Robotica (2012) volume 30, pp. 477–489. © Cambridge University Press 2011
doi:10.1017/S0263574711000841

State adjustment of redundant robot manipulator based on quadratic programming

Kene Li and Yunong Zhang*

School of Information Science and Technology, Sun Yat-sen University, Guangzhou 510006, China
Emails: zhynong@mail.sysu.edu.cn, jallonzyn@sina.com

(Accepted June 21, 2011. First published online: July 25, 2011)

SUMMARY

To achieve desired configuration, a scheme for state adjustment of a redundant robot manipulator with no end-effector task explicitly assigned and referred to as a state-adjustment scheme is proposed in this paper. Owing to the physical limits in an actual robot manipulator, both joint and joint-velocity limits are incorporated into the proposed scheme for practical purposes. In addition, the proposed state-adjustment scheme is formulated as a quadratic program and resolved at the joint-velocity level. A numerical computing algorithm based on the conversion technique of the quadratic program to linear variational inequalities is presented to address the robot state-adjustment scheme. By employing the state-adjustment scheme, the robot manipulator can automatically move to the desired configuration from any initial configuration with the movement kept within its physical limits. Computer simulation and experimental results using a practical six-link planar robot manipulator with variable joint-velocity limits further verify the realizability, effectiveness, accuracy, and flexibility of the proposed state-adjustment scheme.

KEYWORDS: Redundant robot manipulator; State adjustment; Physical limits; Quadratic programming; Numerical computing algorithm.

1. Introduction

A manipulator is said to be redundant when more degrees of freedom (DOF) are available than necessary for a given end-effector path-following task. This definition implies that redundancy can be established simply with respect to some particular tasks.[1] For nonredundant manipulators, the joint motion is uniquely determined by a prescribed end-effector primary task, and thus, there are no redundant freedoms left for executing secondary tasks, such as handling joint physical limits, environmental constraints, and configuration singularities. In contrast, redundant manipulators have wider operational space and meet more functional constraints because an infinite number of feasible joint configurations can be available. Thus, they have been widely applied in areas, such as for cleanup and remediation of nuclear and hazardous materials, and in space or sea exploration.[1–14] Hence, a number of studies have been carried out on the redundant manipulators, and much attention has been paid to motion planning and control of robot manipulators.[1–19]

A redundancy-resolution scheme is a method or algorithm that selects a joint-space solution from an infinite number of possible solutions given the end-effector primary task of following a desired workspace trajectory. This kind of selection is usually used to accomplish secondary tasks for robot manipulators or to generate an optimal solution and achieve some performance criteria. The redundancy-resolution scheme often makes use of optimization techniques, especially quadratic programming (QP). More specifically, such quadratic programs are equivalent to systems of linear variational inequalities (LVIs), which can then be solved by many algorithms, methods, and techniques efficiently, such as numerical methods[20,21] and recurrent neural networks.[4,8,17–19,22]

One of the important issues in controlling actual robot manipulators is the state-adjustment problem, that is, from one state to another. State adjustment appears in many situations. For example, to initialize the execution of a new end-effector task, a robot manipulator has to move from an arbitrary state to the desired starting state of a new task. In addition, in robot motion planning, different end-effector tasks often require different starting joint states.[4–11,16–19] Thus, adjusting the state of the robot manipulator from the final state of the previous task to the desired starting state of the next task is necessary. Furthermore, in repetitive motion planning,[12,23] the task of multiple cycles has to start from the same initial states; nevertheless, due to the existence of nonrepetitive phenomena, the consistency of the initial and final states usually cannot be guaranteed. To complete the repetitive motion task, the states should be adjusted at the end of every cycle. To the best of the authors' knowledge, the QP-based method of state adjustment, which theoretically guarantees that the resultant movement is within the manipulator's physical limits, has not been investigated in present literature. Moreover, the state adjustment of robot manipulator can be done manually by adjusting the joints one by one, but it is evidently much less effective and more time consuming than an automatic state-adjustment scheme. Therefore, the manual scheme is not suitable for highly automated operations and manufacturing, unless it is necessary. In addition, a method of speed-limited trapezoidal trajectories can be designed for the state adjustment, which is discussed in detail in Appendix A. However, such a method is less accurate and less desirable because of its open-loop design property.

* Corresponding author. E-mail: ynzhang@ieee.org

To make the robot manipulator execute tasks more readily and efficiently, a scheme for state adjustment of the redundant robot manipulator with no end-effector task explicitly assigned, called a state-adjustment scheme, is proposed and then unified into a quadratic program. Based on the conversion technique of QP to LVI, a numerical computing algorithm is developed and employed to solve the quadratic program, as well as the original robot scheme. It should be noted that because the numerical computing algorithm adopts the core equations (i.e., Eqs. (4)–(7)) of ref. [20], it is simply termed E47 numerical computing algorithm for presentation convenience. The remainder of this paper is organized into five sections. Section 2 presents the preliminaries and state-adjustment scheme for physically constrained robot manipulators. The scheme is then reformulated into a quadratic program. In Section 3, the LVI-based E47 numerical computing algorithm, as a QP solver, is developed for solving the proposed state-adjustment scheme for robot manipulators. In addition, a pulse-conversion algorithm is presented to convert the resultant joint angles and joint velocities to the motor-driven pulses for controlling the manipulator. Section 4 illustrates the computer simulation results, which are obtained using the proposed state-adjustment scheme, together with the corresponding motion and error analyses. The experimental results based on the same physical robot manipulator are provided in Section 5. Section 6 concludes the paper with final remarks. Before ending this introductory section, it is worth pointing out that this paper has the following new contributions:

- In this paper, a state-adjustment scheme with no end-effector task explicitly assigned is proposed. By employing this scheme, the robot manipulator can be adjusted safely from one state to another with the movement kept within its physical limits, including joint and joint-velocity limits.
- The proposed state-adjustment scheme can be unified into a quadratic program, which can then be solved readily using numerical algorithms or neural networks. This step guarantees the realizability, effectiveness, and accuracy of the proposed state-adjustment scheme.
- The proposed state-adjustment scheme is implemented on a novel planar manipulator, of which the joint-velocity limits are functions of joint angles and thus are time varying. In other words, the proposed scheme is tested on an actual robot with variable joint-velocity limits (VJVLs).
- Zero-initial-velocity boundary constraints based on simple sine-type functions (termed a continuation technique) are designed and incorporated into the unified QP formulation to guarantee that the initial joint velocity is zero, which is more suitable for practical applications.
- An E47 numerical computing algorithm based on LVI conversion is developed to solve the time-varying QP problem constrained by the aforementioned time-varying joint-velocity limits, which is a new and successful attempt. In addition, the E47 algorithm has global linear convergence to optimal QP solutions and is capable of handling general QP problems in an inverse-free manner.
- Computer simulation results based on the practical VJVL robot manipulator are illustrated to demonstrate the accuracy and flexibility of the proposed state-adjustment scheme. More importantly, a corresponding experiment that further demonstrates the realizability and effectiveness of the proposed state-adjustment scheme on the actual redundant robot is conducted.

2. Preliminaries, Scheme Formulation, and Robot Manipulator

References [4–11, 16–19] show that in robot motion planning, different tasks are possibly required to start from different initial joint angles. Thus, adjusting the state of the manipulator from one configuration to another before performing the next task is necessary. In addition to this case, the repetitive motion planning[12] often requires state adjustment because of the existence of the nonrepetitive motion problem. According to previous research on QP-based redundancy-resolution and planning,[17,22] a novel state-adjustment scheme, which minimizes the joint displacement between the current and the desired states, is proposed. The state-adjustment scheme is thus formulated as follows:

$$\text{minimize}\quad \frac{1}{2}((\dot\theta+z)^{\mathrm{T}}(\dot\theta+z))\ \text{with}\ z=\lambda(\theta-\theta_{\mathrm{d}}),\tag{1}$$

$$\text{subject to}\quad \theta^{-}\leq\theta\leq\theta^{+},\tag{2}$$

$$\dot\theta^{-}\leq\dot\theta\leq\dot\theta^{+},\tag{3}$$

where $\theta\in R^n$ denotes the joint-space vector of the manipulator, $\dot\theta\in R^n$ denotes the joint-velocity vector defined as the time derivative of θ, and $\theta_{\mathrm{d}}\in R^n$ denotes the desired state (or the target state). Design parameter $\lambda>0$ is used to scale the convergence rate of the state-adjustment scheme and superscript $^{\mathrm{T}}$ denotes the transpose of a matrix/vector. In addition, $\theta^{\pm}$ and $\dot\theta^{\pm}$ denote the upper and lower limits of the joint-angle and joint-velocity vectors, respectively.

The minimization of the performance index $(\dot\theta+z)^{\mathrm{T}}(\dot\theta+z)/2$ (i.e., Eq. (1)) of the state-adjustment scheme with a global exponential convergence feature can be described in the following theorem:

Theorem 1. Without considering the physical joint limits, starting from any initial state $\theta(0)$, the state vector $\theta(t)$ of the minimization (1) converges to the desired state θ_{d}. Moreover, the exponential convergence can be achieved for the minimization (1) with convergence rate λ. In addition, the joint-velocity vector $\dot\theta(t)$ globally and exponentially converges to zero with convergence rate λ, and the initial joint velocity $\dot\theta(0)$ is $-\lambda[\theta(0)-\theta_{\mathrm{d}}]$.

Proof. By following the neural-dynamic method by Zhang *et al.*,[23,24] the proof is given as the following steps. **Step 1.** The following vector-valued error function $e(t)\in R^n$ is defined:

$$e(t):=\theta(t)-\theta_{\mathrm{d}}.\tag{4}$$

Step 2. To make the error-function (4) converge to zero, the time derivative $\dot e(t)$ of $e(t)$ can be[23,24]

$$\dot e(t):=\frac{\mathrm{d}e(t)}{\mathrm{d}t}=-\lambda e(t).\tag{5}$$

https://doi.org/10.1017/S0263574711000841 Published online by Cambridge University Press

Step 3. As the time derivative of Eq. (4) is $\dot{e}(t) = \dot{\theta}(t)$, substituting it into Eq. (5) yields

$$\dot{\theta}(t) = -\lambda e(t) = -\lambda[\theta(t) - \theta_\mathrm{d}], \tag{6}$$

that is, $\dot{\theta}(t) + \lambda[\theta(t) - \theta_\mathrm{d}] = 0$, which is the theoretical solution of the minimization (1). On the other hand, it follows from Eq. (5) that $e(t) = e(0)\exp(-\lambda t)$, and

$$\theta(t) - \theta_\mathrm{d} = [\theta(0) - \theta_\mathrm{d}]\exp(-\lambda t). \tag{7}$$

Then, the θ_d-difference ratio $\delta(t)$ is defined as follows:

$$\delta(t) := \frac{\|\theta(t) - \theta_\mathrm{d}\|_2}{\|\theta(0) - \theta_\mathrm{d}\|_2} = \exp(-\lambda t), \tag{8}$$

where $\|\cdot\|_2$ denotes the two norm of a vector; that is, $\theta(t)$ converges globally and exponentially to θ_d with convergence rate λ.

Moreover, by differentiating Eq. (7) with respect to time t, the explicit expression about joint velocity $\dot{\theta}$ is as follows:

$$\dot{\theta}(t) = -\lambda[\theta(0) - \theta_\mathrm{d}]\exp(-\lambda t). \tag{9}$$

Based on Eq. (9), joint velocity $\dot{\theta}(t)$ globally and exponentially converges to zero with the convergence rate being λ as well, and the initial joint velocity $\dot{\theta}(0)$ is $-\lambda[\theta(0) - \theta_\mathrm{d}]$. In practice, however, in view of potential requirements on robot motion planning and control (such as joint limits and joint-velocity limits), minimizing $\|\dot{\theta}(t) + \lambda[\theta(t) - \theta_\mathrm{d}]\|_2^2/2$ is better than forcing $\dot{\theta}(t) + \lambda[\theta(t) - \theta_\mathrm{d}] = 0$ directly, and minimizing $\|\dot{\theta}(t) + \lambda[\theta(t) - \theta_\mathrm{d}]\|_2^2/2$ is just Eq. (1). The proof is now complete. □

Hence, in view of Theorem 1, with joint limits $\theta^\pm$ and joint-velocity limits $\dot{\theta}^\pm$ included, the state-adjustment scheme is formulated as Eqs. (1)–(3). Furthermore, using the constraint-combining technique,[9–12] joint physical limits (2) and (3) can be combined and handled via the following bound constraint:

$$\eta^- \le \dot{\theta} \le \eta^+, \tag{10}$$

where the ith elements of $\eta^\pm$ are defined as $\eta_i^+ = \min\{\dot{\theta}_i^+, \kappa(\theta_i^+ - \theta_i)\}$ and $\eta_i^- = \max\{\dot{\theta}_i^-, \kappa(\theta_i^- - \theta_i)\}$, $i = 1, 2, \ldots, n$, with design-parameter $\kappa > 0$ used to scale the feasible region of $\dot{\theta}$ and set as 4 in the computer simulations and experiment performed on the VJVL robot manipulator.

With decision variable vector $x := \dot{\theta} \in R^n$,[10,11] the proposed state-adjustment scheme (1)–(3) is now reformulated as the following quadratic program:

$$\text{minimize} \quad \frac{1}{2}x^\mathrm{T}Wx + z^\mathrm{T}x, \tag{11}$$

$$\text{subject to} \quad \eta^- \le x \le \eta^+, \tag{12}$$

where coefficients $W := I$ and $z = \lambda(\theta - \theta_\mathrm{d})$. According to Theorem 1, the initial joint velocity $\dot{\theta}(0) = -\lambda[\theta(0) - \theta_\mathrm{d}]$ could be very large (even reaching the values of joint-velocity limits sometimes), which is difficult or impossible in practice or causes mechanical damage to the robot. To prevent the occurrence of such a large initial joint velocity, the continuation technique is used by imposing a zero-initial-velocity constraint $x^-(t) \le x \le x^+(t)$ (i.e., $x^-(t) \le \dot{\theta} \le x^+(t)$) with

$$x^-(t) = \sin(\pi t/2\tau)\eta^-, \quad x^+(t) = \sin(\pi t/2\tau)\eta^+,$$

where $\tau > 0$ is the task duration, that is, time $t \in [0, \tau]$. Hence, the above QP problem (11) and (12) is finally modified as the following time-varying quadratic program, with $z(t)$, $x^\pm(t)$, and $x(t)$ all time varying:

$$\text{minimize} \quad \frac{1}{2}x^\mathrm{T}(t)Wx(t) + z^\mathrm{T}(t)x(t), \tag{13}$$

$$\text{subject to} \quad x^-(t) \le x(t) \le x^+(t). \tag{14}$$

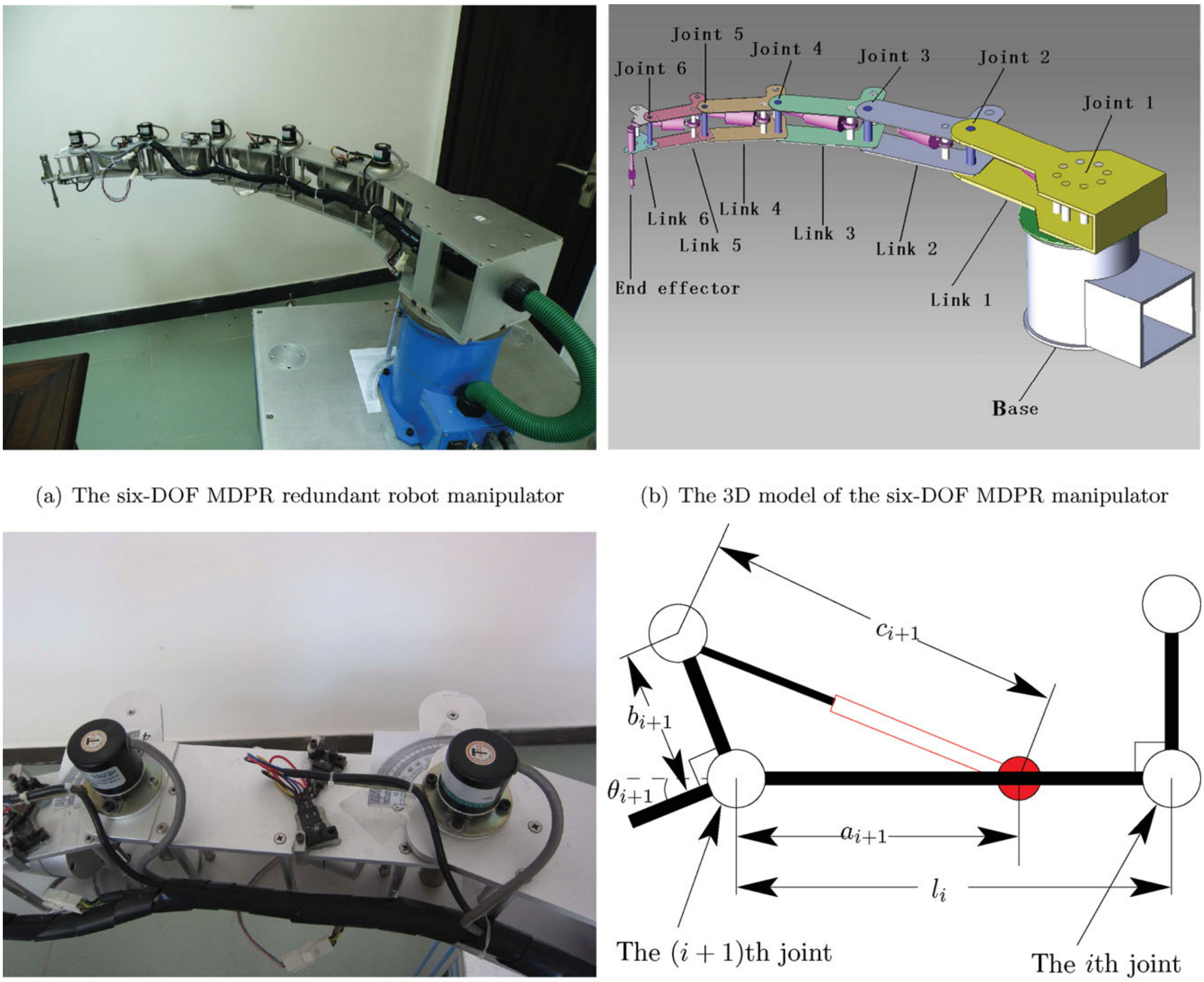
By imposing the zero-initial-velocity constraint depicted in Eq. (14), the zero initial velocity of the motion is guaranteed and is now acceptable in practical applications.

In the ensuing computer simulations and experiment, the proposed state-adjustment scheme is verified on a novel 6-DOF motor-driven push-rod (MDPR) redundant robot manipulator. Figure 1(a) shows the manipulator, which is planar and has six links, with its corresponding three-dimensional (3D) model shown in Fig. 1(b). The first joint is a simple pivot joint. The second to the sixth joints differ from the first joint and are driven by their corresponding push rods. Each joint structure is a triangle, as shown in Fig. 1(c), with three edges of the triangle shown in Fig. 1(d) and denoted by a_{i+1}, b_{i+1}, and c_{i+1} $(i = 1, 2, \ldots, 5)$. The length of the ith link is defined as l_i $(i = 1, 2, \ldots, 6)$ in Fig. 1(d). In addition, the values of the physical parameters of the robot manipulator are given in Table I. To meet the requirement of a wide-range operational space, the first joint is designed to be driven by a high-torque servomotor. The second to the sixth joints are driven by stepping motors for easy-to-control purposes. To reduce the weight of the robot manipulator, the links are made of aluminum alloy. The robot hardware system is composed of the 6-DOF MDPR manipulator and a personal computer (with a Pentium (R) Dual-Core E5300 2.60 GHz CPU, 4 GB DDR3 memory, and a Windows XP Professional operating system), which sends instructions and data to the manipulator motion-control module, that is, a six-axis motion-control card of peripheral component interconnect. The manipulator motion-control module converts the data into actual pulse signals to drive the motors according to the instructions.

Table I. Physical parameters of the 6-DOF MDPR robot manipulator.

i	θ_i^- (rad)	θ_i^+ (rad)	l_i (m)	a_i (m)	b_i (m)
1	0.000	4.587	0.301	–	–
2	0.000	0.816	0.290	0.250	0.080
3	0.035	0.621	0.230	0.250	0.080
4	0.052	0.599	0.225	0.190	0.080
5	0.035	0.599	0.214	0.185	0.080
6	0.000	0.445	0.103	0.174	0.080

https://doi.org/10.1017/S0263574711000841 Published online by Cambridge University Press



(a) The six-DOF MDPR redundant robot manipulator

(b) The 3D model of the six-DOF MDPR manipulator

(c) The $(i+1)$th joint structure, $i = 1, 2, \cdots, 5$

(d) The $(i+1)$th joint structure model

Fig. 1. (Colour online) 6-DOF MDPR redundant robot manipulator and its joint structure. (a) The 6-DOF MDPR redundant robot manipulator. (b) The 3D model of the 6-DOF MDPR manipulator. (c) The $(i + 1)$th joint structure, $i = 1, 2, \ldots, 5$. (d) The $(i + 1)$th joint structure model.

For the 6-DOF MDPR robot manipulator, the VJVLs $\dot{\theta}^{\pm}(\theta)$ are obtained from $\dot{\theta}^{-}(\theta) \leq \dot{\theta} \leq \dot{\theta}^{+}(\theta)$, which change with joint vector θ (i.e., being functions of θ), by analyzing the joint-structure shown in Fig. 1(d) (i.e., following the law of cosines). More specifically, with $i = 1, 2, \ldots, 5$, the formula is as follows:

$$-25\pi/24 \leq \dot{\theta}_1 \leq 25\pi/24, \quad \dot{\theta}^-_{i+1}(\theta_{i+1}) \leq \dot{\theta}_{i+1} \leq \dot{\theta}^+_{i+1}(\theta_{i+1}),$$

where

$$\dot{\theta}^-_{i+1}(\theta_{i+1}) = \frac{v^-_{i+1} s_{i+1}\sqrt{a^2_{i+1} + b^2_{i+1} + 2a_{i+1}b_{i+1}\sin\theta_{i+1}}}{a_{i+1}b_{i+1}\cos\theta_{i+1}}, \tag{15}$$

$$\dot{\theta}^+_{i+1}(\theta_{i+1}) = \frac{v^+_{i+1} s_{i+1}\sqrt{a^2_{i+1} + b^2_{i+1} + 2a_{i+1}b_{i+1}\sin\theta_{i+1}}}{a_{i+1}b_{i+1}\cos\theta_{i+1}}, \tag{16}$$

with v^-_{i+1} and v^+_{i+1} denoting the negative and positive rotation-rate limits of the $(i+1)$th stepping motor, and s_{i+1} denoting the elongation rate of the $(i+1)$th push rod (i.e., the elongation length when the motor moves a full turn). For this manipulator hardware system, $-v^-_{i+1} = v^+_{i+1} = |v_{i+1}|_{\max} = 10$ rot/s and $s_{i+1} = 2.5 \times 10^{-3}$ m/rot. The above VJVL results (i.e., Eqs. (15) and (16)) show that the joint-velocity limits $\dot{\theta}^{\pm}_{i+1}$ are nonlinearly related to the joint angle θ_{i+1}.

3. QP Solver and Control of Robot Manipulator

In this section, a numerical computing algorithm is developed as an efficient QP solver for handling the bound-constraint QP problem (13) and (14) based on the LVI conversion and using the following design method.[3] The developed numerical computing algorithm is also applicable in solving the QP problem (11) and (12). The motors of the robot manipulator are driven by pulse signals. Thus, the sampling gap is set to $h = 0.01$ s for the solution of the QP and the generation of the pulse signals based on the experimental experience of the author and precision requirement.

https://doi.org/10.1017/S0263574711000841 Published online by Cambridge University Press

3.1. E47 numerical computing algorithm

The QP problem (13) and (14) can first be converted into an LVI, that is, $(x - x^*)^{\mathrm{T}}(Wx^* + z) \geq 0$ with x^* being an optimal solution of the QP problem. The derivation of such a conversion is given in Appendix B. In addition, by defining a set $\Omega = \{u|x^- \leq u \leq x^+\} \subset R^n$, the QP problem (13) and (14) is known to be equivalent to the following piecewise-linear equation:

$$\mathcal{P}_\Omega[x - (Wx + z)] - x = 0, \tag{17}$$

where $\mathcal{P}_\Omega[\cdot]$ is a piecewise-linear projection operator with the ith element of $\mathcal{P}_\Omega[u]$ being defined as

$$\begin{cases} x_i^- & \text{if } u_i < x_i^-, \\ u_i & \text{if } x_i^- \leq u_i \leq x_i^+, \quad i \in \{1, 2, \ldots, n\}, \\ x_i^+ & \text{if } u_i > x_i^+. \end{cases}$$

Following ref. [3], a set $\Omega^* \in R^n$ can be defined as the solution set of piecewise-linear projection (17), that is, $\Omega^* = \{x^*|x^* \text{ is a solution of Eq. (17)}\}$. In addition, the related error function of Eq. (17) is defined as

$$E(x) := x - \mathcal{P}_\Omega[x - (Wx + z)]. \tag{18}$$

Following ref. [20], the following vector is further defined as the search direction to determine the zero point of piecewise-linear equation (17):

$$g(x) = W^{\mathrm{T}}E(x) + (Wx + z). \tag{19}$$

For the iteration index $k = 0, 1, 2, \ldots,$ if $x^k \notin \Omega^*$, the recursive formula for the solution of piecewise-linear equation (17) is as follows:

$$x^{k+1} = \mathcal{P}_\Omega[x^k - \rho(x^k)g(x^k)], \tag{20}$$

with

$$\rho(x) = \frac{\|E(x)\|_2^2}{\|(W^{\mathrm{T}} + I)E(x)\|_2^2} \geq \frac{\|E(x)\|_2^2}{\|(W^{\mathrm{T}} + I)\|_2^2\|E(x)\|_2^2} \geq \frac{1}{\|(W^{\mathrm{T}} + I)\|_2^2} = \frac{1}{24}. \tag{21}$$

In summary, Eqs. (18)–(21) constitute the so-called E47 numerical computing algorithm. By following refs. [20, 21], the important lemma on the global convergence of the E47 algorithm, being an effective QP solver for Eqs. (13) and (14), is obtained as follows.

Lemma. By starting with any initial state $x^0 \in R^n$, generated by the E47 numerical computing algorithm depicted in Eqs. (18)–(21), the solution sequence $\{x^k\}$, $k = 1, 2, 3 \ldots$, satisfies

$$\|x^{k+1} - x^*\|_2^2 \leq \|x^k - x^*\|_2^2 - \rho(x^k)\|E(x^k)\|_2^2, \quad \forall x^* \in \Omega^*, \tag{22}$$

with $\rho(x^k) \geq 1/24$ resulting from Eq. (21), guaranteeing the effective solution of piecewise-linear equation (17); that is, the sequence $\{x^k\}$ converges globally and linearly to an optimal solution x^* of the QP optimization problem depicted in Eqs. (13) and (14).

Proof. It can be generalized from refs. [20, 21].

3.2. Control of 6-DOF MDPR robot manipulator

As mentioned in the previous section, the motors of the robot joints are driven by the pulse signals transmitted from the host computer. Hence, the resultant joint variables (i.e., joint angle θ and joint velocity $\dot{\theta}$) should thus be converted into pulses per second (PPS) to control the manipulator. For the first joint driven by a servomotor, the number of PPS is

$$\mathrm{PPS}_1 = \gamma\dot{\theta}_1/(2\pi), \tag{23}$$

where $\gamma = 3.2 \times 10^5$ is the parameter related to the 6-DOF MDPR robot manipulator. For the second through the sixth joint driven by stepping motors, the structure of the $(i + 1)$th joint is shown in Fig. 1(d); following the law of cosines, the rotation rate of the $(i + 1)$th motor is as follows:

$$v_{i+1} = \frac{a_{i+1}b_{i+1}\dot{\theta}_{i+1}\cos\theta_{i+1}}{s_{i+1}\sqrt{a_{i+1}^2 + b_{i+1}^2 + 2a_{i+1}b_{i+1}\sin\theta_{i+1}}},$$

with $i = 1, 2, \ldots, 5$. As the stepping angles of the second joint through the sixth joint are all 0.01π rad and the subdividing multiples are all 32, the number of PPS for the $(i + 1)$th joint $(i = 1, 2, \ldots, 5)$ is

$$\begin{aligned}\mathrm{PPS}_{i+1} &= (2\pi/(0.01\pi/32))v_{i+1} \\ &= \frac{6400a_{i+1}b_{i+1}\dot{\theta}_{i+1}\cos\theta_{i+1}}{s_{i+1}\sqrt{a_{i+1}^2 + b_{i+1}^2 + 2a_{i+1}b_{i+1}\sin\theta_{i+1}}}.\end{aligned} \tag{24}$$

By exploiting Eqs. (23) and (24), the PPS signals are obtained for controlling the robot manipulator.

4. Simulation Studies

In this section, the proposed state-adjustment scheme (1)–(3) (i.e., correspondingly the QP Eqs. (11) and (12) or (13) and (14)) and the E47 numerical computing algorithm (18)–(21) are simulated for the state adjustment of the presented 6-DOF MDPR planar robot manipulator. Specifically, in the simulations, the task of the manipulator is to adjust its configuration from the initial state $\theta(0) = [1.2219, 0.3491, 0.5062, 0.0611, 0.0349, 0.0349]^{\mathrm{T}}$ to the desired state $\theta_{\mathrm{d}} = [0.3491, 0.3491, \pi/12, \pi/12, \pi/12, \pi/12]^{\mathrm{T}}$ in radians (both of which could be set as any values within the joint limits) without loss of generality. In addition, the simulation studies are divided into two subsections. The first one discusses the convergence property of the state-adjustment scheme (11) and (12), which does not impose the zero-initial-velocity constraint. On the other hand, the efficacy, flexibility, and accuracy of the proposed state-adjustment scheme (13) and (14), which impose the zero-initial-velocity constraint for velocity-continuation purposes, are investigated in detail in the second subsection.

https://doi.org/10.1017/S0263574711000841 Published online by Cambridge University Press

Table II. Joint-angle values in the situation of $\lambda = 1$ and $\tau = 30$ s at different time t instants.

t (s)	θ_1 (rad)	θ_2 (rad)	θ_3 (rad)	θ_4 (rad)	θ_5 (rad)	θ_6 (rad)
0.5	0.87712	0.34910	0.40966	0.14037	0.12453	0.12453
1	0.66856	0.34910	0.35126	0.18833	0.17875	0.17875
2	0.46603	0.34910	0.29454	0.23491	0.23140	0.23140
3	0.39190	0.34910	0.27379	0.25196	0.25067	0.25067
6	0.35120	0.34910	0.26240	0.26132	0.26125	0.26125
10	0.34914	0.34910	0.26181	0.26180	0.26180	0.26180
15	0.34910	0.34910	0.26180	0.26180	0.26180	0.26180

4.1. State adjustment without zero-initial-velocity constraint

In this subsection, the state-adjustment scheme (1)–(3) without imposing the zero-initial-velocity constraint, that is, correspondingly the QP Eqs. (11) and (12), is simulated based on the 6-DOF robot manipulator. The simulation results are shown in Figs. 2 and 3, as well as in Table II.

Figure 2 illustrates the transient behaviors of joint-angle and joint-velocity variables of the 6-DOF MDPR robot manipulator in the situation of $\lambda = 1$ and $\tau = 30$ s. The values of the joint angles and joint velocities exponentially converge to their desired values. In addition, Table II shows the joint-angle values at different time instants (i.e., $t = 0.5, 1, 2, 3, 6, 10, 15$ s). With respect to Table II and using Eq. (8), the quantified convergence results are obtained that the θ_{d}-difference ratio $\delta(0.5)$ (i.e., at time instant $t = 0.5$ s) is about 60.50% (which approximates $\exp(-0.5)$), $\delta(1)$ is about 36.60% (which approximates $\exp(-1)$), $\delta(2)$

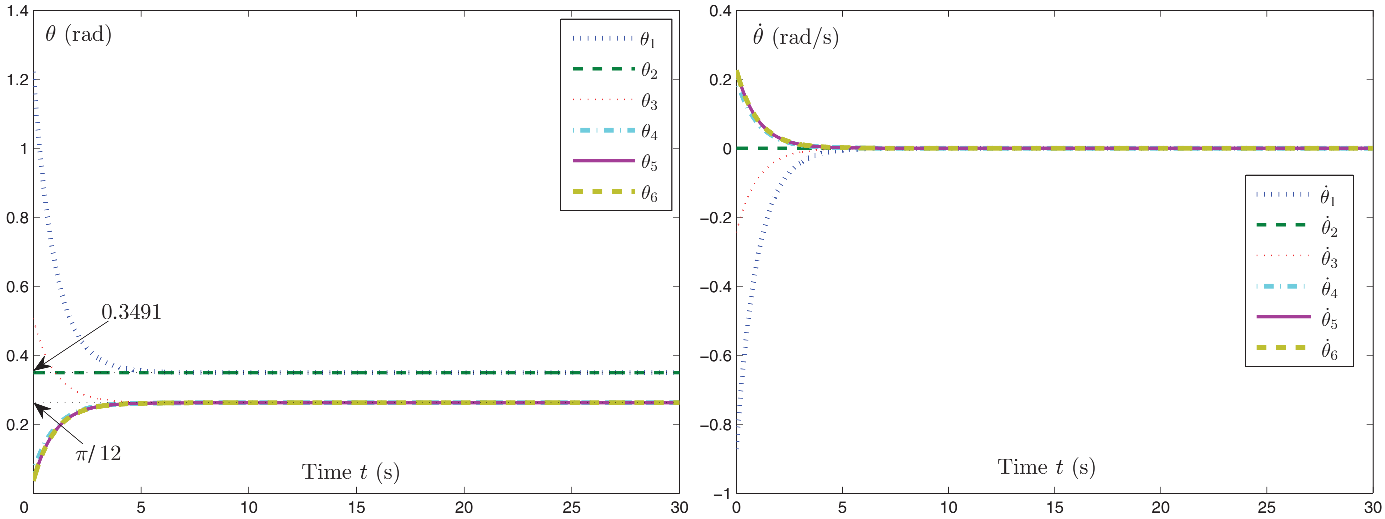


Fig. 2. (Colour online) Profiles of joint variables of the MDPR manipulator synthesized by the state-adjustment scheme (1)–(3) without imposing the zero-initial-velocity constraint and with parameters $\lambda = 1$ and $\tau = 30$ s.

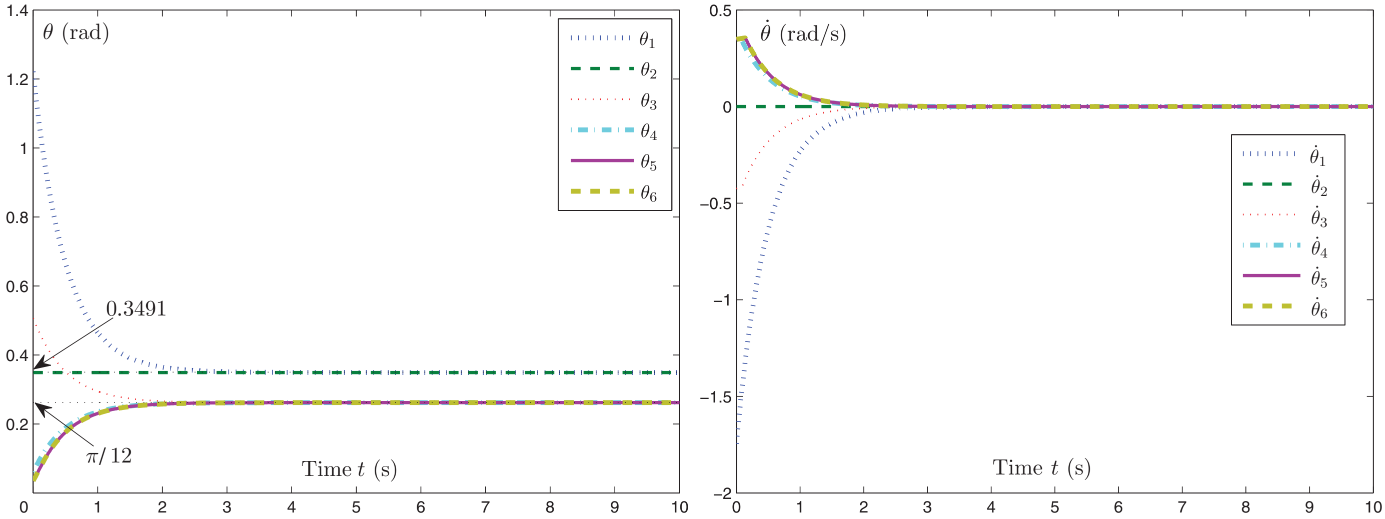


Fig. 3. (Colour online) Profiles of joint variables of the MDPR manipulator synthesized by the state-adjustment scheme (1)–(3) without imposing the zero-initial-velocity constraint and with parameters $\lambda = 2$ and $\tau = 10$ s.

https://doi.org/10.1017/S0263574711000841 Published online by Cambridge University Press

Table III. Joint-angle errors $|e_i(\tau)|$ $(i = 1, 2, \ldots, 6)$ of the state-adjustment task for different λ and τ.

Test #	joint 1	joint 2	joint 3	joint 4	joint 5	joint 6
Situation 1 (10^{-7} rad)	3.32930	3.32929	2.45206	2.45206	2.45206	2.45206
Situation 2 (10^{-5} rad)	7.40662	0.03329	3.84190	2.93454	3.76557	3.72127
Situation 3 (10^{-7} rad)	1.77145	1.66465	1.34676	1.10760	1.05763	1.06175

is about 13.40% (which approximates $\exp(-2)$), $\delta(3)$ is about 4.90% (which approximates $\exp(-3)$), $\delta(6)$ is about 0.24% (which approximates $\exp(-6)$), and $\delta(10)$ is about 0.00%, which implies that $\theta(t)$ actually achieves the desired state θ_{d} from a practical viewpoint. These results confirm the exponential convergence property of Eq. (1) analyzed in Theorem 1. In addition, the initial joint velocity $\dot{\theta}(0) = [-0.8728, 0.0000, -0.2444, 0.2007, 0.2269, 0.2269]^{\mathrm{T}}$ rad/s is the maximal joint velocity during the task, which can be seen from the right subplot of Fig. 2.

For further investigation, a different value of λ (i.e., $\lambda = 2$) is set, and the simulation results are shown in Fig. 3. As the value of λ increases from 1 to 2, the initial velocity $\dot{\theta}(0)$ increases accordingly (compared to the right subplot of Fig. 2), which verifies well $\dot{\theta}(0) = -\lambda(\theta(0) - \theta_{\mathrm{d}})$ (i.e., Eq. (6) with $t = 0$ s). As shown in the right subplot of Fig. 3, some of the initial joint velocities, for example, $\dot{\theta}_4(0)$, $\dot{\theta}_5(0)$, and $\dot{\theta}_6(0)$, meet the manipulator's physical limits, i.e., $\dot{\theta}_4^+(0)$, $\dot{\theta}_5^+(0)$, and $\dot{\theta}_6^+(0)$, respectively. This phenomenon can be observed because $\dot{\theta}_4(0) = 0.4014$ rad/s, $\dot{\theta}_5(0) = 0.4538$ rad/s, and $\dot{\theta}_6(0) = 0.4538$ rad/s as computed by Eq. (6) are larger than the joint-velocity limits $\dot{\theta}_4^+(0) = 0.3470$ rad/s, $\dot{\theta}_5^+(0) = 0.3450$ rad/s and $\dot{\theta}_6^+(0) = 0.3487$ rad/s as computed by Eq. (16), respectively. The above simulations also substantiate the avoidance feature of joint physical limits of the state-adjustment scheme.

4.2. *State adjustment with zero-initial-velocity constraint*

Based on the aforementioned simulation results, the initial joint velocities are nonzero, which may be less desirable for practical applications. As proposed in Section 2, the zero-initial-velocity constraint can be imposed to make the initial joint-velocity zero, which is more acceptable in practice. In this subsection, the state-adjustment scheme (1)–(3) is simulated with the zero-initial-velocity constraint imposed, that is, QP Eqs. (13) and (14), for the state adjustment of the 6-DOF robot manipulator. The computer-simulation results are illustrated in Table III (where the final error of the ith joint $|e_i(\tau)| = |\theta_i(\tau) - \theta_{\mathrm{d}i}|$ $(i = 1, 2, \ldots, 6)$ with $|\cdot|$ denoting the absolute value of a scalar), as well as in Figs. 4–7.

As synthesized by the state-adjustment scheme (1)–(3) with the zero-initial-velocity constraint imposed, Fig. 4 shows the simulated motion trajectories of the 6-DOF MDPR manipulator when it executes a state-adjustment task. The arrow appearing in the figure shows the motion direction. More specifically, the left subplot of Fig. 4 reflects the motion process and change in joint velocity (i.e., from slowness to fastness and then to slowness), whereas the right subplot of Fig. 4 shows the motion trajectories of joints during the task execution. As shown in Fig. 4, the motion velocities are conformable to the zero-initial-velocity constraint (14) and that the joint trajectories are smooth, which are suitable for practical applications. For further investigation, comparison, and illustration, the state-adjustment scheme (1)–(3) is simulated using different situations (i.e., with different values of λ and τ).

Situation 1: $\lambda = 1$ **and** $\tau = 30$ **s**

Figure 5 illustrates the transient behaviors of joint-angle and joint-velocity variables with $\lambda = 1$ and $\tau = 30$ s. As shown in the left subplot of Fig. 5, all joints rapidly converge to their desired state (i.e., θ_{d}). Specifically, $\theta_1(t)$ converges to 0.3491 rad, and $\theta_3(t)$ through $\theta_6(t)$ all converge

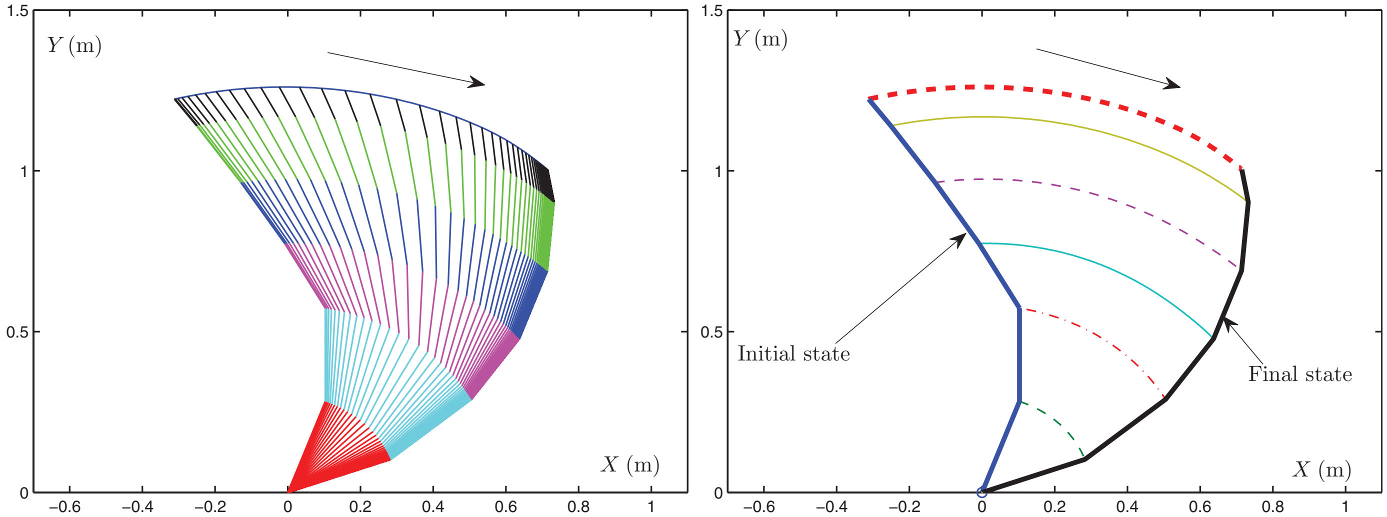


Fig. 4. (Colour online) Motion trajectories of the MDPR manipulator synthesized by the state-adjustment scheme (1)–(3) with zero-initial-velocity constraint imposed (i.e., QP Eqs. (13)–(14)) and with $\lambda = 1$ and $\tau = 10$ s.

https://doi.org/10.1017/S0263574711000841 Published online by Cambridge University Press

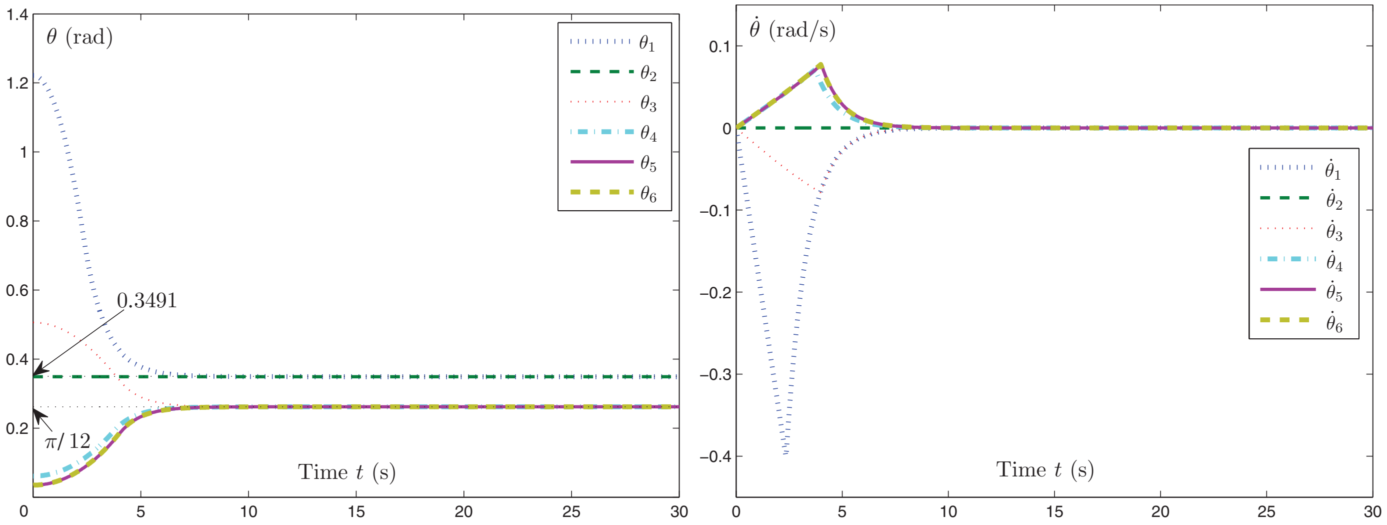


Fig. 5. (Colour online) Profiles of joint variables of the MDPR manipulator synthesized by the state-adjustment scheme (1)–(3) with zero-initial-velocity constraint imposed (i.e., QP Eqs. (13) and (14)) and with $\lambda = 1$ and $\tau = 30$ s.

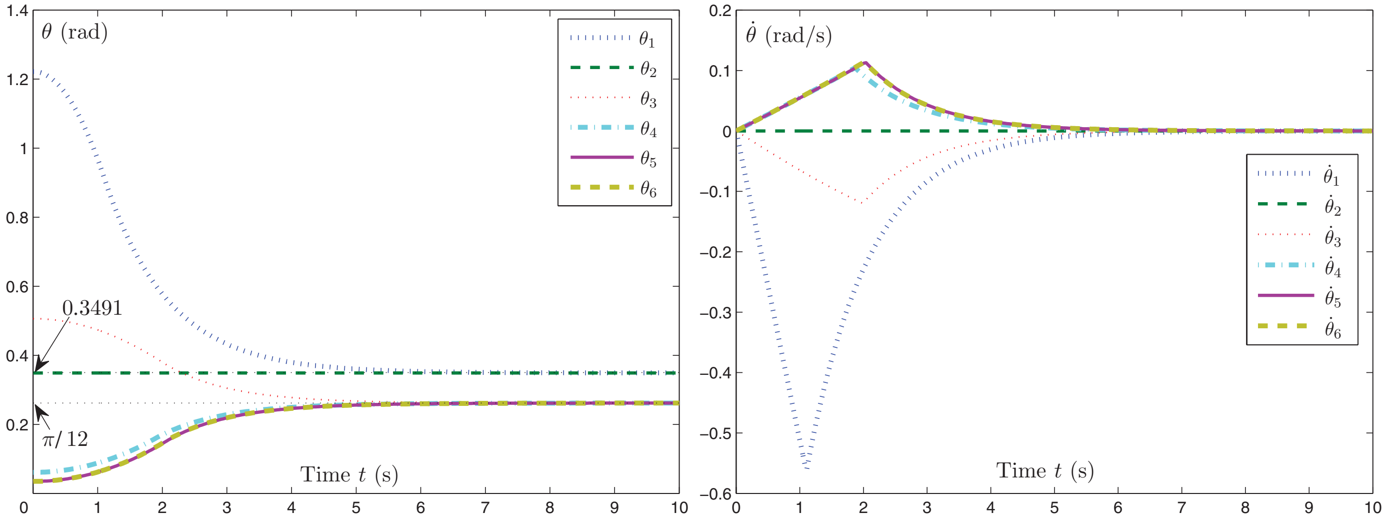


Fig. 6. (Colour online) Profiles of joint variables of the MDPR manipulator synthesized by the state-adjustment scheme (1)–(3) with zero-initial-velocity constraint imposed (i.e., QP Eqs. (13) and (14)) and with $\lambda = 1$ and $\tau = 10$ s.

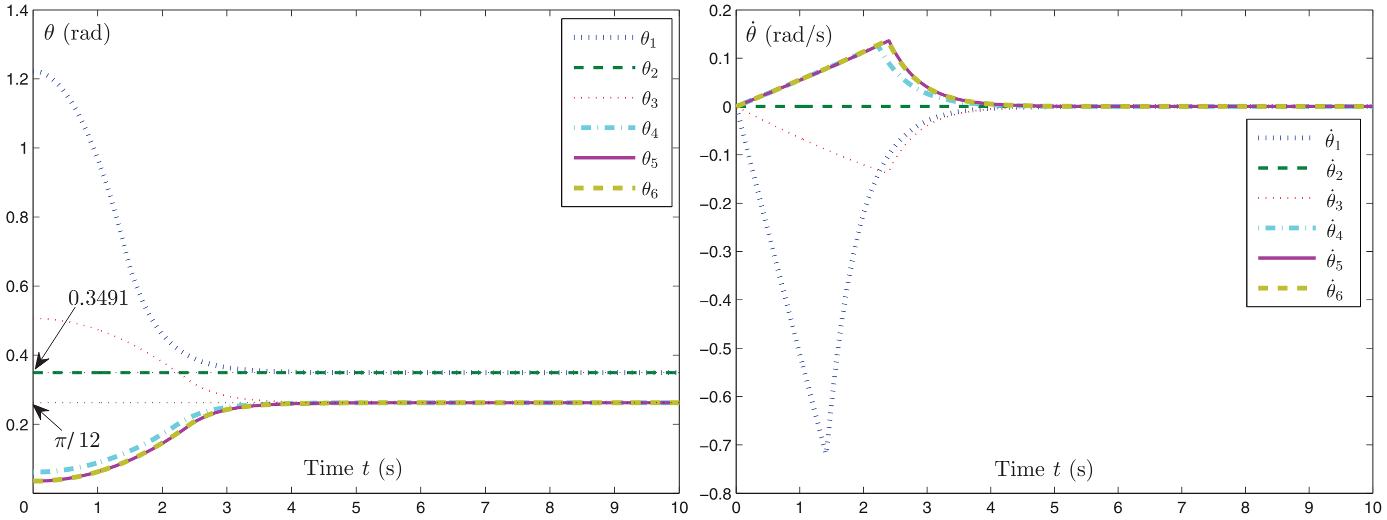


Fig. 7. (Colour online) Profiles of joint variables of the MDPR manipulator synthesized by the state-adjustment scheme (1)–(3) with zero-initial-velocity constraint imposed (i.e., QP Eqs. (13) and (14)) and with $\lambda = 2$ and $\tau = 10$ s.

https://doi.org/10.1017/S0263574711000841 Published online by Cambridge University Press

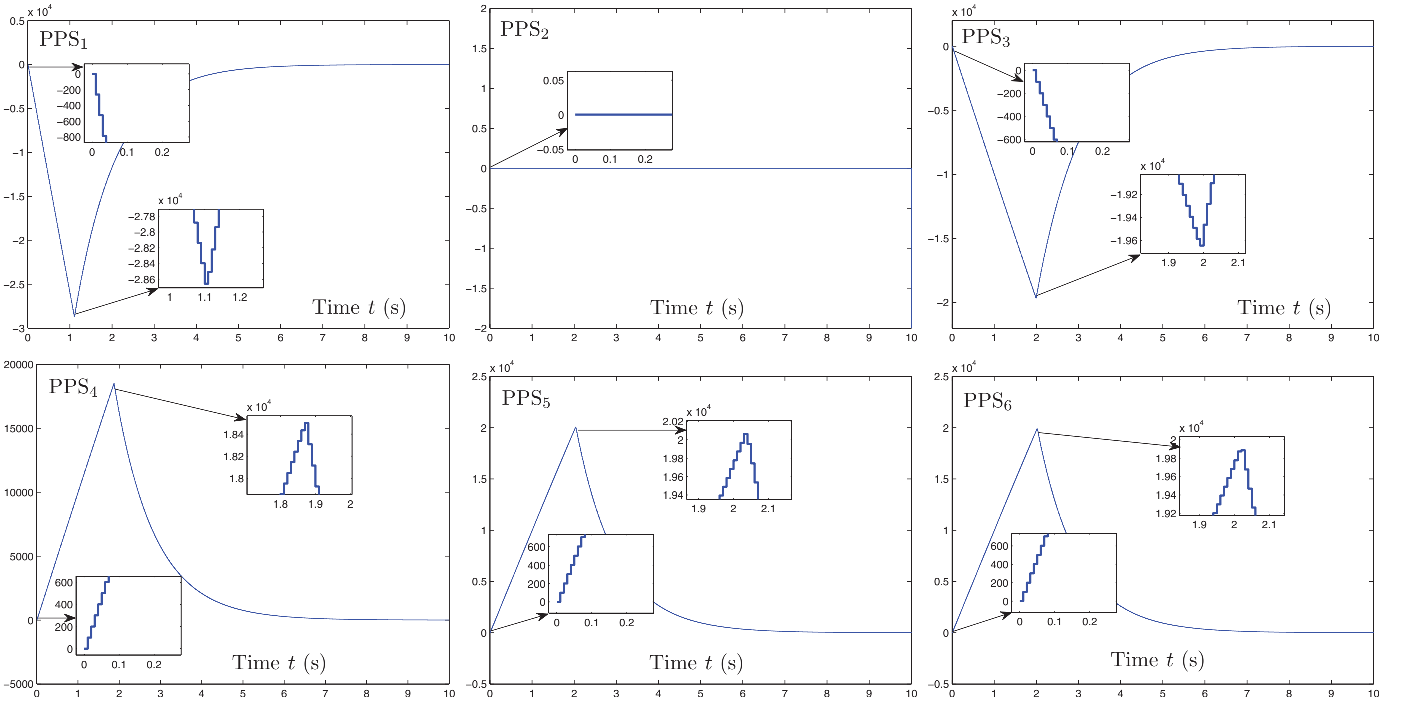


Fig. 8. (Colour online) PPS converted from θ and $\dot{\theta}$ for controlling the robot manipulator.

to $\pi/12$ rad. Meanwhile, $\theta_2(t)$ remains unchanged during the task execution because of $\theta_2(0) = \theta_{d2}$. Correspondingly, as seen from the right subplot of Fig. 5, the joint velocity evidently starts from zero and finally approaches zero within 10 s. As shown in Table III, the maximal joint-angle error in this situation is 3.32930×10^{-7} rad, which is very close to zero, when the execution time t arrives at 30 s. These results demonstrate the efficacy of the proposed state-adjustment scheme QP Eqs. (13) and (14) on the motion control of the robots.

Situation 2: $\lambda = 1$ **and** $\tau = 10$ **s**

As illustrated in the previous situation, the state-adjustment task can be completed in 10 s. Hence, 30 s may be too "time consuming" for the state-adjustment task. Thus, the task duration τ is decreased from 30 to 10 s to investigate the proposed state-adjustment scheme (13) and (14) further. Figure 6 illustrates the profiles of joint variables synthesized by (13) and (14) with $\lambda = 1$ and $\tau = 10$ s. As shown in the left subplot of Fig. 6, the joint-angle variables converge to the desired values within 6.5 s and were shorter than 10 s shown in Fig. 5. The right subplot of Fig. 6 also shows that the joint velocity $\dot{\theta}(t)$ starts from zero, then becomes larger to meet the short time requirement and finally returns to zero, where the maximal joint-velocity magnitude is about 0.58 rad/s. These results substantiate the view that the proposed scheme is flexible for the state-adjustment task, especially under the short time requirement (i.e., $\tau = 10$ s).

Situation 3: $\lambda = 2$ **and** $\tau = 10$ **s**

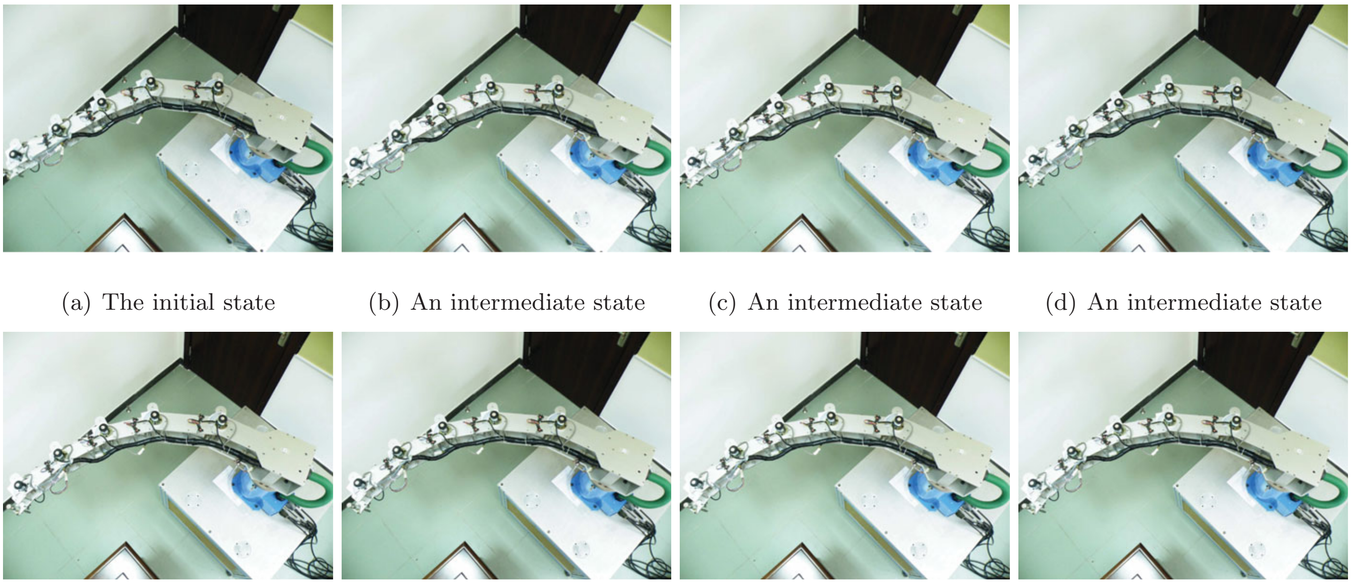
As shown in Table III, the maximal joint-angle error of Situation 2 is 7.40662×10^{-5} rad, which is larger than that of Situation 1. Hence, if the task duration τ is simply decreased to satisfy a shorter execution time requirement, the joint-angle error may increase slightly. To improve the precision while using the short task-execution time, the value of the design parameter λ can be readily increased. Figure 7 shows the profiles of joint variables synthesized by the proposed state-adjustment scheme (13) and (14) with $\lambda = 2$ and $\tau = 10$ s. As shown in the left subplot of Fig. 7, the joint-angle variables converge to the desired values within 5.0 s, which is less than 6.5 s shown in Fig. 6. In addition, as seen from the right subplot of Fig. 7, starting from zero, the joint-velocity variables are larger in magnitude than those in Fig. 6 to meet the short time requirement (i.e., $\dot{\theta}$ converges to zero in shorter time). As shown in Table III, the proposed state-adjustment scheme with $\lambda = 2$ generates a considerably smaller joint-angle error of the state-adjustment task, that is, the maximal joint-angle error is 1.77145×10^{-7} rad.

In summary, the above simulation results, as well as the corresponding analysis, demonstrate the exponential convergence property of the minimization (1) and related joint variables. By imposing the zero-initial-velocity constraint, the initial joint velocity of the motion can be guaranteed to be zero. By choosing λ and τ appropriately, a higher precision and shorter execution time can be achieved. These results show the efficacy, flexibility, and accuracy of the proposed state-adjustment scheme (13) and (14) for the configuration control of redundant robot manipulators.

5. Experiment Results

In this section, the experimental test of the proposed state-adjustment scheme (13) and (14) and E47 numerical computing algorithm (18)–(21) are performed on the actual 6-DOF MDPR robot manipulator shown in Fig. 1. The initial joint-angle vector is set the same as before, that is, $\theta(0) = [1.222, 0.349, 0.506, 0.061, 0.035, 0.035]^{\mathrm{T}}$ rad. To avoid a too large motion scope and to take pictures conveniently, the desired value of θ_1 is set to a more appropriate one that

https://doi.org/10.1017/S0263574711000841 Published online by Cambridge University Press



(a) The initial state (b) An intermediate state (c) An intermediate state (d) An intermediate state

(e) An intermediate state (f) An intermediate state (g) An intermediate state (h) The final state

Fig. 9. (Colour online) Motion transients of the physical 6-DOF MDPR robot manipulator from initial state $\theta(0)$ to desired state θ_{d} synthesized by the state-adjustment scheme (13) and (14) with $\tau = 10$ s. (a) The initial state. (b) An intermediate state. (c) An intermediate state. (d) An intermediate state. (e) An intermediate state. (f) An intermediate state. (g) An intermediate state. (h) The final state.

would allow the robot manipulator to maintain its smaller motion scope of the robot manipulator. Thus, the desired joint-angle vector θ_{d} of the physical robot manipulator is set as $[1.135, 0.349, \pi/12, \pi/12, \pi/12, \pi/12]^{\mathrm{T}}$ in radians. In addition, the task duration $\tau = 10$ s and design parameter $\lambda = 1$.

By exploiting the E47 numerical computing algorithm (18)–(21) to solve the proposed state-adjustment scheme (13) and (14), the joint variables (i.e., θ and $\dot{\theta}$) are obtained. Using Eqs. (23) and (24), the joint variables are converted into PPS for controlling the 6-DOF MDPR robot manipulator, as shown in Fig. 8. With the zero-initial-velocity constraint imposed, the initial PPS of each joint is zero. The plots of PPS are step-like because the E47 numerical computing algorithm (18)–(21) is a discrete-time QP solver and the sampling gap $h = 0.01$ s. The state adjustment of the robot manipulator can be achieved by sending the PPS to drive the motors. Figure 9 shows the motion transients of the physical 6-DOF MDPR robot manipulator during the state-adjustment task execution from the initial state $\theta(0)$ to

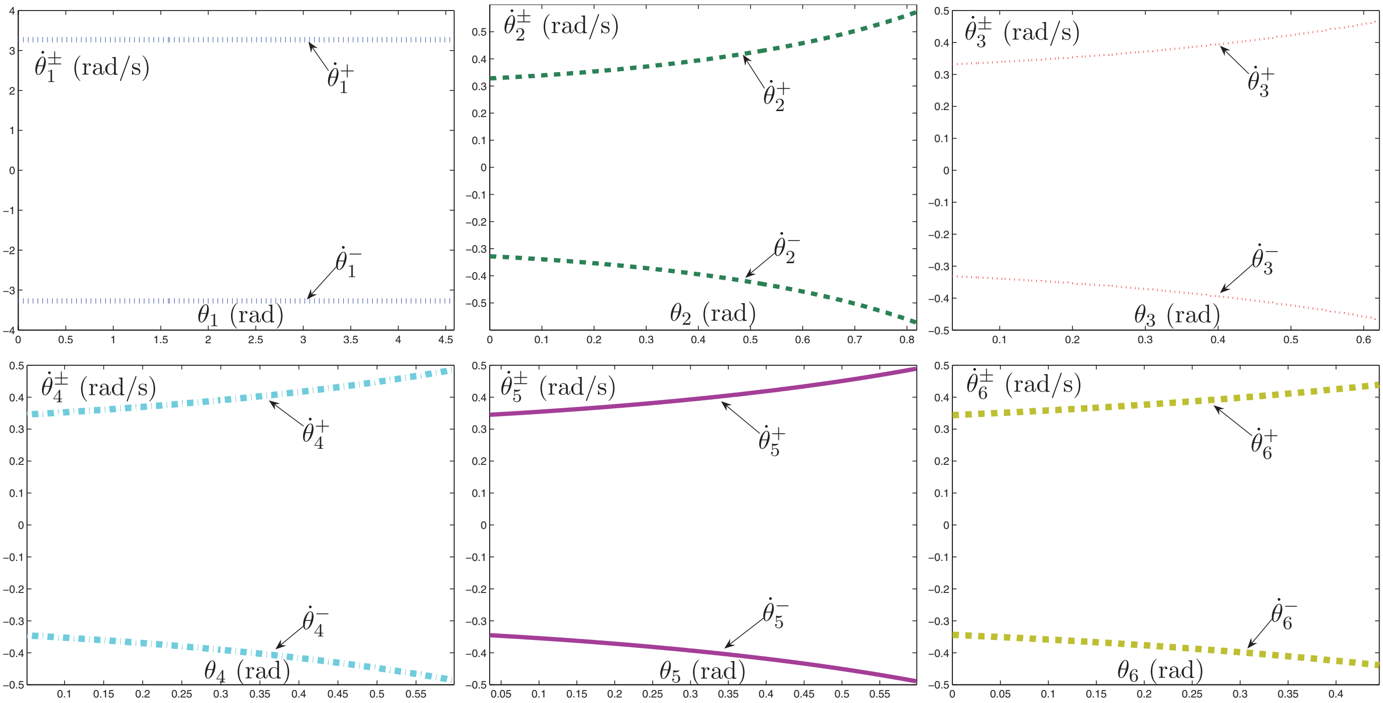


Fig. 10. (Colour online) Relationship between joint-velocity limits $\dot{\theta}^{\pm}$ and joint angle θ.

the final state $\theta(\tau)$ with $\tau = 10\,\mathrm{s}$. As seen from the figure, all joints of the robot manipulator move toward their desired states (i.e., the element values of θ_{d}). Joint 2 of the robot manipulator remains still during the state-adjustment task execution, which fits well with the fact that its desired state $\theta_{\mathrm{d}2}$ equals the initial one $\theta_2(0)$. Moreover, corresponding to Fig. 8, the change in joint velocity is from slowness to fastness and then to slowness (i.e., the dynamic effects of the robot manipulator during the state adjustment), which is similar to the reflection of the left subplot of Fig. 4 (i.e., the sparser the trajectories of the manipulator are, the faster the manipulator moves because all the sampling gaps between every two adjacent trajectories are the same). These results further substantiate the realizability and effectiveness of the proposed state-adjustment scheme (13) and (14). Similar to those in Table III and as shown in the corresponding computer simulation, the final error $[|e_1(\tau)|, \ldots, |e_6(\tau)|]^{\mathrm{T}}$ between $\theta(\tau)$ and θ_{d} is $10^{-5} \times [0.46534, 0.01664, 3.82939, 2.94700, 3.77802, 3.73372]^{\mathrm{T}}$ in radians, and the maximal joint-angle error in this situation (i.e., with $\lambda = 1$, $\tau = 10\,\mathrm{s}$ and the above $\theta(0)$ and θ_{d}) is 3.82939×10^{-5} rad, both of which are small and acceptable in practice.

6. Conclusions

In this paper, a QP-based state-adjustment scheme (13) and (14) with no end-effector task explicitly assigned is proposed and investigated to achieve a desired configuration for robot manipulators. The scheme incorporates the joint physical limits and is formulated as a solvable quadratic program. Being a velocity-continuation technique, the zero-initial-velocity constraint is discussed and employed for the proposed state-adjustment scheme. The state-adjustment scheme and the resultant QP problem (13) and (14) are solved effectively using a so-called E47 numerical computing algorithm (18)–(21) at the joint-velocity level. Both computer simulation and experiment results based on a 6-DOF DMPR robot manipulator demonstrate the realizability, effectiveness, flexibility, and accuracy of the proposed state-adjustment scheme (1)–(3) (or correspondingly, QP Eqs. (13) and (14)).

Acknowledgments

This work is supported by the National Natural Science Foundation of China under Grants 61075121 and 60935001, as well as by the Fundamental Research Funds for the Central Universities of China. In addition, the authors also wish to extend their sincere gratitude to the editors and anonymous reviewers for their constructive comments and suggestions, which have really helped improve the quality of the paper.

Appendix A

From Section 2, VJVLs $\dot{\theta}^{\pm}$ of the 6-DOF MDPR robot manipulator are functions of θ, which is illustrated in Fig. 10 for convenience. According to Eqs. (15) and (16), $\dot{\theta}^{+}$ and $\dot{\theta}^{-}$ are symmetric to each other (illustrated in Fig. 10), and the minimum value of the upper VJVL is $\dot{\theta}^{+}_{\min} = [3.27000, 0.32811, 0.32811, 0.33907, 0.34047, 0.343947]^{\mathrm{T}}$ rad/s. Thus, a method of speed-limited trapezoidal trajectories can be designed as a displacement-related trapezoid-velocity method, with the joint velocity limited by $\pm\dot{\theta}^{+}_{\min}$. Such a design method is based on the displacement (i.e., the angle-change value between the initial and desired joint states). For example, when the task of the robot manipulator is to adjust its configuration from the initial state $\theta(0) = [1.2219, 0.3491, 0.5062, 0.0611, 0.0349, 0.0349]^{\mathrm{T}}$ in radians to the desired state $\theta_{\mathrm{d}} = [0.3491, 0.3491, \pi/12, \pi/12, \pi/12, \pi/12]^{\mathrm{T}}$ in radians, the method of speed-limited trapezoidal trajectories is thus designed based on the value of $(\theta_{\mathrm{d}} - \theta(0))$. For better understanding and further discussion, the state adjustment of the first joint is taken as an example. Evidently, the first joint requires adjusting from 1.2219 rad to 0.3491 rad; that is, the displacement of the first joint is -0.8728 rad for the state adjustment. Then, using $\theta_{\mathrm{d}i}$ to denote the ith element of θ_{d} ($i = 1, 2, \ldots, 6$), the displacement can be divided into three parts for the generation of the trapezoidal velocity: (1) the first part (i.e., $(\theta_{\mathrm{d}1} - \theta_1(0))/6 = -0.1455$ rad) for acceleration, (2) the second part (i.e., $2(\theta_{\mathrm{d}1} - \theta_1(0))/3 = -0.5819$ rad) for constant velocity, and (3) the last part (i.e., $(\theta_{\mathrm{d}1} - \theta_1(0))/6 = -0.1455$ rad) for deceleration. Based on the above analysis, for the ith joint of the manipulator (with $i = 1, 2, \ldots, 6$), the method for the generation of the trapezoidal velocity is as follows:

$$\begin{aligned}\dot{\theta}_i &= \phi(\dot{\theta}^{+}_{\min i})\\ &= \begin{cases} \mathrm{sgn}(\theta_{\mathrm{d}i} - \theta_i(0))4\dot{\theta}^{+}_{\min i} t_i/\tau, & \text{for } 0 \le t_i < t_{i\mathrm{inc}},\\ \mathrm{sgn}(\theta_{\mathrm{d}i} - \theta_i(0))4\dot{\theta}^{+}_{\min i} t_{i\mathrm{inc}}/\tau, & \text{for } t_{i\mathrm{inc}} \le t_i \le 3t_{i\mathrm{inc}},\\ \mathrm{sgn}(\theta_{\mathrm{d}i} - \theta_i(0))4\dot{\theta}^{+}_{\min i}(4t_{i\mathrm{inc}} - t_i)/\tau, & \text{for } 3t_{i\mathrm{inc}} < t_i \le 4t_{i\mathrm{inc}}, \end{cases}\end{aligned} \tag{25}$$

where $\dot{\theta}^{+}_{\min i}$ denotes the ith element of $\dot{\theta}^{+}_{\min}$, $\mathrm{sgn}(\cdot)$ denotes the signum function, and t_i denotes the argument of $\dot{\theta}_i$ corresponding to the state adjustment of the ith joint. $t_{i\mathrm{inc}}$ denotes the time for the acceleration, which can be obtained by solving the integration equation $\int_0^{t_{i\mathrm{inc}}} \dot{\theta}_i \mathrm{d}t_i = (\theta_{\mathrm{d}i} - \theta_i(0))/6$. τ is the task duration (set as $\tau = 10\,\mathrm{s}$ here) and that if $t_{i\mathrm{inc}} > \tau/4$, the task duration τ needs to be increased for the state adjustment.

As synthesized by the displacement-related trapezoid-velocity method, the simulation results of the manipulator for the state adjustment are illustrated in Fig. 11, of which the left subplot shows the joint-velocity profiles and the right subplot shows the corresponding joint-angle profiles. As seen from the figure, the velocity trajectories are trapezoidal and the configuration of the manipulator can be adjusted. However, the final state $\theta(\tau)$ is $[0.36124, \quad 0.34910, 0.26151, 0.25952, 0.26370, 0.25990]^{\mathrm{T}}$ in radians by exploiting the displacement-related trapezoid-velocity method. Thus, the corresponding error between $\theta(\tau)$ and θ_{d} is $[|e_1(\tau)|, \ldots, |e_6(\tau)|]^{\mathrm{T}} = 10^{-2} \times [1.21400, 0.00000, 0.02894, 0.22794, 0.19006, 0.18994]^{\mathrm{T}}$ in radians, which is larger than that of Situation 2 in Section 4.2 (i.e., $10^{-5} \times [7.40662, 0.03329, 3.84190, 2.93454, 3.76557, 3.72127]^{\mathrm{T}}$ in radians). From

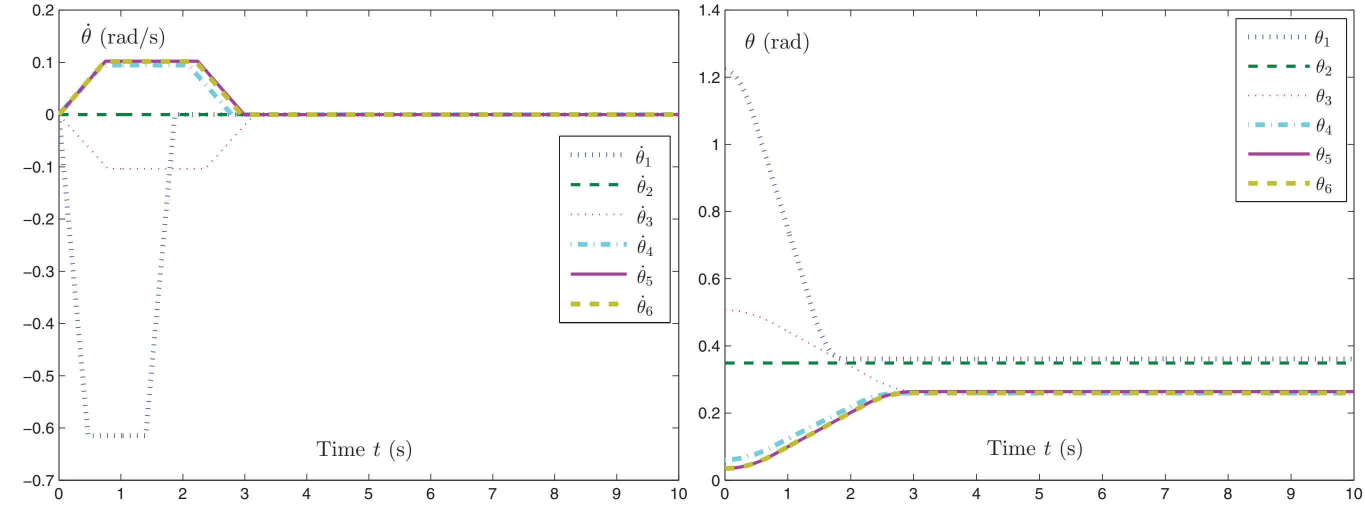


Fig. 11. (Colour online) Profiles of joint variables of the 6-DOF MDPR robot manipulator synthesized by the displacement-related trapezoid-velocity method.

the design process of the displacement-related trapezoid-velocity method, such an undesirable error includes mainly the computational round-off error and the corresponding accumulated error. For example, for the first joint, $t_{1\text{inc}} = 0.471620711823168$ s can be obtained by solving the integration equation $\int_0^{t_{1\text{inc}}} \dot{\theta}_1 \mathrm{d}t_1 = (\theta_{\text{d}1} - \theta_1(0))/6$. As the sampling gap is set as $h = 0.01$ s, $t_{1\text{inc}}$ has to be rounded off and then $t_{1\text{inc}} = 0.47$ s. Thus, $t_{1\text{inc}}$ decreases due to the round-off operation, and $\dot{\theta}_1$ also decreases correspondingly (according to Eq. (25)). In addition, the accumulated error (i.e., the sum of the errors in all sampling gaps) may be large because of the open-loop design property of the displacement-related trapezoid-velocity method. Therefore, after the state adjustment, the error between $\theta(\tau)$ and θ_d is less desirable and less acceptable for practical applications. For the speed-limited trapezoidal trajectory method, the maximum joint velocity is the constant velocity, which is limited by $\pm\dot{\theta}^+_{\text{min}}$. Hence, such a design method (i.e., Eq. (25)) can not make full use of the VJVL and decreases the feasible region of $\dot{\theta}$ in essence.

Remarks. The proposed scheme in this paper differs from the speed-limited trapezoidal trajectory method and takes $(\theta - \theta_\text{d})$ as the criterion to be minimized. Thus, the proposed scheme has θ-feedback property and can eliminate the accumulated round-off error effectively. In addition, the proposed scheme is subject to the bound constraint and makes full use of the VJVL. Therefore, this paper employs the proposed optimization scheme for the state adjustment of the robot manipulator rather than the method of speed-limited trapezoidal trajectories (e.g., Eq. (25)).

Appendix B

In this appendix, the bound-constraint QP problem (13) and (14) is converted into an LVI problem via the following design method.[3] To find a primal equilibrium vector $x^* \in \Omega := \{u | x^- \leq u \leq x^+\} \subset R^n$ such that $\forall x \in \Omega$,

$$(x - x^*)^\text{T}(Wx^* + z) \geq 0. \tag{26}$$

Based on ref. [3], the Lagrangian dual problem of QP Eqs. (13) and (14) can be derived as

$$\text{maximize} \quad -\frac{1}{2}x^\text{T}Wx + x^{-\text{T}}\nu^- - x^{+\text{T}}\nu^+, \tag{27}$$

$$\text{subject to} \quad Wx + z - \nu^- + \nu^+ = 0, \tag{28}$$

$$\nu^- \geq 0, \;\; \nu^+ \geq 0, \tag{29}$$

where ν^- and ν^+ are dual decision variable vectors defined for the left and right parts of bound constraint (14), respectively. Then, a necessary and sufficient condition for the optimum point $(x^*, \nu^{-*}, \nu^{+*})$ of the primal QP problem (13) and (14) and its dual QP problem (27)–(29) is

primal feasibility:

$$x^- \leq x^* \leq x^+;$$

dual feasibility:

$$Wx^* + z - \nu^{-*} + \nu^{+*} = 0, \tag{30}$$

$$\nu^{-*} \geq 0, \nu^{+*} \geq 0;$$

complementarity:

$$\nu^{-*\text{T}}(-x^* + x^-) = 0, \tag{31}$$

$$\nu^{+*\text{T}}(-x^+ + x^*) = 0. \tag{32}$$

To simplify the above necessary and sufficient condition formulation, the dual decision variable vectors ν^{-*} and ν^{+*} in Eqs. (30)–(32) are further studied, which correspond to the original bound constraint (14). Based on Eqs. (31) and (32),

$$\begin{cases} x_i^* = x_i^+ & \text{if } \nu_i^{+*} > 0, \nu_i^{-*} = 0, \\ x_i^- < x_i^* < x_i^+ & \text{if } \nu_i^{+*} = 0, \nu_i^{-*} = 0, \\ x_i^* = x_i^- & \text{if } \nu_i^{+*} = 0, \nu_i^{-*} > 0. \end{cases}$$

By defining $\nu^* = \nu^{-*} - \nu^{+*}$, the ith component of vector $Wx^* + z$ based on dual feasibility condition (30) and the

https://doi.org/10.1017/S0263574711000841 Published online by Cambridge University Press

above analysis is

$$[Wx^* + z]_i = v_i^* \begin{cases} < 0, & x_i^* = x_i^+, \\ = 0, & x_i^* \in (x_i^-, x_i^+), \\ > 0, & x_i^* = x_i^-. \end{cases} \forall i \in \{1, 2, \ldots, n\}, \tag{33}$$

In addition, $\forall x \in \Omega$, the following results are obtained:

$$(x_i - x_i^*) \begin{cases} \leq 0, & x_i^* = x_i^+, \\ \in (x_i^- - x_i^+, x_i^+ - x_i^-), & x_i^* \in (x_i^-, x_i^+), \\ \geq 0, & x_i^* = x_i^-. \end{cases} \forall i \in \{1, 2, \ldots, n\}, \tag{34}$$

By combining Eqs. (33) and (34), the following LVI is obtained; that is, to find an $x^* \in \Omega$ such that

$$(x - x^*)^{\mathrm{T}}(Wx^* + z) \geq 0, \ \forall x \in \Omega. \tag{35}$$

The above resultant LVI (35) is exactly (26), being the converted result of QP Eqs. (13) and (14). The proof is now complete.

References

1. L. Sciavicco and B. Siciliano, *Modelling and Control of Robot Manipulators* (Springer-Verlag, London, UK, 2000).
2. A. Meghdari, D. Naderi and S. Eslami, “Optimal stability of a redundant mobile manipulator via genetic algorithm,” *Robotica* **24**, 739–743 (2006).
3. Y. Zhang, “On the LVI-based Primal-Dual Neural Network for Solving Online Linear and Quadratic Programming Problems,” *Proceedings of the American Control Conference*, Portland, OR, USA (2005) pp. 1351–1356.
4. W. S. Tang and J. Wang, “A recurrent neural network for minimum infinity-norm kinematic control of redundant manipulators with an improved problem formulation and reduced architecture complexity,” *IEEE Trans. Syst. Man Cybern. B* **31**(1), 98–105 (2001).
5. J. P. Puga and L. E. Chiang, “Optimal trajectory planning for a redundant mobile manipulator with non-holonomic constraints performing push-pull tasks,” *Robotica* **26**, 385–394 (2008).
6. U. Ozbay, H. T. Sahin and E. Zergeroglu, “Robust tracking control of kinematically redundant robot manipulators subject to multiple self-motion criteria,” *Robotica* **26** 711–728 (2008).
7. B. Allotta, V. Colla and G. Bioli, “Kinematic control of robots with joint constraints,” *ASME J. Dyn. Syst. Meas. Control* **121**(3), 433–442 (1999).
8. Z. Mao and T. C. Hsia, “Obstacle avoidance inverse kinematics solution of redundant robots by neural networks,” *Robotica* **15**, 3–10 (1997).
9. Y. Zhang and J. Wang, “A dual neural network for constrained joint torque optimization of kinematically redundant manipulators,” *IEEE Trans. Syst. Man Cybern. B* **32**(5), 654–662 (2002).
10. Y. Zhang, J. Wang and Y. Xu, “A dual neural network for bi-criteria kinematic control of redundant manipulators,” *IEEE Trans. Robot. Autom.* **18**(6), 923–931 (2002).
11. Y. Zhang, “A set of nonlinear equations and inequalities arising in robotics and its online solution via a primal neural network,” *Neurocomputing* **70**, 513–524 (2006).
12. Y. Zhang, Z. Tan, Z. Yang and X. Lv, “A Dual Neural Network Applied to Drift-free Resolution of Five-Link Planar Robot Arm,” *Proceedings of the 2008 IEEE International Conference on Information and Automation*, Zhangjiajie, China (2008) pp. 1274–1279.
13. P. S. Donelan, “Singularity-theoretic methods in robot kinematics,” *Robotica* **25**, 641–659 (2007).
14. V. Padois, J.-Y. Fourquet and P. Chiron, “Kinematic and dynamic model-based control of wheeled mobile manipulators: A unified framework for reactive approaches,” *Robotica* **25**, 157–173 (2001).
15. J. Lee, “A structured algorithm for minimum l_∞-norm solutions and its application to a robot velocity workspace analysis,” *Robotica* **19**, 343–352 (2001).
16. C. Qiu, Q. Cao and S. Miao, “An on-line task modification method for singularity avoidance of robot manipulators,” *Robotica* **27**, 539–546 (2009).
17. J. Wang, Q. Hu and D. Jiang, “A Lagrangian network for kinematic control of redundant manipulators,” *IEEE Trans. Neural Netw.* **10**(5), 1123–1132 (1999).
18. H. Ding and S. K. Tso, “A fully neural-network-based planning scheme for torque minimization of redundant manipulators,” *IEEE Trans. Ind. Electron.* **46**(1), 199–206 (1999).
19. H. Ding and J. Wang, “Recurrent neural networks for minimum infinity-norm kinematic control of redundant manipulators,” *IEEE Trans. Syst. Man Cybern. A* **29**(3), 269–276 (1999).
20. B. He, “Solving a class of linear projection equation,” *Numer. Math.* **168**, 71–80 (1994).
21. B. He, “A new method for a class of linear variational inequalities,” *Math. Program.* **166**, 137–144 (1994).
22. J. Wang, “Recurrent neural networks for computing pseudoinverses of rank-deficient matrices,” *SIAM J. Sci. Comput.* **18**(5), 1479–1493 (1997).
23. Y. Zhang, K. Chen and W. Ma, “MATLAB Simulation and Comparison of Zhang Neural Network and Gradient Neural Network for Online Solution of Linear Time-Varying Equations,” *Proceedings of the 2007 International Conference on Life System Modeling and Simulation*, Shanghai, China (2007) pp. 450–454.
24. Y. Zhang, G. Ruan, K. Li and Y. Yang, “Robustness analysis of the Zhang neural network for online time-varying quadratic optimization,” *J. Phys. A-Math. Theor.* **43**, 1–19 (2010) (doi:10.1088/1751-8113/43/24/245202).

https://doi.org/10.1017/S0263574711000841 Published online by Cambridge University Press