Robotica (2019) volume 37, pp. 729–755. © Cambridge University Press 2018
doi:10.1017/S0263574718001303

Tracking Control of Electrically Driven Robots Using a Model-free Observer

Alireza Izadbakhsh†*, Saeed Khorashadizadeh‡ and Payam Kheirkhahan†

† *Department of Electrical Engineering, Garmsar Branch, Islamic Azad University, Garmsar, Iran*
E-mail: kheirkhahan_payam@hotmail.com
‡ *Faculty of Electrical and Computer Engineering, University of Birjand, 615/97175 Birjand, Iran*
E-mail: s.khorashadizadeh@birjand.ac.ir

(Accepted November 08, 2018. First published online: December 18, 2018)

SUMMARY
This paper presents a robust tracking controller for electrically driven robots, without the need for velocity measurements of joint variables. Many observers require the system dynamics or nominal models, while a model-free observer is presented in this paper. The novelty of this paper is presenting a new observer–controller structure based on function approximation techniques and Stone–Weierstrass theorem using differential equations. In fact, it is assumed that the lumped uncertainty can be modeled by linear differential equations. Then, using Stone–Weierstrass theorem, it is verified that these differential equations are universal approximators. The advantage of proposed approach in comparison with previous related works is simplicity and reducing the dimensions of regressor matrices without the need for any information of the systems' dynamic. Simulation results on a 6-degrees of freedom robot manipulator driven by geared permanent magnet DC motors indicate the satisfactory performance of the proposed method in overcoming uncertainties and reducing the tracking error. To evaluate the performance of proposed controller in practical implementations, experimental results on an SCARA manipulator are presented.

KEYWORDS: Cable driven parallel manipulator; Differential equations; Electrically driven robot; Observer-based control; Redundant robot; Robust control; Universal approximator.

1. Introduction

Nowadays, the need for intelligent robots that can perform hard and frequent tasks or even the tasks that may jeopardize the health and safety of human is increasingly felt. There is no doubt that motion control and path planning are two essential requirements to produce such intelligent robots. This is the reason that have placed path planning and motion control in the focus of many active researches all over the world.[1–5] The controllers designed in the field of motion control and path planning can be used in coordination control.[6] Finding collision-free trajectories is an important challenge in the problem of robotic motion planning. In ref. [7], a novel algorithm nicknamed "TrajOpt" has been introduced which outperforms previous approaches such as Covariant Hamiltonian Optimization for Motion Planning.

Motion control of robot manipulators is subjected to various structured and unstructured uncertainties. Structured uncertainties are mainly originated from imprecision in link properties, such as the mass center, mass, inertia, and other parameters. Unstructured uncertainties are caused by unmodeled dynamics, such as friction and external disturbances. As a result, model-based controllers[8]

* Corresponding author. E-mail: izadbakhsh_alireza@hotmail.com

https://doi.org/10.1017/S0263574718001303 Published online by Cambridge University Press

are not suitable choices and more advanced control strategies such as robust controllers[9] or adaptive fuzzy controllers[10] are needed to compensate for these various uncertainties.

During the last decades, we have witnessed an increasing trend toward application of fuzzy systems and neural networks in designing robust nonlinear controllers.[11–17] These researches can be considered as alternatives made toward a common objective which is estimation and compensation of uncertainty.[18] The universal approximation property has been an important motivation for this widespread application of fuzzy systems and neural networks. Many improvements have been reported, but the main idea which is estimating and compensating the uncertainties using the universal approximation property of neural networks and fuzzy systems has been remained unchanged. Fault-tolerance, parallelism, and excellent learning capabilities are other beneficial characteristics of fuzzy systems and neural networks. Although these controllers have been practically successful, their design procedure is not simple. In adaptive fuzzy control, there are many tuning parameters such as the center and width of the Gaussian membership functions and also the weight of each rule. Usually, these parameters are adjusted online using the adaptation laws derived from stability analysis. Nevertheless, the initial values of these parameters and their convergence rate are important issues that considerably affect the controller performance and should be selected carefully.[19] Another important issue is sensing requirements.[20] Most of the aforementioned controllers have been designed based on the assumption that the states of the robot such as joint velocities are available. However, many commercial robots provide the user with only joint position measurements. Thus, practical implementation of these controllers may be difficult. To solve this problem, some observer-based control strategies have been presented.[21–25]

Observer-based adaptive fuzzy control has been presented[21] in which a linear state observer is designed to estimate the velocity signals as the inputs of the fuzzy systems. Extended state observer (ESO)[22] is a simple observer in which the lumped uncertainty is considered as an augmented state of the main system and thus is estimated using the same algorithm that other states are estimated. Recently, it has been applied to a mobile robot with the name of active disturbance rejection control.[25] Combination of ESO and sliding mode has been presented in refs. [26–29]. ESO has been applied to nonlinear servomechanisms in ref. [30]. In ref. [31], combination of projection neural networks and ESO for control of autonomous underwater vehicles has been presented. Disturbance observer-based control of nonlinear systems has been studied extensively. After the pioneering works by refs. [32, 33], there have been various applications of this controller to different systems such as disk drivers, machining centers, DC/AC motors, and manipulators.[34–37] Recently, it has been applied to a robotic exoskeleton.[38] In ref. [39], a combination of disturbance observer and adaptive fuzzy systems has been presented. An overview of disturbance observer-based approaches has been reported in ref. [40]. In ref. [41], an adaptive fuzzy observer has been introduced for strict-feedback nonlinear systems.

In this paper, based on the function approximation technique, a robust controller for robot manipulators is designed in which velocity signals are estimated using a simple linear observer. It is assumed that the lumped uncertainty can be modeled by a linear differential equation. In comparison with previous related works,[42] the design procedure for this controller is simpler, since there is no need to determine the regressor matrix or propose a nominal model for manipulator. The superiority of the proposed method over previous controllers based on function approximation techniques (Function Approximation Techniques-based controllers)[43,44] is reducing the dimensions of regressor matrices. It should be emphasized that these regressors are quite different from the regressors describing the system dynamics. The former depends on the type of approximator selected by the designer and is quite known, while the latter is assumed completely unknown and depends on the system dynamics. In comparison with nonlinear disturbance observer-based control,[45] the proposed observer-based controller is superior, since the observer does not require any information from the system dynamics. In ESO,[22] it is assumed that uncertainty is constant or varies slowly, while the proposed method does not need this assumption.

This paper is organized into seven sections. Following the introduction, in Section 2, a dynamic model of the electrically driven robot is presented. Section 3 presents the observer design. Section 4 presents the controller design. Stability analysis is given in Section 5. Section 6 illustrates the simulation and experimental results, and finally, conclusions are presented in Section 7.

https://doi.org/10.1017/S0263574718001303 Published online by Cambridge University Press

2. Problem Formulation and Dynamic Model

The dynamics equation of a robot manipulator is a set of highly nonlinear coupled differential equations. Using the Lagrange–Euler formulation, the dynamic equation of an n-joint robot arm can be expressed as[46]

$$D(q)\ddot{q} + C(q, \dot{q})\dot{q} + G(q) + F(\dot{q}) + T_l(t) = \tau_l \tag{1}$$

where $q, \dot{q}$, and $\ddot{q}$ are, respectively, the $n \times 1$ vectors of joint angular positions, velocities, and accelerations. $D(q)$ is the $n \times n$ symmetric, positive definite inertia matrix, $C(q, \dot{q})\dot{q}$ is the $n \times 1$ centrifugal and Coriolis force vector, $G(q)$ is the $n \times 1$ gravitational force vector, $F(\dot{q})$ is the vector representing the dynamic effects as nonlinear frictions, $T_l(t)$ is the vector representing the bounded torque disturbance, and τ_l is the $n \times 1$ applied torque vector for the joint actuators. When the dynamics of the actuators are also taken into consideration, a similar dynamics relationship for the *integrated actuator-robotic system* can be derived, which is given by[47,48]

$$M(q)\ddot{q} + N(q, \dot{q})\dot{q} + H(q, \dot{q}) = v(t) \tag{2}$$

where

$$\begin{aligned} M(q) &= RK_m^{-1}(J_m r^{-1} + rD(q)) \\ N(q, \dot{q}) &= RK_m^{-1}(B_m r^{-1} + rC(q, \dot{q})) + K_b r^{-1} \\ H(q, \dot{q}) &= RK_m^{-1} r(G(q) + F(\dot{q}) + T_l(t)) \end{aligned}$$

such that R is the $n \times n$ actuator resistance matrix, K_m is the $n \times n$ motor torque constant matrix, J_m is the $n \times n$ diagonal matrix of the lumped actuator rotor inertia, r is the $n \times n$ diagonal matrix representing reduction gear, B_m denotes the $n \times n$ positive definite damping matrix, K_b is the $n \times n$ matrix characterizing the back EMF constants of the actuator, and $v(t)$ is the $n \times 1$ vector of armature voltages applied for the joint actuators, which is limited to *the voltage space*. As it can be seen, system uncertainties are included in (2) that can increase the control system computational time. This problem becomes hypersensitive especially for robot manipulators with higher degree of freedom (DOF). In addition, there are yet other problems. The controller design for such a highly nonlinear system requires information of the position and velocity of the joints. Practically, joint positions can be directly measured by position sensors, such as potentiometer and encoder. If the joint velocity cannot be directly measured, it can be indirectly calculated from derivative mathematical definition. However, this approach is contaminated with the measurement noise.

With this in mind, the control problem can be stated as designing a linear observer and the control input $v(t)$ such that the robots' joints follow the given bounded reference trajectory q_d, in the presence of possibly large (but bounded) and unknown uncertainties in manipulator dynamics and actuator. Also, all involved signals in closed-loop system should remain bounded. Toward this end, Eq. (2) is rewritten as[47]

$$m_{ii}(q)\ddot{q}_i + (N\dot{q})_i + h_i(q, \dot{q}) + \sum_{j=1,\, j\neq i}^{n} m_{ij}(q)\ddot{q}_j = v_i(t) \qquad \{i = 1, 2, ..., n\} \tag{3}$$

where the subscript "i" indicates the ith element, $m_{ii}(q)$ is the varying effective inertia seen at the ith joint, $(N\dot{q})_i$ is the ith element of $N(q, \dot{q})\dot{q}$, $h_i(q, \dot{q})$ is the ith element of $H(q, \dot{q})$, and $v_i(t)$ is the ith element of $v(t)$. In fact, (3) is the ith row of (2). It can easily be shown that (3) can also be rewritten as the following state-space and output form

$$\dot{x} = Ax + Bv_i(t) + B\delta \tag{4}$$

$$y = C^T x \tag{5}$$

where

$$x = [\, q_i \quad \dot{q}_i \,]^T, \quad A = \begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix}, \quad B = \begin{bmatrix} 0 \\ 1 \end{bmatrix}, \quad C^T = [\, 1 \quad 0 \,] \tag{6}$$

https://doi.org/10.1017/S0263574718001303 Published online by Cambridge University Press

$\delta = (1 - m_{ii}(q))\ddot{q}_i - (N\dot{q})_i - h_i(q, \dot{q}) - \sum_{j=1,\, j\neq i}^{n} m_{ij}(q)\ddot{q}_j$ is the residual uncertainty, x is a 2×1 state vector, and y is a 1×1 output vector. It can be easily checked that the above system is minimal, that is, controllable and observable.

Remark 1. Representation (4)–(6) can be interpreted as a standard computed torque controlled system, when there is no knowledge about the actuated robotic manipulator, that is, $\hat{m}_{ii}(q) = 1$ and $\hat{n}_i(q, \dot{q}) = \hat{h}_i(q, \dot{q}) = 0$. This is the most conservative choice with $(\hat{\bullet})$ denoting estimation value of $(\bullet)$.[49] In this manner, the outer-loop control input is equal to actual control input.

3. Observer Design

As mentioned before, many control strategies are designed based on the assumption that the joint velocities are available for feedback. This assumption limits the application of those approaches, because, in many practical situations only the joint positions are measured. Moreover, almost any observer-based control strategies require the model of system for observer design. In this paper, the control strategy is investigated by introducing an observer to reconstruct the robot state vector. Let us define the following linear state observer:

$$\dot{\hat{x}} = A\hat{x} + Bv_i(t) + L(y - \hat{y}) \tag{7}$$

$$\hat{y} = C^T\hat{x} \tag{8}$$

where $\hat{x} = [\hat{q}_i \ \dot{\hat{q}}_i]^T$ is the estimated state vector, L is the observer gain matrix, and C^T is defined by (6). Using (4), (5), (7), and (8), the state estimation error dynamic can be given by

$$\dot{\tilde{x}} = \left(A - LC^T\right)\tilde{x} + B\delta \tag{9}$$

where $\tilde{x} = x - \hat{x}$ is the vector of state estimation error. Since the pair (A, C^T) is observable, the gain L may be selected to give an arbitrary characteristic equation. The main idea behind our definition is that, uncertainties can be approximated by a pth-order ordinary differential equation of the form[48,50]

$$\delta^{(p)} - \sum_{j=1}^{p} b_j \delta^{(p-j)} = 0 \tag{10}$$

where the order p reflects the dynamic structure of δ, and b_j's are real scalars to be designed. For example, consider a third-order linear differential equation $y^{(3)} - b_1 y^{(2)} - b_2 y^{(1)} - b_3 y(t) = 0$ in which $y^{(j)} = \frac{d^j}{dt^j} y(t)$. This equation can be compacted in the form of $y^{(p)} - \sum_{j=1}^{p} b_j y^{(p-j)} = 0$ which is given by (10).

Remark 2. In fact, (10) is a pth-order ordinary differential equation. It can be easily shown that the solution of this equation is a continuous function as

$$\delta(t) = \sum_{i=1}^{p} c_i e^{\lambda_i t} \cos(\omega_i t + \theta_i) \tag{11}$$

In fact, λ_i and ω_i are, respectively, the real and imaginary parts of the ith root of the characteristic polynomials of the differential equation (10). Also, θ_i is a phase shift parameter and c_i depends on the initial condition of the response function. In Appendix A, it has been shown that (11) is the response of the differential equation (10).

It is interesting to investigate the capability of the last assumption (10), from a function approximation capability point of view. Herein, we will prove that (11) has the universal approximation capability. In the following, we suppose that the input universe of discourse T is a convex set in $\Re$. First, we need the following useful theorem.

Stone–Weierstrass Theorem [51]
Let o be a set of real continuous functions on a convex set T. Consider the following conditions:

1. The set o is algebra, that is the set o is closed under multiplication, addition, and scalar multiplication;

https://doi.org/10.1017/S0263574718001303 Published online by Cambridge University Press

2. The set o separates points of T, that is,

$$\forall t_1, t_2 \in T,\ t_1 \neq t_2,\ \exists\, \delta(t) \in o : \delta(t_1) \neq \delta(t_2) \tag{12}$$

3. The set o vanishes at no point of T, that is,

$$\forall t \in T,\ \exists\, \delta(t) \in o : \delta(t) \neq 0 \tag{13}$$

If the aforementioned conditions are satisfied, then for *any real continuous function* $\delta(t)$ on T and arbitrary $\varepsilon > 0$, there exists a function $f(t)$ in o such that

$$\sup_{t \in T} |f(t) - \delta(t)| < \varepsilon \tag{14}$$

Proposition 1. *(Universal Approximation Theorem) Let $\delta(t)$ be a continuous real function on a convex set T in $\Re$. Then, for each arbitrary $\varepsilon > 0$, there exists a function in the form of*

$$\sum_{i=1}^{p} c_i e^{\lambda_i t} \cos(\omega_i t + \theta_i) \tag{15}$$

such that

$$\sup_{t \in T} \left| \sum_{i=1}^{p} c_i e^{\lambda_i t} \cos(\omega_i t + \theta_i) - \delta(t) \right| < \varepsilon \tag{16}$$

Proof. Please refer to Appendix B.

Remark 3. Proposition 1 proves only existence result. That is, we can always find p and b_j coefficients and therefore c_i, ω_i, and λ_i which can approximate *any continuous function* with arbitrary small accuracy, such that (10) can be satisfied. In other words, it implies that there exist p and b_j coefficients that validate (10) for disturbance dynamic approximation. However, appropriate values of order p and differential equation coefficients b_j should be determined mainly through trial and error related to application's type. For instance, if the disturbance is a sinusoid wave of frequency w, $\delta(t) = \cos(wt)$, then $p = 2$, $b_1 = 0$, and $b_2 = w^2$ are recommended. If disturbance is $\delta(t) = t.\cos(wt)$, then $p = 4$, $b_1 = b_3 = 0$, $b_2 = -2w^2$, and $b_4 = -1 - w^4$ are recommended.[50,52] As a further example, consider the differential equation $\delta^{(2)}(t) = 0$ which is satisfied by $\delta(t) = t$. In the first view, it is possible to think that $\delta(t) = t$ is not in the postulated form (11), which is not true. It can be easily shown that, choosing $c_1 = 0$, $\lambda_1 = 0$, $\omega_1 = 0$, $\theta_1 = 0$, $c_2 = c_2'/\omega_2$, $\lambda_2 = 0$, $\theta_2 = -\pi/2$, *and* $\omega_2 = \omega$, results in $\delta(t) = \frac{c_2'}{\omega}\cos(\omega t - \frac{\pi}{2})$. Since $\cos(\omega t - \frac{\pi}{2}) = \sin(\omega t)$, thus $\delta(t) = c_2' \frac{\sin(\omega t)}{\omega}$. Note that, $\lim_{\omega \to 0} \frac{\sin(\omega t)}{\omega} = t$. Thus, we have $\delta(t) \approx c_2' t$. It is worth to recall that, (11) is a universal approximator for the solution of (10). It is not an exact solution.

4. Controller Design

With regard to the former, a robust control strategy is developed that eliminates elaboration and computational burden inherent in conventional joint space control strategy. The following assumption is useful for this.

Assumption 1. *If δ (constant or time varying) represents uncertainties that include inertia, Coriolis/centrifugal forces, gravity force, external disturbances, and un-modeled dynamics, then δ belongs to a compact set T, in other words $\delta \in T$.*

Now, under the assumption of unknown system dynamics structure, and unknown geometric and mass properties, a control law is developed as

$$v_i(t) = -k\hat{x} + k_0 r \tag{17}$$

where k is a vector of gains chosen to correspond to the coefficients of a Hurwitz polynomial (i.e., a polynomial whose roots lie strictly in the open left half of the complex plane), and k_0 and r are scalar gain and input, respectively. Using this control law, (4) becomes

$$\dot{x} = (A - Bk)x + Bk_0 r + Bk\tilde{x} + B\delta \tag{18}$$

https://doi.org/10.1017/S0263574718001303 Published online by Cambridge University Press

Now, we design an algorithm to adjust the control input r such that the tracking error is reduced even in the presence of uncertainties. Suppose that the desired closed-loop state-space equation is defined as

$$\dot{x}_d = (A - Bk)x_d + Bk_0 r_d \tag{19}$$

$$y_d = C^T x_d \tag{20}$$

where r_d and y_d are the desired path for joints and the desired output in the joint space, respectively. Once the motion specification is given in terms of the desired trajectory r_d, then the motion control objective is to achieve

$$\lim_{t\to\infty} \xi(t) = 0, \quad \lim_{t\to\infty} \tilde{x}(t) = 0 \tag{21}$$

where

$$\xi(t) = y(t) - y_d(t) \tag{22}$$

denotes the position tracking error. With this in mind, and definition of $e(t) = x - x_d$, the system error dynamics can then be expressed as

$$\dot{e}(t) = (A - Bk)e(t) + Bk_0\psi(t) + Bk\tilde{x} + B\delta \tag{23}$$

$$\xi(t) = C^T e(t) \tag{24}$$

where

$$\psi(t) = r(t) - r_d(t) \tag{25}$$

Assumption 2. *The desired reference trajectory r_d is assumed to be bounded and uniformly continuous and has bounded and uniformly continuous derivatives up to a necessary order.*[46]

To determine the influence of input $\psi(t)$ on the tracking error, (23) and (24) are differentiated p times to obtain

$$\dot{E}(t) = (A - Bk)E(t) + Bk_0\vartheta(t) + Bk\wp(t) + B\text{ƛ}(t) \tag{26}$$

$$\xi^{(p)}(t) = C^T E(t) + \sum_{j=1}^{p} b_j \xi^{(p-j)} \tag{27}$$

where

$$E(t) = e^{(p)} - \sum_{j=1}^{p} b_j e^{(p-j)} \tag{28}$$

$$\vartheta(t) = \psi^{(p)} - \sum_{j=1}^{p} b_j \psi^{(p-j)} \tag{29}$$

and

$$\wp(t) = \tilde{x}^{(p)} - \sum_{j=1}^{p} b_j \tilde{x}^{(p-j)} \tag{30}$$

is the solution of

$$\dot{\wp}(t) = (A - LC^T)\wp(t) + B\text{ƛ}(t) \tag{31}$$

Note that $\text{ƛ}(t)$ is the lumped approximation error. It has been assumed that uncertainties can be approximated by a pth-order ordinary differential equation of the form (10). In fact, uncertainty is not always the exact solution of (10). As a result, there exist small differences between the actual

https://doi.org/10.1017/S0263574718001303 Published online by Cambridge University Press

and approximated function denoted by $\dot{\pi}(t)$. According to these, suppose that $\delta = \delta' + \delta''$. Hence, differentiating the last relation gives $\delta'^{(p)} - \sum_{j=1}^{p} b_j \delta'^{(p-j)} + \delta''^{(p)} - \sum_{j=1}^{p} b_j \delta''^{(p-j)} = 0$.

Suppose, $\delta'^{(p)} - \sum_{j=1}^{p} b_j \delta'^{(p-j)} = 0$ is the estimation of $\delta(t)$ such that (16) is satisfied. Thus, we have $\delta''^{(p)} - \sum_{j=1}^{p} b_j \delta''^{(p-j)} = \dot{\pi}(t)$ which is called the residual (lumped) uncertainty.

Now, let us define the following coordinate transformation

$$Z = [\xi \ \dot{\xi} \ \ldots \ \xi^{(p-1)} \ E^T(t)]^T \tag{32}$$

Then, we can write the system state equations in new coordinates as

$$\dot{Z} = \Lambda_0 Z + \kappa_0 \vartheta(t) + \Psi_0 \wp(t) + \Xi(t) \tag{33}$$

$$y = HZ \tag{34}$$

where

$$\Lambda_0 = \begin{bmatrix} 0 & 1 & 0 & \cdots & 0 & 0 \\ 0 & 0 & 1 & \cdots & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & \cdots & \cdots & 1 & 0 \\ b_p & b_{p-1} & \cdots & \cdots & b_1 & C^T \\ 0 & 0 & \cdots & \cdots & 0 & A - Bk \end{bmatrix} \tag{35}$$

$$\kappa_0 = \left[0 \cdots 0 \ (Bk_0)^T\right]^T, \qquad \Xi(t) = \left[0 \cdots 0 \ (B\dot{\pi}(t))^T\right]^T$$

$$H = \left[1 \ 0 \cdots 0\right], \qquad \Psi_0 = \left[0 \cdots 0 \ (Bk)^T\right]^T$$

and $\vartheta(t)$ is defined as

$$\vartheta(t) = -\sum_{j=1}^{p} \mu_j \xi^{(p-j)} - \mu_0 \hat{E}(t) \tag{36}$$

where $\hat{E}(t) = \hat{e}^{(p)} - \sum_{j=1}^{p} b_j \hat{e}^{(p-j)}$. With the definition of $\hat{e} = e - \tilde{x}$, (36) can be rewritten in the form of

$$\vartheta(t) = -\sum_{j=1}^{p} \mu_j \xi^{(p-j)} - \mu_0 E(t) + \mu_0 \wp(t) = -\mu Z + \mu_0 \wp(t) \tag{37}$$

Now, substituting (37) into (33) yields

$$\dot{Z} = \Lambda Z + \kappa \wp(t) + \Xi(t) \tag{38}$$

where

$$\Lambda = (\Lambda_0 - \kappa_0 \mu), \quad \kappa = (\kappa_0 \mu_0 + \Psi_0) \tag{39}$$

The final step is to adjust the control input in (25) to compensate for the effects of modeling errors and external disturbance as follows:

$$r(t) = r_d(t) + \psi(t) \tag{40}$$

where $\psi(t)$ is the auxiliary control input. The last equation says that in order to obtain a high accuracy motion in response to unwanted uncertainties; the control input must be changed into to (40).

https://doi.org/10.1017/S0263574718001303 Published online by Cambridge University Press

More precisely, robot should be controlled to follow the modified signal $r(t)$, instead of $r_d(t)$. If this modification is violated, the error may never converge and driving terms appear in the right-hand side of (31) and (38).

Remark 4. The design procedure can be summarized into the following steps:

1. Calculate k such that $A - Bk$ is Hurwitz.
2. Set $k_0 = k(1, 1)$.
3. Select p (the order of differential equation).
4. Based on Remark 3, calculate b_j.
5. Calculate L such that $A - LC^T$ is Hurwitz. Note that the observer poles should be faster than the plant poles $(A - Bk)$.
6. Calculate μ such that $\Lambda = (\Lambda_0 - \kappa_0\mu)$ in (39) is Hurwitz.
7. Obtain $\psi(t)$ by (29) and (36). In other words,

$$\psi^{(p)} - \sum_{j=1}^{p} b_j \psi^{(p-j)} = -\sum_{j=1}^{p} \mu_j \xi^{(p-j)} - \mu_0 \hat{E}(t).$$

8. Use the observer given in (7) and (8) to estimate the system states.
9. Apply the control law (17) to the system by considering (40).

5. Stability Analysis and Performance Evaluation

To study the stability and analyze performances of closed-loop system, let us choose the following Lyapunov candidate function:

$$V = Z^T P Z + \frac{1}{2} \wp(t)^T P_0 \wp(t) \tag{41}$$

where $P = P^T > 0$ and $P_0 = P_0^T > 0$ are the solutions, for given $Q = Q^T > 0$, and $Q_0 = Q_0^T > 0$, of the Lyapunov equations

$$\Lambda^T P + P\Lambda = -Q \tag{42}$$

$$(A - LC^T)^T P_0 + P_0(A - LC^T) = -Q_0 \tag{43}$$

Differentiation of (41) along (31) and (38) yields

$$\dot{V} = -Z^T Q Z - \frac{1}{2} \wp(t)^T Q_0 \wp(t) + 2Z^T P \kappa \wp(t) + 2Z^T P \Xi + \wp(t)^T P_0 B \dot{\chi}(t) \tag{44}$$

Further, using the following definition

$$Q' = \begin{bmatrix} Q & -P\kappa \\ -P\kappa & \frac{1}{2}Q_0 \end{bmatrix}, \qquad P' = \begin{bmatrix} P & 0 \\ 0 & 0.5P_0 \end{bmatrix} \tag{45}$$

Equation (44) can be rewritten as

$$\begin{aligned} \dot{V} &= -\begin{bmatrix} Z^T & \wp(t)^T \end{bmatrix} Q' \begin{bmatrix} Z \\ \wp(t) \end{bmatrix} + 2\begin{bmatrix} Z^T & \wp(t)^T \end{bmatrix} P' \begin{bmatrix} \Xi \\ B\dot{\chi} \end{bmatrix} \\ &\leq -\lambda_{\min}(Q') \left\| \begin{bmatrix} Z \\ \wp(t) \end{bmatrix} \right\|^2 + 2\lambda_{\max}(P') \left\| \begin{bmatrix} Z \\ \wp(t) \end{bmatrix} \right\| \left\| \begin{bmatrix} \Xi \\ \dot{\chi} \end{bmatrix} \right\| \end{aligned} \tag{46}$$

where $\lambda_{\min}(Q')$ and $\lambda_{\max}(P')$ denote the minimum and maximum eigenvalues of Q' and P', respectively. The two following conditions guarantee that the matrix Q' is positive definite.

$$\begin{aligned} &\lambda_{\min}(Q) > 0 \\ &\lambda_{\min}(Q)\lambda_{\min}(Q_0) > 2\lambda_{\max}^2(P\kappa) \end{aligned} \tag{47}$$

https://doi.org/10.1017/S0263574718001303 Published online by Cambridge University Press

Remark 5. Suppose that appropriate models are used and the approximation error can be ignored. Then, $\dot{V}$ is negative definite, and asymptotic stability of the closed-loop system can be approved.

Remark 6. If the approximation error cannot be ignored, utilizing further manipulations of (46) we have

$$\begin{aligned}\dot{V} \leq & -\frac{1}{2}\left(\sqrt{\lambda_{\min}(Q')}\left\|\begin{bmatrix} Z \\ \wp(t)\end{bmatrix}\right\| - 2\frac{\lambda_{\max}(P')}{\sqrt{\lambda_{\min}(Q')}}\left\|\begin{matrix}\Xi \\ \dot{\chi}\end{matrix}\right\|\right)^2 \\ & -\frac{1}{2}\left(\lambda_{\min}(Q')\left\|\begin{bmatrix} Z \\ \wp(t)\end{bmatrix}\right\|^2 - 4\frac{\lambda^2_{\max}(P')}{\lambda_{\min}(Q')}\left\|\begin{matrix}\Xi \\ \dot{\chi}\end{matrix}\right\|^2\right)\end{aligned} \tag{48}$$

In other words, we have

$$\dot{V} \leq \underbrace{-\frac{1}{2}\lambda_{\min}(Q')\left\|\begin{bmatrix} Z \\ \wp(t)\end{bmatrix}\right\|^2}_{c} + 2\frac{\lambda^2_{\max}(P')}{\lambda_{\min}(Q')}\left\|\begin{matrix}\Xi \\ \dot{\chi}\end{matrix}\right\|^2 \tag{49}$$

We would like to relate (c) to V by considering

$$V \leq \lambda_{\max}(P')\left\|\begin{bmatrix} Z \\ \wp(t)\end{bmatrix}\right\|^2 \tag{50}$$

Now, (49) can be further derived as

$$\dot{V} \leq -\alpha V + \left[\alpha\lambda_{\max}(P') - \frac{1}{2}\lambda_{\min}(Q')\right]\left\|\begin{bmatrix} Z \\ \wp(t)\end{bmatrix}\right\|^2 + 2\frac{\lambda^2_{\max}(P')}{\lambda_{\min}(Q')}\left\|\begin{matrix}\Xi \\ \dot{\chi}\end{matrix}\right\|^2 \tag{51}$$

Pick $\alpha < \frac{\lambda_{\min}(Q')}{2\lambda_{\max}(P')}$, then we have

$$\dot{V} \leq -\alpha V + 2\frac{\lambda^2_{\max}(P')}{\lambda_{\min}(Q')}\left\|\begin{matrix}\Xi \\ \dot{\chi}\end{matrix}\right\|^2 \tag{52}$$

As a result, $\dot{V} < 0$ whenever

$$\left(\xi, \dot{\xi}, ..., \xi^{(p-1)}, E(t), \wp(t)\right) \in F \equiv \left\{V > 2\frac{\lambda^2_{\max}(P')}{\alpha\lambda_{\min}(Q')}\sup_{\tau \geq t_0}\left\|\begin{matrix}\Xi(\tau) \\ \dot{\chi}(\tau)\end{matrix}\right\|^2\right\} \tag{53}$$

This implies that $\left(\xi, \dot{\xi}, ..., \xi^{(p-1)}, E(t), \wp(t)\right)$ is uniformly ultimately bounded. Note that the size of the set $F$is adjustable by proper selection of α, P', and Q'. Smaller size of Fimplies more accurate in output tracking.

Until now, we only demonstrate the boundedness of the closed-loop system, but in practical implementations, the transient performance is also of great importance. Solving (52) yields

$$V(t) \leq e^{-\alpha(t-t_0)}V(t_0) + 2\frac{\lambda^2_{\max}(P')}{\alpha\lambda_{\min}(Q')}\sup_{t_0 \leq \tau \leq t}\left\|\begin{matrix}\Xi(\tau) \\ \dot{\chi}(\tau)\end{matrix}\right\|^2 \tag{54}$$

Therefore, considering (54) and the lower bound of (41) we have

$$\left\|\begin{bmatrix} Z \\ \wp(t)\end{bmatrix}\right\| \leq \sqrt{\frac{V(t_0)}{\lambda_{\min}(P')}}e^{\frac{-\alpha(t-t_0)}{2}} + \sqrt{\frac{2\lambda^2_{\max}(P')}{\alpha\lambda_{\min}(P')\lambda_{\min}(Q')}}\sup_{t_0 \leq \tau \leq t}\left\|\begin{matrix}\Xi(\tau) \\ \dot{\chi}(\tau)\end{matrix}\right\| \tag{55}$$

https://doi.org/10.1017/S0263574718001303 Published online by Cambridge University Press

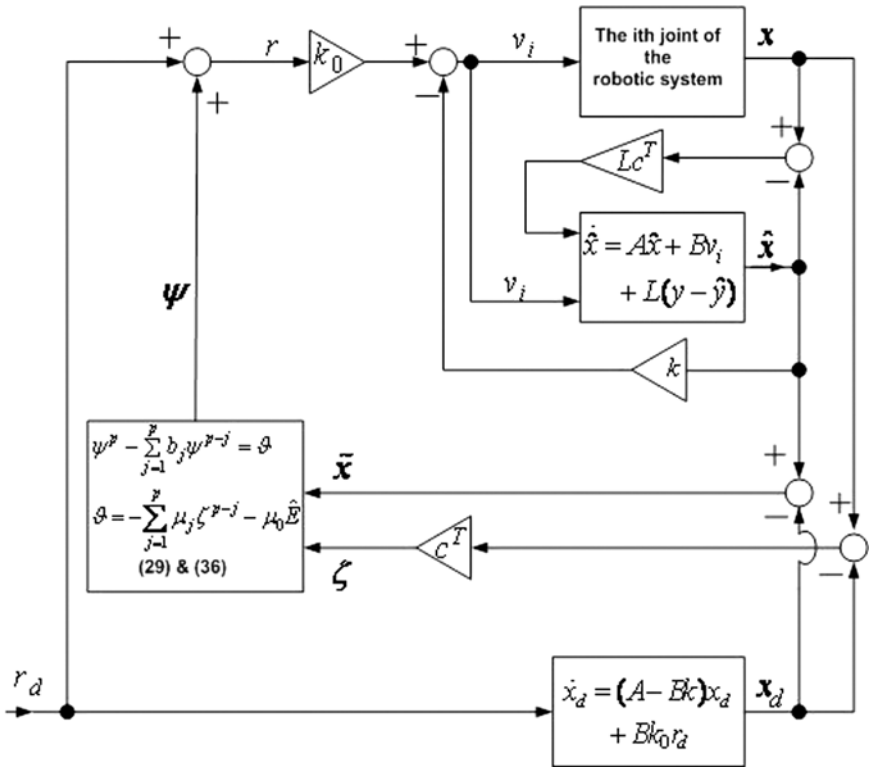


Fig. 1. Block diagram of the proposed observer–controller structure.

that means tracking error is bounded by a weighted exponential function plus a constant. This also implies that by adjusting controller parameters, we may improve output error convergence rate. As a consequence,

$$\lim_{t\to\infty} \left\| \begin{bmatrix} Z \\ \wp(t) \end{bmatrix} \right\| \le \sqrt{\frac{2\lambda_{\max}^2(P')}{\alpha \lambda_{\min}(P')\lambda_{\min}(Q')}} \sup_{t_0 \le \tau \le t} \left\| \begin{matrix} \Xi(\tau) \\ \dot{\lambda}(\tau) \end{matrix} \right\| \tag{56}$$

6. Simulation/Experimental Results

To verify the effectiveness of the proposed control strategy, both computer simulations and experiments are performed. In Section 6.1, simulation results are presented. In Section 6.2, the experimental setup is introduced. Experimental results are shown in Section 6.3.

6.1. Simulation results

In this section, simulation results for a 6-DOF manipulator actuated by geared permanent magnet DC motors are illustrated to verify the satisfactory performance of the proposed controller. The block diagram of the observer–controller structure is illustrated in Fig. 1. The arm which consists of six joints is used to perform the proposed control law. The proposed control scheme is simulated using a Puma560 robot manipulator with the symbolic representation shown in ref. [53]. The Puma robot uses two different sizes of motors – one for the base axes (joints 1–3) and the other for the wrist axes (joints 4–6).[54] The Denavit–Hartenberg parameters are given in ref. [53]. The actuator/robot dynamical parameters are detailed in ref. [55]. The actuator parameters are selected as

$B = diag(1.48 \times 10^{-3},\ 817 \times 10^{-6},\ 1.38 \times 10^{-3},\ 71.2 \times 10^{-6},\ 82.6 \times 10^{-6},\ 36.7 \times 10^{-6})$ [N · m · s/rad], $L = diag(10^{-4},\ 10^{-4},\ 10^{-4},\ 2 \times 10^{-4},\ 2 \times 10^{-4},\ 2 \times 10^{-4})$ [H],
$R = diag(2.1, 2.1, 2.1, 6.7, 6.7, 6.7)$ [Ω],
$J_m = diag(2 \times 10^{-4},\ 2 \times 10^{-4}, 2 \times 10^{-4},\ 33 \times 10^{-6},\ 33 \times 10^{-6},\ 33 \times 10^{-6})$[kg · m^2],
$K_b = K_m = diag(0.189, 0.219, 0.202, 0.075, 0.066, 0.066)$ [N · m/A], and
$r = diag(0.02, 0.02, 0.02, 0.02, 0.02, 0.02)$

https://doi.org/10.1017/S0263574718001303 Published online by Cambridge University Press

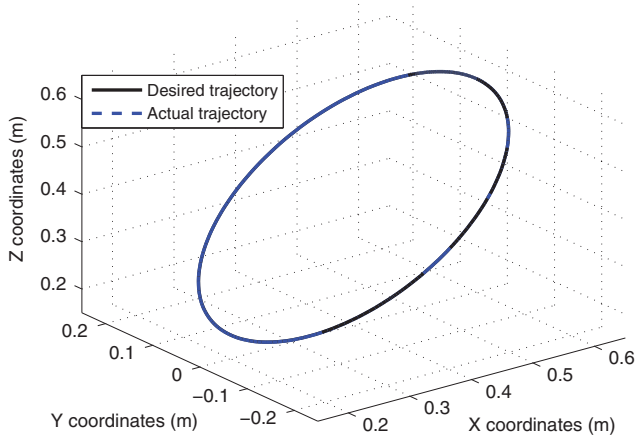


Fig. 2. Trajectory tracking.

The maximum voltage of each motor is set to $u_{\max} = 50$ V. It should be noted that we utilized the actual model of the actuators, describing an interplay of mechanical and electrical part of the motor considering the inductance term (L), during the simulations.

6.1.1. Simulation 1: a simple trajectory. The desired task consists in tracking a circle of radius 0.2 m centered at coordinate (0.4 m, 0, 0.4 m) of the operational space specified by

$$\begin{cases} x_d(t) = 0.2\cos(\pi t) + 0.4 \\ y_d(t) = 0.2\sin(\pi t) \\ z_d(t) = 0.2\cos(\pi t) + 0.4 \end{cases} \tag{57}$$

We have chosen roll, pitch, and yaw angles for representing the orientation of the end-effector. The transformation matrix is given by

$$[R] = R_z(\alpha)R_y(\beta)R_x(\gamma) = \begin{bmatrix} c_\alpha c_\beta & c_\alpha s_\beta s_\gamma - s_\alpha c_\gamma & c_\alpha s_\beta c_\gamma + s_\alpha s_\gamma \\ s_\alpha c_\beta & s_\alpha s_\beta s_\gamma + c_\alpha c_\gamma & s_\alpha s_\beta c_\gamma - c_\alpha s_\gamma \\ -s_\beta & c_\beta s_\gamma & c_\beta c_\gamma \end{bmatrix} \tag{58}$$

where c stands for the cosine function, s represents the sine function, $R_x(\gamma)$, $R_y(\beta)$, and $R_z(\alpha)$ are the basic rotation matrices around the three coordinate axes, respectively. Rotation around the axis Z by an angle of α is called roll. Also, rotation around the axis Y by an angle β is called pitch and rotation around the axis X by an angle of γ is called yaw. Since the present study is devoted to joint space tracking control, it is necessary to translate the desired operational space trajectory in (57) to a suitable desired joint space position trajectory. Such a conversion is usually done by using the inverse kinematics map. Once obtained such a conversion, it is possible to implement a joint space controller so that (21) can be satisfied. The controller parameters are selected as $k = [8000\ 80]$, $L = [3\ 2000] \times 10^3$, $\mu = [249.56\ 5.24\ 0.03]$, and $p = 1$ for each joint. The external disturbance is defined as

$$T_{l_i}(t) = 10\sin\left(\frac{50\pi}{9}t\right) + 5\sin(12.5\pi t) + 5\sin\left(\frac{100\pi}{29}t\right) - 3 \tag{59}$$

which is applied to all motors. According to these, the results of simulations are shown in Figs. 2–7. The tracking response of the end-effector is shown in Fig. 2, in which the real line denotes the desired tracking trajectory and the dash line denotes the actual tracking trajectory. Note that the actual position of the endpoint in operational space is obtained by means of the forward kinematic of the Puma560 robot arm. As shown in this figure, the robot's end-effector can successfully follow circle trajectory without the need for velocity feedback, and the system stability is guaranteed, even in the case where no knowledge of the system is in hand. The norm of tracking errors shows ignorable values in the range of 1.5×10^{-3}(rad) as shown in Fig. 3. Figure 4 shows the norm of

https://doi.org/10.1017/S0263574718001303 Published online by Cambridge University Press

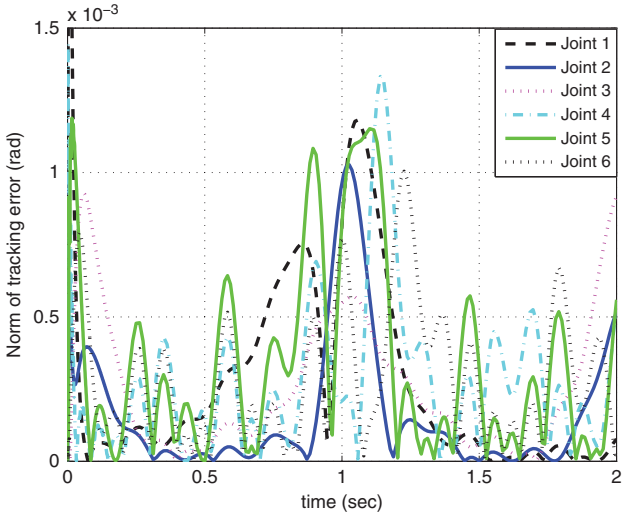


Fig. 3. Norm of tracking error.

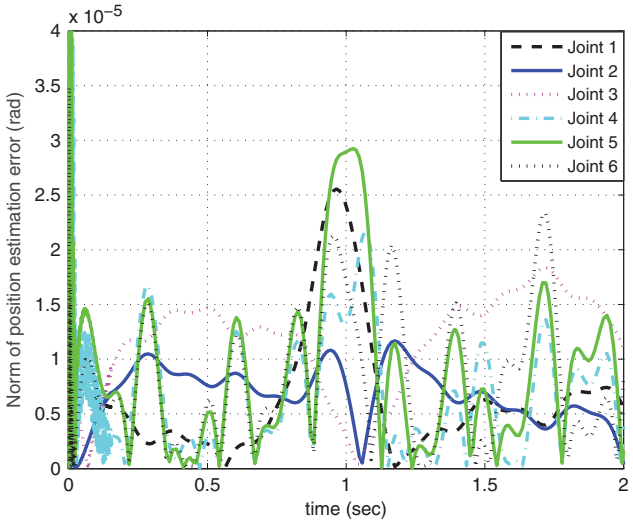


Fig. 4. Norm of position estimation error.

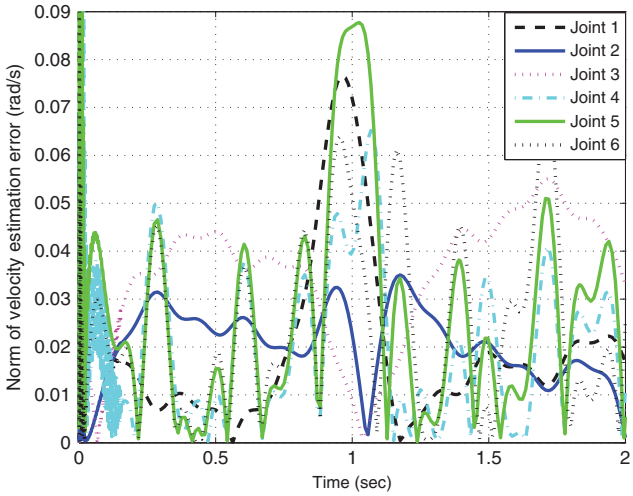


Fig. 5. Norm of velocity estimation error.

https://doi.org/10.1017/S0263574718001303 Published online by Cambridge University Press

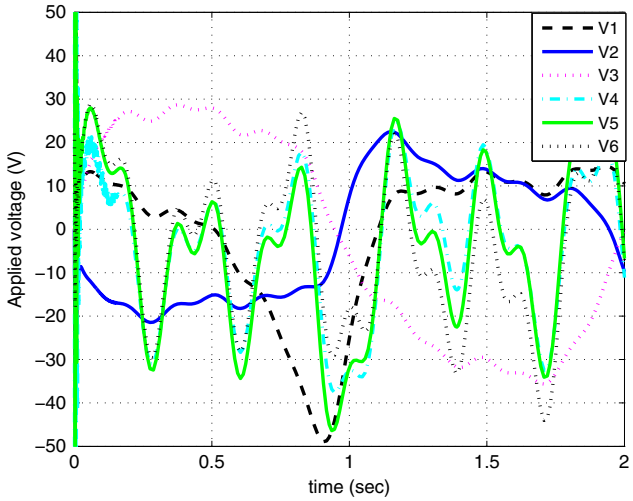


Fig. 6. Applied voltages.

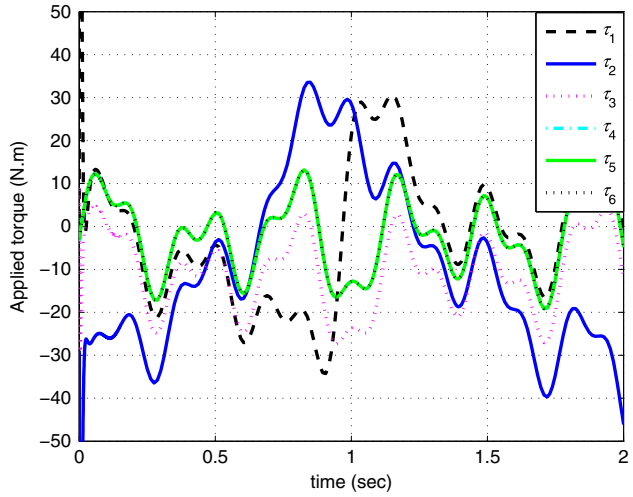


Fig. 7. Applied torques.

position estimation error. According to this figure, the steady state value of position estimation error is in the range of 3×10^{-5}(rad) which is negligible. Norm of estimated velocity errors is also plotted in Fig. 5. As shown in this figure, the steady state value of velocity estimation error is less than 0.09 (rad/s). Overall, it seems that the proposed observer–controller structure performs satisfactorily. Figure 6 shows the applied voltage for the joint actuators. According to this figure, the control law is smooth and does not contain very high frequencies. Finally, Fig. 7 shows the applied torques on the actuators.

For comparison purpose, the proposed controller is compared to an extended form of the work presented by ref. [56] that utilizes Chebyshev neural network (CNN) to estimate the uncertainties and state estimation. Recently, it is proved that CNN has powerful representation capabilities whose input is generated by using a subset of Chebyshev polynomials.[57] The detailed description of CNN and its approximation properties can be found in ref. [58]. Under these circumstances, for the proposed CNN used to estimate the uncertainties, the joint angles, joint velocities, and joint accelerations are required as the input. The number of polynomials selected for functional expansion for each input is 1. Thus, the basis vectors Z_o and Z_c are of dimension $\Re^7$ and $\Re^{19}$, respectively. The initial weights of the CNN are taken to be zero. The control parameters are selected as ref. [20]. Under this setting, Fig. 8 shows tracking response of the end-effector. Figure 9 represents the norm of tracking error. Figure 10 shows the norm of velocity estimation error. Finally, Fig. 11 shows the applied voltages to the actuators. As can be seen, both CNN-based control scheme and model-free control

https://doi.org/10.1017/S0263574718001303 Published online by Cambridge University Press

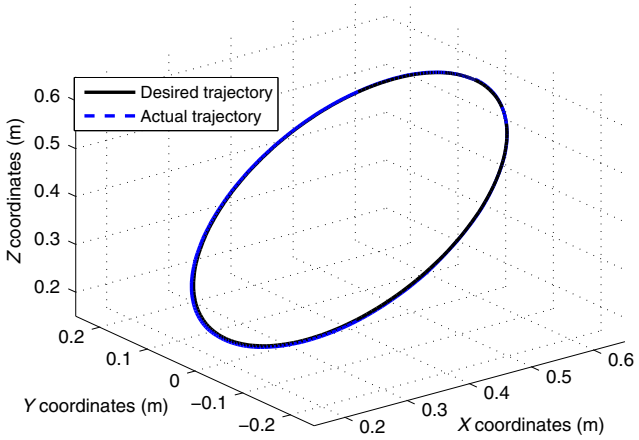


Fig. 8. Trajectory tracking (CNN).

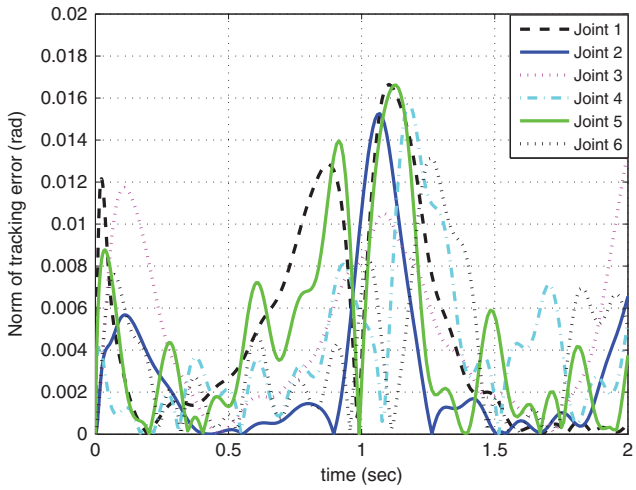


Fig. 9. Norm of tracking error (CNN).

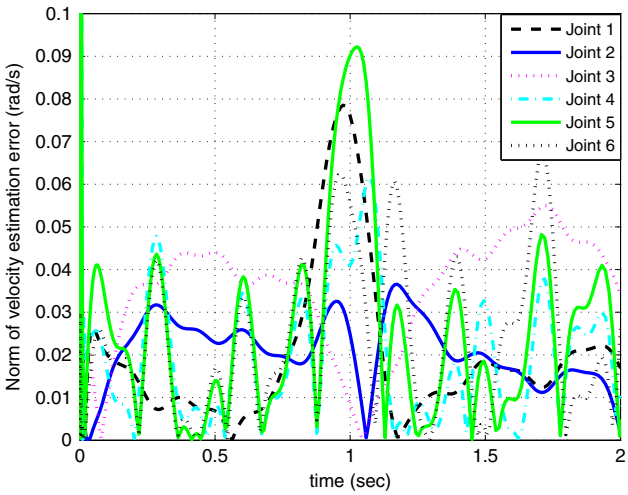


Fig. 10. Norm of velocity estimation error (CNN).

https://doi.org/10.1017/S0263574718001303 Published online by Cambridge University Press

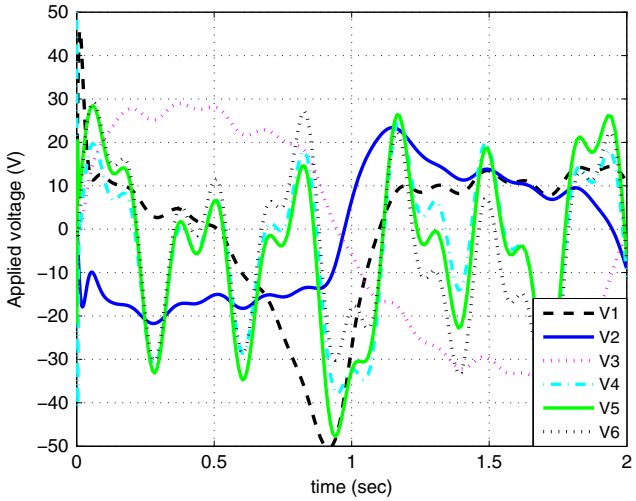


Fig. 11. Applied voltages (CNN).

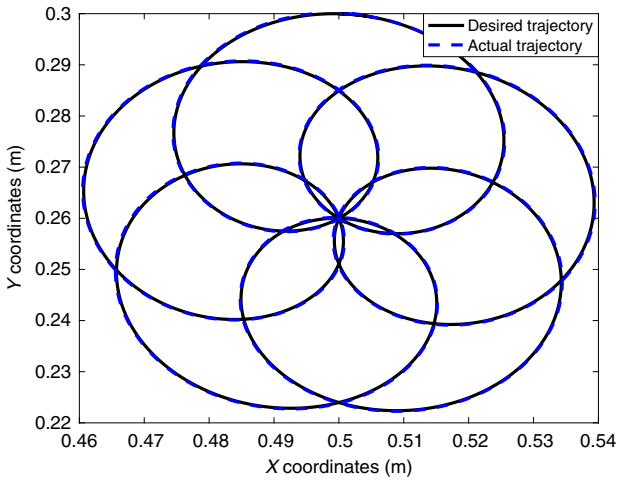


Fig. 12. Tracking performance for the complicated trajectory.

approach give the same result because of universal approximation property of Neural-Network (NN) and differential equations. Being a single-layer neural network, the computational complexity of CNN is less intensive as compared to Multi Layer Perceptron perceptron and can be used for on-line learning.[57] However, the model-free control scheme is much simpler and less computational than CNN, since it does not require various feedbacks as their inputs. The reason is that these estimators consider the uncertainty as a time-varying function rather than a function of different state variable.

6.1.2. Simulation 2: complicated trajectory. Now, consider a fast and complex trajectory specified by

$$\begin{cases} x_d(t) = 0.5 + 0.02\sin(2t) + 0.02\sin(12t) \\ y_d(t) = 0.26 + 0.02\sin(2t + 1.6) + 0.02\sin(12t + 1.6) \\ z_d(t) = 0 \end{cases} \tag{60}$$

The tracking response of the end-effector is shown in Fig. 12 in which the real line denotes the desired tracking trajectory and the dash line denotes the actual tracking trajectory. As shown in this figure, the proposed method is capable of tracking fast and complicated trajectories with high accuracy. The norm of tracking errors is illustrated in Fig. 13 which shows small values in the range of 0.8×10^{-3}(rad). Norms of position estimation errors are also plotted in Fig. 14. According to

https://doi.org/10.1017/S0263574718001303 Published online by Cambridge University Press

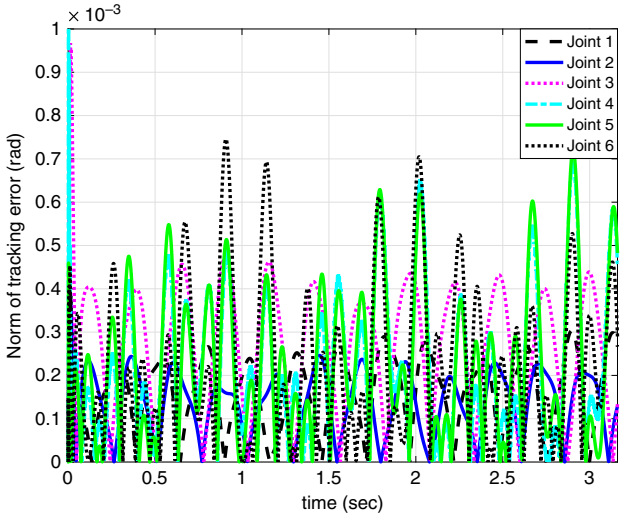


Fig. 13. Norm of tracking errors for the complicated trajectory.

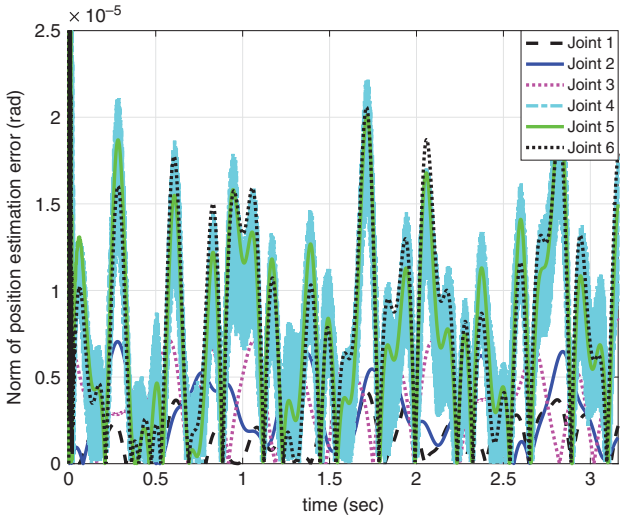


Fig. 14. Norm of position estimation errors for the complicated trajectory.

this figure, the steady state value of position estimation error is in the range of 2.5×10^{-5} (rad) which implies good tuning of the observer parameters. Norms of estimated velocity errors are also presented in Fig. 15. As shown in this figure, the steady state value of velocity estimation error is less than 0.07 (rad/s). Overall, it seems that the proposed observer–controller structure can track complicated trajectories very well. Motor voltages are given in Fig. 16. As it can be seen, the control laws do not exceed the maximum permitted value and the chattering phenomenon is not observed. Finally, Fig. 17 illustrates the applied torques to the joints. The torques are bounded and smooth. As a further work, simulation results related to the extended form of the work presented by ref. [56], using CNN, have been shown in Figs. 18–21. It seems that the results are nearly similar. Nevertheless, further tuning of the controller parameters may improve these results.

6.1.3. Simulation 3: redundant manipulators. In order to test the performance of the proposed method on more complicated robotic systems, it has been applied to a redundant manipulator. A manipulator is termed kinematically redundant when it possesses more DOFs than it is needed to execute a given task. Redundancy can be conveniently exploited to achieve more dexterous robot motions. The inverse kinematic problem is of particular interest in this case since it admits infinite solutions. According to these, simulation results for a cable driven redundant parallel manipulator

https://doi.org/10.1017/S0263574718001303 Published online by Cambridge University Press

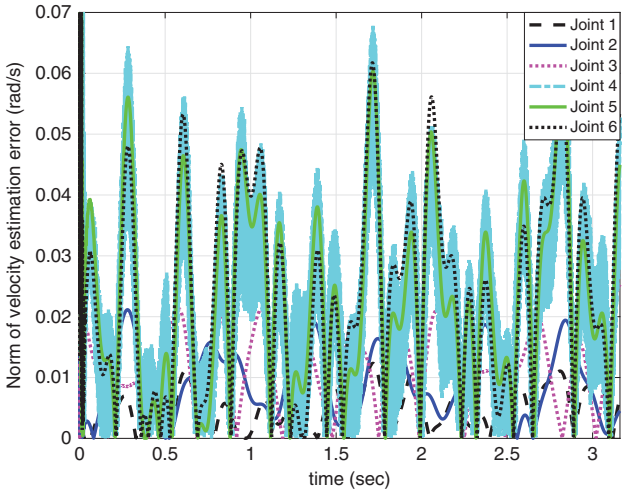


Fig. 15. Norm of velocity estimation errors for the complicated trajectory.

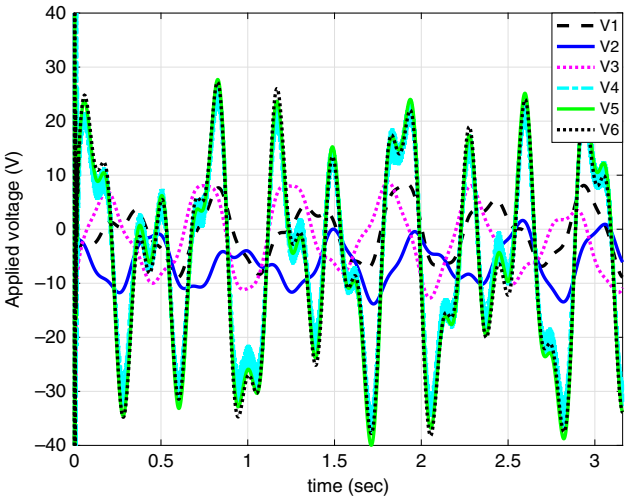


Fig. 16. Control efforts for the complicated trajectory.

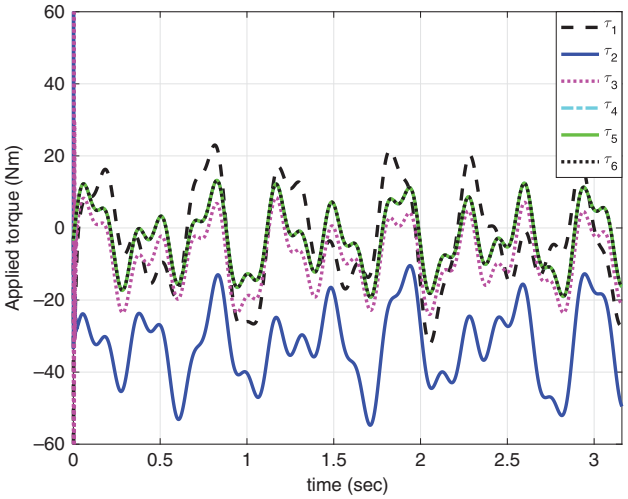


Fig. 17. Applied torques for the complicated trajectory.

https://doi.org/10.1017/S0263574718001303 Published online by Cambridge University Press

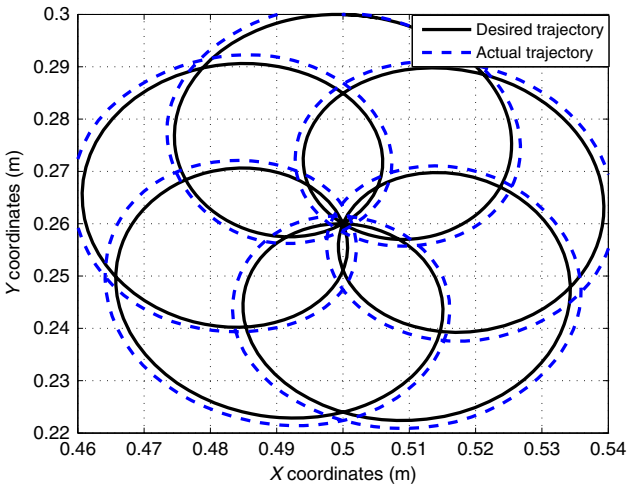


Fig. 18. Tracking performance for the complicated trajectory (CNN).

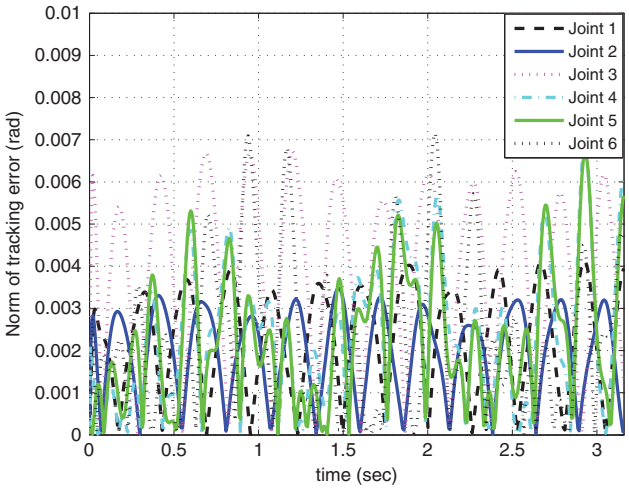


Fig. 19. Norm of tracking errors for the complicated trajectory (CNN).

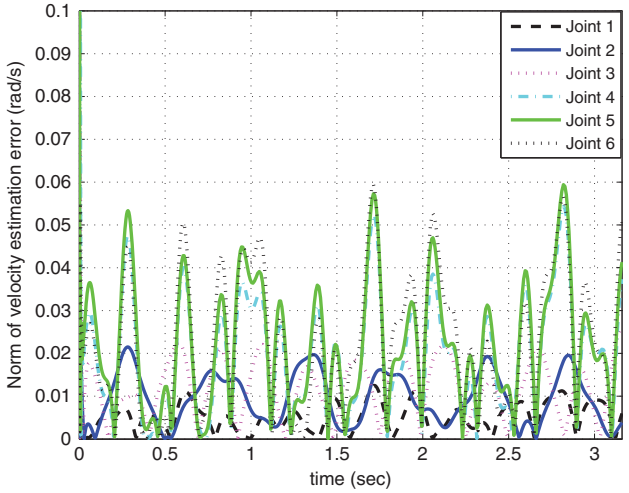


Fig. 20. Norm of velocity estimation errors for the complicated trajectory (CNN).

https://doi.org/10.1017/S0263574718001303 Published online by Cambridge University Press

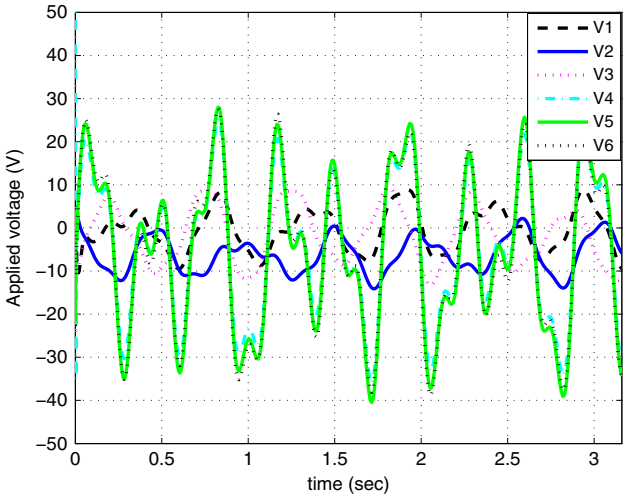


Fig. 21. Applied voltages (CNN).

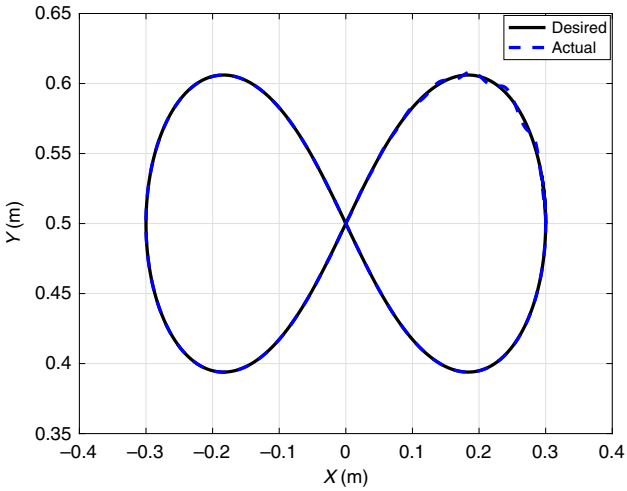


Fig. 22. Trajectory tracking by the end-effector.

have been proposed. The robot dynamics' model and its parameters can be found in ref. [59]. The disturbance term is also given by ref. [60]. The desired trajectory is a lemniscate of Bernoulli in task space specified by

$$\begin{cases} x_d(t) = 0.3\cos(t)/(1+\sin^2(t)) \\ y_d(t) = 0.5 + (0.3\sin(t)\cos(t)/(1+\sin^2(t))) \\ \varphi_d(t) = 0 \end{cases} \tag{61}$$

The control parameters are selected as $K = [1\ 1]$, $L = [0.015\ 50] \times 10^3$, $\mu_0 = [7900\ 160]$, and $p = 1$. Under these settings, Figs. 22 and 23 show the reference and actual trajectory and deviation of $\varphi(t)$ from its zero desired value. As it is observed, tracking performance is very suitable and the position errors in x and y directions are small. Norms of estimated velocity errors are also plotted in Fig. 24. According to this figure, the steady state value of velocity estimation error is in the range of 0.05 (m) which implies good tuning of the observer parameters. Furthermore, as it is shown in Fig. 25, all tensions in the cables are positive and bounded.

6.2. Experimental setup

To validate the proposed controller performance on a real robot manipulator, experimental results are presented in this section. A photograph of the practical control system is depicted in Fig. 26. Three

https://doi.org/10.1017/S0263574718001303 Published online by Cambridge University Press

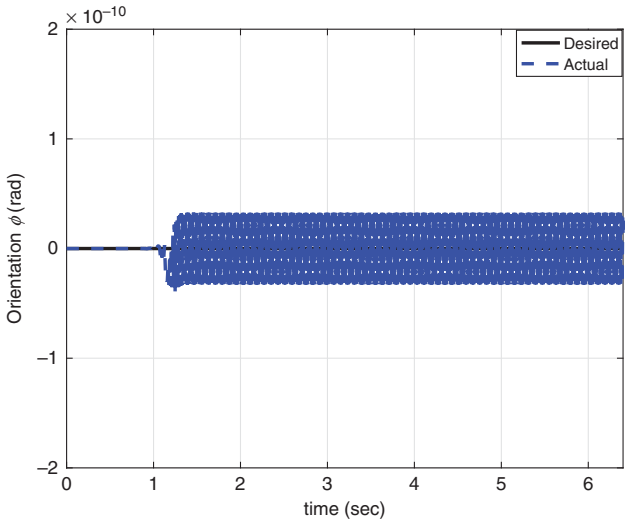


Fig. 23. Simulation results showing the rotation ϕ(t) during trajectory tracking.

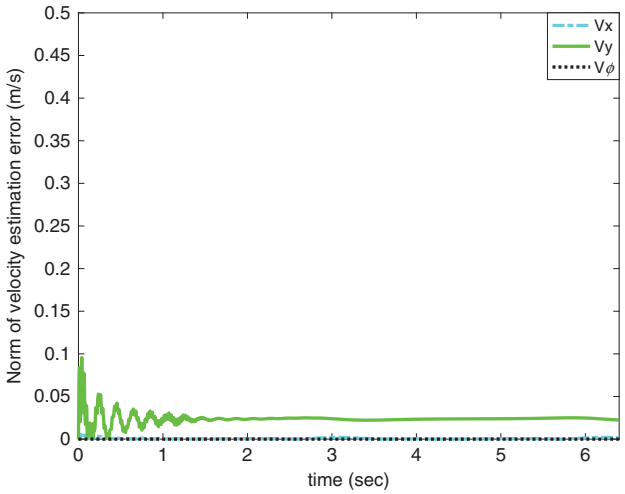


Fig. 24. Norm of velocity estimation errors.

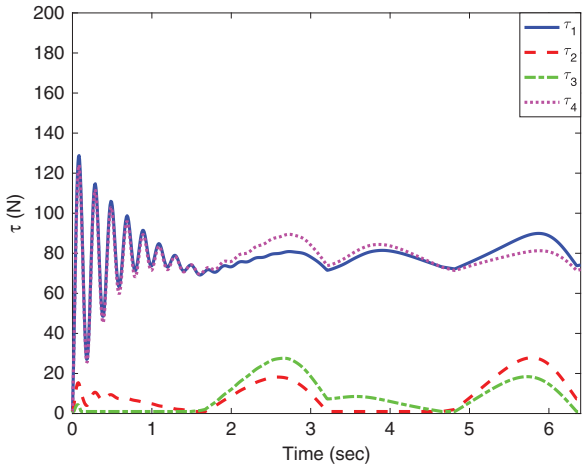


Fig. 25. Cable forces.

https://doi.org/10.1017/S0263574718001303 Published online by Cambridge University Press

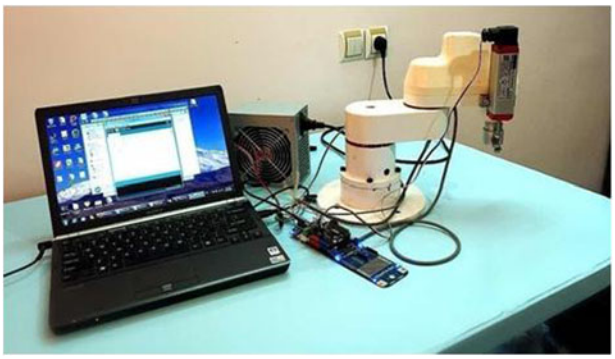


Fig. 26. The experimental setup.

geared DC servomotors are utilized as the actuators. The maximum permitted voltage for motors in the revolute joints is 5 V. The DF robot shop Rover board which is an Arduino-compatible dual motor driver based on ATmega328 has been considered for driving the motors in revolute joints. The third motor is driven using the DF robot L298p Twin V1.1 motor shield. The proposed controller has been implemented using the Arduino programming environment and inserted to the DF robot shop Rover board using the USB connection. In order to provide the controller with position feedbacks from revolute joints, the potentiometer terminals have been extracted from the motor boxes and connected to the analogue inputs of the board. In addition, as shown in Fig. 26, a linear potentiometer measures the linear displacement of the prismatic joint. About the resolution, it should be noted that the maximum angular displacement is about 200° which is converted to 1024 bits. Another important issue is that the velocity signals are not measured. In fact, for evaluation of the observer performance, they have been approximated by $\dot{q}_k \cong \frac{q_k - q_{k-1}}{t_k - t_{k-1}}$ and compared with the observer results. In the Arduino code, we can define various signals such as the voltage calculated by the controller, the desired positions, the actual positions measured by potentiometers, and observer states as outputs. A timer can also be used to measure "time". Finally, when the robot is stopped, we can load the output data into an M-file in MATLAB to plot the figures versus time. An important issue in practical implementations is time delays. Some solutions for this problem are discussed in ref. [61].

6.3. Experimental results

In this test, the set-point problem is studied. The following values for the state feedback vector k, the order of disturbance model p, and the observer parameter L were found to give good results:

$$k = \begin{bmatrix} 400 & 40 \end{bmatrix}, \quad p = 1, \quad b_1 = 0, \quad L = \begin{bmatrix} 650 \\ 6700 \end{bmatrix} \tag{62}$$

These parameters have been applied to all controllers. The data rate in bits per second (baud) for serial data transmission has been set to 38,400. The values of the dynamics parameters in $M(q)$, $N(q, \dot{q})$, $H(q, \dot{q})$, and the actuator dynamic coefficient matrices are assumed to be unknown. By this assumption, Fig. 27 shows the angular position of the first joint in the presence of external disturbances. As shown in this figure, the designed observer can estimate the angular position with ignorable error and also the controller can compensate for various uncertainties and reduce the tracking error to very small values. The observer performance in estimating the first joint angular velocity is also illustrated in Fig. 28. As shown in this figure, the estimated velocity is smooth and more suitable than the actual velocity to be used in the control law. Motor voltage for the first joint is plotted in Fig. 29. According to this figure, the control law is smooth and does not contain very high frequencies. The observer performance in estimating the second joint angular velocity is illustrated in Fig. 30. As shown in this figure, the estimated velocity is smooth and more suitable than the actual velocity to be used in the control law. Figure 31 shows the angular position of the second joint in the presence of external disturbances. As shown in this figure, the error between estimated position and its actual value converges to zero very fast. Also, the controller can compensate for various uncertainties and reduce the tracking error to very small values. Motor voltage for the second joint is plotted in Fig. 32. According to this figure, the control law is smooth and does not contain very high frequencies.

https://doi.org/10.1017/S0263574718001303 Published online by Cambridge University Press

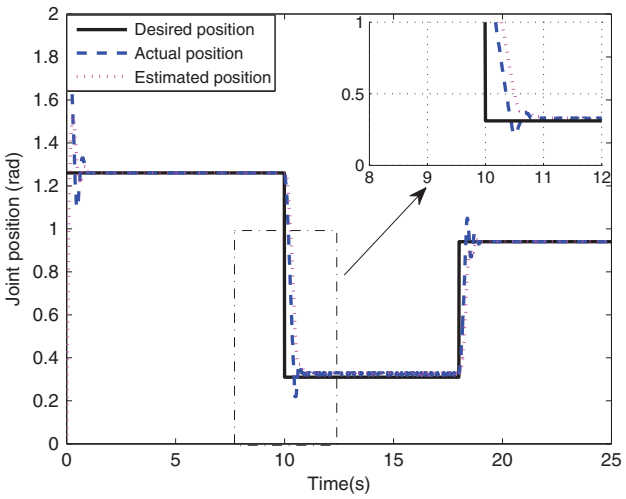


Fig. 27. Comparison of angular positions for the first joint.

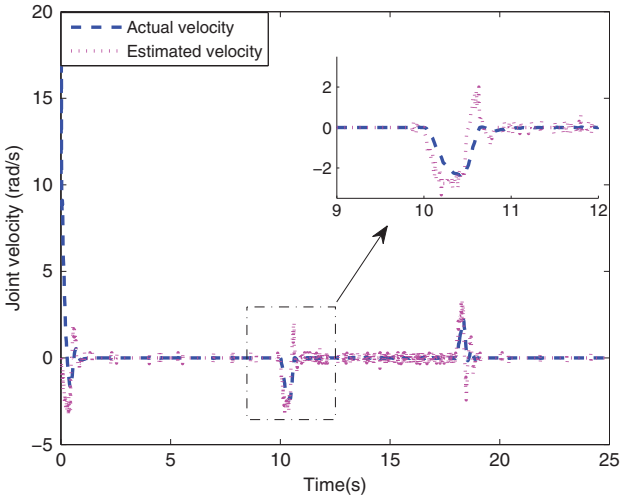


Fig. 28. Comparison of velocities for the first joint.

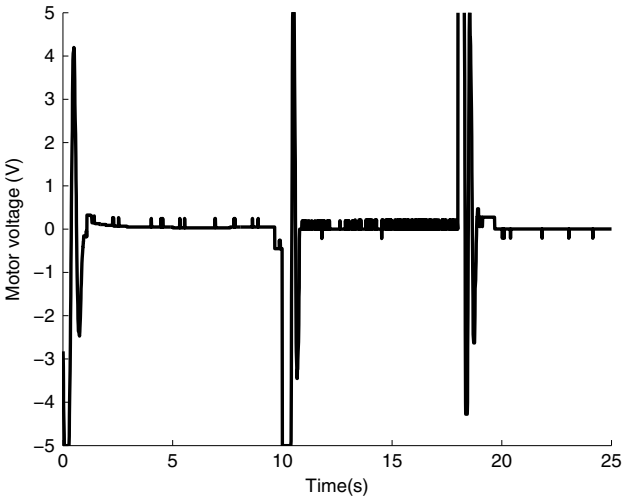


Fig. 29. Motor voltage for the first joint.

https://doi.org/10.1017/S0263574718001303 Published online by Cambridge University Press

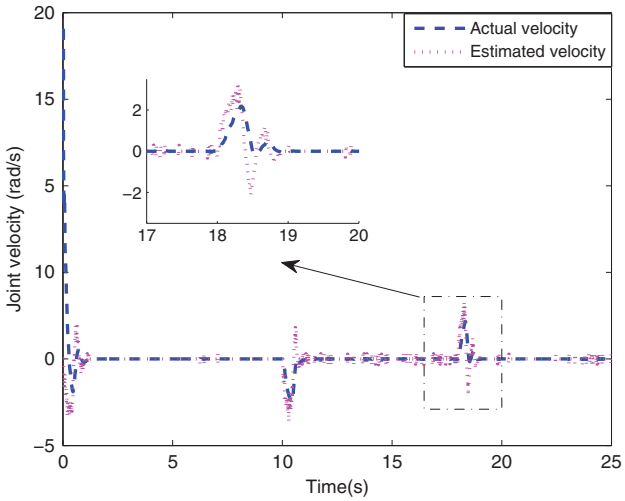


Fig. 30. Comparison of velocities for the second joint.

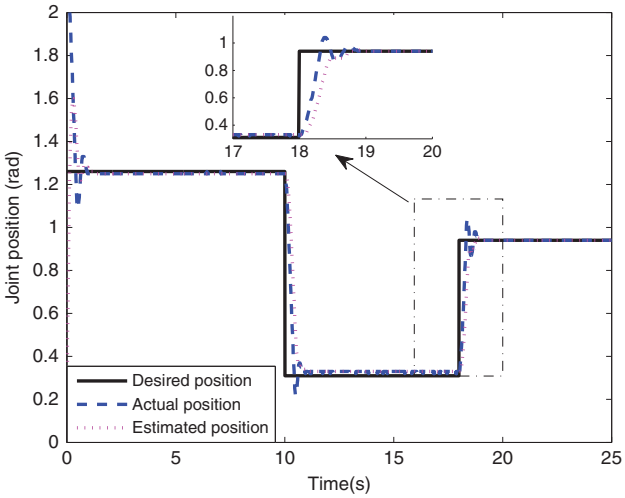


Fig. 31. Comparison of angular positions for the second joint.

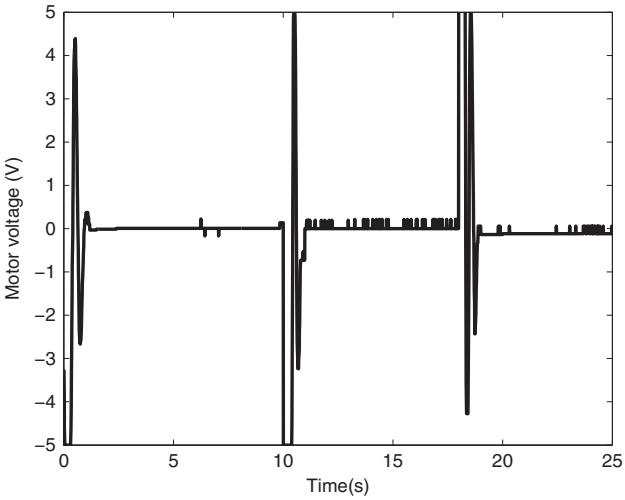


Fig. 32. Motor voltage for the second joint.

https://doi.org/10.1017/S0263574718001303 Published online by Cambridge University Press

7. Conclusion

A robust observer-based control strategy has been proposed for electrically driven robots. The control laws have been derived under the assumption that no information of the system dynamics model is in hand. Uncertainties have been estimated and compensated based on the function approximation technique using differential equations. It has been shown that the responses of linear differential equations satisfy the conditions of Stone–Weierstrass theorem. Thus, they can play the role of a universal approximator in control systems to estimate uncertainties. It is assumed that the joint velocity signals are not available and they are approximated using a linear observer. The observer converges to the system states in spite of the existence of bounded disturbances or parameter uncertainties affecting system dynamics. The system stability has been verified by the Lyapunov method. Both simulation and experimental results have demonstrated the effectiveness of the proposed controller. Future works in this field can deal with more complex forms of differential equations, since just linear ordinary differential equations have been considered in this paper.

References

1. J. Van Den Berg, P. Abbeel and K. Goldberg, "LQG-MP: Optimized path planning for robots with motion uncertainty and imperfect state information," *Int. J. Rob. Res.* **30**, 895–913 (2011).
2. D. Berenson, P. Abbeel and K. Goldberg, "A robot path planning framework that learns from experience," *Rob. Autom. IEEE Int. Conf.* 3671–3678 (2012).
3. J. Schulman, Y. Duan, J. Ho, A. Lee, I. Awwal, H. Bradlow, J. Pan, S. Patil, K. Goldberg and P. Abbeel, "Motion planning with sequential convex optimization and convex collision checking," *Int. J. Rob. Res.* **33**, 1251–1270 (2014).
4. S. Srivastava, E. Fang, L. Riano, R. Chitnis, S. Russell and P. Abbeel, "Combined task and motion planning through an extensible planner-independent interface layer," *IEEE Int. Conf. Rob. Autom.* 639–646 (2014).
5. P. Abbeel, D. Dolgov, A. Y. Ng and S. Thrun, "Apprenticeship learning for motion planning with application to parking lot navigation," *IEEE/RSJ Int. Conf. Intell. Rob. Syst.* 1083–1090 (2008).
6. Y. Shang, "Resilient multiscale coordination control against adversarial nodes," *Energies* **11**, 1–17 (2018). doi:10.3390/en11071844
7. J. Schulman, J. Ho, A. X. Lee, I. Awwal, H. Bradlow and P. Abbeel, "Finding locally optimal, collision-free trajectories with sequential convex optimization," *Rob. Sci. Syst.* **9**, 1–10 (2013).
8. W. C. Ham, "Adaptive control based on explicit model of robot manipulators," *IEEE Trans. Autom. Control* **38**, 654–658 (1993).
9. A. Izadbakhsh and P. Kheirkhahan, "On the voltage-based control of robot manipulators revisited," *Int. J. Control Autom. Syst.* **16**(4), 1887–1894 (2018).
10. L. X. Wang, *A Course in Fuzzy Systems and Control, Englewood Cliffs* (Prentice-Hall, NJ, 1996).
11. S. P. Guzmán, J. M. Valenzuela and V. Santibanez, "Adaptive neural network motion control of manipulators with experimental evaluations," *Sci. World J.* 1–13 (2014). doi:10.1155/2014/694706
12. L. Wang, T. Chai and L. Zhai, "Neural network based terminal sliding-mode control of robotic manipulators including actuator dynamics," *IEEE Trans. Ind. Electron.* **56**, 3296–3304 (2009).
13. J. Peng, J. Wang and Y. Wang, "Neural network based robust hybrid control for robotic system: An H_8 approach," *Nonlinear Dyn.* **65**, 421–431 (2011).
14. H. Fallah Ghavidel and A. Akbarzadeh Kalat, "Observer-based hybrid adaptive fuzzy control for affine and nonaffine uncertain nonlinear systems," *Neural Comput. Appl.* **30**, 1187–1202 (2018).
15. R. J. Wai and R. Muthusamy, "Fuzzy-neural-network inherited sliding-mode control for robot manipulator including actuator dynamics," *IEEE Trans. Neural Networks Learn. Syst.* **24**, 274–287 (2013).
16. A. Izadbakhsh and P. Kheirkhahan, "An alternative stability proof for "Adaptive Type-2 fuzzy estimation of uncertainties in the control of electrically flexible-joint robots"," *J. Vibr. Control*. doi:10.1177/1077546318802694
17. Y. Pan and M. J. Er, "Enhanced adaptive fuzzy control with optimal approximation error convergence," *IEEE Trans. Fuzzy Syst.* **21**, 1123–1132 (2013).
18. S. Khorashadizadeh and M. M. Fateh, "Robust task-space control of robot manipulators using Legendre polynomials for uncertainty estimation," *Nonlinear Dyn.* **79**, 1151–1161 (2015).
19. A. Izadbakhsh, "Robust control design for rigid-link flexible-joint electrically driven robot subjected to constraint: theory and experimental verification," *Nonlinear Dyn.* **85**, 751–765 (2016).
20. A. Izadbakhsh, "FAT-based robust adaptive control of electrically driven robots without velocity measurements," *Nonlinear Dyn.* **89**, 289–304 (2017).
21. N. Gole, A. Gole, K. Barra and T. Bouktir, "Observer-based adaptive control of robot manipulators: Fuzzy systems approach," *Appl. Soft Comput.* **8**, 778–787 (2008).
22. S. E. Talole, J. P. Kolhe and S. B. Phadke, "Extended-state-observer-based control of flexible-joint system with experimental validation," *IEEE Trans. Ind. Electron.* **57**, 1411–1419 (2010).
23. X. Wei and L. Guo, "Composite disturbance-observer-based control and terminal sliding mode control for non-linear systems with disturbances," *Int. J. Control* **82**, 1082–1098 (2009).
24. J. Alvarez, D. Rosas and J. Pena, "Analog implementation of a robust control strategy for mechanical systems," *IEEE Trans. Ind. Electron.* **56**, 3377–3385 (2009).

https://doi.org/10.1017/S0263574718001303 Published online by Cambridge University Press

25. H. Sira-Ramírez, C. López-Uribe and M. Velasco-Villa, "Linear observer-based active disturbance rejection control of the omnidirectional mobile robot," *Asian J. Control* **15**, 51–63 (2013).
26. R. Cui, L. Chen, C. Yang and M. Chen, "Extended state observer-based integral sliding mode control for an underwater robot with unknown disturbances and uncertain nonlinearities," *IEEE Trans. Ind. Electron.* **64**(8), 6785–6795 (2017).
27. Z. Peng and J. Wang, "Output-feedback path-following control of autonomous underwater vehicles based on an extended state observer and projection neural networks," *IEEE Trans. Syst. Man Cybern. Syst.* **48**(4), 535–544 (2018).
28. L. Zhao, Q. Li, B. Liu and H. Cheng, "Trajectory tracking control of a one degree of freedom manipulator based on a switched sliding mode controller with a novel extended state observer framework," *IEEE Trans. Syst. Man Cybern. Syst.* 1–9 (2018).
29. H. Wang, S. Li, Q. Lan, Z. Zhao and X. Zhou, "Continuous terminal sliding mode control with extended state observer for PMSM speed regulation system," *Trans. Inst. Meas. Control* **39**(8), 1195–1204 (2017).
30. S. Wang, X. Ren, J. Na and T. Zeng, "Extended-state-observer-based funnel control for nonlinear servomechanisms with prescribed tracking performance," *IEEE Trans. Autom. Sci. Eng.* **14**(1), 98–108 (2017).
31. Z. Peng and J. Wang, "Output-feedback path-following control of autonomous underwater vehicles based on an extended state observer and projection neural networks," *IEEE Trans. Syst. Man Cybern. Syst.* **48**(4), 535–544 (2018).
32. P. C. Muller and J. Ackermann, "Nichtlineare regelung von elastischen robotern," **In**: *VDI-Berichte 598, Steuerung und Regelung von Roboter* (Springer-Verlag, Berlin, Germany, 1986) pp. 321–333.
33. M. Nakao, K. Ohnishi and K. Miyachi, "Robust decentralized joint control based on interference estimation," *IEEE Int. Conf. Rob. Autom.* **4**, 326–333 (1987).
34. C. J. Kemf and S. Kobayashi, "Disturbance observer and feedforward design for a high-speed direct-drive positioning table," *IEEE Trans. Contr. Syst. Technol.* **7**, 513–526 (1999).
35. Y. H. Huang and W. Messner, "A novel disturbance observer design for magnetic hard drive servo system with rotary actuator," *IEEE Trans. Magn.* **4**, 1892–1894 (1998).
36. J. Ishikawa and M. Tomizuka, "Pivot friction compensation using an accelerometer and a disturbance observer for hard disk," *IEEE/ASME Trans. Mechatron.* **3**, 194–201 (1998).
37. A. Mohammadi, M. Tavakoli, H. J. Marquez and F. Hashemzadeh, "Nonlinear disturbance observer design for robotic manipulators," *Control Eng. Pract.* **21**, 253–267 (2013).
38. Z. Li, C. Y. Su, L. Wang, Z. Chen and T. Chai, "Nonlinear disturbance observer-based control design for a robotic exoskeleton incorporating fuzzy approximation," *IEEE Trans. Ind. Electron.* **62**(9), 5763–5775 (2015).
39. Z. Chu, J. Cui and F. Sun, "Fuzzy adaptive disturbance-observer-based robust tracking control of electrically driven free-floating space manipulator," *IEEE Syst. J.* **8**, 343–352 (2014).
40. W. H. Chen, J. Yang, L. Guo and S. Li, "Disturbance-observer-based control and related methods—An overview," *IEEE Trans. Ind. Electron.* **63**, 1083–1095 (2016).
41. S. Tong and Y. Li, "Observer-based fuzzy adaptive control for strict-feedback nonlinear systems," *Fuzzy Sets Syst.* **160**, 1749–1764 (2009).
42. C. C. Cheah, C. Liu and J. J. E. Slotine, "Adaptive Jacobian tracking control of robots with uncertainties in kinematic, dynamic and actuator models," *IEEE Trans. Autom. Control* **51**, 1024–1029 (2006).
43. A. Izadbakhsh, "A note on the "nonlinear control of electrical flexible-joint robots"," *Nonlinear Dyn.* **89**, 2753–2767 (2017).
44. A. Izadbakhsh, "Robust adaptive control of voltage saturated flexible joint robots with experimental evaluations," *AUT J. Model. Simul.* **50**(1), 31–38 (2018).
45. W. H. Chen, "Disturbance observer based control for nonlinear systems," *IEEE/ASME Trans. Mechatron.* **9**, 706–710 (2004).
46. M. W. Spong, S. Hutchinson and M. Vidyasagar, *Robot Modelling and Control* (Wiley, Hoboken, 2006).
47. A. Izadbakhsh and M. M. Fateh, "Real-time robust adaptive control of robots subjected to actuator voltage constraint," *Nonlinear Dyn.* **78**, 1999–2014 (2014).
48. A. Izadbakhsh and S. Khorashadizadeh, "Robust task-space control of robot manipulators using differential equations for uncertainty estimation," *Robotica* **35**(9), 1923–1938 (2017).
49. Z. Qu and D. M. Dawson, *Robust Tracking Control of Robot Manipulators* (IEEE Press, Inc., New York, 1996).
50. A. Izadbakhsh and S. M. R. Rafiei, "Endpoint perfect tracking control of robots - A robust non inversion-based approach," *Int. J. Control Autom. Syst.* **7**, 888–898 (2009).
51. A. Izadbakhsh, A. Akbarzadeh Kalat, M. M. Fateh and S. M. R. Rafiei, "A robust anti-windup control design for electrically driven robots–Theory and experiment," *Int. J. Control Autom. Syst.* **9**, 1005–1012 (2011).
52. A. Izadbakhsh and S. Khorashadizadeh, "Robust impedance control of robot manipulators using differential equations as universal approximator," *Int. J. Control*. **91**(10), 2170–2186 (2018).
53. A. Izadbakhsh, "Closed-Form Dynamic Model of PUMA560 Robot Arm," *Proceedings of the 4th International Conference on Autonomous Robots and Agents* (2009) pp. 675–680.
54. P. Corke, "The unimation puma servo system," *CSIRO Div. Manuf. Technol.* MTM-226, 1–54 (1994).
55. P. Corke and B. Armstrong-Helouvry, "A search for consensus among model parameters reported for the PUMA560 robot," *IEEE Int. Conf. Rob. Autom.* **2**, 1608–1613 (1994).

https://doi.org/10.1017/S0263574718001303 Published online by Cambridge University Press

56. B. Pratap and S. Purwar, "Real-time implementation of neuro adaptive observer-based robust backstepping controller for twin rotor control system," *J. Control Autom. Electr. Syst.* **25**, 137–150 (2014).
57. S. Purwar, I. N. Kar and A. N. Jha, "Adaptive output feedback tracking control of robot manipulators using position measurements only," *Expert Syst. Appl.* **34**, 2789–2798 (2008).
58. J. C. Patra and A. C. Kot, "Nonlinear dynamic system identification using Chebyshev functional link artificial neural networks," *IEEE Trans. Syst. Man Cybern. Part B* **32**, 505–511 (2002).
59. H. Jabbari Asl and F. Janabi-sharifi, "Adaptive neural network control of cable-driven parallel robots with input saturation," *Eng. Appl. Artif. Intell.* **65**, 252–260 (2017).
60. H. Jabbari Asl and J. Yoon, "Robust trajectory tracking control of cable-driven parallel robots," *Nonlinear Dyn.* **89**(4), 2769–2784 (2017).
61. Y. Shang, "On the delayed scaled consensus problems," *Appl. Sci.* **7**(7), 713 (2017).

Appendix A

In order to show that (11) is the response of (10), consider the roots of the characteristic equation of (10). These roots have the general form of $r_{i,i+1} = \lambda_i \pm j\omega_i$. As a result, the response can be represented as $\delta(t) = \sum_{i=1}^{p}[a_i e^{\lambda_i t}\cos(\omega_i t) + b_i e^{\lambda_i t}\sin(\omega_i t)]$ in which a_i and b_i depend on the initial conditions of $\delta(t)$. In other words, we have $\delta(t) = \sum_{i=1}^{p} c_i e^{\lambda_i t}\sin(\omega_i t + \theta_i')$ in which $c_i = \sqrt{a_i^2 + b_i^2}$ and $\theta_i' = \tan^{-1}\left({}^{a_i}/_{b_i}\right)$.

Since $\sin(\omega_i t + \theta_i') = \cos\left(\omega_i t + \theta_i' - \frac{\pi}{2}\right) = \cos(\omega_i t + \theta_i)$, we can write $\delta(t) = \sum_{i=1}^{p} c_i e^{\lambda_i t}\cos(\omega_i t + \theta_i)$.

Appendix B

Proof of Proposition 1. Let o be a set of continuous function on T and T is convex set in the form of (11). Now, suppose $\delta_1(t)$ and $\delta_2(t)$ are given as

$$\begin{aligned}\delta_1(t) &= \sum_{i=1}^{p} c_i e^{\lambda_i t}\cos(\omega_i t + \theta_i)\\ \delta_2(t) &= \sum_{j=1}^{p} \bar{c}_j e^{\bar{\lambda}_j t}\cos(\bar{\omega}_j t + \bar{\theta}_j)\end{aligned} \tag{A1}$$

we have

$$\begin{aligned}\delta_1(t) + \delta_2(t) &= \sum_{i=1}^{p} c_i e^{\lambda_i t}\cos(\omega_i t + \theta_i) + \sum_{j=1}^{p} \bar{c}_j e^{\bar{\lambda}_j t}\cos(\bar{\omega}_j t + \bar{\theta}_j)\\ \delta_1(t).\delta_2(t) &= \frac{1}{2}\sum_{i=1}^{p}\sum_{j=1}^{p} c_i \bar{c}_j e^{(\lambda_i + \bar{\lambda}_j)t}\cos((\omega_i + \bar{\omega}_j)t + (\theta_i + \bar{\theta}_j))\\ &\quad + \frac{1}{2}\sum_{i=1}^{p}\sum_{j=1}^{p} c_i \bar{c}_j e^{(\lambda_i + \bar{\lambda}_j)t}\cos((\omega_i - \bar{\omega}_j)t + (\theta_i - \bar{\theta}_j))\end{aligned} \tag{A2}$$

Hence, $\delta_1(t) + \delta_2(t)$ and $\delta_1(t).\delta_2(t) \in o$. Finally, for any arbitrary $\sigma \in \Re$ we can get

$$\sigma.\delta(t) = \sum_{i=1}^{p} \sigma c_i e^{\lambda_i t}\cos(\omega_i t + \theta_i) \tag{A3}$$

which is also in the form of (11). So, by considering (A2) and (A3) we can conclude that o is algebra. Therefore, the first condition of Stone–Weierstrass theorem is satisfied for o. We show that o separates points on T. We choose the parameters of the $\delta(t)$ in the form of (11) as follows:

$$c_1 = 1,\ \lambda_1 = -1,\ \omega_1 = 0,\ \theta_1 = 0 \tag{A4}$$

Since $t_1 \neq t_2$, then $e^{-t_1} \neq e^{-t_2}$ and therefore the second condition is also verified. To show that o vanishes at no point of T, we simply observe that any system in the form of (11) with $\omega_i = 0$, $\theta_i = 0$, and $c_i > 0$ has the property of

$$\forall t \in T\ ,\ \delta(t) > 0 \tag{A5}$$

https://doi.org/10.1017/S0263574718001303 Published online by Cambridge University Press

Hence, o vanishes at no point of T. Thus, the three conditions of the Stone–Weierstrass theorem are satisfied. Therefore, the result follows by Stone–Weierstrass theorem.

Remark 7. The interested reader may be considering a class of functions (15) (by plugging $\omega_i = \bar{\omega}_j$ and $\theta_i \neq \bar{\theta}_j$ in (A1)). It is easy to see that this class of functions is closed under multiplication operations. In this manner we have

$$\begin{aligned}\delta_1(t).\delta_2(t) = \frac{1}{2}\sum_{i=1}^{p}\sum_{j=1}^{p} c_i\bar{c}_j e^{(\lambda_i+\bar{\lambda}_j)t}\cos\Big(2\omega_i t + \big(\theta_i + \bar{\theta}_j\big)\Big) \\ + \frac{1}{2}\sum_{i=1}^{p}\sum_{j=1}^{p} c_i\bar{c}_j e^{(\lambda_i+\bar{\lambda}_j)t}\cos\big(\theta_i - \bar{\theta}_j\big)\end{aligned} \tag{A6}$$

As can be seen, the first term of the last equation is in the form of (11). Now, problem is with the second term. This term can be interpreted as the state that (10) has the different real eigenvalues ($\omega_i = \bar{\omega}_j = 0$), which is also in the form of (11).

https://doi.org/10.1017/S0263574718001303 Published online by Cambridge University Press