Kinematic Inversion of Functionally-Redundant Serial Manipulators: Application to Arc-Welding

Liguo Huo and Luc Baron
Department of Mechanical Engineering
École Polytechnique de Montréal
P.O. 6079, station Centre-Ville
Montréal, Québec, Canada, H3C 3A7
liguo.huo@polymtl.ca, luc.baron@polymtl.ca

Abstract
This paper introduces the concept of functional redundancy of serial manipulators, and presents a new resolution scheme to solve such redundant robotic tasks requiring less than six degrees-of-freedom. Instead of projecting the secondary task onto the null space of the Jacobian matrix in order to take advantage of the redundancy, the twist of end-effector is directly decomposed into two orthogonal subspaces where the main and secondary tasks lie, respectively. The algorithm has shown to be computationally efficient and well suited to solve functionally-redundant robotic tasks, such as arc-welding.

Inversion cinématique des manipulateurs séries fonctionnellement redondants: application au soudage à l’arc

Résumé
Cet article introduit le concept de redondance fonctionnelle des manipulateurs séries, et propose une nouvelle approche de résolution de la redondance pour de telles tâches robotiques exigeant moins de six degré-de-liberté. Cette approche est basée sur la décomposition du torseur de vitesse dans deux sous-espaces cartésiens. Au lieu de projeter la solution sur le noyau de la matrice jacobienne comme le font la majorité des autres approches, le torseur de vitesse associé à l'effecteur est décomposé en deux sous-espaces appropriés, l'un est le sous-espace de la tâche principale, l'autre celui de la tâche redondante. L'algorithme est numériquement efficace et bien adapté à la résolution des tâches robotiques fonctionnellement redondants tels que le soudage à l’arc.
1 INTRODUCTION

Since the late sixties, the control of serial robotic manipulators has received great attentions from the robotic research community. The earliest work is probably the one from Pieper [1], who proposed a scheme based on the Newton-Gauss method in order to iteratively converge toward the desired position and orientation, namely pose, of the end-effector (EE). At each iteration, a small displacement in joint space is computed from the inverse of a Jacobian matrix times the desired EE displacement. Whitney [2, 3] proposed to replace the differential form of the Jacobian matrix of Pieper by a more convenient form based on the translational and angular velocity vectors associated to the EE, which resulted into the well-known resolved-motion rate method. Since the inverse of the Jacobian matrix is required, many research works have been conducted on the conditions for obtaining non-singular Jacobian matrices, including the use of manipulators having more than six degrees-of-freedom in order to cope with singularities. Liégeois [4] was the first to propose a method to take advantage of the kinematic redundancy by using the generalized inverse together with the projection onto the null space of the Jacobian matrix of an arbitrary vector chosen as the gradient of an objective function. Nakamura [5, 6] analyzed the kinematic redundancy of manipulators by the use of matrix theory. Klein and Huang [7] reviewed the algorithms for computing the generalized-inverse for redundant manipulators. Angeles et al. [8] introduced an approach-descent algorithm to solve the inverse kinematics of redundant manipulators. Many research works have considered the computational expense, the roundoff-error amplification and different ways to take advantage of the kinematic redundancy. More recently, Arenson et al. [9] proposed a redundancy-resolution scheme that avoid the squaring of roundoff errors while projecting the secondary task onto the null space of the Jacobian. In all these research works, the kinematic redundancy is studied disregarding the task to be performed, and hence, the redundancy comes from the kinematics of the manipulator itself.

In this paper, the sources of kinematic redundancy of a pair of manipulator-task are characterized into two groups, namely, the intrinsic redundancy and the functional redundancy. For the case of functional redundancy, an approach based on the orthogonal decomposition of the twist is proposed in order to take advantage of the redundancy without having to project the secondary task onto the null space of the Jacobian matrix, thus avoiding roundoff-error amplification and superfluous computations.

2 BACKGROUND ON KINEMATIC INVERSION

Before introducing the redundancy-resolution algorithm, a brief review of the kinematic redundancy of a manipulator with respect to a given task and the kinematic inversion of serial manipulators in these contexts is provided.

2.1 Intrinsic And Functional Redundancy

Let $\mathcal{J}$ denote, the joint space of a robotic manipulator having $n+1$ rigid bodies serially connected by $n$ joints, either revolute $R$ or prismatic $P$. The posture of the manipulator in $\mathcal{J}$ is given by the $n$-dimensional vector, namely $\mathbf{\theta}$, and hence, $n = \dim(\mathcal{J}) = \dim(\mathbf{\theta})$.

Moreover, let $\mathcal{O}$ denote, the operational space of the EE of the robotic manipulator resulting from the joint space $\mathcal{J}$. Since any free-moving rigid body in space can have at most six degrees-of-freedom (DOFs), the dimension of $\mathcal{O}$ is also at most six, and hence, $o = \dim(\mathcal{O}) \leq 6$.

Furthermore, let $\mathcal{T}$ denote, the task space such as required by the functional mobility of
the EE, independently of the manipulator’s architecture and hence, \( t = \dim(T) \leq 6 \). Now, let us introduce the three following definitions:

**Definition 2.1: Intrinsic redundancy**

A serial manipulator is said to be intrinsically redundant when the dimension of the joint space \( J \), denoted by \( n = \dim(J) \), is greater than the dimension of the resulting operational space \( O \) of the EE, denoted by \( o = \dim(O) \leq 6 \), i.e., when \( n > o \). The degree of intrinsic redundancy of a serial manipulator, namely \( r_I \), is computed as

\[
r_I = n - o.
\] (1)

**Definition 2.2: Functional redundancy**

A pair of serial manipulator-task is said to be functionally redundant when the dimension of the operational space \( O \) of the EE, denoted by \( o = \dim(O) \leq 6 \), is greater than the dimension of the task space \( T \) of the EE, denoted by \( t = \dim(T) \leq 6 \), while the task space being totally included into the operation space of the manipulator, i.e., \( T \subseteq O \), and hence, \( o > t \). The degree of functional redundancy of a pair of serial manipulator-task, namely \( r_F \), is computed as

\[
r_F = o - t.
\] (2)

**Definition 2.3: Kinematic redundancy**

A pair of serial manipulator-task is said to be kinematically redundant when the dimension of the joint space \( J \), denoted by \( n = \dim(J) \), is greater than the dimension of the task space \( T \) of the EE, denoted by \( t = \dim(T) \leq 6 \), while the task space being totally included into the resulting operational space of the manipulator, i.e., \( T \subseteq O \), and hence, \( n > t \). The degree of kinematic redundancy of a pair of serial manipulator-task, namely \( r_K \), is computed as

\[
r_K = n - t.
\] (3)

Upon substitution of eqs. (1) and (2) into (3), it becomes apparent that the kinematic redundancy come from two sources, the intrinsic and functional redundancies, i.e.,

\[
r_K = r_I + r_F.
\] (4)

**Example 2.1:**

Let us consider a \( P-R-R-R-R \) serial manipulator as shown in Fig. 1. For this manipulator, \( J \) is of dimension five, i.e., \( \dim(J) = n = 5 \), but the resulting \( O \) is only of dimension four (positioning a point of the EE in 3D space and orienting the EE around an axis only), i.e., \( \dim(O) = o = 4 \), and hence, the manipulator has a degree of intrinsic redundancy of one, i.e., \( r_I = n - o = 5 - 4 = 1 \). For a positioning task in 3D space disregarding the orientation of the EE, \( T \) is only of dimension three, i.e., \( \dim(T) = t = 3 \), and hence, the degree of functional redundancy of the pair of manipulator-task is one, i.e., \( r_F = o - t = 4 - 3 = 1 \). Finally, the kinematic redundancy of this pair of manipulator-task is two, because \( r_K = r_I + r_F = 1 + 1 = 2 \).

In the literature, most of the research works focusing on redundancy-resolution of serial manipulators suppose that \( r_F = 0 \), and thus, study \( r_K = r_I \). In this paper, the opposite case is studied, i.e., supposing \( r_I = 0 \), and thus, study \( r_K = r_F \).

---

1 This definition of functional redundancy of serial manipulator-task is expandable to other types of manipulators such as parallel and hybrid ones.
2.2 Non-Redundant Manipulators

For quick reference, the resolved motion-rate method [2, 3], that is commonly used to iteratively solve the inverse kinematics of serial manipulators, is briefly herein. The method is based on the relationship between the EE velocity, called twist and denoted $\mathbf{t}$, and the joint velocities, denoted $\dot{\mathbf{\theta}}$, given by

$$
\mathbf{t} = J \dot{\mathbf{\theta}},
$$

(5)

with $\mathbf{t}$ and $\dot{\mathbf{\theta}}$ defined as

$$
\mathbf{t} \equiv [\omega^T \mathbf{p}^T]^T \in \mathbb{R}^3, \quad \dot{\mathbf{\theta}} \equiv [\dot{\theta}_1 \cdots \dot{\theta}_n]^T \in \mathbb{R}^n, \quad J \equiv \begin{bmatrix} \mathbf{A} \\ \mathbf{B} \end{bmatrix}, \quad \mathbf{A}, \mathbf{B} \in \mathbb{R}^{3 \times n},
$$

(6)

where $\omega \in \mathbb{R}^3$ is the angular velocity vector of the EE, $\dot{\mathbf{p}} \in \mathbb{R}^3$ is the translation velocity vector of a point on the EE, while $\dot{\theta}_i$ is the velocity of joint $i$ of the manipulator. It is noteworthy that $\mathbf{t}$ is not defined as a vector of $\mathbb{R}^6$, but rather as a set of two vectors of $\mathbb{R}^3$ casted into a column array, and hence, $\mathbf{t} \in 2 \times \mathbb{R}^3 \neq \mathbb{R}^6$. This distinction will be further used in section 3.

Upon substituting the finite displacement $\Delta \mathbf{t}$ of a small time interval (see for example [10]) into eq. (5), the finite displacement $\Delta \mathbf{\theta}$ in $\mathcal{J}$ can be computed as

$$
\Delta \mathbf{\theta} = J^{-1} \Delta \mathbf{t},
$$

(7)

where it is apparent that $J$ must be square and non-singular.

2.3 Intrinsically-Redundant Manipulators

For intrinsically-redundant serial manipulators, $J$ always has more columns than rows, and hence, eq. (5) becomes an under-determined linear algebraic system having infinitely
many solutions. In this case, Liégeois [4] proposed to compute the finite displacement $\Delta \theta$ of eq. (7) as
\[
\Delta \theta = (J^\dagger) \Delta t + (1 - J^\dagger J) h,
\]
where $J^\dagger$ is defined as the right-generalized inverse of $J$ such that
\[
J^\dagger \equiv J^T (JJ^T)^{-1},
\]
and $h$ is an arbitrary vector of $\mathcal{J}$ allowing to satisfy a secondary task. The first term in the right-hand side (RHS) of eq. (8) is known as the minimum-norm solution of eq. (5), i.e., the $\Delta \theta_M$ that minimizes $\|\Delta \theta\|$ among all the $\Delta \theta$ that are solutions of eq. (5). The second part of the RHS of eq. (8) is known as the homogeneous solution of eq. (5), i.e., the $\Delta \theta_H$ that produce $\Delta t = 0$, i.e., no displacement of the EE. This joint displacement $\Delta \theta_H$ is also known as the self-motion of the manipulator. It is symbolically computed as the projection of an arbitrary vector $h$ onto the nullspace of $J$ with the orthogonal projector $(1 - J^\dagger J)$. Equation (8) is used to solve the kinematic inversion of intrinsically-redundant manipulators by many researchers (e.g., Siciliano [11], Arenson et al. [9]), including some who had put a special attention in avoiding the squaring of the condition number while solving eq. (8).

2.4 Functionally-Redundant Manipulators

The motion of the EE usually required by a task is the full 6-DOF. However, many industrial tasks such as arc-welding, milling, deburing, laser-cutting, and many others, require less than 6-DOF, because of the presence of a symmetry axis or plane on the EE. For example, the general task of arc-welding requires 3-DOF for the displacement of the end-point of the electrode, but requires only 2-DOF for its orientation. The rotation of the welding-gun around the electrode axis is clearly irrelevant to the view of the task to be accomplished. In order to cope with this problem, Baron [12] proposed to add a virtual joint around the symmetry axis of the electrode, in order to transform the functional redundancy into an intrinsic redundancy thereby solving an augmented Jacobian matrix with eq. (8). However, this augmented approach to solve functionally-redundant robotic tasks suffers from the potential ill-conditioning of $J$ and the additional computational cost required to solve an augmented $J$. Below, a projected approach to solve the same problem is proposed.

3 KINEMATIC INVERSION OF FUNCTIONALLY-REDUNDANT MANIPULATORS

After introducing the orthogonal decomposition of vectors and twists, the inverse kinematics of functionally-redundant manipulators is formulated by projecting the velocity relationship onto the instantaneous-task subspace, thereby producing the so-called projected approach.

3.1 Orthogonal-Decomposition of Three Dimensional Vectors

Decomposing any vector $(\cdot)$ of $\mathbb{R}^3$ into two orthogonal parts, $[\cdot]_M$, the component lying on the subspace, $\mathcal{M}$, and $[\cdot]_{M^\perp}$, the component lying in the orthogonal subspace, $\mathcal{M}^\perp$, using the projector $\mathbf{M}$ and an orthogonal complement of $\mathbf{M}$, namely $\mathbf{M}^\perp$, as follows:
\[
(\cdot) = [\cdot]_M + [\cdot]_{M^\perp} = \mathbf{M}(\cdot) + \mathbf{M}^\perp(\cdot) = (\mathbf{M} + \mathbf{M}^\perp)(\cdot)
\]
It is apparent from eq. (10), that $M$ and $M^\perp$ are related by $M + M^\perp = 1$ and $MM^\perp = O$, where 1 and O are the $3 \times 3$ identity and zero matrices, respectively. The orthogonal complement of $M$ thus defined, $M^\perp$, is therefore unique, and hence, both $M$ and $M^\perp$ are projectors that verify the following properties:

- Symmetry: $[M]^T = M$, $[M^\perp]^T = M^\perp$
- Idempotency: $[M]^2 = M$, $[M^\perp]^2 = M^\perp$
- Rank-complementarity: $\text{rank}(M) + \text{rank}(M^\perp) = 3$
- Subspace-complementarity: $M \oplus M^\perp = \mathbb{R}^3$

The projector $M$ projects vectors of $\mathbb{R}^3$ onto the subspace $\mathcal{M}$, while the orthogonal projector $M^\perp$ projects those vectors onto the orthogonal subspace $\mathcal{M}^\perp$. These projectors are given for the four possible dimensions $i$ of subspaces of $\mathbb{R}^3$ as:

$$M_i = \begin{cases} 1 & i = 3 \Rightarrow 3\text{-D task} \\ P & i = 2 \Rightarrow 2\text{-D task} \\ L & i = 1 \Rightarrow 1\text{-D task} \\ O & i = 0 \Rightarrow 0\text{-D task} \end{cases}, \quad M_i^\perp = \begin{cases} O & i = 3 \Rightarrow 3\text{-D task} \\ L & i = 2 \Rightarrow 2\text{-D task} \\ P & i = 1 \Rightarrow 1\text{-D task} \\ O & i = 0 \Rightarrow 0\text{-D task} \end{cases}$$

(11)

where the plane and line projectors, $P$ and $L$, respectively, are defined as:

$$P \equiv 1 - L, \quad L \equiv ee^T,$$

(12)

in which $e$ is a unit vector along the line $L$ and normal to the plane $P$. The null-projector $O$ is the $3 \times 3$ zero matrix that projects any vector of $\mathbb{R}^3$ onto the null-subspace $O$, while the identity-projector 1 is the $3 \times 3$ identity matrix that projects any vector of $\mathbb{R}^3$ onto itself.

### 3.2 Orthogonal-Decomposition of Twists

Any twist array $(\cdot)$ of $2 \times \mathbb{R}^3$ can also be decomposed into two orthogonal parts, $[\cdot]_T$, the component lying on the task subspace, $T$, and $[\cdot]_{T^\perp}$, the component lying in the orthogonal task subspace (also designated as the redundant subspace), $T^\perp$, using the twist projector $T$ and an orthogonal complement of $T$, namely $T^\perp$, as follows:

$$[\cdot] = [\cdot]_T + [\cdot]_{T^\perp} = T(\cdot) + T^\perp(\cdot) = (T + T^\perp)(\cdot) \quad (13)$$

It is apparent from eq. (13), that $T$ and $T^\perp$ are projectors of twists that must verify all the properties of projectors of section 3.1. However, twists are not vectors of $\mathbb{R}^6$, and hence, projectors of twists cannot be defined as in eqs. (11) and (12), e.g.,

$$T \neq tt^T, \quad T^\perp \neq 1 - tt^T,$$

(14)

but must rather be defined as block diagonal matrices of projectors of $\mathbb{R}^3$, i.e.,

$$T \equiv \begin{bmatrix} M_\omega & O \\ O & M_\nu \end{bmatrix}, \quad T^\perp \equiv 1 - T = \begin{bmatrix} 1 - M_\omega & O \\ O & 1 - M_\nu \end{bmatrix},$$

(15)

where $M_\omega$ and $M_\nu$ are projectors of $\mathbb{R}^3$ defined in eqs. (11) and (12) which allow the projection of the angular and translational velocity vectors, respectively. It is noteworthy that the matrices of eq. (14) do not verify the properties of projectors, and hence, cannot be used for orthogonal decomposition. Finally, eq. (13) becomes

$$t = t_T + t_{T^\perp} = Tt + (1 - T)t.$$

(16)
3.3 Twist Decomposition Algorithm in Solving Functional Redundancy

For functionally-redundant serial manipulators, it is possible to decompose the twist of the EE into two orthogonal parts, one lying into task subspace and another one lying into the redundant subspace. Substituting eq. (16) into eq. (7) yields

\[
\Delta \theta = \begin{pmatrix}
\Delta t \\
\end{pmatrix}_{\text{task displacement}} + \begin{pmatrix}
J^i (1 - T) J h \\
\end{pmatrix}_{\text{redundant displacement}},
\]

where \( h \) is an arbitrary vector of \( J \) allowing to satisfy a secondary task. Vector \( h \) is often chosen as the gradient of an objective function to minimize (Baron [12]). For the avoidance of joint-limits, the objective function \( z \) can be written as to maintain the manipulator as close as possible to the mid-joint position \( \hat{\theta} \), i.e.,

\[
z = \frac{1}{2} (\theta - \hat{\theta})^T W^T W (\theta - \hat{\theta}) \rightarrow \min_{\theta},
\]

with \( \hat{\theta} \) and \( W \) being defined as

\[
\hat{\theta} \equiv \frac{1}{2} (\theta_{max} + \theta_{min}), \quad W \equiv \text{diag} (\theta_{max} - \theta_{min}).
\]

Vector \( h \) is thus chosen as minus the gradient of \( z \), i.e.,

\[
h = -\nabla z.
\]

The first part of the RHS of eq. (17) is the joint displacement required by the task, while the second part is the joint displacement in the redundant subspace (or irrelevant to the task). Clearly, eq. (17) does not require the projection onto the null-space of \( J \) as most of the redundancy-resolution algorithms do, but rather requires an orthogonal projection based on the instantaneous geometry of the task to be accomplished.

---

**Algorithm 3.1: Twist Decomposition Algorithm**

1. \( \theta \leftarrow \) initial joint position;
2. \( \{p_d, Q_d\} \leftarrow \) desired EE position and orientation;
3. \( \{p, Q\} \leftarrow \text{DK}(\theta) \)
4. \( \Delta Q \leftarrow Q^r Q_d \)
5. \( \Delta p \leftarrow p_d - p \)
6. \( \Delta t \leftarrow \begin{bmatrix}
Q \text{vect}(\Delta Q) \\
\Delta p
\end{bmatrix} \)
7. \( \text{DK}(\theta) \Rightarrow \begin{cases}
e \Rightarrow M_\omega \\
f \Rightarrow M_v
\end{cases}, \quad J \)
8. \( T \leftarrow \begin{bmatrix}
M_\omega & O \\
O & M_v
\end{bmatrix}, \)
9. \( \Delta \theta \leftarrow J^i T \Delta t + \bar{J}^i (1 - T) J h \)
10. if \( \|\Delta \theta\| < \epsilon \) then stop; else
11. \( \theta \leftarrow \theta + \Delta \theta \), and go to 3.
As shown in Algorithm 3.1, eq. (17) is used within a resolved-motion rate method. At lines 1-3, the joint position $\theta$ and the desired EE pose $\{p_d, Q_d\}$ are first initialized, then the actual EE pose $\{p, Q\}$ is computed with the direct kinematic model $\text{DK}(\theta)$. At lines 4-6, an EE displacement $\Delta t$ is computed from the difference between the desired and actual EE poses. The $\text{vect}(\cdot)$ at line 6 is the function transforming a $3 \times 3$ rotation matrix into an axial vector as defined in [10] (page 34). At lines 7-8, the instantaneous orthogonal twist projector $T$ is computed from $\text{DK}(\cdot)$. At line 9, the orthogonal decomposition method is used to compute the corresponding joint displacement $\Delta \theta$. Finally, the algorithm is stopped whenever the norm of $\Delta \theta$ is smaller than a certain threshold $\epsilon$.

4 APPLICATION TO ARC-WELDING

When performing arc-welding operations, the electrode of the welding tool has an axis of symmetry around which the welding tool may be rotated without interfering with the task to be performed. This axis describes the geometry of the functional redundancy (or the redundant subspace of twists). The unit vector $e$ denotes the orientation of the axis of symmetry along the electrode. The projection of $\omega$ along $e$ is the irrelevant component of $\omega$, while its projection onto the plane normal to $e$ is the relevant component of $\omega$. For a general arc-welding task around the electrode axis $e$, the twist projector is defined as

$$
T_{\text{weld}} \equiv \begin{bmatrix} (1 - ee^T) & 0 \\ 0 & 1 \end{bmatrix}, \quad T_{\text{weld}}^+ \equiv \begin{bmatrix} ee^T & 0 \\ 0 & 0 \end{bmatrix},
$$

(21)

Now, substituting eq. (21) into eq. (17) yields

$$
\Delta \theta = J^T T_{\text{weld}} \Delta t + J^T \begin{bmatrix} ee^T Ah \\ 0 \end{bmatrix},
$$

(22)

where $A$ is the upper part of $J$ as defined in eq. (6). Equation (22) can be used as line 9 of Algorithm 3.1 in order to solve the inverse kinematics of serial manipulators while performing a general arc-welding task.

As shown in Fig. 2, a PUMA 500 serial manipulator is used to perform a pipe-to-bridge welding task. Its Denavit-Hartenberg parameters are described in Table 1. The welding tool has a transformation matrix $A_{\text{tool}}$ as

$$
A_{\text{tool}} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos(\frac{\pi}{6}) & -\sin(\frac{\pi}{6}) & 0.1 \\ 0 & \sin(\frac{\pi}{6}) & \cos(\frac{\pi}{6}) & 0.501 \\ 0 & 0 & 0 & 1 \end{bmatrix}.
$$

(23)

The EE must perform consecutively the trajectory $\Lambda$ ($T = 285$ sec.), i.e.,

$$
p = \begin{bmatrix} 0.1 \cos(\omega t) \\ 0.6 + 0.1 \sin(\omega t) \\ -0.59 \end{bmatrix}, \quad Q = \begin{bmatrix} \cos \alpha & -\sin \alpha \cos \beta & \sin \alpha \sin \beta \\ \sin \alpha & \cos \alpha \cos \beta & \cos \alpha \sin \beta \\ 0 & \sin \beta & \cos \beta \end{bmatrix},
$$

(24)

with $\alpha = \frac{\pi}{2} + \omega t$, $\beta = \frac{-3\pi}{4}$, $\omega = \frac{2\pi}{T}$. $0 \leq t \leq T$, where distances and angles are expressed in meter and radians, respectively. The orientation of the electrode axis, namely $e$, can be computed as

$$
e = Q_1 Q_2 \cdots Q_6 k, \quad k \equiv [0 \ 0 \ 1]^T.
$$

(25)
The secondary task $h$ is chosen to avoid the joint-limits such that:

$$h = -W(\theta - \theta_0),$$

where $W$ is a positive-definite weighting matrix as in eq. (19) and $\theta_0$ the mean-joint position of the robotic manipulator, i.e.,

$$\theta_0 \equiv \begin{bmatrix} \pi/2 & -\pi/3 & \pi/4 & \pi/3 & \pi \end{bmatrix}^T.$$

Figure 3 shows the joint positions to perform twice the trajectory $\Lambda$ as computed by the resolved-motion rate method without considering the functional redundancy, i.e., using eq. (7) at line 9 of Algorithm 3.1. Apparently, without taking advantage of the axis of symmetry of the electrode, the manipulator is able to perform the first turn of the trajectory $\Lambda$ while the second consecutive turn is not possible without exceeding the joint limits. Figure 4(a) shows the joint positions to perform twice the trajectory $\Lambda$ as computed by the augmented approach, i.e., using eq. (8) at line 9 of Algorithm 3.1. Apparently, the manipulator is able to perform multiple consecutive turns without exceeding the joint limits. However, excessive joint velocities appear at every turn. Figure 4(b) shows the joint positions for two consecutive turns as computed by the projected approach, i.e., using eq. (17) at line 9 of Algorithm 3.1. Apparently, the manipulator is able to perform multiple consecutive turns without exceeding the joint limits. Excessive joint velocities appear only at the first turn, because of the arbitrarily chosen initial conditions, and not thereafter.

For the sake of comparing the performances of the augmented and projected approaches, the trajectory $\Lambda$ is discretized into $n$ segments, for which the average reaching accuracy of their end-point is computed. For this performance evaluation, lines 3 to 10 of Algorithm 3.1 is performed a constant number of iterations rather than until a threshold is reached. The position error of segment $j$, namely $e_{pj}$, is computed as the norm of the difference between the corresponding reached position $p_{rj}$ and desired position $p_{dj}$, i.e.,

$$e_{pj} = \|p_{rj} - p_{dj}\|.$$
and the average position error $\overline{e}_p$ along the whole trajectory $\Lambda$ is given as

$$\overline{e}_p = \frac{1}{n} \sum_{j=1}^{n} e_{pj}. \quad (29)$$

Similarly, the orientation error of segment $j$, namely $e_{ej}$, is computed as the norm of difference between the corresponding reached orientation $Q_{rj}$ and the desired orientation $Q_{dj}$, i.e.,

$$e_{ej} = \|\text{vect}(Q_{rj}^T Q_{dj})\|. \quad (30)$$

Since the rotation around the electrode axis does not affect the welding task, the orientation error around that axis is also irrelevant, and thus, only its projection onto the plane normal to that axis is meaningful. Let the unit vector $e$ be aligned along the electrode axis in the base frame, then the projection of the orientation error of eq. (30) onto the plane normal to $e$ is given as

$$e_{oj} = \|(1 - ee^T)Q_{rj} \text{vect}(Q_{rj}^T Q_{dj})\|, \quad (31)$$

and the average orientation error $\overline{e}_o$ along the whole trajectory $\Lambda$ is given as

$$\overline{e}_o = \frac{1}{n} \sum_{j=1}^{n} e_{oj}. \quad (32)$$

Table 2 shows the average position and orientation errors of the augmented and projected approaches computed with 20 iterations for each segment. It is apparent that the projected approach has much lower position and orientation errors in the task space than the augmented approach. The projected approach produces more accurate solutions than the augmented approach with the same number of iterations. In other words, the projected approach is able to approach the desired posture faster than the augmented approach.
5 CONCLUSIONS

In this paper, the concept of functional redundancy is defined and discussed. The kinematic inversion of functionally-redundant serial manipulators is formulated using the orthogonal decomposition of the twist of the EE into a task subspace and a redundant subspace. The numerical simulation of the arc-welding of a pipe-to-bride with the PUMA 500 serial manipulator has shown that the projected approach is more effective relative to the augmented and non-redundant approaches. The projected approach is applicable to other redundant tasks requiring less than six DOF.

6 ACKNOWLEDGEMENTS

The financial support from the Natural Sciences and Engineering Research Council of Canada under grant OGPIN-203618 and the research contract CDT-P3127 are gracefully acknowledged.

References


<table>
<thead>
<tr>
<th>joint</th>
<th>$\theta_i$</th>
<th>$a_i$</th>
<th>$b_i$</th>
<th>$\alpha_i$</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>$\theta_1$</td>
<td>0.0</td>
<td>0.0</td>
<td>$-\pi/2$</td>
</tr>
<tr>
<td>2</td>
<td>$\theta_2$</td>
<td>0.4318</td>
<td>0.0</td>
<td>0.0</td>
</tr>
<tr>
<td>3</td>
<td>$\theta_3$</td>
<td>-0.0203</td>
<td>0.1491</td>
<td>$\pi/2$</td>
</tr>
<tr>
<td>4</td>
<td>$\theta_4$</td>
<td>0.0</td>
<td>0.4330</td>
<td>$-\pi/2$</td>
</tr>
<tr>
<td>5</td>
<td>$\theta_5$</td>
<td>0.0</td>
<td>0.0</td>
<td>$\pi/2$</td>
</tr>
<tr>
<td>6</td>
<td>$\theta_6$</td>
<td>0</td>
<td>0.055</td>
<td>0</td>
</tr>
<tr>
<td>unit</td>
<td>rad.</td>
<td>m</td>
<td>m</td>
<td>rad.</td>
</tr>
</tbody>
</table>

Table 1: DH parameters of PUMA 500

<table>
<thead>
<tr>
<th>method</th>
<th>$\tau_p$</th>
<th>$\tau_o$</th>
</tr>
</thead>
<tbody>
<tr>
<td>augmented</td>
<td>0.0789</td>
<td>2.7403 x 10^{-5}</td>
</tr>
<tr>
<td>projected</td>
<td>1.0533 x 10^{-6}</td>
<td>1.0123 x 10^{-6}</td>
</tr>
</tbody>
</table>

Table 2: Errors of the augmented and projected approaches


