Robotica (2015) volume 33, pp. 295–313. © Cambridge University Press 2014
doi:10.1017/S0263574714000344

Fuzzy weighted subtask controller for redundant manipulator

Young jun Yoo†, Dae sung Jung†, Yu jin Jang‡ and Sang chul Won†*

†Department of Electronic and Electrical Engineering, POSTECH, Pohang, Gyeongbuk, Republic of Korea
‡Department of Information and Communication Engineering, Dongguk University, Gyeong Ju, Gyeongsangbuk-Do, Republic of Korea

(Accepted January 31, 2014. First published online: February 28, 2014)

SUMMARY

We propose a fuzzy weighted subtask controller for a redundant robot manipulator. To expand the feasibility of the inverse kinematic solution, we introduce a weighted pseudo-inverse that changes the null-space of the Jacobian. The weights of elements in the pseudo-inverse are obtained using fuzzy rules that are related to the null-space velocity tracking error. With the pseudo-inverse, we develop a task space controller to track a desired task space trajectory and subtask control input. We propose a weighted subtask controller for multiple subtasks. The results of a simulation and experiment using a seven-degree-of-freedom whole arm manipulator robot show the effectiveness of the proposed controller with multiple subtasks.

KEYWORDS: Redundant manipulators; Control of robotic systems; Subtask control; Self-motion; Fuzzy weighted pseudo-inverse.

1. Introduction

A redundant manipulator has more degrees of freedom (DOF) than required to execute a given task, and it has multiple inverse kinematic solutions that can be solved using several criteria.[1–24] Two types of approaches have been studied to solve the redundancy resolution; the first solution uses the self-motion of the joint (SMJ)[1–14,19–24] and the second (non-SMJ) uses other criteria.[15–17] The SMJ method uses the null-space of the manipulator's Jacobian without changing the position of the end-effector whereas the non-SMJ method performs the subtask directly using extra DOF.

The SMJ method utilizes link velocity, not affecting task space velocity, in the null-space of the Jacobian. For a desired end-effector trajectory, controllers using the SMJ method should be designed to select a reasonable joint space trajectory that satisfies both control (stability and boundedness of all signals) and mechanical constraints (singularities and joint limit avoidance). A task space controller is designed[3, 4, 19–24] to track both the desired trajectory and the subtask control input that satisfies these constraints. In this controller, the subtask objective is secondary to the tracking objective. A subtask controller is selected using a gradient projection (GP) methods or auxiliary positive function (APF) methods.

The GP[1–7] method is a typical SMJ method. This method is initially applied to avoid joint limits.[7] After the GP is applied, other various subtask objectives are applied to maximize manipulability[1,4] and avoid joint limits[1,4,7] or obstacles.[1,2,18] This conventional method has the property that the gradient value of the subtask function is zero at an extremum point. If the subtask function value is not at an extremum point, then an additional subtask input is applied to the null-space of the Jacobian. Another approach used to project the subtask function on the joint velocity domain is the APF method.[19–24] In this method, an auxiliary signal is defined; an auxiliary signal is bounded when the proposed sufficient conditions are satisfied using the Lyapunov theorem.

* Corresponding author. E-mail: won@postech.ac.kr

https://doi.org/10.1017/S0263574714000344 Published online by Cambridge University Press

In practice, multiple subtasks are required during manipulator operations. For example, avoidance of joint limits and singularities should be performed simultaneously when the manipulator tracks a trajectory. For the GP method, numerous studies have considered how to successfully complete multi-objective subtasks.[2,4,5,8–14] The simplest approach sums subtask functions and applies the GP method;[2,4,5] however, in this approach, subtask objectives may not be adequately met because the GP method determines a locally-optimal solution for each subtask. The extended Jacobian,[12] augmented Jacobian,[13,14] and task priority[8–10] approaches have been studied when the manipulator is required to perform several subtasks. The problem with these approaches however, is that if the number of subtasks is increased, the null-space projection of the Jacobian strongly constrains the motion of the joints and may result in the manipulator failing to complete a task.[17] As a result, these techniques can be applied only when the number of subtasks is smaller than the degree of redundancy.

The ultimate boundedness conditions of the auxiliary signal have been studied using the controller based on APF for both a single subtask[19–21] and multiple subtasks.[24] The subtask objectives are satisfied when the subtask function values are positive. Therefore, the subtask controller input is applied to the null-space of the Jacobian to keep subtask function values positive. However, if the null-space of the Jacobian strongly constrains the input of the subtask controller signal, the subtasks cannot be satisfied when the APF method is applied. The null-space of the Jacobian is determined by the Jacobian and its pseudo-inverse. For a controller based on APF, the pseudo-inverse is the least-norm solution of the same portion of each joint velocity. The pseudo-inverse, however, may cause the null-space of the Jacobian to restrict the subtask control input. To relax the constraints on the null-space, we introduce a weighted pseudo-inverse that considers the joint priority. The weighted pseudo-inverse, based on the null-space error, provides the movement of corresponding joints. This movement relaxes the constraints on the null-space of the Jacobian.

In this paper, a fuzzy weighted subtask controller was proposed for a redundant manipulator. We extended the feasibility of the inverse kinematic solution when a desired trajectory is given, and we constructed a gain matrix using fuzzy rules related to the null-space velocity tracking error. The fuzzy parameters of the membership function were selected by minimizing the norm of the weighted joint velocity to obtain the global resolution of the redundancy. Using the weighted pseudo-inverse, a task space controller was designed to track both the desired trajectory and the subtask control input. A fuzzy weighted subtask controller was proposed for multiple subtasks. The results of the simulation and experiment using a seven-DOF whole arm manipulator showed the feasibility of the proposed controller with multiple subtasks.

This paper is organized as follows. In Section 2, we present preliminary information about the robot and pseudo-inverse. In Section 3, we introduce the controller using APF in the velocity domain. In Section 4, we develop the weighted pseudo-inverse, the task space controller, and the subtask controller. In Section 5, we present a simulation and experiment that show the effectiveness of the proposed controller with multiple subtasks. In Section 6, we present our conclusions.

2. Preliminary

2.1. Kinematic and dynamic models

The kinematic model of the robot manipulator is given by

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

where $x(t) \in \mathbb{R}^m$ is the end-effector position in the task space, $f(q) \in \mathbb{R}^m$ is forward kinematics, and $q(t) \in \mathbb{R}^n$ is the joint position.

Taking the time derivative of (1) yields

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

where $J(q) = \frac{\partial f(q)}{\partial q}$ is the Jacobian and $\dot{q}$ is joint velocity. The dynamics of an n-DOF revolute manipulator is

$$M(q)\ddot{q} + V_m(q, \dot{q})\dot{q} + G(q) + F(\dot{q}) = \tau, \tag{3}$$

https://doi.org/10.1017/S0263574714000344 Published online by Cambridge University Press

where q, $\dot{q}$, $\ddot{q}$ (all$\in \mathbb{R}^n$) represent the position, velocity, and acceleration in the joint-space, respectively. $M(q) \in \mathbb{R}^{n\times n}$ is the inertia matrix, $V_m(q, \dot{q}) \in \mathbb{R}^{n\times n}$ represents the centripetal-Coriolis matrix, $G(q) \in \mathbb{R}^n$ denotes the gravity effects, $F(\dot{q}) \in \mathbb{R}^n$ represents the friction effect, and $\tau(t) \in \mathbb{R}^n$ is the input torque.

The properties found in refs. [19]–[21] are utilized in control development.

Property 1. *The inertia matrix $M(q)$ is symmetric and positive-definite and satisfies the following inequality:*

$$m_1||\zeta||^2 \leq \zeta^T M(q)\zeta \leq m_2||\zeta||^2, \tag{4}$$

where $m_1, m_2 \in \mathbb{R}$ are positive constants and $||\cdot||$ is the Euclidean norm.

Property 2. *The left-hand side of (3) can be linearly parameterized as*

$$M(q)\ddot{q} + V_m(q, \dot{q})\dot{q} + G(q) + F(\dot{q}) = Y(q, \dot{q}, \ddot{q})\phi, \tag{5}$$

where $Y(q, \dot{q}, \ddot{q}) \in \mathbb{R}^{n\times r}$ is the regression matrix and $\phi \in \mathbb{R}^r$ contains constant system parameters (e.g., mass, inertia).

2.2. The pseudo-inverse and its properties

To resolve the redundancy of a redundant manipulator, the pseudo-inverse of $J(q)$ is needed. The weighted pseudo-inverse of $J(q)$ is denoted by

$$J_w^+ \overset{\triangle}{=} W^{-1}J^T(JW^{-1}J^T)^{-1} \in \mathbb{R}^{n\times m}, \tag{6}$$

where $W = \text{diag}(w_1, w_2, \dots, w_n) \in \mathbb{R}^{n\times n}$ and $w_i \in \mathbb{R}^+$ is the ith component of W. This weighted pseudo-inverse is obtained by solving the least norm problem,

$$\min_{\dot{q}} \frac{1}{2}\dot{q}^T W\dot{q}, \tag{7}$$
$$\text{subject to } \dot{x} = J\dot{q}.$$

Note that w_i in (6) denotes the gain of the ith joint in (7). For example, if W is the identity matrix I_n, the pseudo-inverse is obtained by considering the same portion of each joint; the pseudo-inverse of J is $J_w^+ = J^T(JJ^T)^{-1}$, which minimizes $\frac{1}{2}\dot{q}^T\dot{q}$.

The pseudo-inverse J_w^+ satisfies the following condition:

$$JJ_w^+ = I_m, \tag{8}$$

where I_m is the m×m identity matrix. The pseudo-inverse J_w^+ also satisfies the following Moore–Penrose conditions:

$$JJ_w^+J = J, \quad J_w^+JJ_w^+ = J_w^+, \tag{9}$$
$$(J_w^+J)^T = J_w^+J, \quad (JJ_w^+)^T = JJ_w^+. \tag{10}$$

The matrix $(I_n - J_w^+J)$ satisfies the following conditions:

$$(I_n - J_w^+J)(I_n - J_w^+J) = (I_n - J_w^+J), \tag{11}$$
$$J(I_n - J_w^+J) = 0_{m\times 1}, \tag{12}$$
$$(I_n - J_w^+J)^T = (I_n - J_w^+J), \tag{13}$$
$$(I_n - J_w^+J)J_w^+ = 0_{n\times 1}, \tag{14}$$

where I_n is the n×n identity matrix, $0_{n\times 1}$ and $0_{m\times 1}$ are zero vectors.

https://doi.org/10.1017/S0263574714000344 Published online by Cambridge University Press

2.3. The assumptions for subsequent development

Subsequent development is based on the following assumptions.

Assumption 1. *The weight matrix denoted by $W = diag(w_1, w_2, \ldots, w_n) \in \mathbb{R}^{n\times n}$ satisfies the following statements:*

$$\lambda_{W_m}||\zeta||^2 \le \zeta^T W \zeta \le \lambda_{W_M}||\zeta||^2, \tag{15}$$

where $\zeta \in \mathbb{R}^n$, and $\lambda_{W_m}(\lambda_{W_M})$ satisfying $0 < \lambda_{W_m} \le \lambda_{W_M} < \infty$ is the minimum (maximum) eigenvalue of W.

Remark 1. *During control development, $J_w^+(q) \in \mathcal{L}_\infty$ by* **Assumption 1** *and subtask control of manipulability [4,20,21].*

Assumption 2. *The dynamic and kinematic terms of the revolute robot manipulator, denoted by $M(q)$, $V_m(q, \dot{q})$, and $J(q)$ depend on $q(t)$ as arguments of trigonometric functions; hence, residuals are bounded for all $q(t) \in \mathcal{L}_\infty$ and $\dot{q}(t) \in \mathcal{L}_\infty$.* **Assumption 1** *and* **Remark 1** *imply that if $x(t)$ is bounded, then $q(t)$ is a bounded signal.*

3. Subtask Control in the Velocity Domain Based on APF and its Feasibility

In this section, we introduce the subtask controller based on APF methods in the joint velocity domain. We also discuss the feasibility of the inverse kinematic solution. A controller based on APF can perform multiple subtask objectives since its solution has greater feasibility than a controller based on GP methods. However, a controller based on APF is restricted by the null-space of the Jacobian. In the next section, this restriction is relaxed by the proposed controller.

Based on (2), the following expression can be obtained for redundancy resolution in the joint velocity domain:

$$\dot{q} = J_w^+(q)\dot{x} + (I_n - J_w^+ J)g, \tag{16}$$

where $g(t) \in \mathbb{R}^n$ is a subtask controller. Multiplying both sides of (16) on the left by J shows that g changes the joint velocity, but it does not affect the task space velocity $\dot{x}$. This property enables us to perform various subtask objectives such as maximizing manipulability[1,4] and avoiding joint limits[1,4,7] or obstacles[1,2,18].

3.1. Method of auxiliary positive function

Subtask controllers using APF for a single subtask are introduced.[20] The controller proposed in ref. [20] is based on $W = I_n$ and $J_w^+ = J^T(JJ^T)^{-1}$, which minimizes $\frac{1}{2}\dot{q}^T\dot{q}$.

To develop the controller, an auxiliary signal is introduced. The APF y is given by

$$y \overset{\triangle}{=} \exp(-k_s\beta(q)), \tag{17}$$

where $k_s \in \mathbb{R}$ is a positive constant and $\beta(q)$ is a subtask function. Taking the time derivative of (17) yields

$$\dot{y} = J_s\dot{q}, \tag{18}$$

where $J_s = \frac{\partial y}{\partial q}$.

Adding and subtracting terms $J_s(I_n - J_w^+ J)(g - \dot{q})$ to the right-hand side of Eq. (18) yields

$$\begin{aligned}\dot{y} &= J_s\dot{q} + J_s(I_n - J_w^+ J)(g - \dot{q}) - J_s(I_n - J_w^+ J)(g - \dot{q})\\ &= J_s\dot{q} - J_s(I_n - J_w^+ J)\dot{q} + J_s(I_n - J_w^+ J)g - J_s\dot{e}_N\\ &= J_sJ_w^+ J\dot{q} + J_s(I_n - J_w^+ J)g - J_s\dot{e}_N\\ &= J_sJ_w^+\dot{x} + J_s(I_n - J_w^+ J)g - J_s\dot{e}_N,\end{aligned} \tag{19}$$

https://doi.org/10.1017/S0263574714000344 Published online by Cambridge University Press

where

$$\dot{e}_N = (I_n - J_w^+ J)(g - \dot{q}) \tag{20}$$

is the null-space velocity tracking error defined in ref. [3].

The subtask controller for a single subtask can be designed according to[20]

$$g = -k_s [J_s(I_n - J_w^+ J)]^T y. \tag{21}$$

Multiple subtask objectives are required when the robot operates. Extending APF methods from a single subtask to multiple subtasks is required. The auxiliary signal of the ith subtask is

$$y_i \overset{\triangle}{=} \exp(-k_s \beta_i(q)), \tag{22}$$

where $k_s \in \mathbb{R}$ is a positive constant and $\beta_i(q)$ is the ith subtask function. The time derivative of (22) is given by

$$\dot{y}_i = J_{si}\dot{q}, \tag{23}$$

where $J_{si} = \frac{\partial y_i}{\partial q}$.

By adding and subtracting $J_{si}(I_n - J_w^+ J)(g - \dot{q})$ to the right-hand side of Eq. (23), the time derivatives of the auxiliary signal can be extended to the ith subtask as

$$\dot{y}_i = J_{si} J_w^+ \dot{x} + J_{si}(I_n - J_w^+ J)g - J_{s1}\dot{e}_N. \tag{24}$$

The auxiliary signal vector is defined as $\mathbf{y} = [y_1^T \; y_2^T, \ldots, y_j^T]^T \in \mathbb{R}^j$. The time derivatives of $\mathbf{y}$ can be written as

$$\dot{\mathbf{y}} = \mathbb{J} J_w^+ \dot{x} + \mathbb{J}(I_n - J_w^+ J)g - \mathbb{J}\dot{e}_N, \tag{25}$$

where $\mathbb{J} = [J_{s1}^T, J_{s2}^T, \ldots, J_{sj}^T]^T \in \mathbb{R}^{j \times n}$ and j is the number of subtasks.

Finally, the subtask controller g for multiple subtasks based on APF[24] is designed as

$$g = -k_s \sum_{i=1}^{j} [J_{si}(I_n - J_w^+ J)]^T y_i, \tag{26}$$

where $J_w^+ = J^T(JJ^T)^{-1}$.

With (26) in hand, we discuss the feasibility of the inverse kinematic solution using APF in the next subsection.

Remark 2. *To perform subtask objectives, the subtask controller g includes the subtask function $\beta(q)$. For APF, g also includes the function $\beta(q)$ which will be kept positive. Gradient information is utilized to maximize $\beta(q)$ when $\beta(q) < 0$. However, when $\beta(q) > 0$, y results in small input to g since* $0 < exp(-k_s\beta(q)) < 1$.

3.2. The feasibility of APF with multiple subtasks

In this subsection, we analyze the subtask controller based on APF from the aspect of the inverse kinematic solution's feasibility. When multiple subtasks are applied, the objective function $\beta = \sum_{i=1}^{j} \beta_i(q)$ is the summation of each subtask function where $\beta_i(q)$ is the ith subtask function and j is the number of subtasks. The summation of each subtask function may cause the extremum point of β to change; this modification can result in a failure to satisfy the subtasks.

Each subtask function is quadratic and has a minimum point (Fig. 1(a), (b), (d), and (e)). The first subtask (Fig. 1(a) and (d)) is cyclicity.[1] The second (Fig. 1(b) and (e)) is joint limit avoidance.[1] β_1 obtains a minimum value of 0 at $[q_1, q_2] = [0.8, 1]$. β_2 obtains a minimum value of -0.75 at $[q_1, q_2] = [-1, -1]$. However, note that the summation of subtask functions changes the minimum

https://doi.org/10.1017/S0263574714000344 Published online by Cambridge University Press

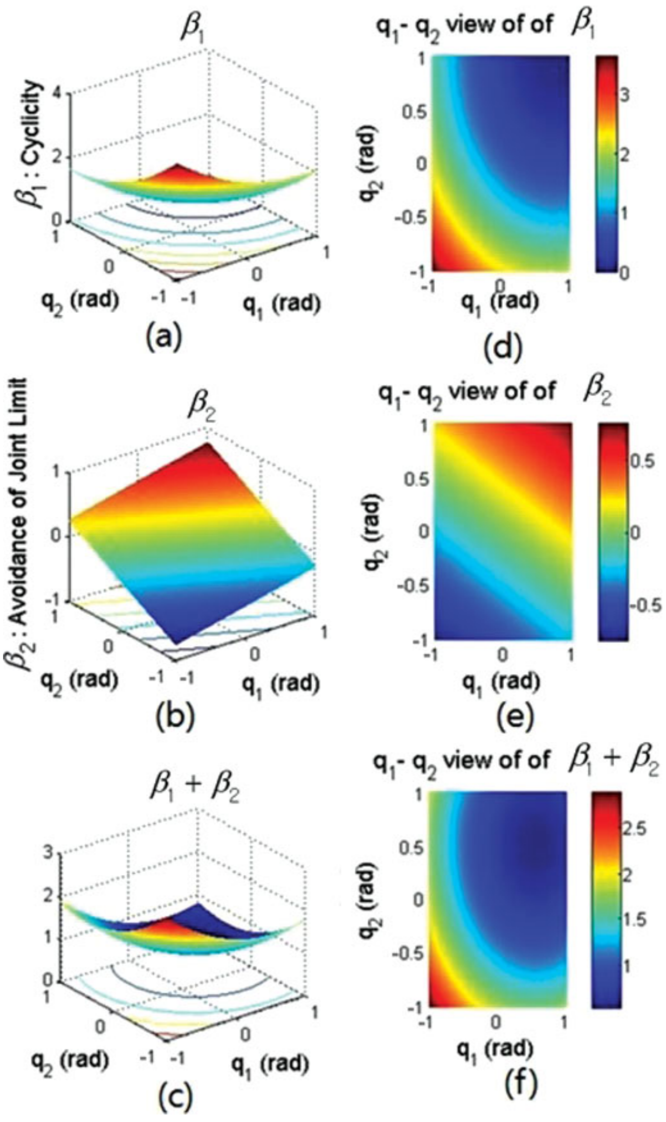


Fig. 1. Example of changing the extremum point for $q \in \mathbb{R}^2$: (a) cyclicity; (b) joint limit avoidance; (c) summation of the two subtask functions; (d) top view of β_1; (e) top view of β_2; (f) top view of the $\beta_1 + \beta_2$.

point (0.5347 at $[q_1, q_2] = [0.56, 0.51]$) (Fig. 1(c) and (f)). When this point changes, the joint limit avoidance requirement may not be satisfied because the minimum point is close to the joint limit.

Although the maximum or minimum point is changed, a controller based on APF can still perform multiple subtask objectives because it has time-varying gain that reduces the coupling effects of other subtask functions. The coupling effect can be analyzed by examining the time derivatives of the subtask function vector $\mathbb{B} = [\beta_1^T \beta_2^T \ldots, \beta_j^T]^T \in \mathbb{R}^j$. The Jacobian of y_i is

$$J_{si} = \frac{\partial y_i}{\partial q} = \frac{\partial}{\partial q}\exp(-k_s\beta_i(q)) = -k_s\frac{\partial \beta_i}{\partial q}y_i. \tag{27}$$

Substituting (27) into (26), the subtask controller based on APF can be rewritten as

$$\begin{aligned} g &= -k_s\sum_{i=1}^{j}[J_{si}(I_n - J_w^+ J)]^T y_i \\ &= (I_n - J_w^+ J)\sum_{i=1}^{j}\frac{\partial \beta_i}{\partial q}^T k_s^2 y_i^2. \end{aligned} \tag{28}$$

https://doi.org/10.1017/S0263574714000344 Published online by Cambridge University Press

Table I. Fuzzy Rules for weight component w_i.

	$\sum_{i=1}^{n} \|\|\dot{e}_N(i)\|\|$		
$\|\|\dot{e}_N(i)\|\|$	Small	Medium	Large
Small	$w_i =$ Medium	$w_i =$ Small	$w_i =$ Small
Large	$w_i =$ Large	$w_i =$ Large	$w_i =$ Medium

The time derivative of $\mathbb{B}$ with (28) is given by

$$\begin{aligned} \dot{\mathbb{B}} &= \frac{\partial \mathbb{B}}{\partial q}(J_w^+\dot{x} + (I_n - J_w^+J)g) \\ &= \frac{\partial \mathbb{B}}{\partial q}(J_w^+\dot{x} + (I_n - J_w^+J)\frac{\partial \mathbb{B}}{\partial q}^T (k_s^2\mathbb{Y})\mathbf{1}), \end{aligned} \tag{29}$$

where $\mathbb{Y} \overset{\triangle}{=} \text{diag}(y_1^2, y_2^2, \dots, y_j^2) \in \mathbb{R}^{j\times j}$ and $\mathbf{1} \overset{\triangle}{=} [1, 1, \dots, 1, 1]^T \in \mathbb{R}^j$. Therefore, the time derivative of the lth subtask function is

$$\dot{\beta}_l = \frac{\partial \beta_l}{\partial q} J_w^+\dot{x} + \frac{\partial \beta_l}{\partial q}(I_n - J_w^+J)\sum_{i=1}^{j} \frac{\partial \beta_i}{\partial q}^T k_s^2 y_i^2. \tag{30}$$

The term $\frac{\partial \beta_l}{\partial q}(I_n - J_w^+J)\sum_{i=1}^{j} \frac{\partial \beta_i}{\partial q}^T k_s^2 y_i^2$ is the coupling effect of β_l and the other subtask β_i (for $i = 1, 2, \dots, j$). However, when the other subtask is satisfied, the coupling effect of β_l is reduced because $\beta_i > 0$ making $0 < y_i < 1$ ($\frac{\partial \beta_i}{\partial q}^T k_s^2 y_i^2 \simeq 0$). This property reduces the coupling effect of β_l on the other subtask β_i.

If the null-space of the Jacobian strongly constrains the subtask input g, the subtasks cannot be satisfied when the controller based on APF is applied. The subtask control input g can be divided into two parts: the first part is $\sum_{i=1}^{j}(I_n - J_w^+J)J_{si}^T$ which is related to the null-space of the Jacobian and the second part is $(-k_s y_i)$ which is a function of the subtask function β_i. If the null-space of J restricts g (one element of $\sum_{i=1}^{j}(I_n - J_w^+J)J_{si}^T \simeq 0$) in the first part, the subtask input g is not sufficient to satisfy the subtask objective even though $y_i > 1$ ($\beta_i < 0$) in the second part. Therefore, we relax the condition $W = I$ (each joint velocity has the same gain) to change the null-space of the Jacobian $(I_n - J_w^+J)$.

4. The Proposed Controllers Using a Fuzzy Weighted Matrix

In this section, we propose a weighted subtask controller for a redundant manipulator. To relax the constraints on the null-space of the Jacobian, we use fuzzy rules to define the coefficient of W related to the null-space velocity error given by (20). Based on this matrix, we obtain the weighted pseudo-inverse $J_w^+ = W^{-1}J^T(JW^{-1}J^T)^{-1}$. We then develop a task space controller which tracks x_d and g. After the task space controller is developed, we propose a subtask controller for multiple subtasks.

4.1. The fuzzy weighted matrix

To design the weighted subtask controller, we first develop the coefficients of the weighted matrix $W = \text{diag}(w_1, w_2, \dots, w_n)$. The coefficient w_i ($i = 1, \dots, n$) is related to the gain of $\dot{q}_i$ in (7). If w_i is large, the weighted pseudo-inverse J_w^+ is the least-norm solution considering a large portion of $\dot{q}_i$ movement. We design w_i to relax the condition that $W = I_n$ in the APF methods. We design w_i in this manner to decide which $\dot{q}_i$ will have priority. The priority of $\dot{q}_i$ can be expressed using fuzzy rules (Table I), where $\dot{e}_N(i)$ is the ith component of $\dot{e}_N$ (20).

We assign the priority w_i using fuzzy rules that are related to the terms $||\dot{e}_N(i)||$ and $\sum_{i=1}^{n} ||\dot{e}_N(i)||$. A large value of $||\dot{e}_N(i)||$ indicates that $\dot{q}_i$ cannot follow the ith component of g. We increase the weight of the ith joint to follow the ith component of the subtask controller. For the reason, w_i is large

Table II. Fuzzy operations.

And	Implication	Aggregation	Defuzzification
Min	Prod	Max	Centroid

when $||\dot{e}_N(i)||$ is large, and $\sum_{i=1}^{n} ||\dot{e}_N(i)||$ is small or medium (Table I). However, if both $||\dot{e}_N(i)||$ and $\sum_{i=1}^{n} ||\dot{e}_N(i)||$ are large, the possibility exists that in addition to the ith component joint being unable to follow the ith component of subtask controller g, other joint components may have a large null-space velocity error ($||\dot{e}_N(k)|| \gg 0$). In this case, we assign the medium gain. If $||\dot{e}_N(i)||$ is small and $\sum_{i=1}^{n} ||\dot{e}_N(i)||$ is medium or large, we do not need to assign a large gain because the ith joint follows the ith component of g with a small error. However, when both $||\dot{e}_N(i)||$ and $\sum_{i=1}^{n} ||\dot{e}_N(i)||$ are small, all joints follow g, and therefore, we assign the medium gain to each joint (Table I).

The value of w_i is determined using fuzzy rules (Table I) and operations (Table II). From the input membership function $\mu_{R_k:||\dot{e}_N||}$ (Fig. 1(a)), $\mu_{R_k:\sum||\dot{e}_N||}$ (Fig. 1(b)) and the output membership function $\mu_{R_k:w_i}$ (Fig. 1(c)), the kth fuzzy rule R_k is implemented using the fuzzy implication function $\mu_{R_k} \overset{\triangle}{=} [\,\mu_{R_k:||\dot{e}_N||}\ \mathit{And}\ \mu_{R_k:\sum||\dot{e}_N||}] \rightarrow \mu_{R_k:w_i}$. The fuzzy rules are aggregated for the output fuzzy set as

$$\mu_{w_i} = \mathrm{Agg}(\mu_{R_1}, \mu_{R_2}, \ldots, \mu_{R_6}), \tag{31}$$

where Agg(·) is the aggregation operator. Using centroid defuzzification (Table II), w_i is computed as

$$w_i = \frac{\int \mu_{w_i}(w_i) w_i dw_i}{\int \mu_{w_i}(w_i) dw_i}. \tag{32}$$

Figure 2 shows the input and output membership functions. The parameters z_p ($p = 1, \ldots, 6$) can be found by minimizing the following quadratic performance criterion:

$$C = \int_{t_0}^{t_f} \dot{q}^T W \dot{q}\, dt, \tag{33}$$

$$\text{subject to } z_p \in [z_p^{\min}, z_p^{\max}] \text{ for } p = 1 \text{ to } 6,$$

where t_0 is the initial time and t_f is the final time of robot operations. Optimization techniques with an integral-type performance index (33) are introduced to obtain the globally optimal resolution of redundancy of the robot manipulators.[25–27] By solving (33) using the optimization algorithm, uDEAS,[28–30] the parameters z_p are determined.

The overall process of the proposed controller is illustrated in Fig. 3. The dynamic and kinematic models of the robot with the proposed controller, discussed in the next sections, generate the null-space tracking error $\dot{e}_N$. The proposed fuzzy rules (Table I) give the fuzzy weighted matrix. This matrix is used to calculate J_w^+. The pseudo-inverse of J is subsequently used to generate $\dot{e}_N$ that provides W repeatedly.

4.2. *Task space controller*

The most important control objective is that the end effector position of the manipulator tracks a desired trajectory. To track the trajectory, the controller is designed to apply the torque input.

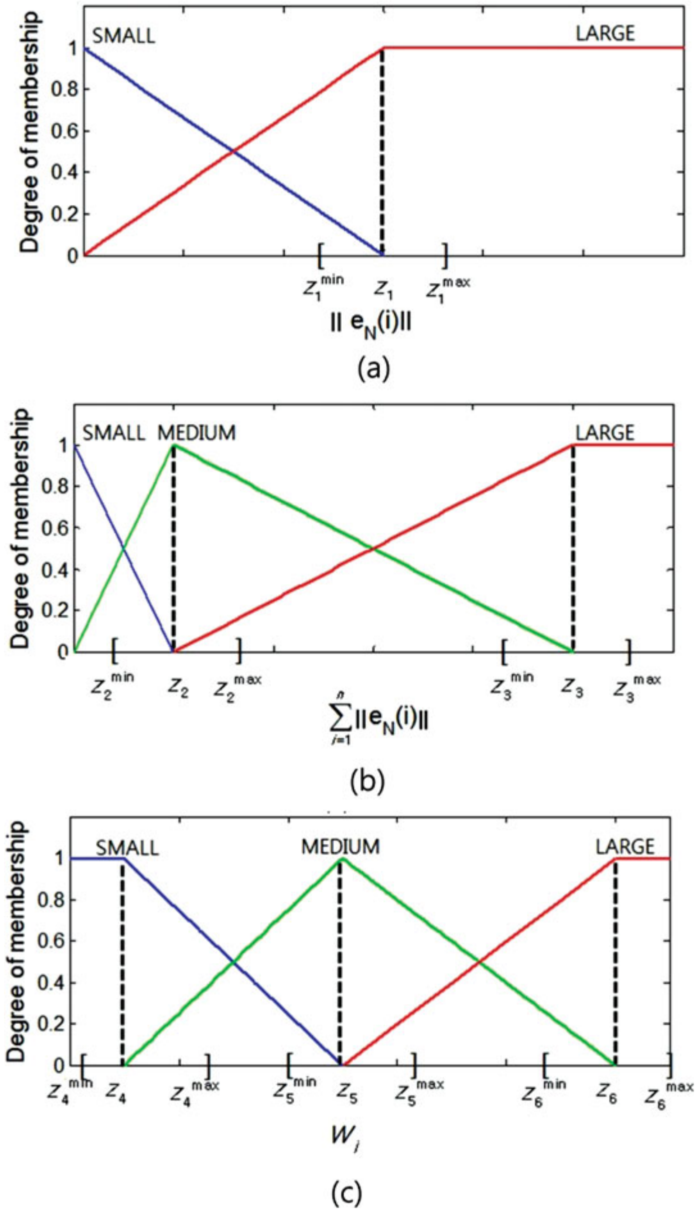
The task-space error $e(t) \in \mathbb{R}^m$ is defined as

$$e = x_d - x, \tag{34}$$

where $x_d(t) \in \mathbb{R}^m$ is the desired task-space trajectory. Differentiating (34) with respect to time and substituting (2), we obtain the following equation:

$$\dot{e} = \dot{x}_d - J\dot{q} = -\alpha_k e + \alpha_k e + \dot{x}_d - J\dot{q}, \tag{35}$$

https://doi.org/10.1017/S0263574714000344 Published online by Cambridge University Press



(a)

(b)

(c)

Fig. 2. Membership functions of (a) $||\dot{e}_N(i)||$; (b) $\sum_{i=1}^{n} ||\dot{e}_N(i)||$; (c) w_i.

where $\alpha_k \in \mathbb{R}^{m\times m}$ is a diagonal, positive-definite gain matrix. Using pseudo-inverse properties (8) and (12), (35) can be rewritten as

$$\dot{e} = -\alpha_k e + J[J_w^+(\alpha_k e + \dot{x}_d) - \dot{q} + (I_n - J_w^+ J)g]. \tag{36}$$

The filtered tracking error signal $r(t)$ is defined as

$$r = J_w^+(\alpha_k e + \dot{x}_d) - \dot{q} + (I_n - J_w^+ J)g. \tag{37}$$

After multiplying by $(I_n - J_w^+ J)$ and using (14), (20) is rewritten as

$$\dot{e}_N = (I_n - J_w^+ J)r. \tag{38}$$

Therefore, when $r(t)$ is regulated, $\dot{e}_N$(t) is also regulated so that subtask control is achieved.

https://doi.org/10.1017/S0263574714000344 Published online by Cambridge University Press

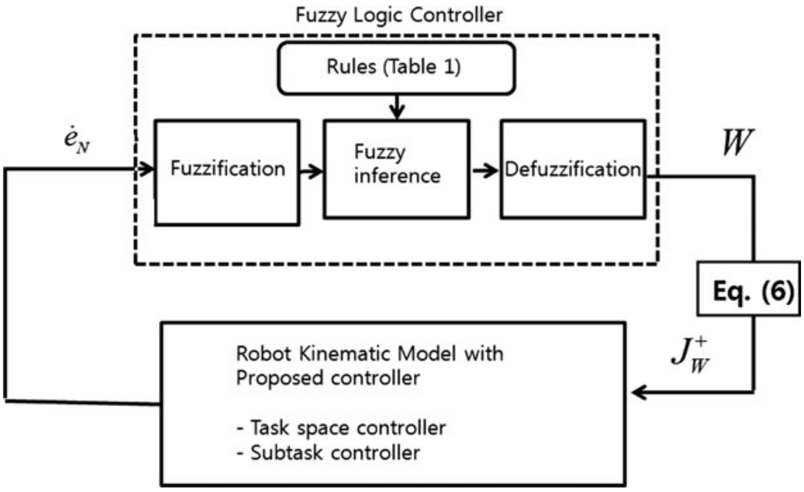


Fig. 3. Overall process of the proposed controller.

Substituting (37) into (36) yields

$$\dot{e} = -\alpha_k e + Jr. \tag{39}$$

By differentiating (37), pre-multiplying by $M(q)$ and using (3) yields

$$M\dot{r} = -V_m r + Y\phi - \tau, \tag{40}$$

where $Y\phi = M\frac{d}{dt}[J_w^+(\dot{x}_d + \alpha_k e) + (I_n - J_w^+ J)g] + V_m[J_w^+(\dot{x}_d + \alpha_k e) + (I_n - J_w^+ J)g] + G(q) + F(\dot{q})$.

The control torque input $\tau(t)$ is designed as follows:

$$\tau = Y\phi + Kr + J^T e, \tag{41}$$

where K is a positive constant gain matrix. Then, we propose the following **Theorem** for tracking x_d and g.

Theorem 1. *The control input (41) ensures that both signals $e(t)$ and $\dot{e}_N(t)$ are bounded by an exponential envelope.*

PROOF: The Lyapunov candidate is chosen as

$$V_1 = \frac{1}{2}\begin{bmatrix} e \\ r \end{bmatrix}^T \begin{bmatrix} I_m & 0 \\ 0 & M \end{bmatrix} \begin{bmatrix} e \\ r \end{bmatrix}. \tag{42}$$

Substituting (41) into (40) yields

$$M\dot{r} = -V_m r - Kr - J^T e. \tag{43}$$

Taking the time derivative of (42), then substituting (39) and (43) yields

$$\begin{aligned} \dot{V}_1 &= \begin{bmatrix} e \\ r \end{bmatrix}^T \begin{bmatrix} I_m & 0 \\ 0 & M \end{bmatrix} \begin{bmatrix} \dot{e} \\ \dot{r} \end{bmatrix} + \frac{1}{2}\begin{bmatrix} e \\ r \end{bmatrix}^T \begin{bmatrix} 0 & 0 \\ 0 & \dot{M} \end{bmatrix} \begin{bmatrix} e \\ r \end{bmatrix} \\ &= -\begin{bmatrix} e \\ r \end{bmatrix}^T \begin{bmatrix} \alpha_k & 0 \\ 0 & K \end{bmatrix} \begin{bmatrix} e \\ r \end{bmatrix} < 0. \end{aligned} \tag{44}$$

https://doi.org/10.1017/S0263574714000344 Published online by Cambridge University Press

4.3. Subtask controller with multiple subtasks

In this section, we develop a fuzzy weighted subtask controller using the weighted pseudo-inverse from Section 4.1 for multiple subtasks.

The proposed subtask controller is designed as

$$g = -k_s \sum_{i=1}^{j} [J_{si}(I_n - J_w^+ J)]^T y_i. \tag{45}$$

The following **Theorem** relates to the weighted subtask controller when multiple subtasks are given.

Theorem 2. *If the following sufficient conditions hold*

$$||\mathbb{J}(I_n - J_w^+ J)||^2 > \bar{\delta}_w, \tag{46}$$

$$||\mathbb{J} J_w^+ \dot{x} - \mathbb{J} \dot{e}_N|| \leq \delta_{1w}, \tag{47}$$

$$k_s > \frac{1}{\delta_{2w}\bar{\delta}_w}, \tag{48}$$

then the subtask controller g of (45) makes **y** *ultimately bounded in the following sense:*

$$||\mathbf{y}|| \leq \left[||\mathbf{y}(t_0)||^2 exp(-2\gamma_w t) + \frac{\varepsilon_w}{\gamma_w} \right]^{\frac{1}{2}}, \tag{49}$$

where $\varepsilon_w = \delta_{1w}^2 \delta_{2w}$ *and* $\gamma_w = (k_s \bar{\delta}_w - \frac{1}{\delta_{2w}})$.

PROOF: The Lyapunov candidate is

$$V_2 = \frac{1}{2}\mathbf{y}^T \mathbf{y}. \tag{50}$$

The time derivative of V_2 is

$$\begin{aligned} \dot{V}_2 &= \mathbf{y}^T \mathbb{J}(J_w^+ \dot{x} + (I_n - J_w^+ J)g - \dot{e}_N) \\ &= \mathbf{y}^T \mathbb{J}(J_w^+ \dot{x} - k_s(I_n - J_w^+ J)\mathbb{J}^T \mathbf{y} - \dot{e}_N) \\ &= -k_s \mathbf{y}^T ||\mathbb{J}(I_n - J_w^+ J)||\mathbf{y} + \mathbf{y}^T [\mathbb{J} J_w^+ \dot{x} - \mathbb{J}\dot{e}_N]. \end{aligned} \tag{51}$$

From **Assumption 2** and **Remark 1**, $J(q)$ and $J_w^+(q) \in \mathcal{L}_\infty$ have full ranks. **Assumption 2** shows that $q(t) \in \mathcal{L}_\infty$; therefore, $J_{si}(q) \in \mathcal{L}_\infty$ for all subtasks $(i = 1, 2, \ldots, j)$. From this, the following condition is derived:

$$||\mathbb{J}(I_n - J_w^+ J)|| > \bar{\delta}_w, \tag{52}$$

where $\bar{\delta}_w \in \mathbb{R}$ is a positive constant.

The following upper bound on the desired trajectory is assumed to be

$$||\mathbb{J} J_w^+ \dot{x} - \mathbb{J} \dot{e}_N|| \leq \delta_{1w}, \tag{53}$$

where $\delta_{1w} \in \mathbb{R}$ is a positive constant. After utilizing the bounds δ_{1w} and $\bar{\delta}_w$, (51) can be expressed as

$$\dot{V}_2 \leq -k_s \bar{\delta}_w \mathbf{y}^T \mathbf{y} + \delta_{1w} \mathbf{y}. \tag{54}$$

Using the following inequality,

$$|\delta_{1w}\mathbf{y}| \leq \frac{1}{\delta_{2w}}\mathbf{y}^T \mathbf{y} + \delta_{1w}^2 \delta_{2w}, \tag{55}$$

https://doi.org/10.1017/S0263574714000344 Published online by Cambridge University Press

Table III. The DH parameters of the WAM.

	i (joint)						
	1	2	3	4	5	6	7
a_i	0	0	0.045	−0.045	0	0	0
α_i	$-\pi/2$	$\pi/2$	$-\pi/2$	$\pi/2$	$-\pi/2$	$\pi/2$	0
d_i	0	0	0.55	0	0.3	0	0.060
q_i	q_1	q_2	q_3	q_4	q_5	q_6	q_7

(54) can be written as

$$\dot{V}_2 \leq -\left(k_s\bar{\delta}_w - \frac{1}{\delta_{2w}}\right)\mathbf{y}^T\mathbf{y} + \delta_{1w}^2\delta_{2w}, \tag{56}$$

where $\delta_{2w} \in \mathbb{R}$ is a positive constant. The values of k_s,$\bar{\delta}_w$ and δ_{2w} are chosen to satisfy

$$\gamma_w = \left(k_s\bar{\delta}_w - \frac{1}{\delta_{2w}}\right) > 0. \tag{57}$$

If (57) is satisfied, (56) can be expressed as

$$\dot{V}_2 \leq -\gamma_w\mathbf{y}^T\mathbf{y} + \varepsilon_w, \tag{58}$$

where γ_w and ε_w are bounding constants. Substituting (50) into (58), the following equation can be derived:

$$\dot{V}_2 \leq -2\gamma_w V_2 + \varepsilon_w. \tag{59}$$

Integrating both sides of Eq. (59) yields

$$V_2(t) \leq V_2(t_0)\exp(-2\gamma_w t) + \frac{\varepsilon_w}{2\gamma_w}(1 - \exp(-2\gamma_w t)). \tag{60}$$

After utilizing (60), the following upper bound of $\mathbf{y}$ can be obtained as

$$|\mathbf{y}| \leq [|\mathbf{y}(t_0)|^2\exp(-2\gamma_w t) + \frac{\varepsilon_w}{\gamma_w}]^{\frac{1}{2}}; \tag{61}$$

thus, $\mathbf{y}(t) \in \mathcal{L}_\infty$. From (45), we see that g is in $\mathcal{L}_\infty$. Using J_{si},J,$J_w^+ \in \mathcal{L}_\infty$, (45), (61) and **Assumption 2**, $\dot{\mathbf{y}}$ is also in $\mathcal{L}_\infty$. After taking the time derivative of (45), $\frac{\partial g}{\partial q}$ is clearly in $\mathcal{L}_\infty$.

5. Simulation and Experimental Results

In this section, we use an example to demonstrate the effectiveness of the proposed controller. The WAM (Barrett Technology^{TM}), commercial seven-DOF robot is used for the computer simulation and experiment. The Denavit-Hartenburg (DH) parameters of the WAM are listed in Table III.

The main task was defined as the end-effector that tracks the desired "figure-eight"-shaped trajectory,[20]

$$x_d = \begin{bmatrix} -0.1 + 0.1\sin(0.2t) \\ 0.3 \\ 0.3 + 0.2\cos(0.1t) \end{bmatrix}.$$

https://doi.org/10.1017/S0263574714000344 Published online by Cambridge University Press

Table IV. Maximum and minimum joint angle limits.

	i (joint)						
	1	2	3	4	5	6	7
$q_i^{\max}$(rad)	−2.6	−2.0	−2.8	−0.9	−4.8	−1.6	−2.2
$q_i^{\min}$(rad)	2.6	2.0	2.8	3.1	1.3	1.6	2.2

The subtasks were defined to avoid joint limits and keep the manipulator away from the singularity.

5.1. Simulation results

We defined the subtask objective functions that avoid the joint angle limits[4] as

$$\beta_i\ (i = 1,\ldots,7)\ = \left(\frac{q_i - q_i^{\min}}{q_i^{\max} - q_i^{\min}}\right)\left(\frac{q_i^{\max} - q_i}{q_i^{\max} - q_i^{\min}}\right), \tag{62}$$

where $q_i^{\max}$ is the maximum ith joint angle limit and $q_i^{\min}$ is the minimum ith joint angle limit (Table IV). In order for the joint angles to remain within their limits, β_i must be positive for the duration of operation.

For singularity avoidance,[4,20,21] we defined

$$\beta_8 = \sqrt{\det(JJ^T)}. \tag{63}$$

When β_8 is close to zero, the robot is close to the singular configuration. In order for J_w^+ to exist, β_8 should not be zero.

The diagonal elements of W are found using the fuzzy rules in Table I and the membership functions in Fig. 2. The range of parameters z_i was chosen as follows:

$$[z_1^{\min}, z_1^{\max}] = [0.1,\ 2], \qquad [z_2^{\min}, z_2^{\max}] = [0.1,\ 4],$$
$$[z_3^{\min}, z_3^{\max}] = [8,\ 15], \qquad [z_4^{\min}, z_4^{\max}] = [0.1,\ 0.2],$$
$$[z_5^{\min}, z_5^{\max}] = [0.3,\ 0.5], \qquad [z_6^{\min}, z_6^{\max}] = [0.6,\ 1].$$

By minimizing (33), the membership function parameters are found to be $z_1 = 1.8959$, $z_2 = 0.1002$, $z_3 = 8.0866$, $z_4 = 0.1508$, $z_5 = 0.3145$, and $z_6 = 0.6014$. For minimization, the uDEAS optimization algorithm[28–30] was used. The gains were chosen as

$$\alpha_k = \mathrm{diag}(2, 2, 2),\ k_s = 5,$$
$$\text{and}\ \ K = \mathrm{diag}(40, 40, 40, 40, 40, 40, 40).$$

The subtask control input,

$$g = -k_s \sum_{i=1}^{8} (I_n - J_w^+ J) J_{si}^T y_i,$$

were applied to (16) according to the following conditions:

$$\begin{cases} W = I_7 \\ \text{(conventional APF controller)}, \\ W = \mathrm{diag}(w_1, w_2, \ldots, w_7) \\ \text{(proposed controller)}. \end{cases}$$

https://doi.org/10.1017/S0263574714000344 Published online by Cambridge University Press

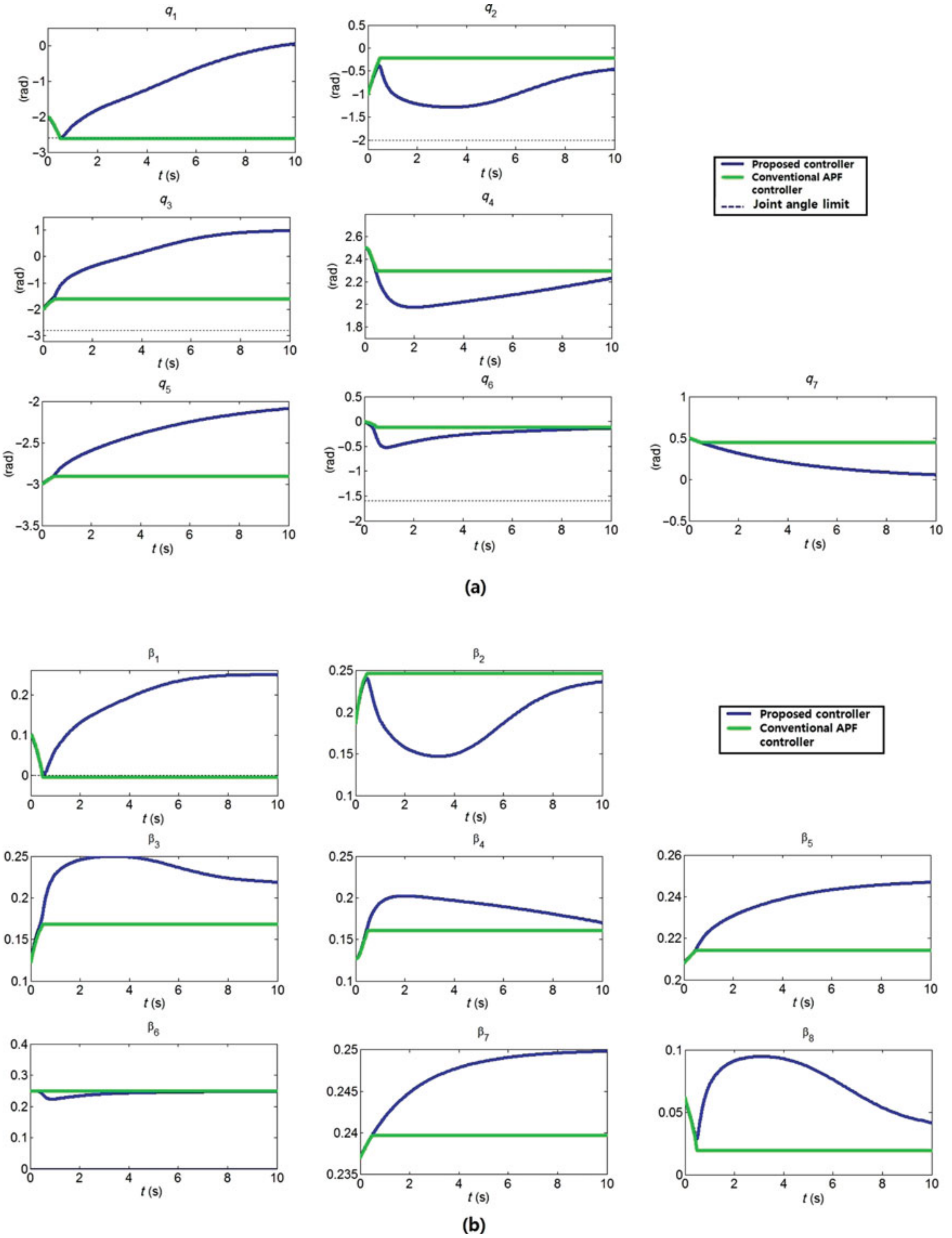


Fig. 4. (a) Joint angles ($q_1, \ldots, q_7$) of the simulation example; (b) Subtask functions for the simulation example where β_i $(i = 1, \ldots, 7)$ is the avoidance of the i-th joint limit and β_8 is manipulability.

The initial condition of q was chosen to be

$$q(t_0) = \begin{bmatrix} -2, & -1, & -2, & 2.5, & -3, & 0, & 0.5 \end{bmatrix}^T \quad \text{(rad)}. \tag{64}$$

The simulation results are summarized in Figs. 4 and 5. Figure 4(a) shows joint angles $q_1, \ldots,$ q_7. As one can see in the figure, q_1 of the conventional APF controller is over its joint limit whereas

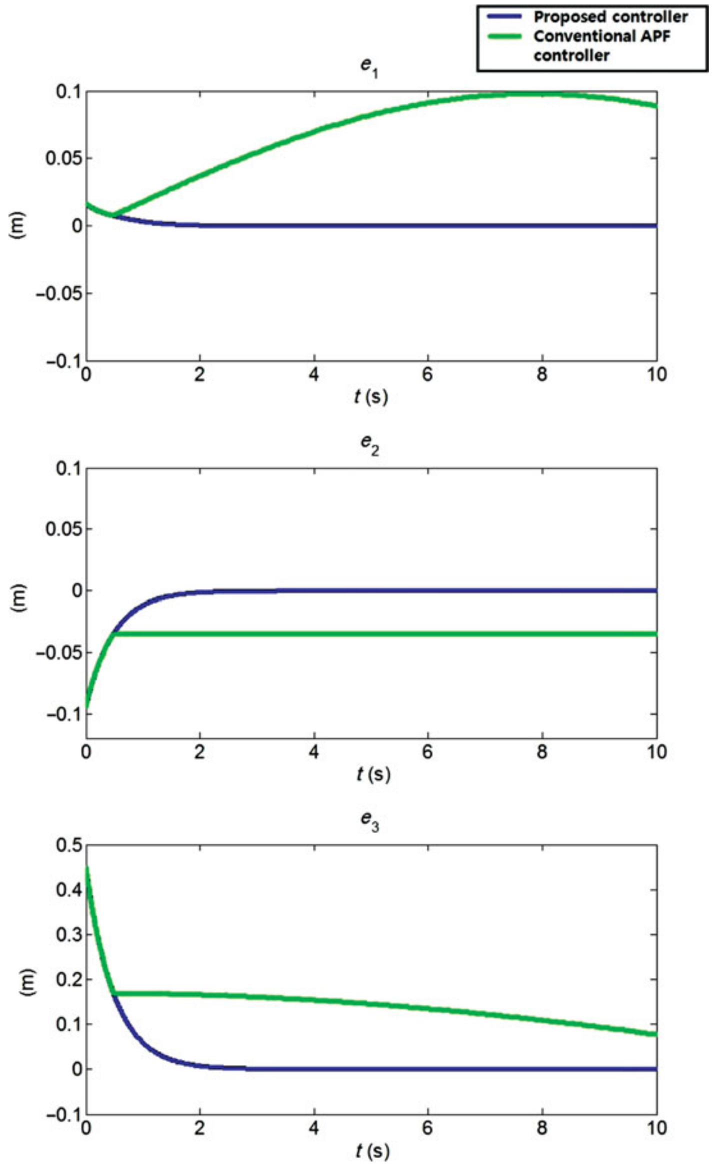


Fig. 5. e_1, e_2, and e_3 of the simulation example when the controllers (26) and (45) are applied for $t \in [0, 10]$.

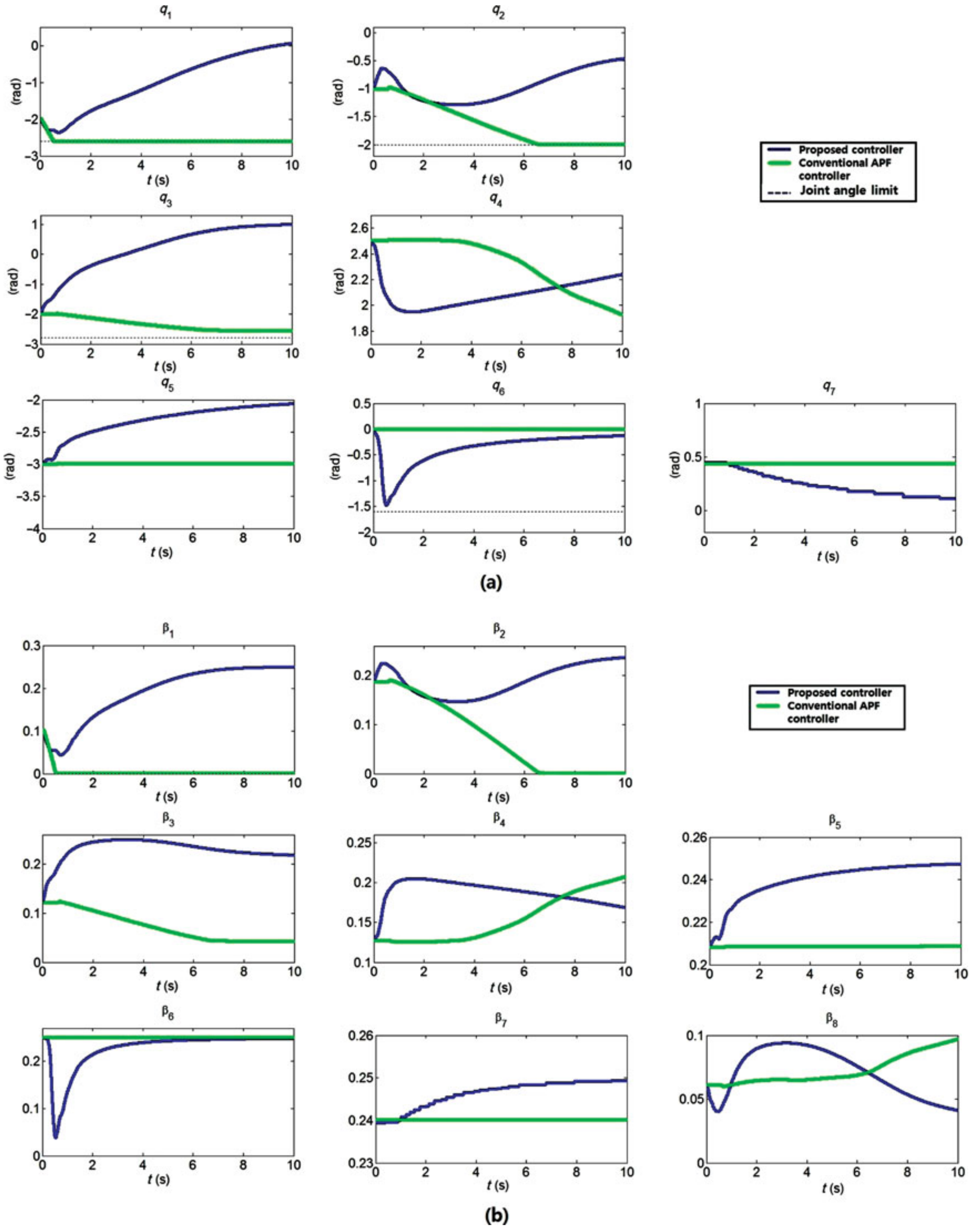
the proposed controller maintains all joint angles within their joint limits. Figure 4(b) shows that the subtask objectives were performed successfully using the proposed controller (all $\beta_i > 0$). Figure 5 shows the tracking errors $e_1 \overset{\triangle}{=} x_{d1} - x_1$, $e_2 \overset{\triangle}{=} x_{d2} - x_2$, and $e_3 \overset{\triangle}{=} x_{d3} - x_3$. The tracking errors of the proposed controller converge to zero whereas the conventional APF controller stops operation at t = 0.4816 s after the operation. The conventional APF controller could not track the desired trajectory.

5.2. *Experimental results*

We performed an experiment for the same tasks described in Section 5.1. The same membership function parameters z_p $(p = 1, \ldots, 6)$ used in the simulation were also used in this experiment. The Barrett WAM control library (based on C++) was used to implement the proposed controller.

When the conventional APF controller (26) was applied to the WAM, it attempted to cross the limit of joint angle q_1 and stopped operation at t = 0.512 s. The WAM has a safety system that stops operations when the control torque current exceeds the fault (3.91 A) levels. On the other hand, the proposed controller made a successful operation for both main and subtasks. Figure 6(a)

https://doi.org/10.1017/S0263574714000344 Published online by Cambridge University Press



(a)

(b)

Fig. 6. (a) Joint angles (q_1, ..., q_7) measured during the experiment; (b) Subtask functions of the example for the experiment where β_i (i = 1 to 7) is the avoidance of the i-th joint limit and β_8 is manipulability.

shows the joint angles q_i (i = 1, . . .,7). In this figure, the proposed controller keeps all joint angles within their joint limits. However, the conventional APF controller caused q_1 to reach its joint angle limit (−2.6 rad). When q_1 of the WAM reached its joint limit, the operation was stopped; however, motion continued only by gravitational force and eventually stopped (Fig. 6(a)). Using the proposed controller, all subtask functions were positive (Fig. 6(b)), which implies that the subtask objectives were successfully performed. Figure 7 shows the tracking errors e_1, e_2, and e_3. The proposed controller

https://doi.org/10.1017/S0263574714000344 Published online by Cambridge University Press

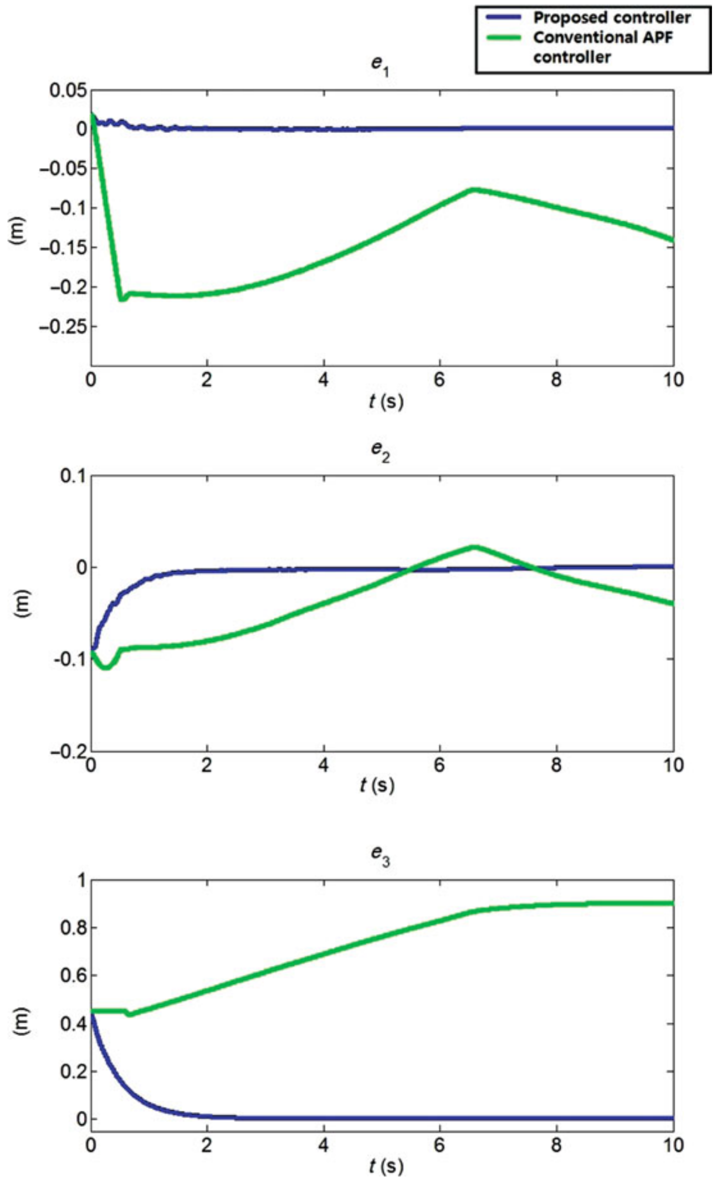


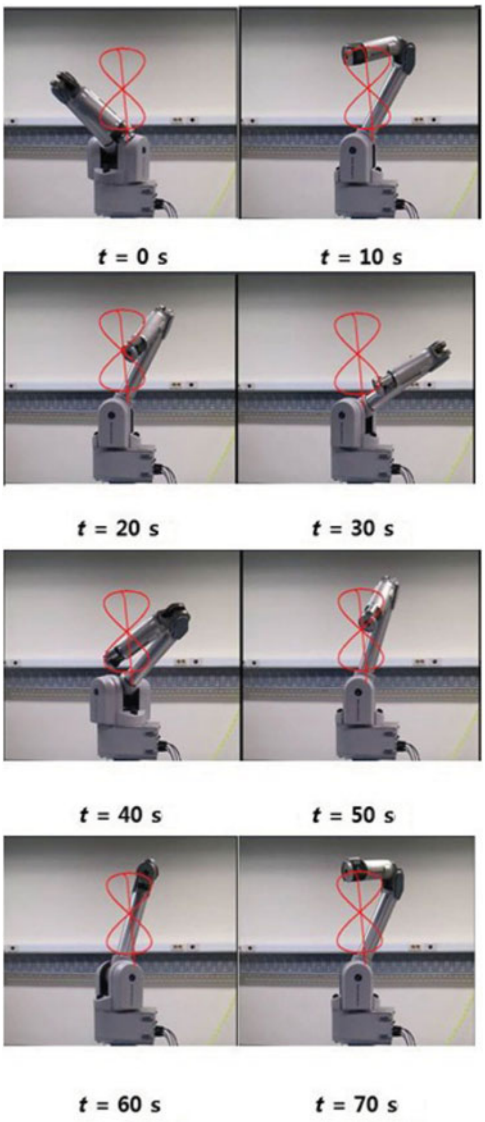
Fig. 7. e_1, e_2, and e_3 of the experiment when controllers (26) and (45) are applied for $t \in [0, 10]$.

made the tracking error converge to zero whereas the conventional APF controller made the robot stop operations. Figure 8 shows the sequence of manipulator motions tracking given the “figure-eight”-shaped trajectory.

6. Conclusion

We proposed the fuzzy weighted subtask controller using APF. We introduced the weighted pseudo-inverse obtained using fuzzy rules related to the null-space velocity tracking error. With this pseudo-inverse, we proposed weighted subtask controllers using APF. A seven-DOF WAM robot was used for the simulation and experiment. Multiple subtasks such as manipulability and avoidance of joint limits were tested. In both the simulation and experiment, the proposed controllers tracked the desired trajectory and satisfied multiple subtasks (positive subtask function values) effectively.

https://doi.org/10.1017/S0263574714000344 Published online by Cambridge University Press



t = 0 s *t* = 10 s

t = 20 s *t* = 30 s

t = 40 s *t* = 50 s

t = 60 s *t* = 70 s

Fig. 8. Manipulator motion sequence in the experiment using the proposed controller.

References

1. W. Risse and M. H. Hiller, "Dextrous Motion Control of a Redundant SCARA Robot," *Proceedings of the Industrial Electronics Conference (IECON 1998)*, Aachen, Germany (Aug. 31–Sep. 4, 1998) pp. 2446–2451.
2. B. Le Boudec, M. Saad and V. Nerguizian, "Modeling and adaptive control of redundant robots," *Math. Comput. Simul.* **71**(4–6), 395–403 (2006).
3. E. Zergeroglu, D. M. Dawson, I. Walker and A. Behal, "Nonlinear tracking control of kinematically redundant robot manipulators," *IEEE/ASME Trans. Mechatronics* **9**(1), 129–132 (2004).
4. U. Ozbay, H. T. Sahin and E. Zergeroglu, "Robust tracking control of kinematically redundant robot manipulators subject to multiple self-motion criteria," *Robotica* **26**(6), 711–728 (2008).
5. K. R. Cleary and D. Tesar, "Incorporating Multiple Criteria in the Operation of Redundant Manipulators of Robotics and Automation," *Proceedings of the IEEE International Conference*, Cincinnati, Ohio (May. 13–18, 1999), pp. 618–624.
6. B. Siciliano, "Kinematic control of redundant robot manipulators: A tutorial," *J. Intell. Robot. Syst.* **3**(3), 201–212 (1990).
7. A. Liegeois, "Automatic supervisory control of the configuration and behavior of multibody mechanisms," *IEEE Trans. Syst. Man Cybern.* **SMC-7**(12), 868–871, (1977).
8. Y. Nakamura, H. Hanafusa and T. Yoshikawa, "Task-priority based redundancy control of robot manipulators," *Int. J. Robot. Res.* **6**(2), 3–15 (1987).

https://doi.org/10.1017/S0263574714000344 Published online by Cambridge University Press

9. S. Chiaverini, "Singularity-robust task-priority redundancy resolution for real-time kinematic control of robot manipulators," *IEEE Trans. Robot. Autom.* **13**(3), 398–410 (1997).
10. P. Chiacchio, S. Chiaverini and L. Sciavicco, "Closed-loop inverse kinematics schemes for constrained redundant manipulators with task space augmentation and task priority strategy," *Int. J. Robot. Res.* **10**(4) , 410–425 (1991).
11. G. Antonelli, "Stability analysis for prioritized closed-loop inverse kinematic algorithms for redundant robotic systems," *IEEE Trans. Robot.* **25**(5), 985–994 (2009).
12. J. Baillieul, "Kinematic Programming Alternatives for Redundant Manipulators," *Proceedings of the IEEE International Conference of Robotics and Automation*, St. Louis, Missouri (Mar. 25–28, 1985) pp. 722–728.
13. O. Egeland, "Task-space tracking with redundant manipulators," *IEEE J. Robot. Autom.* **RA-3**(5), 471–475 (1987).
14. L. Sciavicco and B. Siciliano, "A solution algorithm to the inverse kinematic problem for redundant manipulators," *IEEE J. Robot. Autom.* **4**(4), 403–410 (1988).
15. 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.* **11**(2), 286–292 (1995).
16. W. Shen and J. Gu, "Multi-criteria Kinematics Control for the pa10-7c Robot Arm with Robust Singularities," *Proceedings of the IEEE International Conference of Robotics and Biomimetics*, Yalong Bay, Sanya, China (Dec. 15–18, 2007) pp. 1242–1248.
17. J. Xiang, C. Zhong and W. Wei, "General-weighted least-norm control for redundant manipulators," *IEEE Trans. Robot.* **26**(4), 660–669 (2010).
18. A. A. Maciejewski and C. A. Klein, "Obstacle avoidance for kinematically redundant manipulators in dynamically varying environments," *Int. J. Robot. Res.* **4**, 109–117 (1985).
19. E. Tatlicioglu, M. McIntyre, D. Dawson and I. Walker, "Adaptive Nonlinear Tracking Control of Kinematically Redundant Robot Manipulators with Sub-Task Extensions," *Proceedings of the 44th IEEE Conference on Decision and Control, and the European Control Conference*, CDC-ECC '05 Seville, Spain (Dec. 12–15, 2005) pp. 4373–4378.
20. E. Tatlicioglu, M. McIntyre and D. Dawson, "Adaptive nonlinear tracking control of kinematically redundant robot manipulators with sub-task extensions," Clemson University CRB Technical Report (2005).
21. E. Tatlicioglu, D. Braganza, T. C. Burg and D. M. Dawson, "Adaptive control of redundant robot manipulators with sub-task objectives," *Robotica* **27**(6) , 873–881 (2009).
22. N. Nath, E. Tatlicioglu and D. M. Dawson, "Teleoperation with kinematically redundant robot manipulators with sub-task objectives," *Robotica* **27**(7) , 1027–1038 (2009).
23. N. Nath, E. Tatlicioglu and D. M. Dawson, "Teleoperation with Kinematically Redundant Robot Manipulators with Sub-Task Objectives," *Proceedings of the IEEE Conference on Decision and Control*, Cancun, Maxico (Dec. 9–11, 2008) pp. 4320–4325.
24. Y. J. Yoo, D. S. Jung and S. C. Won "Multi-subtask Controllers of the Redundant Robot Manipulator," *Proceedings of the 37th Annual Conference on IEEE Industrial Electronics Society (IECON 2011)*, Melbourne, Australia (Nov. 7–10, 2011) pp. 227–232.
25. B. W. Choi, J. H. Won and M. J. Chung, "Optimal redundancy resolution of a kinematically redundant manipulator for a cyclic task," *J. Robot. Syst.* **9**(4), 481–503 (1992).
26. K. C. Suh and J. M. Hollerbach, "Local Versus Global Torque Optimization of Redundant Manipulators," *IEEE International Conference on Robotics and Automation*, Raleigh, North Caroliina (Mar. 31–Apr. 3, 1987) pp. 614–624.
27. D. P. Martin, J. Baillieul and J. M. Hollerbach, "Resolution of kinematic redundancy using optimization techniques," *IEEE Trans. Robot. Autom.* **5**(4), 529–533 (1989).
28. Y. J. Jang and S. W. Kim, "An estimation of a billet temperature during reheating furnace operation," *Int. J. Control Autom. Syst.* **5**(1), 43–50 (2007).
29. J. W. Kim, T. G. Kim, Y. S. Park and S. W. Kim, "On-load motor parameter identification using univariate dynamic encoding algorithm for searches," *IEEE Tran. Energy Convers.* **23**(3), 804–813 (2008).
30. J. W. Kim, T. G. Kim, J. Y. Choi and S. W. Kim, "On the global convergence of univariate dynamic encoding algorithm for searches (uDEAS)," *Int. J. Control Autom. Syst.* **6**(4), 571–582 (2008).

https://doi.org/10.1017/S0263574714000344 Published online by Cambridge University Press