Robotica (2026), **44**, pp. 2336–2351
doi:10.1017/S0263574726103956

CAMBRIDGE UNIVERSITY PRESS

RESEARCH ARTICLE

Oriented maximum velocity evaluation in serial robots using translational and rotational decomposition

Kai Wu[1], Hengyi An[1] and Mingfeng Wang[2]

[1]SHIEN-MING WU School of Intelligent Engineering, South China University of Technology, Guangzhou, 510000, China
[2]Department of Engineering, Brunel University of London, London, UB83PH, UK
Corresponding author: Mingfeng Wang; Email: mingfeng.wang@brunel.ac.uk

Received: 28 July 2025 UTC; **Revised:** 23 July 2026 UTC; **Accepted:** 19 August 2026 UTC;
First published online: 1 October 2026 UTC

Keywords: serial robots; kinematic performance evaluation; oriented maximum velocity

Abstract

The reachable velocity of a serial robot within its workspace is crucial for planning complex robotic tasks. Traditional methods often struggle to accurately assess the maximum reachable velocity of both redundant and nonredundant serial robots in directions involving coupled translational and rotational motion. This paper proposes a method for evaluating the maximum oriented velocity of serial robots. The approach separates the motion into pure translational and pure rotational components. It converts the summation of joint velocities that satisfy only translational and rotational velocity constraints into an objective function. Consequently, solving the robot's maximum oriented velocity is transformed into a numerical optimization problem using linear programming. The proposed method is applicable to both nonredundant and redundant serial robots. A comparative experiment demonstrates that the proposed method exhibits high accuracy. Especially in the redundant serial robots, the proposed method has an average error of 0.67%, while the existing decomposed twist feasibility method has an average error of 12.56%.

1. Introduction

In industrial scenarios, motion along a specified Cartesian path is one of the most common tasks performed by robots. To minimize the cycle time, the robot trajectory must be executed at the highest feasible speed. For time-optimal path planning, the objective is to enable the robot to move as fast as possible without violating any constraints. To achieve this, accurately evaluating the maximum reachable velocity of the end-effector in any direction at each configuration is essential [1]. It also helps in finding the best joint configuration and velocity during task execution, solving for the optimal task deployment strategy, and enhancing the robot's operational flexibility within its workspace [2, 3]. For example, in multi-robot collaborative tasks, the motion velocities of multiple robots must be coordinated to ensure the successful completion of the task. In a teleoperated robotic system, the slave robot must synchronize its motion speed with the master device to ensure good coordination and compliance.

Evaluation methods and indices of robot velocity performance are effective approaches for assessing the reachable velocity of robots. Klein et al. [4]. found that the minimum singular value of the Jacobian matrix undergoes more drastic changes near singular configurations compared to other singular values. This trend suggests that the minimum singular value can be used as a metric to evaluate robot velocity performance. Salisbury et al. [5] defined the product of the norm of the Jacobian matrix and the norm of its inverse as the Condition Number, which is used to measure the robot's dexterous motion capability. However, these metrics only reflect the robot's behavior near singular configurations, and there is no explicit mapping between the results and the robot's actual reachable velocity. Dubey et al. [6] proposed measuring a robot's ability to move in a given direction by the ratio of the end-effector velocity vector norm to the joint velocity vector norm. However, these indices are based on the Jacobian matrix and do not consider the robot's actual kinematic constraints. Considering actual joint velocity constraints, the

© The Author(s), 2026. Published by Cambridge University Press. This is an Open Access article, distributed under the terms of the Creative Commons Attribution licence (https://creativecommons.org/licenses/by/4.0/), which permits unrestricted re-use, distribution and reproduction, provided the original article is properly cited.

https://doi.org/10.1017/S0263574726103956 Published online by Cambridge University Press

ratio of the minimum to the maximum value of the robot's maximum reachable velocity in any direction within its workspace is defined as the Isotropy Index. However, the computed values of the above indices do not have practical physical significance.

Yoshikawa [7] introduced the concept of manipulability to evaluate a robot's operational capability. Building on this, Yoshikawa [8] further proposed the concept of the velocity ellipsoid, dividing it into the translational velocity ellipsoid and rotational velocity ellipsoid to separately assess a robot's reachable translational and rotational velocities. Since the velocity ellipsoid constraints joint velocities using the Euclidean norm, its calculation only provides an approximation of the robot's actual reachable velocity. When controlling a robot to perform tasks at a specified velocity in a given direction, these performance indices cannot be used as reliable references to ensure task success.

The velocity polytope maps the actual joint velocity constraints into the workspace through the Jacobian matrix, representing the robot's reachable velocity in any direction, and its calculation is more accurate [9–13]. However, the velocity polytope is difficult to apply due to geometric dimensionality limitations when the motion direction exceeds three dimensions. Moreover, the maximum feasible velocity of a robot along a prescribed path depends not only on the joint velocity limits, but also on the available joint accelerations and actuator torque capabilities at the current state [14, 15]. In the case of separating translation and rotation, the vector expansion method [16] and the kinematic directional index [17–19] can effectively evaluate the maximum velocity of the robot in a specified direction. Since rotational and translational velocities have inconsistent physical units, these methods cannot evaluate the maximum velocity in coupled translational and rotational directions. Moreover, when evaluating the maximum translational (rotational) velocity, it is impossible to maintain control over the rotational (translational) velocity. Additionally, the kinematic directional index cannot be applied to both nonredundant and redundant robots. To overcome the issue of physical unit inconsistency, Mansour et al. [20, 21] proposed the power index, which unifies the measurement of rotational and translational motion performance to evaluate a robot's motion capability. However, the power index does not have a clear correlation with the reachable velocity requirements of actual tasks. Stradovnik [22] introduced the decomposed twist feasibility (DTF) method to solve for the maximum reachable velocity of nonredundant serial robots in a specified direction. This method is not applicable for solving the maximum velocity in redundant cases.

Therefore, in this paper, to evaluate the maximum achievable velocity of a serial robot's end-effector in a specific joint configuration and a designated direction, that is, the oriented maximum reachable velocity, we propose a novel method based on the decomposition of translational and rotational motions.This method accurately assesses the oriented maximum velocity of both nonredundant and redundant serial robots in specified coupled translational and rotational directions. The approach decomposes the translational and rotational motions, then solves for the corresponding joint velocities and combines them to obtain the oriented maximum velocity of the serial robot, subject to joint velocity constraints. Since this method essentially converts each dimension of the robot's motion in the specified direction into a constraint, it can be applied to cases with varying motion dimensionality. More importantly, the linear-programming formulation establishes a flexible framework for future extensions, in which additional physical constraints, such as joint acceleration and actuator torque limits, may be incorporated into the feasible set. Experimental results based on the nonredundant UR5e robot and the redundant Sawyer robot demonstrate the accuracy of the proposed method.

The remainder of this paper is structured as follows: In Section 2, we analyze the limitations of the traditional method for evaluating the oriented maximum velocity of serial robots, namely, the DTF method. In Section 3, we propose an evaluation method based on the decomposition of translational and rotational motions. In Section 4, we design comparative experiments to verify the accuracy of the proposed method. Finally, conclusions are presented in Section 5.

2. Related works

2.1. Inverse kinematics (IK) for redundant manipulators

The main methods for solving the inverse kinematics problem of redundant robots can be broadly classified into three categories: joint-velocity-based methods, joint-position-based methods, and AI-based

https://doi.org/10.1017/S0263574726103956 Published online by Cambridge University Press

methods [23]. Among them, the most representative velocity-based method is the gradient projection method. Colomé [24] provided a detailed overview of research on the gradient projection method, such as minimizing joint-torque energy [25] and enhancing dexterity [26], including joint-torque energy minimization and dexterity enhancement. JongHo Kim [27] reformulated redundant inverse kinematics as a gradient-based constrained optimization problem, numerically solved using sequential quadratic programming. AI-based methods establish the pose-to-joint mapping relationship through offline training, which can effectively avoid the computational burden of real-time calculation. However, their performance is limited by sample dependency and model generalization capability.

2.2. Decomposition of oriented maximum velocity

In serial robots, the relationship between the robot's joint velocity and workspace velocity can be expressed as:

$$\dot{X} = \begin{bmatrix} \dot{x}_T \\ \dot{x}_R \end{bmatrix} = \begin{bmatrix} J_T \\ J_R \end{bmatrix} \dot{\theta} = J\dot{\theta} \tag{1}$$

where $\dot{\theta} \in \mathbb{R}^{n\times 1}$ is the joint velocity vector, n is the number of joints, and $\dot{X} \in \mathbb{R}^{m\times 1}$ is an m-dimensional column vector composed of the translational velocity $\dot{x}_T$ and the rotational velocity $\dot{x}_R$ of the robot's end effector. The value of m depends on the degrees of freedom of the robot's end-effector workspace. $J \in \mathbb{R}^{m\times n}$ is the Jacobian matrix composed of the translational Jacobian J_T and the rotational Jacobian J_R. Without loss of generality, we assume that the robot's end-effector workspace is 6-dimensional, i.e., $m = 6$, in the following discussions.

The robot's joint velocity vector is constrained by the maximum velocity of each joint, expressed by the infinity norm:

$$\left\|\dot{\theta}\right\|_\infty \le 1 \tag{2}$$

The joint velocity vector of the robot can be described as the sum of the joint velocity vector that generates only translational velocity $\dot{\theta}_T$ in the workspace and the joint velocity vector that generates only rotational velocity $\dot{\theta}_R$ in the workspace, as shown in

$$\dot{\theta} = \dot{\theta}_T + \dot{\theta}_R \tag{3}$$

The translational-only joint velocity vector satisfies:

$$J_R\dot{\theta}_T = 0 \tag{4}$$

Similarly, the rotational-only joint velocity vector satisfies:

$$J_T\dot{\theta}_R = 0 \tag{5}$$

2.3. Solving maximum velocity using the DTF method

After the velocity decomposition in Eq. (3), the DTF method is the current method used to solve for the maximum velocity. This method maps the end-effector velocity to the joint velocity using the generalized inverse matrix and expresses the joint velocity vector under the constraints of each joint's maximum velocity in the form of the infinity norm.

The translation velocity $\hat{\dot{x}}_T$ under the condition of zero rotational velocity is:

$$\hat{\dot{x}}_T = J_T\left(I_n - J_R^+ J_R\right)\dot{\theta} = \tilde{J}_T\dot{\theta} \tag{6}$$

where $J_R^+ \in \mathbb{R}^{n\times 3}$ is the pseudo Moore–Penrose inverse of the rotational Jacobian matrix J_R, and $\tilde{J}_T$ represents the augmented translational Jacobian matrix under the zero rotational velocity constraint, $I_n$is the n-dimensional identity matrix.

https://doi.org/10.1017/S0263574726103956 Published online by Cambridge University Press

Similarly, the rotational velocity $\dot{\hat{x}}_R$ under the condition of zero translational velocity is:

$$\dot{\hat{x}}_R = J_R\left(I_n - J_T^+ J_T\right)\dot{\theta} = \tilde{J}_R\dot{\theta} \tag{7}$$

where $J_T^+ \in \mathbb{R}^{n\times 3}$ is the pseudo Moore–Penrose inverse of the translational Jacobian matrix J_T, and $\tilde{J}_R$ represents the augmented rotational Jacobian matrix under the zero translational velocity constraint.

From (6) and (7), (3) can be transformed into:

$$\dot{\theta} = \dot{\theta}_T + \dot{\theta}_R = \tilde{J}_T^+\dot{x}_T + \tilde{J}_R^+\dot{x}_R \tag{8}$$

where $\tilde{J}_T^+$ is the pseudo Moore–Penrose inverse of the augmented translational Jacobian matrix $\tilde{J}_T$ considering the constraint of zero rotational velocity, and $\tilde{J}_R^+$is the pseudo Moore–Penrose inverse of the augmented rotational Jacobian matrix $\tilde{J}_R$ considering the constraint of zero translational velocity.

For a redundant manipulator, the solution consists of a particular solution plus a null-space component:

$$\dot{\theta} = J^+\dot{x} + \left(I_n - J^+J\right)z = J^+\dot{x} + n \tag{9}$$

The joint velocities that produce only end-effector translational motion $\dot{\theta}_T$ and only end-effector rotational motion $\dot{\theta}_R$ are expressed as follows:

$$\dot{\theta}_T = \tilde{J}_T^+\dot{x}_T + \alpha n \tag{10}$$

$$\dot{\theta}_R = \tilde{J}_R^+\dot{x}_R + \beta n \tag{11}$$

Here, $\widetilde{J_T}^+\dot{x}_T$ and $\widetilde{J_R}^+\dot{x}_R$ represent the particular solutions for pure translation and pure rotation, respectively. $n \in \mathrm{Null}(J)$ denotes the null-space solution vector, and α and β represent the contributions of the null-space vectors in the translational and rotational directions, respectively. The sum of these contributions is constrained by the joint velocity limits (Eq. 2)

Therefore, Eq. (8) in the case of redundancy is expressed as follows:

$$\dot{\theta} = \left(\dot{\theta}_T + \dot{\theta}_R\right) = \tilde{J}_T^+\dot{x}_T + \tilde{J}_R^+\dot{x}_R + (\alpha+\beta)n \tag{12}$$

Because the DTF method does not address the null-space issue under redundancy, in our subsequent experiments—when computing in MATLAB—the solver effectively considered only the particular solutions, i.e., $\hat{J}_T^+\dot{\mathbf{x}}_T$ and $\hat{J}_R^+\dot{\mathbf{x}}_R$. For the null-space components in (10) and (11), it implicitly set the null-space term to zero during the solution process. This approach yields the minimum ℓ_2-norm solution; it only guarantees, for the current configuration, minimal effort (or joint activity) when moving along the prescribed target direction, and does not necessarily maximize the velocity along that direction.

After discussing the handling of the DTF method in both nonredundant and redundant cases, it is clear that DTF is only applicable to nonredundant cases. Therefore, the following discussion will focus on the nonredundant case.

Let $h \in \mathbb{R}$ denote the relative ratio between the magnitudes of the end-effector's translational and rotational velocities:

$$\tilde{h} = \tilde{v}/\tilde{\omega} \tag{13}$$

Then, the translation velocity vector $\dot{\hat{x}}_T$ and the rotational velocity vector $\dot{\hat{x}}_R$ of the end-effector can be expressed as:

$$\dot{x}_T = \|\dot{x}_T\|_2 \; u_T = \tilde{v}\, u_T \tag{14}$$

$$\dot{x}_R = \|\dot{x}_R\|_2 \; u_R = \tilde{\omega}\, u_R \tag{15}$$

where $\tilde{v}$ is the magnitude of the translational velocity and $\tilde{\omega}$ is the magnitude of the rotational velocity, $u_T, u_R \in \mathbb{R}^{3\times 1}$ are the directional vectors of translational velocity and rotational velocity in the workspace, respectively.

https://doi.org/10.1017/S0263574726103956 Published online by Cambridge University Press

Substituting Eqs. (8), (20), (21), (22) into the constraints of Eq. (2) yields:

$$\left\| \tilde{v}\, J_T^+ u_T + \tilde{\omega}\, J_R^+ u_R \right\|_\infty \leq 1 \tag{16}$$

Thus, the maximum translational velocity $\tilde{v}_{\max}$ and the maximum rotational velocity $\tilde{\omega}_{\max}$ can be expressed as:

$$\begin{aligned} \tilde{v}_{\max} &= \left\| \tilde{J}_T^+ u_T + h^{-1} \tilde{J}_R^+ u_R \right\|_\infty^{-1} \\ \tilde{\omega}_{\max} &= \left\| h\, \tilde{J}_T^+ u_T + \tilde{J}_R^+ u_R \right\|_\infty^{-1} \end{aligned} \tag{17}$$

Therefore, given the motion parameters (u_T, u_R, h) and joint velocity constraints, the maximum velocity in the specified direction at a target configuration can be approximately determined.

However, the DTF method has the following limitations:

- The DTF method computes the ℓ_2-norm solution using the Moore–Penrose pseudoinverse, which generally corresponds to the minimum joint effort (or minimum joint activity) when moving along the prescribed target direction. However, in redundant cases, this criterion does not match the objective of maximizing the end-effector velocity.
- When joint limits are unequal, it is unclear which joint dominates the infinity norm, complicating constraint handling.

3. Proposed algorithm

Unlike the DTF method, this paper proposes a method that accurately solves for the maximum velocity of both nonredundant and redundant serial robots in a specified direction after the velocity decomposition in Equation (3).

3.1. Algorithmic procedure and flowchart

As shown in Figure 1, we take the pure translational motion of the robot as an example to illustrate the detailed implementation procedure of the proposed algorithm. To clearly present the theoretical basis and solution procedure of the proposed method, this section is organized into three parts. First, in 3.2, the joint velocity decomposition formulations for nonredundant and redundant robots are derived, where the joint velocity is decoupled into the component responsible for pure translational motion of the end-effector and the component responsible for pure rotational motion. Then, in 3.3, the joint velocity is mapped into the end-effector workspace through the velocity Jacobian matrix, and this mapping relationship is further expressed as a function of the unit vector of the specified motion direction, the ratio between translational and rotational motions, and the maximum velocity. Finally, in 3.4, the constructed function is further converted into a set of linear programming (LP) equations, thereby enabling the exact calculation of the maximum velocity along the specified direction

3.2. Joint velocity analysis

The null-space solution of the Jacobian matrix J can be expressed as:

$$n = \sum_{i=1}^{n-r(J)} t_i\, \sigma_i \tag{18}$$

The solution vector $\dot{\theta}_T$ that satisfies Eq. (10) can be expressed in terms of the fundamental solution set of the system of equations:

$$\dot{\theta}_T = \tilde{J}_T^+ \dot{x}_T + \alpha \sum_{i=1}^{n-r(J)} t_i\, \sigma_i \tag{19}$$

https://doi.org/10.1017/S0263574726103956 Published online by Cambridge University Press

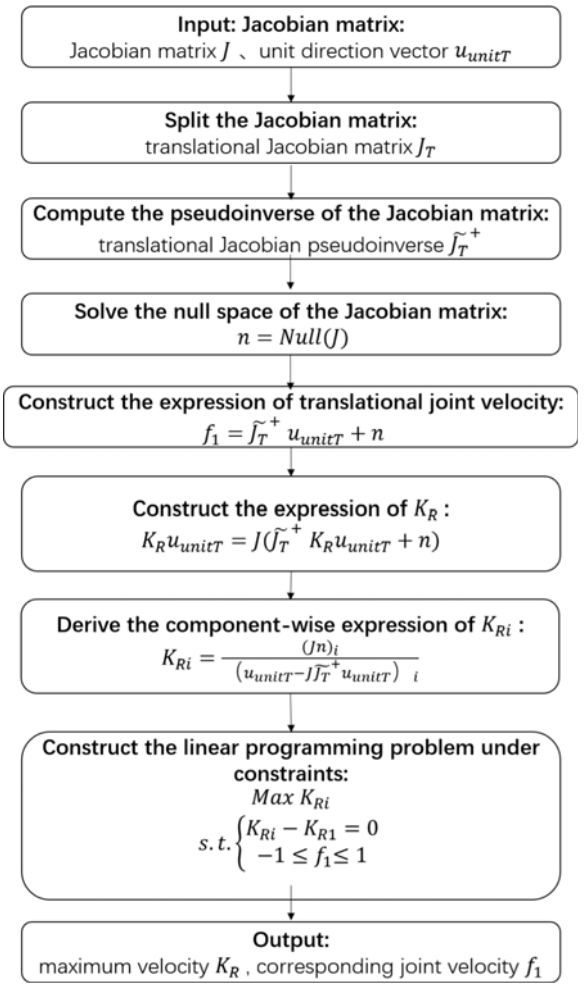


Figure 1. *Algorithm flowchart.*

where σ_i represents the *i-th* column of the null space matrix $N(J)$ of the Jacobian matrix J, with $N(J) \in \mathbb{R}^{n\times(n-r(J))}$, and $r(J)$ denoting the rank of J. t_i is an arbitrary constant, where $i = 1, 2, \cdots, n - r(J)$.

Similarly, the joint velocity vector generating only rotational velocity is:

$$\dot{\theta}_R = \tilde{J}_R^+ \dot{x}_R + \beta \sum_{i=1}^{n-r(J)} t_i \sigma_i \tag{20}$$

From Eq. (12), $\dot{\theta}$ can be simplified as a function of the unknowns t_i, α, and β, as follows:

$$\dot{\theta}(t_i, \alpha, \beta) = \tilde{J}_T^+ \dot{x}_T + \tilde{J}_R^+ \dot{x}_R + (\alpha + \beta) \sum_{i=1}^{n-r(J)} t_i \sigma_i \tag{21}$$

Without loss of generality, we take the case where the maximum joint velocity for each joint is set to ± 1 rad/s as an example. This constraint can be obtained by substituting Eq. (22) into Eq. (2), as follows:

$$\left\| \tilde{J}_T^+ \dot{x}_T + \tilde{J}_R^+ \dot{x}_R + (\alpha + \beta) \sum_{i=1}^{n-r(J)} t_i \sigma_i \right\|_\infty \le 1 \tag{22}$$

In fact, different joint velocity limits can be adjusted in Eq. (35).

https://doi.org/10.1017/S0263574726103956 Published online by Cambridge University Press

3.3. Joint velocity analysis velocity analysis in workspace

The robot's end-effector velocity vector $\dot{X} \in \mathbb{R}^{m\times 1}$ can be decomposed into the translational velocity vector $\dot{x}_T$ and the rotational velocity vector $\dot{x}_R$. Then, $\dot{x}_T$ and $\dot{x}_R$ can be expressed as the product of a constant k_T and k_R, and the corresponding unit directional vector for translation $\dot{x}_{unitT}$ and rotation $\dot{x}_{unitR}$, respectively.

$$\dot{X} = \begin{bmatrix} \dot{x}_T \\ \dot{x}_R \end{bmatrix} = \begin{bmatrix} k_T \dot{x}_{\text{unit}T} \\ k_R \dot{x}_{\text{unit}R} \end{bmatrix} \tag{23}$$

Let $d \in \mathbb{R}$ denote the relative ratio between the magnitudes of the end-effector's translational and rotational velocities:

$$d = \frac{k_T}{k_R} \tag{24}$$

From Eqs. (1) and (3):

$$\dot{X} = k_R \begin{bmatrix} d \cdot \dot{x}_{\text{unit}T} \\ \dot{x}_{\text{unit}R} \end{bmatrix} = J\dot{\theta} = J\left(\dot{\theta}_T + \dot{\theta}_R\right) \tag{25}$$

Substituting (22) into (26):

$$k_R \begin{bmatrix} d \cdot \dot{x}_{\text{unit}T} \\ \dot{x}_{\text{unit}R} \end{bmatrix} = J\left(\tilde{J}_T^{+}\dot{x}_T + \tilde{J}_R^{+}\dot{x}_R + (\alpha+\beta)\sum_{i=1}^{n-r(J)} t_i\,\sigma_i\right) \tag{26}$$

Define:

$$f\left(t_i, \alpha, \beta\right) = J\left(\tilde{J}_T{}^{+}\dot{x}_T + \tilde{J}_R{}^{+}\dot{x}_R + (\alpha+\beta)\sum_{i=1}^{n-r(J)} t_i\,\sigma_i\right) \tag{27}$$

Let $\dot{x}_{new}$ represent $\begin{bmatrix} d \cdot \dot{x}_{\text{unit}T} \\ \dot{x}_{\text{unit}R} \end{bmatrix}$. Since $\dot{x}_{new}$ is an m-dimensional vector, Eq. (28) can be regarded as a system of m equations. Substituting Eq. (28) into (27), the system of equations can be expressed as:

$$\begin{aligned} k_R \cdot \dot{x}_{\text{new}g} &= f_g\left(t_i, \alpha, \beta\right) \\ g &= 1, 2, \ldots, m \\ i &= 1, 2, \ldots, n - r\left(J\right) \end{aligned} \tag{28}$$

where g denotes the *g-th* equation in Eq. (28), $\dot{x}_{newg}$ is the g-th element in the column vector $\dot{x}_{new}$ and $f_g(t_i, \alpha, \beta)$ represent the g-th row of $f(t_i, \alpha, \beta)$. Equation (29) exhibits the relationship between the velocity coefficient k_R and the variables t_i, α, and β. Therefore, the maximum velocity of the robot along the specified coupled translational and rotational motion direction $\dot{X}$ is $k_{R_{\max}}\dot{x}_{new}$, where $k_{R_{\max}}$ is the maximum value that k_R can take under the constraint conditions specified in Eq. (27).

3.4. Conversion into linear programming equations

Transform the first equation containing k_R in Eq. (29) into an objective function of k_R in terms of t_i. Assuming the selected equation is the first equation, we have

$$\begin{aligned} k_R = F(t_i, \alpha, \beta) &= \frac{f_1(t_i, \alpha, \beta)}{\dot{x}_{\text{new}1}} \\ i &= 1, 2, \ldots, n - r\left(J\right) \end{aligned} \tag{29}$$

https://doi.org/10.1017/S0263574726103956 Published online by Cambridge University Press

By eliminating k_R from the m equations in Eq. (34), m-1 equations without k_R can be obtained. These m-1 equations can be used as equality constraints and can be expressed as:

$$\text{s.t.} \quad \begin{cases} \dfrac{f_1(t_i,\alpha,\beta)}{\dot{x}_{\text{new1}}}\dot{x}_{\text{new2}} - f_2(t_i,\alpha,\beta) = 0, \\ \dfrac{f_1(t_i,\alpha,\beta)}{\dot{x}_{\text{new1}}}\dot{x}_{\text{new3}} - f_3(t_i,\alpha,\beta) = 0, \\ \vdots \\ \dfrac{f_1(t_i,\alpha,\beta)}{\dot{x}_{new1}}\dot{x}_{new_m} - f_m(t_i,\alpha,\beta) = 0. \end{cases} \tag{30}$$

For simplicity of expression, we represent the above equation in the following form:

$$\begin{gathered} y_p(t_i,\alpha,\beta) = 0 \\ p = 1,2,\ldots,m-1 \\ i = 1,2,\ldots,n-r(J) \end{gathered} \tag{31}$$

where $y_p(t_i,\alpha,\beta)$ represents the p-th of the m-1 equations after eliminating k_R, and the Eq. (17) can be transformed into n inequality constraints, as follows:

$$-1 \le \dot{\theta}_q(t_i,\alpha,\beta) \le 1, \quad q = 1,2,\ldots,n \tag{32}$$

It should be noted that in Eq. (23), we use a joint velocity limit of ± 1 rad/s as an example, resulting in the Eq. (33). In fact, the joint velocity limits at the corresponding joints can be adjusted by modifying the velocity range of any row in Eq. (27). $\dot{\theta}_q(t_i,\alpha,\beta)$ represents the q-th row of $\dot{\theta}(t_i,\alpha,\beta)$ in Eq. (22).

In the case of heterogeneous joint-velocity limits:

$$-\dot{\theta}_{q\,\max} \le \dot{\theta}_q \le \dot{\theta}_{q\,\max} \tag{33}$$

Here, $\dot{\theta}_{qmax}$ denotes the maximum (limit) velocity of the q-th joint.

When a component of $\dot{\theta}$ reaches its joint-velocity limit, the robot's speed along $\dot{\boldsymbol{x}}$ attains its maximum as well. Accordingly, the maximum speed scaling factor k_{Rmax} is computed as follows:

$$k_{R\,\max} = \min_i \left\{ \left| \frac{\dot{\theta}_{q\,\max}}{\dot{\theta}_q} \right| \right\}, \quad i = 1,2,\ldots,n \tag{34}$$

Therefore, the problem of solving the robot's maximum motion velocity $k_{Rmax}\dot{x}_{new}$ in the specified direction can be transformed into a LP problem for maximizing $k_{R_{max}}$:

$$\begin{gathered} \max \; F(t_i,\alpha,\beta) \\ \text{s.t.} \quad \begin{cases} y_p\,(t_i,\alpha,\beta) = 0, \\ -1 \le \dot{\theta}_q\,(t_i,\alpha,\beta) \le 1 \end{cases} \\ p = 1,2,\ldots,m-1 \\ q = 1,2,\ldots,n \\ i = 1,2,\ldots,n-r(J) \end{gathered} \tag{35}$$

Therefore, given the robot end-effector's motion parameters ($\dot{x}_{unitT}$, $\dot{x}_{unitR}$, d) and joint velocity constraints, the robot's maximum motion velocity in the specified direction at the target joint configuration can be approximately determined. Notably, we adopt the classical interior-point method for LP, which is particularly effective for small- to medium-scale LP problems, offering minimal computational overhead and strong robustness.

$$k_{R\,\max} = \min_i \left\{ \left| \frac{\dot{\theta}_{q\,\max}}{\dot{\theta}_q} \right| \right\}, \quad i = 1,2,\ldots,n \tag{36}$$

https://doi.org/10.1017/S0263574726103956 Published online by Cambridge University Press

Since the LP method is used for solving, this approach takes more time to compute compared to the DTF method. The advantage of this method lies in the fact that for serial robots with different degrees of freedom, the entire solving process remains unchanged. The only variations throughout the process are the degrees of freedom m of the workspace and the number of joints n of the robot. Therefore, this method is applicable to both nonredundant and redundant robots. Moreover, as shown in Eq. (36), since this method considers the velocity constraints of each joint, it can be applied to scenarios where the joint velocity limits of the serial robot are different.

4. Discussion

Our objective is to maximize k_R. When all joint velocity limits are identical, the solution that satisfies Eq. (36) corresponds to the minimum infinity-norm solution of the system. In the nonredundant case, and away from singularities, the Jacobian J is full rank, and therefore, the solution is unique. In contrast, for redundant manipulators, except at singular configurations, the solution consists of a particular solution plus a null-space component. The proposed method searches over the entire solution space and uses LP to eliminate those solutions that violate the joint-limit constraints. The joint-velocity vector obtained in this way is thus the minimum infinity-norm solution of the equation.

$$k_{R\max} = \min_i \left\{ \left| \frac{\dot{\theta}_{q\max}}{\dot{\theta}_q} \right| \right\}, \quad i = 1, 2, \ldots, n \tag{37}$$

When the joint-velocity limits are not identical, the solution obtained in the redundant case is not necessarily the minimum infinity-norm solution. For example, suppose the minimum infinity-norm solution yields a joint-velocity vector whose largest component occurs at the third joint with magnitude 0.5 rad/s, while the joint limit of that joint is 1 rad/s; this gives $k_{R_{max}}$=2. Another feasible solution may yield its largest component at the fourth joint with magnitude 1 rad/s, whose limit is 4 rad/s; this yields $k_{R_{max}}$=4. In this situation, searching within the solution space via LP to maximize $k_{R_{max}}$ provides an effective way to handle unequal joint-velocity limits.

4.1. Experiment and analysis

In this section, experiments were carried out to verify the accuracy of the proposed method for solving the oriented maximum velocity of serial robots, by comparison with the DTF method. A set of joint configurations is randomly selected first, and multiple directions in the robot's workspace are then used to measure the maximum translational and rotational velocities. Two robotic platforms, i.e., UR5e (6-DoF) and Sawyer (7-DoF), are utilized in the experiments to evaluate the performance in both nonredundant and redundant serial robots, applied to scenarios where the joint velocity limits of the serial robot are different.

4.2. Experimental design

In the experiment series, the initial joint configurations for the nonredundant robot (i.e., UR5e) experimental joint configuration is selected as [−163.46°, −88.43°, −103.69°, −54.15°, 84.15°, −25.78°], while for the redundant robot (i.e., Sawyer) is [−193.46°, −88.43°, −193.69°, −54.20°, 84.15°, −95.78°, 50°]. For the directions, we used the spherical coordinate system of the end-effector. Namely, with a step size of 10°, the polar angle is taken from 0° to 180°, and the azimuth angle is taken from 0° to 350°, resulting in a total of 648 directions. Since 34 of these directions overlap, the remaining 614 unique directions are selected as the experimental directions after removing the overlapping ones, as illustrated with red arrow in Figure 2.

For the translational velocity, the robot end-effector moves in the direction indicated by the arrow. For angular velocity, its direction is determined by the axis of rotation, and the positive sense follows

https://doi.org/10.1017/S0263574726103956 Published online by Cambridge University Press

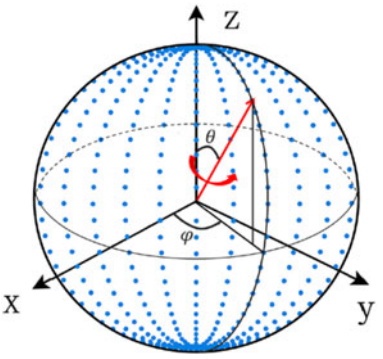


Figure 2. *Experimental directions.*

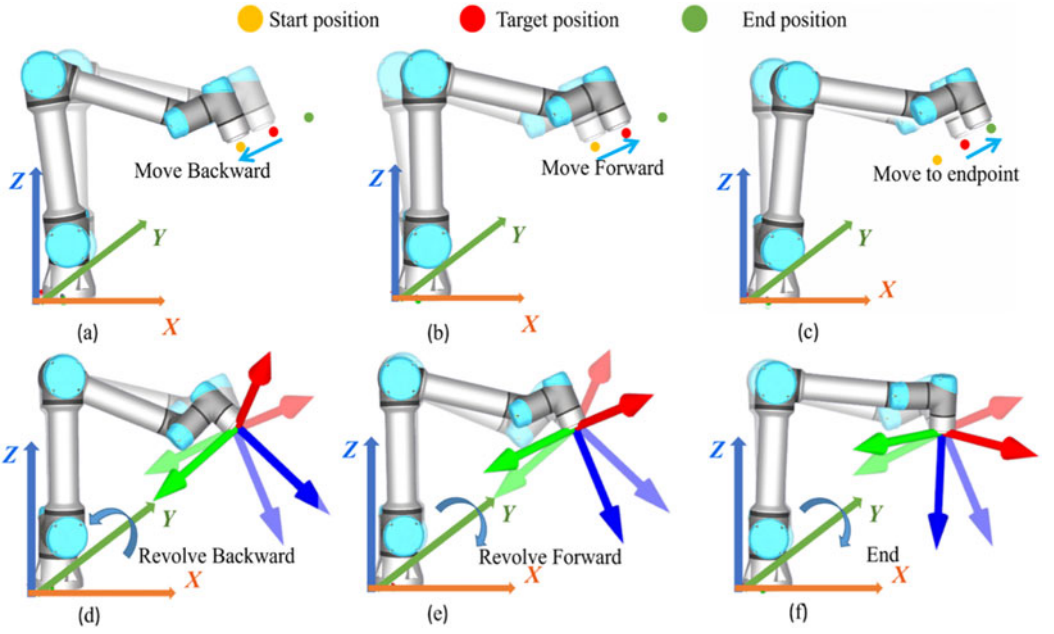


Figure 3. *Experimental motion sequence. (a), (b), (c) Translational experiments and (d), (e), (f) rotational experiments.*

the right-hand rule: when the right thumb points along this axis vector, the curl of the remaining fingers indicates the positive direction of rotation.

To ensure that the robot reaches its maximum speed upon arriving at the target position, we prescribe an acceleration phase so that it attains the desired peak velocity.

In the experiments, we defined three positions:

1. **Target position:** the position at which the experiment records the maximum speed to be reported as the result.
2. **Start position:** the position obtained by moving from the target position for 2 s in the direction opposite to the prescribed direction.
3. **End position:** the position obtained by moving from the target position for 2 s along the prescribed direction (Figure 3).

https://doi.org/10.1017/S0263574726103956 Published online by Cambridge University Press

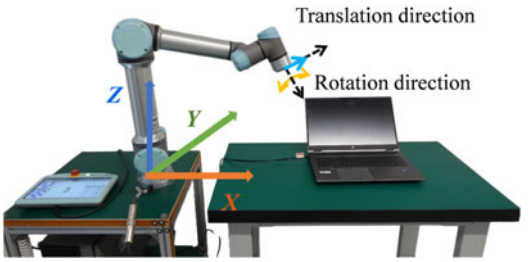


Figure 4. *UR 5e robot.*

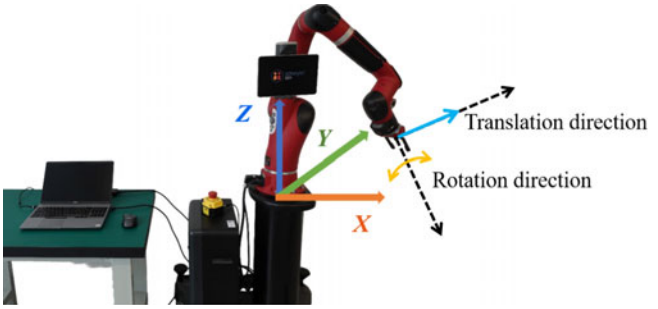


Figure 5. *Sawyer Robot.*

The experiment procedure involves:

- Move robot to target position.
- Move(Revolve) backward at maximum translational(rotational) velocity for 2 s. After reaching the target position, the robot moves for 2 s at its maximum speed in the direction opposite to the experimental direction to arrive at the start position, thereby leaving sufficient room to accelerate to maximum speed upon returning to the target.
- Move(Revolve) forward at maximum translational(rotational) velocity. Starting from the departure position, the robot moves along the experimental direction at maximum speed, and the speed at the instant it passes the target position is recorded.
- The motion stops when the robot continues to move and reaches the end position

$$e = \left| \frac{\dot{x}_e - \dot{x}_t}{\dot{x}_e} \right| \tag{38}$$

Where $\dot{x}_e$ is the experimental velocity and $\dot{x}_t$ is the theoretical velocity.

4.3. Robotic experimental platform

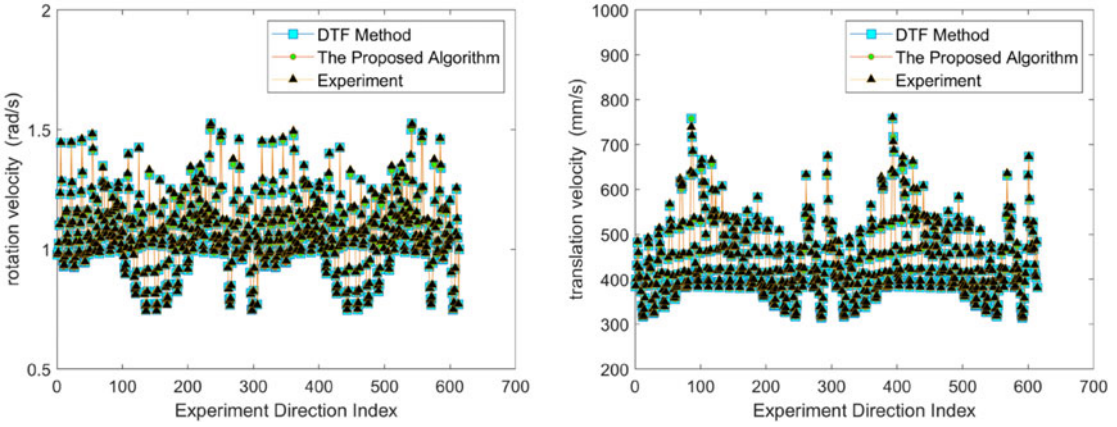
In this section, the real-world search experiment was verified in the laboratory environment. As shown in Figure 4, the UR5e robot is a six-degree-of-freedom serial robot. The host computer, equipped with an i7-10700 CPU and 16 GB RAM, communicates with the UR5e robot via RTDE over Ethernet. It controls robot motion and receives real-time joint and TCP data at 500 Hz. Joint velocity limits are set via the teach pendant, and all control and data acquisition programs are developed in Python.

As shown in Figure 5, the Sawyer robot is a seven-degree-of-freedom serial robot. The host computer runs Ubuntu with ROS and MoveIt! for controlling the Sawyer robot via Ethernet. It uses the MoveGroupCommander interface for motion planning and executes commands through ROS nodes. Joint limits are set in joint_limits.yaml, and real-time joint and TCP data are collected at 500 Hz. All programs are implemented in Python.

https://doi.org/10.1017/S0263574726103956 Published online by Cambridge University Press

Table I. *Comparison between DTF and our proposed methods in a nonredundant case (UR5e)*

	Computation error		Computation time (ms)	
	DTF	**Ours**	**DTF**	**Ours**
Maximum value	2.5%	2.5%	0.15	4.37
Minimum value	0.01%	0.01%	0.02	1.06
Mean value	0.44%	0.44%	0.02	1.38
Standard deviation	0.32%	0.32%	0.01	0.35



(a) (b)

Figure 6. *Results comparison between two calculated (both DTF and ours) and experimental oriented maximum velocity by a nonredudant robot (UR5e): (a) rotational movement and (b) translational movement.*

4.4. Experimental results analysis

Based on the aforementioned experimental design, tests were conducted on both robots, and results are analyzed with focus on two aspects: the accuracy of the velocity and the computation time.

In the nonredundant robot experiment, Figure 6(a) and (b) illustrates the distribution of calculated maximum velocity values using the two methods for the 614 selected directions, along with the experimental values. The results show that the proposed method and the DTF method produce identical results (Table I), which is because, in the nonredundant case, the Jacobian matrix of the selected configuration in the experiment is invertible, and the solution corresponding to Eq. (22) is unique. Therefore, the generalized inverse solution used in the DTF method and the fundamental solution set used in this paper yield the same results. The maximum error is 2.50%, with a mean error of 0.44% and a standard deviation of 0.32%.

In terms of computation time, as shown in Figure 7, the overall computation time of the proposed method is higher than that of the DTF method, with average computation times of 1.38 ms and 0.02 ms, respectively (Table I). This is consistent with the theoretical analysis. Unlike nonredundant robots, the proposed algorithm shows significant differences in the accuracy of maximum velocity calculations for redundant robots.

Figure 8(a) and (b) shows the distribution of calculated maximum velocity values using the two methods for the 614 selected directions, along with the experimental values. The proposed method has a maximum error of 2.20%, a mean error of 0.67%, and a standard deviation of 0.38%. In contrast,

https://doi.org/10.1017/S0263574726103956 Published online by Cambridge University Press

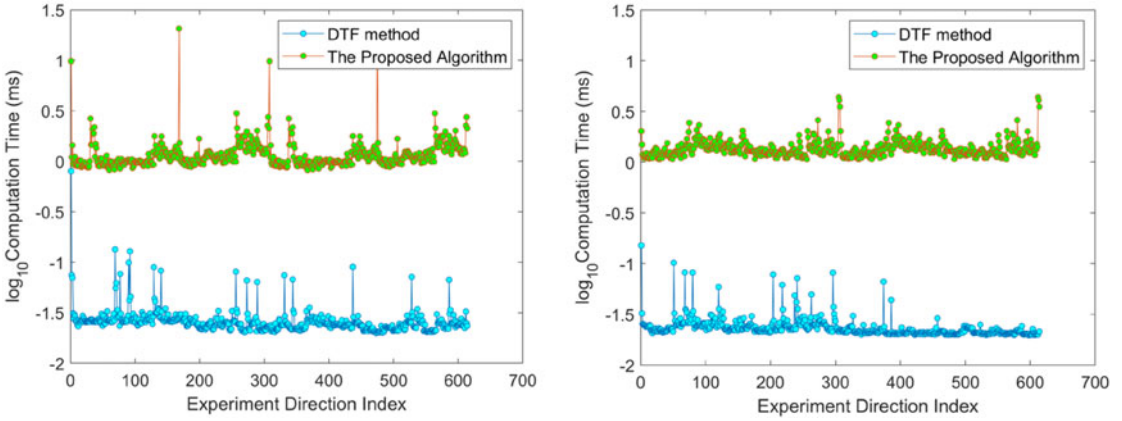


Figure 7. *Computation time comparison between DTF and ours method when performing oriented maximum velocity by a nonredudant robot (UR5e): (a) rotational movement and (b) tansliational movement.*

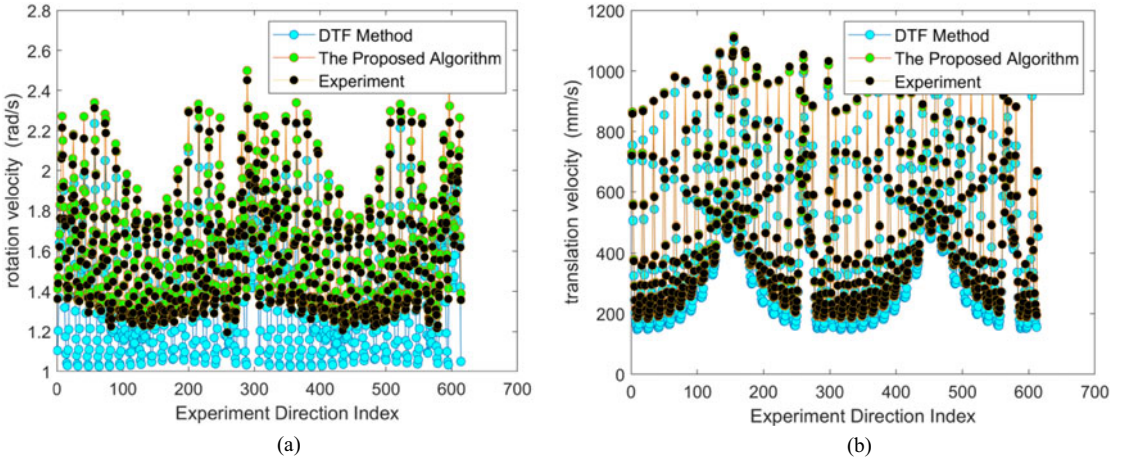


Figure 8. *Results comparison between two calculated (both DTF and ours) and experimental oriented maximum velocity by a redudant robot (Saywyer): (a) rotational movement and (b) tansliational movement.*

the DTF method shows a maximum error of 24.3%, a mean error of 12.56%, and a standard deviation of 7.04%. Overall, our proposed method performs better than the DTF method. This is because, in the redundant case, the Jacobian matrix is noninvertible and the workspace velocity corresponding to the sum of $\dot{\theta}_T$ and $\dot{\theta}_R$, solved using the generalized inverse matrix, does not necessarily represent the maximum workspace velocity among the infinite set of joint velocity solutions, leading to inaccuracies in the solution.

In terms of computation time, as shown in Figure 9, the proposed algorithm's experimental results show that the nonredundant robot's computation time is overall higher than that of the DTF method, which is consistent with the results for the redundant robot. The average computation times are 1.74 ms and 0.04 ms, respectively. Compared to the nonredundant robot, the computation time for both algorithms increases overall when applied to the redundant robot.

https://doi.org/10.1017/S0263574726103956 Published online by Cambridge University Press

Table II. *Comparison between DTF and our proposed methods in a redundant case (Sawyer).*

	Computation error		Computation time (ms)	
	DTF	**Ours**	**DTF**	**Ours**
Maximum value	24.3%	2.2%	5.07	24.2
Minimum value	0.22%	0.01%	0.02	1.14
Mean value	12.56%	0.67%	0.04	1.74
Standard deviation	7.04%	0.38%	0.20	1.36

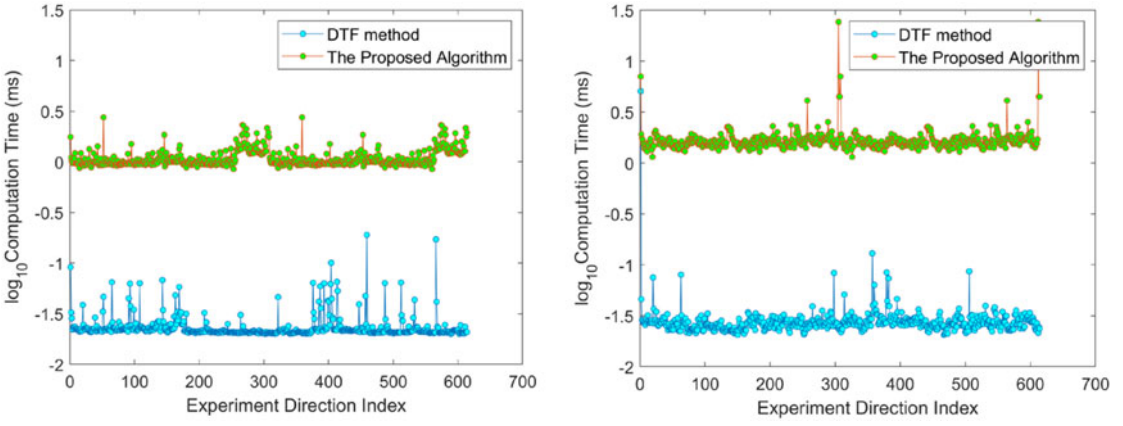


Figure 9. *Computation time comparison between DTF and ours method when performing oriented maximum velocity by a redudant robot (Sawyer): (a) rotational movement and (b) tanslational movement.*

Table I summarizes the performance of the two methods applied to the nonredundant robot. The proposed method transforms the problem of solving the oriented maximum velocity into an equivalent standard LP problem, and the computed result can converge to the optimal solution within the feasible region. Therefore, the calculated result is theoretically the true value, with minimal error compared to the experimental value, and the error mainly originates from experimental inaccuracies. The DTF method's results closely match those of the proposed algorithm in terms of computation error. This is because the Jacobian matrix is full rank in nonredundant robots at nonsingular positions, and the solution is unique. Theoretical analysis aligns with the experimental results.

Table II shows the performance of the two methods applied to the redundant robot. The proposed method exhibits minor calculation errors compared to the experimental values, and the computational performance is roughly the same as that of the nonredundant robot. However, the DTF method shows a more significant deviation from the experimental results than the proposed algorithm. At nonsingular configurations, regardless of redundancy, the augmented Jacobian (under the respective constraint) is generally noninvertible, but a solution can be obtained using the Moore–Penrose pseudoinverse. The cause of the error in the obtained solution has already been described following Eq. (12) in Section 2.3. The theoretical analysis is consistent with the experimental results.

Tables I and II also show the computation times of the two methods applied to different robots. Since the proposed algorithm uses a LP approach and considers all solutions within the fundamental solution set, the computation time is longer than the DTF method.

https://doi.org/10.1017/S0263574726103956 Published online by Cambridge University Press

5. Conclusion

In this paper, we propose a method for solving the oriented maximum velocity of serial robots based on the decomposition of translational and rotational motions. Theoretically, compared to the kinematic directional index, this method can be directly applied to both nonredundant and redundant serial robots. It avoids the issue of inconsistent physical units and has a relatively simple structure. Unlike velocity ellipsoids, velocity polytopes and the kinematic directional index can only evaluate translational and rotational velocities separately. This method can accurately assess the maximum motion velocity of serial robots in coupled translational and rotational directions. Compared to the existing DTF method, it can accurately evaluate the maximum velocity of serial robots in a specified motion direction while considering precise joint velocity limits and specific task requirements. The results of experimental analysis on redundant and nonredundant robots show that, although the proposed algorithm takes longer to compute compared to the existing DTF method, the average computation time across 614 directions does not exceed 2 ms. The proposed algorithm demonstrates a significant advantage in terms of accuracy for redundant robots, with an average error of 0.67%, which is much lower than the DTF method's 12.56%. The experimental results are consistent with the theoretical analysis, indicating that the method can be effectively applied to evaluate robotic motion performance. In the current study, only joint velocity limits are incorporated into the optimization model. Nevertheless, the proposed linear-programming framework provides good extensibility. In future work, joint acceleration limits and actuator torque limits will be further introduced to evaluate the dynamically reachable directional velocity of serial robots from a given motion state. This extension is expected to provide a more realistic assessment of robot velocity capability by accounting for both kinematic feasibility and dynamic realizability.

Author contributions. Material preparation, data collection, and writing were performed by Kai Wu, Hengyi An. Conception and design, writing, and review were performed by Kai Wu and Shaoxiong Feng. Review and editing were performed by Mingfeng Wang. All authors commented on previous versions of the manuscript. All authors read and approved the final manuscript.

Financial support. Chunhui Project Foundation of the Education Department of China, Grant 202201789. GJYC program of Guangzhou, Grant 2024D03j0005.

Competing interest. The authors declare that Kai Wu is an Associate Editor for the journal but was completely excluded from the review process. The data that support the findings of this study are available from the corresponding author upon reasonable request.

Ethical approval. Not applicable.

References

[1] T. Ardeshiri, M. Norrlöf, J. Löfberg and A. Hansson, "Convex optimization approach for time-optimal path tracking of robots with speed dependent constraints," *IFAC Proc.* **44**(1), 14648–14653 (2011).

[2] W. J. Marais, S. B. Williams and O. Pizarro, "Anisotropic disturbance rejection for kinematically redundant systems with applications on an UVMS," *IEEE Robot. Autom. Lett.* **6**(4), 7017–7024 (2021).

[3] L. Liu, F. Guo, Z. Zou and V. G. Duffy, "Application, development and future opportunities of collaborative robots (Cobots) in manufacturing: A literature review," *Int. J. Human–Comput. Interact.* **40**(4), 915–932 (2024).

[4] C. A. Klein and B. E. Blaho, "Dexterity measures for the design and control of kinematically redundant manipulators," *Int. J. Robot. Res.* **6**(2), 72–83 (1987).

[5] J. K. Salisbury and J. J. Craig, "Articulated hands: Force control and kinematic issues," *The Int. J. Robot. Res.* **1**(1), 4–17 (1982).

[6] R. Dubey and J. Y. S. Luh, "Redundant robot control using task based performance measures," *J. Robot. Syst.* **5**(5), 409–432 (1988).

[7] T. Yoshikawa. Manipulability and redundancy control of robotic mechanisms. **In:** *1985 IEEE International Conference on Robotics and Automation Proceedings* (Mar. 1985) pp. 1004–1009.

[8] T. Yoshikawa, "Translational and rotational manipulability of robotic manipulators," *1990 American Control Conference*, 228–233 (1990). doi: 10.23919/ACC.1990.4790733.

https://doi.org/10.1017/S0263574726103956 Published online by Cambridge University Press

[9] J. Lee. A study on the manipulability measures for robot manipulators. **In:** *Proceedings of the 1997 IEEE/RSJ International Conference on Intelligent Robot and Systems. Innovative Robotics for Real-World Applications. IROS '97*, **3**, (1997) pp. 1458–1465.
[10] H.-B. Choi and J. Ryu, "Convex hull-based velocity transmission capability of parallel manipulators," *Int. J. Manuf. Technol.* **65**(5), 695–704 (2013).
[11] Y.-S. Hwang, J. Lee and C. Hsia. A recursive dimension-growing method for computing robotic manipulability polytope. **In:** *Proceedings 2000 ICRA. Millennium Conference. IEEE International Conference on Robotics and Automation. Symposia Proceedings, Apr*, **3**, (2000) pp. 2569–2574.
[12] J. Lee, "Velocity workspace analysis for multiple arm robot systems," *Robotica* **19**(5), 581–591 (2001).
[13] J. Lee, "A structured algorithm for minimum l∞-norm solutions and its application to a robot velocity workspace analysis," *Robotica* **19**(3), 343–352 (2001).
[14] H. Pham and Q.-C. Pham, "A new approach to time-optimal path parameterization based on reachability analysis," *IEEE Trans. Robot.* **34**(3), 645–659 (2018). doi: 10.1109/TRO.2018.2819195.
[15] A. Bowling and O. Khatib, "The dynamic capability equations: A new tool for analyzing robotic manipulator performance," *IEEE Trans. Robot.* **21**(1), 115–123 (2005). doi: 10.1109/TRO.2004.837243.
[16] C. Pholsiri, C. Kapoor and D. Tesar. Real-Time Robot Capability Analysis. **In:** *2005 International Design Engineering Technical Conferences and Computers and Information in Engineering Conference*, **47446**, (2005) pp. 973–982.
[17] G. Boschetti, "A novel kinematic directional index for industrial serial manipulators," *Appl. Sci.* **10**(17), 5953 (2020).
[18] G. Boschetti and R. Minto, "Directional index for the performance of redundant manipulators," *Robotica* **41**(10), 2964–2984 (2023).
[19] G. Boschetti, R. Rosa and A. Trevisani, "Parallel robot translational performance evaluation through Direction-Selective Index (DSI)," *J. Robot.* **2011**(1), 129506 (2011). doi: 10.1155/2011/129506
[20] I. Mansouri and M. Ouali, "A new homogeneous manipulability measure of robot manipulators, based on power concept," *Mechatronics* **19**(6), 927–944 (2009).
[21] I. Mansouri and M. Ouali, "The power manipulability – A new homogeneous performance index of robot manipulators," *Robot. Com.-Int. Manuf.* **27**(2), 434–449 (2011).
[22] S. Stradovnik and Hace A., "Task-oriented evaluation of the feasible kinematic directional capabilities for robot machining," *Sensors* **22**(11), 4267 (2022). doi: 10.3390/s22114267
[23] A. A. Hassan, M. El-Habrouk and S. Deghedie, "Inverse kinematics of redundant manipulators formulated as quadratic programming optimization problem solved using recurrent neural networks: A review," *Robotica* **38**(8), 1495–1512 (2020).
[24] A. Colomé and C. Torras, "Closed-loop inverse kinematics for redundant robots: Comparative assessment and two enhancements," *IEEE/ASME Trans. Mechatron.* **20**(2), 944–955 (2015).
[25] J. Zhao, X. Yang, Z. Zhao and H. Liu, "A trajectory planning method for load-carrying capacity improvement of redundant space manipulator with large external force," **In:** *Intelligent Robotics and Applications. ICIRA 2021. Lecture Notes in Computer Science* (X. J. Liu, Z. Nie, J. Yu, F. Xie, and R. Song, eds.), vol. 13016 (Springer, Cham, 2021). doi: 10.1007/978-3-030-89092-6_34
[26] S. Patel and T. Sobh, "Manipulator performance measures - a comprehensive literature survey," *J. Intell. Robot. Syst.* **77**(3), 547–570 (2015).
[27] J. H. Kim, Y. J. Shin and I. G. Jang, "Evaluating the maximum directional kinematic capability of a redundant manipulator based on allowable velocity and force," *IEEE Access* **9**, 88085–88097 (2021).

Cite this article: K. Wu, H. An and M. Wang (2026). "Oriented maximum velocity evaluation in serial robots using translational and rotational decomposition", Robotica **44**, 2336–2351. https://doi.org/10.1017/S0263574726103956

https://doi.org/10.1017/S0263574726103956 Published online by Cambridge University Press