Received January 16, 2019, accepted February 6, 2019, date of publication February 19, 2019, date of current version March 8, 2019.

Digital Object Identifier 10.1109/ACCESS.2019.2900086

Iteratively Successive Projection: A Novel Continuous Approach for the Task-Based Control of Redundant Robots

PEI JIANG[1], SHUIHUA HUANG[2], JI XIANG[3], (Senior Member, IEEE), AND MICHAEL Z. Q. CHEN[4], (Senior Member, IEEE)

[1]College of Mechanical Engineering, University of Chongqing, Chongqing 400030, China
[2]Hangzhou Hikvision Digital Technology Co., Ltd., Hangzhou 310052, China
[3]Department of System Science and Engineering, Zhejiang University, Hangzhou 3100027, China
[4]School of Automation, Nanjing University of Science and Technology, Nanjing 210094, China

Corresponding authors: Pei Jiang (peijiang@cqu.edu.cn) and Michael Z. Q. Chen (mzqchen@outlook.com)

This work was supported in part by the National Natural Science Foundation of China under Grant 51705050 and Grant 51709023, and in part by the Technology Innovation and Application Demonstration Project of Chongqing under Grant cstc2018jszx-cyzdx0147.

ABSTRACT The task-based control approach with an activation task mechanism endows a robot the possibility to act complex behaviors in an unstructured environment. However, the task-based control approach based on the classical inverse may suffer from discontinuities as tasks switch between inactive and active states. In this paper, an iteratively successive projection method is proposed to construct a null space projection operator by iteratively successive projecting the space between the null space of each task. And the null space projection operator constructed by the iteratively successive projection method should converge to the null space projection operator obtained by the classical inverse, as the iteration time increases to infinity. By introducing activation factors, a continuous projection operator is obtained with a finite-time iteration. Based on the continuous projection operator, a closed-loop control algorithm is presented, which ensures both the continuity of the joint motion and the boundedness of the tracking error. The simulations on a six-link planar manipulator for the trajectory tracking task in the presence of obstacles show the efficacy of the proposed method in continuity, stability, and calculation-time consumption.

INDEX TERMS Robot kinematics, redundancy, motion control, null-space projection, discontinuity.

I. INTRODUCTION

The classical control of a robot is to preplan a desired trajectory in the joint space to realized an objective task, usually a target-tracking task of the end-effector [1]. This control framework is free of discontinuity in joint motion by properly preplanning. However, the joint space method is limited for the cases with subtasks in the task space, such as obstacle avoidance and visual servoing. Due to the limitation of the control methods in the joint space, the control approaches in task space have been proposed [2]–[5]. In the task-based approach, the gradient project method (GPM) was first adopted to achieve joint-limit avoidance task in [6], and obstacle avoidance task in [7]. The augmented Jacobian technique was proposed in [8] and [9], which treats other tasks equally with the tracking task by adding the gradient vectors of other tasks into the Jacobian matrix. The weighted least-norm based methods were proposed to realize the tracking tasks under multiple constraints [3], [10]–[13]. All these methods work directly in the desired task space, which produces a more intuitive control manner.

The task-based approaches extend the scope of application of robots to the unstructured environments, which requires the execution of multiple tasks simultaneously or sequentially under the limited degree of freedom (DOF). To effectively utilize the DOFs, the task set of the task-based approach should not be constant, for example, when a robot is far away from obstacles, the obstacle avoidance tasks do not need to be taken into consideration. Some research has already been conducted to handle the task-based control with a varying task set. The task is removed from the task set as sensor-visibility is lost which cannot provide the feature for visual servoing [14]. Meanwhile, a task can also be removed from the task set when it is sufficiently close to the target to release more DOFs for other tasks [15].

The associate editor coordinating the review of this manuscript and approving it for publication was Luigi Biagiotti.

However, in those task-based approaches with activation task mechanism, discontinuity happens as a task is removed or added to the task set. In [16], a mathematical proof was given to demonstrate that discontinuity generally exists under the control methods based on the pseudo-inverse, as the activity of a task changes. The explanation based on the task projection onto other tasks was presented in [17]. How to solve the discontinuous problem becomes the key problem in the task-based control with activation task mechanism. Mansard proposed a continuous inverse, which is a linear combination of all the inverses of all the possible tasks [16]. And the Mansard's inverse has two characteristics: first, the inverse is continuous during transition of the task state switch; second, the inverse is equal to the pseudo-inverse when all the tasks are inactive or fully active. This guarantees the stability of the system, as all the tasks are in inactive or fully active state. The Mansard's inverse was applied to the multiple tasks with different priorities [18], and the selective damping was introduced to dealing singularities [19]. However, the calculation of the Mansard's inverse is computationally intensive. And an intermediate desired value approach was also presented to solve the discontinuity problem, which directly modifies all the tasks to the intermediate desired values to ensure the continuity in the transition period [17]. A smooth control method based on successive null space projection operator is proposed to solve the discontinuity as the task state switches at a low-computation cost, but the method cannot guarantee the stability of the system, even if all the tasks are in inactive or fully active state [20].

Motivated by the continuous characteristic of successive null space projection, this paper presents a method to construct the null space by iteratively successive projection, which iteratively projects the space between the null space of each subtask in sequence. Based on the iteratively successive projection, a continuous projection operator is presented by introducing the activation factors. A new continuous algorithm based on the continuous projection operator is proposed to solve the task-based control problem with a task activation mechanism to ensure both continuity and stability.

It is worth pointing out the main contributions of this paper as follows.

1) This paper presents an iteratively successive projection method to form the null space projection operator, which has not yet been reported in the literature. And a geometric interpretation for iteratively successive projection method is also given.
2) A continuous inverse based on the iteratively successive projection is proposed, and the iteration times balance the continuity and the degree of approximation to the Moore-Penrose.
3) A continuous closed-loop control algorithm is proposed to solve the task-based control problem of the redundant robot with the activation task mechanism, which can ensure both continuity and stability at the same time.

The paper is organized as follows. Section II reviews classical task-based control methods for robots, and a brief analysis for discontinuity caused by task state switch is also presented in this section. The method of iteratively successive projection is presented in the Section III. Section IV presents the continuous closed-loop control algorithm for the task-based control. The performances of the proposed method and two other method in case of obstacle avoidance are compared in the Section V. Finally, Section VI concludes the paper.

II. BACKGROUND

A. KINEMATIC CONTROL

Let $q \in \mathbb{R}^n$ is the vector of joint position and $x_e \in \mathbb{R}^{m_e}$ be the vector in the task space. A task can be formulated as:

$$x_e = f(q), \tag{1}$$

where $f(q) : \mathbb{R}^n \mapsto \mathbb{R}^{m_e}$ is the task function, and $m_e \in \mathbb{R}$ is the number of the dimension of the task vector, i.e. if x_e is the position vector of the end-effector, equation (1) is the forward kinematic equation of the robot. Differentiating (1) yields the task formulation at the velocity level as:

$$\dot{x}_e = \frac{\partial f(q)}{\partial q}\dot{q} = J_e\dot{q}, \tag{2}$$

where J_e is the Jacobian of the tasks. Besides the tracking task, a redundant robot may have to realize several other tasks in the motion process, such as obstacle avoidance, joint-limit avoidance, visual servoing, etc. Assume there are k tasks, all the tasks can be combined to form a kinematic equation of the robot as

$$\dot{x} = J\dot{q}, \tag{3}$$

where $\dot{x} = [\dot{x}_1^T, \dot{x}_2^T, \cdots, \dot{x}_k^T]^T$ is the velocity of all tasks, $\dot{x}_i$ is the velocity of the i^{th} task. And J is the augmented Jacobian of all tasks, which is defined as:

$$J = \begin{bmatrix} J_1^T & J_2^T & \cdots & J_k^T \end{bmatrix}^T, \tag{4}$$

where $J_i \in \mathbb{R}^{m_i \times n}$ is the Jacobian of the i^{th} task, m_i is the dimension of the i^{th} task. In this paper, only serial redundant robotic systems are studied, the number of joints is larger than the dimension of the tasks.

The inverse technique is adopted to obtain the least-norm solution of (3) as follows:

$$\dot{q} = J^{\dagger}\dot{x}, \tag{5}$$

where $J^{\dagger} = J^T(JJ^T)^{-1}$ is the pseudo-inverse of Jacobian J. $J^{\dagger}\dot{x}$ is the least-norm solution to obtain the minimum-norm joint velocities, which satisfies

$$\dot{q} = \arg\min\{z^T z\} \quad \text{subject to: } \dot{x} = Jz. \tag{6}$$

For the redundant robot, $m = \sum_{i=1}^{k} m_i < n$ holds, there exists nonzero homogenous solutions of (3). Combining the homogenous solution and the right-hand side of (5) yields the general solution as follows:

$$\dot{q} = J^{\dagger}\dot{x} + Pz, \tag{7}$$

where $z \in \mathbb{R}^n$ is a free vector, P is the classical null space projection operator of J, which is defined in the following form:

$$P = I_n - J^{\dagger}J. \tag{8}$$

The singular value decomposition (SVD) of the task Jacobian can be formulated in the form:

$$J = U\Sigma V^T = \sum_{i=1}^{m} \sigma_i u_i v_i^T, \tag{9}$$

where $U \in \mathbb{R}^{m\times m}$ is the orthonormal matrix of the output singular vectors $u_i \in \mathbb{R}^m$, $V \in \mathbb{R}^{n\times n}$ is the orthonormal matrix of the input singular vectors $v_i \in \mathbb{R}^n$, and $\Sigma = [S\ O] \in \mathbb{R}^{m\times n}$ with $S = \text{diag}(\sigma_1, \sigma_2, \ldots, \sigma_m)$. Assume $\text{rank}(J) = r$, there is

- $\sigma_1 \geq \sigma_2 \cdots \geq \sigma_r > \sigma_{r+1} = \cdots = 0$.
- $\mathcal{R}(J^T) = \text{span}\{v_1, \cdots, v_r\}$.
- $\mathcal{N}(J) = \text{span}\{v_{r+1}, \cdots, v_n\}$.

where $\mathcal{N}(A)$ is the null space of matrix A (or the kernel of matrix A), and $\mathcal{R}(A)$ is the range of matrix A [1]. With the SVD, P can be written as

$$P = I_n - J^{\dagger}J = \sum_{i=r+1}^{n} v_i v_i^T, \tag{10}$$

which can project a vector onto the null space of Jacobian. This ensures Pz in (7) is always in the null space of Jacobian, regardless what z is chosen, and the task velocity $\dot{x}$ will not be effected.

B. ACTIVATION OF TASKS AND DISCONTINUITY PHENOMENON

In dynamic environment, some tasks are not always in the active state during the whole motion of the robot. For example, the obstacle avoidance task is active, only when the robot is close to the obstacle. When the robot is a long distance away from the obstacle, the task does not need to be taken into consideration, namely, the corresponding Jacobian should be removed from (4).

An activation matrix is introduced to represent the activation of the each task, which is defined as:

$$H = \begin{bmatrix} h_1 I_{m_1} & O & \cdots & O \\ O & h_2 I_{m_2} & \cdots & O \\ O & O & \ddots & \vdots \\ O & O & \cdots & h_k I_{m_k} \end{bmatrix}, \tag{11}$$

where $h_i \in [0, 1]$ is the activation factor of the i^{th} task, which represents the activation of the i^{th} task. If $h_i \in (0,\ 1)$, the task is active, if $h_i = 1$, the task is fully active, if $h_i = 0$, the corresponding task is totally inactive. The kinematic equation under the activation matrix H is

$$HJ\dot{q} = H\dot{x}. \tag{12}$$

The activation matrix is used for removing the Jacobian of inactive task from the augmented Jacobian J smoothly. The pseudo-inverse can be adopted to solve (12), and the solutions can be formulated as:

$$\dot{q} = (HJ)^{\dagger}H\dot{x} \quad \text{or} \tag{13}$$

$$\dot{q} = (\tilde{H}J)^{\dagger}\tilde{H}\dot{x}, \tag{14}$$

where $\tilde{H}$ is a diagonal matrix, and is defined as:

$$\tilde{H} = \begin{bmatrix} \tilde{h}_1 I_{m_1} & O & \cdots & O \\ O & \tilde{h}_2 I_{m_2} & \cdots & O \\ O & O & \ddots & \vdots \\ O & O & \cdots & \tilde{h}_k I_{m_k} \end{bmatrix}, \tag{15}$$

where the diagonal element $\tilde{h}_i$ is defined as:

$$\tilde{h}_i = \begin{cases} 1 & \text{if } h_i \in (0, 1] \\ 0 & \text{if } h_i = 0. \end{cases} \tag{16}$$

Provided J is full rank, for the redundant robot, the rank of HJ (or $\tilde{H}J$) is not larger than n. The solution (13) and (14) are the least-norm solutions of $HJ\dot{q} = H\dot{x}$ and $\tilde{H}J\dot{q} = \tilde{H}\dot{x}$, respectively. From perspective of the system of linear equations, $\tilde{H}J\dot{q} = \tilde{H}\dot{x}$ is equivalent to (12), the solution (14) should also be the solution of (12), and vice versa. The solution (13) is equivalent to (14) for the redundant robot, according to the uniqueness property of the least-norm solution [21].

However, solution (13) and (14) are adopted to solve the task-based control problem with an activation task mechanism, and discontinuities arise, even though (13) ensures the continuity of HJ and $H\dot{x}$, as the activities of the tasks change. The reason is that pseudo-inverse operator is continuous when the rank of the matrix is constant, which is a sufficient condition to ensure the continuity of the solution [16]. For example, assume the task $\dot{x} = [\dot{x}_1^T,\ \dot{x}_2^T]^T$, as the task $\dot{x}_2$ evolves from inactive state to active state in the k^{th} control period, and the solution (14) is adopted, the activation matrix switches from

$$\tilde{H}(k) = \begin{bmatrix} I_{m_1} & O \\ O & O \end{bmatrix} \text{ to } \tilde{H}(k+1) = I_m,$$

and the corresponding solutions are:

$$\dot{q}_k = J^{\dagger}\dot{x}_1 \text{ and } \dot{q}_{k+1} = \begin{bmatrix} J_1 \\ J_2 \end{bmatrix}^{\dagger} \begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \end{bmatrix}.$$

Thereby, a discontinuity arises, as the activation matrix switches from $\tilde{H}(k)$ to $\tilde{H}(k+1)$. If the activation matrix acts in a continuous manner, the discontinuity still exists, since (13) is equivalent to (14) for the redundant robot.

Hence the fundamental issue for the task-based control with activation task mechanism is how to construct a continuous inverse operator. When $m < n$, the construction of a continuous inverse can be converted to the construction of a continuous null space projection operator based on the equation $J^{\dagger} = (I_n - P)J^{\dagger}$.

In [20], continuous projection operator of the i^{th} task is given as $P_{s_i} = I_n - h_i J_i^{\dagger} J_i$. For k tasks activated by H_k, the null

space projection operator of J is formulated as:

$$P_s = \prod_{i=1}^{i=k} P_{s_i}. \tag{17}$$

The projection operator of J is obtained by using successive projection onto null space of each task, which is endowed with the merits of simple formulation and low computational complexity. However, (17) is an approximate formulation of $I_n - J^\dagger J$, and the error of approximation increases as the number of the tasks increases. The algorithm is limited to the case of two tasks. In view of the merits of continuous projection operator obtained by successive projecting, we intend to construct a new continuous projection operator based on the successive projection in order to achieve a continuous behavior with a better performance.

III. ITERATIVELY SUCCESSIVE PROJECTION

A. ITERATIVELY SUCCESSIVE PROJECTION

In order to build a continuous null space projection operator, we first analyze the relation between the null space projection operator of augmented Jacobian and the null space of each subtask. And the activation mechanism of the tasks is ignored first. The null space projection operator P of J defined in (8) has the following properties [1], which will be used in the analysis.

Property 1: The following equalities hold

- $(I_n - J^\dagger J)^T = I_n - J^\dagger J$.
- $(I_n - J^\dagger J)(I_n - J^\dagger J) = I_n - J^\dagger J$.
- $\|I_n - J^\dagger J\| = 1$.

In this paper $\|\cdot\|$ represents the Euclidean norm. The relation between the null space of the augmented Jacobian and the null space of a single task Jacobian is discussed first, and the following lemmas are presented first:

Lemma 1: $(I - J_i^\dagger J_i)(I - J^\dagger J) = I - J^\dagger J$, *and* $(I - J^\dagger J)(I - J_i^\dagger J_i) = I - J^\dagger J$, $i = 1, 2, \cdots, k$.

Proof: Since the null space of augmented Jacobian $\mathcal{N}(J) \subseteq \mathcal{N}(J_i)$. For any vector $z \in \mathbb{R}^n$, there is $J_i(I - J^\dagger J)z = O$, namely, $J_i(I - J^\dagger J) = O$ holds. Thus, $J_i^\dagger J_i(I - J^\dagger J) = O$, and $(I - J_i^\dagger J_i)(I - J^\dagger J) = I - J^\dagger J$ can be obtained by adding $I - J^\dagger J$ on both sides of the equation.

Based on the symmetrical property of $I - J^\dagger J$, there is $I_n - J^\dagger J = (I_n - J^\dagger J)^T = ((I - J_i^\dagger J_i)(I - J^\dagger J))^T = (I - J^\dagger J)(I - J_i^\dagger J_i)$. □

Remark 1: Based on Lemma 1, by reformulating $(I - J_i^\dagger J_i)(I - J^\dagger J) = (I - J^\dagger J)(I - J_i^\dagger J_i)$. $J_i^\dagger J_i J^\dagger J = J^\dagger J J_i^\dagger J_i$ *is obtained. Substracting* $J^\dagger J$ *from the both sides yields the following equation:*

$$(I - J_i^\dagger J_i)J^\dagger J = J^\dagger J(I - J_i^\dagger J_i). \tag{18}$$

Lemma 2: $\left\| \prod_{i=1}^{k} (I - J_i^\dagger J_i) J^\dagger J \right\| < 1.$

Proof: If $J = O$ or any of $I - J_i^\dagger J_i$ equals to O, the result holds.

Otherwise, according to the Euclidean norm, it follows:

$$\left\| \prod_{i=1}^{k} (I - J_i^\dagger J_i) J^\dagger J \right\| \le \prod_{i=1}^{k} \left\| I - J_i^\dagger J_i \right\| \left\| J^\dagger J \right\| = 1. \tag{19}$$

To prove $\left\| \prod_{i=1}^{k} (I - J_i^\dagger J_i) J^\dagger J \right\| \neq 1$, the method of contradiction is adopted. Assume $\left\| \prod_{i=1}^{k} (I - J_i^\dagger J_i) J^\dagger J \right\| = 1$ holds, there exist a nonzero vector $z \in \mathbb{R}^n$, satisfying $\| \prod_{i=1}^{k} (I - J_i^\dagger J_i) J^\dagger J z \| = \|z\|$.

Considering one task, for any nonzero vector z, there is $\|z\|^2 = \|(I - J_i^\dagger J_i)z\|^2 + \|J_i^\dagger J_i z\|^2$, such that $\|(I - J_i^\dagger J_i)z\| = \|z\|$ holds, only when $z \in \mathcal{N}(J_i)$. Similarly, we obtain the result $\|J^\dagger J z\| = \|z\|$, only when $z \in \mathcal{R}(J^T)$. Since $\|I_n - J_i^\dagger J_i\| = 1$ and $\|J^\dagger J\| = 1$, $\| \prod_{i=1}^{k} (I - J_i^\dagger J_i) J^\dagger J z \| = \|z\|$, only when $z \in \mathcal{N}(J_1) \cap \cdots \cap \mathcal{N}(J_i) \cap \cdots \cap \mathcal{N}(J_k) \cap \mathcal{R}(J^T)$. Hence $\mathcal{N}(J_1) \cap \cdots \cap \mathcal{N}(J_i) \cap \cdots \cap \mathcal{N}(J_k) = \mathcal{N}(J), \mathcal{N}(J) \cap \mathcal{R}(J^T) = \emptyset$. Namely, there does not exist a nonzero vector z satisfying $\| \prod_{i=1}^{k} (I - J_i^\dagger J_i) J^\dagger J z \| = \|z\|$, which contradicts with the assumption. Therefore, $\| \prod_{i=1}^{k} (I - J_i^\dagger J_i) J^\dagger J \| < 1$ is proved. □

And the iteratively successive projection operator can be defined as follows:

$$P_N = \left(\prod_{i=1}^{k} (I - J_i^\dagger J_i) \right)^N, \tag{20}$$

where $N \in \mathbb{N}^*$ is the iteration times.

The iteratively successive projection operator can be obtained in two steps: first, multiply the classical null space projection operator of each subtask in sequence to obtain the successive projection operator; then the successive projection operator must be multiplied by itself N times to form the iteratively successive projection operator. And the successive projection operator P_s defined in (17) can be considered as a special case using iteratively successive projection principle, where the time N of successive projection is merely 1. According to Lemma 1 the following theorem holds.

Theorem 1: $\lim_{N \to \infty} P_N = I - J^\dagger J$.

Proof: According to (18), $\prod_{i=1}^{k} (I - J_i^\dagger J_i)$ can be reformulated as:

$$\begin{aligned} &\prod_{i=1}^{k} (I - J_i^\dagger J_i) \\ &\quad = J^\dagger J \prod_{i=1}^{k} (I - J_i^\dagger J_i) + (I - J^\dagger J) \prod_{i=1}^{k} (I - J_i^\dagger J_i) \\ &\quad = \prod_{i=1}^{k} (I - J_i^\dagger J_i) J^\dagger J + (I - J^\dagger J) \prod_{i=1}^{k} (I - J_i^\dagger J_i). \end{aligned} \tag{21}$$

Since $J^\dagger J(I - J^\dagger J) = O$, it implies $\prod_{i=1}^{k}(I - J_i^\dagger J_i)J^\dagger J(I - J^\dagger J)\prod_{i=1}^{k}(I - J_i^\dagger J_i) = O$. Based on (21), we obtain:

$$\begin{aligned}
&\lim_{N\to\infty}\left(\prod_{i=1}^{k}(I - J_i^\dagger J_i)\right)^N \\
&= \lim_{N\to\infty}\left(\prod_{i=1}^{k}(I - J_i^\dagger J_i)J^\dagger J + (I - J^\dagger J)\prod_{i=1}^{k}(I - J_i^\dagger J_i)\right)^N \\
&= \lim_{N\to\infty}\left(\prod_{i=1}^{k}(I - J_i^\dagger J_i)J^\dagger J\right)^N \\
&\quad + \lim_{N\to\infty}\left((I - J^\dagger J)\prod_{i=1}^{k}(I - J_i^\dagger J_i)\right)^N . \qquad (22)
\end{aligned}$$

According to Lemma 2, $\rho(\prod_{i=1}^{k}(I - J_i^\dagger J_i)J^\dagger J) \leq \| \prod_{i=1}^{k}(I - J_i^\dagger J_i)J^\dagger J \| < 1$, where $\rho(\cdot)$ denotes the spectral radius. Thus $\lim_{N\to\infty}(\prod_{i=1}^{k}(I - J_i^\dagger J_i)J^\dagger J)^N = O$. Based on Lemma 1, $(I - J^\dagger J)\prod_{i=1}^{k}(I - J_i^\dagger J_i) = I - J^\dagger J$ can be obtained. Therefore, $\lim_{N\to\infty}(\prod_{i=1}^{k}(I - J_i^\dagger J_i))^N = I - J^\dagger J$ holds. □

Theorem 1 demonstrates the relation between the null space of the augmented Jacobian and the successive projection between null space of each subtask. Iteratively successive projection operator consists of two parts: one part equals to the classical null space projection operator defined in (10), the other part is the exponentiation of a matrix with spectral radius less than 1. As the iteration times increase to infinity, the second part converges to a zero matrix, where the iteratively successive projection operator equals to the classical null space projection operator.

There is a geometric interpretation for iteratively successive projection method: on the condition that the spectral radius of $I_n - J_i^\dagger J_i$ is less than 1, if a vector is successively projected onto the null space of each task's Jacobian, the vector component out of the intersection of each task's null space will be diminished gradually. By repeating successive projection iteratively, the component of the vector out of the intersection of each task's null space will be totally removed finally. Fig. 1 illustrates the iteratively successive projection between two tasks. Vector $v \in \mathbb{R}^3$, subtask Jacobian $J_1 \in \mathbb{R}^{1\times 3}$ and $J_2 \in \mathbb{R}^{1\times 3}$, plane A and B are perpendicular to J_1 and J_2, respectively. v_1 is the projection of v onto the plane B, which equals $(I_n - J_2^\dagger J_2)v$. Similarly, projecting v_1 onto the plane A yields $(I_n - J_1^\dagger J_1)(I_n - J_2^\dagger J_2)v$. By repeating the successive projection infinite times, the vector converges to v_p finally, which equals the classical null space projection $(I_n - J^\dagger J)v$.

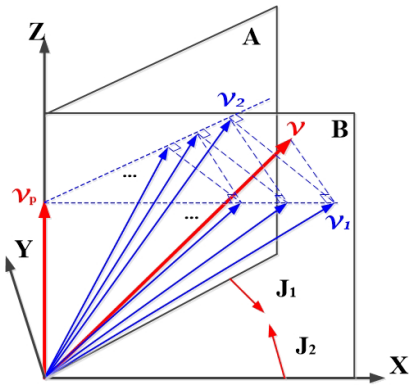


FIGURE 1. A geometric interpretation for iteratively successive projection with two subtasks.

B. CONTINUOUS PROJECTION OPERATOR

The case without activation mechanism is discussed in the previous section, in this section activation factors are introduced into (20) to obtain the continuous projection operator, which can be formulated in the following form:

$$P_{NH} = \left(\prod_{i=1}^{k}(I_n - h_i J_i^\dagger J_i)\right)^N , \qquad (23)$$

where h_i is the diagonal element of matrix H defined in (11), which represents activation of the i^{th} task Jacobian J_i.

For the projection operator $I_n - h_i J_i^\dagger J_i$, h_i determines whether the projection onto the row space of J_i is totally or partially removed. Therefore, the values of h_i directly affect the continuity of P_{NH}, however, the value of the iteration times N can affect the continuity of P_{NH} either, and the following two properties hold.

Property 2: Assume N is finite, P_{NH} is continuous with respect to the activation factors.

Proof: P_{NH} can be reformulated as a polynomial form of h_i. However, a polynomial is always continuous on the condition that N is finite, which ensures the continuity of P_{NH} with respect to the activation factors. □

Property 3: Assume N is infinite, P_{NH} is discontinuous with respect to the activation factors. And there is $\lim_{N\to\infty} P_{NH} = I_n - (\tilde{H}J)^\dagger \tilde{H}J$.

Proof: The residual item between P_{NH} and $I_n - (\tilde{H}J)^\dagger \tilde{H}J$ is

$$\begin{aligned}
R &= P_{NH} - (I_n - (\tilde{H}J)^\dagger \tilde{H}J) \\
&= (I_n - (\tilde{H}J)^\dagger \tilde{H}J)P_{NH} + (\tilde{H}J)^\dagger \tilde{H}JP_{NH} \\
&\quad - (I_n - (\tilde{H}J)^\dagger \tilde{H}J). \qquad (24)
\end{aligned}$$

Based on Lemma 1, it implies

$$(I_n - (\tilde{H}J)^\dagger \tilde{H}J)(I_n - h_i J_i^\dagger J_i) = I_n - (\tilde{H}J)^\dagger \tilde{H}J . \qquad (25)$$

According to the definition of P_{NH}, substituting (25) into (24) yields

$$R = (\tilde{H}J)^\dagger \tilde{H}JP_{NH} . \qquad (26)$$

Equation $(\tilde{H}J)^{\dagger}\tilde{H}J(I_n - h_i J_i^{\dagger}J_i) = (I_n - h_i J_i^{\dagger}J_i)(\tilde{H}J)^{\dagger}\tilde{H}J$ can be obtained in the same way as (18). According to the definition of pseudo-inverse, there is $(\tilde{H}J)^{\dagger}\tilde{H}J(\tilde{H}J)^{\dagger} = (\tilde{H}J)^{\dagger}$. Based on those equations, (26) can be rewritten as:

$$\begin{aligned} R &= (\tilde{H}J)^{\dagger}\tilde{H}J\left(\prod_{i=1}^{k}(I_n - h_i J_i^{\dagger}J_i)\right)^N \\ &= \left((\tilde{H}J)^{\dagger}\tilde{H}J\right)^N\left(\prod_{i=1}^{k}(I_n - h_i J_i^{\dagger}J_i)\right)^N \\ &= \left(\prod_{i=1}^{k}(I_n - h_i J_i^{\dagger}J_i)(\tilde{H}J)^{\dagger}\tilde{H}J\right)^N. \end{aligned} \tag{27}$$

Since $\|I_n - h_i J_i^{\dagger}J_i\| \leq 1$ and $\|(\tilde{H}J)^{\dagger}\tilde{H}J\| \leq 1$, $\|\prod_{i=1}^{k}(I_n - h_i J_i^{\dagger}J_i)(\tilde{H}J)^{\dagger}\tilde{H}J\| \leq 1$. $\|\prod_{i=1}^{k}(I_n - h_i J_i^{\dagger}J_i)(\tilde{H}J)^{\dagger}\tilde{H}J\| < 1$ can be proved in the same way as Lemma 2. Hence $\lim_{N\to\infty} R = O$, there is $\lim_{N\to\infty} P_{NH} = I_n - (\tilde{H}J)^{\dagger}\tilde{H}J$.

According to [16], $I_n - (\tilde{H}J)^{\dagger}\tilde{H}J$ is discontinuous with respect to the activation factors, which proves the property. □

Property 3 also demonstrates the relation between P_{NH} and $I_n - (\tilde{H}J)^{\dagger}\tilde{H}J$. P_{NH} consists of two parts: one part equals to $I_n - (\tilde{H}J)^{\dagger}\tilde{H}J$; the other part is the residual item R defined in (24), which ensures the continuity of P_{NH} as h_i switch between the zero value and the nonzero value. However, R converges to zero matrix, as N increases to infinity, and discontinuity happens. Hence, the value of N determines the continuity of P_{NH}.

IV. CONTINUOUS ALGORITHM FOR TASK-BASED CONTROL USING THE ITERATIVELY SUCCESSIVE PROJECTION

A. CONSTRUCTION OF A CONTINUOUS INVERSE OPERATOR

In view of that $J^{\dagger} = (I_n - P)J^{\dagger}$ holds for the redundant robot, a continuous inverse operator of J activated by H can be defined on the basis of the definition of the continuous projection operator as follows:

$$J^{\#H} = (I_n - P_{NH})J^{\dagger}, \tag{28}$$

where H is defined in (11), and P_{NH} is defined in (23). Since $J^{\dagger}$ is continuous, $J^{\#H}$ is continuous according to Property 2, on the condition that the iteration times N of P_{NH} is finite.

Note that the solution based on the classical inverse for the kinematic equation (12) is $(HJ)^{\dagger}H$, which is featured with the property $HJ(HJ)^{\dagger}H = H$ (this holds for $(\tilde{H}J)^{\dagger}\tilde{H}$ either). This endows the corresponding solution the ability to ensure all the active tasks, if the total dimension of all active tasks is not larger than the DoF of the robot. However, the continuous inverse operator $J^{\#H}$ is the approximation of $(\tilde{H}J)^{\dagger}\tilde{H}$, which is equivalent to $(HJ)^{\dagger}H$. To evaluate the approximate degree between $J^{\#H}$ and $(\tilde{H}J)^{\dagger}\tilde{H}$, the difference between $\tilde{H}JJ^{\#H}$ and $\tilde{H}J(\tilde{H}J)^{\dagger}\tilde{H}$ is calculated as follows:

$$\begin{aligned} &\tilde{H}JJ^{\#H} - \tilde{H}J(\tilde{H}J)^{\dagger}\tilde{H} \\ &= \tilde{H}J(I_n - P_{NH})J^{\dagger} - \tilde{H} \\ &= \tilde{H}J\left((\tilde{H}J)^{\dagger}\tilde{H}J - R\right)J^{\dagger} - \tilde{H} \\ &= \tilde{H}J(\tilde{H}J)^{\dagger}\tilde{H}JJ^{\dagger} - \tilde{H}JRJ^{\dagger} - \tilde{H} \\ &= -\tilde{H}JRJ^{\dagger}, \end{aligned} \tag{29}$$

where R is defined in (26). The proof of Property 3 demonstrates that R converges to a zero matrix as the value of N increases to infinity. On the condition that J is nonsingular, $\|J\|$ and $\|J^{\dagger}\|$ are both bounded, $\tilde{H}JJ^{\#H}$ will gradually converge to $\tilde{H}$ as the value of N increases. This implies the value of N balances the continuity and the approximate degree, a small value of N leads to a smooth behavior but a poor approximation to $\tilde{H}JJ^{\#H}$; a high value of N leads to a discontinuous behavior but a good approximation to $\tilde{H}JJ^{\#H}$.

B. CONTINUOUS CONTROL LAW FOR THE TASK-BASED CONTROL

Assume the total dimension of all active tasks is not larger than the DoF of the robot, a continuous solution to the kinematic equation (12) is proposed based on the definition of the continuous inverse (28) as follows:

$$\dot{q} = J^{\#H}\dot{x}. \tag{30}$$

Because $J^{\#H}$ is an approximation to $(\tilde{H}J)^{\dagger}\tilde{H}$, and the open-loop control (30) based on $J^{\#H}$ cannot guarantee the active task. Given a desired end-effector trajectory in terms of position, velocity, a closed-loop control law is presented to improve the accuracy as follows:

$$\dot{q} = J^{\#H}(\dot{x}_d - K_e e) = (I_n - P_{NH})J^{\dagger}(\dot{x}_d - K_e e), \tag{31}$$

where $e = x - x_d$ is defined as the tracking error, x_d and $\dot{x}_d$ are the position and velocity of the desired trajectory, respectively. And the diagonal matrix K_e is the error-feedback gain matrix, whose diagonal elements are positive.

According to the task-based control with the activation mechanism, some tasks are not active, only the active tasks are considered in stability analysis. Hence, considering the interval when $\tilde{H}$ is constant, the stability of the active task tracking error $\tilde{e} = \tilde{H}e$ is discussed, and the following theorem holds.

Theorem 2: Assume J is full rank, the active task tracking error is bounded under the control law (31).

Proof: At the beginning, let the Lyapunov function be

$$V = \frac{1}{2}\tilde{e}^T P\tilde{e}, \tag{32}$$

where the constant diagonal matrix P is positive definite.

First, the boundedness of the tracking error in the period without task activity changing is analyzed. Without loss of generality, assume the activities of all tasks keep steady in the interval $[t_0, t_1]$. Since the activities of the tasks do not change,

$\tilde{H}$ is constant within the period, the time derivative of V can be written as

$$\begin{aligned}\dot{V} &= \tilde{e}^T P\tilde{H}\left(JJ^{\#H}(\dot{x}_d - K_e e) - \dot{x}_d\right)\\ &= \tilde{e}^T P\Big(\big(\tilde{H}J(\tilde{H}J)^\dagger \tilde{H}JJ^\dagger - \tilde{H}JRJ^\dagger\big)(\dot{x}_d - K_e e) - \tilde{H}\dot{x}_d\Big)\\ &= -\tilde{e}^T PK_e\tilde{e} - \tilde{e}^T P\tilde{H}JRJ^\dagger(\dot{x}_d - K_e e). \end{aligned} \tag{33}$$

Since $\tilde{H}J(\tilde{H}J)^\dagger\tilde{H}J = \tilde{H}J$, according to (18) and (26), the following equation holds

$$R = R(\tilde{H}J)^\dagger\tilde{H}J. \tag{34}$$

Substituting (34) into (33) yields

$$\begin{aligned}\dot{V} &= -\tilde{e}^T PK_e\tilde{e} - \tilde{e}^T P\tilde{H}JR(\tilde{H}J)^\dagger\tilde{H}JJ^\dagger(\dot{x}_d - K_e e)\\ &= -\tilde{e}^T PK_e\tilde{e} - \tilde{e}^T P\tilde{H}JR(\tilde{H}J)^\dagger(\tilde{H}\dot{x}_d - K_e\tilde{e}). \end{aligned} \tag{35}$$

On the condition that J is full rank, J is bounded. According to Weyl's theorem [21], $\sigma_m \le \|\tilde{H}J\| \le \sigma_M$ holds, where σ_M and σ_m denote the maximum singular value and the minimum singular value of J, respectively. Therefore, Eq. (35) can be simplified as:

$$\begin{aligned}\dot{V} &\le -\alpha_m k_{em}\|\tilde{e}\|^2 + \alpha_M k_{eM}\|\tilde{H}J\|\|R\|\|(\tilde{H}J)^\dagger\|\|\tilde{e}\|^2\\ &\quad + \alpha_M\|\tilde{H}J\|\|R\|\|(\tilde{H}J)^\dagger\|\|\tilde{H}\dot{x}_d\|\|\tilde{e}\|\\ &= -(\alpha_m k_{em} - \alpha_M k_{eM}\frac{\sigma_M}{\sigma_m}r)\|\tilde{e}\|^2 + \alpha_M\frac{\sigma_M}{\sigma_m}rv_d\|\tilde{e}\|, \end{aligned} \tag{36}$$

where α_m (α_M) and k_{em} (k_{eM}) denote the minimum (maximum) element of the diagonal matrix P and K_e, respectively. v_d is the upper bound of $\|\tilde{H}\dot{x}_d\|$, r is the upper bound of $\|R\|$.

According to the proof of Property 3, $\|R\|$ converges to 0 as the value of N increases. Thus, by properly selecting N and K_e, the following equation holds

$$r < \frac{\sigma_m\alpha_m k_{em}}{\sigma_M\alpha_M k_{eM}}. \tag{37}$$

And (36) can be rewritten as

$$\dot{V} \le -\gamma_1(\|\tilde{e}\| - \gamma_2)\|\tilde{e}\|, \tag{38}$$

where

$$\gamma_1 = \frac{\sigma_m\alpha_m k_{em} - \sigma_M\alpha_M k_{eM} r}{\sigma_m}, \tag{39}$$

$$\gamma_2 = \frac{\sigma_M\alpha_M r v_d}{\sigma_m\alpha_m k_{em} - \sigma_M\alpha_M k_{eM} r}. \tag{40}$$

This implies the boundedness of the norm of $\|\tilde{e}\|$. If $\|\tilde{e}\| > \gamma_2$, $\dot{V} < 0$, $\|\tilde{e}\|$ decreases until γ_2. Once $\|\tilde{e}(t_0)\| \le \gamma_2$, then $\|\tilde{e}(t)\| \le \gamma_2$, $t \in [t_0,\ t_1]$. It demonstrates $\|\tilde{e}(t)\| \le \max\{\|\tilde{e}(t_0)\|, \gamma_2\},, t \in [t_0,\ t_1]$.

Then, the case of task activity switching is considered. Assume the activity of the i^{th} task switches at the moment $t = t_1$, the active task tracking error before and after the task switching are denoted as $\tilde{e}(t)|_{t_1-}$ and $\tilde{e}(t)|_{t_1+}$, respectively. If the i^{th} task switches from active state to inactive state at the moment $t = t_1$, the corresponding element is removed from the active task tracking error $\tilde{e}$, therefore, $\|\tilde{e}(t)|_{t_1+}\| \le \|\tilde{e}(t)|_{t_1-}\|$. If the i^{th} task switches from inactive state to active state, the corresponding element in the active task tracking error equals 0 at the moment $t = t_1$, because the initial tracking task error should be zero as it is activated, consequently, $\|\tilde{e}(t)|_{t_1+}\| = \|\tilde{e}(t)|_{t_1-}\|$. Thus, the activity switching of a single task would not increase the value of $\|\tilde{e}\|$. And the same conclusion can be drawn when several tasks switch at the same moment. Since the initial value of $\|\tilde{e}\| = 0$, the upper bound of $\|\tilde{e}\|$ is γ_2, which proves the theorem. □

Compared with the control law (31), the the control law proposed in [16] can only guarantee the convergence of the active task tracking error in the activity steady period (all tasks are inactive or fully active) while the proposed control law can ensure the boundedness of the active task tracking error in the whole tracking process.

Remark 2: The proof of Theorem 2 implies that the bound of $\|\tilde{e}\|$ is influenced by the value $\|R\|$: the larger the value of $\|R\|$ is, the smaller the bound of $\|\tilde{e}\|$ is. According to the analysis in Sec. III-B, the value of $\|R\|$ converges to 0, as N increases. Hence selecting N is also a compromise between continuity and stability. A large N leads to a small bound of $\|\tilde{e}\|$ at the price of poor continuity as task switches from inactivity to activity, or vice versa, while a small N makes joint velocity act in a continuous manner with a large active task tracking error. The following method can be adopted to determine the lower bound of N. Assume the desired bound of $\|\tilde{e}\|$ is b_e, the upper bound of $\|\prod_{i=1}^{k}(I_n - h_i J_i^\dagger J_i)(\tilde{H}J)^\dagger\tilde{H}J\|$ is denoted as b_r. According to (27) *and* (40)*, $\|\tilde{e}\| \le b_e$ holds, if the following inequality holds*

$$b_e \ge \frac{\sigma_M\alpha_M b_r^N v_d}{\sigma_m\alpha_m k_{em} - \sigma_M\alpha_M k_{eM} b_r^N}. \tag{41}$$

Since $\|b_r\| < 1$, N can ensure $\|\tilde{e}\| \le b_e$ on the condition that the following inequality holds:

$$N \ge \log_{b_r}\frac{b_e\alpha_m\sigma_m k_{em}}{\sigma_M\alpha_M(b_e k_{eM} + v_d)}. \tag{42}$$

C. COMPUTATION OF THE CONTROL LAW

Since the definition of $J^{\#H}$ is based on the pseudo-inverse of J, the closed-loop control law (31) might suffer from the singularity problem, as the rank of J loses. The damping factor is introduced to obtain a feasible solution at the cost of a lower precision, as proposed in [22]. The solution with a damping factor can be formulated as:

$$\dot{q} = (I_n - P_{NH})J^T(JJ^T + \lambda^2 I)^{-1}(\dot{x}_d - K_e e). \tag{43}$$

where the damping factor $\lambda > 0$ is set as follows:

$$\lambda^2 = \begin{cases} 0 & \sigma_{\min} \ge \epsilon \\ \left(1 - \left(\frac{\sigma_{\min}}{\epsilon}\right)^2\right)\lambda_{\max}^2 & \text{otherwise,} \end{cases} \tag{44}$$

where $\sigma_{\min}$ denotes the minimal non-zero singular value of HJ, ϵ is the length of the singular interval to avoid, $\lambda_{\max}$ denotes the maximum damping parameter, which are selected in the same way as [23]. Note that the pseudo-inverse is also used in (23) to obtain the null-space projection operator of each task. Nevertheless, the calculation of the null space

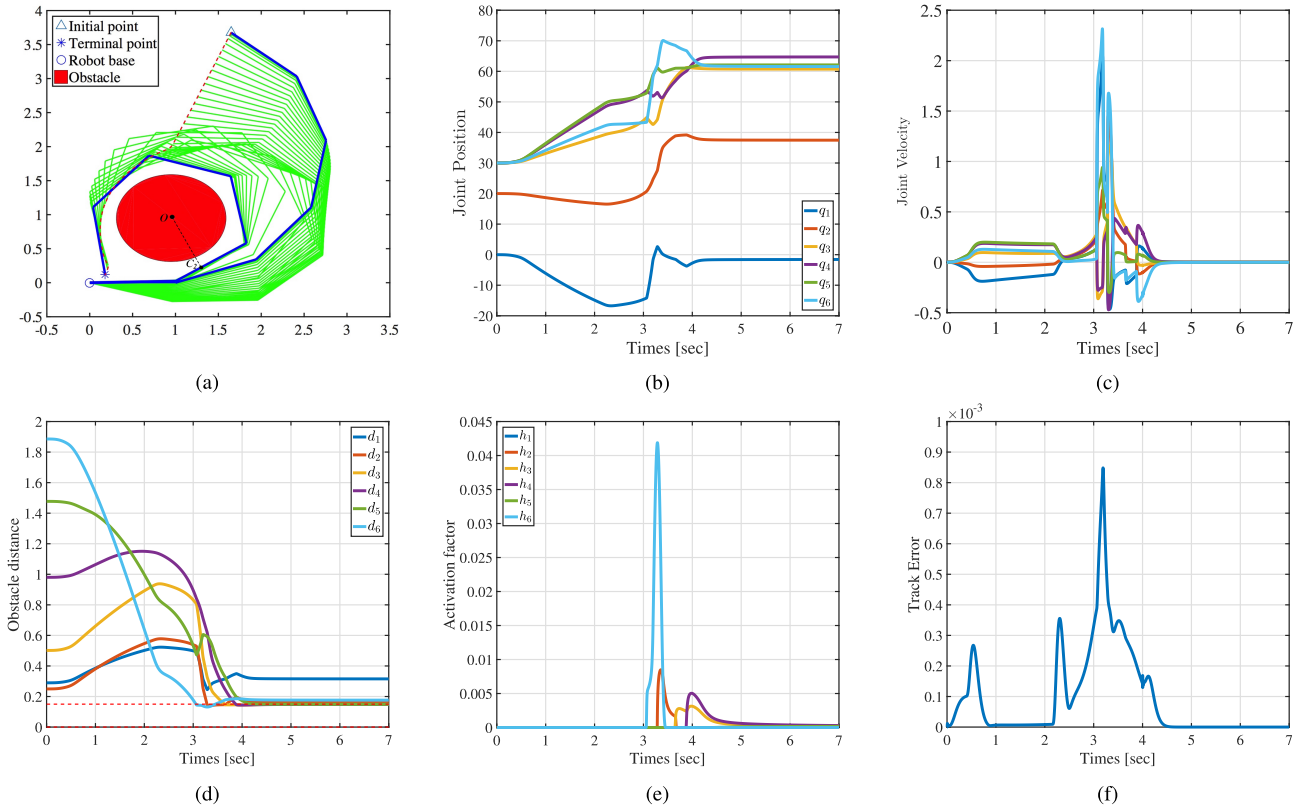


FIGURE 2. Simulation of the redundant manipulator under the control algorithm based on the classical inverse. (a) Motion of the manipulator. (b) Joint position trajectory. (c) Joint velocity trajectory. (d) Distance to the obstacle. (e) Activation Factor. (f) Tracking error.

projection operator of each task will not be effected by the singularity, which can be illustrated according to (10).

It takes N times successive projecting to obtain the P_{NH}, according to (23). Since the control law (43) is defined based on P_{NH}, it is computationally intensive to obtain $J^{\#H}$ using (23) and (28). For simplicity, N is set 2^n, and algorithm 1 details the computation of the continuous inverse $J^{\#H}$.

Algorithm 1 Calculate $\dot{q}$

1: Calculate the activation factor h_i and Jacobian matrix J_i of each task.
2: Set $G = I_n$
3: **for** $i = 1; i \leq k; i++$ **do**
4: $\quad G = G \times (I_n - h_i J_i^{\dagger} J_i)$;
5: **end for**
6: **for** $i = 1; i \leq n; i++$ **do**
7:

$$P_{NH} = \begin{cases} G & i = 1 \\ P_{NH} \times P_{NH} & i > 1. \end{cases}$$

8: **end for**
9: Calculate λ by (44).
10: Calculate $\dot{q} = (I_n - P_{NH})J^T(JJ^T + \lambda^2 I)^{-1}(\dot{x}_d - K_e e)$.

V. SIMULATION RESULTS

To validate the control algorithm proposed in the previous section, simulations are conducted on 6-DoF planar manipulator to perform the trajectory tracking task with obstacle. In trajectory-tracking process, the obstacle avoiding task is activated, only when the link is close to the obstacle. The comparison with the classical inverse method and the Mansard's methods are presented to demonstrate the advantage of the proposed control algorithm.

All link lengths are 1, and the initial configuration is set as $q_i = [0.0\ 20.0\ 30.0\ 30.0\ 30.0\ 30.0]^T$. The coordinate of the terminal point in the Cartesian space is $x_f = [0.18\ 0.12]^T$, and the center of column obstacle is point O at (0.95 0.95) with the radius $r_o = 0.66$. The desired trajectory of the end-effector is preplanned with a high-order polynomial function. As the end-effector tracks the trajectory, all the links have the risk of colliding to the obstacle. And the distance from the i^{th} link to the obstacle can be obtained as follows:

$$d_i = \sqrt{(C_{ix} - O_x)^2 + (C_{iy} - O_y)^2} - r_o, \tag{45}$$

where C_{ix} and C_{iy} are coordinate of the point $C_i(t)$ on the i^{th} link, which is the closet to the obstacle, shown in Fig. 2(a). And O_x, O_y is the center of the obstacle, r_o is the radius of the obstacle. $d_i > 0$ should hold to guarantee the i^{th} link not to collide the obstacle. In order to avoid collision, the obstacle

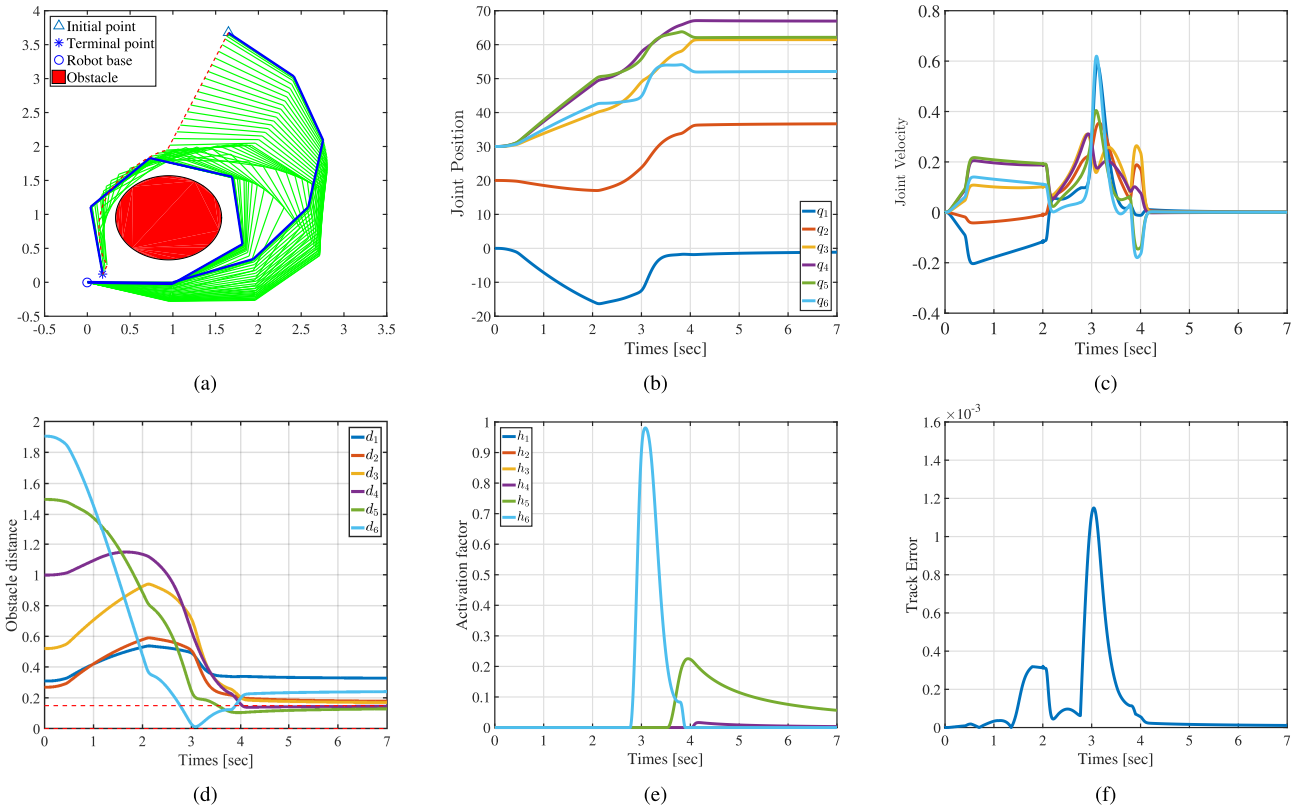


FIGURE 3. Simulation of the redundant manipulator under the control algorithm based on the iteratively successive projection. (a) Motion of the manipulator. (b) Joint position trajectory. (c) Joint velocity trajectory. (d) Distance to the obstacle. (e) Activation Factor. (f) Tracking error.

avoidance task of the i^{th} link can be formulated as:

$$J_o^i \dot{q} = \frac{\overrightarrow{OC_i}^T}{\|\overrightarrow{OC_i}\|} \cdot J_{ci}\dot{q} = v \quad i = 1, \cdots, 6 \tag{46}$$

where J_{ci} is the Jacobian of the point $C_i(t)$ on the i^{th} link, $\overrightarrow{OC_i}$ is the vector from the point O to point $C_i(t)$. And $v > 0$ is obstacle avoiding velocity. Since only four degrees of redundancy are available for the obstacle avoidance of six links, the activation mechanism is adopted for the obstacle avoiding tasks, the activation factor h_i is introduced as follows:

$$h_i = \begin{cases} 0 & \text{if } d_i > \tilde{d} \\ f_a\left(\dfrac{d_i}{\tilde{d}}\right) & \text{if } 0 \le d_i \le \tilde{d} \\ 1 & \text{else,} \end{cases} \tag{47}$$

where $\tilde{d}$ denotes the region where the task of avoiding collision becomes active, and the function $f_a(x)$ is defined as:

$$f_a(x) = 3x^2 - 2x^3, \quad \forall x \in [0,\ 1]. \tag{48}$$

The obstacle avoidance tasks of the links and tracking task are combined together, the augmented Jacobian and the velocity vector of the tasks are set as follows:

$$J = \begin{bmatrix} J_e \\ J_o^1 \\ \vdots \\ J_o^6 \end{bmatrix}, \quad \dot{x} = \begin{bmatrix} \dot{x} \\ v \\ \vdots \\ v \end{bmatrix} \tag{49}$$

where J_e is the Jacobian of the end-effector. And the activation matrix is set as $H = \text{diag}\{I_2,\ h_1,\ \cdots,\ h_6\}$. Therefore, the trajectory tracking problem with an obstacle avoiding task can be formulated as (12). In the simulation, the control period is set $T = 5$ ms, the error-feedback gain matrix $K_e = 20I_n$, and the avoiding collision region length $\tilde{d} = 0.15$.

A. THE CLASSICAL INVERSE

First, the control algorithm based on the classical inverse is adopted to solve the problem. In view of the singular case, the damping factor is introduced to (13), and the control law is formulated as:

$$\dot{q} = J^T H(HJJ^T H + \lambda^2 I)^{-1} H\dot{x}. \tag{50}$$

where is a damping factor λ is defined the same as (44), ϵ is the length of the singular interval to avoid, which is set to 0.005 in the simulation.

The results are shown in Fig. 2. The end-effector tracked the preplanned trajectory in the Cartesian space with a

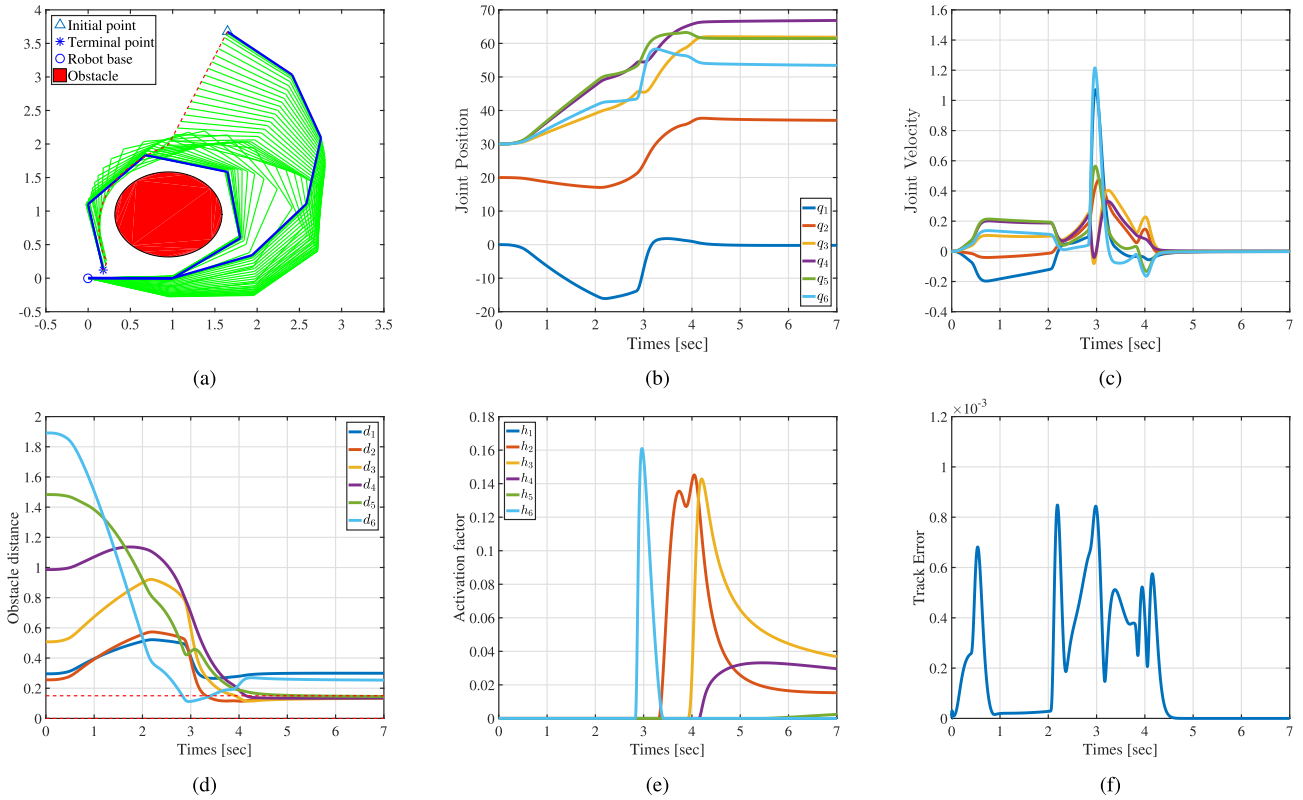


FIGURE 4. Simulation of the redundant manipulator under the control algorithm based on Mansard's inverse (a) Motion of the manipulator. (b) Joint position trajectory. (c) Joint velocity trajectory. (d) Distance to the obstacle. (e) Activation Factor. (f) Track error.

smooth path. At the interval $t \in [0,\ 3.1)$ s, all the links were far away from the obstacle. Correspondingly, only the trajectory tracking task was fully activated, the rank of HJ was 2, and the joint velocity was smooth. At $t = 3.1$ s, the sixth link of the manipulator came into the avoiding collision region, shown in Fig. 2(d), the obstacle avoiding task became active, which suppressed the further motion of the six link to the obstacle. Consequently, the rank of the matrix HJ increased to 3 abruptly, and the rank discontinuity of HJ caused the drastic switch in joint velocity. Similar discontinuous situations happened at the moment $t = 3.3$ s, 3.5 s, 3.7 s, 3.9 s, when an obstacle avoiding task switched from inactivity to activity or vice versa, shown in Fig. 2(c). This demonstrates the discontinuity of the classical inverse. Although there were several obstacle avoiding tasks happened during the trajectory tracking process, however, there were not more than four obstacle avoiding tasks were activated simultaneously, the rank of HJ was always less than 6, there was not any singularity happened, and the tracking error converged as the end-effector reached the terminal point.

B. THE ITERATIVELY SUCCESSIVE PROJECTION

In this section, the proposed control algorithm is adopted to solve the manipulator control problem. P_{NH} is obtained by Algorithm 1, the iteration parameter n in Algorithm 1 is set 10 based on (42).

The result is shown in Fig. 3. At the beginning, the tasks of obstacle avoidance were not active, all joints were only driven for the trajectory tracking task. At the moment $t = 2.8$ s, the sixth link came close to the obstacle, the corresponding task of obstacle avoidance started to come active, and the activation factor increased gradually, which did not affect the continuity of the joint velocity. Compared with the classical inverse, the activation factor of the proposed method is much larger. The reason for this phenomena is that the proposed method did not fully activate the obstacle avoiding subtask to guarantee the continuity, as the link came near the obstacle. However, the classical inverse method fully activated the avoiding subtasks to push the corresponding links out of the avoiding collision regions immediately. This is the reason why the classical inverse method took a shorter time to push the link away from the obstacle than proposed method.

It is worth noting that the track error was about 1.1×10^{-5}, as the end-effector came to the terminal point, shown in Fig. 3(f). However, this is not caused by the singularity, but instead it is due to the iteratively successive projection, which does not converge of the tracking error. Meanwhile iteratively successive projection guarantees the boundedness of the track error, which can keep the track error in an acceptable region, and according to (40), the bounded of the tracking error can be further reduced by increasing the error-feedback gain.

C. THE MANSARD'S INVERSE

For comparison, the Mansard's inverse proposed in [16] is adopted in the simulation. The control law can be written as:

$$J_{co}^{\oplus H} = \sum_{\beta \in \phi} \left(\prod_{i \in \beta} h_i \right) \prod_{i \notin \beta} (1 - h_i) J_\beta^T (J_\beta J_\beta^T + \lambda^2 I)^{-1}, \quad (51)$$

where ϕ are the set of all task combinations without repetitions. And the Jacobian matrix J_β of all the tasks included in the task set β, damping factor λ is set in the same way as in Sec. IV-C. The result is shown in Fig. 4, which demonstrates the approach based on the Mansard's inverse can successfully realize the trajectory tracking task, while guarantees the obstacle avoidance tasks.

TABLE 1. Simulation time of the methods based on different inverse.

	total time [s]	average time [ms]
Classical inverse	1.452	1.037
The proposed inverse	2.402	1.716
Mansard's inverse	62.893	44.924

Table 1 reveals that the calculation time of the Mansard's approach (51) is 25 times more than that of the control law based on iteratively successive projection and classical inverse. Mansard's inverse is the linear weighted sum of pseudo-inverse of all task combinations. Suppose there are l active tasks, the Mansard's inverse based method needs $2^l - 1$ times pseudo-inverse operation to calculate Mansard's inverse. While the proposed method needs $l + 1$ pseudo-inverse operation and n times matrix multiplications to calculate the inverse based iteratively successive projection. Compared with the continuous algorithm based on iteratively successive projection, the Mansard's inverse based approach is shown to be more computationally intensive, as the number of active tasks increase.

VI. CONCLUSION

This paper proposes an iteratively successive projection method, where the space is projected onto the null space of each task's null space in sequence iteratively to obtain the null space projection operator. This guarantees the continuity of the null space projection operator as the tasks switch over between inactive and active states by taking the finite-times iteration. Based on the iteratively successive projection method, a continuous inverse operator is built. A continuous closed-loop control algorithm is presented by means of the continuous inverse operator, which can ensure the continuity of the joint motion and the boundedness of the tracking errors of the active task. Compared with the classical task-based control algorithm and Mansard's unified approach, simulation results demonstrate the advantages of the proposed approach in continuity, stability and calculation-time consumption.

REFERENCES

[1] B. Siciliano and O. Khatib, *Springer Handbook of Robotics*. New York, NY, USA: Springer-Verlag, 2008.

[2] Y. Nakamura, H. Hanafusa, and T. Yoshikawa, "Task-priority based redundancy control of robot manipulators," *Int. J. Robot. Res.*, vol. 6, no. 2, pp. 3–15, 1987.

[3] J. Xiang, C. Zhong, and W. Wei, "General-weighted least-norm control for redundant manipulators," *IEEE Trans. Robot.*, vol. 26, no. 4, pp. 660–669, Aug. 2010.

[4] M. Keshmiri and W.-F. Xie, "Image-based visual servoing using an optimized trajectory planning technique," *IEEE/ASME Trans. Mechatronics*, vol. 22, no. 1, pp. 359–370, Feb. 2017.

[5] T. Liu *et al.*, "Iterative Jacobian-based inverse kinematics and open-loop control of an mri-guided magnetically actuated steerable catheter system," *IEEE/ASME Trans. Mechatronics*, vol. 22, no. 4, pp. 1765–1776, Aug. 2017.

[6] A. Liégeois, "Automatic supervisory control of the configuration and behavior of multibody mechanisms," *IEEE Trans. Syst., Man, Cybern., Syst.*, vol. SMCC-7, no. 12, pp. 868–871, Dec. 1977.

[7] A. A. Maciejewski and C. A. Klein, "Obstacle avoidance for kinematically redundant manipulators in dynamically varying environments," *Int. J. Robot. Res.*, vol. 4, no. 3, pp. 109–117, 1985.

[8] O. Egeland, "Task-space tracking with redundant manipulators," *IEEE J. Robot. Autom.*, vol. RA-3, no. 5, pp. 471–475, Oct. 1987.

[9] P. Chiacchio, S. Chiaverini, L. Sciavicco, and B. Siciliano, "Closed-loop inverse kinematics schemes for constrained redundant manipulators with task space augmentation and task priority strategy," *Int. J. Robot. Res.*, vol. 10, no. 4, pp. 410–425, 1991.

[10] T. F. Chan and R. V. Dubey, "A weighted least-norm solution based scheme for avoiding joint limits for redundant joint manipulators," *IEEE Trans. Robot. Autom.*, vol. 11, no. 2, pp. 286–292, Apr. 1995.

[11] P. Jiang, J. Xiang, W. Wei, and C. Shan, "General-weighted least-norm control for redundant manipulators under time-dependent constraint," *Int. J. Adv. Robot. Syst.*, vol. 12, no. 5, 2015. doi: 10.5772/60128.

[12] S. Huang, J. Xiang, W. Wei, and M. Z. Q. Chen, "On the virtual joints for kinematic control of redundant manipulators with multiple constraints," *IEEE Trans. Control Syst. Technol.*, vol. 26, no. 1, pp. 65–76, Jan. 2018.

[13] P. Jiang, S. Huang, J. Xiang, and M. Z. Q. Chen, "A unified approach for second-order control of the manipulator with joint physical constraints," *J. Mech. Robot.*, vol. 9, no. 4, 2017, Art. no. 041009.

[14] N. Garcia-Aracil, E. Malis, R. Aracil-Santonja, and C. Perez-Vidal, "Continuous visual servoing despite the changes of visibility in image features," *IEEE Trans. Robot.*, vol. 21, no. 6, pp. 1214–1220, Dec. 2005.

[15] A. Remazeilles, N. Mansard, and F. Chaumette, "A qualitative visual servoing to ensure the visibility constraint," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst. (IROS)*, Oct. 2006, pp. 4297–4303.

[16] N. Mansard, A. Remazeilles, and F. Chaumette, "Continuity of varying-feature-set control laws," *IEEE Trans. Autom. Control*, vol. 54, no. 11, pp. 2493–2505, Nov. 2009.

[17] J. Lee, N. Mansard, and J. Park, "Intermediate desired value approach for task transition of robots in kinematic control," *IEEE Trans. Robot.*, vol. 28, no. 6, pp. 1260–1277, Dec. 2012.

[18] N. Mansard, O. Khatib, and A. Kheddar, "A unified approach to integrate unilateral constraints in the stack of tasks," *IEEE Trans. Robot.*, vol. 25, no. 3, pp. 670–685, Jun. 2009.

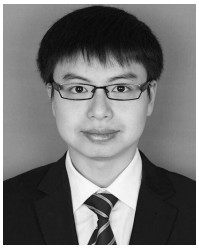
[19] A. Colomé and C. Torras, "Closed-loop inverse kinematics for redundant robots: Comparative assessment and two enhancements," *IEEE/ASME Trans. Mechatronics*, vol. 20, no. 2, pp. 944–955, Apr. 2015.

[20] T. Petrič and L. Žlajpah, "Smooth continuous transition between tasks on a kinematic control level: Obstacle avoidance as a control problem," *Robot. Auton. Syst.*, vol. 61, no. 9, pp. 948–959, 2013.

[21] R. A. Horn and C. R. Johnson, *Matrix Analysis*. New York, NY, USA: Cambridge Univ. Press, 1990.

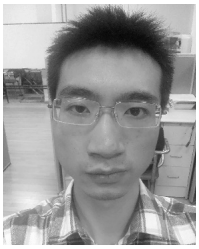
[22] S. Chiaverini, O. Egeland, and R. K. Kanestrom, "Achieving user-defined accuracy with damped least-squares inverse kinematics," in *Proc. 5th Int. Conf. Adv. Robot. 'Robots Unstruct. Environ. (ICAR)*, Jun. 1991, pp. 672–677.

[23] S. Huang, Y. Peng, W. Wei, and J. Xiang, "Clamping weighted least-norm method for the manipulator kinematic control with constraints," *Int. J. Control*, vol. 89, no. 11, pp. 2240–2249, 2016.



PEI JIANG received the B.S. and Ph.D. degrees in control science and engineering from Zhejiang University, Hangzhou, China, in 2008 and 2015, respectively.

He is currently a Lecturer with the College of Mechanical Engineering, University of Chongqing, Chongqing, China. His research interests include robotic systems, soft robotics, and energy modeling of industrial robots.



SHUIHUA HUANG received the B.S. and Ph.D. degrees in control theory and control engineering from Zhejiang University, Hangzhou, China, in 2011 and 2016, respectively.

He is currently an Engineer with Hangzhou Hikvision Digital Technology Co., Ltd., China. His current research interests include robotic manipulator control, visual servoing, and real-time system design.

JI XIANG (M'09–SM'13) was born in Wenzhou, China, in 1975. He received the B.S. degree from the North China University of Technology, Beijing, China, in 1996, and the M.S. and Ph.D. degrees from Zhejiang University, Hangzhou, China, in 1999 and 2005, respectively.

From 2005 to 2007, he was a Postdoctoral Researcher with Zhejiang University and visited the City University of Hong Kong for three months. Under the support of a Gleddens Senior Visiting Fellowship, he visited The University of Western Australia, in 2008. In 2013, he visited The University of Sydney as a Visiting Scholar. He is currently a Professor with the Department of System Science and Engineering, Zhejiang University. His research interests include networked dynamic systems, microgrids, and underwater vehicles.

Dr. Xiang is a Senior Member of the IEEE Control Systems Society.

MICHAEL Z. Q. CHEN (M'08–SM'16) received the B.Eng. degree in electrical and electronic engineering from Nanyang Technological University, Singapore, and the Ph.D. degree in control engineering from Cambridge University, Cambridge, U.K.

He is currently a Professor with the School of Automation, Nanjing University of Science and Technology, Nanjing, China.

Dr. Chen is a Fellow of the Cambridge Philosophical Society. He is a Guest Associate Editor of the *International Journal of Bifurcation and Chaos*.