

Chapter 14

Application to Fixed-Base Robot RMP

Abstract In this chapter, the ZD approach presented in the previous chapters is applied to repetitive motion planning (RMP) of fixed-base redundant robot manipulators at the joint-acceleration level. Specifically, by introducing two different ZFs and by exploiting the ZD design formula, an acceleration-level RMP performance index is proposed, developed, and investigated. The resultant RMP scheme, which incorporates joint-angle, joint-velocity and joint-acceleration limits, is further presented and investigated to remedy the joint-angle drift phenomenon of fixed-base redundant robot manipulators. Such a scheme is then reformulated as a quadratic program, which is solved by a primal–dual neural network. With three path-tracking examples, simulation results based on PUMA560 robot manipulator substantiate well the effectiveness and accuracy of the proposed acceleration-level RMP scheme, as well as show the application prospect of the presented ZD approach.

14.1 Introduction

In recent years, robotic researchers have focused on solving a variety of tasks requiring sophisticated motion in complex environment via various advanced robots based on different planning and/or control methods [1–7]. Redundant robot manipulators are robots having more degrees of freedom (DOF) than required to perform a given end-effector primary task [6, 8, 9]. It has been argued that redundancy can improve the performance and versatility of a robot manipulator in various aspects such as obstacle avoidance [5, 10, 11], joint limits avoidance [6, 12, 13], and repetitive motion planning (RMP) [6, 9, 14]. Therefore, a multipurpose robot manipulator needs to be redundant if it is to be implemented effectively; e.g., a six-DOF PUMA560 robot manipulator has 3 redundant DOF when we consider only the end-effector’s positioning, and it can thus perform various subtasks in addition to the end-effector’s primary path-tracking task [6].

One fundamental issue on operating such a robot system is the redundancy-resolution problem [6]. The conventional solution to such a redundancy-resolution problem is the pseudoinverse-based method [15, 16]. The researches in recent years [5, 6, 8–10, 13] show that the redundancy-resolution problem can be solved in a more

favorable manner via optimization techniques based on quadratic programming. Generally speaking, such optimization techniques are usually unified and expressed as a quadratic program (QP) which can incorporate equality, inequality, and bound constraints [5, 10]. The resultant QP problem can be solved by many methods and techniques, such as dual neural network (DNN) [6, 8, 17] and primal–dual neural network (PDNN) [6, 9, 18]. Note that, in [19], comparisons between the DNN and the PDNN are demonstrated, which validates the latter has more advantages (e.g., PDNN is matrix-inversion free).

A redundancy-resolution scheme is called repetitive, if it maps closed paths in the task space (i.e., cyclic sequences of tasks) to closed trajectories in the configuration space (i.e., cyclic sequences of configurations) [6, 9, 14, 17, 18, 20, 21]. By contrast, the non-repetitive problem is that the joint angles may not return to their initial values when the end-effector traces a closed path in its workspace [6, 9, 14, 17, 18]. Note that the non-repetitive problem results in a joint angle drift phenomenon and may induce a problem that the manipulator's behavior is hard to predict [6, 9, 17, 18]; and it is then less efficient to readjust the manipulator's configuration after every cycle via self-motion [6, 9, 17, 18].

The previous researches on solving the non-repetitive problem are mainly at the joint-velocity level [6, 14, 17, 18, 20, 21]. However, these may not be applicable to the manipulators which are controlled at the acceleration and/or torque levels. In addition, the joint-acceleration physical limits of the manipulators cannot be incorporated in the scheme resolved at the joint-velocity level [6, 17, 18]. Thus, the RMP performance index at the joint-acceleration level is a very appealing and interesting topic in robotics research domain. Moreover, the acceleration-level scheme can effectively prevent the instability/divergence problem of joint accelerations and torques caused by some velocity-level scheme in long-range motion [9, 22].

In this chapter, based on the ZD approach presented in the previous chapters, we propose and investigate a novel RMP scheme at the joint-acceleration level to remedy the joint-angle drift phenomenon. Specifically, by introducing two different ZFs and by exploiting the ZD design formula [23], an acceleration-level RMP performance index is proposed, developed, and investigated. To the best of the authors' knowledge, such a new RMP performance index at the joint-acceleration level has never been investigated before by others. In addition, the proposed acceleration-level RMP scheme, which incorporates joint-angle, joint-velocity and joint-acceleration limits, is reformulated as a QP, and is then solved by a PDNN [6, 9, 18]. Moreover, simulation results based on PUMA560 robot manipulator performing different types of end-effector path-tracking tasks substantiate well the effectiveness and accuracy of the proposed acceleration-level RMP scheme, as well as show the application prospect of the presented ZD approach. Besides, it is worth pointing out here that our previous book [6] only presents and investigates RMP of redundant robot manipulators at the joint-velocity level. By contrast, in this book (or specifically, in this chapter), we focus on the investigation of RMP at the joint-acceleration level. Evidently, this is a great contribution and improvement, as it promotes the RMP research from joint-velocity level to joint-acceleration level.

14.2 RMP Performance Index Derived via Different ZFs

In this section, the ZD approach presented in the previous chapters is applied to deriving the RMP performance index at the joint-acceleration level.

To lay a basis for further discussion, some well-known essential equations for redundant robot manipulators can be given below directly [9, 15, 16]:

$$f(\theta) = \mathbf{r}, \quad (14.1)$$

$$J(\theta)\dot{\theta} = \dot{\mathbf{r}}, \quad (14.2)$$

$$J(\theta)\ddot{\theta} = \ddot{\mathbf{r}} - \dot{J}(\theta)\dot{\theta}. \quad (14.3)$$

For the above equations, (14.1) describes the relationship between the end-effector position-and-orientation vector $\mathbf{r} \in \mathbb{R}^m$ and joint-angle vector $\theta \in \mathbb{R}^n$, where $f(\cdot)$ is a differentiable nonlinear function with a structure and parameters which are known for a given robot manipulator. $\dot{\mathbf{r}}$ and $\dot{\theta}$ in (14.2) [by differentiating (14.1) with respect to time t] denote respectively the end-effector velocity vector and the joint-velocity vector, and $J(\theta) \in \mathbb{R}^{m \times n}$ is the Jacobian matrix defined as $J(\theta) = \partial f(\theta)/\partial \theta$. In (14.3) [by differentiating (14.2)], $\ddot{\mathbf{r}}$ and $\ddot{\theta}$ denote, respectively, the end-effector acceleration vector and the joint-acceleration vector, and $\dot{J}(\theta)$ is the time derivative of Jacobian matrix $J(\theta)$. For redundant robot manipulators (i.e., $m < n$), (14.1)–(14.3) are all underdetermined and generally admit an infinite number of solutions in terms of inverse kinematics. Besides, it is worth mentioning here that the RMP schemes investigated in [6, 17, 18] are based on (14.2) (i.e., at the joint-velocity level), whereas the RMP scheme presented in this chapter is based on (14.3) (i.e., at the joint-acceleration level).

Now we present the specific design procedure to obtain the acceleration-level RMP performance index as follows:

- Firstly, to achieve RMP of both end-effector and joints, we require that $r(T_d) = r(0)$ and $\theta(T_d) = \theta(0)$, where T_d denotes the task duration of both path-tracking and RMP, and $\theta(0)$ denotes the initial value of $\theta(t)$ [i.e., $\theta(0) = \theta(t=0)$]. Thus, by following the ZD approach, it is natural to define the following vector-valued joint-displacement function (i.e., the first ZF):

$$\mathbf{e}(t) = \theta(t) - \theta(0) \in \mathbb{R}^n.$$

- Secondly, to eliminate every entry of the vector-valued joint-displacement function $\mathbf{e}(t)$ over $[0, T_d]$, we exploit the ZD design formula [i.e., (4.2)] as follows:

$$\dot{\mathbf{e}}(t) = -\gamma \mathbf{e}(t) = -\gamma(\theta(t) - \theta(0)) \quad (14.4)$$

where design parameter $\gamma > 0 \in \mathbb{R}$ is used to adjust the exponential convergence rate of $\mathbf{e}(t)$ to zero. Since the time derivative of $\mathbf{e}(t)$ is $\dot{\mathbf{e}}(t) = \dot{\theta}(t)$, (14.4) is rewritten as

$$\dot{\theta}(t) + \gamma(\theta(t) - \theta(0)) = \mathbf{0} \in \mathbb{R}^n. \quad (14.5)$$

- Thirdly, to achieve (14.5), we define the second ZF as follows:

$$\mathbf{e}(t) = \dot{\theta}(t) + \gamma(\theta(t) - \theta(0)) \in \mathbb{R}^n.$$

By exploiting the ZD design formula (4.2) again, we have $\ddot{\theta}(t) + \gamma\dot{\theta}(t) = -\gamma(\dot{\theta}(t) + \gamma(\theta(t) - \theta(0)))$, which is reformulated as

$$\ddot{\theta}(t) + 2\gamma\dot{\theta}(t) + \gamma^2(\theta(t) - \theta(0)) = \mathbf{0} \in \mathbb{R}^n.$$

- Finally, as the end-effector's task requirement and joint physical limits should be considered, it is better to minimize the performance index $\|\ddot{\theta}(t) + 2\gamma\dot{\theta}(t) + \gamma^2(\theta(t) - \theta(0))\|_2^2/2$, rather than using $\ddot{\theta}(t) + 2\gamma\dot{\theta}(t) + \gamma^2(\theta(t) - \theta(0)) = \mathbf{0}$ directly. Therefore, with $\mathbf{h} = 2\gamma\dot{\theta}(t) + \gamma^2(\theta(t) - \theta(0))$ defined, we obtain

$$\|\ddot{\theta}(t) + 2\gamma\dot{\theta}(t) + \gamma^2(\theta(t) - \theta(0))\|_2^2/2 = \|\ddot{\theta}(t) + \mathbf{h}\|_2^2/2 = (\ddot{\theta} + \mathbf{h})^T(\ddot{\theta} + \mathbf{h})/2, \quad (14.6)$$

which is the acceleration-level RMP performance index for fixed-base redundant robot manipulators.

In summary, by using the ZD approach, we have developed the RMP performance index (14.6) at the joint-acceleration level (showing the application prospect of such a ZD approach). Note that, by minimizing the acceleration-level performance index (14.6) [i.e., “minimize $(\ddot{\theta} + \mathbf{h})^T(\ddot{\theta} + \mathbf{h})/2$ ”], the RMP purpose is thus achieved for fixed-base redundant robot manipulators.

14.3 Scheme and QP Formulations

In this section, based on the proposed performance index (14.6), a novel RMP scheme is further developed and investigated for fixed-base redundant robot manipulators at the joint-acceleration level. In addition, such an acceleration-level RMP scheme is reformulated as a QP, which is solved by a primal–dual neural network [6, 9, 18, 19].

14.3.1 Acceleration-Level RMP Scheme

With joint physical limits (i.e., joint-angle, joint-velocity, and joint-acceleration limits) being considered, the acceleration-level RMP scheme is proposed as follows:

$$\text{minimize } (\ddot{\theta} + \mathbf{h})^T (\ddot{\theta} + \mathbf{h})/2 \quad (14.7)$$

$$\text{subject to } J(\theta)\ddot{\theta} = \ddot{\mathbf{r}}_d - \dot{J}(\theta)\dot{\theta}, \quad (14.8)$$

$$\theta^- \leq \theta \leq \theta^+, \quad (14.9)$$

$$\dot{\theta}^- \leq \dot{\theta} \leq \dot{\theta}^+, \quad (14.10)$$

$$\ddot{\theta}^- \leq \ddot{\theta} \leq \ddot{\theta}^+, \quad (14.11)$$

where $\ddot{\mathbf{r}}_d$ denotes the twice time derivative of the desired end-effector path $\mathbf{r}_d \in \mathbb{R}^m$. In addition, θ^\pm , $\dot{\theta}^\pm$ and $\ddot{\theta}^\pm$ denote the upper and lower limits of the joint-angle, joint-velocity, and joint-acceleration vectors, respectively.

14.3.2 Bound Constraint Transformation Technique

Since the proposed RMP scheme (14.7)–(14.11) is resolved at the joint-acceleration level, the constraints in (14.9)–(14.11) have to be converted to the expressions in terms of joint acceleration $\ddot{\theta}$. In view of the inertia movement, the avoidance of the upper limit of the i th joint (i.e., θ_i^+) in (14.9) can be converted as

$$\ddot{\theta}_i \leq \kappa_\alpha (\lambda \theta_i^+ - \theta_i), \quad (14.12)$$

and the avoidance of the lower limit of the i th joint (i.e., θ_i^-) in (14.9) can be converted as

$$\ddot{\theta}_i \geq \begin{cases} \kappa_\alpha (\lambda \theta_i^- - \theta_i), & \text{for } \theta_i^- < 0 \\ \kappa_\alpha (\theta_i^- + \vartheta - \theta_i), & \text{for } \theta_i^- \geq 0 \end{cases} \quad (14.13)$$

where design parameters $\lambda \in (0, 1)$ and $\vartheta > 0 \in \mathbb{R}$ are selected (e.g., $\lambda = 0.9$ and $\vartheta = 0.0524$ rad), to define critical regions $[\theta_i^-, \lambda \theta_i^-]$ or $[\theta_i^-, \theta_i^- + \vartheta]$ and $[\lambda \theta_i^+, \theta_i^+]$ for joint position variables such that there will appear a deceleration when the robot manipulator enters them [9, 22]. In addition, $\kappa_\alpha > 0 \in \mathbb{R}$ determines the magnitude of such a deceleration. Similarly, the avoidance of the i th joint-velocity limits $\dot{\theta}_i^\pm$ in (14.10) can be converted as

$$\kappa_\beta (\dot{\theta}_i^- - \dot{\theta}_i) \leq \ddot{\theta}_i \leq \kappa_\beta (\dot{\theta}_i^+ - \dot{\theta}_i), \quad (14.14)$$

which guarantees that joint acceleration changes its direction gradually as the joint velocity approaches its limit. Design parameters κ_α and κ_β are selected such that the feasible region of $\ddot{\theta}$ made by the conversion of joint-angle limits and

joint-velocity limits [i.e., (14.9) and (14.10)] are normally not smaller than the original one made by joint-acceleration limits, i.e., bound constraint (14.11). By using (14.12)–(14.14), the acceleration-level avoidance of joint physical limits (14.9)–(14.11) becomes $\zeta^- \leq \ddot{\theta} \leq \zeta^+$. Here, ζ^- and ζ^+ denote, respectively, the resultant lower bound and upper bound synthesized by the joint-angle limits, joint-velocity limits, and joint-acceleration limits. In addition, the i th elements of ζ^- and ζ^+ are defined respectively as

$$\zeta_i^- = \begin{cases} \max\{\kappa_\alpha(\lambda\theta_i^- - \theta_i), \kappa_\beta(\dot{\theta}_i^- - \dot{\theta}_i), \ddot{\theta}_i^-\}, & \text{for } \theta_i^- < 0 \\ \max\{\kappa_\alpha(\theta_i^- + \vartheta - \theta_i), \kappa_\beta(\dot{\theta}_i^- - \dot{\theta}_i), \ddot{\theta}_i^-\}, & \text{for } \theta_i^- \geq 0 \end{cases}$$

$$\zeta_i^+ = \min\{\kappa_\alpha(\lambda\theta_i^+ - \theta_i), \kappa_\beta(\dot{\theta}_i^+ - \dot{\theta}_i), \ddot{\theta}_i^+\}.$$

14.3.3 QP Reformulation

In (14.7), $(\ddot{\theta} + \mathbf{h})^T(\ddot{\theta} + \mathbf{h})/2 = (\ddot{\theta}^T\ddot{\theta} + \ddot{\theta}^T\mathbf{h} + \mathbf{h}^T\ddot{\theta} + \mathbf{h}^T\mathbf{h})/2$. Since the proposed RMP scheme is resolved at the joint-acceleration level and the decision variable vector is joint acceleration $\ddot{\theta}$, the parameter \mathbf{h} [i.e., $\mathbf{h} = 2\gamma\dot{\theta} + \gamma^2(\theta - \theta(0))$ in (14.7)] is viewed as a constant in the performance index. In this situation, $\mathbf{h}^T\mathbf{h}/2$ is also viewed as a constant (with respect to $\ddot{\theta}$) and $\mathbf{h}^T\mathbf{h}/2$ is thus set aside from the performance index. Therefore, the minimization of (14.7) is equivalent to the minimization of $\ddot{\theta}^T\ddot{\theta}/2 + \mathbf{h}^T\ddot{\theta}$ (note that $\ddot{\theta}^T\mathbf{h} = \mathbf{h}^T\ddot{\theta}$).

In light of the above minimization formula and the above bound constraint conversion, with $\mathbf{x} = \ddot{\theta} \in \mathbb{R}^n$ and $W = I \in \mathbb{R}^{n \times n}$, the proposed acceleration-level RMP scheme (14.7)–(14.11) for physically-constrained redundant robot manipulators is reformulated finally as the following QP:

$$\text{minimize } \mathbf{x}^T W \mathbf{x} / 2 + \mathbf{h}^T \mathbf{x} \quad (14.15)$$

$$\text{subject to } C \mathbf{x} = \mathbf{d}, \quad (14.16)$$

$$\zeta^- \leq \mathbf{x} \leq \zeta^+, \quad (14.17)$$

where $C = J(\theta)$ and $\mathbf{d} = \ddot{\mathbf{r}}_d - \dot{J}(\theta)\dot{\theta}$.

14.3.4 QP Solver

According to [6, 9, 18, 19], the presented QP problem (14.15)–(14.17) can be solved by the following primal–dual neural network (PDNN):

$$\dot{\mathbf{u}} = \nu(I + M^T)\{P_\Omega(\mathbf{u} - (M\mathbf{u} + \mathbf{p})) - \mathbf{u}\} \quad (14.18)$$

where design parameter $\nu > 0 \in \mathbb{R}$ is used to scale the convergence rate of the neural network. The piecewise-linear activation-function array $P_\Omega(\cdot)$ can be implemented by using operational amplifiers. In addition, $\Omega = \{\mathbf{u} \in \mathbb{R}^{n+m} \mid \mathbf{u}^- \leq \mathbf{u} \leq \mathbf{u}^+\} \subset \mathbb{R}^{n+m}$,

$$\mathbf{u} = \begin{bmatrix} \mathbf{x} \\ \mathbf{y} \end{bmatrix} \in \mathbb{R}^{n+m}, \quad \mathbf{u}^- = \begin{bmatrix} \zeta^- \\ -\varpi \mathbf{1}_\nu \end{bmatrix} \in \mathbb{R}^{n+m}, \quad \mathbf{u}^+ = \begin{bmatrix} \zeta^+ \\ \varpi \mathbf{1}_\nu \end{bmatrix} \in \mathbb{R}^{n+m},$$

$$M = \begin{bmatrix} W & -C^T \\ C & 0 \end{bmatrix} \in \mathbb{R}^{(n+m) \times (n+m)}, \quad \mathbf{p} = \begin{bmatrix} \mathbf{h} \\ -\mathbf{d} \end{bmatrix} \in \mathbb{R}^{n+m}, \quad \mathbf{1}_\nu = [1, \dots, 1]^T \in \mathbb{R}^m.$$

It is worth mentioning that $P_\Omega(\cdot) : \mathbb{R}^{n+m} \rightarrow \Omega$ is a projection operator, with the i th element of $P_\Omega(\mathbf{u})$ defined as

$$\begin{cases} u_i^-, & \text{if } u_i < u_i^-, \\ u_i, & \text{if } u_i^- \leq u_i \leq u_i^+, \forall i \in \{1, 2, 3, \dots, n+m\}. \\ u_i^+, & \text{if } u_i > u_i^+, \end{cases}$$

$\mathbf{y} \in \mathbb{R}^m$ is the dual decision vector defined for equality constraint (14.16), and $\varpi \gg 0$ is defined sufficiently large (e.g., $\varpi = 10^6$) to replace $+\infty$ numerically. Furthermore, we have the following theorem which guarantees that PDNN (14.18) can globally generate optimal solution x^* to QP (14.15)–(14.17) [6, 9, 18, 19].

Theorem 14.1 *Assume the existence of optimal solution \mathbf{x}^* to QP (14.15)–(14.17). Starting from any initial state $\mathbf{u}(0)$, state vector $\mathbf{u}(t)$ of PDNN (14.18) converges to equilibrium point \mathbf{u}^* , of which the first n elements constitute the optimal solution \mathbf{x}^* to QP (14.15)–(14.17).*

14.4 Illustrative Examples

In this section, to substantiate the efficacy of the proposed acceleration-level RMP scheme (14.7)–(14.11), computer simulations are performed based on PUMA560 robot manipulator, of which the mechanical configuration is shown in Fig. 14.1 and the joint physical parameters (i.e., joint-angle limits θ^\pm , joint-velocity limits $\dot{\theta}^\pm$, joint-acceleration limits $\ddot{\theta}^\pm$) used in the ensuing simulations are given in Table 14.1. In the simulations, the end-effector of the PUMA560 robot manipulator is required to track three different paths, i.e., a pentagram path, an East-Asian character looking like symbol “日” which means “the sun” in Chinese, and the initial letter “V” of the English word “VICTORY.” Note that, when we apply PDNN (14.18) to solving

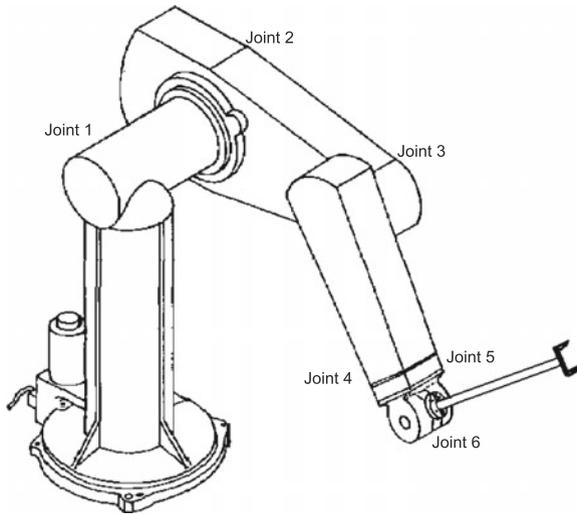


Fig. 14.1 Mechanical configuration of PUMA560 robot manipulator used in simulations

Table 14.1 Joint physical limits used in the PUMA560 simulations

#	θ^+ (rad)	θ^- (rad)	$\dot{\theta}^+$ (rad/s)	$\dot{\theta}^-$ (rad/s)	$\ddot{\theta}^+$ (rad/s ²)	$\ddot{\theta}^-$ (rad/s ²)
1	+2.775	-2.775	+1.5	-1.5	+6.0	-6.0
2	+0.750	-3.892	+1.5	-1.5	+6.0	-6.0
3	+4.049	-0.905	+1.5	-1.5	+6.0	-6.0
4	+2.967	-1.919	+1.5	-1.5	+6.0	-6.0
5	+1.745	-1.745	+1.5	-1.5	+6.0	-6.0
6	+4.625	-4.625	+1.5	-1.5	+6.0	-6.0

the presented QP problem (14.15)–(14.17) [as well as the proposed acceleration-level RMP scheme (14.7)–(14.11)] for controlling the PUMA560 robot manipulator, design parameter $\nu = 10^5$ is used throughout this chapter.

14.4.1 Pentagram-Path Tracking

In the computer simulations of this subsection, the PUMA560 robot manipulator's end-effector is expected to move along a pentagram path with the side length being 0.0707 m. The X-axis, Y-axis, and Z-axis acceleration functions of the desired pentagram path are

$$\ddot{\mathbf{x}}_X(t) = \begin{cases} -\frac{\sqrt{2}l\pi}{T^2} \cos(\frac{\pi}{5}) \sin(\frac{2\pi t}{T}), & \forall t \in [0, T] \\ -\frac{\sqrt{2}l\pi}{T^2} \cos(\frac{\pi}{5}) \sin[\frac{2\pi(t-T)}{T}], & \forall t \in [T, 2T] \\ \frac{\sqrt{2}l\pi}{T^2} \cos(\frac{2\pi}{5}) \sin[\frac{2\pi(t-2T)}{T}], & \forall t \in [2T, 3T] \\ -\frac{\sqrt{2}l\pi}{T^2} \cos(\frac{\pi}{5}) \sin[\frac{2\pi(t-3T)}{T}], & \forall t \in [3T, 4T] \\ \frac{\sqrt{2}l\pi}{T^2} \sin[\frac{2\pi(t-4T)}{T}], & \forall t \in [4T, 5T] \\ \frac{\sqrt{2}l\pi}{T^2} \sin(\frac{\pi}{10}) \sin[\frac{2\pi(t-5T)}{T}], & \forall t \in [5T, 6T] \\ \frac{\sqrt{2}l\pi}{T^2} \sin(\frac{\pi}{10}) \sin[\frac{2\pi(t-6T)}{T}], & \forall t \in [6T, 7T] \\ \frac{\sqrt{2}l\pi}{T^2} \sin[\frac{2\pi(t-7T)}{T}], & \forall t \in [7T, 8T] \\ -\frac{\sqrt{2}l\pi}{T^2} \cos(\frac{\pi}{5}) \sin[\frac{2\pi(t-8T)}{T}], & \forall t \in [8T, 9T] \\ \frac{\sqrt{2}l\pi}{T^2} \cos(\frac{2\pi}{5}) \sin[\frac{2\pi(t-9T)}{T}], & \forall t \in [9T, 10T] \end{cases}$$

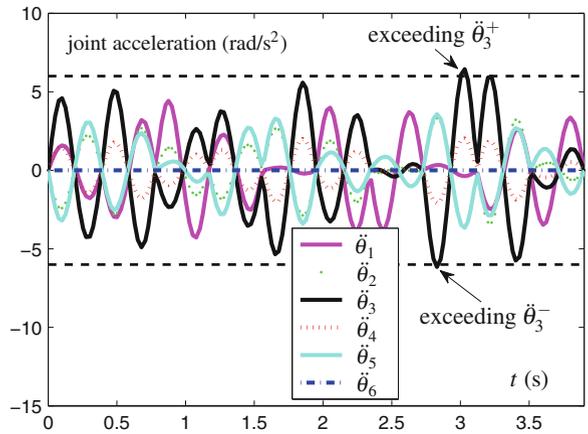
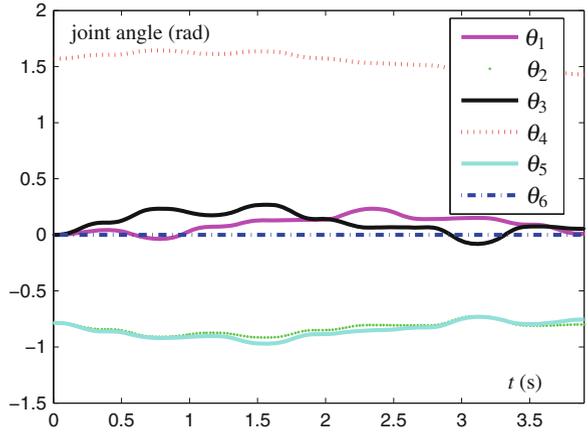
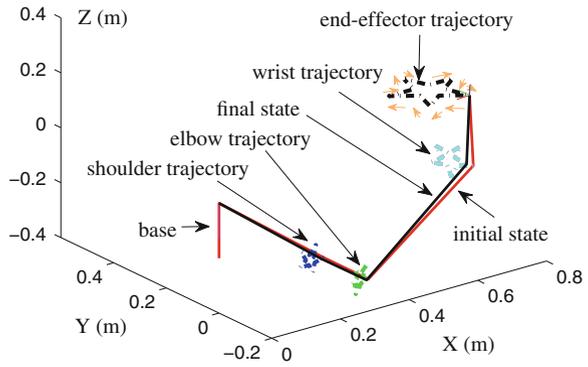
$$\ddot{\mathbf{y}}_Y(t) = \begin{cases} -\frac{\sqrt{2}l\pi}{T^2} \sin(\frac{\pi}{5}) \sin(\frac{2\pi t}{T}), & \forall t \in [0, T] \\ -\frac{\sqrt{2}l\pi}{T^2} \sin(\frac{\pi}{5}) \sin[\frac{2\pi(t-T)}{T}], & \forall t \in [T, 2T] \\ \frac{\sqrt{2}l\pi}{T^2} \sin(\frac{2\pi}{5}) \sin[\frac{2\pi(t-2T)}{T}], & \forall t \in [2T, 3T] \\ \frac{\sqrt{2}l\pi}{T^2} \sin(\frac{\pi}{5}) \sin[\frac{2\pi(t-3T)}{T}], & \forall t \in [3T, 4T] \\ 0, & \forall t \in [4T, 5T] \\ \frac{\sqrt{2}l\pi}{T^2} \cos(\frac{\pi}{10}) \sin[\frac{2\pi(t-5T)}{T}], & \forall t \in [5T, 6T] \\ -\frac{\sqrt{2}l\pi}{T^2} \cos(\frac{\pi}{10}) \sin[\frac{2\pi(t-6T)}{T}], & \forall t \in [6T, 7T] \\ 0, & \forall t \in [7T, 8T] \\ -\frac{\sqrt{2}l\pi}{T^2} \sin(\frac{\pi}{5}) \sin[\frac{2\pi(t-8T)}{T}], & \forall t \in [8T, 9T] \\ -\frac{\sqrt{2}l\pi}{T^2} \sin(\frac{2\pi}{5}) \sin[\frac{2\pi(t-9T)}{T}], & \forall t \in [9T, 10T] \end{cases}$$

$$\ddot{z}_Z(t) = 0, \forall t \in [0, 10T]$$

where the task duration T_d is $10T$, and parameter l should be set appropriately according to the desired length of the line segment in a path-tracking task. Specifically, in the computer simulation of this subsection, $l = 0.1$ m and the task duration T_d is 3.9 s (i.e., $T = 0.39$ s). In addition, during such a task execution, the joints of PUMA560 robot manipulator are expected to start from initial state $\theta(0) = [0, -\pi/4, 0, \pi/2, -\pi/4, 0]^T$ rad, and finally return to the initial state.

First, the inverse kinematics problem of PUMA560 robot manipulator is handled via QP (14.15)–(14.17) with neither joint physical limits nor RMP criterion considered. That is, θ^\pm in (14.9), $\dot{\theta}^\pm$ in (14.10) and $\ddot{\theta}^\pm$ in (14.11) are set as $\pm\varpi 1_v$, and in (14.7), the RMP coefficient $\gamma = 0$. The corresponding simulation results are shown in Fig. 14.2. It is seen from the upper graph of Fig. 14.2 that the end-effector's trajectory is the desired pentagram curve, which shows that the end-effector's primary task

Fig. 14.2 PUMA560 end-effector tracks a pentagram path with neither joint physical limits nor RMP criterion considered



is completed. However, the final state of the robot manipulator (denoted in black) has not returned to the initial one (denoted in red). This can also be seen from the middle graph of Fig. 14.2 which illustrates the θ profiles over the task duration. The exact actual values of the final joints, $\theta(3.9)$, are shown in the second column of Table 14.2, which are evidently different from the initial ones, $\theta(0)$, in the third column. That is to say, the joint angle drift phenomenon has happened. It is worth mentioning that this phenomenon is not desired in industrial applications because we need extra self-motion to readjust the manipulator's configuration. This would thus lead to low efficiency. Besides, the θ and $\dot{\theta}$ profiles during the task execution are shown, respectively, in the middle and lower graphs of Fig. 14.2. Compared with the joint physical limits in Table 14.1, the joint angles and joint velocities in the simulations do not reach their corresponding limits; i.e., the joint-angle limits and joint-velocity limits have not been activated. However, joint acceleration $\ddot{\theta}_3(t)$ exceeds its lower limit -6 rad/s^2 at about $t = 2.83 \text{ s}$ and its upper limit $+6 \text{ rad/s}^2$ at about $t = 3.03 \text{ s}$, which may lead to the damage to the manipulator and is less desirable in applications.

Second, for comparison and for illustration, the simulation results with both joint physical limits and RMP criterion considered are shown in Fig. 14.3. That is, θ^\pm , $\dot{\theta}^\pm$, and $\ddot{\theta}^\pm$ are set correctly according to Table 14.1, $\lambda = 0.9$, $\kappa_\alpha = \kappa_\beta = 20$, and $\gamma = 6$. As seen from the upper left graph of Fig. 14.3, the end-effector of PUMA560 moves along a pentagram path, which is sufficiently close to the desired one. In addition, this solution is repetitive and applicable for PUMA560 working with a cyclic pentagram-path tracking requirement, because the final and initial states coincide well with each other (in the upper right graph of Fig. 14.3). Furthermore, as seen from Fig. 14.3 and Table 14.3, all joint variables are kept within their limited ranges. Besides, the lower left graph of Fig. 14.3 shows the motion trajectories of PUMA560 over the task duration. The end-effector positioning error $\varepsilon = r_d - f(\theta)$ is shown in the lower right graph of Fig. 14.3, illustrating small deviations of the end-effector from the desired path r_d in the X-axis, Y-axis, and Z-axis of the base frame (i.e., ε_X , ε_Y and ε_Z , respectively). As seen from the figure, the maximal component of the positioning error is less than $5.0 \times 10^{-4} \text{ m}$, which illustrates the accuracy of the PUMA560 robot

Table 14.2 Joint drifts (rad) with neither joint limits nor RMP criterion considered when PUMA560 end-effector tracks the pentagram path

Joint	$\theta(3.9)$	$\theta(0)$	$\theta(3.9) - \theta(0)$
θ_1	+0.00957950482	+0.00000000000	+0.00957950482
θ_2	-0.80021692574	-0.78539816340	-0.01481876234
θ_3	+0.05479385946	+0.00000000000	+0.05479385946
θ_4	+1.42816130434	+1.57079632680	-0.14263502246
θ_5	-0.75586229472	-0.78539816340	+0.02953586868
θ_6	+0.00000000000	+0.00000000000	+0.00000000000

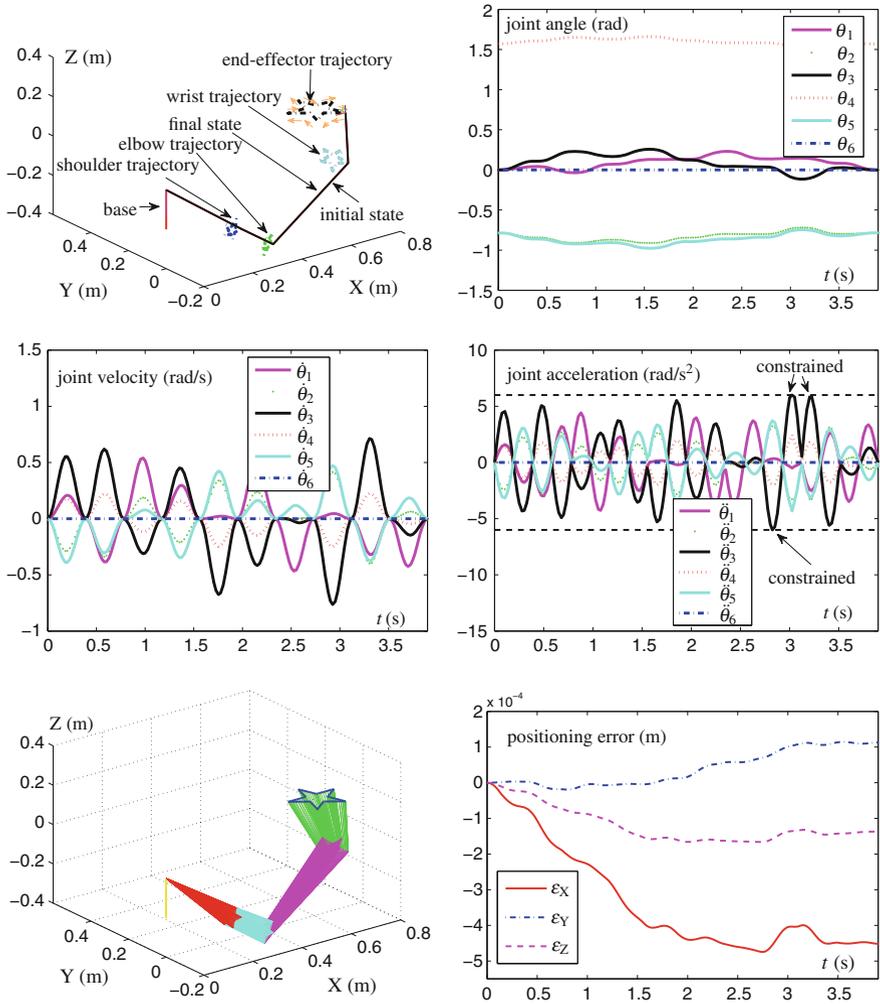


Fig. 14.3 PUMA560 end-effector tracks the pentagram path with both joint physical limits and RMP criterion considered, where PUMA560 final state coincides with its initial state

manipulator synthesized by the proposed acceleration-level RMP scheme when the end-effector tracks the desired pentagram path.

Third, comparing Tables 14.2 and 14.3, we can see that the joint angle drifts in the former (i.e., Table 14.2) are large; in contrast, the joint angle drifts in the latter (i.e., Table 14.3) are quite small (less than 3.41×10^{-3} rad). This shows quantitatively the repetitive accuracy of the proposed RMP scheme. In summary, this example has substantiated well the efficacy of the proposed acceleration-level RMP scheme (14.7)–(14.11) and its QP solver (14.18) (i.e., a PDNN) on redundancy resolution

Table 14.3 Joint drifts (rad) with both joint limits and RMP criterion considered when PUMA560 end-effector tracks a pentagram path

Joint	$\theta(3.9)$	$\theta(0)$	$\theta(3.9) - \theta(0)$
θ_1	+0.00000629838	+0.00000000000	$+6.29838 \times 10^{-6}$
θ_2	-0.78469374929	-0.78539816340	$+7.04414 \times 10^{-4}$
θ_3	-0.00096763343	+0.00000000000	-9.67633×10^{-4}
θ_4	+1.57045580322	+1.57079632680	-3.40524×10^{-3}
θ_5	-0.78474817868	-0.78539816340	$+6.49985 \times 10^{-4}$
θ_6	+0.00000000000	+0.00000000000	+0.00000000000

of physically-constrained robot manipulators, and more importantly, has shown the application prospect of the presented ZD approach.

14.4.2 East-Asian Character Writing

In the simulations of this subsection, the motion trajectory of the PUMA560 end-effector is expected to be an East-Asian character looking like symbol “日” which means “the sun” in Chinese. The word width is 0.1697 m and the word height is 0.3394 m. The X-axis, Y-axis, and Z-axis acceleration functions of the path are

$$\ddot{r}_X(t) = \begin{cases} -\frac{\sqrt{2}l\pi}{T^2} \sin\left(\frac{2\pi t}{T}\right), & \forall t \in [0, T] \\ 0, & \forall t \in [T, 2T] \\ \frac{\sqrt{2}l\pi}{T^2} \sin\left[\frac{2\pi(t-2T)}{T}\right], & \forall t \in [2T, 3T] \\ 0, & \forall t \in [3T, 4T] \\ -\frac{\sqrt{2}l\pi}{T^2} \sin\left[\frac{2\pi(t-4T)}{T}\right], & \forall t \in [4T, 5T] \\ 0, & \forall t \in [5T, 6T] \\ \frac{\sqrt{2}l\pi}{T^2} \sin\left[\frac{2\pi(t-6T)}{T}\right], & \forall t \in [6T, 7T] \end{cases}$$

$$\ddot{r}_Y(t) = \begin{cases} 0, & \forall t \in [0, T] \\ \frac{\sqrt{2}l\pi}{T^2} \sin\left[\frac{2\pi(t-T)}{T}\right], & \forall t \in [T, 2T] \\ 0, & \forall t \in [2T, 3T] \\ -\frac{2\sqrt{2}l\pi}{T^2} \sin\left[\frac{2\pi(t-3T)}{T}\right], & \forall t \in [3T, 4T] \\ 0, & \forall t \in [4T, 5T] \\ \frac{\sqrt{2}l\pi}{T^2} \sin\left[\frac{2\pi(t-5T)}{T}\right], & \forall t \in [5T, 6T] \\ 0, & \forall t \in [6T, 7T] \end{cases}$$

$$\ddot{r}_Z(t) = 0, \forall t \in [0, 7T]$$

where the task duration T_d is $7T$, and parameter ι should be set appropriately according to the desired length of the line segment in a path-tracking task. Specifically, in the simulations of this subsection, design parameter $\iota = 0.24\text{m}$ and the task duration T_d is 4.55 s (i.e., $T = 0.65\text{ s}$). In addition, the initial joint state $\theta(0) = [0, -\pi/4, 0, \pi/2, -\pi/4, 0]^T$ rad.

First, we show, in Fig. 14.4, the redundancy-resolution results with joint physical limits considered (i.e., $\lambda = 0.9$ and $\kappa_\alpha = \kappa_\beta = 20$) but without considering cyclic motion criterion (i.e., $\gamma = 0$). As seen from the upper left graph of Fig. 14.4, the end-effector of the PUMA560 robot manipulator moves along a “ \square ” path, which indicates that the robot manipulator completes the desired path-tracking task. From the upper left graph of Fig. 14.4, we can also see that such a solution is not repetitive, because the final and initial states of the manipulator are not equal. Thus, if such a non-repetitive solution is exploited to control the PUMA560 robot manipulator, then an additional self-motion readjustment is needed, which would be of low efficiency in engineering applications.

Second, for comparison as well as for illustration, the inverse kinematics problem is finally solved via the proposed acceleration-level RMP scheme (14.7)–(14.11) with both joint physical limits and RMP criterion considered (i.e., $\lambda = 0.9$, $\kappa_\alpha = \kappa_\beta = 20$ and $\gamma = 5$). The corresponding simulation results are shown in the rest graphs of Fig. 14.4. As seen from the upper right graph of Fig. 14.4, the end-effector of

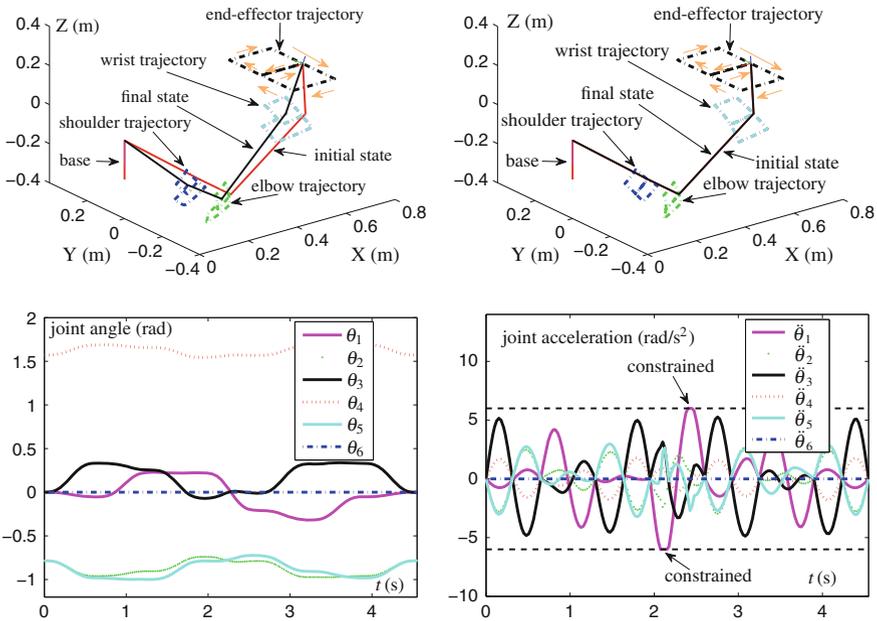


Fig. 14.4 PUMA560 end-effector tracks a path of an East-Asian character looking like “ \square ” synthesized by the scheme without and with RMP criterion considered

PUMA560 robot manipulator tracks a “ \square ” path, which is sufficiently close to the desired one. In addition, this solution is repetitive and applicable for PUMA560 robot manipulator working with a cyclic motion requirement, because the final and initial states of the robot manipulator coincide with each other as shown in the upper right graph of Fig. 14.4, which can also be seen from the θ profiles in the lower left graph of Fig. 14.4. Furthermore, as seen from the lower two graphs of Fig. 14.4 and Table 14.1, the joint angles and joint accelerations are kept within their limited ranges. Note that the joint velocities are also within their ranges and the corresponding figure is omitted due to the space limitation. It is worth pointing out that the maximum component of the robot end-effector’s positioning error is also small during the “ \square ” path tracking task execution, i.e., less than 8×10^{-4} m. The efficacy of the proposed RMP scheme (14.7)–(14.11) at the joint-acceleration level subject to joint-angle limits, joint-velocity limits and joint-acceleration limits is thus well substantiated.

14.4.3 “V” Path Path Tracking

In order to further demonstrate the efficacy and the general applicability of the proposed acceleration-level RMP scheme (14.7)–(14.11) as well as the corresponding

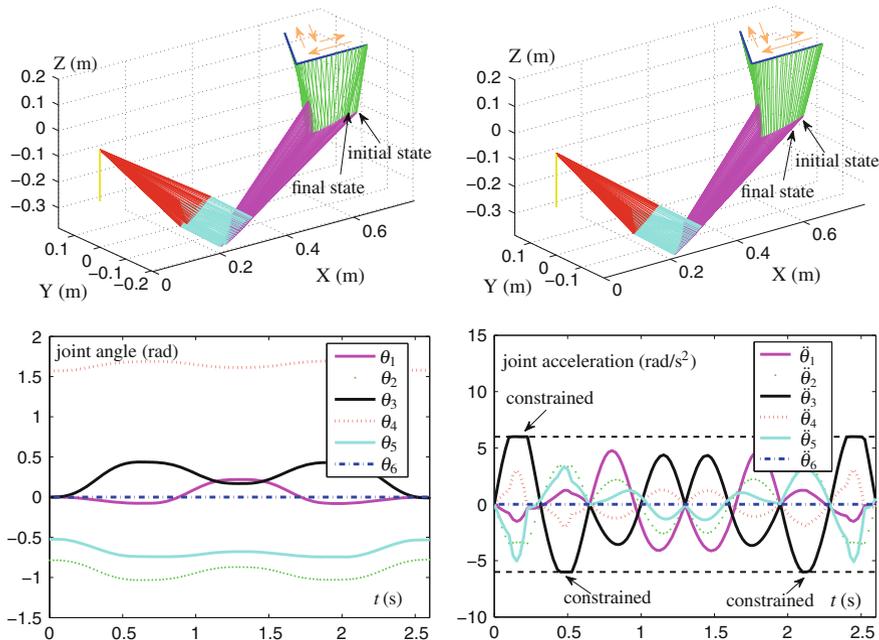


Fig. 14.5 PUMA560 end-effector tracks the path of the initial letter “V” of English word “VICTORY” synthesized by scheme (14.7)–(14.11) without and with RMP criterion considered (i.e., corresponding to $\gamma = 0$ and $\gamma \neq 0$)

QP solver [i.e., PDNN (14.18)], the initial letter “V” writing of English word “VICTORY” is also performed based on PUMA560 robot manipulator in this subsection.

Figure 14.5 illustrates the simulation results of PUMA560 robot manipulator’s end-effector tracking the “V” path. Specifically, the upper left graph of Fig. 14.5 shows the simulation result with joint physical limits considered but without considering RMP criterion (i.e., $\gamma = 0$). From such a figure, we can see that the initial and final states of the robot manipulator do not match. For comparison, the simulation results with both joint physical limits and RMP criterion considered are shown in the rest graphs of Fig. 14.5. These simulation results illustrate that, by applying the proposed acceleration-level RMP scheme (14.7)–(14.11) to the PUMA560 robot manipulator, the joint drift phenomenon is remedied, all joint variables [i.e., joint angle $\theta(t)$, joint velocity $\dot{\theta}(t)$ and joint acceleration $\ddot{\theta}(t)$] are kept within their limited ranges, and that the positioning error of the robot end-effector is small. These comparisons of simulation results further substantiate the efficacy and accuracy of the proposed RMP scheme (14.7)–(14.11), as well as the effectiveness of the PDNN (14.18) as a QP solver.

14.4.4 Comparisons with Velocity-Level RMP Scheme

To substantiate the superiority of the proposed acceleration-level RMP scheme (14.7)–(14.11), comparisons between the RMP schemes at the joint-acceleration and joint-velocity levels are shown in this subsection. As presented in [6, 17, 18], the velocity-level RMP scheme is formulated as follows:

$$\text{minimize} \quad (\dot{\theta} + \rho)^T(\dot{\theta} + \rho)/2 \quad (14.19)$$

$$\text{subject to} \quad J(\theta)\dot{\theta} = \dot{\mathbf{r}}_d, \quad (14.20)$$

$$\theta^- \leq \theta \leq \theta^+, \quad (14.21)$$

$$\dot{\theta}^- \leq \dot{\theta} \leq \dot{\theta}^+, \quad (14.22)$$

where $\rho = \gamma(\theta(t) - \theta(0))$, and $\dot{\mathbf{r}}_d \in \mathbb{R}^m$ is the time derivative of the desired end-effector path \mathbf{r}_d .

Comparing these two different level RMP schemes, we find that the proposed acceleration-level RMP scheme (14.7)–(14.11) can incorporate the joint-acceleration limits into the scheme formulation readily, but the presented velocity-level RMP scheme (14.19)–(14.22) can not do it. In other words, the redundancy-resolution results generated by the velocity-level scheme probably exceed the physical acceleration limits sometimes. To explain this point, without loss of generality, we still take tracking the pentagram-path for example. The initial state $\theta(0)$, task duration T_d and the corresponding design parameters κ_α , κ_β , λ , and γ are set the same as before. In this comparison example, $l = 0.12$ m. The acceleration profiles of the RMP scheme resolved, respectively, at the velocity level and acceleration level when the end-effector of PUMA560 robot manipulator tracks the pentagram path

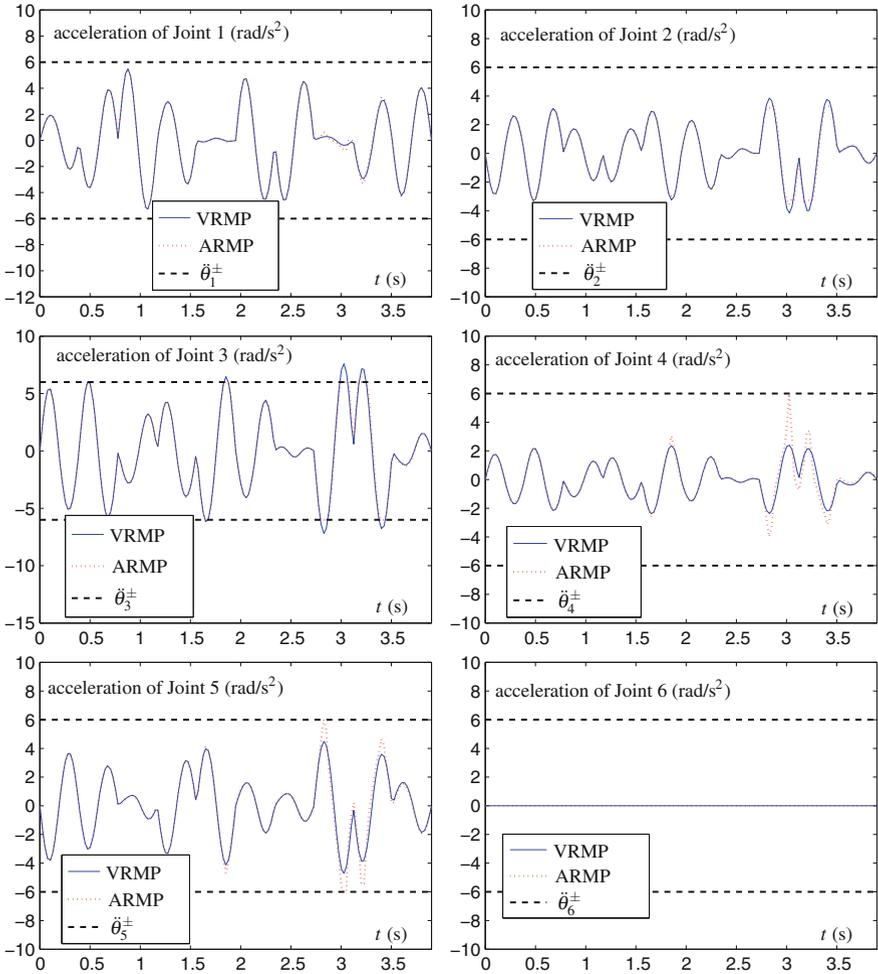


Fig. 14.6 Profiles of the joint acceleration $\ddot{\theta} \in \mathbb{R}^6$: comparisons between the RMP schemes, respectively, resolved at the joint-velocity and joint-acceleration levels

are presented in Fig. 14.6. Note that, in Fig. 14.6, VRMP denotes the RMP scheme at the joint-velocity level [i.e., (14.19)–(14.22)] and ARMP denotes the RMP scheme at the joint-acceleration level [i.e., (14.7)–(14.11)]. As shown in Fig. 14.6, all the joint-acceleration profiles generated by ARMP, denoted by the red dot line, are constrained within the acceleration limits $\ddot{\theta}_i^\pm$ (with $i = 1, 2, \dots, 6$) denoted by the black dash line. Even some joint accelerations (i.e., $\ddot{\theta}_3$, $\ddot{\theta}_4$ and $\ddot{\theta}_5$) reach their limits in some durations, the bound constraints keep them within their physical limits. For instance, $\ddot{\theta}_3$ reaches its upper or lower limit many times (e.g., during [1.828, 1.875], [2.801, 2.875], [2.978, 3.075], [3.175, 3.253] and [3.378, 3.428] s)

as shown in Fig. 14.6, and it stops increasing in these durations. This illustrates that the bound constraint (14.11) is activated effectively. On the contrary, joint acceleration $\ddot{\theta}_3$ indirectly resolved by VRMP, denoted by the blue solid line, exceeds the limits $\ddot{\theta}_3^{\pm}$ because the joint-acceleration limits have not been considered in the VRMP. In actual application, the case of exceeding physical limits is unallowed, because it may lead to the acceleration saturation and even destroy the physical robot. From this point, the RMP scheme at the joint-acceleration level is readily applied and preferred. Another advantage of the proposed acceleration-level RMP scheme (14.7)–(14.11) is that, in applications, some robot manipulators are controlled by acceleration (such as the robot in [24]). This kind of robot cannot directly use the redundancy-resolution results generated by the presented velocity-level RMP scheme (14.19)–(14.22), and it is less efficient to transform the resolution results (i.e., joint velocity $\dot{\theta}$) of the velocity-level RMP scheme to joint acceleration $\ddot{\theta}$.

In summary, compared with the presented velocity-level RMP scheme (14.19)–(14.22), the proposed acceleration-level RMP scheme (14.7)–(14.11) is safer and more applicable (showing again the successful application of the presented ZD approach to RMP of fixed-base redundant robot manipulators).

14.5 Summary

In this chapter, by defining two different ZFs and by exploiting the ZD design formula, the acceleration-level RMP performance index (14.9) has been proposed, developed, and investigated. Based on such a performance index, the acceleration-level RMP scheme (14.7)–(14.11) has been further presented and investigated to remedy the joint-angle drift phenomenon of fixed-base redundant robot manipulators, which is reformulated as a QP (14.15)–(14.17) and then is solved by PDNN (14.18). Computer simulation results based on PUMA560 robot manipulator performing three different types of end-effector path-tracking tasks have substantiated well the effectiveness, accuracy, and safety of the proposed acceleration-level RMP scheme for physically-constrained redundant robot manipulators, and more importantly, have shown the application prospect of the presented ZD approach to robotic redundancy resolution.

References

1. Janabi-Sharifi F, Hassanzadeh I (2011) Experimental analysis of mobile-robot teleoperation via shared impedance control. *IEEE Trans Syst Man Cybern B* 41(2):591–606
2. Su J, Xie W (2011) Motion planning and coordination for robot systems based on representation space. *IEEE Trans Syst Man Cybern B* 41(1):248–299
3. Guo D, Zhang Y (2014) Li-function activated ZNN with finite-time convergence applied to redundant-manipulator kinematic control via time-varying Jacobian matrix pseudoinversion. *Appl Soft Comput* 24:158–168

4. Sun C, Xu WL, Bronlund JE, Morgenstern M (2014) Dynamics and compliance control of a linkage robot for food chewing. *IEEE Trans Ind Electron* 61(1):377–386
5. Guo D, Zhang Y (2014) Acceleration-level inequality-based MAN scheme for obstacle avoidance of redundant robot manipulators. *IEEE Trans Ind Electron* 61(12):6903–6914
6. Zhang Y, Zhang Z (2013) Repetitive motion planning and control of redundant robot manipulators. Springer, New York
7. Okadome Y, Nakamura Y, Ishiguro H (2014) Predictive control method for a redundant robot using a non-parametric predictor. *Adv Robot* 28(10):647–657
8. Zhang Y, Wang J, Xia Y (2003) A dual neural network for redundancy resolution of kinematically redundant manipulators subject to joint limits and joint velocity limits. *IEEE Trans Neural Netw* 14(3):658–667
9. Zhang Z, Zhang Y (2012) Acceleration-level cyclic-motion generation of constrained redundant robots tracking different paths. *IEEE Trans Syst Man Cybern B* 42(4):1257–1269
10. Guo D, Zhang Y (2012) A new inequality-based obstacle-avoidance MVN scheme and its application to redundant robot manipulators. *IEEE Trans Syst Man Cybern C* 42(6):1326–1340
11. Maciekewski AA, Klein CA (1985) Obstacle avoidance for kinematically redundant manipulators in dynamically varying environments. *Int J Robot Res* 4(3):109–117
12. Cheng FT, Sheu RJ, Chen TH (1995) The improved compact QP method for resolving manipulator redundancy. *IEEE Trans Syst Man Cybern* 25(11):1521–1530
13. Zhang Z, Zhang Y (2013) Variable joint-velocity limits of redundant robot manipulators handled by quadratic programming. *IEEE/ASME Trans Mech* 18(2):674–686
14. Tchou K, Jakubiak J (2005) A repeatable inverse kinematics algorithm with linear invariant subspaces for mobile manipulators. *IEEE Trans Syst Man Cybern B* 35(5):1051–1057
15. Siciliano B, Khatib O (2008) Springer handbook of robotics. Springer, Heidelberg
16. Siciliano B, Sciavicco L, Villani L, Oriolo G (2009) Robotics: modelling, planning and control. Springer, London
17. Zhang Y, Tan Z, Chen K, Yang Z, Lv X (2009) Repetitive motion of redundant robots planned by three kinds of recurrent neural networks and illustrated with a four-link planar manipulator's straight-line example. *Robot Auton Syst* 57(6–7):645–651
18. Zhang Y, Lv X, Li Z, Yang Z, Chen K (2008) Repetitive motion planning of PA10 robot arm subject to joint physical limits and using LVI-based primal-dual neural network. *Mechatronics* 18(9):475–485
19. Cai B, Zhang Y (2010) Bi-criteria optimal control of redundant robot manipulators using LVI-based primal-dual neural network. *Optim Control Appl Methods* 31(3):213–229
20. Roberts RG, Maciejewski AA (1993) Repeatable generalized inverse control strategies for kinematically redundant manipulators. *IEEE Trans Autom Control* 38(5):689–698
21. Shamir T, Yomdin Y (1988) Repeatability of redundant manipulators: mathematical solution of the problem. *IEEE Trans Autom Control* 33(11):1004–1009
22. Ge SS, Zhang Y, Lee TH (2004) An acceleration-based weighting scheme for minimum-effort inverse kinematics of redundant manipulators. In: Proceedings of the IEEE international symposium on intelligent control, pp 275–280
23. Zhang Y, Yi C (2011) Zhang neural networks and neural-dynamic method. Nova Science Publishers, New York
24. Xue F, Hou Z, Deng H (2011) Balance control for an acrobat. In: Proceedings of the IEEE international conference on control decision, pp 3426–3429