International Journal of Systems Science, 2001, volume 32, number 1, pages 9–23

Model-based and neural-network-based adaptive control of two robotic arms manipulating an object with relative motion

S. S. GE†, L. HUANG‡ and T. H. LEE§

In the study of constrained multiple robot control, the relative motion between the constraint object and the end effectors of manipulators are usually neglected in the literature. However, in many industrial applications, such as assembly and machining, the constraint object is required to move with respect to not only the world coordinates but also the end effectors of the robotic arms. In this paper, dynamic modelling of two robotic arms manipulating an object with relative motion is presented first, then a model-based adaptive controller and a model-free neural network controller are developed. Both controllers guarantee the asymptotic tracking of the constraint object and the boundedness of the constraint force. Asymptotic convergence of the constraint force can also be achieved under certain conditions. Simulation studies are conducted to verify the effectiveness of the approaches.

1. Introduction

A constrained robotic system is normally governed by a set of nonlinear differential–algebraic equations which describe both the positional information and the force developed between the constraint object and the end effectors of the robots. A nonlinear transformation is normally used to transform the original dynamic model into a set of differential equations describing an unconstrained robot motion on the constraint manifold, and an algebraic equation describing the relationship between the force and the system dynamics (McClamroch and Wang 1988). The concept of pseudo-velocity is usually used to reduce the order of the system (Kankaanranta and Kovio 1988). The techniques of both nonlinear transformation and order reduction were further utilized in the control of constrained multiple manipulators (Ramadorai *et al.* 1992, Unseren 1992, Yao *et al.* 1994). Based on these models, many attractive control strategies have been developed, such as nonlinear feedback control (McClamroch and Wang 1988), variable structure control (Yao *et al.* 1994) and adaptive control (Slotine and Li 1989, Yuan 1996), among others.

In the control of constrained multiple manipulators, although the constraint object is moving, it is usually assumed to be held tightly and thus has no relative motion with respect to the end effectors of the manipulators for the ease of analysis. These assumptions are not applicable to some applications which require both the motion of the object and its relative motion with respect to the end effectors of the manipulators. In some machining processes such as deburring, grinding and polishing, the motion of the part with respect to the manipulators can also be utilized to cope with the limited operational space and to increase the efficiency of the work (Ge *et al.* 1997).

In this paper, we shall investigate one particular application where one robotic arm (called manipulator 1) does the assembly or the machining task on the workpiece which is held tightly by another robotic arm (called manipulator 2). Manipulator 2 is to be controlled in such a manner that the constraint object follows the planned motion trajectory, while manipulator 1 is to be controlled such that its end effector follows a planned trajectory on the workpiece with a desired contact force.

The rest of the paper is organized as follows. In § 2, the kinematics and dynamic models of the above system are derived. In § 3, a model-based adaptive controller is presented first, then it is extended to a model-free

Received 8 June 1999. Revised 22 November 1999. Accepted 30 November 1999.

† Department of Electrical Engineering, National University of Singapore, Singapore 119260. e-mail: elegesz@nus.edu.sg.

‡ Department of Electrical and Communication Engineering, Singapore Polytechnic, Singapore 139651. e-mail: loulin@sp.edu.sg.

§ Department of Electrical Engineering, National University of Singapore, Singapore 119260.

International Journal of Systems Science ISSN 0020–7721 print/ISSN 1464–5319 online © 2001 Taylor & Francis Ltd
http://www.tandf.co.uk/journals

neural-network-based adaptive controller. Both controllers are designed to control the positions of the constraint object and the end effectors asymptotically, and the constraint forces follow their desired trajectories. In §4, intensive simulation studies are used to show the effectiveness of the controllers.

2. Kinematics and dynamics

2.1. *Kinematics and force model*

The system under study is schematically shown in figure 1. The object is held tightly by the end effector of manipulator 2 and can be moved as required in space. The end effector of manipulator 1 follows a trajectory on the surface of the workpiece, and at the same time exerts a certain desired force on the workpiece.

2.1.1. *Notation for figure 1.* The following notation is used to describe the system in figure 1.

O_c	contact point between the end effector of manipulator 1 and the object
O_h	point where the end effector of manipulator 2 holds the object
O_o	mass centre of the object
$O_cX_cY_cZ_c$	frame fixed with the tool of manipulator 1 with its origin at the contact point O_c
$O_hX_hY_hZ_h$	frame fixed with the end effector or hand of manipulator 2 with its origin at point O_h
$O_oX_oY_oZ_o$	frame fixed with the object with its origin at the mass centre O_o
$OXYZ$	world coordinates
$r_c = [x_c^T \quad \theta_c^T]^T \in R^6$	vector describing the posture of frame $O_cX_cY_cZ_c$
$r_h = [x_h^T \quad \theta_h^T]^T \in R^6$	vector describing the posture of frame $O_hX_hY_hZ_h$
$r_o = [x_o^T \quad \theta_o^T]^T \in R^6$	vector describing the posture of frame $O_oX_oY_oZ_o$
$r_{co} = [x_{co}^T \quad \theta_{co}^T]^T \in R^6$	vector describing the posture of frame $O_cX_cY_cZ_c$ expressed in $O_oX_oY_oZ_o$
$r_{ho} = [x_{ho}^T \quad \theta_{ho}^T]^T \in R^6$	vector describing the posture of frame $O_hX_hY_hZ_h$ expressed in $O_oX_oY_oZ_o$
$q_1 \in R^{n_1}$	joint variables of manipulator 1
$q_2 \in R^{n_2}$	joint variables of manipulator 2
$x_c \in R^3$	position vector of O_c, the origin of frame $O_cX_cY_cZ_c$
$x_h \in R^3$	position vector of O_h, the origin of frame $O_hX_hY_hZ_h$
$x_o \in R^n$	position vector of O_o, the origin of frame $O_oX_oY_oZ_o$
$x_{co} \in R^3$	position vector of O_c, the origin of frame $O_cX_cY_hZ_c$ expressed in $O_oX_oY_oZ_o$
$x_{ho} \in R^3$	position vector of O_h, the origin of frame $O_hX_hY_hZ_h$ expressed in $O_oX_oY_oZ_o$
$\theta_c \in R^3$	orientation vector of frame $O_cX_cY_cZ_c$
$\theta_h \in R^3$	orientation vector of frame $O_hX_hY_hZ_h$
$\theta_o \in R^3$	orientation vector of frame $O_oX_oY_oZ_o$
$\theta_{co} \in R^3$	orientation vector of frame $O_cX_cY_cZ_c$ expressed in $O_oX_oY_oZ_o$
$\theta_{ho} \in R^3$	orientation vector of frame $O_hX_hY_hZ_h$ expressed in $O_oX_oY_oZ_o$
$\Phi(r_{co}) = 0$	trajectory expressed in the object frame $O_oX_oY_oZ_o$

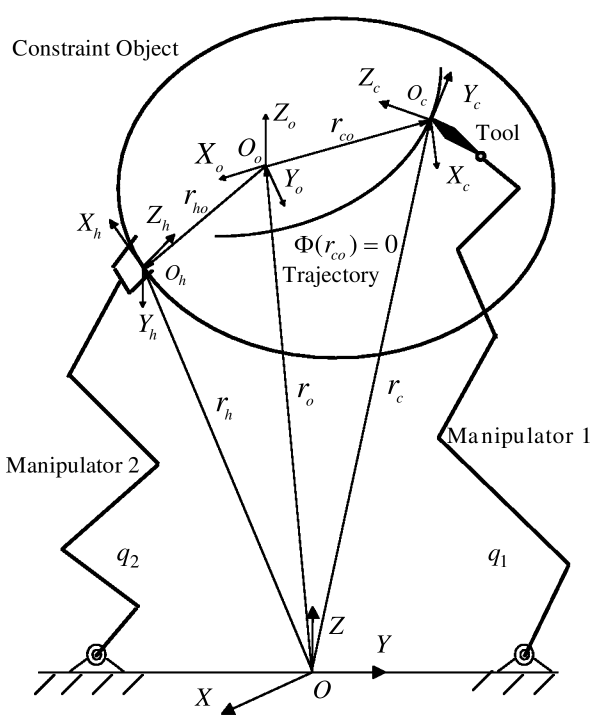

Figure 1. Coordinated operation of two robots.

2.1.2. *Equations for figure 1.* The closed kinematic relationships of the system are given by the following equations:

$$x_c = x_o + R_o(\theta_o)x_{co}, \tag{1}$$

$$x_h = x_o + R_o(\theta_o)x_{ho}, \tag{2}$$

$$R_c = R_o(\theta_o)R_{co}(\theta_{co}), \tag{3}$$

$$R_h = R_o(\theta_o), \tag{4}$$

where $R_o(\theta_o) \in R^{3\times 3}$ and $R_{co}(\theta_{co}) \in R^{3\times 3}$ are the rotation matrices of θ_o and θ_{co} respectively; $R_c \in R^{3\times 3}$ and $R_h \in R^{3\times 3}$ given above are the rotation matrices of frames $O_cX_cY_cZ_c$ and $O_hX_hY_hZ_h$ with respect to the world coordinate respectively.

Differentiating the above equations with respect to time t and considering the facts that the object is held by manipulator 2 tightly (accordingly, $\dot{x}_{ho} = 0$ and $\omega_{ho} = 0$), we have

$$\dot{x}_c = \dot{x}_o + R_o(\theta_o)\dot{x}_{co} - S(R_o(\theta_o)x_{co})\omega_o, \tag{5}$$

$$\dot{x}_h = \dot{x}_o - S(R_o(\theta_o)x_{ho})\omega_o, \tag{6}$$

$$\omega_c = \omega_o + R_o(\theta_o)\omega_{co}, \tag{7}$$

$$\omega_h = \omega_o, \tag{8}$$

with

$$S(u) := \begin{bmatrix} 0 & -u_3 & u_2 \\ u_3 & 0 & -u_1 \\ -u_2 & u_1 & 0 \end{bmatrix}$$

for a given vector $u = [u_1\ u_2\ u_3]^T$.

Define $v_c = [\dot{x}_c^T\ \ \omega_c^T]^T$, $v_h = [\dot{x}_h^T\ \ \omega_h^T]^T$, $v_o = [\dot{x}_o^T\ \ \omega_o^T]^T$, $v_{co} = [\dot{x}_{co}^T\ \ \omega_{co}^T]^T$ and $v_{ho} = [\dot{x}_{ho}^T\ \ \omega_{ho}^T]^T$. From (5)–(8), we have the following relationships:

$$v_c = Av_o + R_A v_{co}, \tag{9}$$

$$v_h = Bv_o, \tag{10}$$

where

$$R_A = \begin{bmatrix} R_o(\theta_o) & 0 \\ 0 & R_o(\theta_o) \end{bmatrix},$$

$$A = \begin{bmatrix} I^{3\times 3} & -S(R_o(\theta_o)x_{co}) \\ 0 & I^{3\times 3} \end{bmatrix},$$

$$B = \begin{bmatrix} I^{3\times 3} & -S(R_o(\theta_o)x_{ho}) \\ 0 & I^{3\times 3} \end{bmatrix}.$$

As $R_o(\theta_o)$ is a rotation matrix, $R_o(\theta_o)R_o^T(\theta_o) = I^{3\times 3}$ and $R_A R_A^T = I^{6\times 6}$. It is obvious that A and B are of full rank.

Assume that the end effector of manipulator 1 follows the trajectory $\Phi(r_{co}) = 0$ in the object coordinates. The contact force f_c is given by

$$f_c = n_c\lambda, \tag{11}$$

$$n_c = \frac{R_A(\partial\Phi/\partial r_{co})^T}{\|(\partial\Phi/\partial r_{co})^T\|}, \tag{12}$$

where λ is a Lagrange multiplier related to the magnitude of the force.

The resulting force f_o due to f_c is thus derived as follows:

$$f_o = -A^T f_c = -A^T n_c\lambda. \tag{13}$$

2.2. *Dynamic modelling*

In obtaining the dynamic model of manipulator 2, the constraint object is treated as part of the end effector. The dynamic models of manipulators 1 and 2 are described by the following equations (Ge *et al.* 1998):

$$M_1(q_1)\ddot{q}_1 + C_1(q_1, \dot{q}_1)\dot{q}_1 + G_1(q_1) = \tau_1 + J_1^T(q_1)f_c = \tau_1 + J_1^T(q_1)n_c\lambda, \tag{14}$$

$$M_2(q_2)\ddot{q}_2 + C_2(q_2, \dot{q}_2)\dot{q}_2 + G_2(q_2) = \tau_2 + J_2^T(q_2)f_o = \tau_2 - J_2^T(q_2)A^T n_c\lambda, \tag{15}$$

where $M_i(q_i)$ are the inertia matrices, $C_i(q_i, \dot{q}_i)$ are the Coriolis and centrifugal force matrices, $G_i(q_i)$ are the gravitational forces, τ_i are the joint torques and $J_i(q_i)$ are the Jacobian matrices ($i = 1,2$).

Combining (14) and (15) gives the following compact dynamic equation:

$$M(q)\ddot{q} + C(q, \dot{q})\dot{q} + G(q) = \tau + J^T(q)n_c\lambda, \tag{16}$$

where

$$M(q) = \begin{bmatrix} M_1(q_1) & 0 \\ 0 & M_2(q_2) \end{bmatrix},$$

$$C(q, \dot{q}) = \begin{bmatrix} C_1(q_1\ \dot{q}_1) & 0 \\ 0 & C_2(q_2\ \dot{q}_2) \end{bmatrix}$$

$$G(q) = \begin{bmatrix} G_1(q_1) \\ G_2(q_2) \end{bmatrix}, \quad q = \begin{bmatrix} q_1 \\ q_2 \end{bmatrix}, \quad \tau = \begin{bmatrix} \tau_1 \\ \tau_2 \end{bmatrix},$$

$$J(q) = [J_1(q_1)\quad -AJ_2(q_2)].$$

Because the motion of the system is constrained by a set of holonomic constraints represented by (9) and (10), some degrees of freedom of the system are lost and the order of the dynamics is reduced accordingly. Assume a set of *independent* n coordinates $q^1 = [q_1^1\ \ \dots\ \ q_n^1]^T$ is chosen from the joint variables q, such that q is the function of q^1, that is

$$q = q(q^1). \tag{17}$$

Differentiating (17) with respect to time t, we have

$$\dot{q} = L(q^1)\dot{q}^1, \tag{18}$$

$$\ddot{q} = L(q^1)\ddot{q}^1 + \dot{L}(q^1)\dot{q}^1, \tag{19}$$

where $L(q^1) = \partial q/\partial q^1$. It is obvious that $L(q^1)$ is of full column rank.

Substituting (18) and (19) into (16), we obtain the following reduced-order dynamic model of the system

$$M^1(q^1)\ddot{q}^1 + C^1(q^1, \dot{q}^1)\dot{q}^1 + G^1(q^1) = \tau + J^{1\mathrm{T}}(q^1)n_c\lambda, \tag{20}$$

where

$$M^1(q^1) = M(q^1)L(q^1),$$
$$C^1(q^1, \dot{q}^1) = M(q^1) \quad (\dot{L}(q^1) + C(q^1, \dot{q}^1)L(q^1)),$$
$$G^1(q^1) = G(q(q^1)) \quad \text{and} \quad J^1(q^1) = J(q(q^1)).$$

To facilitate controller design, the structural properties of dynamic model (20) are listed as follows.

Property 1: *Terms $L(q^1)$, $J^1(q^1)$ and n_c satisfy the relationship $L^{\mathrm{T}}(q^1)J^{1\mathrm{T}}(q^1)n_c = 0$.*

Proof: Solving for v_{co} from (9) yields

$$v_{co} = R_A^{-1}(v_c - \mathrm{A}v_o) = R_A^{\mathrm{T}}(v_c - Av_o). \tag{21}$$

From the kinematics of the robots, we have

$$v_c = J_1(q_1)\dot{q}_1, \tag{22}$$

$$v_o = J_2(q_2)\dot{q}_2. \tag{23}$$

Substituting (22) and (23) into (21) and noting (18), we have

$$v_{co} = R_A^{\mathrm{T}} J^1(q^1)L(q^1)\dot{q}^1. \tag{24}$$

As v_{co} and n_{co} are orthogonal to each other, we have

$$n_{co}^{\mathrm{T}} v_{co} = 0. \tag{25}$$

From (21)–(24), we obtain

$$n_c^{\mathrm{T}} J^1(q^1)L(q^1)\dot{q}^1 = 0. \tag{26}$$

As q^1 are independent variables, the following equation holds:

$$L^{\mathrm{T}}(q^1)J^{1\mathrm{T}}(q^1)n_c = 0. \tag{27}$$

□

Property 2: *$M_L(q^1) \triangleq L^{\mathrm{T}}(q^1)M^1(q^1)$ is symmetric positive definite and bounded above and below.*

Property 3: *Define $C_L(q^1, \dot{q}^1) = L^{\mathrm{T}}(q^1)C^1(q^1, \dot{q}^1)$; then $N_L = \dot{M}_L(q^1) - 2C_L(q^1, \dot{q}^1)$ is skew-symmetric if $C_i(q_i, \dot{q}_i)(i = 1,2)$ is in the Christoffel form, that is $x^{\mathrm{T}} N_L x = 0$, $\forall x \in R^n$.*

Property 4: *The dynamics described by* (20) *are linear in parameters, that is*

$$M^1(q^1)\ddot{\chi} + C^1(q^1, \dot{q}^1)\dot{\chi} + G^1(q^1) = \Psi P, \tag{28}$$

where $P \in R^l$ are the parameters of interest, $\Psi = \Psi(q^1, \dot{q}^1, \dot{\chi}, \ddot{\chi}) \in R^{n\times l}$ is the regressor matrix, and $\dot{\chi}, \ddot{\chi} \in R^n$.

Properties 2–4 are similar to the properties of the dynamic model of a single robot (Slotine and Li 1987, Yuan 1996, Murray *et al*. 1993, Sciavicco and Siciliano 1996, Kankaanranta and Kovio 1988).

3. Controller design

In this section, the model-based adaptive controller is developed for the case when the parameters are unknown, followed by the model-free neural-network-based adaptive controller in an effort to reduce further the need for the derivation of the known regressor $\Psi(*)$ in (28).

Let $r_{od}(t)$ be the desired trajectory of the object, $r_{cod}(t)$ be the desired trajectory on the workpiece and $\lambda_d(t)$ be the desired constraint force. The first control objective is to drive the manipulators such that $r_o(t)$ and $r_{co}(t)$ track their desired trajectories $r_{od}(t)$ and $r_{cod}(t)$ respectively. Accordingly it is only necessary to make $q^1(t)$ track the desired trajectory $q_d^1(t)$ since $q^1(t)$ completely determines $r_o(t)$ and $r_{co}(t)$. The second objective is to make $\lambda(t)$ track its desired trajectory $\lambda_d(t)$.

In practice, the parameters of the system are usually unknown. Let $\hat{P}$ be the estimates of parameters P, and $\tilde{P} = P - \hat{P}$. Define the following variables for the ease of discussion:

$$e^1 = q_d^1 - q^1, \tag{29}$$

$$e_\lambda = \lambda_d - \lambda, \tag{30}$$

$$r^1 = \dot{e}^1 + K_e e^1, \tag{31}$$

$$\dot{q}_r^1 = \dot{q}_d^1 + K_e e^1, \tag{32}$$

where constant $K_e \in R^{n\times n}$ is positive definite. It is obvious that

$$r^1 = \dot{q}_r^1 - \dot{q}^1. \tag{33}$$

3.1. *Model-based adaptive control*

The model-based adaptive controller is to solve the control problem when the parameters of the dynamic system are unknown, but it still requires the exact model structure.

For the dynamic system (20), consider the following controller:

$$\tau = \Psi_r \hat{P} - J^{1\mathrm{T}}(q^1) n_c \left(\lambda_d + k_\lambda \int_0^t e_\lambda(\tau)\,\mathrm{d}\tau \right) + L^{-\mathrm{T}}(q^1) K r^1, \quad (34)$$

where the constants $k_\lambda \in R$ and $K \in R^{n\times n}$ are all positive definite, and

$$\Psi_r = \Psi(q^1, \dot{q}^1, \dot{q}_r^1, \ddot{q}_r^1).$$

Applying the control law (34) to the dynamic system (20), the closed-loop dynamics are obtained:

$$M^1(q^1)\ddot{q}^1 + C^1(q^1, \dot{q}^1)\dot{q}^1 + G^1(q^1) = \Psi_r \hat{P} - J^{1\mathrm{T}}(q^1) n_c \left(e_\lambda + k_\lambda \int_0^t e_\lambda(\tau)\,\mathrm{d}\tau \right) + L^{-\mathrm{T}}(q^1) K r^1. \quad (35)$$

From property 4, we know that

$$M^1(q^1)\ddot{q}^1 + C^1(q^1, \dot{q}^1)\dot{q}^1 + G^1(q^1) = \Psi_0 P \quad (36)$$

where

$$\Psi_0 = \Psi(q^1, \dot{q}^1, \dot{q}^1, \ddot{q}^1). \quad (37)$$

Combining (35) and (36) leads to

$$J^{1\mathrm{T}}(q^1) n_c \left(e_\lambda + k_\lambda \int_0^t e_\lambda(\tau)\,\mathrm{d}\tau \right) = L^{-\mathrm{T}}(q^1) K r^1 - \Psi_r \tilde{P} + (\Psi_r - \Psi_0) P. \quad (38)$$

Pre-multiplying both sides of (35) and using property 1, we have

$$M_L(q^1)\ddot{q}^1 + C_L(q^1, \dot{q}^1)\dot{q}^1 + G_L(q^1) = L^{\mathrm{T}}(q^1)\Psi_r \hat{P} + K r^1. \quad (39)$$

Pre-multiplying (28) by $L^{\mathrm{T}}(q^1)$ and noting the change in variables, we have

$$M_L(q^1)\ddot{q}_r^1 + C_L(q^1, \dot{q}^1)\dot{q}_r^1 + G_L(q^1) = L^{\mathrm{T}}(q^1)\Psi_r P. \quad (40)$$

By subtracting (39) from (40) and using (33), it yields

$$M_L(q^1)\dot{r}^1 + C_L(q^1, \dot{q}^1) r^1 + K r^1 = L^{\mathrm{T}}(q^1)\Psi_r \tilde{P} \quad (41)$$

Note that (41) describes the dynamic behaviour of the tracking errors r^1, whereas (38) describes the behaviour of the force tracking error e_λ. It is obvious that r^1 is mainly affected by the parameter estimation errors $\tilde{P}$, while the force error e_λ is affected by both $\tilde{P}$ and the term $\Psi_r - \Psi_0$ resulting from the tracking errors e^1. For the convergence of the tracking errors e^1 and e_λ, we have the following theorem.

Theorem 1: *For the closed-loop dynamic system* (39), *if the parameters are updated by*

$$\dot{\hat{P}} = \Gamma \Psi_r^{\mathrm{T}} L(q^1) r^1, \quad (42)$$

where Γ is a constant positive definite matrix, then $e^1 \to 0$ and e_λ is bounded as $t \to \infty$, and all the closed-loop signals are bounded.

Proof: Choose the following Lyapunov function candidate:

$$V = \tfrac{1}{2} r^{1\mathrm{T}} M_L(q^1) r^1 + \tfrac{1}{2}\tilde{P}^{\mathrm{T}}\Gamma^{-1}\tilde{P}. \quad (43)$$

Differentiating (43) with respect to time t gives rise to

$$\dot{V} = r^{1\mathrm{T}} M_L(q^1)\dot{r}^1 + \tfrac{1}{2} r^{1\mathrm{T}}\dot{M}_L(q^1) r^1 + \tilde{P}^{\mathrm{T}}\Gamma^{-1}\dot{\tilde{P}}. \quad (44)$$

From property 3, we have

$$\dot{V} = r^{1\mathrm{T}}[M_L(q^1)\dot{r}^1 + C_L(q^1, \dot{q}^1) r^1] + \tilde{P}^{\mathrm{T}}\Gamma^{-1}\dot{\tilde{P}}. \quad (45)$$

From (41), we obtain

$$\dot{V} = \tilde{P}^{\mathrm{T}}\Gamma^{-1}[\Gamma\Psi_r^{\mathrm{T}} L(q^1) r^1 - \dot{\hat{P}}] - r^{1\mathrm{T}} K r^1, \quad (46)$$

where the fact that $\dot{\tilde{P}} = -\dot{\hat{P}}$ has been used.

Substituting the adaptation law (42) into the above equation leads to

$$\dot{V} = -r^{1\mathrm{T}} K r^1. \quad (47)$$

As $K > 0$, $\dot{V} \leqslant 0$; thus $r^1 \in L_2^n$. From the definition of r^1 in (31), $e^1 \to 0$, $q^1(t) \to q_d^1(t)$ as $t \to \infty$, and $\dot{e}^1 \in L_2^n$. From the closed kinematics (17), we can conclude that $q \to q_d$ when $t \to \infty$. Obviously the same conclusion cannot be made for $\tilde{P}$, but it is bounded in the sense of Lyapunov stability.

Because $e^1 \to 0$, $\dot{e}^1 \in L_2^n$ and $\tilde{P}$ is bounded, from (41) it can be concluded that $\dot{r}^1 \in L_\infty^n$. It has been proven that $r^1 \in L_2^n$; thus $r^1 \to 0$ as $t \to \infty$. From the definition of r^1 in (31), we have $\dot{e}^1 \to 0$, $e^1 \to 0$ as $t \to \infty$.

Because $r^1 \to 0$, $e^1 \to 0$, $\dot{e}^1 \to 0$ and $\tilde{P}$ is bounded when $t \to \infty$, from the definitions of $\dot{q}_r^1$, r, Ψ_r and Ψ_0, we can conclude that the right-hand side of (38) is bounded, thus e_λ is bounded and its size can be adjusted by choosing a proper gain matrix k_λ. The integral of the force error is for reducing the static error. □

Controller (34) and adaptation law (42) guarantee that $e^1 \to 0$, but they can only make the force error e_λ be bounded. Before proceeding on the ways to make e_λ converge to zero, the following definitions and lemma are reproduced for completeness (Sadegh and Horowitz 1990).

Definition 1—Almost everywhere uniform continuity: A function $f(t): R^+ \to R^n$ is said to be uniformly continuous almost everywhere if for any given t_0 and any given ε there exist $\delta(\varepsilon)$ such that

$$\|f(t) - f(t_0)\| \leqslant \varepsilon \quad \text{for all } t \in [t_0, t_0 + \delta] \text{ or } t \in [t_0 - \delta, t_0]. \quad (48)$$

□

Definition 2—Persistent excitation: A matrix function $W(t) : R^+ \to R^{m\times n} (m \leqslant n)$ is said to be persistently exciting if there exist a $\delta > 0$ and an $\alpha > 0$ such that for all $t \in R^+$ we have

$$\int_t^{t+\delta} W^{\mathrm{T}}(\tau) W(\tau)\,\mathrm{d}\tau \geqslant \alpha I. \tag{49}$$

□

Lemma 1: *Let $f(t) : R^+ \to R^n$ be a uniformly continuous almost everywhere function. Then, for any $p_0 > 0$,*

$$\lim_{t\to\infty} [f(t)] = 0 \quad \text{iff} \lim_{t\to\infty}\left(\int_t^{t+p} f(\tau)\,\mathrm{d}\tau\right) = 0 \quad \textit{for all } 0 < p < p_0 \tag{50}$$

Now we are ready to present the following theorem about the convergence of e_λ.

Theorem 2: *For the closed-loop system consisting of dynamic model* (20)*, control law* (34) *and adaptation law* (42)*, if*

(*a*) *$\ddot{q}_{\mathrm{d}}^1$ is uniformally continuous almost everywhere and*
(*b*) *$\Psi_{L\mathrm{d}} = L^{\mathrm{T}}(q_d^1)\Psi(q_{\mathrm{d}}^1, \dot{q}_{\mathrm{d}}^1, \dot{q}_{\mathrm{d}}^1, \ddot{q}_{\mathrm{d}}^1)$ is persistently exciting,*

then $e_\lambda \to 0$ as $t \to \infty$.

Proof: For clarity, define the following terms:

$$\Psi_{Lr} = L^{\mathrm{T}}(q^1)\Psi(q^1, \dot{q}^1, \dot{q}_r^1, \ddot{q}_r^1),$$
$$f(t) = \Psi_{Lr}\tilde{P}(t),$$
$$f_1(t) = M_L(q^1)\dot{r}^1(t),$$
$$f_2(t) = C_L(q^1,\ \dot{q}^1)r^1(t) + Kr^1(t).$$

Thus (41) can be rewritten as

$$f_1(t) + f_2(t) = f(t). \tag{51}$$

Integrating both sides of (51) in the interval $[t, t+p]$ $(0 \leqslant p \leqslant p_0)$, it follows that

$$\int_t^{t+p} f_1(\tau)\,\mathrm{d}\tau + \int_t^{t+p} f_2(\tau)\,\mathrm{d}\tau = \int_t^{t+p} f(\tau)\,\mathrm{d}\tau \tag{52}$$

and

$$\int_t^{t+p} f_1(\tau)\,\mathrm{d}\tau = \int_t^{t+p} M_L(q^1)(\tau)\dot{r}^1(\tau)\,\mathrm{d}\tau, \tag{53}$$

$$\int_t^{t+p} f_2(\tau)\,\mathrm{d}\tau = \int_t^{t+p} [C_L(q^1,\ \dot{q}^1)r^1(\tau) + Kr^1(\tau)]\,\mathrm{d}\tau. \tag{54}$$

By expanding the integral $\int_t^{t+p} f_1(\tau)\,\mathrm{d}\tau$, we have

$$\int_t^{t+p} f_1(\tau)\,\mathrm{d}\tau = M_L(q^1)(t+p)r^1(t+p) - M_L(q^1)(t)r^1(t) - \int_t^{t+p} \dot{M}_L(q^1)(\tau)r^1(\tau)\,\mathrm{d}\tau, \tag{55}$$

which leads to

$$\left\|\int_t^{t+p} f_1(\tau)\,\mathrm{d}\tau\right\| \leqslant \sup_{q^1}\|M_L(q^1)\|(\|r^1(t+p)\| + \|r^1(t)\|) + p_0\left(\sup_{t\leqslant\tau\leqslant t+p}\right)\|\dot{M}_L(q^1)(\tau)\|\|r^1(\tau)\|). \tag{56}$$

Note that $\dot{M}_L(q^1)(t)$ can be written as

$$\dot{M}_L(q^1)(\tau) = \sum_{i=1}^n \frac{\partial M_L(q_i^1)}{\partial q_i^1}(\tau)\dot{q}_i^1. \tag{57}$$

As proved in theorem 1, $r^1 \to 0$, $e^1 \to 0$ and $\dot{e}^1 \to 0$ when $t \to \infty$; thus $\dot{M}_L(q^1)(\tau)$ is bounded. In addition, $\sup_{q^1}\|M_L(q^1)\|$ is bounded from property 1. Therefore

$$\lim_{t\to\infty}\left(\int_t^{t+p} f_1(\tau)\,\mathrm{d}\tau\right) = 0. \tag{58}$$

From lemma 1, we have

$$\lim_{t\to\infty}[f_1(t)] = 0. \tag{59}$$

From the fact that $r^1 \to 0$ when $t \to \infty$, it is obvious that

$$\lim_{t\to\infty}[f_2(t)] = 0 \tag{60}$$

From (59) and (60), we have

$$\lim_{t\to\infty}[f(t)] = \lim_{t\to\infty}[f_1(t) + f_2(t)] = \lim_{t\to\infty}[\Psi_{Lr}\tilde{P}(t)] = 0. \tag{61}$$

Consider the following inequality:

$$\|\Psi_{L\mathrm{d}}\tilde{P}\| \leqslant \|\Psi_{L\mathrm{d}} - \Psi_{Lr}\|\|\tilde{P}\| + \|\Psi_{Lr}\tilde{P}\|. \tag{62}$$

For $r^1 \to 0$ when $t \to \infty$, and $\dot{q}_r^1 = \dot{q}_{\mathrm{d}}^1 + K_e e^1$, $\ddot{q}_r^1 = \ddot{q}_{\mathrm{d}}^1 + K_e\dot{e}^1$, we have

$$\lim_{t\to\infty}\|\Psi_{Lr} - \Psi_{L\mathrm{d}}\| = 0. \tag{63}$$

From (61)–(63), we conclude that

$$\Psi_{L\mathrm{d}}\tilde{P} \to 0 \quad \text{when } t \to \infty.$$

Let $Q(t, t+\delta) = \int_t^{t+\delta} \Psi_{L\mathrm{d}}^{\mathrm{T}}(\tau)\Psi_{L\mathrm{d}}(\tau)\,\mathrm{d}\tau$. Since $\Psi_{L\mathrm{d}}$ is persistent exciting, then, for some $\delta > 0$ and all t, we have

$$Q(t, t+\delta) \geqslant \alpha I > 0. \tag{64}$$

From adaptation law (42) and the integration by parts, we obtain

$$\tilde{P}^{\mathrm{T}}(t)Q(t,t+\delta)\tilde{P}(t) = -2\int_t^{t+\delta} \tilde{P}^{\mathrm{T}}(\tau)Q(\tau,\tau+\delta)\Gamma\Psi_{Lr}r^1(\tau)\,\mathrm{d}\tau + \int_t^{t+\delta} \tilde{P}^{\mathrm{T}}(\tau)\Psi_{Ld}^{\mathrm{T}}(\tau)\Psi_{Ld}(\tau)\tilde{P}(\tau)\,\mathrm{d}\tau.$$

From (64) and the fact that $r^1 \to 0$ when $t \to \infty$ proven in theorem 1, we can see that the right-hand side of the above equation converges to zero as $t \to \infty$. Since $Q(t,t+\delta) \geqslant \alpha I > 0$, then it can be concluded that $\tilde{P} \to 0$ as $t \to \infty$.

It has been proven that $r^1 \to 0$, $e^1 \to 0$ and $\dot{e}^1 \to 0$ as $t \to \infty$ in theorem 1. With $\tilde{P} \to 0$, we can conclude that $\Psi_r - \Psi_0 \to 0$ as $t \to \infty$. Thus, from (38), we have

$$J^{1\mathrm{T}}(q^1)n_{\mathrm{c}}\left(e_\lambda + k_\lambda \int_0^t e_\lambda(\tau)\,\mathrm{d}\tau\right) \to 0. \quad (65)$$

As $J^{1\mathrm{T}}(q^1)$ is of full column rank, we conclude that

$$e_\lambda + k_\lambda \int_0^t e_\lambda(\tau)\,\mathrm{d}\tau \to 0, \quad (66)$$

which leads to $e_\lambda(\tau) \to 0$ as $t \to \infty$ for $k_\lambda > 0$. □

Remark 1: The condition for the convergence of force is more stringent than those for the convergence of position. It requires that the trajectory q_{d}^1 be planned such that $\ddot{q}_{\mathrm{d}}^1$ and $L^{\mathrm{T}}(q_{\mathrm{d}}^1)\Psi(q_{\mathrm{d}}^1, \dot{q}_{\mathrm{d}}^1, \dot{q}_{\mathrm{d}}^1, \ddot{q}_{\mathrm{d}}^1)$ meet the conditions listed in theorem 2. □

Remark 2: The above model-based adaptive controller relies on accurate dynamic modelling of the system. The regressor matrix Ψ and Jacobian J are very time consuming and tedious in derivation and calculation. To eliminate the need for dynamic modelling, a model-free adaptive neural network controller is presented next. □

3.2. *Neural network controller*

It is well known that the Gaussian radial basis function (RBF) neural network can be used to approximate any smooth function (Haykin 1994). For a given smooth function $F(x): R^n \to R^m$, there exist optimal parameters $w_{ji} \in R$ such that

$$\hat{F}_j(x) = \sum_{i=1}^{l} w_{ji}a_i(x) = w_j^{\mathrm{T}}a(x) \quad (j = 1,2,\ldots,m) \quad (67)$$

$$\hat{F}(x) = [\hat{F}_1(x)\ \hat{F}_2(x) \ldots \hat{F}_m(x)]^{\mathrm{T}}, \quad (68)$$

$$F(x) = \hat{F}(x) + \epsilon(x), \quad (69)$$

where $\epsilon(x)$ is the minimum approximation error and $a_i(x)$ $(i = 1,2,\ldots,l)$ are the Gaussian functions defined as

$$a_i(x) = \exp\left(\frac{-(x-\mu_i)^{\mathrm{T}}(x-\mu_i)}{\sigma^2}\right), \quad (70)$$

with $\mu_i \in R^n$ being the centres of the functions, and $\sigma^2 \in R$ being the variance.

Equation (67) can be expressed in a matrix form as follows:

$$\hat{F}(x) = W^{\mathrm{T}}a(x), \quad (71)$$

where $W = [w_1 \quad w_2 \quad \ldots \quad w_m]^{\mathrm{T}}$.

The above RBF neural network is schematically shown in figure 2. It has the input layer, the hidden layer and the output layer. In the hidden layer, each node contains a Gaussian function $a_i(x)$. Note that only the connections between the hidden layer and the output layer are weighted by w_{ji}.

Consider the reduced dynamic model (20) and let $m_{kj}^1(q^1)$ and $c_{kj}^1(q^1, \dot{q}^1)$ denote the kjth elements of matrices $M^1(q^1)$ and $C^1(q^1, \dot{q}^1)$ respectively, and $g_k^1(q^1)$ be the kth element of $G^1(q^1)$. According to the above discussion, $M^1(q^1)$, $C^1(q^1, \dot{q}^1)$ and $G^1(q^1)$ can be approximated by the following RBF neural networks (Ge *et al.* 1998):

$$m_{kj}^1(q^1) = \theta_{kj}^{\mathrm{T}}\xi_{kj}(q^1) + \epsilon_{mkj}(q^1), \quad (72)$$

$$c_{kj}^1(q^1, \dot{q}^1) = \alpha_{kj}^{\mathrm{T}}\zeta_{kj}(z) + \epsilon_{ckj}(z), \quad (73)$$

$$g_k^1(q^1) = \beta_k^{\mathrm{T}}\eta_k + \epsilon_{gk}(q^1), \quad (74)$$

where $z = [(q^1)^{\mathrm{T}}, (\dot{q}^1)^{\mathrm{T}}]^{\mathrm{T}} \in R^{2n}$, $\xi_{kj}(q^1) \in R^{l_{Mkj}}$, $\zeta_{kj}(z) \in R^{l_{Ckj}}$ and $\eta_k(q^1) \in R^{l_{Gk}}$ are the vectors of Gaussian functions of the form defined in (70); $\theta_{kj} \in R^{l_{Mkj}}$, $\alpha_{kj} \in R^{l_{Ckj}}$ and $\beta_k \in R^{l_{Gk}}$ are the vectors of optimal weights of the neural network which make the modelling errors $\epsilon_{mkj}(q^1)$, $\epsilon_{ckj}(z)$ and $\epsilon_{gk}(q^1)$ a minimum.

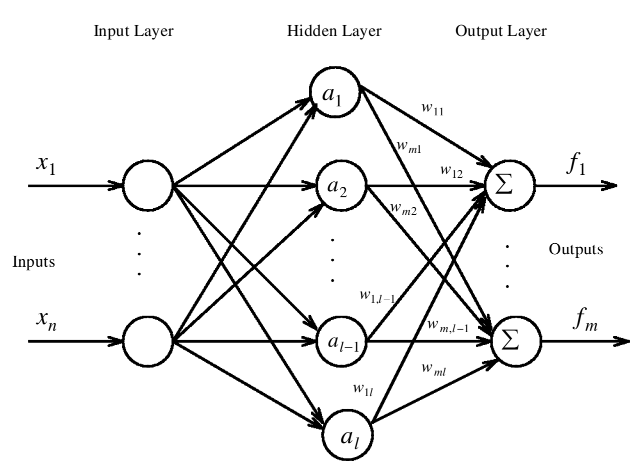

Figure 2. RBF neural network.

To simplify the above algebraic expressions of neural networks, we adopt the notation of Ge-Lee matrix (Ge *et al.* 1998) in the following discussion. A GL matrix is normally expressed in the form $\{*\}$ to differentiate it from a normal matrix $[*]$. The unique characteristics of the GL matrices are that the transposes and the product of the matrices are obtained 'locally'. For example, for two GL matrices $\{\Theta\}$ and $\{\Xi(q^1)\}$ and a normal square matrix Γ_k as follows

$$\{\Theta\} = \begin{Bmatrix} \theta_{11} & \theta_{12} & \cdots & \theta_{1n} \\ \theta_{21} & \theta_{22} & \cdots & \theta_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ \theta_{n1} & \theta_{n2} & \cdots & \theta_{nn} \end{Bmatrix} = \begin{Bmatrix} \{\theta_1\} \\ \{\theta_2\} \\ \vdots \\ \{\theta_n\} \end{Bmatrix},$$

$$\{\Xi(q^1)\} = \begin{Bmatrix} \xi_{11}(q^1) & \xi_{12}(q^1) & \cdots & \xi_{1n}(q^1) \\ \xi_{21}(q^1) & \xi_{22}(q^1) & \cdots & \xi_{2n}(q^1) \\ \vdots & \vdots & \ddots & \vdots \\ \xi_{n1}(q^1) & \xi_{n2}(q^1) & \cdots & \xi_{nn}(q^1) \end{Bmatrix} = \begin{Bmatrix} \{\xi_1\} \\ \{\xi_2\} \\ \vdots \\ \{\xi_n\} \end{Bmatrix},$$

$$\Gamma_k = \Gamma_k^{\mathrm{T}} = [\gamma_{k1} \quad \gamma_{k2} \quad \cdots \quad \gamma_{kn}],$$

we have

$$\{\Theta\}^{\mathrm{T}} = \begin{Bmatrix} \theta_{11}^{\mathrm{T}} & \theta_{12}^{\mathrm{T}} & \cdots & \theta_{1n}^{\mathrm{T}} \\ \theta_{21}^{\mathrm{T}} & \theta_{22}^{\mathrm{T}} & \cdots & \theta_{2n}^{\mathrm{T}} \\ \vdots & \vdots & \ddots & \vdots \\ \theta_{n1}^{\mathrm{T}} & \theta_{n2}^{\mathrm{T}} & \cdots & \theta_{nn}^{\mathrm{T}} \end{Bmatrix},$$

$$\{\Theta\}^{\mathrm{T}} \cdot \{\Xi(q^1)\} = \begin{bmatrix} \theta_{11}^{\mathrm{T}}\xi_{11}(q^1) & \theta_{12}^{\mathrm{T}}\xi_{12}(q^1) & \cdots & \theta_{1n}^{\mathrm{T}}\xi_{1n}(q^1) \\ \theta_{21}^{\mathrm{T}}\xi_{21}(q^1) & \theta_{22}^{\mathrm{T}}\xi_{22}(q^1) & \cdots & \theta_{2n}^{\mathrm{T}}\xi_{2n}(q^1) \\ \vdots & \vdots & \ddots & \vdots \\ \theta_{n1}^{\mathrm{T}}\xi_{n1}(q^1) & \theta_{n2}^{\mathrm{T}}\xi_{n2}(q^1) & \cdots & \theta_{nn}^{\mathrm{T}}\xi_{nn}(q^1) \end{bmatrix},$$

$$\Gamma_k \cdot \{\xi_k\} = \{\Gamma_k\} \cdot \{\xi_k\} := [\gamma_{k1}\xi_{k1} \quad \gamma_{k2}\xi_{k2} \quad \cdots \quad \gamma_{kn}\xi_{kn}],$$

where $\{*\} \cdot \{*\}$ represents the multiplication of two GL matrices and $[*] \cdot \{*\}$ represents the multiplication of a square matrix with a GL matrix of compatible dimension. Note that their products are all normal matrices.

Let α_{kj} and ζ_{kj} be the kjth elements of GL matrices $\{A\}$ and $\{Z(z)\}$ respectively, and β_k and η_k be the kth elements of the GL matrices $\{B\}$ and $\{H(q^1)\}$ respectively. By using these defined GL matrices, (72)–(74) can be rewritten as follows:

$$M^1(q^1) = [\{\Theta\}^{\mathrm{T}} \cdot \{\Xi(q^1)\}] + E_M(q^1), \tag{75}$$

$$C^1(q^1, \dot{q}^1) = [\{A\}^{\mathrm{T}} \cdot \{Z(z)\}] + E_C(z), \tag{76}$$

$$G^1(q^1) = [\{B\}^{\mathrm{T}} \cdot \{H(q^1)\}] + E_G(q^1), \tag{77}$$

where $E_M(q^1)$, $E_C(z)$ and $E_G(q^1)$ are matrices with $\epsilon_{mkj}(q^1)$, $\epsilon_{ckj}(z)$ and $\epsilon_{gk}(q^1)$ being their elements respectively.

Let the estimates of Θ, A and B be $\hat{\Theta}$, $\hat{A}$ and $\hat{B}$ respectively. The neural network estimates of $M^1(q^1)$, $C^1(q^1, \dot{q}^1)$ and $G^1(q^1)$ are expressed as follows:

$$\hat{M}_{\mathrm{nn}}^1(q^1) = [\{\hat{\Theta}\}^{\mathrm{T}} \cdot \{\Xi(q^1)\}], \tag{78}$$

$$\hat{C}_{\mathrm{nn}}^1(q^1, \dot{q}^1) = [\{\hat{A}\}^{\mathrm{T}} \cdot \{Z(z)\}], \tag{79}$$

$$\hat{G}_{\mathrm{nn}}^1(q^1) = [\{\hat{B}\}^{\mathrm{T}} \cdot \{H(q^1)\}]. \tag{80}$$

Consider the following neural-network-based controller:

$$\begin{aligned} \tau = {} & \hat{M}_{\mathrm{nn}}^1(q^1)\ddot{q}_r^1 + \hat{C}_{\mathrm{nn}}^1(q^1, \dot{q}^1)\dot{q}_r^1 + \hat{G}_{\mathrm{nn}}^1(q^1) \\ & - J^{1\mathrm{T}}(q^1)n_{\mathrm{c}}\left(\lambda_d + k_\lambda \int_0^t e_\lambda \, \mathrm{d}t\right) \\ & + L^{-\mathrm{T}}(q^1)[Kr^1 + K_s \operatorname{sgn}(r^1)], \end{aligned} \tag{81}$$

where the control parameters k_λ, K and K_s are all positive definite.

Applying control law (81) to the dynamic system (20), we have

$$\begin{aligned} & M^1(q^1)\ddot{q}^1 + C^1(q^1, \dot{q}^1)\dot{q}^1 + G^1(q^1) \\ & \quad = \hat{M}_{\mathrm{nn}}^1(q^1)\ddot{q}_r^1 + \hat{C}_{\mathrm{nn}}^1(q^1, \dot{q}^1)\dot{q}_r^1 + \hat{G}_{\mathrm{nn}}^1(q^1) \\ & \qquad - J^{1\mathrm{T}}(q^1)n_{\mathrm{c}}\left(e_\lambda + k_\lambda \int_0^t e_\lambda \, \mathrm{d}t\right) \\ & \qquad + L^{-\mathrm{T}}(q^1)[Kr^1 + K_s \operatorname{sgn}(r^1)]. \end{aligned} \tag{82}$$

Multiplying both sides of (82) by $L^{\mathrm{T}}(q^1)$ and making use of property 1, we have

$$\begin{aligned} & L^{\mathrm{T}}(q^1)[M^1(q^1)\ddot{q}^1 + C^1(q^1, \dot{q}^1)\dot{q}^1 + G^1(q^1)] \\ & \quad = L^{\mathrm{T}}(q^1)[\hat{M}_{\mathrm{nn}}^1(q^1)\ddot{q}_r^1 + \hat{C}_{\mathrm{nn}}^1(q^1, \dot{q}^1)\dot{q}_r^1 + \hat{G}^1(q^1)] \\ & \qquad + L^{\mathrm{T}}(q^1)[Kr^1 + K_s \operatorname{sgn}(r^1)]. \end{aligned} \tag{83}$$

Substituting (75)–(80) and through some simple algebraic manipulations, the above equation is rewritten as

$$\begin{aligned} & L^{\mathrm{T}}(q^1)(M^1(q^1)\dot{r}^1 + C^1(q^1, \dot{q}^1)r^1) + Kr^1 + K_s \operatorname{sgn}(r^1) \\ & \quad = L^{\mathrm{T}}([\{\tilde{\Theta}\}^{\mathrm{T}} \cdot \{\Xi(q^1)\}]\ddot{q}_r^1 + [\{\tilde{A}\}^{\mathrm{T}} \cdot \{Z(z)\}]\dot{q}_r^1 \\ & \qquad + [\{\tilde{B}\} \cdot \{H(q^1)\}]) + L^{\mathrm{T}}(q^1)E, \end{aligned} \tag{84}$$

where

$$E = L^{\mathrm{T}}(q^1)[E_M(q^1)\ddot{q}_r^1 + E_C(q^1, \dot{q}^1)\dot{q}_r^1 + E_G(q^1))] \quad (85)$$

and $(\tilde{*}) = (*) - (\hat{*})$.

Equation (84) describes the dynamic behaviour of the tracking errors r^1 under the proposed controller. The right-hand side of the equation is a function of neural network estimations. The dynamic behaviour of the force variable λ is described in (82), which is directly affected by the errors r^1.

For the convergence of r^1, e^1 and the boundedness of e_λ, we have the following theorem.

Theorem 3: *For the closed-loop dynamic system* (84) *with* $K_s > \|E\|$ *, if terms* $M^1(q^1)$*,* $C^1(q^1, \dot{q}^1)$ *and* $G(q^1)$ *are approximated by the neural networks* (78), (79) *and* (80) *respectively with the weight matrices being updated according to*

$$\dot{\hat{\theta}}_k = \Gamma_k \cdot \{\xi_k(q^1)\}\ddot{q}_r^1(L(q^1)r^1)_k, \quad (86)$$

$$\dot{\hat{\alpha}}_k = Q_k \cdot \{\zeta_k(z)\}\dot{q}_r^1(L(q^1)r^1)k, \quad (87)$$

$$\dot{\hat{\beta}}_k = U_k\eta_k(q^1)(L(q^1)r^1)_k, \quad (88)$$

where $\Gamma_k = \Gamma_k^{\mathrm{T}} > 0$, $Q_k = Q_k^{\mathrm{T}} > 0$ *and* $U_k = U_k^{\mathrm{T}} > 0$, *and* $\hat{\theta}_k$*,* $\hat{\alpha}_k$*,* $\hat{\beta}_k$*,* $\xi_k(q^1)$*,* $\zeta_k(z)$*, and* $\eta_k(q^1)$ *represents the kth column vector of the corresponding matrices* $\{\hat{\Theta}\}$*,* $\{\hat{A}\}$ *,* $\{\hat{B}\}$*,* $\{\Xi(q^1)\}$*,* $\{Z(z)\}$ *and* $\{H(q^1)\}$ *respectively, then* $\hat{\theta}_k$*,* $\hat{\alpha}_k$*,* $\hat{\beta}_k \in L_\infty$*,* $e^1 \in L_2^n \cap L_\infty^n$*,* e^1*,* $\dot{e}^1 \to 0$ *and* e_λ *is bounded when* $t \to \infty$*.*

Proof: Choose a Lyapunov function

$$V = \tfrac{1}{2} r^{1\mathrm{T}} M_L(q^1) r^1 + \tfrac{1}{2}\sum_{k=1}^{n}(\tilde{\theta}_k^{\mathrm{T}}\Gamma_k^{-1}\tilde{\theta}_k + \tilde{\alpha}_k^{\mathrm{T}} Q_k^{-1}\tilde{\alpha}_k + \tilde{\beta}_k^{\mathrm{T}} U_k^{-1}\tilde{\beta}_k), \quad (89)$$

where $\tilde{\theta}_k = \theta_k - \hat{\theta}_k$, $\tilde{\alpha}_k = \alpha_k - \hat{\alpha}_k$ and $\tilde{\beta}_k = \beta_k - \hat{\beta}_k$.

Differentiating (89), and noting that $\dot{\tilde{\theta}}_k = -\dot{\hat{\theta}}_k$, $\dot{\tilde{\alpha}}_k = -\dot{\hat{\alpha}}_k$ and $\dot{\tilde{\beta}}_k = -\dot{\hat{\beta}}_k$, we have

$$\dot{V} = r^{1\mathrm{T}} M_L(q^1)\dot{r}^1 + \tfrac{1}{2} r^{1\mathrm{T}}\dot{M}_L(q^1) r^1 - \sum_{k=1}^{n}(\tilde{\theta}_k^{\mathrm{T}}\Gamma_k^{-1}\dot{\hat{\theta}}_k - \tilde{\alpha}_k^{\mathrm{T}} Q_k^{-1}\dot{\hat{\alpha}}_k + \tilde{\beta}_k^{\mathrm{T}} U_k^{-1}\dot{\hat{\beta}}_k). \quad (90)$$

From (86)–(88) and property 3, it follows that

$$\dot{V} = r^{1\mathrm{T}} M_L(q^1)\dot{r}^1 + r^{1\mathrm{T}} C_L(q^1\ \dot{q}^1) r^1 - \sum_{k=1}^{n}\{\tilde{\theta}_k\}^{\mathrm{T}} \cdot \{\xi_k(q^1)\}\ddot{q}_r^1(L(q^1)r^1)_k - \sum_{k=1}^{n}\{\tilde{\alpha}_k\}^{\mathrm{T}} \cdot \{\zeta_k(z)\}\dot{q}_r^1(L(q^1)r^1)_k - \sum_{k=1}^{n}\tilde{\beta}_k^{\mathrm{T}}\eta_k(q^1)(L(q^1)r^1)_k. \quad (91)$$

From the definitions of the product of GL matrices, we have

$$\sum_{k=1}^{n}\{\tilde{\theta}_k\}^{\mathrm{T}} \cdot \{\xi_k(q^1)\}\ddot{q}_r^1(L(q^1)r^1)_k = r^{1\mathrm{T}} L^{\mathrm{T}}(q^1)[\{\tilde{\Theta}\}^{\mathrm{T}} \cdot \{\Xi(q^1)\}]\ddot{q}_r^1, \quad (92)$$

$$\sum_{k=1}^{n}\{\tilde{\alpha}_k\}^{\mathrm{T}} \cdot \{\zeta_k(z)\}\dot{q}_r^1(L(q^1)r^1)_k = r^{1\mathrm{T}} L^{\mathrm{T}}(q^1)[\{\tilde{A}\}^{\mathrm{T}} \cdot \{Z(z)\}]\dot{q}_r^1, \quad (93)$$

$$\sum_{k=1}^{n}\tilde{\beta}_k^{\mathrm{T}}\eta_k(q^1)(L(q^1)r^1)_k = r^{1\mathrm{T}} L^{\mathrm{T}}(q^1)[\{\tilde{B}\} \cdot \{H(q^1)\}]. \quad (94)$$

Substituting (92)–(94) into (91), we have

$$\dot{V} = r^{1\mathrm{T}} L^{\mathrm{T}}(q^1)(M^1(q^1)\dot{r}^1 + C^1(q^1\ \dot{q}^1)r^1) - r^{1\mathrm{T}} L^{\mathrm{T}}(q^1)([\{\tilde{\Theta}\}^{\mathrm{T}} \cdot \{\Xi(q^1)\}]\ddot{q}_r^1 + [\{\tilde{A}\}^{\mathrm{T}} \cdot \{Z(z)\}]\dot{q}_r^1 + [\{\tilde{B}\} \cdot \{H(q^1)\}]). \quad (95)$$

From (84), we have

$$\dot{V} = -r^{1\mathrm{T}} K r^1 - [r^{1\mathrm{T}} K_s \operatorname{sgn}(r^1) - r^{1\mathrm{T}} E)]. \quad (96)$$

As $K > 0$ and $K_s > \|E\|$, thus

$$\dot{V} \leqslant -r^{1\mathrm{T}} K r^1 \leqslant 0. \quad (97)$$

As $V > 0$ and $\dot{V} \leqslant 0$, $V \in L_\infty$. From the definition of V, it follows that $r^1 \in L_2^n$ and θ_k, α_k, $\beta_k \in L_\infty$. From the definition of r^1 in (31), $e^1 \to 0$, $q^1(t) \to q_{\mathrm{d}}^1(t)$ as $t \to \infty$, and $\dot{e}^1 \in L_2^n$ (Desoer and Vidyasagar 1975). From the closed kinematics (17), we can conclude that $q \to q_{\mathrm{d}}$ when $t \to \infty$.

Because $e^1 \to 0$, $\dot{e}^1 \in L_2^n$ and θ_k, α_k , $\beta_k \in L_\infty$ as proved above, $E \in L_\infty$ from its definition (85). From (84), $\dot{r}^1 \in L_\infty^n$. It has been proven that $r^1 \in L_2^n$, thus $r^1 \to 0$ as $t \to \infty$. From the definition of r^1 in (31), we have $\dot{e}^1 \to 0$, $e^1 \to 0$ as $t \to \infty$. Based on the above conclusions, it is obvious that e_λ is bounded from (82). □

Remarks 3: The controller can only guarantee the boundedness of the force error e_λ. More stringent conditions are required to make it converge to zero including that $\ddot{q}_{\mathrm{d}}^1$ must be uniformally continuous almost everywhere as discussed in theorem 2. □

Remark 4: The weights of the neural networks are updated on line by the tracking errors. The time-consuming off-line training of neural networks is thus not required. □

Remark 5: The chattering caused by the sign function sgn (r^1) is inevitable. Many effective methods are available to diminish the chattering, one of which is to introduce a boundary layer into the controller as suggested by Slotine and Li (1987) and Ge *et al.* (1998). □

Remark 6: Both the model-based controller (34) and the neural network controller (81) neglect the dynamics of the actuators of the robot and use the joint torques as the inputs. For better control performance at high operational speed, the actuator dynamics have to be taken into consideration (Ge and Postlethwaite 1994, Su and Stepanenko 1995). □

4. Simulation

The system used for simulation is schematically shown in figure 3. The rectangular object is held rigidly by manipulator 2 which has only one degree of freedom and moves in the horizontal plane. The end effector of manipulator 1 of two degrees of freedom is to track a specified trajectory on the object.

In figure 3, XOY is the world coordinates, the object coordinate $X_o O_o Y_o$ is at the object mass centre O_o, and the length, the mass and the moment of inertia of each link of manipulator 1 are denoted by d_i, m_i and I_i $(i = 1,2)$ respectively. Let l_i $(i = 1,2)$ be the distance of the mass centre of each link from the respective joint. The mass of manipulator 2 together with the object is M_2. The joint variables for the two manipulators are $q_1 = [\theta_1\ \theta_2]^T$ and $q_2 = x$ respectively. Note that x is actually a linear displacement of the object in the horizontal plane. The gravitational acceleration is denoted by $g = 9.8\ \mathrm{m/s^2}$. For simplicity, let $c_i = \cos(\theta_i)$, $s_i = \sin(\theta_i)$, $c_{ij} = \cos(\theta_i + \theta_j)$, and $s_{ij} = \sin(\theta_i + \theta_j)$. From figure 3, the following position vectors are derived:

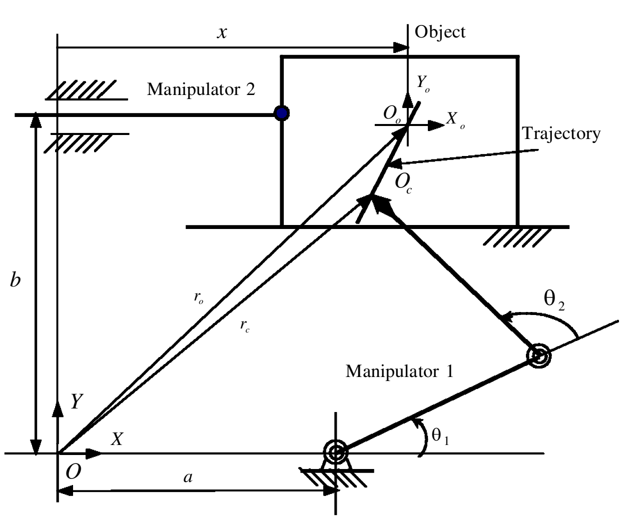

Figure 3. Simulation example.

$$r_c = [d_1 c_1 + d_2 c_{12} + a \quad d_1 s_1 + d_2 s_{12}]^T, \tag{98}$$

$$r_{co} = [d_1 c_1 + d_2 c_{12} - q_2 + a \quad d_1 s_1 + d_2 s_{12} - b]^T, \tag{99}$$

$$r_o = [q_2 \ \ b]^T, \tag{100}$$

where r_c and r_o are described with respect to the world coordinates, while r_{co} is described with respect to the object frame.

The trajectory on the object is assumed to be a straight line with reference to the object frame

$$\Phi(r_{co}) = x_{co} - y_{co} = 0 \tag{101}$$

The inverse kinematic equations of manipulator 1 are given by

$$\theta_1 = \arctan\left(\frac{s_1}{c_1}\right), \tag{102}$$

$$\theta_2 = \arctan\left(\frac{s_2}{c_2}\right), \tag{103}$$

where

$$c_2 = \frac{(x_c - a)^2 + y_c^2 - d_1^2 - d_2^2}{2 d_1 d_2}, \tag{104}$$

$$s_2 = \pm(1 - c_2^2)^{1/2}, \tag{105}$$

$$s_1 = \frac{(d_1 + d_2 c_2) y_c - d_2 s_2 (x_c - a)}{(x_c - a)^2 + y_c^2}, \tag{106}$$

$$c_1 = \frac{(d_1 + d_2 c_2) y_c + d_2 s_2 (x_c - a)}{(x_c - a)^2 + y_c^2)}. \tag{107}$$

Choose $q^1 = [\theta_1 \quad \theta_2]^T$. It is obvious that

$$q_1(q^1) = q^1, \tag{108}$$

$$q_2(q^1) = d_1(c_1 - s_1) + d_2(c_{12} - s_{12}) + a + b. \tag{109}$$

From the above equations, the following quantities are derived:

$$A = R_o = I^{2\times 2}, \quad n_{co} = n_c = \begin{bmatrix} \frac{1}{2^{1/2}} \\ \frac{-1}{2^{1/2}} \end{bmatrix},$$

$$J_2(q_2) = \begin{bmatrix} 1 \\ 0 \end{bmatrix},$$

$$J_1(q_1) = \begin{bmatrix} -d_1 s_1 - d_2 s_{12} & -d_2 s_{12} \\ d_1 c_1 + d_2 c_{12}, & d_2 c_{12} \end{bmatrix},$$

$$L(q^1) = \begin{bmatrix} 1 & 0 \\ 0 & 1 \\ -d_1(s_1+c_1)-d_2(s_{12}+c_{12}) & -d_2(s_{12}+c_{12}) \end{bmatrix},$$

$$\dot{L}(q^1) = \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ d_1(s_1-c_1)\dot{\theta}_1+d_2(s_{12}-c_{12})\dot{\theta}_{12} & d_2(s_{12}-c_{12})\dot{\theta}_{12} \end{bmatrix}.$$

It can be verified that $L^{\mathrm{T}}(q^1)J^{1\mathrm{T}}(q^1)n_{\mathrm{c}} = 0$ as stated in property 1.

The dynamic model for manipulator 1 (two-link arm) is given by

$$M_1(q_1)\ddot{q}_1 + C_1(q_1,\dot{q}_1)(q_1,\ \dot{q}_1)\dot{q}_1 + G_1(q_1)(q_1) = \tau_1 + J_1^{\mathrm{T}}(q_1)f_{\mathrm{c}} = \tau_1 + J_1^{\mathrm{T}}(q_1)n_{\mathrm{c}}\lambda \tag{110}$$

where

$$M_1(q_1) = \begin{bmatrix} I_1+m_1l_1^2+I_2+m_2(d_1^2+l_2^2+2d_1l_2c_2) & I_2+m_2(l_2^2+d_1l_2c_2) \\ I_2+m_2(l_2^2+d_1l_2c_2) & I_2+m_2l_2^2 \end{bmatrix},$$

$$C_1(q_1,\dot{q}_1) = \begin{bmatrix} -m_2d_1l_2s_2\dot{\theta}_2 & -m_2d_1l_2s_2(\dot{\theta}_1+\dot{\theta}_2) \\ m_2d_1l_2s_2\dot{\theta}_1 & 0 \end{bmatrix},$$

$$G_1(q^1) = [(m_1l_1+m_2d_1)gc_1 + m_2l_2gc_{12} \quad m_2l_2gc_{12}]^{\mathrm{T}}.$$

The dynamic model for manipulator 2 is as follows:

$$M_2\ddot{q}_2 = \tau_2 - J_2^{\mathrm{T}}(q_2)A^{\mathrm{T}}n_{\mathrm{c}}\lambda. \tag{111}$$

The reduced dynamic model is described by

$$M^1(q^1)\ddot{q}^1 + C^1(q^1,\dot{q}^1)\dot{q}^1 + G^1(q^1) = \tau + J^{1\mathrm{T}}(q^1)n_{\mathrm{c}}\lambda, \tag{112}$$

with all the terms as defined in (20).

Define $q_r^1 = [q_{r1}^1\ q_{r2}^1]^{\mathrm{T}}$. The regressor matrix $\Psi(q^1,\dot{q}^1,\dot{q}_r^1\ \ddot{q}_r^1)$ and the parameter vector P in (28) are as follows:

$$\Psi(q^1,\dot{q}^1,\dot{q}_r^1,\ddot{q}_r^1) = \begin{bmatrix} \ddot{\theta}_1 & \Psi_{12} & \ddot{q}_{r2}^1 & c_1 & c_{12} & 0 & 0 \\ 0 & \Psi_{22} & \Psi_{23} & 0 & c_{12} & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & \Psi_{36} & \Psi_{37} \end{bmatrix},$$

$$P = [p_1 \quad m_2d_1l_2 \quad I_2+m_2l_2^2 \quad (m_1l_1+m_2d_1)g \quad m_2l_2g \quad M_2d_1 \quad M_2d_2]^{\mathrm{T}}$$

where

$$\Psi_{12} = 2c_2\ddot{q}_{r1}^1 + c_2\ddot{q}_{r2}^1 - s_2(\dot{\theta}_1+\dot{\theta}_2)\dot{q}_{r2}^1 - s_2\dot{\theta}_2\dot{q}_{r1}^1,$$
$$\Psi_{22} = c_2\ddot{q}_{r1}^1 + s_2\dot{\theta}_1\dot{q}_{r1}^1,$$
$$\Psi_{23} = \ddot{q}_{r1}^1 + \ddot{q}_{r2}^1,$$
$$\Psi_{36} = (s_1-c_1)\dot{\theta}_1\dot{q}_{r1}^1 - (s_1+c_1)\ddot{q}_{r1}^1,$$
$$\Psi_{37} = (s_{12}-c_{12})(\dot{\theta}_1+\dot{\theta}_2)(\dot{q}_{r1}^1+\dot{q}_{r2}^1) - (s_{12}+c_{12})(\ddot{q}_{r1}^1+\ddot{q}_{r2}^1),$$
$$p_1 = I_1 + m_1l_1^2 + I_2 + m_2(d_1^2+l_2^2).$$

Assume that the geometric parameters are $d_1 = d_2 = 0.3\,\mathrm{m}$, $l_1 = l_2 = 0.15\,\mathrm{m}$, $a = 0.2$ and $b = 0.5\,\mathrm{m}$. The true values of the mass and inertia parameters are assumed to be $m_1 = m_2 = 0.1\,\mathrm{kg}$, $M_2 = 0.2\mathrm{kg}$, $I_1 = I_2 = 0.3\,\mathrm{kg\,m^2}$, which are unknown for controller design. The true parameters are $P = [0.6\ \ 0.04\ \ 0.3\ \ 0.44\ \ 0.15\ \ 0.06\ \ 0.06]^{\mathrm{T}}$, while the initial estimates of parameters are $\hat{P}(0) = [0.2\ \ 0.01\ \ 0.4\ \ 0.4\ \ 0.1\ \ 0.2\ \ 0.2]^{\mathrm{T}}$.

The trajectory on the object is given by

$$x_{\mathrm{co}} = y_{\mathrm{co}} = -\tfrac{1}{12}\cos(t+2). \tag{113}$$

The trajectory for the object to follow is given by

$$x_{\mathrm{o}} = \tfrac{1}{10}[1-\sin(4t+12)]. \tag{114}$$

The desired force is $\lambda_{\mathrm{d}} = 2\,\mathrm{N}$. From (104) and (105), we can obtain the desired q_{d}^1, $\dot{q}_{\mathrm{d}}^1$ and $\ddot{q}_{\mathrm{d}}^1$ which are required by the controller.

4.1. *Simulation of the adaptive control scheme*

The gain matrices are chosen as $K_e = \mathrm{diag}\,[20] \in R^{2\times2}$, $K = \mathrm{diag}\,[15] \in R^{2\times2}$ and $k_\lambda = 15$. The adaptation gain matrix Γ in adaptation law (42) is chosen as $\Gamma = \mathrm{diag}\,[15] \in R^{7\times7}$. The position tracking performances of the object and the force tracking performances are plotted in figures 4 and 5 respectively. The control torques for the manipulators are given in figure 6. From the figures, it can be seen that the position and force tracking errors approach zero. The torque of the robotic arms are also in the reasonable range. We can conclude that the proposed adaptive controller effectively control the position and the force although the true parameters are unknown.

4.2. *Simulation of the neural network control scheme*

Based on the planned trajectories, the range of the angular displacement q^1 is $[-1.2, 1.7]$ rad and the range of the angular velocity $\dot{q}^1$ is $[-1.0, 1.0]\,\mathrm{rad\,sec^{-1}}$. The two-dimensional input space for $\hat{M}_{\mathrm{nn}}(q^1)$ and $\hat{G}_{\mathrm{nn}}(q^1)$ is spanned by q^1 and the four-dimensional

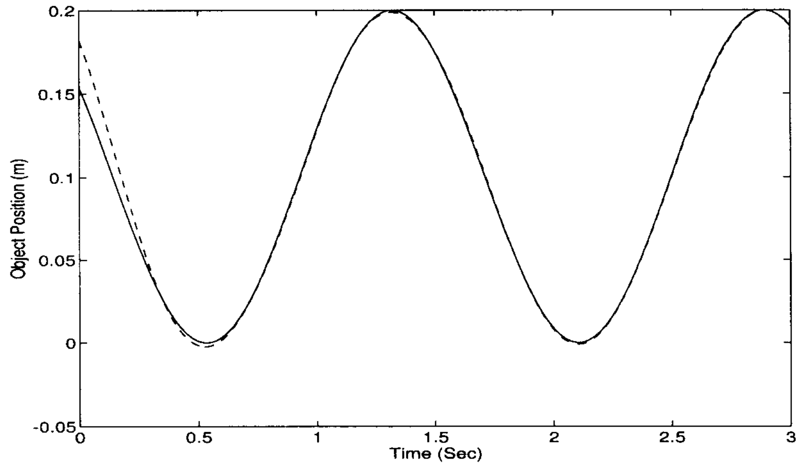

Figure 4. **Position tracking under adaptive control: (——), $r_d(t)$; (- - - - -), $r(t)$.**

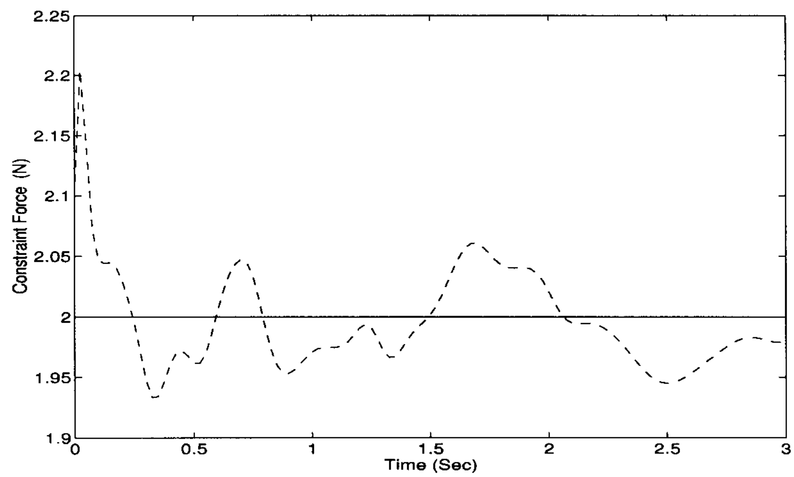

Figure 5. **Constraint force tracking under adaptive control: (——), $\lambda_d(t)$; (- - - - -), $\lambda(t)$.**

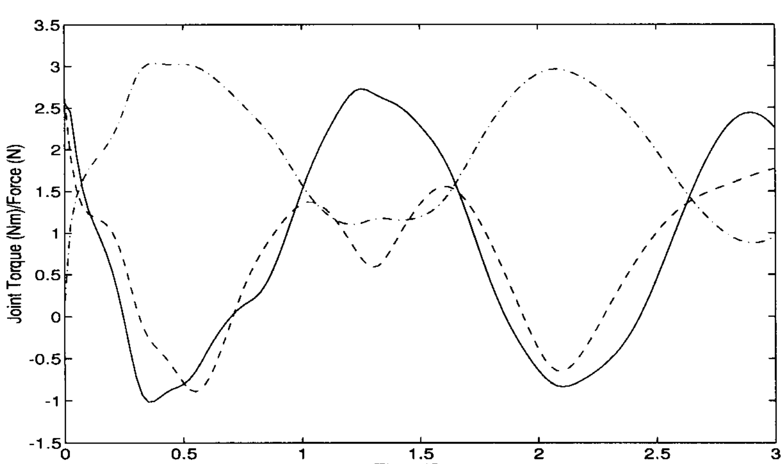

Figure 6. **Torques and forces of the manipulators under adaptive control: (——), (- - - - -) τ_1; (– · –), τ_2.**

input space for $\hat{C}_{nn}(q^1, \dot{q}^1)$ is spanned by $[q^1 \quad \dot{q}^1]^T$. The centres of the RBF functions in the neural network are the crossing point of the grids evenly distributed in the input spaces of $\hat{M}_{nn}(q^1)$, $\hat{G}_{nn}(q^1)$ and $\hat{C}_{nn}(q^1, \dot{q}^1)$ respectively (Ge *et al.* 1998). In the simulation, a 120-node neural network with $\delta^2 = 40$ is used to estimate each element of $M^1(q^1)$, $C^1(q^1, \dot{q}^1)$ and $G^1(q^1)$ respectively. The controller gain matrices are chosen as $K_e = \text{diag}[20] \in R^{2\times2}$, $K = \text{diag}[15] \in R^{2\times2}$ and $k_\lambda = 15$. The boundary layer is chosen as $\|\Delta\| = 0.01$. The updating of the weights of the neural works are activated with $\Gamma_{kij} = 0.1$, $Q_{kij} = 0.2$ and $U_{kij} = 5.0$, $i = 120, k = 3, j = 2$. The position tracking performances of the object and the force tracking performances are plotted in figures 7 and 8 respectively. The control torques for the manipulators are given in figure 9. The neural network approximation performances are also shown from figures 10–12. From the simulation results, it can be seen that under the proposed adaptive neural network controller, the positions and the forces converge to their desired values and the torques are in the reasonable ranges. While $\hat{G}^1_{nn}$ almost converge to its true value G, $\hat{M}^1_{nn}$ and $\hat{C}^1_{nn}$ do not converge to M and C respectively. The approximation errors are affected by the persistent excitation condition. The overall performance of the controller is satisfactory with the model of the system unknown.

Comparing the performance of the neural-network-based adaptive controller with that of the model-based adaptive controller, there is not much difference in the accuracy and the speed of position tracking. While the force error is bounded without being convergent to zero in both controllers, the magnitudes of the fluctuations of the force signals are larger under neural-network-based adaptive control than those under model-based adaptive control, especially at the initial stage of control. The

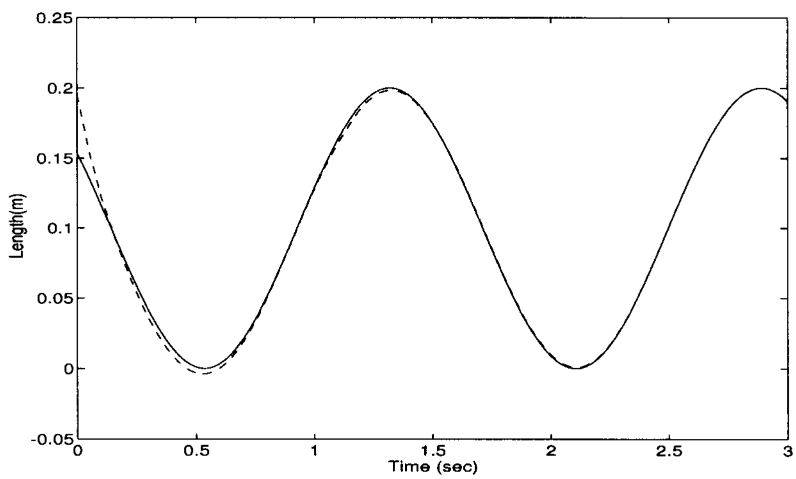

Figure 7. Object position tracking under neural network control: (——), $r_d(t)$; (- - - - -), $r(t)$.

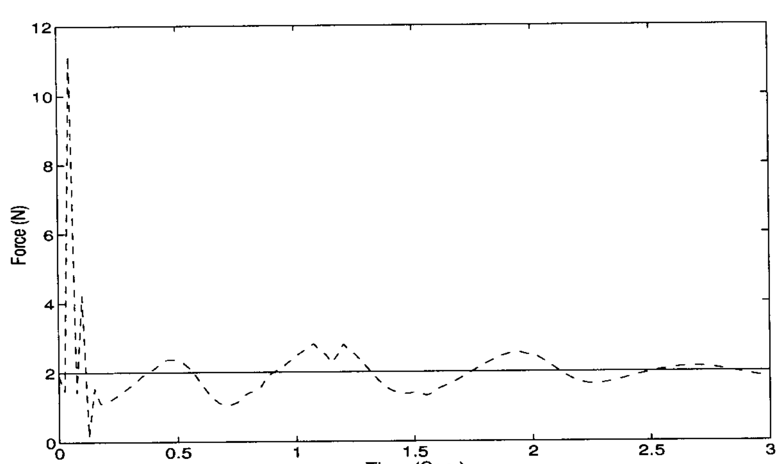

Figure 8. Constraint force tracking under neural network control: (——), $\lambda_d(t)$; (- - - - -), $\lambda(t)$.

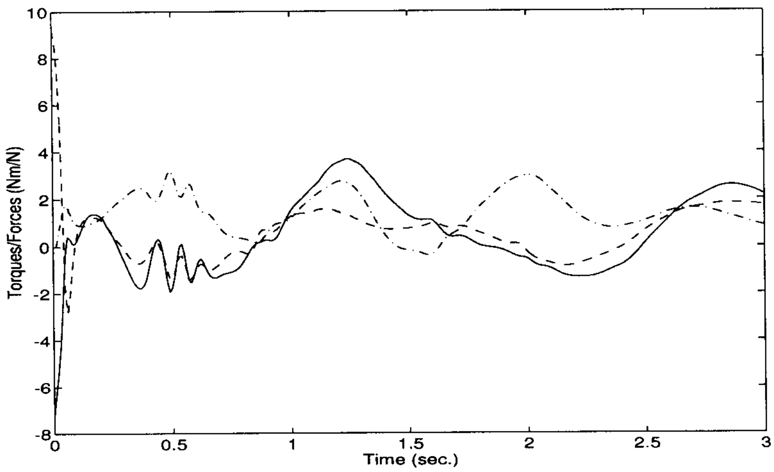

Figure 9. Torques and forces of the manipulators under neural network control: (——), (- - - - -) τ_1; (– · –), τ_2.

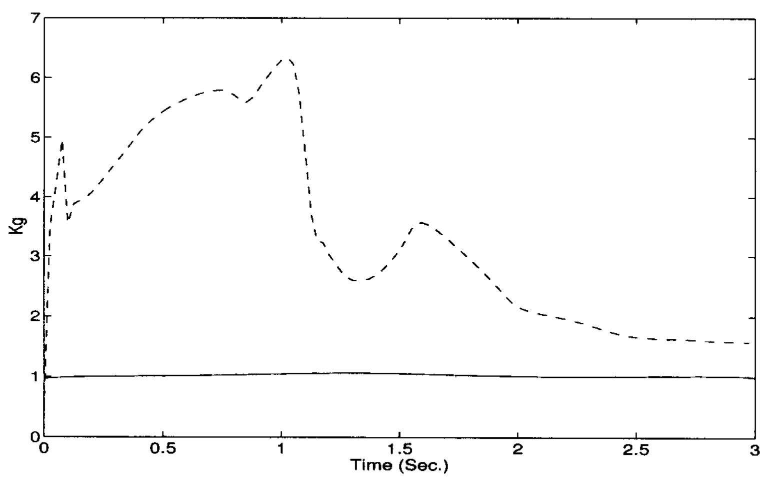

Figure 10. The approximation of M^1: (——), $\|M^1\|$; (- - - - -), $\|\hat{M}^1\|$.

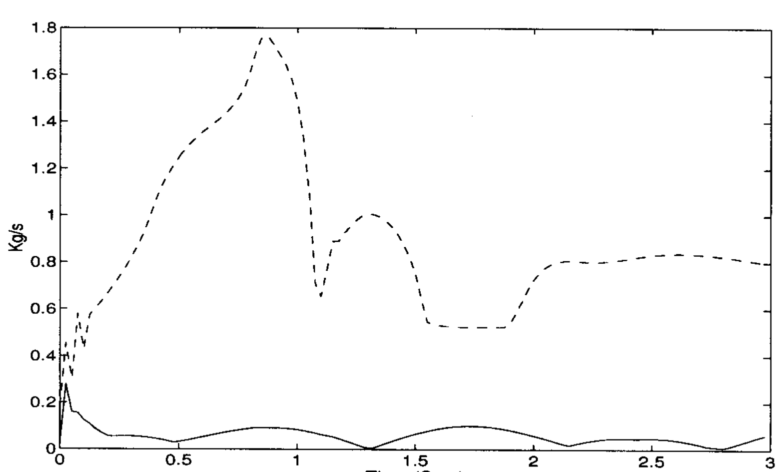

Figure 11. The approximation of C^1: (——), $\|C^1\|$; (- - - - -), $\|\hat{C}^1\|$.

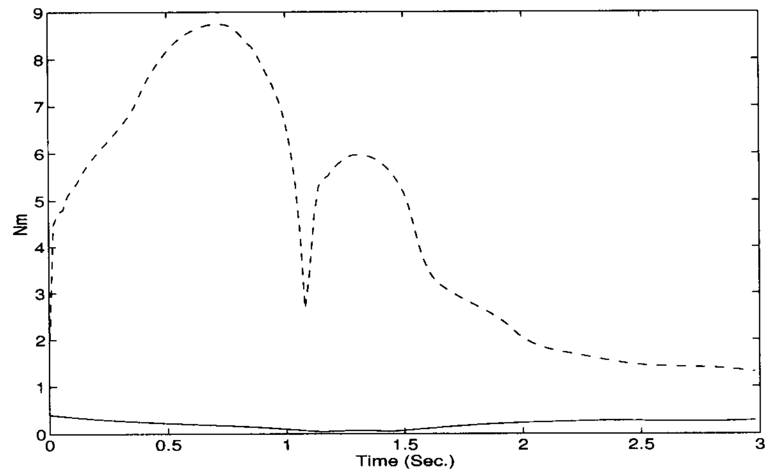

Figure 12. The approximation of G^1: (——), $\|G^1\|$; (- - - - -), $\|\hat{G}^1\|$.

neural-network-based controller involves more matrix manipulations than the model-based adaptive controller does and its computing efficiency is relatively lower.

5. Conclusion

In this paper, dynamic modelling and control of two robotic arms manipulating a constraint object have been discussed. In addition to the motion of the object with respect to the world coordinates, its relative motion with respect to the manipulators is also taken into consideration. The dynamic model of such a system is established and its properties are discussed. Both model-based and neural network based adaptive controllers have been developed which can guarantee the asymptotic convergence of positions and boundedness of the constraint force. The condition for the convergence of the constraint force has also been discussed. Simulation results showed that the proposed controllers work quite well.

References

DESOER, C., and VIDYASAGAR, M., 1975, *Feedback Systems: Input-Output Properties* New York: Academic.

GE, S. S., and POSTLETHWAITE, I., 1994, Non-linear adaptive control of robots including motor dynamics. *Proceedings of the Institution of Mechanical Engineers*, *Part I, Journal of Systems and Control Engineering*, **209,** 89–99; 1996, Adaptive controller design for flexible joint manipulators. *Automatica*, **32,** 273–278.

GE, S. S., CHEN, X. Q., XIE, S., and GU, D. L., 1997, Motion and force control of a Cartesian arm and a rotary table. *Proceedings of the IEEE Singapore International Symposium on Control Theory and Applications*, Singapore, 1997, pp. 286–289.

GE, S. S., LEE, T. H., and HARRIS, C. J., 1998, *Adaptive Neural Network Control of Robotic Manipulators* (Singapore: World Scientific).

HAYKIN, S., 1994, *Neural Networks—A Comprehensive Foundation* (New York: Macmillan College).

KANKAANRANTA, R. K., and KOVIO, H. N., 1988, Dynamics and simulation of compliant motion of a manipulator. *IEEE Journal of Robotics and Automation*, **4,** 163–173.

MCCLAMROCH, N. H., and WANG, D. W., 1988, Feedback stabilization and tracking of constrained robot. *IEEE Transactions on Automatic Control*, **33,** 419–426.

MURRAY, R., LI, Z., and SASTRY, S., 1993, *A Mathematical Introduction to Robotic Manipulation* (Boca Raton, Florida: CRC Press).

RAMADORAI, A. K., TARN, T. J., and BEJCZY, A. K., 1992, Task definition, decoupling and redundancy resolution by nonlinear feedback in multi-robot object handling. Report SSM-RL-92-16, Robotic Laboratory, Washington University.

SADEGH, N., and HOROWITZ, R., 1990, Stability and robustness analysis or a class of adaptive controller for robotic manipulators. *International Journal of Robotics Research*, **9,** 74–92.

SCIAVICCO, L., and SICILIANO, B., 1996, *Modeling and Control of Robot Manipulators* (New York: McGraw-Hill).

SLOTINE, J.-J., and LI, W., 1987, On the adaptive control of robot manipulators. *International Journal of Robotics Research*, **6,** 147–157; 1989, Composite adaptive control of robot manipulators. *Automatica*, **25,** 509–519.

SU, C.-Y., and STEPANENKO, Y., 1995, Hybrid adaptive/robust motion control of rigid-link electrically-driven robot manipulators. *IEEE Transactions on Robotics and Automation*, **11,** 426–432.

UNSEREN, M. A., 1992, A rigid body model and decoupled control architecture for two manipulators holding a complex object. *Robotics and Autonomous Systems*, (10), 115–131.

YAO, B., CHAN, S. P., and WANG, D. W., 1994, Unified formulation of variable structure control schemes for robot manipulators. *IEEE Transactions on Automatic Control*, **39,** 371–376.

YUAN, J., 1996, Composite adaptive control of constrained robots. *IEEE Transactions on Robotics and Automation*, **12,** 640–645.