International Journal of Robotics and Automation, Vol. 23, No. 1, 2008

RESOLVE REDUNDANCY WITH CONSTRAINTS FOR OBSTACLE AND SINGULARITY AVOIDANCE SUBGOALS

C. Qiu,* Q. Cao,* and Y. Sun*

Abstract

In this work we analyse the general form solution of the Inverse Redundant Kinematics (IRK) problem from a geometric point of view. We propose two new methods, Velocity Projection Constraints Method (VPCM) and On-line Task Modification Method (OTMM), which formulate the obstacle and singularity avoidance as constraints, and will be incorporated in the standard Quadratic Programming (QP) method to resolve the IRK problem. This will avoid the compromise or conflict between weighted optimization criteria for the obstacle and singularity avoidance subgoals, as well as the complicated weight adjustment process. The OTMM is effective not only to redundant manipulators but also to nonredundant manipulators. Numerical simulation examples validate and prove the effectiveness of the proposed methods.

Key Words

Inverse redundant kinematics, obstacle avoidance, singularity avoidance, dynamic constraints, Quadratic Programming method

1. Introduction

Due to the remarkable ability to reconfigure the manipulator without changing the end-effector pose, redundant manipulators can perform specific tasks while satisfying multiple restrictions such as singularities [1], obstacles [2–14], joint motion limits (joint range and velocity limit) [2–6], and at the same time optimize dynamic performance criteria [2–10]. Redundant manipulators have more degrees of freedom (DOF) than that required by the task, and due to this fact there exist an infinite number of solutions for the Inverse Redundant Kinematics (IRK) problem. Thus, in order to take full advantage of their dexterity, redundant manipulators are the object of development and implementation of many methods in current robotics research, such as the pseudoinverse method [15], the Extended Jacobian Method (EJM) [16], the Gradient Projection Method (GPM) [17], the Lagrange Multiplier Method (LMM) [18], the Quadratic Programming (QP) method [4–7], the Direct Search Method (DSM) [8–10], the Artificial Neural Network Method (ANNM) [11–13, 19–22], and the Linear Programming (LP) method [19, 23].

GPM extends the pseudoinverse solution by projecting the Gradient of Optimization Criteria Function (GOCF) in the null space of Jacobian matrix, and then calculates the joint velocity to optimize the criteria for the Obstacle Avoidance (OA) [2, 3] and/or Singularity Avoidance (SA) [1, 24]. In applications, much effort must be invested in adjusting the nonnegative scalar of the projected velocity in order to obtain joint velocity smoothness and to meet the joint motion limits requirement. Baillieul introduced the EJM method [16], which constrains the GOCF to be in the null space of the Jacobian matrix. Chang [18] presented the LMM method, which constrains the GOCF to be in the range space of the Jacobian matrix. Charles [25] has given a deep analysis and proved that the EJM and LMM are equivalent. EJM and LMM augment task space with subtask and impose strong limitations on manipulator's configuration. Rigorous requirements for the initial configuration, limited ability for realizing multi-subtasks, algorithmic singularity, and less consideration of the joint motion limit are their main drawbacks [8, 10, 16, 25, 26]. However, the closed-form solution can be found with EJM and LMM under nonsingularity situations [25].

For the compact QP method of Cheng *et al.* [4–7], the OA, SA, and drift-free were formulated as weighed quadratic objective function together with solution feasibility objective function, such as minimum 2-norm of joint velocity. Ding *et al.* [20] and Ho *et al.* [23] formulate the IRK as an LP problem. The main difference, compared to the QP based method, is the type of the objective function. Ho *et al.* [23] pointed out that an additional smoothness measure must be considered in the LP method to avoid the jittering of the joint trajectory.

On configuration manifold, the DSM [8–10] searches for the solution vector, which reduces the tracking error of the end-effector and optimizes performance criteria. The DSM is more suitable and comprehensive than the QP method to evaluate the configuration control quality in view of operation task fulfilment, but is less efficient from the real-time implementation point of view.

* School of Mechanical Engineering, Shanghai Jiao Tong University, 200240, Shanghai, China; e-mail: {cw_qiu, qxcao, sunyj}@sjtu.edu.cn

Recommended by Dr. Hani Hagras
(paper no. 206-3041)

In recent years, new interest has arisen in using Artificial Neural Networks (ANN) to IRK research due to their excellent nonlinear mapping attribute, potential fast computational time under parallel computation mechanisms, and optimization ability. Research has been done in realizing the forward and inverse kinematics mapping of robot manipulators with neural networks [26]. Mao and Hsia [11] showed the effectiveness of the multilayer neural network approach to solve the IRK problem. Ding *et al.* [12] obtained the collision-free IRK solution using a Hopfield network with a weighted form objective performance function. Wang *et al.* [19] presented a Lagrange network for the kinematic resolution without considering physical limit constraints. Ding and Wang [20] proposed the use of a recurrent form linear programming network, named primal-dual ANN, to find the minimum infinity-norm solution of redundant manipulators. The methods proposed by Ding and Wang [20] and Wang *et al.* [19] can deal with the case when the Jacobian is in or near singular configuration. Zhang *et al.* [13] presented an improved formulation under recurrent ANN strategy in the sense that the OA requirement is represented with dynamically updated inequality constraints. In order to reduce computation complexity and to increase convergence rate, Xia *et al.* [21, 22] proposed a novel recurrent form ANN for solving the kinematic resolution problem while considering the physical limit constraints. The proposed neural network has a one-layer structure and is not required for computing the inverse matrix.

Among the kinds of requirements for utilization of the dexterity of redundant robots, OA, SA, and joint motion limits avoidance are the essential requirements for the redundant robots to complete a task. In most of the research work the essential requirements are formulated as performance functions [8–10] that are then combined in a weighed-sum function to resolve the IRK problem [3–7, 11, 12]. From the point of view of real-time applications, the ANN-based methods, primarily the ones that use online training for the recurrent network, are especially attractive for redundant robot control. However, using recurrent neural networks for finding the solution of a class of optimization problems with constraints, while guaranteeing at all times the stability of the system, is an open research area [20, 27–29, 30].

The weighted-sum method that was used in the case of multi subtasks requirements [4–7, 11, 12, 20] have two drawbacks: The first is the compromise between weighted parts with different physical units in performance function [8, 29, 31]. The weighted-more parts in performance function become stronger and the other, weighted-less parts become weaker. However, this does not mean that the original term will be optimized as the weighted one. The second drawback consists in the additional computational burden to calculate, at every time sample, the inverse of the Jacobian minor or the pseudoinverse of the Jacobian matrix, the performance function and its derivatives [32]. Ho *et al.* have shown [23] that the LP method is a suitable approach for controlling articulated systems with a large number of DOFs and constraints for real-time applications, as the LP method has a lower computation time than the pseudoinverse method. Goldbach had given extensive numerical results [32], which indicate that the QP method is potentially more efficient than the LP method when considering the same input and linear constraints.

The weighted quadratic performance function is superior to the linear one with system dynamics–consistent attribute. In this paper we construct the general form of solution for the first-order IRK problem based on the geometry analysis by making use of differential geometry tools. We propose a Velocity Projection Constraints Method (VPCM) for OA and an On-line Task Modification Method (OTMM) for SA. OA and SA requirements are formulated as, respectively, dynamic inequality and equality constraints, which are incorporated in the standard QP algorithm to resolve IRK problem. These constraints are then combined with the joint motion limits in order to model the feasible subspace of the configuration space of the manipulator, such that the drawback of the penalty method [29] will be avoided. At the same time, the method avoids the compromise, complicated weight adjustment [3–8], and noncommensurate, which was defined by Schwartz *et al.* [31] for blending different optimizing objectives with different physical unit, factor of the weighed-sum method. Numerical examples show the validity of our method.

In Section we present the general formulation of the IRK in a differential geometry setting, when the joint motion limits are modelled as linear inequality constraints. Section 3 proposes intuitive geometry formulations to set up the inequality constraints to define the OA subspace. Section 4 gives the online singularity avoidance method. Section 5 shows the simulation results for obstacle and singularity avoidance when two planar manipulators are considered. Conclusions are presented in Section 6.

2. Generalized Formulation of the IRK Problem

In this section we give a fundamental basis for understanding the differential transformations that are implicit in solving the IRK problem. We adopt differential geometry tools to find the general-form solution of the IRK problem by making an analysis of the relationship between the tangent and cotangent spaces of the configuration manifold and the Jacobian matrix.

2.1 Analysis and Formulation

For the linear attribute of the first-order kinematics mapping, the majority of efforts have been focused on finding the solution of the redundancy at velocity level. The kinematics of manipulators is frequently represented as:

$$x = f(q) \tag{1}$$

$$\dot{x} = J(q) \cdot \dot{q} \tag{2}$$

where $x \in R^n$ represents the task in Cartesian space, $f(q) \in R^n$ is a vector function expressing the forward kinematics relation, $q \in R^m (m > n)$ represents the joint variables in configuration space, and $J(q) = [J_1^T \cdots J_n^T]^T \in R^{n \times m}$ is the end-effector Jacobian matrix, which consists of joint twist respect to Cartesian space. Let N be the

n-dimensional manifold, which defines the task in Cartesian space, and let M be the corresponding m-dimensional manifold in the configuration space. We can then find $q \in M$ and $x \in N$. Equation (2) expresses the mapping, f_*, between the tangent space $TM_q \in R^m$ located at point q on the manifold M and the tangent space $TN_x \in R^n$ located at point x on manifold N [33]. Let the vector field $\dot{x} \in TN_X$ and $\dot{q} \in TM_q$ be given by:

$$\dot{x} = x^1 \frac{\partial}{\partial x_1} + \cdots + x^n \frac{\partial}{\partial x_n} \tag{3}$$

$$\dot{q} = q^1 \frac{\partial}{\partial q_1} + \cdots + q^m \frac{\partial}{\partial q_m} \tag{4}$$

where $x^i,\ i = 1, 2, \ldots, n$, and $q^j,\ j = 1, 2, \ldots, m$, are the coordinates of $\dot{x}$ and $\dot{q}$ in the canonical bases $\partial/\partial x_i$ and, respectively, $\partial/\partial q_j$. Associated with the tangent space TM_q is the dual space $TM_q^* \in R^m$, also named cotangent space, which is expressed with the set of linear functions on TM_q. The row vectors of $J(q)$ span the subspace $\Delta_r = span\{J_1 \cdots J_n\} \in TM_q^*$. We prove (see Appendix) that $J_i = f_*^{-1}(\partial/\partial x_i)$. Thus the geometric meaning of (2) is that the ith coordinate of $\dot{x}$ is the projection of the vector $\dot{q}$ on the vector J_i, which is the f_* induced map $f_*^{-1}(\partial/\partial x_i)$ at point q. Assuming the dimension of Δ_r is r_J, where $r_J \le n$, we can find a set of linearly independent vector fields $N_t, \{t = 1, \ldots, m - r_J\}$, which are independent to Δ_r and span a subspace $\Delta_n \in TM_q^*$. The distribution $\Delta = span\{\Delta_r + \Delta_n\} \in TM_q^*$ spans a dimension m space at point q. We can construct the vector $\dot{q}$ in space $\Delta \in R^m$ as:

$$\dot{q} = [J_1, \ldots, J_{r_J}, N_1, \ldots, N_{m-r_J}] \cdot \lambda = [J_r^T N_q] \cdot \begin{bmatrix} \lambda_a \\ \lambda_b \end{bmatrix} \tag{5}$$

where $\lambda \in R^m$ represents the coordinates of the vector $\dot{q}$ in the non-normalized affine coordinates frame (q, Δ); λ_a is the coordinate vector of $\dot{q}$ in the m-dimensional subspace Δ_r, and λ_b is the coordinate vector of $\dot{q}$ in the m-dimensional subspace Δ_n. It is worth noting that N_t is unnecessary orthogonal to the space Δ_r; however, there is a premise that the vectors N_t must be linearly independent to each other and to the row vectors of the Jacobian matrix $J(q)$.

Now solving the inverse kinematics of a redundant robot reduces to finding the velocity vector $\dot{q}$ in the space TM_q^* under the condition that the coordinates λ_a of $\dot{q}$ are invariant on the space Δ_r, namely, it satisfies the end-effector velocity $\dot{x}$. Many additional objectives to be achieved with a redundant manipulator can be expressed in terms of minimizing a criterion function H [2–11, 16]. If we assure $-\nabla H^T \cdot \dot{q} > 0$ by choosing properly the λ_b, the corresponding solution $\dot{q}$ will make H increase.

2.2 Solution of the IRK Problem

Equation (5) gives the generalized solution for the first-order IRK problem. The IRK solutions at velocity level are particular forms of (5). Suppose $Rank(J) = n$ and $\lambda_b = 0_{(m-n)\times 1}$; we can get the pseudoinverse solution by setting $\lambda_a = (JJ^T)^{-1}\dot{x}$. Similarly, any other velocity level solution variants, such as solution with methods EJM [16], GPM [17], LMM [18], and QP [4–7], can be formed by selecting properly λ in (5).

The standard optimized solution under equality and inequality constraints for the IRK at velocity level can be given as:

$$\text{Minimize } H(\lambda) \tag{6}$$

$$\text{Subject to } J \cdot [J_{\mathrm{r}}^T N_q] \cdot \lambda = \dot{x} \tag{7}$$

$$A \cdot \lambda \le B \tag{8}$$

where $H(\lambda) \in R$ is a performance function. Equation (7) represents the relationship between the end-effector velocity and the joint velocity. The function $H(\lambda)$ can be linear or nonlinear form [3–13, 20–22]. Due to system dynamics consistent attribute, the quadratic form performance function is preferred, such as $H(\lambda) = 1/2\lambda^T W\lambda + \varphi\lambda$, where $W \in R^{m\times m}$ is a positive-definite cost matrix, and $\varphi \in R^{1\times m}$ is a linear cost vector. The joint motion constraints were formulated as inequalities in (8). The authors of [4] introduced the definition of joint range limit and joint velocity limit; we modify it and add acceleration limits into the constraints formulation.

Let $q_l(q_u)$ denote the lower (upper) limits of joint range and $\dot{q}_l(\dot{q}_u)$ denote the lower (upper) joint velocity limits of joint while $\ddot{q}_l(\ddot{q}_u)$ represent the lower (upper) joint acceleration limits. The available joint range, joint velocity and joint acceleration can be converted to velocity constraints as:

$$\frac{\ell q_l - q(t)}{\Delta t} \le \dot{q}(t) \le \frac{hq_u - q(t)}{\Delta t} \tag{9}$$

$$\beta\dot{q}_l \le \dot{q}(t) \le \beta\dot{q}_u \tag{10}$$

$$\dot{q}(t - \Delta t) + \Delta t \cdot \gamma\ddot{q}_l \le \dot{q}(t) \le \dot{q}(t - \Delta t) + \Delta t \cdot \gamma\ddot{q}_u \tag{11}$$

where $\ell = diog[\ell_1, \ldots, \ell_m],\ h = diog[h_1, \ldots, h_m],\ \beta = diag[\beta_1, \ldots, \beta_m],\ \gamma = diag[\gamma_1, \ldots, \gamma_m],\ 0 < \ell_i \le 1,\ 0 < h_i \le 1,\ 0 < \beta_i \le 1,\ 0 < \gamma_i \le 1$. The joint torque limits consideration can be realized by adjusting γ and replacing $\ddot{q}_l(\ddot{q}_u)$ with the online minimum (maximum) acceleration output under the joint torque limits. We can combine (9), (10), and (11) to a matrix form as (8); then A and B have the form:

$$A = \begin{bmatrix} -I_m \\ I_m \end{bmatrix}_{2m\times m} \cdot \begin{bmatrix} J_r \\ N_q^T \end{bmatrix}^T \tag{12}$$

$$B = \begin{bmatrix} -\max\left(\dfrac{\ell q_l - q(t)}{\Delta t}, \beta\dot{q}_l, \dot{q}(t - \Delta t) + \Delta t \cdot \gamma\ddot{q}_l\right) \\ \min\left(\dfrac{hq_u - q(t)}{\Delta t}, \beta\dot{q}_u, \dot{q}(t - \Delta t) + \Delta t \cdot \gamma\ddot{q}_u\right) \end{bmatrix}_{2m\times 1} \tag{13}$$

In the case in which several performance functions need to be optimized, the weighted-sum method is the usual way to combine them [34]. However, it is very possible that the terms that are weighted more will not be better satisfied than the terms that are weighted less in the entire weighted sum [29]. The noncommensurate [31], which exists in blending different optimizing objectives with different physical unit, compromise, and conflict [8] between performance functions, makes the weight adjustment a difficult issue [6, 29].

Many subtasks will determine limits to the joint configuration space. However, as long as the limits are not violated and there are no conflicts with the joint motion limits, the requirement of subtasks (e.g., OA, SA, and joint range availability) are satisfied.

3. Obstacle Avoidance Constraint

The controller of the manipulator must properly avoid the obstacles presented in the environment in order to prevent possible damage to the manipulator. The location of the obstacle can be determined from the information regarding the distance between the obstacle objects and the links of the manipulator. But arbitrary and complex shapes of obstacle objects make the minimum distance detection between the boundary hulls of obstacles and the manipulator links highly computation consuming. Simple boundary hulls, such as cube, sphere, and column, which envelope the obstacle and links' boundary, can decrease the proximity checking computation burden. In this work, we adopted spherical and column bounding boxes as envelopes for the obstacle and manipulator links' boundary. We also simplify the minimum distance checking problem as checking the shortest distance between a point and a line segment. Excepting in the situation showed in Fig. 1(a), the critical point for possible collision, denoted by C, is located only at the joint of the manipulator; Figs. 1(b) and (c) show the situation. A similar example has been given in Zhang *et al.* [13].

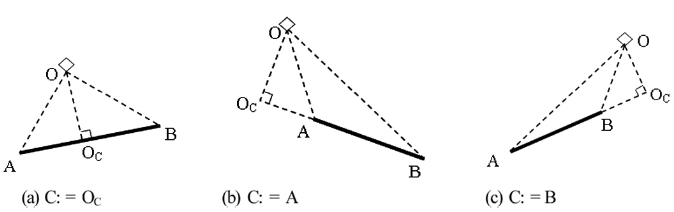

Figure 1. Location of critical point C according to three possible situations for obstacle point and the link L_{AB}.

Let P_A and P_B represent the joint A and B; O represents the obstacle point; the vector $\vec{V}_{AB}$ corresponding to the link L_{AB} will be:

$$\vec{V}_{AB} = [v_1 \;\; v_2 \;\; v_3]^T = \overrightarrow{P_A P_B} \tag{14}$$

The point O_C can be easily located by solving the system of equations:

$$\begin{cases} v_1(x - O_x) + v_2(y - O_y) + v_z(z - O_z) = 0 \\ \dfrac{x - P_{Ax}}{v_1} = \dfrac{y - P_{Ay}}{v_2} = \dfrac{z - P_{Az}}{v_3} \end{cases} \tag{15}$$

The three cases presented in Fig. 1 can be distinguished by using the following pseudocode:

If $O_{Cx} \in [P_{Ax}, P_{Bx}]$, *then case (a)*
Else if $\|\overrightarrow{O_C P_A}\|_2 \le \|\overrightarrow{O_C P_B}\|_2$, *then case (b)*
Else case (c)
End

In [19] the inequality constraints limit stringently the solution space, because only the sign of vector $\overrightarrow{OO_C}$ was used to formulate the constraints for the velocity of the critical point on the manipulator link. In order to reduce the nonessential limit on the available solution space, projection of critical point velocity on the vector $\overrightarrow{OO_C}$ will be used to set up the geometry intuitive constraints. The unit vector $\overrightarrow{OO_C}$ is given as:

$$r_{oc} = [X_{O_C} - X_O \;\; Y_{O_C} - Y_O \;\; Z_{O_C} - Z_O]^T / \|\overrightarrow{OO_C}\| \tag{16}$$

In the case in which the distance, d, from the obstacle to the manipulator link is less than the safety margin, d_S, we must adopt obstacle avoidance measure in order to eliminate possible damage to the manipulator. Considering the relation between the critical point velocity and the vector r_{oc}, we determine that the critical point will escape the possible collision zone when the projection value of the critical point velocity on r_{oc}, P, is positive. Thus, the obstacle avoidance inequality constraints can be formulated as:

$$P = r_{oc}^T \cdot (J_O \cdot \dot{q} - v_o) \ge 0 \tag{17}$$

where J_O is the Jacobian matrix with the critical point O_C as velocity reference point; $v_o \in R^3$ is the velocity of the nearest obstacle point. We can find that a discontinuity in the joint velocity will occur when $P < 0$ and the constraint (17) was suddenly applied into (8). Therefore, the smoothing measure $P \ge t$ must be adopted to avoid the discontinuity. We define t as:

$$t = \begin{cases} P & d_S + \sigma_1 < d \le d_S + \sigma_2 \\ \rho(d) \cdot P_C & d_S < d \le d_S + \sigma_1 \\ 0 & d \le d_S \end{cases} \tag{18}$$

where $\rho(d) \in [0, 1]$ is a bell shape function, $\sigma_1 \in R^+, \sigma_2 \in R^+$ are the buffer distance, and $\sigma_1 < \sigma_2$, $P_C = \{P | d(q,t) = d_S + \sigma_1\}$. When $d > d_S + \sigma_2$, the constraint described by (18) is not considered to alleviate the computation burden. If there are s obstacle points that need to be considered, and the manipulator has enough dexterity or redundant DOF, we can formulate the s OA constraints as:

$$r_{o_i c}^T (J_{O_i} \dot{q} - v_{o_i}) \ge t_i, \quad i = 1, \ldots, s \tag{19}$$

The matrices A and B can be extended by combining (12), (13), (18), and (19).

The above-presented process formulates the OA requirements as feasible range of joint velocities by setting up the critical Velocity Projection Constraints Method (VPCM). However, there are cases in which the constraints are over-stringent, and this means that the constraints (7) or (8) must be broken in order first to fulfill the safety consideration and then to perform the task. In these cases there must be applied high-level task-planning algorithms for the purpose of performing the task and at the same time satisfying the physical limits and safety requirements. It is also well known that infeasible solutions may be obtained when the end-effector moves close to a singularity [1, 4, 8, 16, 24]. In the following section, we present a general SA method. A comparison with the well-known damped least-squares method [35] will also be given.

4. Singularity Avoidance Consideration

For nonredundant manipulators, singularity-free motion can be achieved with offline path planning. However, it is required to have a priori knowledge of all the singular configurations of the manipulator. For a redundant manipulator, self-motion can realize the avoidance of escapable singularities, but this cannot be achieved for the case of inescapable singularities [36, 37]. A similar method to avoid the obstacle collision configuration can be used to avoid the escapable singularities by self-motion based on the knowledge of all the singular configurations of the manipulator. Let the escapable singularity configurations be expressed as q_{es}; we can avoid the escapable singularity configurations by steering the joint velocity with constraints $(q(t)-q_{es})\cdot\dot{q}\geq 0$. When the desired direction of the end-effector motion locates in the range space of the singular Jacobian matrix, the method of solving the IRK problem presented in Section 2 will always compute a satisfying solution. However, the desired time increment motion $\dot{x}(t)=(x_d(t+\Delta t)-x(t))/\Delta t$ will often produce deviation in the range space of the singular Jacobian matrix, and this will result in failure in accomplishing the trajectory following task. An online SA method will avoid that problem and reduce the burden of checking if the singular configuration can be avoided by self-motion. This is due to the fact that the singularities can be directly identified based on the singular values of the Jacobian matrix. The singular value decomposition of the Jacobian matrix is given by:

$$J = U\sum V^T \tag{20}$$

where $U=[u_1,u_2,\ldots,u_m]$ and $V=[v_1,v_2,\ldots,v_n]$ are orthogonal matrices and $\sum=[diag(\sigma_1,\sigma_2,\ldots,\sigma_m)|0]$, $(m<n)$ is a $m\times n$ matrix whose diagonal elements are the ordered singular values of J.

A singular direction vector, u_S, in the workspace is identified from the fact that its corresponding singular value is zero; the vector u_S can then be used to modify the velocity of the end-effector to avoid the singularity. This modification method is formulated as:

$$\dot{x}_R=\dot{x}-k(1-p(\sigma_{\min}))(u_S^T\dot{x})u_S,\quad k=\begin{cases}0 & \sigma_{\min}>\sigma_S\\ 1 & \sigma_{\min}\leq\sigma_S\end{cases} \tag{21}$$

where $\dot{x}_R$ is the revised end-effector velocity; $\sigma_{\min}\in R$ is the minimum singular value of the manipulator Jacobian matrix; σ_S is the low bound for allowable singular values; and $p(\sigma_{\min})\in[0,1]$ is an monotone decreasing function. The above formulation realizes SA by online modifying the task velocity of the end-effector, and is named Online Task Modification Method (OTMM). OTMM is not only effective for redundant manipulators but also for nonredundant manipulators.

The damped least-squares method [35] is often used to get the singularity robust solution and is defined as:

$$\dot{q}=J^T(JJ^T+\mu I)^{-1}\dot{x} \tag{22}$$

where $\mu>0$. Now using the matrix formula $(C+D)^{-1}=C^{-1}(I-(C^{-1}+D^{-1})^{-1}C^{-1})$ (where $C\in R^{n\times n}$ and $D\in R^{n\times n}$), we can find that the equation (22) revises the velocity of the end-effector such that the singularity will be avoided. However, (22) requires the computation of the matrix inverse and thus has higher computation burden than (21). At the same time, (21) is more intuitive and easier to incorporate into the IRK resolution method presented in Section 2.

We formulated the OA and SA requirements as, respectively, inequality and equality constraints with VPCM and OTMM. These constraints were incorporated in the standard QP algorithm to avoid the compromise, complicated weight adjustment, and noncommensurate factor in weighed sum method [3–8, 31].

5. Simulation Examples

Two simulation examples are given in this section in order to demonstrate the effectiveness of the presented methods. Matlab was used to implement the control algorithm and perform the numerical simulation for the robotic arm movement.

The first simulation example considers a four-link planar manipulator following a circular trajectory while avoiding a moving circular obstacle with radius 0.1 m. The parameters of the manipulator and its initial states are $[l_1,l_2,l_3,l_4]=[1,0.8,0.8,0.6]$ m, $[q_1,q_2,q_3,q_4]=[-1.2763,1.1927,0.7517,0.4086]$ rad. The physical limits of the manipulator are defined as $q_u=-q_l=[3,3,3,3]^T$rad, $\dot{q}_u=-\dot{q}_l=[0.9,1,1.2,1.5]^T$ rad/s, and $\ddot{q}_u=-\ddot{q}_l=[6,7,8,9]^T$ rad/s^2. The desired path of the end-effector is a circle with radius 0.6 m and centre (1.8, 0), which must be tracked in a counter-clockwise movement within 12 seconds. The centre of the obstacle moves from point (3.1, −0.65) to point (0.7, −0.65) at even speed 0.175 m/s. The other parameters are $d_s=0.2$, $\sigma_1=0.2$, and $\sigma_s=0.05$. The function $p(\sigma_{\min})$ has form $p(\sigma_{\min})=\sqrt[3]{\sigma_{\min}/\sigma_s}$.

Fig. 2 shows the simulation results for minimum 2-norm of the joint velocity without obstacle and singularity

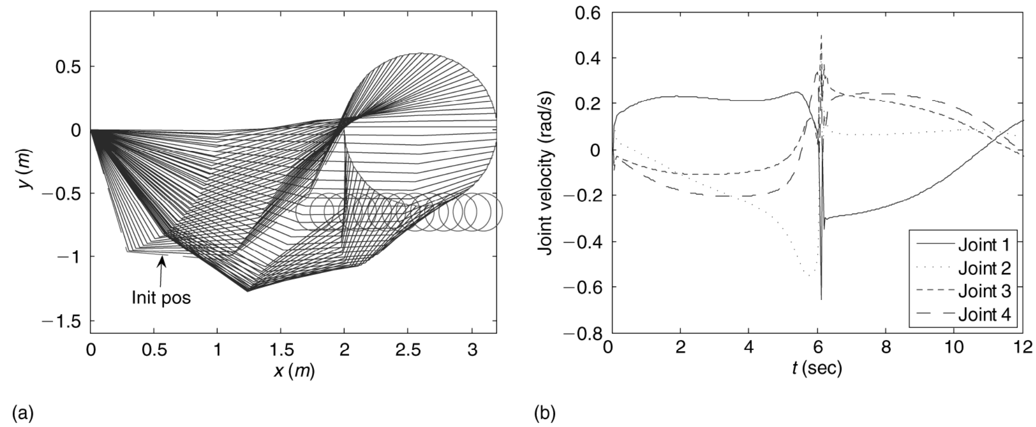

Figure 2. Simulation results for optimized minimum 2-norm of joint velocity without OA and SA consideration for four-link planar manipulator.

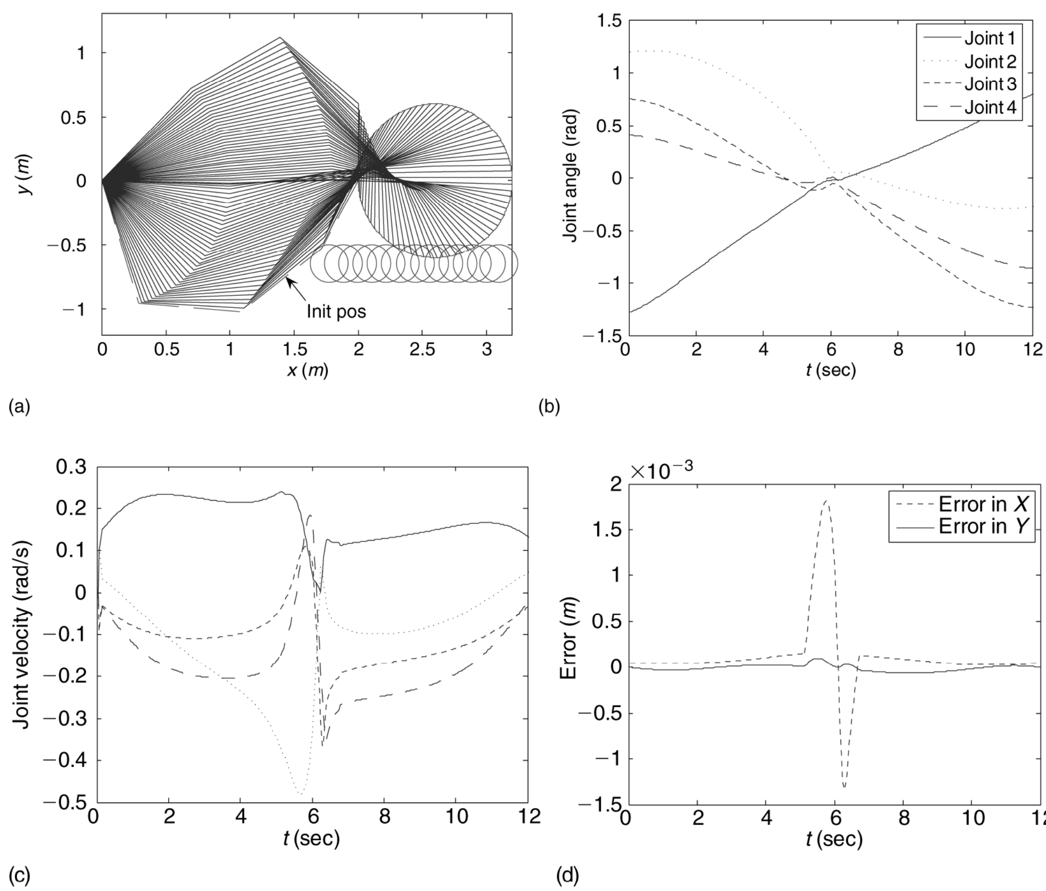

Figure 3. Simulation results for optimized minimum 2-norm of joint velocity with obstacle and singularity avoidance consideration for four-link planar manipulator.

avoidance consideration. One can see that there exists chattering and the joint velocity leap when the manipulator moves to a singularity. Also, it results in collision between the dynamic obstacle and manipulator link. Fig. 3 shows the simulation results for minimum 2-norm of the joint velocity when OA and SA are described as constraints. Fig. 3(a) shows that the QP method with OA constraints is successful in avoiding the dynamic obstacle. Figs. 3(c) and (d) show that smooth joint velocity and only small errors result while passing through the singularity.

The second simulation example is a six-link planar manipulator following a line trajectory while avoiding a point and triangular obstacles within 12 seconds. The parameters of the manipulator and its initial states are $[l_1, l_2, l_3, l_4, l_5, l_6] = [1, 1, 1, 1, 1, 1]\,m$, $[q_1, q_2, q_3, q_4, q_5, q_6] = [0, 0, 0, 0, 0, 0]$ rad. The physical limits of the manipulator are defined as $q_u = -q_l = [3, 3, 3, 3, 3, 3]^T$ rad, $\dot{q}_u = -\dot{q}_l = [0.8, 1, 1.2, 1.2, 1.5, 1.8]^T$ rad/s, and $\ddot{q}_u = -\ddot{q}_l = [6, 6, 8, 10, 12, 12]^T$ rad/s^2. The line trajectory is from point (6, 0) to point (0, −3) with bell

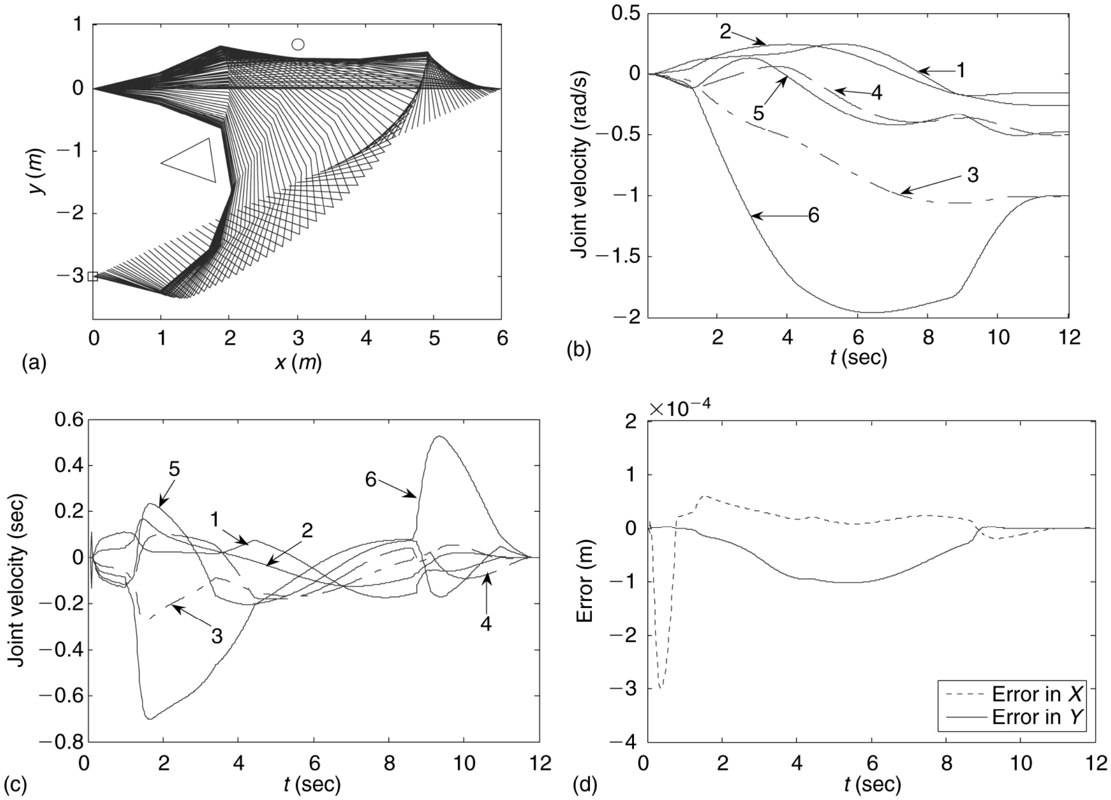

Figure 4. Simulation results for optimized minimum 2-norm of joint velocity with obstacle and singularity avoidance consideration for six-link planar manipulator.

shape acceleration law. The other parameters are $d_s = 0.2$, $\sigma_1 = 0.2$, and $\sigma_s = 0.06$.

Fig. 4 shows the simulation results for the minimum 2-norm of joint velocity with obstacle and singularity avoidance constraints for the six-link manipulator. The initial configuration of the manipulator is at a singularity. The task velocity is dependant on the column vectors of the Jacobian matrix at initial configuration; therefore no feasible solution can be found at that point. In our algorithm, this problem is resolved by OTMM given by equation (21). The point and triangular obstacles are successfully avoided with VPCM.

The two examples are implemented in Matlab R14 environment on a PC with a 2.4 GHz Pentium 4 CPU and 512M RAM. The average computation cycle time is, respectively, 31.25 ms and 36.16 ms for the two simulation examples.

6. Conclusion

In this work, the general-form solution of the IRK problem was given by means of differential geometry tools. OA, SA, and joint motion limits avoidance are the essential requirements for the redundant robots to complete a task. In order to avoid the compromise, complicated weight adjustment, and noncommensurate factor in weighed-sum method while realizing the OA and SA subgoals for the IRK problem, we proposed VPCM and OTMM to construct OA as inequality constraints and SA as equality constraints. These constraints were incorporated into QP method, and the IRK problem was solved successfully. The proposed online singularity avoidance method, OTMM, is not only suitable for the control of redundant manipulators but can be also applied for nonredundant manipulators. When applied to redundant manipulators, the method alleviates the burden of checking the nature of the singularity points. Two simulation results were presented to demonstrate the validity of the proposed method.

Appendix

For any point $q \in M$ and $x \in N$, the map relation between $\partial/\partial q_i$ and $\partial/\partial x_j$ [33, p. 18] is:

$$f_* \left(\frac{\partial}{\partial q_i} \right) = \sum_{\beta=1}^{n} \left(\frac{\partial f^\beta}{\partial q_i} \right) \frac{\partial}{\partial x_\beta} \tag{23}$$

Thus, the ith column of Jacobian matrix $J(q)$ represents the components of the map of $\partial/\partial q_i$ in TN_X on the canonical basis of TN_X. $dq_i, i \in (1, 2, \ldots, m)$ and $dx_j, j \in (1, 2, \ldots, l)$ are the canonical bases of the cotangent spaces TM_q^* and TN_X^*, which are dual space of, respectively, TM_q and TN_X. Considering the bilinear map $\langle \cdot, \cdot \rangle : TN \times TN^* \to R$, we denote:

$$\begin{aligned} \left\langle \frac{\partial}{\partial x_\beta}, \mathrm{dx}_\beta \right\rangle &= \left\langle \frac{\partial}{\partial x_\beta}, d(x_\beta \circ f) \right\rangle \\ &= \left\langle \frac{\partial}{\partial x_\beta}, \sum_{i=1}^{n} \frac{\partial f^\beta}{\partial q_i} dq_i \right\rangle \end{aligned}$$

$$= \sum_{i=1}^{n} \left\langle \frac{\partial}{\partial x_\beta}, dq_i \right\rangle \frac{\partial f^\beta}{\partial q_i}$$

$$= \sum_{i=1}^{n} \left\langle \frac{\partial}{\partial q_i}, dx_\beta \right\rangle \frac{\partial f^\beta}{\partial q^i} \tag{24}$$

$$= \left\langle \sum_{i=1}^{n} \frac{\partial f^\beta}{\partial q_i} \frac{\partial}{\partial q_i}, dx_\beta \right\rangle$$

Therefore, we obtain $f_*^{-1}(\partial/\partial x_\beta) = \sum_{i=1}^{n} (\partial f^\beta/\partial q_i)$ $(\partial/\partial q_i)$. This equation expresses the fact that the β-th row of Jacobian matrix $J(q)$ consists of the components of the inverse map of $\partial/\partial x_\beta$ in the canonical basis $\partial/\partial q_i$. One can now express the β-th component of $\dot{x} \in TN_X$ as:

$$x^\beta = \left\langle \sum_{i=1}^{n} \frac{\partial f^\beta}{\partial q_i} \frac{\partial}{\partial q_i}, \dot{q} \right\rangle = row_\beta(J(q)) \bullet \dot{q} \tag{25}$$

Acknowledgment

The authors are grateful for comments and suggestions from Draguna Vrabie.

References

[1] R.V. Mayorga & A.K. Wong, A singularities prevention approach for redundant robot manipulators, *Proc. IEEE Int. Conf. on Robotics and Automation*, Cincinnati, OH, USA, 1990, 812–817.

[2] A.A. Maciejewski & C.A. Klein, Obstacle avoidance for kinematically redundant manipulators in dynamically varying environments, *International Journal of Robotics Research*, *4*(4), 1985, 109–117.

[3] Y. Nakamura, H. Hanafusa, & T. Yoshikawa, Task priority based redundancy control of robot manipulators, *International Journal of Robotics Research*, *6*(2), 1987, 3–15.

[4] F.-T. Cheng, T.-H. Chen, & Y.-Y. Sun, Efficient algorithm for resolving manipulator redundancy: The compact QP method, *Proc. IEEE Int. Conf. on Robotics and Automation*, Nice, France, 1992, 508–513.

[5] F.-T. Cheng & M.S. Shih, Multiple-goal priority considerations of redundant manipulators, *Robotica*, *15*(6), 1997, 675–691.

[6] F.-T. Cheng, M.-S. Shih, F.-C. Kung, & Y.-Y. Sun, The improved parallel scheme for multiple-goal priority considerations of redundant manipulators, *Proc. IEEE Int. Conf. on Robotics and Automation*, Albuquerque, NM, USA, 1997, 2409–2414.

[7] F. -T. Cheng, T.-H. Chen, Y.-S. Wang, & Y.-Y. Sun, Obstacle avoidance for redundant manipulators using the compact QP method, *Proc. IEEE Int. Conf. on Robotics and Automation*, Atlanta, GA, vol. 3, 1993, 262–269.

[8] C. Pholsiri, *Task-based decision making and control of robotic manipulators*, doctoral diss., University of Texas, Austin, TX, 2004.

[9] M. Cetin, *Performance identification and multi-criteria redundancy resolution for robotic systems*, doctoral diss., University of Texas, Austin, TX, 1999.

[10] M. Pryor, *Task-based resource allocation for improving the reusability of redundant manipulators*, doctoral diss., University of Texas, Austin, TX, 2002.

[11] Z. Mao & T.C. Hsia, Obstacle avoidance inverse kinematics solution of redundant robots by neural networks, *Robotica*, *15*, 1997, 3–10.

[12] H. Ding & S.E. Chang, A real-time planning algorithm for obstacle avoidance of redundant robots, *Journal of lntelligent and Robotic Systems*, *16*, 1996, 229–243.

[13] Y. Zhang & J. Wang, Obstacle avoidance for kinematically redundant manipulators using a dual neural network, *IEEE Trans. on Systems, Man, and Cybernetics*, *34*, 2004, 752–759.

[14] S.I. Choi & B.K. Kim, Obstacle avoidance control for redundant manipulators using collidability measure, *Robotica*, *18*, 2000, 143–151.

[15] D.E. Whitney, Resolved motion rate control of manipulators and human prostheses, *IEEE Trans. on Man-Machine Systems*, *MMS-10*, 1969, 47–53.

[16] J. Baillieul, Kinematic programming alternative for redundant manipulators, *Proc. IEEE Int. Conf. on Robotics and Automation*, St. Louis, MO, 1985, 722–728.

[17] A. Liegeois, Automatic supervisory control of the configuration and behavior of multibody mechanisms, *IEEE Trans. on Systems, Man, and Cybernetics*, *SMC-7*(12), 1977, 868–871.

[18] P.H. Chang, A closed-form solution for the inverse kinematics of robot manipulators with redundancy, *IEEE Journal of Robotics and Automation*, *RA-3*(5), 1987, 393–403.

[19] J. Wang, Q. Hu, & D. Jiang, A Lagrangian network for kinematic control of redundant robot manipulators, *IEEE Trans. on Neural Networks*, *10*(5), 1999, 1123–1132.

[20] H. Ding & J. Wang, Recurrent neural networks for minimum infinity-norm kinematic control of redundant manipulators, *IEEE Trans. on Systems, Man, and Cybernetics A, Syst. Humans*, *29*(3), 1999, 269–276.

[21] Y. Xia & J. Wang, A dual neural network for kinematic control of redundant robot manipulators, *IEEE Trans. on Systems, Man, and Cybernetics*, *31*(1), 2001, 147–154.

[22] Y. Xia, G. Feng, & J. Wang, A primal-dual neural network for online resolving constrained kinematic redundancy in robot motion control, *IEEE Trans. Systems, Man, and Cybernetics*, *35*(1), 2005, 54–64.

[23] E.S.L. Ho, T. Komura, & R.W.H. Lau, Computing inverse kinematics with linear programming, *Proc. ACM Symp. on Virtual Reality Software and Technology*, Monterey, CA, 2005, 163–166.

[24] T. Yoshikawa, Manipulability of robotic mechanisms, *International Journals of Robotics Research*, *4*(2), 1985, 3–9.

[25] A.K. Charles & L. Chu, Comparison of extended Jacobian and Lagrange multiplier based methods for resolving kinematic redundancy, *Journal of Intelligent and Robotic Systems*, *19*, 1997, 39–54.

[26] J.F. Gardner *et al.*, Applications of neural networks for coordinate transformations in robotics, *Journal of Intelligent and Robotic Systems*, *8*, 1993, 361–373.

[27] P.W. Protzel, D.L. Palumbo, & M.K. Arras, Performance and fault-tolerance of neural networks for optimization, *IEEE Trans. on Neural Networks*, *4*, 1993, 600–614.

[28] G. Serpen & D.L. Livingston, Analysis of the relationship between weight parameters and stability of solutions in Hopfield networks from dynamic systems viewpoint, *Neural, Parallel and Scientific Computations*, *2*, 1994, 361–372.

[29] Z. Li Stan, Improving convergence and solution quality of Hopfield-Type neural networks with augmented Lagrange multipliers, *IEEE Trans. on Neural Networks*, *7*(6), 1996, 1507–1516.

[30] Z. Zeng & J. Wang, Improved conditions for global exponential stability of recurrent neural networks with time-varying delays, *IEEE Trans. on Neural Networks*, *17*(3), 2006, 623–635.

[31] E. Schwartz, R. Manseur, & K. Doty, Noncommensurate systems in robotics, *International Journal of Robotics and Automation*, *17*(2), 2002, 86–92.

[32] R. Goldbach, Some randomized algorithms for convex quadratic programming, *Applied Mathematics & Optimization*, *39*(1), 1999, 121–142.

[33] S.S. Chern & W.H. Chern, *Lecture notes of differential geometry* (Beijing: University of Beijing, 2001).

[34] R. Fletcher, *Practical methods of optimization* (New York: Wiley, 1987).

[35] C.W. Wampler, Manipulator inverse kinematic solutions based on vector formulations and damped least-squares methods, *IEEE Trans. on Systems, Man, and Cybemetics*, *SMC-16*, 1986, 93–101.

[36] N.S. Bedrossian, Classification of singular configurations for redundant manipulators, *Proc. IEEE Int. Conf. on Robotics and Automation*, New York, 1990, 818–823.

[37] J. Seng, O. Neil, & A. Kevin, Escapability of singular configuration for redundant manipulators via self-motion, *Proc. IEEE/RSJ Int. Conf. on Intelligent Robots and Systems*, Pittsburgh, PA, USA, Part 3, 1995, 78–83.

Biographies

Changwu Qiu received his B.Sc. and M.Sc. degrees in mechanical engineering and automation from the Northeastern University, China, in 2000 and 2003, respectively. He is now a doctoral graduate at the Research Institute of Robotics, Shanghai Jiao Tong University, China. His research interests are in the dynamics control of redundant manipulators, modelling and control of mobile service dual arm robot, and 3D simulation of robot dynamics.

Qixin Cao is a professor at the School of Mechanical and Power Energy Engineering, Shanghai Jiao Tong University, China. His current research interests are machine vision, autonomous mobile robot control, and pattern recognition.

Yijun Sun is a doctoral graduate at the Research Institute of Robotics, Shanghai Jiao Tong University, China. His research interests are in the dynamic walking of humanoid robot and 3D simulation of robots.