Robotica (2007) volume 25, pp. 103–112. © 2006 Cambridge University Press
doi:10.1017/S0263574706003018 Printed in the United Kingdom

Adaptive path-constrained control of a robotic manipulator in a task space

Mirosław Galicki

Institute of Medical Statistics, Computer Sciences and Documentation, Friedrich Schiller University Jena, Bachstrasse 18, D-07740 Jena, Germany
Department of Mechanical Engineering, University of Zielona Góra, Podgórna 50, 65-246 Zielona Góra, Poland
E-mail: galicki@imsid.uni-jena.de

(Received in Final Form: July 19, 2006, First published online: September 4, 2006)

SUMMARY
This study addresses the problem of adaptive controlling of both a nonredundant and a redundant robotic manipulator with state-dependent constraints. The task of the robot is to follow a prescribed geometric path given in the task space, by the end-effector. The aforementioned robot task has been solved on the basis of the Lyapunov stability theory, which is used to derive the control scheme. A new adaptive Jacobian controller is proposed in the paper for the path following of the robot, with both uncertain kinematics and dynamics. The numerical simulation results carried out for a planar redundant three-DOF (three degrees of freedom) manipulator whose end-effector follows a prescribed geometric path given in a two-dimensional (2D) task space, illustrate the trajectory performance of the proposed control scheme.

KEYWORDS: Robotic manipulator; Adaptive control; Lyapunov stability.

1. Introduction

Recently, there has been an increase in the application of both nonredundant and redundant robotic manipulators to useful practical tasks that are specified in terms of a geometric path to be followed by the end-effector. Such tasks require high precision and stability of performance. In order to fulfil these requirements, control algorithms should take into account the dynamic nonlinearities resulting from the robot equations. However, they are never known exactly in practice. Therefore, it is particularly important to design control strategies that ensure accurate and stable path following despite the uncertainties and unknown payloads. In most practical situations, geometric paths are expressed in the robot task space. Consequently, it seems natural to develop task space control schemes, which, as opposed to the well-known joint space control techniques, do not require the knowledge of reference joint trajectories and their time derivatives.

Thus, the aim of path following control is to move the end-effector of the manipulator (from its arbitrary initial configuration) along the geometric path given in a task space.

Application of redundant manipulators to such tasks creates additional possibilities during the path following by the end-effector. Redundant degrees of freedom (DOF) make it possible to achieve some useful objectives such as collision avoidance with obstacles in the work space, joint limit avoidance, and/or avoiding the singular configurations when the manipulator moves. Application of redundant manipulators to such tasks complicates their performance since these manipulators provide, in general, no unique solutions. Due to unknown path parameterization, path following by the end-effector of a nonredundant manipulator becomes nonunique, too. Consequently, some objective criteria should be specified to solve the robot tasks uniquely. Minimization of the performance time is mostly considered in the literature. One may find several approaches in this context. The task of time-optimal control of nonredundant manipulators whose end-effector moves along a prescribed geometric path, has been solved both theoretically and practically (by means of effective algorithms) by reducing the whole optimization problem to path coordinates.[1–7] However, these algorithms are not applicable in the case of other useful tasks to be accomplished simultaneously, and require the full knowledge of robot kinematics and dynamics.

Using the concept of a regular trajectory and the extended state space, the structure of path-constrained time-optimal controls has been studied[8,9] for kinematically redundant manipulators. Moreover, the efficient numerical procedures to find such controls have also been proposed.[10,11] Nevertheless, these algorithms require full knowledge of manipulator Jacobian matrix and robot dynamic equations. Although all the aforementioned algorithms produce optimal solutions, they are not suitable for real-time computations due to their computational complexity. It results from either a two-point boundary value problem to be solved in the case of algorithms proposed in refs. 1–7 or generating a weakly convergent sequence of controls being the corresponding solutions of continuous linear optimal control problems for the case of computational methods.[8–11] Consequently, it is obvious that these schemes require the determination of controls in the whole time interval of task accomplishment and then their application to robotic manipulators. Therefore, in order to control the robot, it is obvious to attempt other techniques that do not require the knowledge of controls in the whole time interval. They are based on solving the Cauchy problem (known from the theory of ordinary differential equations), which is more suitable for real-time computations than a two-point boundary value problem.

https://doi.org/10.1017/S0263574706003018 Published online by Cambridge University Press

Using online trajectory time scaling, and dynamic and computed torque laws, respectively, a nearly time-optimal path-tracking control for nonredundant robotic manipulators with partially uncertain dynamics has been presented.[12,13] However, these algorithms require the solution of inverse kinematic problem along the path.

A technique that avoids solving an inverse of the robot kinematic equations and uses the exact Jacobian matrix has been offered in refs. 14 and 15 for determining a collision-free trajectory of redundant manipulators operating in both a static environment[14] and in a dynamic one.[15] Recently, a generalized transpose Jacobian controller with gravity compensation and a nonlinear (saturating) derivative term has been introduced in ref. 16 to generate robot controls subject to geometric path and actuator constraints.

As is known, many robotic controllers have been proposed to solve both a setpoint control problem (a regulation task)[17–27] and the task of trajectory tracking.[28–35] However, most of these controllers assume full knowledge of manipulator kinematic equations. Recently, several approximate Jacobian set point controllers have been proposed[36–38] to tackle uncertainties in both robot kinematics and dynamics. The controllers proposed do not require the exact knowledge of Jacobian matrix and dynamic equations. However, the results in refs. 36–38 are applicable only to a set point control of a robot.

This paper introduces a new class of adaptive path-following controllers that do not require the full knowledge of kinematic and dynamic equations in the control laws and are based on the solution of the Cauchy problems. Consequently, they are suitable for real-time controlling of uncertain robotic manipulators. Motivated in part by the dissipativity and adaptivity methodology,[28] we develop path-following controllers whose structure is composed of transpose adaptive Jacobian controller plus a nonlinear term including an estimated control. Under the assumption of a full-rank adaptive Jacobian matrix, the proposed control scheme has been derived based on the Lyapunov stability theory. By using sensory feedback of the end-effector position, it is also shown that the end-effector is able to follow a prescribed geometric path for robots with both uncertain kinematics and dynamics. Furthermore, new adaptive laws extending the adaptive algorithm from ref. 28 to tackle kinemetic uncertainties are proposed. In fact, to the best of our knowledge, a convergent adaptive path-following controller for manipulators with both uncertain kinematics and dynamics has not yet been proposed so far in the robotics literature, although it is important in motion control. It should be noted that approximate Jacobian set point controllers from refs. 36–38 cannot be directly applied to our task. The reason is that the approximate Jacobian matrix in refs. 36–38 does not include kinematic parameters to be adapted, and the error of approximation is *a priori* bounded. On the other hand, the controller proposed in our work adaptively varies the kinematic parameters of the Jacobian matrix and the dynamic ones in such a way as to have a stable following of a geometric path by the end-effector.

The paper is structured as follows. Section 2 formulates the problem of path-following control in the task space and summarizes useful properties of robot kinematic and dynamic equations. The main result is presented in Section 3, where a class of transpose adaptive Jacobian controllers with estimated control term is proposed, and conditions on the controller gains to ensure asymptotic stability are provided. Computer simulations carried out in Section 4 on a redundant manipulator comprising three revolute joints and operating in a two-dimensional (2D) task space confirm theoretical results obtained in Section 3. Finally, some concluding remarks are given in Section 5.

2. Problem Formulation

The control scheme designed in the next section is applicable to holonomic mechanical systems comprising both nonredundant and redundant manipulators considered here, which are described, in general, by the following dynamic equations, expressed in generalized coordinates (joint coordinates) $\boldsymbol{q} = (q_1, \ldots, q_n)^T \in \mathbb{R}^n$

$$\boldsymbol{M}(\boldsymbol{q})\ddot{\boldsymbol{q}} + \boldsymbol{C}(\boldsymbol{q}, \dot{\boldsymbol{q}})\dot{\boldsymbol{q}} + \boldsymbol{g}(\boldsymbol{q}) = \boldsymbol{u} \tag{1}$$

where $\boldsymbol{M}(\boldsymbol{q})$ is the $n \times n$ inertia matrix, $\boldsymbol{C}(\boldsymbol{q}, \dot{\boldsymbol{q}})\dot{\boldsymbol{q}}$ is the n-dimensional vector representing centrifugal and Coriolis forces, $\boldsymbol{g}(\boldsymbol{q}) = (g_1(\boldsymbol{q}), \ldots, g_n(\boldsymbol{q}))^T$ stands for the n-dimensional vector of gravity forces, $\mathbb{R}^n \ni \boldsymbol{u} = (u_1, \ldots, u_n)^T$ is the vector of controls (torques/forces), and n denotes the number of kinematic pairs of the Vth class. Several important properties of dynamic Eq. (1) may be derived as follows.[39]

1. The inertia matrix $\boldsymbol{M}(\boldsymbol{q})$ is symmetric and positive-definite for all $\boldsymbol{q} \in \mathbb{R}^n$.
2. Matrix $\frac{\dot{\boldsymbol{M}}(\boldsymbol{q})}{2} - \boldsymbol{C}(\boldsymbol{q}, \dot{\boldsymbol{q}})$ is skew-symmetric so that

$$\forall \boldsymbol{v},\ \boldsymbol{q},\ \dot{\boldsymbol{q}} \in \mathbb{R}^n \quad \left\langle \boldsymbol{v},\ \left(\frac{\dot{\boldsymbol{M}}(\boldsymbol{q})}{2} - \boldsymbol{C}(\boldsymbol{q}, \dot{\boldsymbol{q}}) \right) \boldsymbol{v} \right\rangle = \boldsymbol{0} \tag{2}$$

   where $\langle , \rangle$ is the scalar product of vectors.
3. The dynamic Eq. (1) is linear with respect to an ordered set of physical parameters $\boldsymbol{X} = (X_1, \ldots, X_d)^T$, i.e.,

$$\boldsymbol{M}(\boldsymbol{q})\ddot{\boldsymbol{q}} + \boldsymbol{C}(\boldsymbol{q}, \dot{\boldsymbol{q}})\dot{\boldsymbol{q}} + \boldsymbol{g}(\boldsymbol{q}) = \boldsymbol{D}(\boldsymbol{q}, \dot{\boldsymbol{q}}, \dot{\boldsymbol{q}}, \ddot{\boldsymbol{q}})\boldsymbol{X} \tag{3}$$

   where $\boldsymbol{D}(\boldsymbol{q}, \dot{\boldsymbol{q}}, \dot{\boldsymbol{q}}, \ddot{\boldsymbol{q}})$ is called the $n \times d$ dynamic regressor matrix; d stands for the number of the physical parameters such as link masses and inertias.

In most applications of robotic manipulators, a desired path for the end-effector is specified in the task space such as the visual space or the Cartesian space. The aim is that the end-effector follows a prescribed geometric path (given in the m-dimensional task space) described by

$$\boldsymbol{p}(\boldsymbol{q}) - \boldsymbol{\theta}(s) = \boldsymbol{0} \tag{4}$$

where $\boldsymbol{p} : \mathbf{R}^n \longrightarrow \mathbf{R}^m$ denotes an m-dimensional, nonlinear (with respect to vector $\boldsymbol{q}$) mapping constructed from the kinematic equations of the manipulator; $\boldsymbol{p}(\boldsymbol{q}) = (p_1(\boldsymbol{q}), \ldots, p_m(\boldsymbol{q}))^T$; $\boldsymbol{\theta}(s) = (\theta_1(s), \ldots, \theta_m(s))^T$ stands for a given geometric path; s is the current parameter of the

https://doi.org/10.1017/S0263574706003018 Published online by Cambridge University Press

path (e.g., its length); $s \in [0, s_{\max}]$; $s_{\max}$ is the maximal path length. The mapping $\boldsymbol{\theta}$ is assumed to be bounded together with the first and the second derivative and is not degenerated, i.e., $\|\frac{\mathrm{d}\boldsymbol{\theta}}{\mathrm{d}s}\| > 0$. The kinematic equations of a manipulator are independent (rank $(\frac{\partial \boldsymbol{p}}{\partial \boldsymbol{q}}) = m$). In general (i.e., when rank $(\frac{\partial \boldsymbol{p}}{\partial \boldsymbol{q}}) \leq m$), we should require that rank $(\frac{\partial \boldsymbol{p}}{\partial \boldsymbol{q}})$ = rank $([\frac{\partial \boldsymbol{p}}{\partial \boldsymbol{q}} \ \frac{\mathrm{d}\boldsymbol{\theta}}{\mathrm{d}s}])$ in order to guarantee consistency of the robotic task [Eq. (4)]. The problem is to determine control $\boldsymbol{u}$, which generates manipulator trajectory $\boldsymbol{q} = \boldsymbol{q}(t)$ and path parameterization $s = s(t)$, satisfying Eq. (4) for each $t \in [0, T]$, where T denotes an (unknown) time horizon of task performance. It is natural to assume that at the initial moment $t = 0$, for which $s(0) = 0$, a given (by definition) initial configuration $\boldsymbol{q}(0) = \boldsymbol{q}_0$ satisfies Eq. (4), i.e.,

$$\boldsymbol{p}(\boldsymbol{q}_0) - \boldsymbol{\theta}(0) = \boldsymbol{0}. \tag{5}$$

The final path parameterization fulfils the equality

$$s(T) - s_{\max} = 0. \tag{6}$$

Furthermore, at the initial and the final time moment, the manipulator and path velocities equal zero, i.e.,

$$\dot{\boldsymbol{q}}(0) = \dot{\boldsymbol{q}}(T) = \boldsymbol{0} \tag{7}$$

and

$$\dot{s}(0) = \dot{s}(T) = 0. \tag{8}$$

From the view point of the control theory, Eqs. (4)–(8) present state-dependent equality constraints, where the state vector is defined as an ordered quadruple consisting of joint and path coordinates together with their time derivatives, i.e., manipulator and path velocities. As is known, the task space velocity $\dot{\boldsymbol{p}}$ is related to the joint space velocity $\dot{\boldsymbol{q}}$ as

$$\dot{\boldsymbol{p}} = \boldsymbol{J}(\boldsymbol{q}, \boldsymbol{Y})\dot{\boldsymbol{q}} \tag{9}$$

where $\boldsymbol{J}(\boldsymbol{q}, \boldsymbol{Y}) = \frac{\partial \boldsymbol{p}}{\partial \boldsymbol{q}}$ is the $m \times n$ Jacobian matrix; $\boldsymbol{Y}$ stands for an ordered set of kinematic parameters $\boldsymbol{Y} = (Y_1, \ldots, Y_k)^T$ such as link lengths and joint offsets, where k denotes the number of kinematic parameters. The property of differential Eq. (9) is given below.

4. The right-hand side of Eq. (9) is linear with respect to $\boldsymbol{Y}$. Consequently, Eq. (9) can be expressed as

$$\dot{\boldsymbol{p}} = \boldsymbol{J}(\boldsymbol{q}, \boldsymbol{Y})\dot{\boldsymbol{q}} = \boldsymbol{K}(\boldsymbol{q}, \ \dot{\boldsymbol{q}})\boldsymbol{Y} \tag{10}$$

where $\boldsymbol{K}(\boldsymbol{q}, \ \dot{\boldsymbol{q}})$ is called the $m \times k$ kinematic regressor matrix.

In order to simplify further computations, $s_{\max}$ is assumed to be equal to 1, i.e., $s_{\max} = 1$. Let us define errors $\boldsymbol{e}$ and e_{m+1} of path following (task errors) as

$$\begin{aligned} \boldsymbol{e} &= (e_1, \ldots, e_m)^T = \boldsymbol{p}(\boldsymbol{q}) - \boldsymbol{\theta}(s) \\ &= (p_1 - \theta_1, \ldots, p_m - \theta_m)^T e_{m+1} = s - 1. \end{aligned} \tag{11}$$

Many commercial sensors are available for the measurement of end-effector position $\boldsymbol{p}$ (and consequently task error $\boldsymbol{e}$), such as vision systems, electromagnetic measurement systems, position-sensitive detectors, or laser tracking systems. There are two reasons to check the error in the path variable. First, based on a given value of s, the current point on the path $\boldsymbol{\theta}(s)$ is determined, which makes it possible to measure task error $\boldsymbol{e}$. Second, the path error $e_{m+1} = s - 1$ provides information about how far the end of geometric path $\boldsymbol{\theta}(1)$ is from the current path point $\boldsymbol{\theta}(s)$ pursued by the end-effector. For the revolute kinematic pairs considered here, mapping $\boldsymbol{p}(\cdot)$ is bounded. Consequently, we have the property given below.

5. Boundedness of mappings $\boldsymbol{p}(\cdot)$, $\boldsymbol{\theta}(\cdot)$ implies that the task error $\boldsymbol{e}$ is bounded.

Equations (1) and (4)–(8) formulate the robot task as a control problem. The fact is that there are state constraints that make the solution of this problem difficult. The next section presents an approach that renders it possible to solve the control problem [Eqs. (1) and (4)–(8)] making use of the Lyapunov stability theory.

3. Adaptive Path Control of a Robotic Manipulator

In this section, we present the adaptive Jacobian path-following controllers for robots with both uncertain kinematics and dynamics. In our approach, the exact knowledge of the robot dynamic equations and the Jacobian matrix is not required for updating the uncertain parameters. In the presence of kinematic uncertainty, the parameters of the Jacobian matrix are uncertain, and hence Eq. (10) can be expressed as

$$\widehat{\boldsymbol{J}}\dot{\boldsymbol{q}} = \boldsymbol{K}(\boldsymbol{q}, \ \dot{\boldsymbol{q}})\widehat{\boldsymbol{Y}} \tag{12}$$

where $\widehat{\boldsymbol{J}} = \boldsymbol{J}(\boldsymbol{q}, \widehat{\boldsymbol{Y}}) \in \mathbb{R}^{m \times n}$ is an adaptive Jacobian matrix and $\widehat{\boldsymbol{Y}}$ stands for the vector of the estimated kinematic parameters. In order to show the stability of the path-following control system in the presence of kinematic and dynamic uncertainties, we define an adaptive joint-space sliding vector $\boldsymbol{z}$ using quantities $\widehat{\boldsymbol{Y}}$ and $\widehat{\boldsymbol{J}}$, as

$$\boldsymbol{z} = \lambda \widehat{\boldsymbol{J}}^T \boldsymbol{e} + \dot{\boldsymbol{q}} \tag{13}$$

where λ is a scalar positive gain coefficient chosen in such a way that the unit of $\boldsymbol{z}$ is radians/second. Let us note that vector mapping [Eq. (13)] transforms task coordinates $\boldsymbol{e} \in \mathbf{R}^m$ into joint space coordinates $\boldsymbol{z} \in \mathbf{R}^n$. In the sequel, we will use the structure of Eq. (13) to design the control strategy which ensures that the task error $\boldsymbol{e}$ and the sliding vector $\boldsymbol{z}$ converge to zero. Consequently, it will follow from Eq. (13) that $\dot{\boldsymbol{q}}$ too tends to approach zero. It means that the manipulator movement will pursue the strategy of redundancy resolution at the velocity level. The adaptivity of vector $\boldsymbol{z}$ is understood in the sense that the parameters of the adaptive Jacobian matrix will be updated by a parameter update law, defined later. Differentiating Eq. (13) with respect to time yields

$$\dot{\boldsymbol{z}} = \lambda \dot{\widehat{\boldsymbol{J}}}^T \boldsymbol{e} + \lambda \widehat{\boldsymbol{J}}^T \dot{\boldsymbol{e}} + \ddot{\boldsymbol{q}}. \tag{14}$$

https://doi.org/10.1017/S0263574706003018 Published online by Cambridge University Press

Based on Eqs. (1) and (13)–(14), we can easily express the robot dynamic equations by means of vector z and its time derivative, as

$$\boldsymbol{M}(\boldsymbol{q})\dot{z} + \boldsymbol{C}(\boldsymbol{q}, \dot{\boldsymbol{q}})z + \boldsymbol{M}(\boldsymbol{q})\ddot{\boldsymbol{q}}_r + \boldsymbol{C}(\boldsymbol{q}, \dot{\boldsymbol{q}})\dot{\boldsymbol{q}}_r + \boldsymbol{g}(\boldsymbol{q}) = \boldsymbol{u} \quad (15)$$

where $\ddot{\boldsymbol{q}}_r = (\ddot{q}_{r,1}, \ldots, \ddot{q}_{r,n})^T = -\lambda \widehat{\boldsymbol{J}}^T \boldsymbol{e} - \lambda \dot{\widehat{\boldsymbol{J}}}^T \boldsymbol{e}$ and $\dot{\boldsymbol{q}}_r = (\dot{q}_{r,1}, \ldots, \dot{q}_{r,n})^T = -\lambda \widehat{\boldsymbol{J}}^T \boldsymbol{e}$. Moreover, we know from Eq. (3) that the last three terms of Eq. (15) become linear with respect to vector $\boldsymbol{X}$, and hence they can be written as

$$\boldsymbol{M}(\boldsymbol{q})\ddot{\boldsymbol{q}}_r + \boldsymbol{C}(\boldsymbol{q}, \dot{\boldsymbol{q}})\dot{\boldsymbol{q}}_r + \boldsymbol{g}(\boldsymbol{q}) = \boldsymbol{D}(\boldsymbol{q}, \dot{\boldsymbol{q}}, \dot{\boldsymbol{q}}_r, \ddot{\boldsymbol{q}}_r)\boldsymbol{X}. \quad (16)$$

Inserting the right-hand side of Eq. (16) into Eq. (15), it takes the following form:

$$\boldsymbol{M}(\boldsymbol{q})\dot{z} + \boldsymbol{C}(\boldsymbol{q}, \dot{\boldsymbol{q}})z + \boldsymbol{D}(\boldsymbol{q}, \dot{\boldsymbol{q}}, \dot{\boldsymbol{q}}_r, \ddot{\boldsymbol{q}}_r)\boldsymbol{X} = \boldsymbol{u}. \quad (17)$$

Based on dependencies [Eqs. (13) and (17)], and the subsequent stability analysis, we design an adaptive controller as

$$\boldsymbol{u} = -kz - k_P \widehat{\boldsymbol{J}}^T \boldsymbol{e} + \boldsymbol{D}(\boldsymbol{q}, \dot{\boldsymbol{q}}, \dot{\boldsymbol{q}}_r, \ddot{\boldsymbol{q}}_r)\widehat{\boldsymbol{X}} \quad (18)$$

where $\widehat{\boldsymbol{X}}$ is the estimated physical parameter vector defined later; k and k_P are some positive scalars, which could be replaced by diagonal matrices of positive constants without affecting the stability results obtained further on (this should lead to improved performance). Here, scalar constants are chosen for simplicity of presentation. Let us note that the first two terms of the control scheme in Eq. (18) present a generalization to the adaptive case of a well-known transpose Jacobian controller, and the last term on the right-hand side of Eq. (18) is an estimated control including estimations of unknown robot dynamic parameters. The aim of control (18) is to decrease the task error $\boldsymbol{e}$, i.e., to attain geometric path $\boldsymbol{\theta}(s)$ for arbitrary s by the end-effector. However, in order to simultaneously follow the whole path $\boldsymbol{\theta}(\cdot)$ with $\boldsymbol{e}$ tending to approach zero, the path parameter s should vary (increase) as a function of time from 0 to 1. Based on the task error $\boldsymbol{e}$, geometric path $\boldsymbol{\theta}(\cdot)$, and subsequent stability analysis, we design the time parameterization of s as

$$\ddot{s} = -k_s \dot{s} - k_P \left(\left\langle \boldsymbol{e}, -\frac{\mathrm{d}\boldsymbol{\theta}(s)}{\mathrm{d}s} \right\rangle + \gamma(s-1) + \frac{1}{2}\frac{\mathrm{d}\gamma}{\mathrm{d}s}(s-1)^2 \right) \quad (19)$$

where k_s is a positive coefficient, γ is assumed to be a strictly positive function ($\inf\{\gamma\} > 0$) of s with bounded first and second derivatives for any $|s| \leq 1 + \sqrt{\frac{\gamma_0}{\inf\{\gamma\}}}$, where $\gamma_0 = \gamma(0)$ (as will be seen further on, γ_0 determines the upper bound on the accuracy of the path following and may be specified by the user). The choice of function γ is crucial for computational effectiveness of the control scheme given in Eqs. (18) and (19). One possibility is $\gamma = \gamma(s)$ with $\frac{\mathrm{d}\gamma}{\mathrm{d}s} > 0$. An alternative choice could be $\gamma = \gamma(\boldsymbol{e}^2)$, where γ attains its maximum for $\boldsymbol{e} = \boldsymbol{0}$ and smoothly decreases as $\|\boldsymbol{e}\|$ increases. For simplicity of further considerations, we take the first form of γ. Let us note that for a task error equal to zero, $s = 1$ is a stationary point of differential Eq. (19).

Assumption 1. Function γ is not required to satisfy differential equation $\gamma + \frac{1}{2}\frac{\mathrm{d}\gamma}{\mathrm{d}s}(s-1) = 0$.

As will be seen further on, Assumption 1 results in asymptotic convergence of s to 1. Estimated kinematic parameters $\widehat{\boldsymbol{Y}}$ of the adaptive Jacobian matrix $\widehat{\boldsymbol{J}} = \boldsymbol{J}(\boldsymbol{q}, \widehat{\boldsymbol{Y}})$ are updated according to the following law:

$$\dot{\widehat{\boldsymbol{Y}}} = w_k k_P \boldsymbol{K}^T(\boldsymbol{q}, \dot{\boldsymbol{q}})\boldsymbol{e} \quad (20)$$

and the estimated physical parameters $\widehat{\boldsymbol{X}}$ of the dynamic equations are updated by

$$\dot{\widehat{\boldsymbol{X}}} = -w_d \boldsymbol{D}^T(\boldsymbol{q}, \dot{\boldsymbol{q}}, \dot{\boldsymbol{q}}_r, \ddot{\boldsymbol{q}}_r)z \quad (21)$$

where w_k and w_d are the positive gains (scalars) that could be replaced by diagonal matrices of positive constants without affecting the stability of the controller in Eq. (18). Although some kinematic parameters appear in $\widehat{\boldsymbol{X}}$, we should adapt on them separately in $\widehat{\boldsymbol{Y}}$ to preserve linearity. The estimated kinematic parameters $\widehat{\boldsymbol{Y}}$ (updated according to the rule given in Eq. (20)) are then used to compute approximate Jacobian $\boldsymbol{J}(\boldsymbol{q}, \widehat{\boldsymbol{Y}})$ and its time derivative $\dot{\widehat{\boldsymbol{J}}} = \dot{\boldsymbol{J}}(\boldsymbol{q}, \widehat{\boldsymbol{Y}})$ using the right-hand side of Eq. (20), which is only a mapping of $\boldsymbol{q}$, $\boldsymbol{e}$ and $\dot{\boldsymbol{q}}$. Having obtained the adaptive Jacobian matrix and its time derivative, we determine quantities $\dot{\boldsymbol{q}}_r$ and $\ddot{\boldsymbol{q}}_r$. It is worth noticing that their computation does not require any pseudoinverse of matrix $\widehat{\boldsymbol{J}}$, which results in the numerical stability of the controller given in Eq. (18). Finally, based on $\boldsymbol{q}$, $\dot{\boldsymbol{q}}$, $\dot{\boldsymbol{q}}_r$, and $\ddot{\boldsymbol{q}}_r$, we may determine the dynamic regressor matrix $\boldsymbol{D}(\boldsymbol{q}, \dot{\boldsymbol{q}}, \dot{\boldsymbol{q}}_r, \ddot{\boldsymbol{q}}_r)$, which is then used to update the estimated physical parameter vector $\widehat{\boldsymbol{X}}$. Let us note, that the setpoint controllers proposed in refs. 36–38, which are somewhat computationally simpler, cannot be applied to our task. The reason is that the error of approximation in refs. 36–38 is bounded by a constant, and the approximate Jacobian matrix does not include parameters to be adapted. Due to the adaptation law given in Eq. (20), one cannot guarantee in our task the satisfaction of assumption regarding the approximation error made in refs. 36–38.

The closed-loop error dynamics is obtained by inserting the right-hand side of Eq. (18) into Eq. (17) as

$$\begin{aligned}
\boldsymbol{M}(\boldsymbol{q})\dot{z} &= -\boldsymbol{C}(\boldsymbol{q}, \dot{\boldsymbol{q}})z + \boldsymbol{D}(\boldsymbol{q}, \dot{\boldsymbol{q}}, \dot{\boldsymbol{q}}_r, \ddot{\boldsymbol{q}}_r)\widetilde{\boldsymbol{X}} - kz - k_P \widehat{\boldsymbol{J}}^T \boldsymbol{e} \\
\dot{\boldsymbol{e}} &= \boldsymbol{J}(\boldsymbol{q}, \boldsymbol{Y})\dot{\boldsymbol{q}} - \frac{\mathrm{d}\boldsymbol{\theta}(s)}{\mathrm{d}s}\dot{e}_{m+1} \\
\ddot{e}_{m+1} &= -k_s \dot{e}_{m+1} - k_P \left(\left\langle \boldsymbol{e}, -\frac{\mathrm{d}\boldsymbol{\theta}(s)}{\mathrm{d}s} \right\rangle + \gamma e_{m+1} + \frac{1}{2}\frac{\mathrm{d}\gamma}{\mathrm{d}s} e_{m+1}^2 \right) \\
\dot{\widetilde{\boldsymbol{Y}}} &= w_k k_P \boldsymbol{K}^T(\boldsymbol{q}, \dot{\boldsymbol{q}})\boldsymbol{e} \\
\dot{\widetilde{\boldsymbol{X}}} &= -w_d \boldsymbol{D}^T(\boldsymbol{q}, \dot{\boldsymbol{q}}, \dot{\boldsymbol{q}}_r, \ddot{\boldsymbol{q}}_r)z
\end{aligned} \quad (22)$$

where $\widetilde{\boldsymbol{Y}} = \widehat{\boldsymbol{Y}} - \boldsymbol{Y}$ and $\widetilde{\boldsymbol{X}} = \widehat{\boldsymbol{X}} - \boldsymbol{X}$. Applying the Lyapunov stability theory, we derive the following result.

Theorem 1. *If there exists a solution to the problem given in Eqs.* (1) *and* (4)–(8)*, the adaptive Jacobian matrix* $\widehat{\boldsymbol{J}}$ *is*

https://doi.org/10.1017/S0263574706003018 Published online by Cambridge University Press

nonsingular along the end-effector path [*Eq.* (4)]*, and the function* γ *fulfils Assumption 1, then the control scheme given in Eq.* (18) *generates manipulator trajectory whose limit point* $(\dot{\boldsymbol{q}}(\infty),\ \dot{e}_{m+1}(\infty),\ \boldsymbol{e}(\infty),\ e_{m+1}(\infty)) = (\boldsymbol{0},\ 0,\ \boldsymbol{0},\ 0)$*, i.e., satisfying state constraints given in Eqs.* (5)–(8)*, is asymptotically stable.*

Proof. Consider a Lyapunov function candidate

$$V = \frac{1}{2}\langle \boldsymbol{M}z, z\rangle + \frac{1}{2w_d}\widetilde{\boldsymbol{X}}^2 + \frac{1}{2w_k}\widetilde{\boldsymbol{Y}}^2 + \frac{1}{2}\dot{e}_{m+1}^2 + \frac{1}{2}k_P\gamma e_{m+1}^2 + \frac{1}{2}k_P\boldsymbol{e}^2. \tag{23}$$

The time derivative of V is given by

$$\dot{V} = \langle z, \boldsymbol{M}\dot{z}\rangle + \left\langle \frac{\dot{\boldsymbol{M}}}{2}z, z\right\rangle + k_P\langle \boldsymbol{J}^T\boldsymbol{e}, \dot{\boldsymbol{q}}\rangle + \frac{1}{w_d}\langle \widetilde{\boldsymbol{X}}, \dot{\widetilde{\boldsymbol{X}}}\rangle + \frac{1}{w_k}\langle \widetilde{\boldsymbol{Y}}, \dot{\widetilde{\boldsymbol{Y}}}\rangle + \dot{e}_{m+1}\ddot{e}_{m+1} + k_P\left\langle \boldsymbol{e}, -\frac{\mathrm{d}\boldsymbol{\theta}}{\mathrm{d}s}\right\rangle\dot{e}_{m+1} + k_P\gamma e_{m+1}\dot{e}_{m+1} + k_P\frac{1}{2}\frac{\mathrm{d}\gamma}{\mathrm{d}s}e_{m+1}^2\dot{e}_{m+1}.$$

Substituting $\boldsymbol{M}\dot{z}$, $\dot{\boldsymbol{e}}$, $\ddot{e}_{m+1}$, $\dot{\widetilde{\boldsymbol{Y}}}$ and $\dot{\widetilde{\boldsymbol{X}}}$ from $\dot{V}$ for the right-hand sides of closed-loop error dynamics given in Eq. (22) and using the skew-symmetric property of matrix $\frac{\dot{\boldsymbol{M}}}{2} - \boldsymbol{C}$ [property (2)], after simple calculations, we obtain

$$\dot{V} = -k\langle z, z\rangle - k_P\langle \widehat{\boldsymbol{J}}\widehat{\boldsymbol{J}}^T\boldsymbol{e}, \boldsymbol{e}\rangle - k_s\dot{e}_{m+1}^2.$$

Since $\dot{V} \le 0$, function V is bounded. Therefore, z, $\widetilde{\boldsymbol{X}}$, and $\widetilde{\boldsymbol{Y}}$ are bounded vectors. This implies that $\widehat{\boldsymbol{X}}$ and $\widehat{\boldsymbol{Y}}$ are bounded, too. Consequently, it follows from Eq. (13) that $\dot{\boldsymbol{q}}$ is also bounded. Moreover, s and $\dot{s}$ are bounded, too. As can be seen, $\dot{V}$ is negative for all $(z,\ \boldsymbol{e},\ \dot{e}_{m+1}) \neq 0$, and is zero only when $(z,\ \boldsymbol{e},\ \dot{e}_{m+1}) = 0$, which implies (using LaSalle–Yoshizawa invariant theorem[40]) that $(z,\ \boldsymbol{e},\ \dot{e}_{m+1})$ tend asymptotically to zero, i.e. $z(T) \to \boldsymbol{0}$, $\boldsymbol{e}(T) \to \boldsymbol{0}$ and $\dot{e}_{m+1}(T) \to 0$, as $T \to \infty$. By differentiating $\ddot{e}_{m+1}$ in Eq. (22) with respect to time, it is also easy to see that $\frac{\mathrm{d}^3 e_{m+1}}{\mathrm{d}t^3}$ is a function bounded by assumptions regarding $\boldsymbol{\theta}$ and γ. This means that $\ddot{e}_{m+1}$ is uniformly continuous. Hence, $\ddot{e}_{m+1}(T) \to 0$, as $T \to \infty$, too. The convergence of path velocity and acceleration yields

$$\gamma e_{m+1}(\infty) + \frac{1}{2}\frac{\mathrm{d}\gamma}{\mathrm{d}s}e_{m+1}^2(\infty) = 0.$$

Consequently, $e_{m+1}(\infty) = 0$ or $\gamma + \frac{1}{2}\frac{\mathrm{d}\gamma}{\mathrm{d}s}e_{m+1}(\infty) = 0$. On account of Assumption 1, the second equality is not fulfilled. Thus, $e_{m+1}(\infty) = 0$ (or equivalently $s(\infty) = 1$). On account of Eq. (13), $\dot{\boldsymbol{q}}(T) \to \boldsymbol{0}$, as $T \to \infty$, too. Consequently, the boundary conditions given in Eqs. (5)–(8) are (asymptotically) fulfilled, and the limit point $(\dot{\boldsymbol{q}}(\infty),\ \dot{e}_{m+1}(\infty),\ \boldsymbol{e}(\infty),\ e_{m+1}(\infty)) = (\boldsymbol{0},\ 0,\ \boldsymbol{0},\ 0)$ is asymptotically stable. Finally, it should be emphasized that the chosen Lyapunov function does not guarantee the convergence of parameter estimations $\widehat{\boldsymbol{X}}$ and $\widehat{\boldsymbol{Y}}$ to their true values. □

On account of Eqs. (5)–(8), we have $V_{t=0} = \frac{\widetilde{\boldsymbol{X}}_{t=0}^2}{2w_d} + \frac{\widetilde{\boldsymbol{Y}}_{t=0}^2}{2w_k} + \frac{k_P\gamma_0}{2}$. For sufficiently large w_d and w_k, the first two terms in this equality may be omitted. Hence, we obtain $V_{t=0} \simeq \frac{k_P\gamma_0}{2}$. Since $\dot{V}$ is not positive, function V fulfils the inequality $V \le \frac{k_P\gamma_0}{2}$. Consequently, the following bound on $\|\boldsymbol{e}\|$ may easily be obtained, based on Eq. (23) and the last dependence

$$\|\boldsymbol{e}\| \le \sqrt{\gamma_0}. \tag{24}$$

An important remark may be derived from the proof given earlier. The inequality given in Eq. (24) presents an upper bound (path independent) on the accuracy of the path following by the end-effector according to the control law [Eq. (18)]. Let us note that the estimation of the upper bound on path-following error given in Eq. (24) is very conservative. Consequently, the control gains w_d and w_k do not require large values to achieve a good path-following accuracy, as the numerical simulations (given in the next section) show. Moreover, several observations can be made regarding the control strategy [Eq. (18)]. First note that the proposed control law requires no information concerning the robot kinematic and dynamic equations. Second, the choice of controller parameters k, k_P, k_s, w_d, and w_k according to the dependencies given in Eqs. (18)–(21) guarantees asymptotic stability of the closed-loop error dynamics [Eq. (22)] during the manipulator movement. Moreover, the transpose of $\widehat{\boldsymbol{J}}$ (instead of a pseudoinverse) in the control scheme given in Eq. (18) does not result in numerical instabilities due to (possible) kinematic singularities met on the robot trajectory. Nevertheless, Eq. (18) has been derived under the assumption of a full-rank adaptive Jacobian matrix along the path. Furthermore, controller given in Eq. (18) does not require the knowledge of task space velocity. Due to conservative estimation of the path-following accuracy, the control algorithm given in Eq. (18) results in a better accuracy of path following as compared to the upper bound given in Eq. (24), as the numerical computations carried out in the next section show. In order to prevent control (torque) oscillations at the very beginning of time histories (caused by, e.g., the nonzero initial path-following error) k_s from Eq. (19) should be a bounded, quickly decreasing time-dependent function as $t \to \infty$ (see Section 4).

Due to the real-time nature of the robot controller [Eq. (18)], we shall try to estimate the number of arithmetic operations required to implement the algorithm presented in this section. The dimension of the robot task space is assumed to be constant in estimation. Operations required for the computation of sine, cosine, and $\boldsymbol{\theta}(\cdot)$ functions are not taken into account. Furthermore, matrices $\boldsymbol{J}(\boldsymbol{q}, \widehat{\boldsymbol{Y}})$, $\boldsymbol{K}(\boldsymbol{q},\ \dot{\boldsymbol{q}})$, and $\boldsymbol{D}(\boldsymbol{q}, \dot{\boldsymbol{q}}, \dot{\boldsymbol{q}}_r, \ddot{\boldsymbol{q}}_r)$ are assumed to be given in estimation. Moreover, estimations are carried out at any time instant of the robot task accomplishment. It follows from Eqs. (13) and (18) that terms kz, $k_P\widehat{\boldsymbol{J}}^T\boldsymbol{e}$ require $O(n)$ operations. Computation of the right-hand sides of Eqs. (19) and (20) involves $O(1)$ and $O(n)$ operations, respectively, assuming that $k = O(n)$. Computational complexity for the right-hand side of Eq. (21) equals $O(n^2)$ by the assumption that $d = O(n)$. Computation of the estimated control $\boldsymbol{D}(\boldsymbol{q}, \dot{\boldsymbol{q}}, \dot{\boldsymbol{q}}_r, \ddot{\boldsymbol{q}}_r)\widehat{\boldsymbol{X}}$ also requires the same order of complexity, i.e., $O(n^2)$ operations. Finally, the computational

https://doi.org/10.1017/S0263574706003018 Published online by Cambridge University Press

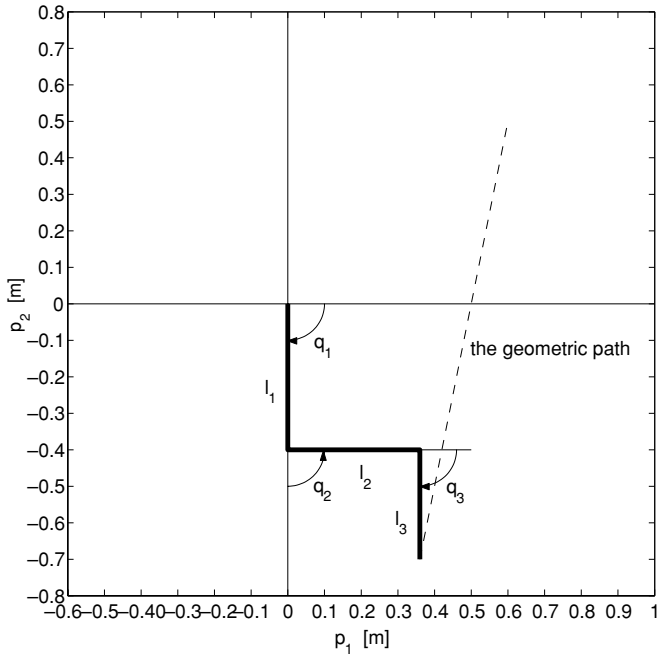


Fig. 1. A kinematic scheme of the manipulator and the task to be accomplished.

complexity of the whole robot controller [Eq. (18)] is of the order $O(n^2)$.

4. A Numerical Example

In this section, we illustrate the performance of the proposed adaptive control algorithm using a dynamic three-joint direct-drive arm ($n = 3$) of a SCARA-type robotic manipulator whose kinematic scheme is shown in Fig. 1. In the simulations, SI units are used. The link masses of links 1–3 were selected to be $m_1 = 3.6$, $m_2 = 2.6$, and $m_3 = 2.0$. The corresponding link lengths were selected as $l_1 = 0.4$, $l_2 = 0.36$, $l_3 = 0.3$, respectively, and the center of mass of each link was selected to be the midpoint of each link. The components of dynamic equations of this manipulator are as follows[39]:

$$\boldsymbol{M} = \begin{bmatrix} X_1 + 2X_4c2 + X_5c23 + X_6c3 & X_2 + X_4c2 + X_5c23 & X_3 \\ X_2 + X_4c2 + X_5c23 & X_2 + 2X_6c3 & X_3 + X_6c3 \\ X_3 & X_3 + X_6c3 & X_3 \end{bmatrix}$$

$$\begin{aligned} \boldsymbol{C} &= [C_{ij}]_{1\le i,\ j\le 3} \\ C_{11} &= -(X_4s2 + X_5s12)\dot{q}_2 - (X_5s12 + X_6s12)\dot{q}_3 \\ C_{12} &= -(X_4s2 + X_5s12)(\dot{q}_1 + \dot{q}_2) - (X_5s12 + X_6s12)\dot{q}_3 \\ C_{13} &= (X_5s12 + X_6s12)(-\dot{q}_1 + \dot{q}_2 + \dot{q}_3) \\ C_{21} &= (X_4s2 + X_5s12)\dot{q}_1 + X_6s3\dot{q}_3 \\ C_{22} &= -(X_5s12 + X_6s12)\dot{q}_3 \\ C_{23} &= -X_6s3(3\dot{q}_1 + \dot{q}_2 + \dot{q}_3) \\ C_{31} &= (X_4s2 + X_5s12)\dot{q}_1 - X_6s3\dot{q}_2 \\ C_{32} &= X_6s3(\dot{q}_1 + \dot{q}_2) \\ C_{33} &= 0 \end{aligned} \tag{25}$$

$$\boldsymbol{g} = g \begin{pmatrix} X_7c1 + X_8c12 + X_9c123 \\ X_8c12 + X_9c123 \\ X_9c123 \end{pmatrix}$$

where $ci = \cos(q_i)$, $si = \sin(q_i)$, $cij = \cos(q_i + q_j)$, $sij = \sin(q_i + q_j)$, $cijk = \cos(q_i + q_j + q_k)$, $sijk = \sin(q_i + q_j + q_k)$, g is the gravity acceleration and parameters X_i, $i = 1 : 9$ take the following nominal values:

$$\begin{aligned} X_1 &= 1.1956 \\ X_2 &= 0.3946 \\ X_3 &= 0.0512 \\ X_4 &= 0.4752 \\ X_5 &= 0.1280 \\ X_6 &= 0.1152 \\ X_7 &= (m_1p_1 + m_2l_1 + m_3l_1)g = 15.116 \\ X_8 &= (m_2p_2 + m_3l_2)g = 11.6543 \\ X_9 &= m_3p_3g = 2.9430 \end{aligned}$$

where $p_i = l_i/2$ denotes the location of the mass center in the ith coordinate system of the ith link. The Jacobian matrix $\boldsymbol{J}(\boldsymbol{q},\ \boldsymbol{Y})$ equals

$$\begin{aligned} &\boldsymbol{J}(\boldsymbol{q}, \boldsymbol{Y}) \\ &= \begin{bmatrix} -Y_1s1 - Y_2s12 - Y_3s123 & -Y_2s12 - Y_3s123 & -Y_3s123 \\ Y_1c1 + Y_2c12 + Y_3c123 & Y_2c12 + Y_3c123 & Y_3c123 \end{bmatrix} \end{aligned} \tag{26}$$

where $Y_1 = l_1 = 0.4$, $Y_2 = l_2 = 0.36$, $Y_3 = l_2 = 0.3$, and the kinematic regressor matrix takes the following form:

$$\boldsymbol{K}(\boldsymbol{q},\ \dot{\boldsymbol{q}}) = \begin{bmatrix} -s1\dot{q}_1 & -s12(\dot{q}_1 + \dot{q}_2) & -s123(\dot{q}_1 + \dot{q}_2 + \dot{q}_3) \\ c1\dot{q}_1 & c12(\dot{q}_1 + \dot{q}_2) & c123(\dot{q}_1 + \dot{q}_2 + \dot{q}_3) \end{bmatrix}. \tag{27}$$

Hence, in the presence of kinematic uncertainties, the approximate Jacobian matrix is equal to

$$\begin{aligned} &\boldsymbol{J}(\boldsymbol{q}, \widehat{\boldsymbol{Y}}) \\ &= \begin{bmatrix} -\widehat{Y}_1s1 - \widehat{Y}_2s12 - \widehat{Y}_3s123 & -\widehat{Y}_2s12 - \widehat{Y}_3s123 & -\widehat{Y}_3s123 \\ \widehat{Y}_1c1 + \widehat{Y}_2c12 + \widehat{Y}_3c123 & \widehat{Y}_2c12 + \widehat{Y}_3c123 & \widehat{Y}_3c123 \end{bmatrix}. \end{aligned} \tag{28}$$

The end-effector position $\boldsymbol{p} = (p_1\ \ p_2)^T$ (see Fig. 1) represents the task space coordinates ($m = 2$) in the simulations. The upper bound on the accuracy of the path following in all the computer simulations is assumed to be equal to $\sqrt{\gamma_0} = 6 \times 10^{-2}$, where $\gamma(s) = 2 \times 10^{-3} + 2 \times 10^{-3}e^{2.7s}$. Let us introduce the path-following errors

$$\begin{pmatrix} e_1 \\ e_2 \\ e_3 \end{pmatrix} = \begin{pmatrix} p_1 - \theta_1(s) \\ p_2 - \theta_2(s) \\ s - 1 \end{pmatrix} \tag{29}$$

to evaluate the performance of the robot controller [Eq. (18)]. Based on Eqs. (25), (28), and (29), the dynamic regressor

https://doi.org/10.1017/S0263574706003018 Published online by Cambridge University Press

matrix takes the following form:

$$\boldsymbol{D} = \begin{bmatrix} D_{11} & D_{12} & D_{13} & D_{14} & D_{15} & D_{16} & D_{17} & D_{18} & D_{19} \\ D_{21} & D_{22} & D_{23} & D_{24} & D_{25} & D_{26} & D_{27} & D_{28} & D_{29} \\ D_{31} & D_{32} & D_{33} & D_{34} & D_{35} & D_{36} & D_{37} & D_{38} & D_{39} \end{bmatrix} \quad (30)$$

where $D_{11} = \ddot{q}_{r,1}$, $D_{12} = \ddot{q}_{r,2}$, $D_{13} = \ddot{q}_{r,3}$, $D_{14} = \ddot{q}_{r,1}2c2 + \ddot{q}_{r,2}c2 - \dot{q}_{r,1}s2\dot{q}_2 - \dot{q}_{r,2}(\dot{q}_1+\dot{q}_2)s2$, $D_{15} = \ddot{q}_{r,1}c23+\ddot{q}_{r,2}c23 - \dot{q}_{r,1}s12\dot{q}_2 - \dot{q}_{r,1}s12\dot{q}_3 - \dot{q}_{r,2}s12(\dot{q}_1 + \dot{q}_2) - \dot{q}_{r,2}s12\dot{q}_3 + \dot{q}_{r,3}s12(-\dot{q}_1 + \dot{q}_2 + \dot{q}_3)$, $D_{16} = \ddot{q}_{r,1}c3 - \dot{q}_{r,1}s12\dot{q}_3 - \dot{q}_{r,2}s12\dot{q}_3 - \dot{q}_{r,3}s12(-\dot{q}_1 + \dot{q}_2 + \dot{q}_3)$, $D_{17} = c1$, $D_{18} = c12$, $D_{19} = c123$, $D_{21} = 0$, $D_{22} = \ddot{q}_{r,1} + \ddot{q}_{r,2}$, $D_{23} = \ddot{q}_{r,3}$, $D_{24} = \ddot{q}_{r,1}c2 + \dot{q}_{r,1}s2\dot{q}_1$, $D_{25} = \ddot{q}_{r,1}c23 + \ddot{q}_{r,2}2c3 + \dot{q}_{r,1}s12\dot{q}_1 - \dot{q}_{r,2}s12\dot{q}_3$, $D_{26} = \ddot{q}_{r,1}c3+\dot{q}_{r,1}s3\dot{q}_3-\dot{q}_{r,2}s12\dot{q}_3 - \dot{q}_{r,3}s3(3\dot{q}_1 + \dot{q}_2 + \dot{q}_3)$, $D_{27} = 0$, $D_{28} = c12$, $D_{29} = c123$, $D_{31} = 0$, $D_{32} = 0$, $D_{33} = \ddot{q}_{r,1} + \ddot{q}_{r,3}$, $D_{34} = \dot{q}_{r,1}s2\dot{q}_1$, $D_{35} = \dot{q}_{r,1}s12\dot{q}_1$, $D_{36} = \ddot{q}_{r,2}c3 - \dot{q}_{r,1}s3\dot{q}_2 + \dot{q}_{r,2}s3(\dot{q}_1 + \dot{q}_2)$, $D_{37} = 0$, $D_{38} = 0$, $D_{39} = c123$. In order to examine the effects of both kinematic and dynamic uncertainties, the initial values for vectors $\widehat{\boldsymbol{X}}$ and $\widehat{\boldsymbol{Y}}$ were set in both the simulations as $\widehat{\boldsymbol{X}}(0) = (3\ \ 2\ \ 1\ \ 1\ \ 0.8\ \ 0.5\ \ 30\ \ 20\ \ 10)^T$ and $\widehat{\boldsymbol{Y}}(0) = (0.55\ 0.45\ 0.4)^T$.

The task of the robot is to transfer the end-effector along the geometric path (the dotted line in Fig. 1), expressed by the following equations:

$$\begin{aligned} \theta_1(s) &= 0.36 + 0.24s \\ \theta_2(s) &= -0.7 + 1.2s \end{aligned} \quad (31)$$

where, $s \in [0, 1]$. The initial configuration $\boldsymbol{q}_0$ equals $\boldsymbol{q}_0 = (-\pi/2\ \pi/2\ \ -\pi/2)^T$. Parameters $k = 10$, $\lambda = 1$, $k_P = 3 \times 10^3$, and $k_s = 6.9(1 + 100e^{-10t}) + 30$ have been chosen experimentally to achieve practically reasonable time horizon of task performance and relatively small controls with $w_k = 10$ and $\boldsymbol{w}_d = \text{diag}(4.85\ 4.85\ 4.85\ 4.85\ 4.85\ 4.85\ 4.85\ 8.55$

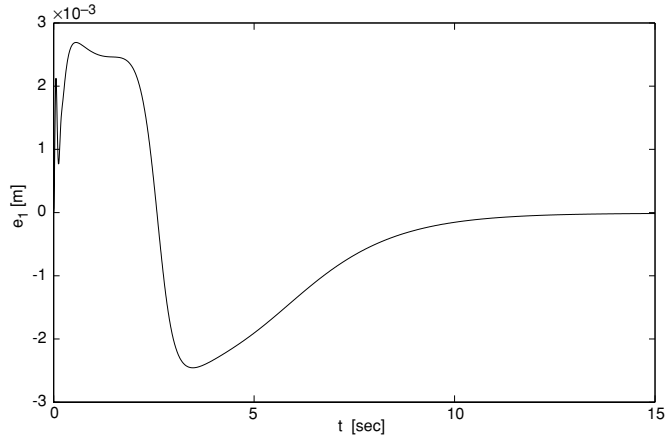


Fig. 2. Path following error e_1.

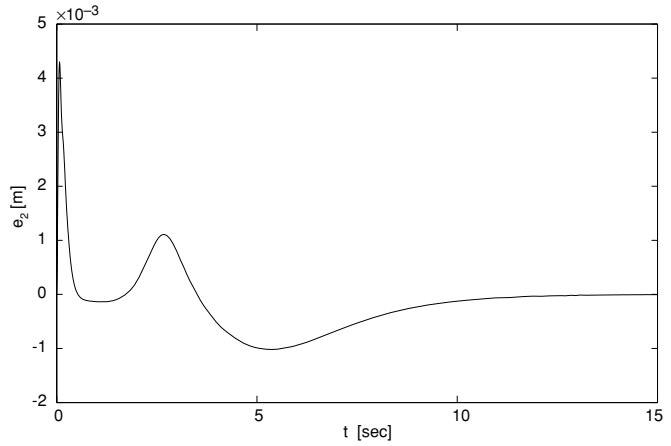


Fig. 3. Path following error e_2.

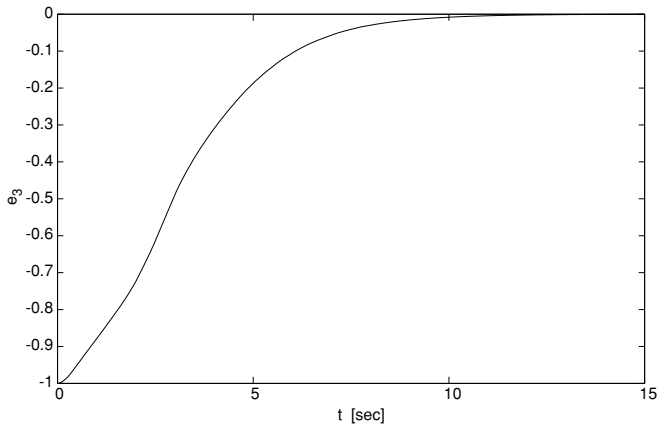


Fig. 4. Path following error e_3.

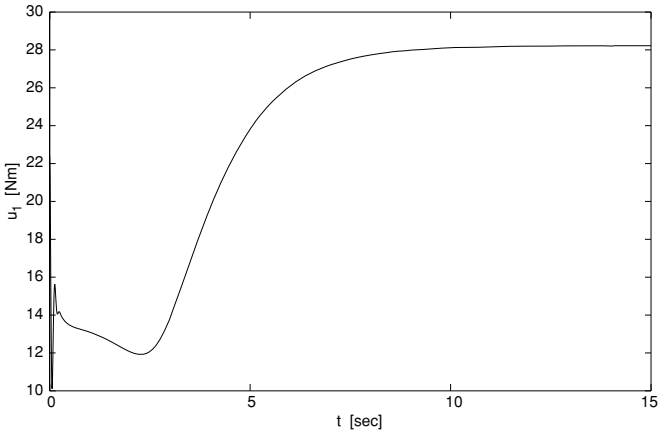


Fig. 5. Input torque u_1 of joint 1.

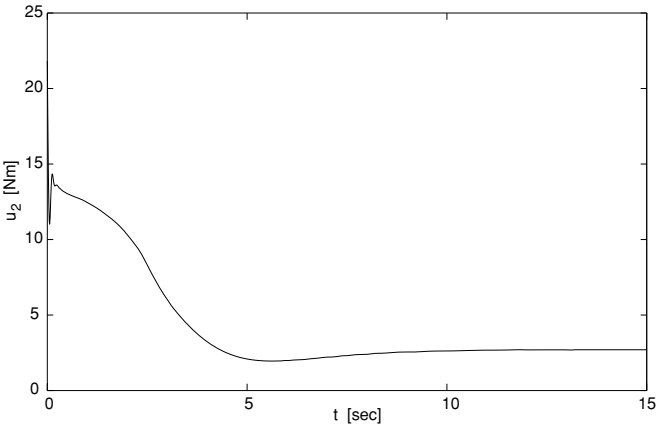


Fig. 6. Input torque u_2 of joint 2.

https://doi.org/10.1017/S0263574706003018 Published online by Cambridge University Press

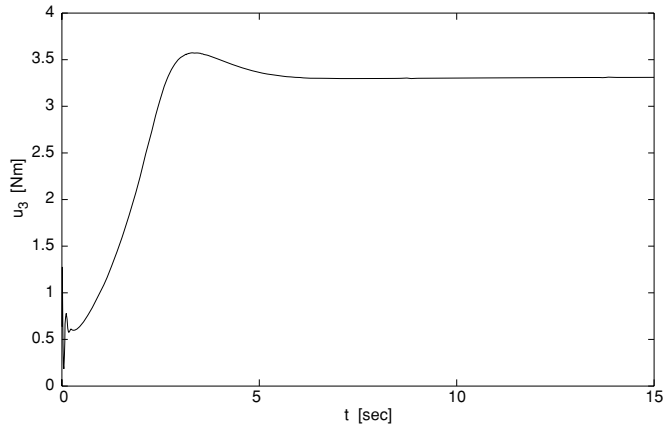


Fig. 7. Input torque u_3 of joint 3.

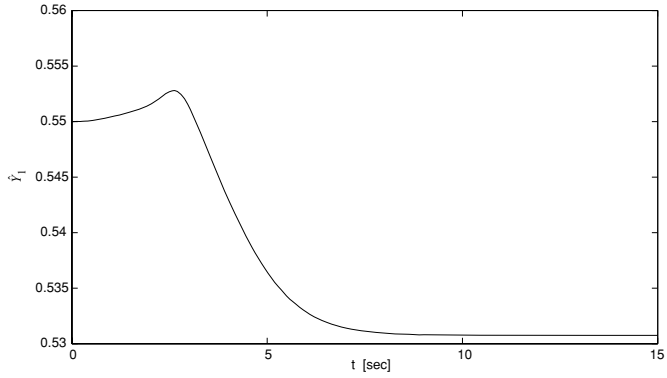


Fig. 8. Time course of estimated kinematic parameter $\widehat{Y}_1$.

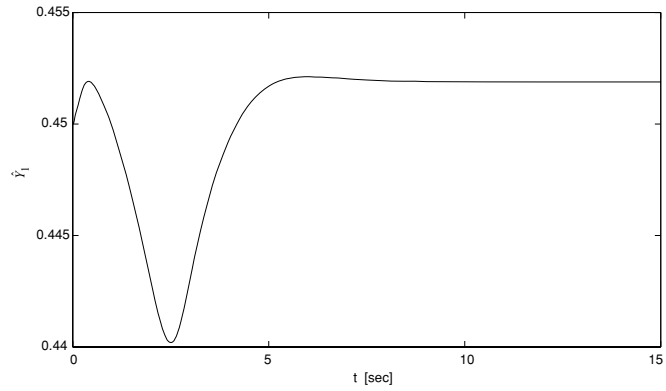


Fig. 9. Time course of estimated kinematic parameter $\widehat{Y}_2$.

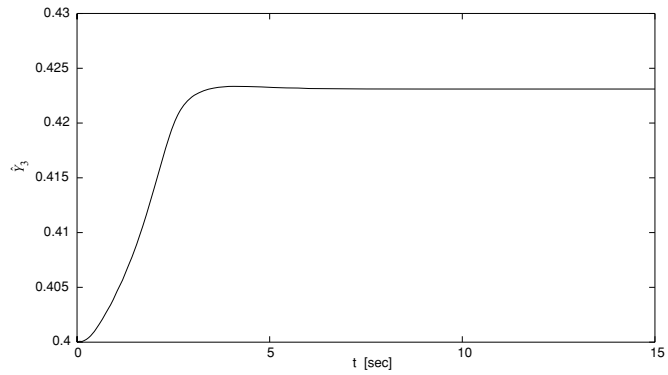


Fig. 10. Time course of estimated kinematic parameter $\widehat{Y}_3$.

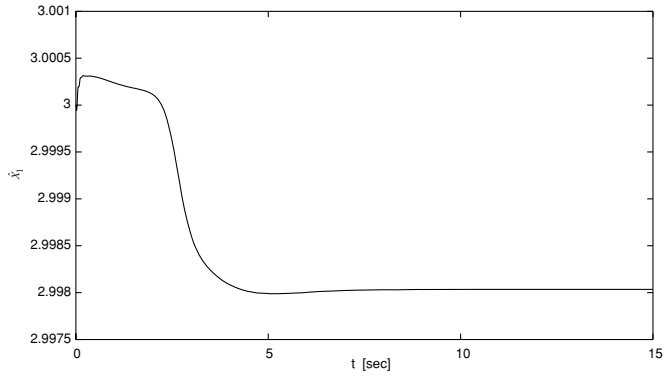


Fig. 11. Time course of estimated dynamic parameter $\widehat{X}_1$.

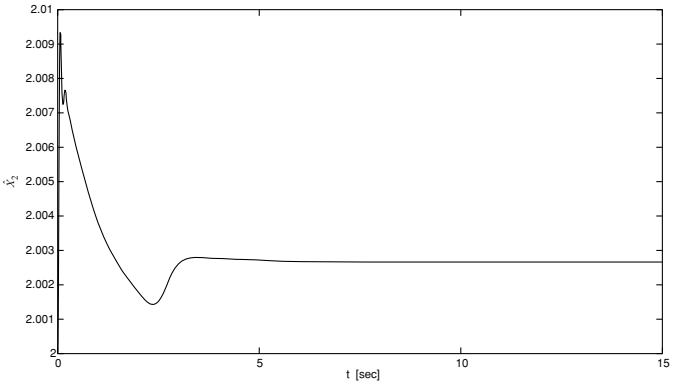


Fig. 12. Time course of estimated dynamic parameter $\widehat{X}_2$.

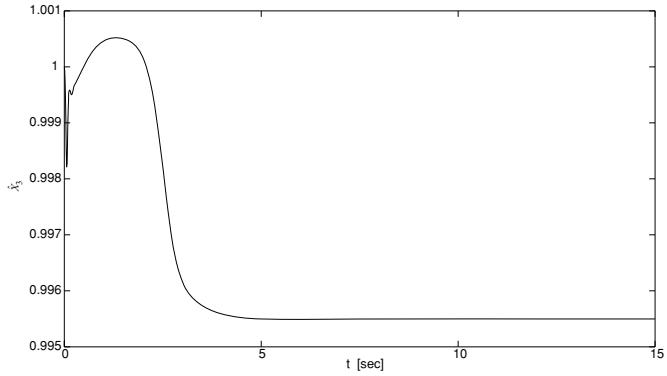


Fig. 13. Time course of estimated dynamic parameter $\widehat{X}_3$.

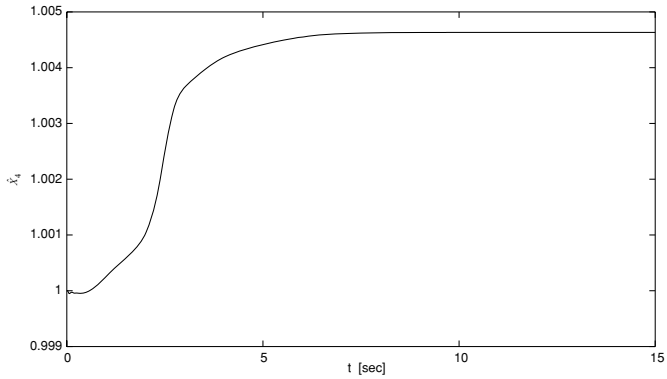


Fig. 14. Time course of estimated dynamic parameter $\widehat{X}_4$.

https://doi.org/10.1017/S0263574706003018 Published online by Cambridge University Press

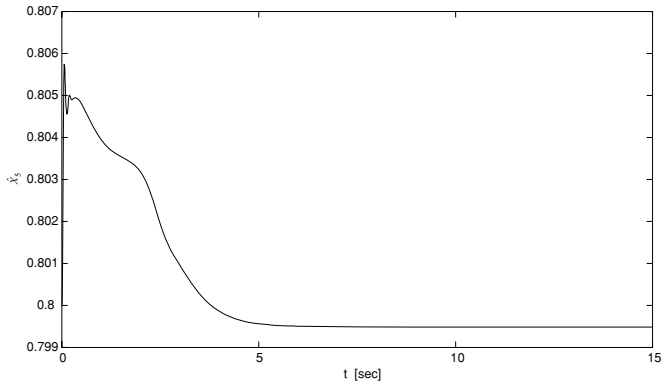


Fig. 15. Time course of estimated dynamic parameter $\widehat{X}_5$.

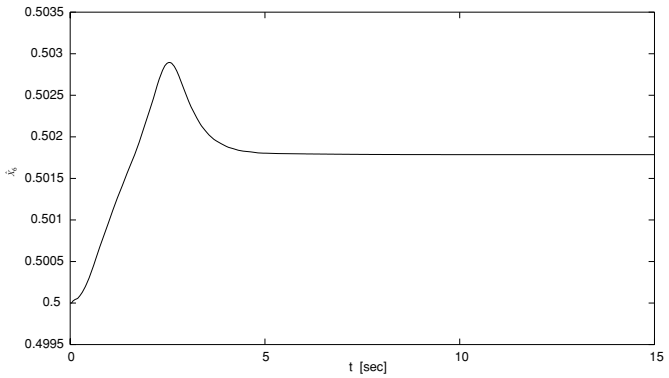


Fig. 16. Time course of estimated dynamic parameter $\widehat{X}_6$.

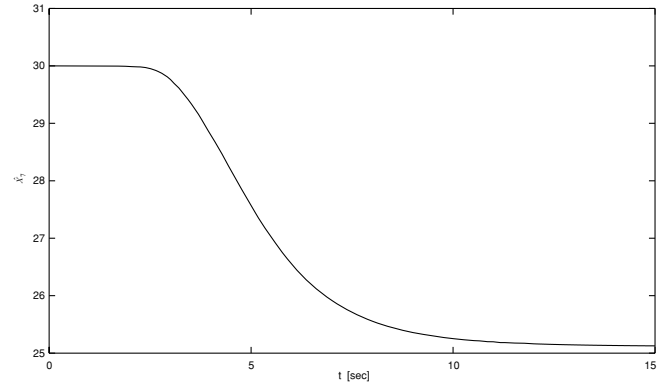


Fig. 17. Time course of estimated dynamic parameter $\widehat{X}_7$.

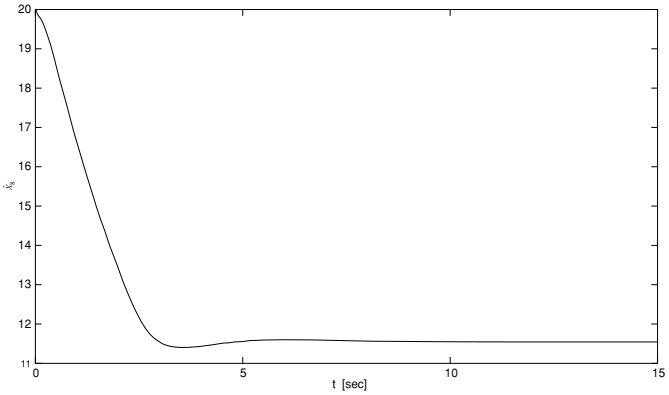


Fig. 18. Time course of estimated dynamic parameter $\widehat{X}_8$.

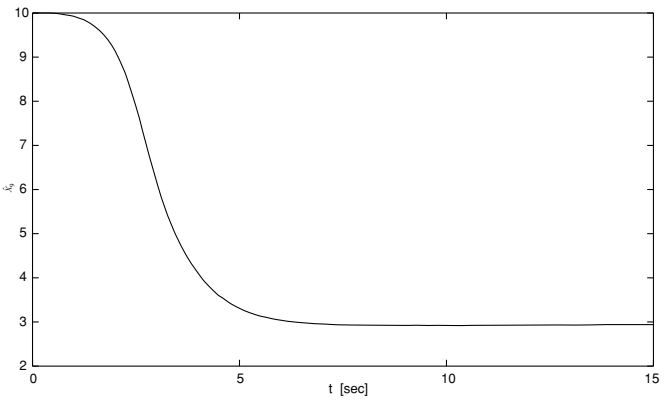


Fig. 19. Time course of estimated dynamic parameter $\widehat{X}_9$.

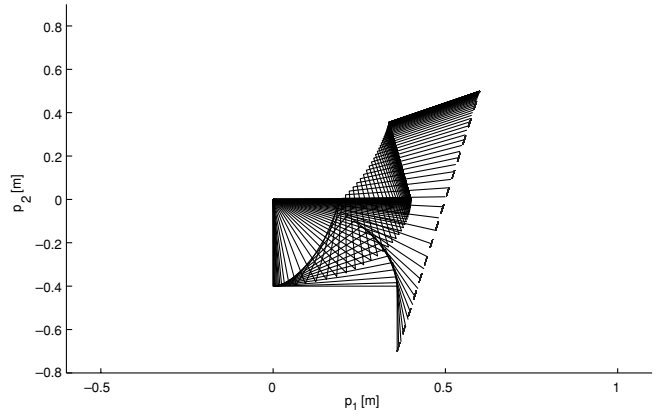


Fig. 20. Manipulator motion along the path.

4.76). The results of computer simulation are presented in Figs. 2–20. As might be expected, the path-following errors from Figs. 2 and 3 are much smaller than those obtained from the conservative dependence given in Eq. (24). Moreover, as one can observe from Figs. 2–7, the time-dependent damping function k_s decreases (eliminates) errors and torques oscillations at the very beginning of the time histories. Furthermore, as seen from Figs. 8–19, estimations $\widehat{\boldsymbol{X}}$ and $\widehat{\boldsymbol{Y}}$ do not converge to their real (nominal) values.

5. Conclusion

This study has presented a nonlinear adaptive robot controller for path following by the end-effector with uncertain kinematics and dynamics. The control generation scheme has been derived using the Lyapunov stability theory. An advantage of the proposed control law given in Eq. (18) is that it requires a minimal amount of knowledge regarding the robot dynamic equations. The control strategy given in Eq. (18) is also shown to be asymptotically stable (by fulfilment of practically reasonable assumptions). The proposed robot controller has been applied to a planar redundant manipulator of three revolute kinematic pairs operating in a 2D task space. Numerical simulations have shown that the results obtained are in accordance with the theoretical analysis. The novelty of the proposed strategy lies in its simplicity in design, program code, and real-time implementation. The approach presented here may also be directly applied to kinematically redundant manipulators operating in a 3D work space including obstacles.

https://doi.org/10.1017/S0263574706003018 Published online by Cambridge University Press

Acknowledgment
This work was supported by the DFG Ga 652/1–1,2.

References

1. J. E. Bobrow, S. Dubowsky and J. S. Gibson, "Time-optimal control of robotic manipulators," *Int. J. Robot. Res.* **4**(3), 3–17 (1985).
2. F. Pfeiffer and R. Johanni, "A concept for manipulator trajectory planning," *IEEE J. Robot. Autom.* **RA-3**(2), 115–123 (1987).
3. Z. Shiller and S. Dubowsky, "Robust computation of path constrained time optimal motions," *Proceedings of the IEEE Conference on Robotics and Automation*, Cincinnati, OH (1990) pp. 144–149.
4. Z. Shiller and H. H. Lu, "Computation of path constrained time optimal motions with dynamic singularities," *ASME J. Dynam. Syst., Measure. Control* **114**(2), 34–40 (1992).
5. Y. Chen and A. A. Desrochers, "Structure of minimum-time control law for robotic manipulators with constrained paths," *Proceedings of the IEEE Conference on Robotics and Automation*, Scottsdale, OH (1989) pp. 971–976.
6. J. M. McCarthy and J. E. Bobrow, "The number of saturated actuators and constraint forces during time-optimal movement of a general robotic system," *IEEE Trans. Robot. Autom.* **8**(3), 407–409 (1992).
7. Z. Shiller, "On singular time-optimal control along specified paths," *IEEE Trans. Robot. Autom.* **10**(4), 561–566 (1994).
8. M. Galicki, "The structure of time optimal controls for kinematically redundant manipulators with end-effector path constraints," *Proceedings of the IEEE Conference on Robotics and Automation*, Leuven, Belgium (1998) pp. 101–106.
9. M. Galicki, "Time-optimal controls of kinematically redundant manipulators with geometric constraints," *IEEE Trans. Robot. Autom.* **16**(1), 89–93 (2000).
10. M. Galicki, "The planning of robotic optimal motions in the presence of obstacles," *Int. J. Robot. Res.* **17**(3), 248–259 (1998).
11. M. Galicki and I. Pajak, "Optimal motion of redundant manipulators with state equality constraints," *Proceeding of the IEEE International Symposium on Assembly and Task Planning*, Porto, Portugal (1999) pp. 181–185.
12. O. Dahl, "Path-constrained robot control with limited torques—Experimental evaluation," *IEEE J. Robot. Autom.* **10**, 658–669 (1994).
13. J. Kiefer, A. J. Cahill and M. R. James, "Robust and accurate time-optimal path-tracking control for robot manipulators," *IEEE Trans. Robot. Autom.* **13**, 880–890 (1997).
14. M. Galicki, "Real-time trajectory generation for redundant manipulators with path constraints," *Int. J. Robot. Res.* **20**(7), 673–690 (2001).
15. M. Galicki, "Path following by the end-effector of a redundant manipulator operating in a dynamic environment," *IEEE Trans. Robot.* **20**(6), 1018–1025 (2004).
16. M. Galicki, "Path-constrained control of a redundant manipulator in a task space," *Robot. Auton. Syst.* **54**, 234–243 (2006).
17. M. Takegaki and S. Arimoto, "A new feedback method for dynamic control of manipulators," *Trans. ASME: J. Dynam. Syst., Measure., Control* **102**, 119–125 (1981).
18. O. Khatib, "A unified approach for motion and force control of robot manipulators: The operational space formulation," *IEEE J. Robot. Autom.* **RA-3**, 43–53 (1987).
19. F. Miyazaki and S. Arimoto, "Sensory feedback for robot manipulators," *J. Robot. Syst.* **2**(1), 53–71 (1985).
20. S. Arimoto, *Control Theory of Non-Linear Mechanical Systems* (Clarendon, Oxford, UK, 1996).
21. C. Canudas de Wit, B. Siciliano and G. Bastin, *Theory of Robot Control* (Springer-Verlag, New York, 1996).
22. L. Sciavicco and B. Siciliano, *Modeling and Control of Robot Manipulators* (McGraw-Hill, New York, 1996).
23. S. Arimoto, "Design of robot control systems," *Adv. Robot.* **4**(1), 79–97 (1990).
24. Y. Masutani, F. Miyazaki and S. Arimoto, "Sensory feedback control for space manipulators," *Proceedings of the IEEE International Conference on Robotics and Automation*, Scottsdale (1989) pp. 1346–1351.
25. R. Kelly, "Regulation of manipulators in generic task space: An energy shaping plus damping injection approach," *IEEE Trans. Robot. Autom.* **15**(2), 381–386 (1999).
26. M. Galicki, "Motion control of robotic manipulators in task space," *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems*, Lausanne (2002) pp. 2061–2066.
27. M. Galicki, "Collision-free control of robotic manipulators in the task space," *J. Robot. Syst.* **22**(8), 439–455 (2005).
28. J. J. E. Slotine and W. Li, "On adaptive control of robot manipulators," *Int. J. Robot. Res.* **6**, 49–59 (1987).
29. J. J. E. Slotine and W. Li, "Adaptive manipulator control: A case study," *IEEE Trans. Autom. Control* **33**(11), 995–1003 (1988).
30. J. J. E. Slotine and W. Li, *Applied Nonlinear Control* (Prentice Hall, Englewood Cliffs, NJ, 1991).
31. G. Feng and M. Palaniswami, "Adaptive control of robot manipulators in task space," *IEEE Trans. Autom. Control* **38**(1), 100–104 (1992).
32. H. Berghuis, R. Ortega and H. Nijmeijer, "A robust adaptive robot controller," *IEEE Trans. Robot. Autom.* **9**(6), 825–830 (1993).
33. F. L. Lewis, C. T. Abdallach and D. M. Dawson, *Control of Robot Manipulators* (Macmillan, New York, 1993).
34. K. Kozlowski, *Modelling and Identification in Robotics* (Springer-Verlag, Berlin, Germany, 1998).
35. P. Tomei, "Robust adaptive friction compensation for tracking control of robot manipulators," *IEEE Trans. Autom. Control* **45**(11), 2164–2169 (2000).
36. C. C. Cheah, S. Kawamora and S. Arimoto, "Feedback control for robotic manipulators with an uncertain Jacobian matrix," *J. Robot. Syst.* **6**(2), 119–134 (1999).
37. H. Yazarel and C. C. Cheach, "Task-space adaptive control of robotic manipulators with uncertainties in gravity regressor matrix and kinematics," *IEEE Trans. Autom. Control* **47**(9), 1580–1585 (2002).
38. C. C. Cheach, M. Hirano, S. Kawamura and S. Arimoto, "Approximate Jacobian control for robots with uncertain kinematics and dynamics," *IEEE Trans. Robot. Autom.* **19**(4), 692–702 (2003).
39. M. Spong and M. Vidyasagar, *Robot Dynamics and Control* (Wiley, New York, 1989).
40. M. Krstic, I. Kanellakopoulos and P. Kokotovic, *Nonlinear and Adaptive Control Design* (Wiley, New York, 1995).

https://doi.org/10.1017/S0263574706003018 Published online by Cambridge University Press