Robotica (2022), **40**, pp. 3415–3434
doi:10.1017/S0263574722000303

CAMBRIDGE UNIVERSITY PRESS

RESEARCH ARTICLE

FAT-based robust adaptive controller design for electrically direct-driven robots using Phillips q-Bernstein operators

Alireza Izadbakhsh[1,*], Ali Akbarzadeh Kalat[2] and Nazila Nikdel[3]

[1]Department of Electrical Engineering, Garmsar Branch, Islamic Azad University, Garmsar, Iran, [2]Faculty of Electrical Engineering, Shahrood University of Technology, Shahrood, Iran, and [3]Faculty of Electrical and Computer Engineering, Urmia University, Urmia, Iran
*Corresponding author. E-mail: izadbakhsh_alireza@hotmail.com

Received: 23 August 2021; **Revised:** 4 January 2022; **Accepted:** 10 February 2022; **First published online:** 15 March 2022

Keywords: function approximation technique, industrial manipulator, uncertainties, Philips q-Bernstein operator, adaptive control, Chebyshev neural network

Abstract

This article proposes a robust and adaptive controller for industrial robot arms with multiple degrees of freedom without the need for velocity measurement. Many of the controllers designed for manipulators are model-based and require detailed knowledge of the system model. In contrast to these methods, this paper proposes a model-free controller using the Philips q-Bernstein operator as universal approximator. The designed controller can approximate uncertainties including external disturbances and unmodeled dynamics based on its universal approximation capability. Besides, most of the controllers revealed for robot arms are torque-based, which is not a realistic presumption from a practical point of view. In the proposed control method, the voltage applied to the actuator is considered as the control signal. However, unlike many voltage-based methods, the need to know the exact models of the system and the actuator has been eliminated in the presented method. Also, adaptive rules are extracted during the Lyapunov analysis to ensure system stability. Finally, to analyze the performance of the presented controller, this method is simulated for an industrial robot arm, and the results are analyzed. The proposed methodology is also compared to those of a strong state-of-the-art approximator, the Chebyshev neural network.

1. Introduction

One of the factors for the expansion and growth of industries in the last decade is the use of advanced robotic systems, especially industrial manipulators. In most of these applications, robots interact with human operators, so they need to be controlled efficiently and robustly. Otherwise, the failure of these systems may cause serious damages. Therefore, in recent years, various studies have been conducted to control robotic systems. The proposed controllers can be generally classified into model-based and model-free categories. Model-based techniques are designed based on accurate knowledge of system dynamics and will fail if the system model is inaccurate or there are uncertainties [1, 2, 3].

To eliminate the drawbacks of model-based methods, model-free approaches for controlling robotic arms have been proposed. These controllers are mainly designed based on adaptive methods, fuzzy logic, or neural networks [4, 5, 6, 7, 8]. In ref. [5], a model-free adaptive controller was presented for manipulators with varying loads by augmenting the fractional-order operators into the sliding-mode scheme. The finite-time stability of the system was also confirmed using the Lyapunov lemmas. However, the presented controller is torque-based, and the motor dynamics were not considered while designing the control scheme. In ref. [6], an adaptive control method was proposed for an uncertain robot arm based on the backstepping scheme and by proposing a barrier Lyapunov function. A concurrent method was also introduced to identify the system model using the torque filtering approach. Even though the

© The Author(s), 2022. Published by Cambridge University Press.

https://doi.org/10.1017/S0263574722000303 Published online by Cambridge University Press

designed controller eliminated the need to measure acceleration, it still required velocity measurements. Reference [7] introduced an adaptive controller for flexible manipulators by augmenting fuzzy type-II logic and neural networks. The controller parameters were optimized, and the boundedness of error signals was proved. In ref. [8], an adaptive torque-based controller was developed for the Timoshenko arm subject to input dead zone. The uniformly ultimately stability of the controlled system was assured using the Lyapunov stability theorem.

The mentioned controllers were designed based on the assumption that system states were measured and available. However, measuring states such as velocities and accelerations in practical systems are not cost-effective due to the high cost of required equipment. Moreover, there may be uncertainties and errors in measured values [9]. The presence of noise in the state measurements may also severely affect the system performance and reduce its bandwidth [10]. Therefore, model-free observer-based controllers have been designed to estimate these states and control the system. In ref. [11], first, a fixed-time state observer was designed for approximating unmeasured velocities. Then, the fuzzy logic and sliding-mode control scheme were integrated to form an adaptive control method for high-order systems, including manipulators. The actuator model was not considered in the design procedure, and the joint torque was obtained as the control input. Ahmed et al. introduced an adaptive fractional-order controller for robot arms in ref. [12] based on the super-twisting sliding-mode scheme. The problem of state measurement was addressed in this study by introducing an exact differentiator, which estimated states based on the measured position. However, the controller required the acceleration in addition to the velocity approximation, which increased the computational load of the system. In ref. [13], a torque-based controller was designed for robot arms based on a high-gain observer. The observer was utilized to estimate the system speeds, and then, based on the acquired estimations, the nonsingular second-order fast sliding-mode controller was developed. Niu et al. presented a valuable observer-based controller to stabilize a general class of nonlinear systems, including manipulators in ref. [14]. The designed controller required system state values; therefore, a switched observer was first introduced to estimate all unmeasured states. The adaptive output feedback controller was then proposed based on neural networks.

Although the problem of state measurement is eliminated by designing state observers, the computational load and complexity of the system are severely increased. Uncertainties are other undesirable factors that may negatively affect the performance of industrial manipulators. Therefore, the controllers should be robustly designed to attenuate their effects. Some articles have focused on designing a controller based on uncertainty/unmodeled dynamics approximators to solve this problem [15, 16, 17, 18, 19, 20]. In ref. [15], unmodeled dynamics of a manipulator were approximated using the Takagi−Sugeno fuzzy logic, and an adaptive algorithm was developed using the small gain theorem to fulfill the trajectory tracking. According to the universal approximation feature of fuzzy logic, a satisfactory approximation of uncertainties was acquired. However, the actuator dynamics were neglected while designing the controller, and the controller was proposed using the assumption that joint velocities were available. Reference [16] presented an adaptive controller to address trajectory tracking of the robot arm, which was subject to contact with the environment. A neuro-fuzzy approach was adopted to approximate the unknown system dynamics based on the broad learning methodology. Joint velocities are assumed to be available as the inputs of the approximator. Then, a barrier Lyapunov function was utilized to design the input joint torque, which controls the system. A vast number of studies have adopted neural networks as the universal approximators to deal with the problem of uncertainty approximation [17 18, 19, 20]. In ref. [17], a radial basis function neural network was utilized to confront the model uncertainty issue of robot arms. The system was stabilized based on the backstepping scheme, and a funnel boundary was employed to limit the output overshoot. A similar neural network was used by Truong et al. [18] to compensate for the unmodeled dynamics of industrial manipulators. An adaptive torque-based controller was designed utilizing the Lyapunov lemma to guarantee the convergence of all system errors.

Fuzzy and neural approaches are commonly used as approximators due to their universal approximation property based on the Stone−Weierstrass theorem. However, these methods have a complex structure, and due to existence of numerous parameters, the controller design based on fuzzy logic or

https://doi.org/10.1017/S0263574722000303 Published online by Cambridge University Press

neural networks is not straightforward and requires sufficient design knowledge and experience. On the other hand, most of these methods usually require state feedback for uncertainty approximation or system control, which in turn increases the computational cost of the system. The need for projection algorithms to maintain the values of various parameters in certain intervals is one of the other weaknesses of these methods and is a computational load for the system [21, 22].

One of the model-free control approaches that have been considered in recent years to confront the drawbacks of fuzzy or neural network approaches is the function approximation technique (FAT). In this approach, Szász−Mirakyan operators, Fourier series, and polynomials are used for approximating functions such as uncertainty [23, 24, 25, 26]. These approaches have simpler structures in comparison with fuzzy or neural network methods, and therefore, are ideal choices for controlling industrial systems. Their computational costs are much lower than fuzzy methods and neural networks. Moreover, their design does not require much expertise and experience. Also, unlike fuzzy approaches and neural networks, the FAT-based controllers do not require multiple-state feedback to approximate uncertainties. The need for projection algorithms is also eliminated in these methods [22].

Inspired by the features of the FAT-based methods, a control structure based on the Philips q-Bernstein operators is presented for industrial manipulators in this paper. Based on the universal approximation feature, the proposed approach can efficiently approximate uncertainties and confirm the robustness of the system despite the unavailability of the exact system model and the actuator model. The proposed method also has other advantages. Many studies have used arm torque as a control input, and the design of torque-based controllers for industrial robot arms is common [27, 28, 29, 30]. However, from a realistic point of view, using torque as a control input is not realistic because it cannot be directly implemented to the robot joint motors. In the proposed controller, voltage is considered as a control input, but unlike many voltage-based controllers, the requirement to have the exact model of the actuator is also removed in this method. The dependence on measuring velocity and acceleration states, and even their estimations, which is one of the major drawbacks of fuzzy/neural network controllers, has also been eliminated in the proposed method. The presented control structure also guarantees bounded tracking error according to the Lyapunov criteria. The main contribution of the presented approach can be summarized as follows: (1) the current research is the first application of the Philips q-Bernstein operators in designing an approximator-based controller for electrically driven manipulators and (2) unlike conventional Philips q-Bernstein operator-based application, the coefficient vector of the proposed approach is tuned adaptively and is not dependent on the actual system model. Consequently, the introduced controller does not have the drawbacks of traditional approaches; (3) the proposed technique does not require velocity measurements, and position is the only measured state. The reason behind this is that adding velocity sensors like tachometers raises the hardware cost. Furthermore, one of the major issues with dexterous manipulators is the difficulty of integrating sensors into a small robot architecture, particularly as degrees of freedom (DOF) increases. Additionally, due to the existence of noise, velocity measurement reduces the system's bandwidth. Velocity has been calculated in some articles by differentiating the joint positions. This method, however, is ineffective since it amplifies the noise signal.

This article consists of six sections as follows: Philips q-Bernstein operators are described in Section 2. Section 3 is devoted to modeling the industrial robot arm with multiple DOFs. Section 4 clarifies the design procedure of the proposed robust controller. Section 5 explains the results of applying the controller to an industrial manipulator, and Section 6 gives the concluding remarks.

2. Function Approximation Using the Philips q-Bernstein Operators

Polynomials are one of the most powerful mathematical tools that can accurately approximate complex functions. The simplicity of the structure of these mathematical operators allows them to be quickly calculated by a computer. Also, by combining polynomials, spline curves can be formed to approximate any function with the desired degree of accuracy. In other words, using these operators, the approximation of each real-valued function is obtained with an acceptable desired approximation error. A new

https://doi.org/10.1017/S0263574722000303 Published online by Cambridge University Press

q-variation of the Bernstein operator called the q-Bernstein operator was proposed by Philips in 1997, as follows [31, 32]:

$$B_{N,q}(f;t)=\sum_{k=0}^{N} f\left(\frac{[k]_q}{[N]_q}\right)\begin{bmatrix}N\\k\end{bmatrix}_q t^k\prod_{s=0}^{N-k-1}(1-q^s t)\,,\ 0\le t\le 1 \tag{1}$$

where $B_{N,q}\colon C[0,1]\to C[0,1]$ is defined for any $N\in\mathbb{N}$ with $\mathbb{N}$ showing the set of natural number, and any function $f\in C[0,1]$. In Eq. (1), $\begin{bmatrix}N\\k\end{bmatrix}_q$ represents the q-binomial coefficient. In other words,

$$\begin{bmatrix}N\\k\end{bmatrix}_q=\frac{[N]_q!}{[N-k]_q![k]_q!} \tag{2}$$

where N and k are integers that satisfy $N\ge k\ge 0$. The q-factorial is represented by $[k]_q!$ and is described as follows:

$$[k]_q!=\begin{cases}[k]_q\,[k-1]_q\ldots[1]_q, & k\ge 1\\ 1, & k=0\end{cases} \tag{3}$$

where

$$[k]_q=\begin{cases}\left(1-q^k\right)/(1-q), & q\ne 1\\ k, & q=1\end{cases} \tag{4}$$

denotes the q-integer of the number $k\in\mathbb{N}\cup\{0\}$. The q-Bernstein polynomials can be used as shaping parameters to construct any free-form surface or curve. According to Theorem 2 of ref. [33], regarding any function $f\in C[0,1]$, it is confirmed that $B_{N,q}(f;t)$ converges uniformly to $f(t)$ on $[0,1]$, where q_N is a sequence that satisfies $0<q_N<1$ and $\lim_{N\to\infty}q_N=1$ [33]. It can be shown that

$$B_{N,q}(f;t)=\mathbf{W}_f^T\mathbf{Z}_f \tag{5}$$

where

$$\mathbf{W}_f=\begin{bmatrix}f(0) & f\left(\dfrac{1}{[N]_q}\right) & \ldots & f(1)\end{bmatrix}^T\in\Re^{N+1} \tag{6}$$

is a vector consisting of tuneable parameters. It is notable that, in conventional methods, which used Philips q-Bernstein operators, the coefficient vector $\mathbf{W}_f$ is formed using the known system model. However, in this paper, this vector is tuned adaptively and is not dependant on the exact system model. In Eq. (5), $\mathbf{Z}_f$ is defined as follows:

$$\mathbf{Z}_f=\begin{bmatrix}\begin{bmatrix}N\\0\end{bmatrix}_q\prod_{s=0}^{N-1}(1-q^s t) & \begin{bmatrix}N\\1\end{bmatrix}_q t\prod_{s=0}^{N-2}(1-q^s t) & \ldots & \begin{bmatrix}N\\N\end{bmatrix}_q t^N\end{bmatrix}^T\in\Re^{N+1} \tag{7}$$

is the vector of basis functions. Equation (1) can be shown in the linear form (5), which is a common form in adaptive control.

3. Robot Arm Dynamics

Equations (8) and (9) give the dynamics of an n-DOF electrically driven robot arm [34, 35]:

$$\mathbf{D}(\mathbf{q})\ddot{\mathbf{q}}+\mathbf{C}(\mathbf{q},\dot{\mathbf{q}})\dot{\mathbf{q}}+\mathbf{g}(\mathbf{q})=\mathbf{HI} \tag{8}$$

$$\mathbf{L}\dot{\mathbf{I}}+\mathbf{RI}+\mathbf{K}_b\dot{\mathbf{q}}+\boldsymbol{\eta}(\mathrm{t})=\mathbf{u}(t) \tag{9}$$

where $\mathbf{q}$, $\dot{\mathbf{q}}$, and $\ddot{\mathbf{q}}$ stand for the $n\times 1$ vectors of joint position, velocity, and acceleration, respectively. $\mathbf{D}(\mathbf{q})\in\Re^{n\times n}$ is the symmetric and positive-definite inertia matrix. $\mathbf{C}(\mathbf{q},\dot{\mathbf{q}})\in\Re^n$ and $\mathbf{g}(\mathbf{q})\in\Re^n$ are the matrices of centrifugal/Coriolis forces and the vector of gravitational forces, respectively. Since the

https://doi.org/10.1017/S0263574722000303 Published online by Cambridge University Press

dynamic model of the actuator is also considered in designing the controller, there is a need to convert mechanical parts to electrical components. Therefore, the current to torque vector conversion is performed utilizing the diagonal constant matrix $\mathbf{H} \in \Re^{n\times n}$. $\mathbf{I} \in \Re^n$ represents the currents of the actuator armature. $\mathbf{R} \in \Re^{n\times n}$ and $\mathbf{L} \in \Re^{n\times n}$ are the electrical resistance and inductance, respectively. $\mathbf{K}_b \in \Re^{n\times n}$ denotes the actuator back EMF effects. $\mathbf{u}(t) \in \Re^n$ represents the joint actuator voltage, which is considered as the control input, and finally $\boldsymbol{\eta}(t)$ represents the effect of external disturbance.

The following properties are held for the dynamics (8) and will be used for designing the controller.

Property 1. The following bounding inequality is satisfied for the positive-definite and symmetric matrix $\mathbf{D}(\mathbf{q})$ regarding all $\mathbf{q} \in \Re^n$:

$$d_m \mathbf{I}_n \leq \mathbf{D}(\mathbf{q}) \leq d_M \mathbf{I}_n$$

where $\mathbf{I}_n$ stands for the $n \times n$ identity matrix, and d_m and d_M are positive constants. It is presumed that d_m and d_M are known.

Property 2. The centripetal/Coriolis matrix $\mathbf{C}(\mathbf{q}, \dot{\mathbf{q}})$ is linear in $\dot{\mathbf{q}}$, and its dependence on $\mathbf{q}$ is similar to that of $\mathbf{D}(\mathbf{q})$. So, it follows that:

$$\|\mathbf{C}(\mathbf{q}, \dot{\mathbf{q}})\| \leq \xi_c \|\dot{\mathbf{q}}\|, \qquad \forall \mathbf{q}, \dot{\mathbf{q}} \in \Re^n$$

where ξ_c is a positive and known constant.

Property 3. Based on the Lagrangian formulation of the arm dynamics, it is derived that the matrix $\dot{\mathbf{D}}(\mathbf{q}) - 2\mathbf{C}(\mathbf{q}, \dot{\mathbf{q}})$ is skew-symmetric [36, 37, 38, 39]. That is,

$$\mathbf{y}^T \dot{\mathbf{D}}(\mathbf{q})\mathbf{y} = 2\mathbf{y}^T \mathbf{C}(\mathbf{q}, \dot{\mathbf{q}})\, \mathbf{y}, \qquad \forall \mathbf{y}, \mathbf{q}, \dot{\mathbf{q}} \in \Re^n$$

Remark 1. Property 1 is used in Appendix A. Properties 2 and 3 are utilized in proving the controlled system stability.

4. Control Design

Equation (8) is rewritten as follows by defining the tracking error of the motor current as $\mathbf{e}_I = (\mathbf{I} - \mathbf{I}_d) \in \Re^n$:

$$\mathbf{D}(\mathbf{q})\ddot{\mathbf{q}} + \mathbf{C}(\mathbf{q}, \dot{\mathbf{q}})\dot{\mathbf{q}} + \mathbf{g}(\mathbf{q}) = \mathbf{H}\mathbf{e}_I + \mathbf{H}\mathbf{I}_d \tag{10}$$

where $\mathbf{I}_d$ denotes the virtual control law. $\mathbf{e}_I$ stands for the equivalent disturbances inserted in the dynamic model. Suppose the link tracking error is defined as follows:

$$\mathbf{e}_q = \mathbf{q} - \mathbf{q}_d \in \Re^n \tag{11}$$

A possible desired current can be defined as follows:

$$\mathbf{I}_d = \mathbf{H}^{-1} \left\{ \hat{\mathbf{D}}(\mathbf{q})\ddot{\mathbf{q}}_d + \hat{\mathbf{C}}(\mathbf{q}, \dot{\mathbf{q}}_d)\dot{\mathbf{q}}_d + \hat{\mathbf{g}}(\mathbf{q}) - \gamma^2 \boldsymbol{\Gamma} \left(\boldsymbol{\psi} + \kappa \mathbf{e}_q \right) \right\} \tag{12}$$

In this equation, κ and γ are positive gains; $\boldsymbol{\Gamma} = \boldsymbol{\Gamma}^T = diag(\boldsymbol{\Gamma}_1, \boldsymbol{\Gamma}_2, \ldots, \boldsymbol{\Gamma}_n)$ is an $n \times n$ constant and positive-definite matrix, $\hat{(\cdot)}$ represents approximation of $(\bullet)$, and $\boldsymbol{\psi} \in \Re^n$ is an intermediate vector described as:

$$\dot{\boldsymbol{\psi}} \triangleq -2\gamma \boldsymbol{\psi} + \gamma^2 \dot{\mathbf{e}}_q \tag{13}$$

By replacing (12) into (10), subtracting and adding $\mathbf{C}(\mathbf{q}, \dot{\mathbf{q}}_d)\dot{\mathbf{q}}_d$, and utilizing definition (11), the following robot tracking error equation results

$$\begin{aligned} &\mathbf{D}(\mathbf{q})\ddot{\mathbf{e}}_q + \mathbf{C}(\mathbf{q}, \dot{\mathbf{q}})\dot{\mathbf{e}}_q + \mathbf{D}(\mathbf{q})\ddot{\mathbf{q}}_d + \mathbf{C}(\mathbf{q}, \dot{\mathbf{q}})\dot{\mathbf{q}}_d + \mathbf{g}(\mathbf{q}) = \mathbf{H}\mathbf{e}_I \\ &+ \hat{\mathbf{D}}(\mathbf{q})\ddot{\mathbf{q}}_d + \hat{\mathbf{C}}(\mathbf{q}, \dot{\mathbf{q}}_d)\dot{\mathbf{q}}_d + \hat{\mathbf{g}}(\mathbf{q}) - \gamma^2 \boldsymbol{\Gamma} \boldsymbol{\psi} - \kappa \gamma^2 \boldsymbol{\Gamma} \mathbf{e}_q + \mathbf{C}(\mathbf{q}, \dot{\mathbf{q}}_d)\dot{\mathbf{q}}_d - \mathbf{C}(\mathbf{q}, \dot{\mathbf{q}}_d)\dot{\mathbf{q}}_d \end{aligned} \tag{14}$$

https://doi.org/10.1017/S0263574722000303 Published online by Cambridge University Press

Some mathematical simplifications yield

$$\mathbf{D}(\mathbf{q})\ddot{\mathbf{e}}_q = -\gamma^2\boldsymbol{\Gamma}\boldsymbol{\psi} - \kappa\gamma^2\boldsymbol{\Gamma}\mathbf{e}_q + \left(\hat{\mathbf{D}}(\mathbf{q}) - \mathbf{D}(\mathbf{q})\right)\ddot{\mathbf{q}}_d + \hat{\mathbf{g}}(\mathbf{q}) - \mathbf{g}(\mathbf{q}) + \left(\hat{\mathbf{C}}(\mathbf{q}, \dot{\mathbf{q}}_d) - \mathbf{C}(\mathbf{q}, \dot{\mathbf{q}}_d)\right)\dot{\mathbf{q}}_d + \mathbf{H}\mathbf{e}_I - \mathbf{C}(\mathbf{q}, \dot{\mathbf{q}})\dot{\mathbf{e}}_q + \mathbf{C}(\mathbf{q}, \dot{\mathbf{q}}_d)\dot{\mathbf{q}}_d - \mathbf{C}(\mathbf{q}, \dot{\mathbf{q}})\dot{\mathbf{q}}_d \tag{15}$$

It is straightforward to present that $\mathbf{C}(\mathbf{q}, \dot{\mathbf{q}}_d)\dot{\mathbf{q}}_d - \mathbf{C}(\mathbf{q}, \dot{\mathbf{q}})\dot{\mathbf{q}}_d = -\mathbf{C}\left(\mathbf{q}, \dot{\mathbf{e}}_q\right)\dot{\mathbf{q}}_d$, So, Eq. (15) is simplified as (16):

$$\mathbf{D}(\mathbf{q})\ddot{\mathbf{e}}_q = -\gamma^2\boldsymbol{\Gamma}\boldsymbol{\psi} - \kappa\gamma^2\boldsymbol{\Gamma}\mathbf{e}_q + \left(\hat{\mathbf{D}}(\mathbf{q}) - \mathbf{D}(\mathbf{q})\right)\ddot{\mathbf{q}}_d + \hat{\mathbf{g}}(\mathbf{q}) - \mathbf{g}(\mathbf{q}) + \left(\hat{\mathbf{C}}(\mathbf{q}, \dot{\mathbf{q}}_d) - \mathbf{C}(\mathbf{q}, \dot{\mathbf{q}}_d)\right)\dot{\mathbf{q}}_d + \mathbf{H}\mathbf{e}_I - \mathbf{C}(\mathbf{q}, \dot{\mathbf{q}})\dot{\mathbf{e}}_q - \mathbf{C}\left(\mathbf{q}, \dot{\mathbf{e}}_q\right)\dot{\mathbf{q}}_d \tag{16}$$

Now, $\mathbf{D}(\mathbf{q})$, $\mathbf{C}(\mathbf{q}, \dot{\mathbf{q}}_d)$, and $\mathbf{g}(\mathbf{q})$ are presented as linear combinations of basis functions using the universal approximation theorem, as (17):

$$\begin{aligned} \mathbf{D}(\mathbf{q}) &= \mathbf{W}_D^T\mathbf{Z}_D + \boldsymbol{\varepsilon}_D \\ \mathbf{C}(\mathbf{q}, \dot{\mathbf{q}}_d) &= \mathbf{W}_C^T\mathbf{Z}_C + \boldsymbol{\varepsilon}_C \\ \mathbf{g}(\mathbf{q}) &= \mathbf{W}_g^T\mathbf{Z}_g + \boldsymbol{\varepsilon}_g \end{aligned} \tag{17}$$

where $\mathbf{W}_D \in \Re^{n^2(N+1)\times n}$, $\mathbf{W}_C \in \Re^{n^2(N+1)\times n}$, and $\mathbf{W}_g \in \Re^{n(N+1)\times n}$ are weighting vectors. The number of utilized basis functions is represented by N. The basis function vectors are represented by $\mathbf{Z}_D \in \Re^{n^2(N+1)}$, $\mathbf{Z}_C \in \Re^{n^2(N+1)}$, and $\mathbf{Z}_g \in \Re^{n(N+1)}$. The errors of approximation are $\boldsymbol{\varepsilon}_D, \boldsymbol{\varepsilon}_C$, and $\boldsymbol{\varepsilon}_g$. By utilizing the same set of basis functions, (18) is presented:

$$\begin{aligned} \hat{\mathbf{D}}(\mathbf{q}) &= \hat{\mathbf{W}}_D^T\mathbf{Z}_D \\ \hat{\mathbf{C}}(\mathbf{q}, \dot{\mathbf{q}}_d) &= \hat{\mathbf{W}}_C^T\mathbf{Z}_C \\ \hat{\mathbf{g}}(\mathbf{q}) &= \hat{\mathbf{W}}_g^T\mathbf{Z}_g \end{aligned} \tag{18}$$

where $\hat{\mathbf{W}}_D \in \Re^{n^2(N+1)\times n}$, $\hat{\mathbf{W}}_C \in \Re^{n^2(N+1)\times n}$, and $\hat{\mathbf{W}}_g \in \Re^{n(N+1)\times n}$ are approximation of $\mathbf{W}_D, \mathbf{W}_C$, and $\mathbf{W}_g$, respectively. By replacing (17) and (18) into (16), the output tracking loop dynamics are obtained as:

$$\mathbf{D}(\mathbf{q})\ddot{\mathbf{e}}_q = -\gamma^2\boldsymbol{\Gamma}\boldsymbol{\psi} - \kappa\gamma^2\boldsymbol{\Gamma}\mathbf{e}_q - \tilde{\mathbf{W}}_D^T\mathbf{Z}_D\ddot{\mathbf{q}}_d - \boldsymbol{\varepsilon}_D\ddot{\mathbf{q}}_d - \tilde{\mathbf{W}}_g^T\mathbf{Z}_g - \boldsymbol{\varepsilon}_g - \tilde{\mathbf{W}}_C^T\mathbf{Z}_C\dot{\mathbf{q}}_d - \boldsymbol{\varepsilon}_C\dot{\mathbf{q}}_d + \mathbf{H}\mathbf{e}_I - \mathbf{C}(\mathbf{q}, \dot{\mathbf{q}})\dot{\mathbf{e}}_q - \mathbf{C}\left(\mathbf{q}, \dot{\mathbf{e}}_q\right)\dot{\mathbf{q}}_d \tag{19}$$

In this equation, $\tilde{\mathbf{W}}_{(\bullet)} = \mathbf{W}_{(\bullet)} - \hat{\mathbf{W}}_{(\bullet)}$ denotes the error of the weight approximation. It should be noted that similar methods have been applied in the cooperative system control [40]. Equation (20) is the derived state-space model of (13) and (19):

$$\dot{\mathbf{x}} = -\mathbf{A}\mathbf{x} + \mathbf{B}\left\{\mathbf{H}\mathbf{e}_I - \mathbf{C}(\mathbf{q}, \dot{\mathbf{q}})\dot{\mathbf{e}}_q - \mathbf{C}\left(\mathbf{q}, \dot{\mathbf{e}}_q\right)\dot{\mathbf{q}}_d - \boldsymbol{\varepsilon}_1 - \tilde{\mathbf{W}}_D^T\mathbf{Z}_D\ddot{\mathbf{q}}_d \quad -\tilde{\mathbf{W}}_C^T\mathbf{Z}_C\dot{\mathbf{q}}_d - \tilde{\mathbf{W}}_g^T\mathbf{Z}_g\right\} \tag{20}$$

where

$$\mathbf{x} \triangleq \left[\dot{\mathbf{e}}_q^T \quad \boldsymbol{\psi}^T \quad \mathbf{e}_q^T\right]^T \in \Re^{3n}$$

$$\mathbf{A} = \begin{bmatrix} \mathbf{0}_n & \gamma^2\mathbf{D}^{-1}(\mathbf{q})\boldsymbol{\Gamma} & \kappa\gamma^2\mathbf{D}^{-1}(\mathbf{q})\boldsymbol{\Gamma} \\ -\gamma^2\mathbf{I}_n & 2\gamma\mathbf{I}_n & \mathbf{0}_n \\ -\mathbf{I}_n & \mathbf{0}_n & \mathbf{0}_n \end{bmatrix} \in \Re^{3n\times 3n}, \tag{21}$$

$$\mathbf{B} = \begin{bmatrix} \mathbf{D}^{-1}(\mathbf{q}) \\ \mathbf{0}_n \\ \mathbf{0}_n \end{bmatrix} \in \Re^{3n\times n}$$

https://doi.org/10.1017/S0263574722000303 Published online by Cambridge University Press

and $\boldsymbol{\varepsilon}_1 = \boldsymbol{\varepsilon}(\boldsymbol{\varepsilon}_D, \boldsymbol{\varepsilon}_g, \boldsymbol{\varepsilon}_C)$ denotes the lumped approximation error. In (21), $\mathbf{0}_n$ stands for the null matrix of dimension n.

4.1. The current tracking control loop

The object of this subsection is determining a control law $\mathbf{u}(t)$ such that the perfect actuator current in (12), which reduces the error $\mathbf{e}_I$ as much as possible, is obtained. Therefore, the following control law is presented:

$$\mathbf{u}(t) = \hat{\mathbf{f}} - \mathbf{K}_c \mathbf{e}_I \tag{22}$$

where $\hat{\mathbf{f}} \in \Re^n$ is the approximation of $\mathbf{f} = \mathbf{L}\dot{\mathbf{I}}_d + \mathbf{R}\mathbf{I} + \mathbf{K}_b\dot{\mathbf{q}} + \boldsymbol{\eta}(\mathrm{t})$; and $\mathbf{K}_c \in \Re^{n\times n}$ is a diagonal, constant, and positive matrix. Then, the dynamic model of the current tracking loop with the controller, introduced in (9), is rewritten as follows:

$$\mathbf{L}\dot{\mathbf{e}}_I + \mathbf{K}_c \mathbf{e}_I = \hat{\mathbf{f}} - \mathbf{f} \tag{23}$$

Following the procedure similar to the previous section, the function approximation representation is applied:

$$\mathbf{f} = \mathbf{W}_f^T \mathbf{Z}_f + \boldsymbol{\varepsilon}_f \tag{24}$$

In this equation, $\mathbf{W}_f \in \Re^{n(N+1)\times n}$ is the weighting matrix. The vector of basis functions is presented by $\mathbf{Z}_f \in \Re^{n(N+1)}$. The vector of approximation error is represented by $\boldsymbol{\varepsilon}_f \in \Re^n$. Using the same set of basis functions, $\mathbf{f}$ is approximated as:

$$\hat{\mathbf{f}} = \hat{\mathbf{W}}_f^T \mathbf{Z}_f \tag{25}$$

The vector of the parameter error is defined as $\tilde{\mathbf{W}}_f = \mathbf{W}_f - \hat{\mathbf{W}}_f$. So, (23)–(25) are represented as follows:

$$\mathbf{L}\dot{\mathbf{e}}_I + \mathbf{K}_c \mathbf{e}_I = -\tilde{\mathbf{W}}_f^T \mathbf{Z}_f - \boldsymbol{\varepsilon}_f \tag{26}$$

4.2. Stability analysis

The following presumption is needed to perform the stability analysis.

Assumption 1. The desired path $\mathbf{q}_d(t)$ and its derivatives $\dot{\mathbf{q}}_d(t)$ and $\ddot{\mathbf{q}}_d(t)$ are bounded by P_M, V_M, and A_M, respectively, as follows:

$$P_M = \sup_{t\geq 0} \|\mathbf{q}_d(t)\|\,, V_M = \sup_{t\geq 0} \|\dot{\mathbf{q}}_d(t)\|\,, A_M = \sup_{t\geq 0} \|\ddot{\mathbf{q}}_d(t)\| \tag{27}$$

Now, to accomplish the closed-loop system stability analysis, a Lyapunov-like candidate function is considered as follows:

$$V\left(\mathbf{x}, \mathbf{e}_I, \tilde{\mathbf{W}}_D, \tilde{\mathbf{W}}_C, \tilde{\mathbf{W}}_g, \tilde{\mathbf{W}}_f\right) = \frac{1}{2}\left[\mathbf{e}_I^T \mathbf{L}\mathbf{e}_I + \mathbf{x}^T \mathbf{P}\mathbf{x} + Tr\left(\tilde{\mathbf{W}}_D^T \mathbf{Q}_D \tilde{\mathbf{W}}_D\right) + Tr\left(\tilde{\mathbf{W}}_C^T \mathbf{Q}_C \tilde{\mathbf{W}}_C\right) + Tr\left(\tilde{\mathbf{W}}_g^T \mathbf{Q}_g \tilde{\mathbf{W}}_g\right) + Tr\left(\tilde{\mathbf{W}}_f^T \mathbf{Q}_f \tilde{\mathbf{W}}_f\right)\right] \tag{28}$$

where $\mathbf{Q}_D \in \Re^{n^2(N+1)\times n^2(N+1)}$, $\mathbf{Q}_C \in \Re^{n^2(N+1)\times n^2(N+1)}$, $\mathbf{Q}_g \in \Re^{n(N+1)\times n(N+1)}$, and $\mathbf{Q}_f \in \Re^{n(N+1)\times n(N+1)}$ are positive-definite matrices. The trace operator is denoted by $Tr(\bullet)$:

$$\mathbf{P} = \begin{bmatrix} \mathbf{D}(\mathbf{q}) & \frac{-1}{\gamma}\mathbf{D}(\mathbf{q}) & \frac{\kappa}{\gamma}\mathbf{D}(\mathbf{q}) \\ \frac{-1}{\gamma}\mathbf{D}(\mathbf{q}) & \boldsymbol{\Gamma} & \mathbf{0}_n \\ \frac{\kappa}{\gamma}\mathbf{D}(\mathbf{q}) & \mathbf{0}_n & \kappa\gamma^2\boldsymbol{\Gamma} \end{bmatrix} \in \Re^{3n\times 3n} \tag{29}$$

https://doi.org/10.1017/S0263574722000303 Published online by Cambridge University Press

is the positive-definite and symmetric matrix that holds (30):

$$\mathbf{A}^T\mathbf{P}+\mathbf{P}\mathbf{A}=2\mathbf{Q} \tag{30}$$

where

$$\mathbf{Q}=\begin{bmatrix}\left(\gamma-\dfrac{\kappa}{\gamma}\right)\mathbf{D}(\mathbf{q}) & -\mathbf{D}(\mathbf{q}) & \mathbf{0}_n\\ -\mathbf{D}(\mathbf{q}) & \gamma\boldsymbol{\Gamma} & \mathbf{0}_n\\ \mathbf{0}_n & \mathbf{0}_n & \kappa^2\gamma\boldsymbol{\Gamma}\end{bmatrix}\in\Re^{3n\times 3n} \tag{31}$$

Both $\mathbf{P}$ and $\mathbf{Q}$ matrices are positive-definite if γ is large enough. The positive-definiteness of $\mathbf{P}$ and $\mathbf{Q}$ matrices and derivation of their minimum and maximum eigenvalues are presented in lemmas 1 and 2 of Appendix A. Differentiating (28) results in

$$\begin{aligned}\dot{V}\left(\mathbf{x},\mathbf{e}_I,\tilde{\mathbf{W}}_D,\tilde{\mathbf{W}}_C,\tilde{\mathbf{W}}_g,\tilde{\mathbf{W}}_f\right)=&\frac{1}{2}\dot{\mathbf{x}}^T\mathbf{P}\mathbf{x}+\frac{1}{2}\mathbf{x}^T\dot{\mathbf{P}}\mathbf{x}+\frac{1}{2}\mathbf{x}^T\mathbf{P}\dot{\mathbf{x}}+\mathbf{e}_I^T\mathbf{L}\dot{\mathbf{e}}_I-Tr\left(\tilde{\mathbf{W}}_D^T\mathbf{Q}_D\dot{\hat{\mathbf{W}}}_D\right)-Tr\left(\tilde{\mathbf{W}}_C^T\mathbf{Q}_C\dot{\hat{\mathbf{W}}}_C\right)\\&-Tr\left(\tilde{\mathbf{W}}_g^T\mathbf{Q}_g\dot{\hat{\mathbf{W}}}_g\right)-Tr\left(\tilde{\mathbf{W}}_f^T\mathbf{Q}_f\dot{\hat{\mathbf{W}}}_f\right)\Big]\end{aligned} \tag{32}$$

Based on (20), (26), and (30), and with some simplifications, (33) is resulted:

$$\begin{aligned}\dot{V}\left(\mathbf{x},\mathbf{e}_I,\tilde{\mathbf{W}}_D,\tilde{\mathbf{W}}_C,\tilde{\mathbf{W}}_g,\tilde{\mathbf{W}}_f\right)=&-\mathbf{x}^T\mathbf{Q}\mathbf{x}+\frac{1}{2}\mathbf{x}^T\dot{\mathbf{P}}\mathbf{x}+\mathbf{x}^T\mathbf{P}\mathbf{B}\left\{\mathbf{H}\mathbf{e}_I-\mathbf{C}(\mathbf{q},\dot{\mathbf{q}})\dot{\mathbf{e}}_q\right.\\&\left.-\mathbf{C}(\mathbf{q},\dot{\mathbf{e}}_q)\dot{\mathbf{q}}_d-\boldsymbol{\varepsilon}_1\right\}-\mathbf{e}_I^T\mathbf{K}_c\mathbf{e}_I-\mathbf{e}_I^T\boldsymbol{\varepsilon}_f-Tr\left(\tilde{\mathbf{W}}_D^T\left(\mathbf{Q}_D\dot{\hat{\mathbf{W}}}_D+\mathbf{Z}_D\ddot{\mathbf{q}}_d\mathbf{x}^T\mathbf{P}\mathbf{B}\right)\right)\\&-Tr\left(\tilde{\mathbf{W}}_C^T\left(\mathbf{Q}_C\dot{\hat{\mathbf{W}}}_C+\mathbf{Z}_C\dot{\mathbf{q}}_d\mathbf{x}^T\mathbf{P}\mathbf{B}\right)\right)-Tr\left(\tilde{\mathbf{W}}_g^T\left(\mathbf{Q}_g\dot{\hat{\mathbf{W}}}_g+\mathbf{Z}_g\mathbf{x}^T\mathbf{P}\mathbf{B}\right)\right)\\&-Tr\left(\tilde{\mathbf{W}}_f^T\left(\mathbf{Q}_f\dot{\hat{\mathbf{W}}}_f+\mathbf{Z}_f\mathbf{e}_I^T\right)\right)\end{aligned} \tag{33}$$

The update rules are chosen as follows:

$$\begin{aligned}\hat{\mathbf{W}}_D=&-\mathbf{Q}_D^{-1}\mathbf{Z}_D\ddot{\mathbf{q}}_d\mathbf{e}_q^T+\mathbf{Q}_D^{-1}\int\left(\dot{\mathbf{Z}}_D\ddot{\mathbf{q}}_d+\mathbf{Z}_D\dddot{\mathbf{q}}_d\right)\mathbf{e}_q^T dt+\frac{1}{\gamma}\mathbf{Q}_D^{-1}\int\mathbf{Z}_D\ddot{\mathbf{q}}_d\boldsymbol{\psi}^T dt\\&-\frac{\kappa}{\gamma}\mathbf{Q}_D^{-1}\int\mathbf{Z}_D\ddot{\mathbf{q}}_d\mathbf{e}_q^T dt-\sigma_D\mathbf{Q}_D^{-1}\int\hat{\mathbf{W}}_D dt\end{aligned} \tag{34}$$

$$\begin{aligned}\hat{\mathbf{W}}_C=&-\mathbf{Q}_C^{-1}\mathbf{Z}_C\dot{\mathbf{q}}_d\mathbf{e}_q^T+\mathbf{Q}_C^{-1}\int\left(\dot{\mathbf{Z}}_C\dot{\mathbf{q}}_d+\mathbf{Z}_C\ddot{\mathbf{q}}_d\right)\mathbf{e}_q^T dt+\frac{1}{\gamma}\mathbf{Q}_C^{-1}\int\mathbf{Z}_C\dot{\mathbf{q}}_d\boldsymbol{\psi}^T dt\\&-\frac{\kappa}{\gamma}\mathbf{Q}_C^{-1}\int\mathbf{Z}_C\dot{\mathbf{q}}_d\mathbf{e}_q^T dt-\sigma_C\mathbf{Q}_C^{-1}\int\hat{\mathbf{W}}_C dt\end{aligned} \tag{35}$$

$$\begin{aligned}\hat{\mathbf{W}}_g=&-\mathbf{Q}_g^{-1}\mathbf{Z}_g\mathbf{e}_q^T+\mathbf{Q}_g^{-1}\int\dot{\mathbf{Z}}_g\mathbf{e}_q^T dt+\frac{1}{\gamma}\mathbf{Q}_g^{-1}\int\mathbf{Z}_g\boldsymbol{\psi}^T dt-\frac{\kappa}{\gamma}\mathbf{Q}_g^{-1}\int\mathbf{Z}_g\mathbf{e}_q^T dt\\&-\sigma_g\mathbf{Q}_g^{-1}\int\hat{\mathbf{W}}_g dt\end{aligned} \tag{36}$$

$$\dot{\hat{\mathbf{W}}}_f=-\mathbf{Q}_f^{-1}\left(\mathbf{Z}_f\mathbf{e}_I^T+\sigma_f\hat{\mathbf{W}}_f\right) \tag{37}$$

In these equations, $\sigma_{(\bullet)}$s are positive constants. Then, (33) is rewritten as:

$$\begin{aligned}\dot{V}\left(\mathbf{x},\mathbf{e}_I,\tilde{\mathbf{W}}_D,\tilde{\mathbf{W}}_C,\tilde{\mathbf{W}}_g,\tilde{\mathbf{W}}_f\right)\le&-\gamma\underline{\lambda}(\mathbf{Q})\|\mathbf{x}\|^2-\underline{\lambda}(\mathbf{K}_c)\|\mathbf{e}_I\|^2+\wp_0\overline{\lambda}^{-1}(\mathbf{H})\|\mathbf{x}\|\|\boldsymbol{\varepsilon}_1\|+\|\mathbf{e}_I\|\|\boldsymbol{\varepsilon}_f\|\\&+\sigma_D Tr\left(\tilde{\mathbf{W}}_D^T\hat{\mathbf{W}}_D\right)+\sigma_C Tr\left(\tilde{\mathbf{W}}_C^T\hat{\mathbf{W}}_C\right)+\sigma_g Tr\left(\tilde{\mathbf{W}}_g^T\hat{\mathbf{W}}_g\right)+\sigma_f Tr\left(\tilde{\mathbf{W}}_f^T\hat{\mathbf{W}}_f\right)\\&+\mathbf{x}^T\mathbf{P}\mathbf{B}\mathbf{H}\mathbf{e}_I+\frac{1}{2}\mathbf{x}^T\dot{\mathbf{P}}\mathbf{x}-\mathbf{x}^T\mathbf{P}\mathbf{B}\left\{\mathbf{C}(\mathbf{q},\dot{\mathbf{q}})\dot{\mathbf{e}}_q+\mathbf{C}(\mathbf{q},\dot{\mathbf{e}}_q)\dot{\mathbf{q}}_d\right\}\end{aligned} \tag{38}$$

https://doi.org/10.1017/S0263574722000303 Published online by Cambridge University Press

Remark 2. The update rules (34)–(37) are given in detail. It is clear that joint velocity measurements are not needed for the implementation of the controller. However, the proposed method is slightly complex.

The last line of Eq. (38) is investigated in the remaining part of this proof. It is straightforward to show that the inequalities (39)–(41) are held (see Appendix B):

$$\mathbf{x}^T\mathbf{PBHe}_I \leq \wp_0\|\mathbf{x}\|\|\mathbf{e}_I\| \tag{39}$$

$$\frac{1}{2}\mathbf{x}^T\dot{\mathbf{P}}\mathbf{x} - \mathbf{x}^T\mathbf{PBC}(\mathbf{q},\dot{\mathbf{q}})\dot{\mathbf{e}}_q \leq \frac{\wp_1}{\gamma}\|\dot{\mathbf{q}}\|\|\mathbf{x}\|^2 \tag{40}$$

and

$$-\mathbf{x}^T\mathbf{PBC}\left(\mathbf{q},\dot{\mathbf{e}}_q\right)\dot{\mathbf{q}}_d \leq \wp_2\|\mathbf{x}\|^2 \tag{41}$$

By replacing (39)–(41) into (38) and some further simplifications, (42) is resulted:

$$\begin{aligned}\dot{V}\left(\mathbf{x},\mathbf{e}_I,\tilde{\mathbf{W}}_D,\tilde{\mathbf{W}}_C,\tilde{\mathbf{W}}_g,\tilde{\mathbf{W}}_f\right) \leq &-\begin{bmatrix}\|\mathbf{x}\|\\ \|\mathbf{e}_I\|\end{bmatrix}^T\mathbf{\Lambda}\begin{bmatrix}\|\mathbf{x}\|\\ \|\mathbf{e}_I\|\end{bmatrix}+\begin{bmatrix}\|\mathbf{x}\|\\ \|\mathbf{e}_I\|\end{bmatrix}^T\mathbf{P}'\begin{bmatrix}\|\boldsymbol{\varepsilon}_1\|\\ \|\boldsymbol{\varepsilon}_f\|\end{bmatrix}\\ &+\sigma_D Tr\left(\tilde{\mathbf{W}}_D^T\hat{\mathbf{W}}_D\right)+\sigma_C Tr\left(\tilde{\mathbf{W}}_C^T\hat{\mathbf{W}}_C\right)+\sigma_g Tr\left(\tilde{\mathbf{W}}_g^T\hat{\mathbf{W}}_g\right)+\sigma_f Tr\left(\tilde{\mathbf{W}}_f^T\hat{\mathbf{W}}_f\right)\end{aligned} \tag{42}$$

where

$$\mathbf{\Lambda} = \begin{bmatrix}\left(\gamma\underline{\lambda}(\mathbf{Q}) - \wp_2 - \frac{\wp_1}{\gamma}\|\dot{\mathbf{q}}\|\right) & -0.5\wp_0\\ -0.5\wp_0 & \underline{\lambda}(\mathbf{K}_c)\end{bmatrix} \tag{43}$$

and $\mathbf{P}' = \begin{bmatrix}\wp_0\overline{\lambda}^{-1}(\mathbf{H}) & 0\\ 0 & 1\end{bmatrix} \in \Re^{2\times 2}$. To confirm the negative-definiteness of $\dot{V}\left(\mathbf{x},\mathbf{e}_I,\tilde{\mathbf{W}}_D,\tilde{\mathbf{W}}_C,\tilde{\mathbf{W}}_g,\tilde{\mathbf{W}}_f\right)$, the positive-definiteness of the matrix $\mathbf{\Lambda}$ should be guaranteed. $\mathbf{\Lambda}$ is positive-definite if (44) is satisfied:

$$\gamma\underline{\lambda}(\mathbf{Q}) > \frac{\wp_0^2}{4\underline{\lambda}(\mathbf{K}_c)} + \wp_2 + \frac{\wp_1}{\gamma}\left\{V_M + \sqrt{\frac{2V(0)}{\underline{\lambda}(\mathbf{\Phi})}}\right\} \tag{44}$$

For sufficiently large values of γ, (44) is initially satisfied, and consequently, (43) will be positive-definite. It is straightforward to show that the following inequality holds

$$\begin{aligned}&-\begin{bmatrix}\|\mathbf{x}\|\\ \|\mathbf{e}_I\|\end{bmatrix}^T\mathbf{\Lambda}\begin{bmatrix}\|\mathbf{x}\|\\ \|\mathbf{e}_I\|\end{bmatrix}+\begin{bmatrix}\|\mathbf{x}\|\\ \|\mathbf{e}_I\|\end{bmatrix}^T\mathbf{P}'\begin{bmatrix}\|\boldsymbol{\varepsilon}_1\|\\ \|\boldsymbol{\varepsilon}_f\|\end{bmatrix} \leq -\frac{1}{2}\underline{\lambda}(\mathbf{\Lambda})\left\|\begin{bmatrix}\mathbf{x}\\ \mathbf{e}_I\end{bmatrix}\right\|^2\\ &+\frac{\wp_3^2}{2\underline{\lambda}(\mathbf{\Lambda})}\left\|\begin{bmatrix}\boldsymbol{\varepsilon}_1\\ \boldsymbol{\varepsilon}_f\end{bmatrix}\right\|^2, Tr\left(\tilde{\mathbf{W}}_{(\bullet)}^T\hat{\mathbf{W}}_{(\bullet)}\right) \leq \frac{1}{2}Tr\left(\mathbf{W}_{(\bullet)}^T\mathbf{W}_{(\bullet)}\right) - \frac{1}{2}Tr\left(\tilde{\mathbf{W}}_{(\bullet)}^T\tilde{\mathbf{W}}_{(\bullet)}\right)\end{aligned} \tag{45}$$

where $\wp_3 = \max\left(1, \wp_0\overline{\lambda}^{-1}(\mathbf{H})\right)$. A similar derivation is done in ref. [41]. Together with (46)

$$\begin{aligned}V\left(\mathbf{x},\mathbf{e}_I,\tilde{\mathbf{W}}_D,\tilde{\mathbf{W}}_C,\tilde{\mathbf{W}}_g,\tilde{\mathbf{W}}_f\right) \leq &\frac{1}{2}\left[\overline{\lambda}(\mathbf{\Phi})\left\|\begin{bmatrix}\mathbf{x}\\ \mathbf{e}_I\end{bmatrix}\right\|^2+\overline{\lambda}(\mathbf{Q}_D)\,Tr\left(\tilde{\mathbf{W}}_D^T\tilde{\mathbf{W}}_D\right)\right.\\ &\left.+\overline{\lambda}(\mathbf{Q}_C)\,Tr\left(\tilde{\mathbf{W}}_C^T\tilde{\mathbf{W}}_C\right)+\overline{\lambda}(\mathbf{Q}_g)\,Tr\left(\tilde{\mathbf{W}}_g^T\tilde{\mathbf{W}}_g\right)+\overline{\lambda}(\mathbf{Q}_f)\,Tr\left(\tilde{\mathbf{W}}_f^T\tilde{\mathbf{W}}_f\right)\right]\end{aligned} \tag{46}$$

https://doi.org/10.1017/S0263574722000303 Published online by Cambridge University Press

where $\mathbf{\Phi} = \begin{bmatrix} \mathbf{P} & \mathbf{0}_n \\ \mathbf{0}_n & \mathbf{L} \end{bmatrix}$, Eq. (42) is rewritten in the form of (47):

$$\begin{aligned}\dot{V}\left(\mathbf{x}, \mathbf{e}_I, \tilde{\mathbf{W}}_D, \tilde{\mathbf{W}}_C, \tilde{\mathbf{W}}_g, \tilde{\mathbf{W}}_f\right) \leq & -\mu V + \frac{1}{2}\left(\mu\bar{\lambda}(\mathbf{\Phi}) - \underline{\lambda}(\mathbf{\Lambda})\right)\left\|\begin{bmatrix}\mathbf{x}\\ \mathbf{e}_I\end{bmatrix}\right\|^2 \\ & + \frac{1}{2}\left(\mu\bar{\lambda}(\mathbf{Q}_D) - \sigma_D\right) Tr\left(\tilde{\mathbf{W}}_D^T\tilde{\mathbf{W}}_D\right) + \frac{1}{2}\left(\mu\bar{\lambda}(\mathbf{Q}_C) - \sigma_C\right) Tr\left(\tilde{\mathbf{W}}_C^T\tilde{\mathbf{W}}_C\right) \\ & + \frac{1}{2}\left(\mu\bar{\lambda}(\mathbf{Q}_g) - \sigma_g\right) Tr\left(\tilde{\mathbf{W}}_g^T\tilde{\mathbf{W}}_g\right) + \frac{1}{2}\left(\mu\bar{\lambda}(\mathbf{Q}_f) - \sigma_f\right) Tr\left(\tilde{\mathbf{W}}_f^T\tilde{\mathbf{W}}_f\right) \\ & + \frac{1}{2}\left[\frac{\wp_3^2}{\underline{\lambda}(\mathbf{\Lambda})}\left\|\begin{bmatrix}\boldsymbol{\varepsilon}_1\\ \boldsymbol{\varepsilon}_f\end{bmatrix}\right\|^2 + \sigma_D Tr\left(\mathbf{W}_D^T\mathbf{W}_D\right) + \sigma_C Tr\left(\mathbf{W}_C^T\mathbf{W}_C\right)\right. \\ & \left. + \sigma_g Tr(\mathbf{W}_g^T\mathbf{W}_g) + \sigma_f Tr(\mathbf{W}_f^T\mathbf{W}_f)\right] \end{aligned} \tag{47}$$

By choosing $\mu \leq \min\left\{\frac{\underline{\lambda}(\mathbf{\Lambda})}{\bar{\lambda}(\mathbf{\Phi})}, \frac{\sigma_D}{\bar{\lambda}(\mathbf{Q}_D)}, \frac{\sigma_C}{\bar{\lambda}(\mathbf{Q}_C)}, \frac{\sigma_g}{\bar{\lambda}(\mathbf{Q}_g)}, \frac{\sigma_f}{\bar{\lambda}(\mathbf{Q}_f)}\right\}$, (47) is further simplified as:

$$\begin{aligned}\dot{V}\left(\mathbf{x}, \mathbf{e}_I, \tilde{\mathbf{W}}_D, \tilde{\mathbf{W}}_C, \tilde{\mathbf{W}}_g, \tilde{\mathbf{W}}_f\right) \leq & -\mu V + \frac{1}{2}\left[\frac{\wp_3^2}{\underline{\lambda}(\mathbf{\Lambda})}\left\|\begin{bmatrix}\boldsymbol{\varepsilon}_1\\ \boldsymbol{\varepsilon}_f\end{bmatrix}\right\|^2 + \sigma_D Tr\left(\mathbf{W}_D^T\mathbf{W}_D\right)\right. \\ & \left. + \sigma_C Tr\left(\mathbf{W}_C^T\mathbf{W}_C\right) + \sigma_g Tr\left(\mathbf{W}_g^T\mathbf{W}_g\right) + \sigma_f Tr\left(\mathbf{W}_f^T\mathbf{W}_f\right)\right] \end{aligned} \tag{48}$$

Therefore, $\dot{V} < 0$, whenever

$$V > \frac{1}{2\mu}\left[\frac{\wp_3^2}{\underline{\lambda}(\mathbf{\Lambda})}\sup_{t_0<\tau}\left\|\begin{bmatrix}\boldsymbol{\varepsilon}_1(\tau)\\ \boldsymbol{\varepsilon}_f(\tau)\end{bmatrix}\right\|^2 + \sigma_D Tr\left(\mathbf{W}_D^T\mathbf{W}_D\right) + \sigma_C Tr\left(\mathbf{W}_C^T\mathbf{W}_C\right) + \sigma_g Tr\left(\mathbf{W}_g^T\mathbf{W}_g\right) + \sigma_f Tr\left(\mathbf{W}_f^T\mathbf{W}_f\right)\right] \tag{49}$$

So, it is proven that $\left(\mathbf{x}, \mathbf{e}_I, \tilde{\mathbf{W}}_D, \tilde{\mathbf{W}}_C, \tilde{\mathbf{W}}_g, \tilde{\mathbf{W}}_f\right)$ are uniformly ultimately bounded.

4.3. Performance assessment

In previous subsections, the boundedness of the closed-loop system errors was demonstrated. From the practical point of view, transient performance is also important. Solving (48) leads to

$$\begin{aligned} V(t) \leq & V(t_0)\, e^{-\mu(t-t_0)} + \frac{1}{2\mu}\left[\frac{\wp_3^2}{\underline{\lambda}(\mathbf{\Lambda})}\sup_{t_0<\tau}\left\|\begin{bmatrix}\boldsymbol{\varepsilon}_1(\tau)\\ \boldsymbol{\varepsilon}_f(\tau)\end{bmatrix}\right\|^2 + \sigma_D Tr\left(\mathbf{W}_D^T\mathbf{W}_D\right)\right. \\ & \left. + \sigma_C Tr\left(\mathbf{W}_C^T\mathbf{W}_C\right) + \sigma_g Tr\left(\mathbf{W}_g^T\mathbf{W}_g\right) + \sigma_f Tr\left(\mathbf{W}_f^T\mathbf{W}_f\right)\right] \end{aligned} \tag{50}$$

Based on the lower bound of the Lyapunov function (28), it is obvious that

$$\left\|\begin{bmatrix}\mathbf{x}\\ \mathbf{e}_I\end{bmatrix}\right\| \leq \sqrt{\frac{2V}{\underline{\lambda}(\mathbf{\Phi})}} \tag{51}$$

https://doi.org/10.1017/S0263574722000303 Published online by Cambridge University Press

which enables us to define

$$\left\|\begin{bmatrix}\mathbf{x}\\ \mathbf{e}_I\end{bmatrix}\right\| \leq \sqrt{\frac{2V(t_0)}{\underline{\lambda}(\mathbf{\Phi})}}e^{-0.5\mu(t-t_0)} + \frac{1}{\sqrt{\mu\underline{\lambda}(\mathbf{\Phi})}}\left[\frac{\wp_3^2}{\underline{\lambda}(\mathbf{\Lambda})}\sup_{t_0<\tau}\left\|\begin{bmatrix}\boldsymbol{\varepsilon}_1(\tau)\\ \boldsymbol{\varepsilon}_f(\tau)\end{bmatrix}\right\|^2\right.$$

$$\left.+\sigma_D Tr\left(\mathbf{W}_D^T\mathbf{W}_D\right)+\sigma_C Tr\left(\mathbf{W}_C^T\mathbf{W}_C\right)+\sigma_g Tr\left(\mathbf{W}_g^T\mathbf{W}_g\right)+\sigma_f Tr\left(\mathbf{W}_f^T\mathbf{W}_f\right)\right]^{\frac{1}{2}} \tag{52}$$

In other words, the error of tracking is bounded by the summation of a constant and a weighted exponential function. Consequently, proper adjustment of the control parameters may enhance the convergence rate of the tracking error. Therefore,

$$\lim_{t\to\infty}\left\|\begin{bmatrix}\mathbf{x}\\ \mathbf{e}_I\end{bmatrix}\right\| \leq \frac{1}{\sqrt{\mu\underline{\lambda}(\mathbf{\Phi})}}\left[\frac{\wp_3^2}{\underline{\lambda}(\mathbf{\Lambda})}\sup_{t_0\leq\tau\leq t}\left\|\begin{bmatrix}\boldsymbol{\varepsilon}_1(\tau)\\ \boldsymbol{\varepsilon}_f(\tau)\end{bmatrix}\right\|^2+\sigma_D Tr\left(\mathbf{W}_D^T\mathbf{W}_D\right)\right.$$

$$\left.+\sigma_C Tr\left(\mathbf{W}_C^T\mathbf{W}_C\right)+\sigma_g Tr\left(\mathbf{W}_g^T\mathbf{W}_g\right)+\sigma_f Tr\left(\mathbf{W}_f^T\mathbf{W}_f\right)\right]^{\frac{1}{2}} \tag{53}$$

5. Results and Analysis

The designed FAT-based controller is simulated for a 2-DOF industrial manipulator to evaluate its performance. The robot joints are actuated by DC motors. This system is dynamically modeled as (8), and the details are as follows [42]:

$$\mathbf{D}(\mathbf{q}) = \begin{bmatrix} d_{11} & d_{12}\\ d_{21} & d_{22}\end{bmatrix}$$

$$d_{11} = m_1 l_{c1}^2 + I_1 + m_2\left(l_1^2 + l_{c2}^2 + 2l_1 l_{c2}cos(q_2)\right) + I_2$$

$$d_{21} = d_{12} = m_2 l_{c2}^2 + m_2 l_1 l_{c2}cos(q_2) + I_2$$

$$d_{22} = m_2 l_{c2}^2 + I_2 \tag{54}$$

$$\mathbf{C}(\mathbf{q},\dot{\mathbf{q}}) = \begin{bmatrix} -m_2 l_1 l_{c2}sin(q_2)\dot{q}_2 & -m_2 l_1 l_{c2}sin(q_2)(\dot{q}_1+\dot{q}_2)\\ m_2 l_1 l_{c2}sin(q_2)\dot{q}_1 & 0\end{bmatrix}$$

$$\mathbf{g}(\mathbf{q}) = \begin{bmatrix} m_1 l_{c1} gcos(q_1) + m_2 g(l_{c2}cos(q_1+q_2) + l_1 gcos(q_1))\\ m_2 l_{c2} gcos(q_1+q_2)\end{bmatrix}$$

where q_1 and q_2 stand for the first and second joint angular positions, respectively. The centers of mass of the arm links are symbolized by $l_{c1} = l_{c2} = 0.375$(m), and their lengths are represented by $l_1 = l_2 = 0.75$(m). Link mass values and inertias are set to $m_1 = m_2 = 0.5$(kg) and $I_1 = I_2 = 0.0234(\text{kg} - \text{m}^2)$, respectively. The actuator parameters are numerically assigned as (55):

$$\mathbf{R} = \begin{bmatrix}1 & 0\\ 0 & 1\end{bmatrix}\Omega, \mathbf{L} = \begin{bmatrix}1 & 0\\ 0 & 1\end{bmatrix}\times 10^{-3}H, \mathbf{K}_b = \begin{bmatrix}1 & 0\\ 0 & 1\end{bmatrix}\ volt/rad/s,$$

$$\mathbf{H} = \begin{bmatrix}10 & 0\\ 0 & 10\end{bmatrix}\ Nm/A \tag{55}$$

Based on the stated numerical values, it is concluded that $d_M = 1.4998\ \text{kg} - \text{m}^2$ and $\xi_c = 0.5625\ \text{kg} - \text{m}^2$.$\mathbf{q}(0) = \begin{bmatrix}0.0691 & 1.4326\end{bmatrix}^T$(rad) is considered as the initial desired and actual vector of the joint positions. It is considered that a desired sinusoidal trajectory should be tracked by joints of the manipulator for 4 s. The controller gains are assigned to $\gamma = 20, \kappa = 10$, and $\mathbf{\Gamma} = diag(10^{-2}, 10^{-2})$.

https://doi.org/10.1017/S0263574722000303 Published online by Cambridge University Press

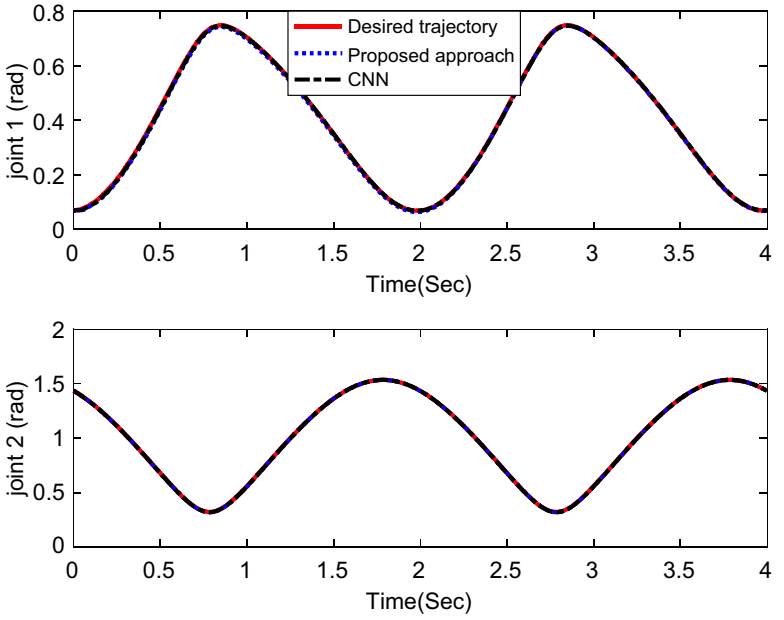


Figure 1. *The desired and actual paths.*

The matrix $\mathbf{K}_c$ appeared in (22) is set to $\mathbf{K}_c = \text{diag}(50, 50)$. It is presumed that the information of the system model (i.e., the matrices $\mathbf{D}(\mathbf{q})$, $\mathbf{C}(\mathbf{q}, \dot{\mathbf{q}})$, and $\mathbf{g}(\mathbf{q})$) is unavailable. To confront the problem of uncertainties, the first two terms of the Phillips q-Bernstein operators are elected as the regressor for approximating $\mathbf{g}(\mathbf{q})$, $\mathbf{C}(\mathbf{q}, \dot{\mathbf{q}})$, $\mathbf{D}(\mathbf{q})$, and $\mathbf{f}$. It is concluded that$\hat{\mathbf{W}}_D \in \Re^{8\times 2}$, $\hat{\mathbf{W}}_C \in \Re^{8\times 2}$, $\hat{\mathbf{W}}_g \in \Re^{4\times 2}$, and $\hat{\mathbf{W}}_f \in \Re^{4\times 2}$. The adaptive parameters are initially set to 0. The matrices $\mathbf{Q}_D, \mathbf{Q}_C, \mathbf{Q}_g$, and $\mathbf{Q}_f$, which present the convergence rates, are selected as $\mathbf{Q}_D = I_8, \mathbf{Q}_C = I_8, \mathbf{Q}_g = 10^{-3} I_4$, and $\mathbf{Q}_f = 10^{-2} I_4$, where $I_{(\bullet)}$ is a$(\bullet) \times (\bullet)$ identity matrix. The approximation error is presumed to be negligible, which means that $\sigma_D = \sigma_C = \sigma_g = \sigma_f = 0$.

The suggested Philips q-Bernstein operator-based controller performance is compared to another existing approximator to evaluate its efficiency. To this mean, the Chebyshev neural network (CNN) is integrated into the proposed control method, as the approximator, to approximate uncertainties. The detailed description of CNN and its approximation properties can be found in ref. [43]. For the proposed CNN used to approximate the nonlinear functions $\mathbf{D}(\mathbf{q})$ and $\mathbf{g}(\mathbf{q})$, the joint position $\mathbf{q}$ is given as the input. For approximation of $\mathbf{C}(\mathbf{q}, \dot{\mathbf{q}}_d)$, the joint position $\mathbf{q}$ and desired joint velocities $\dot{\mathbf{q}}_d$ are required as the input of the regressor vector. Furthermore, the actuator current $\mathbf{I}$, the time derivative of virtual control law $\mathbf{I}_d$, and the joint velocities $\dot{\mathbf{q}}$ are required signals for approximation of $\mathbf{f}$. The number of Chebyshev polynomials selected for functional expansion for each input is 1. Thus, $\mathbf{Z}_D \in \Re^{6\times 2}$, $\mathbf{Z}_C \in \Re^{10\times 2}$, $\mathbf{Z}_G \in \Re^3$, and $\mathbf{Z}_f \in \Re^7$. The initial weights of the CNN are taken to be 0. The values of the controller parameters are considered the same as previously stated values except that $\mathbf{Q}_\text{D} = 0.5 \times \text{I}_6$, $\mathbf{Q}_\text{C} = \text{I}_{10}$, $\mathbf{Q}_\text{G} = 10^{-3} \times \text{I}_3$, and $\mathbf{Q}_f = 10^{-2} \times I_7$. These values are achieved through trial and error and are not optimized. Also, we suppose that the external disturbance (56) is injected to the robotic system:

$$\boldsymbol{\eta}(\text{t}) = \begin{bmatrix} 12\sin\left(\dfrac{5\pi t}{4}\right) + 8\sin(1.8\pi t) + 12 \\ 12\sin\left(\dfrac{5\pi t}{4}\right) + 8\sin(1.8\pi t) + 12 \end{bmatrix} \tag{56}$$

Under these settings, the desired and actual paths of the controlled system are depicted in Fig. 1. It is clear that both methods can track the desired path with an acceptable tracking error. Being a single-layer

https://doi.org/10.1017/S0263574722000303 Published online by Cambridge University Press

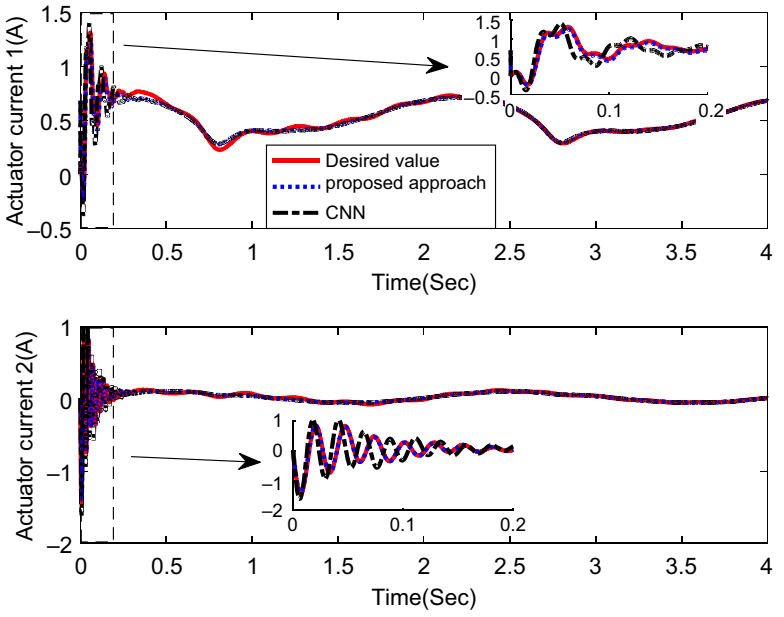


Figure 2. *Current profiles.*

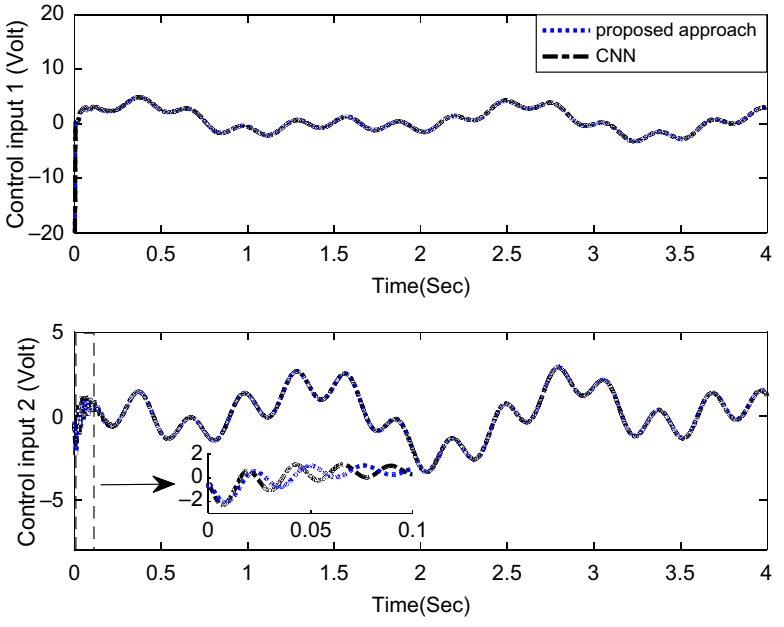


Figure 3. *Motor voltages.*

neural network, the computational complexity of CNN is less intensive as compared to multilayered perceptron and can be used for online learning [44]. However, the Phillips q-Bernstein operator-based 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 approximators consider the uncertainty as a time-varying function rather than a function of different state variables. Figure 2 presents the current control evaluation. Based on this figure, the current values are limited, and the error of current tracking negligible. The control inputs, that is, the voltages applied on the joint actuators, are illustrated in Fig. 3. According

https://doi.org/10.1017/S0263574722000303 Published online by Cambridge University Press

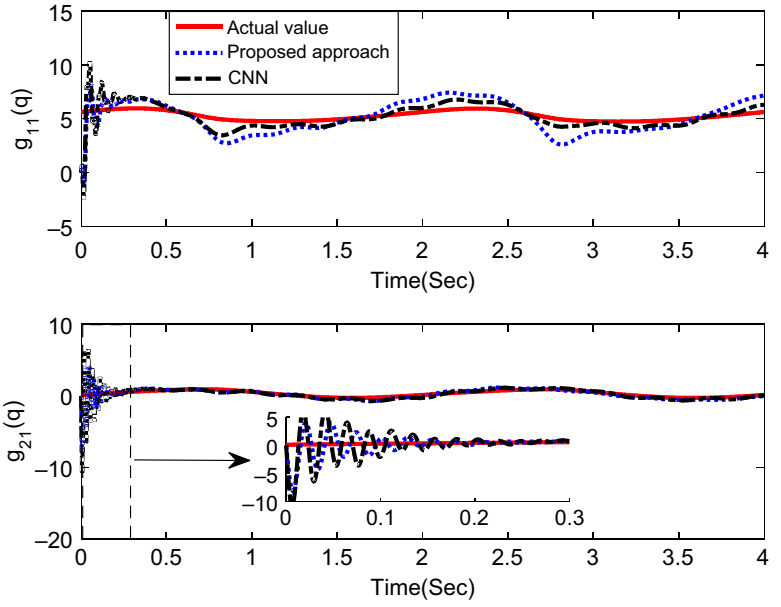


Figure 4. *Evaluation of the FAT-based method performance in* $\mathbf{g}(\mathbf{q})$ *approximation.*

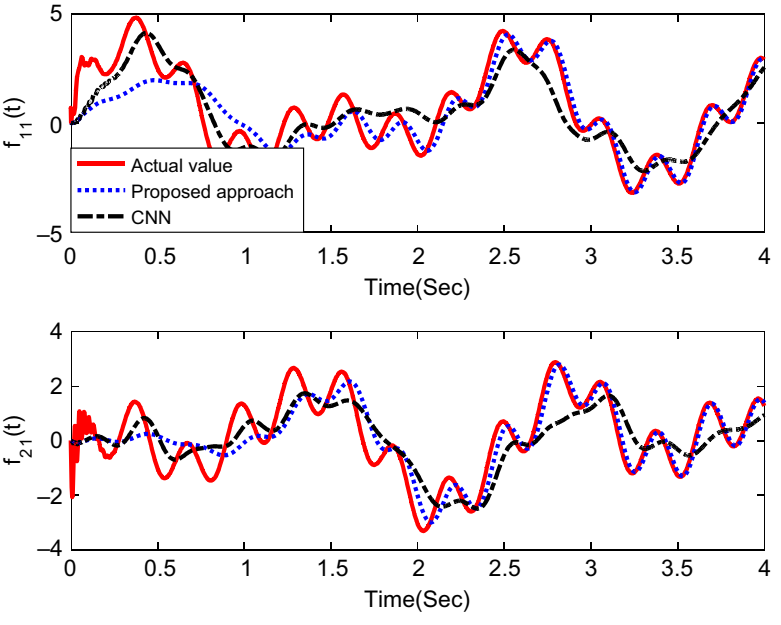


Figure 5. *Evaluation of the FAT-based method performance in* $\mathbf{f}(t)$ *approximation.*

to the figures, the oscillations of signals are quickly diminished, and the signals have no chattering. Besides, the control inputs are bounded. Finally, the approximation capability of both controllers are evaluated and presented in Figs. 4 and 5. Based on Fig. 4, the performances of the proposed method and CNN in approximating $\mathbf{g(q)}$ are almost the same. According to Fig. 5, CNN fails in approximating $\mathbf{f}(t)$ precisely. On the other hand, the proposed method performs better in approximating the function. Even though there is an error in the first 2 s of approximation, this error decreases to zero over time, and the FAT-based method can approximate the function precisely.

https://doi.org/10.1017/S0263574722000303 Published online by Cambridge University Press

6. Conclusion

An adaptive control structure for industrial robot arms based on the Philips q-Bernstein operator is presented in this paper. Unlike many common controllers, this method is designed by considering the actuator dynamics. However, the presented robust controller does not depend on the knowledge of the system or actuator model. It also eliminates the requirement for measurement or estimation of velocities, which is unavoidable in most model-free controllers. Lyapunov's approach has been utilized to prove the system stability, and comprehensive mathematical analysis has shown that all error signals are bounded. The results of simulating the adaptive controller structure for the industrial manipulator show its efficiency. Simple structure, reduction of computational load, independence from system, and actuator model, as well as velocity/acceleration measurements, and robust performance make the proposed adaptive controller a suitable option for controlling industrial robotic arms. As suggestions for future research, the proposed structure can be implemented in impedance control.

Compliance with ethical standards. The authors declare that this paper has never been submitted to other journals for simultaneous review processes, and it has not been published before (partially or completely). Moreover, this paper has not been divided into different sections in order to have more submissions. The data have not been fabricated or changed in favor of the conclusions. In addition, no theory, data, or text belonging to other authors and publications was included as if it was our own and proper acknowledgement to other works has been provided.

Funding. The authors did not receive any funding from any organization.

Conflict of interest. No conflict of interest exists among the authors.

Research involving human participants and/or animals. The authors state that this submission does not require any informed consent due to the fact that the results were acquired by computer simulation.

Data availability statements. The data that support the findings of this study are available from the corresponding author upon reasonable request.

References

[1] M. H. Korayem, S. R. Nekoo and S. Kazemi, "Finite-time feedback linearization (FTFL) controller considering optimal gains on mobile mechanical manipulators," *J. Intell. Robot. Syst.* **94**(3), 727–744 (2019).
[2] A. Dumlu, K. Erentürk, A. Kaleli and K. K. Ayten, "A comparative study of two model-based control techniques for the industrial manipulator," *Robotica* **35**(10), 2036–2055 (2017).
[3] Y. Taira, S. Sagara and M. Oya, "Model-based motion control for underwater vehicle-manipulator systems with one of the three types of servo subsystems," *Artif. Life Robot.* **25**(1), 133–148 (2020).
[4] A. Izadbakhsh, S. Khorashadizadeh and P. Kheirkhahan, "Real-time fuzzy fractional-order control of electrically driven flexible-joint robots," *AUT J. Model. Simul.* **52**(1) (2020). doi: 10.22060/MISCJ.2018.13523.5075.
[5] S. Ahmed, H. Wang and Y. Tian, "Adaptive fractional high-order terminal sliding mode control for nonlinear robotic manipulator under alternating loads," *Asian J. Control* **23**(4), 1900–1910 (2021).
[6] C. Li, F. Liu, Y. Wang and M. Buss, "Concurrent learning-based adaptive control of an uncertain robot manipulator with guaranteed safety and performance," *IEEE Trans. Syst. Man Cybern. Syst.* (2021). doi: 10.1109/TSMC.2021.3064971.
[7] M. U. Khan and T. Kara, "Adaptive control of a two-link flexible manipulator using a type-2 neural fuzzy system," *Arab. J. Sci. Eng.* **45**(3), 1949–1960 (2020).
[8] S. Chen, Z. Zhao, D. Zhu, C. Zhang and H. X. Li, "Adaptive robust control for a spatial flexible Timoshenko manipulator subject to input dead-zone," *IEEE Trans. Syst. Man Cybern. Syst.* **52**(3), 1395–1404 (2022). doi: 10.1109/TSMC.2020.3020326.
[9] Y. Shang, Leader-following consensus problems with a time-varying leader under measurement noises (2011). arXiv preprint arXiv:0909.4349.
[10] A. Izadbakhsh and N. Nikdel, "Robust adaptive controller-observer scheme for robot manipulators: A Bernstein-Stancu approach," *Robotica* **2015**, 1–17 (2021). doi: 10.1017/S0263574721001120.
[11] A. S. S. Abadi, P. A. Hosseinabadi and S. Mekhilef, "Fuzzy adaptive fixed-time sliding mode control with state observer for a class of high-order mismatched uncertain systems," *Int. J. Control, Autom. Syst.* **18**(10), 2492–2508 (2020).
[12] S. Ahmed, A. Ahmed, I. Mansoor, F. Junejo and A. Saeed, "Output feedback adaptive fractional-order super-twisting sliding mode control of robotic manipulator," *Iran. J. Sci. Technol. Trans. Electr. Eng.* **45**(1), 335–347 (2021).
[13] H. Liu, J. Sun, J. Nie and L. Zou, "Observer-based adaptive second-order non-singular fast terminal sliding mode controller for robotic manipulators," *Asian J. Control* **23**(4), 1845–1854 (2021).

https://doi.org/10.1017/S0263574722000303 Published online by Cambridge University Press

[14] B. Niu, C. K. Ahn, H. Li and M. Liu, "Adaptive control for stochastic switched non lower triangular nonlinear systems and its application to a one-link manipulator," *IEEE Trans. Syst. Man Cybern. Syst.* **48**(10), 1701–1714 (2017).
[15] Y. Fan, Y. An, W. Wang and C. Yang, "TS fuzzy adaptive control based on small gain approach for an uncertain robot manipulators," *Int. J. Fuzzy Syst.* **22**(3), 930–942 (2020).
[16] A. Izadbakhsh and S. Khorashadizadeh, "Polynomial-based robust adaptive impedance control of electrically driven robots," *Robotica* **39**, 1–21 (2020). doi: 10.1017/S0263574720001009.
[17] J. Bao, H. Wang and P. Xiaoping Liu, "Adaptive finite-time tracking control for robotic manipulators with funnel boundary," *Int. J. Adapt. Control Signal Process* **34**(5), 575–589 (2020).
[18] L. V. Truong, S. D. Huang, V. T. Yen and P. Van Cuong, "Adaptive trajectory neural network tracking control for industrial robot manipulators with deadzone robust compensator," *Int. J. Control Autom. Syst.* **18**(9), 2423–2434 (2020).
[19] 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. Vib. Control* **25**(5), 977–983 (2019).
[20] A. Izadbakhsh and S. Khorashadizadeh, "Neural control of robot manipulators considering motor voltage saturation: performance evaluation and experimental validation," *COMPEL - Int. J. Comput. Math. Electr. Electron. Eng.* **40**(1), 27–29 (2021). doi: 10.1108/COMPEL-03-2020-0127.
[21] A. Izadbakhsh and P. Kheirkhahan, "On the voltage-based control of robot manipulators revisited," *Int. J. Control Autom. Syst.* **16**(4), 1887–1894 (2018).
[22] A. Izadbakhsh, "FAT-based robust adaptive control of electrically driven robots without velocity measurements," *Nonlinear Dyn.* **89**(1), 289–304 (2017).
[23] A. Izadbakhsh, I. Zamani and S. Khorashadizadeh, "Szász-Mirakyan-based adaptive controller design for chaotic synchronization," *Int. J. Robust Nonlinear Control* **31**(5), 1689–1703 (2020).
[24] P. Kheirkhahan and A. Izadbakhsh, "Observer-based adaptive fractional-order control of flexible-joint robots using the Fourier series expansion: theory and experiment," *J. Braz. Soc. Mech. Sci. Eng.* **42**(10), 1–10 (2020).
[25] A. Deylami and A. Izadbakhsh, "FAT-based robust adaptive control of cooperative multiple manipulators without velocity measurement," *Robotica* **25**, 1–31 (2021). doi: 10.1017/S0263574721001338.
[26] 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).
[27] T. Yang, N. Sun, Y. Fang, X. Xin and H. Chen, "New adaptive control methods for n-link robot manipulators with online gravity compensation: design and experiments," *IEEE Trans. Ind. Electron.* **69**(1), 539–548 (2021).
[28] T. Beckers, J. Umlauft and S. Hirche, "Stable model-based control with Gaussian process regression for robot manipulators," *IFAC-PapersOnLine* **50**(1), 3877–3884 (2017).
[29] A. Abooee, M. MoravejKhorasani and M. Haeri, "Finite time control of robotic manipulators with position output feedback," *Int. J. Robust Nonlinear Control* **27**(16), 2982–2999 (2017).
[30] H. Zhang, H. Fang, Q. Zou and D. Zhang, "Dynamic modeling and adaptive robust synchronous control of parallel robotic manipulator for industrial application," *Complexity* **2020**(11), 1–23 (2020). doi: 10.1155/2020/5640246.
[31] G. M. Phillips, "Bernstein polynomials based on the q-integers, the heritage of P.L. Chebyshev," *Ann. Numer. Math.* **4**, 511–518 (1997).
[32] K. Khan, D. K. Lobiyal and A. Kilicman, A de Casteljau algorithm for Bernstein type polynomials based on (p,q)-integers, 2015). arXiv preprint arXiv:1507.04110.
[33] G. M. Phillips, "A generalization of the Bernstein polynomials based on the q-integers," *ANZIAM J* **42**(1), 79–86 (2000).
[34] A. Izadbakhsh, S. Khorashadizadeh and S. Ghandali, "Robust adaptive impedance control of robot manipulators using Szasz-Mirakyan operator as universal approximator," *ISA Trans.* **106**(5), 1–11 (2020).
[35] A. Izadbakhsh, P. Kheirkhahan and S. Khorashadizadeh, "FAT-based robust adaptive control of electrically driven robots in interaction with environment," *Robotica* **37**(5), 779–800 (2019).
[36] M. W. Spong, S. Hutchinson and M. Vidyasagar. *Robot Modeling and Control*, vol. **3** (Wiley, New York, 2006) pp. 75–118.
[37] A. Izadbakhsh and M. M. Fateh, "Robust Lyapunov-based control of flexible-joint robots using voltage control strategy," *Arabian J. Sci. Eng.* **39**(4), 3111–3121 (2014).
[38] A. Izadbakhsh, "Robust control design for rigid-link flexible-joint electrically driven robot subjected to constraint: Theory and experimental verification," *Nonlinear Dyn.* **85**(2), 751–765 (2016).
[39] L. Sciavicco and B. Siciliano. *Modelling and Control of Robot Manipulators* (Springer Science & Business Media, 2001).
[40] Y. Shang, "Finite-time consensus for multi-agent systems with fixed topologies," *Int. J. Syst. Sci.* **43**(3), 499–506 (2012).
[41] Y. Shang, "Couple-group consensus of continuous-time multi-agent systems under Markovian switching topologies," *J. Franklin Inst.* **352**(11), 4826–4844 (2015).
[42] A. C. Huang and M. C. Chien. *Adaptive Control of Robot Manipulators: A Unified Regressor-Free Approach* (World Scientific, 2010).
[43] J. C. Patra and A. C. Kot, "Nonlinear dynamic system identification using Chebyshev functional link artificial neural networks," *IEEE Trans. Syst. Man Cybern. Syst.* **32**(4), 505–511 (2002).
[44] 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**(4), 2789–2798 (2008).
[45] G. W. Stewart. *Introduction to Matrix Computations* (Elsevier, 1973).

https://doi.org/10.1017/S0263574722000303 Published online by Cambridge University Press

Appendix A

Lemma 1. *Suppose that the following equalities are satisfied.*

$$s_1 = d_m - \frac{(\kappa+1)}{\gamma} d_M > 0 \tag{A1}$$

$$s_2 = \underline{\lambda}(\mathbf{\Gamma}) - \frac{1}{\gamma} d_M > 0 \tag{A2}$$

$$s_3 = \kappa^2 \gamma \underline{\lambda}(\mathbf{\Gamma}) - \frac{\kappa}{\gamma} d_M > 0 \tag{A3}$$

As a result, $\mathbf{P}$ *is positive-definite, satisfying (A4).*

$$\underline{\lambda}(\mathbf{P})\|\mathbf{x}\|^2 \leq \mathbf{x}^T \mathbf{P} \mathbf{x} \leq \overline{\lambda}(\mathbf{P})\|\mathbf{x}\|^2 \tag{A4}$$

where

$$\overline{\lambda}(\mathbf{P}) = \max\{s_4, s_5, s_6\} \tag{A5}$$

$$\underline{\lambda}(\mathbf{P}) = \min\{s_1, s_2, s_3\} \tag{A6}$$

and

$$s_4 = \left(1 + \frac{(\kappa+1)}{\gamma}\right) d_M \tag{A7}$$

$$s_5 = \overline{\lambda}(\mathbf{\Gamma}) + \frac{1}{\gamma} d_M \tag{A8}$$

$$s_6 = \kappa \gamma^2 \overline{\lambda}(\mathbf{\Gamma}) + \frac{\kappa}{\gamma} d_M \tag{A9}$$

Proof. The lemma will be proved using the Gershgorin theorem [45]. Consider $\mathbf{T}$ as a transformation such that

$$\mathbf{D} = \mathbf{T}^{-1} \mathbf{\Xi} \mathbf{T} \tag{A10}$$

where $\mathbf{\Xi} = \text{diag}\{a_1, a_2, \ldots, a_n\}$ and a_is are the point-wise eigenvalues of $\mathbf{D}(\mathbf{q})$. The transformation is applied to (29), which results in

$$\begin{bmatrix} \mathbf{T}^{-1} & 0 & 0 \\ 0 & \mathbf{T}^{-1} & 0 \\ 0 & 0 & \mathbf{T}^{-1} \end{bmatrix} \mathbf{P} \begin{bmatrix} \mathbf{T} & 0 & 0 \\ 0 & \mathbf{T} & 0 \\ 0 & 0 & \mathbf{T} \end{bmatrix} = \begin{bmatrix} \mathbf{\Xi} & -\frac{1}{\gamma}\mathbf{\Xi} & \frac{\kappa}{\gamma}\mathbf{\Xi} \\ -\frac{1}{\gamma}\mathbf{\Xi} & \mathbf{\Gamma} & \mathbf{0}_n \\ \frac{\kappa}{\gamma}\mathbf{\Xi} & \mathbf{0}_n & \kappa\gamma^2\mathbf{\Gamma} \end{bmatrix} \tag{A11}$$

All the sub-matrices that appeared on the right-hand side of (A11) are diagonal. Based on the Gershgorin theorem, it is known that the following inequalities hold for the eigenvalues $\aleph_j, j = 1,\ldots,3n$, of the matrix on the right-hand side of (A11).

$$\begin{aligned} |\aleph_i - a_i| &\leq \frac{(\kappa+1)}{\gamma} a_i \\ |\aleph_{n+i} - \Gamma_i| &\leq \frac{1}{\gamma} a_i \\ \left|\aleph_{2n+i} - \kappa\gamma^2\Gamma_i\right| &\leq \frac{\kappa}{\gamma} a_i \qquad \text{i} = 1, 2, \ldots, \text{n} \end{aligned} \tag{A12}$$

https://doi.org/10.1017/S0263574722000303 Published online by Cambridge University Press

Or similarly

$$\begin{gathered} a_i - \frac{(\kappa+1)}{\gamma} a_i \le \aleph_i \le a_i + \frac{(\kappa+1)}{\gamma} a_i \\ \Gamma_i - \frac{1}{\gamma} a_i \le \aleph_{n+i} \le \Gamma_i + \frac{1}{\gamma} a_i \\ \kappa\gamma^2\Gamma_i - \frac{\kappa}{\gamma} a_i \le \aleph_{2n+i} \le \kappa\gamma^2\Gamma_i + \frac{\kappa}{\gamma} a_i \end{gathered} \tag{A13}$$

for $i = 1,\ldots,n$. The lemma proof is accomplished by noting that $d_m \le a_i \le d_M$ and $\underline{\lambda}(\mathbf{\Gamma}) \le \Gamma_i \le \overline{\lambda}(\mathbf{\Gamma})$ for $i = 1, 2,..,n$. □

Lemma 2. *Suppose the following equations are hold.*

$$s_1^* = \left(d_m - \frac{\kappa}{\gamma^2} d_M - \frac{1}{\gamma} d_M \right) > 0 \tag{A14}$$

$$s_2^* = \underline{\lambda}(\mathbf{\Gamma}) - \frac{1}{\gamma} d_M > 0 \tag{A15}$$

$$s_3^* = \kappa^2 \underline{\lambda}(\mathbf{\Gamma}) \tag{A16}$$

Then, $\mathbf{Q}$ *is positive-definite, and (A17) holds for it.*

$$\gamma \underline{\lambda}(\mathbf{Q}) \|x\|^2 \le x^T \mathbf{Q} x \le \overline{\lambda}(\mathbf{Q}) \|x\|^2 \tag{A17}$$

where

$$\underline{\lambda}(\mathbf{Q}) = \min\left\{ s_1^*, s_2^*, s_3^* \right\} \tag{A18}$$

$$\overline{\lambda}(\mathbf{Q}) = \max\left\{ s_4^*, s_5^*, s_6^* \right\} \tag{A19}$$

and

$$s_4^* = \left(1 + \gamma - \frac{\kappa}{\gamma} \right) d_M \tag{A20}$$

$$s_5^* = d_M + \gamma \overline{\lambda}(\mathbf{\Gamma}) \tag{A21}$$

$$s_6^* = \kappa^2 \gamma \overline{\lambda}(\mathbf{\Gamma}) \tag{A22}$$

Proof. The proof of this lemma is the same as Lemma 1. Based on the Gershgorin theorem, it is known that the following equations hold for the eigenvalues $\aleph_j^*, j = 1,\ldots,3n$.

$$\begin{gathered} \left(\gamma - \frac{\kappa}{\gamma} - 1 \right) a_i \le \aleph_i^* \le \left(\gamma - \frac{\kappa}{\gamma} + 1 \right) a_i \\ \gamma\Gamma_i - a_i \le \aleph_{n+i}^* \le \gamma\Gamma_i + a_i \\ \aleph_{2n+i}^* = \kappa^2 \gamma \Gamma_i \end{gathered} \tag{A23}$$

for $i = 1,\ldots, n$. The lemma proof is fulfilled by noting that $d_m \le a_i \le d_M$, and $\underline{\lambda}(\mathbf{\Gamma}) \le \Gamma_i \le \overline{\lambda}(\mathbf{\Gamma})$ for $i = 1,2,.., n$. □

https://doi.org/10.1017/S0263574722000303 Published online by Cambridge University Press

Appendix B

1. Derivation of (39)

Using the definitions of $\mathbf{P}$ and $\mathbf{B}$, it is straightforward to present that

$$\mathbf{x}^T\mathbf{PBHe}_I = \begin{bmatrix} \dot{\mathbf{e}}_q^T & \boldsymbol{\psi}^T & \mathbf{e}_q^T \end{bmatrix} \begin{bmatrix} \mathbf{D}(\mathbf{q}) & \frac{-1}{\gamma}\mathbf{D}(\mathbf{q}) & \frac{\kappa}{\gamma}\mathbf{D}(\mathbf{q}) \\ \frac{-1}{\gamma}\mathbf{D}(\mathbf{q}) & \boldsymbol{\Gamma} & \mathbf{0}_n \\ \frac{\kappa}{\gamma}\mathbf{D}(\mathbf{q}) & \mathbf{0}_n & \kappa\gamma^2\boldsymbol{\Gamma} \end{bmatrix} \begin{bmatrix} \mathbf{D}^{-1}(\mathbf{q}) \\ \mathbf{0}_n \\ \mathbf{0}_n \end{bmatrix} \mathbf{He}_I \tag{B1}$$

which is simplified as

$$\mathbf{x}^T\mathbf{PBHe}_I = \begin{bmatrix} \dot{\mathbf{e}}_q^T & \boldsymbol{\psi}^T & \mathbf{e}_q^T \end{bmatrix} \begin{bmatrix} \mathbf{I}_n \\ \frac{-1}{\gamma}\mathbf{I}_n \\ \frac{\kappa}{\gamma}\mathbf{I}_n \end{bmatrix} \mathbf{He}_I \leq \wp_0 \|\mathbf{x}\| \|\mathbf{e}_I\| \tag{B2}$$

where $\wp_0 = \overline{\lambda}(\mathbf{H})\sqrt{1+\frac{1}{\gamma^2}(1+\kappa^2)}$ stands for a positive gain. As a result, Eq. (39) is held.

2. Derivation of (40)

Using the definitions of $\mathbf{P}$ and $\mathbf{B}$, it is easy to present that

$$\begin{aligned} \frac{1}{2}\mathbf{x}^T\dot{\mathbf{P}}\mathbf{x} - \mathbf{x}^T\mathbf{PBC}(\mathbf{q},\dot{\mathbf{q}})\dot{\mathbf{e}}_q = & \frac{1}{2}\begin{bmatrix} \dot{\mathbf{e}}_q^T & \boldsymbol{\psi}^T & \mathbf{e}_q^T \end{bmatrix} \begin{bmatrix} \dot{\mathbf{D}}(\mathbf{q}) & \frac{-1}{\gamma}\dot{\mathbf{D}}(\mathbf{q}) & \frac{\kappa}{\gamma}\dot{\mathbf{D}}(\mathbf{q}) \\ \frac{-1}{\gamma}\dot{\mathbf{D}}(\mathbf{q}) & \mathbf{0}_n & \mathbf{0}_n \\ \frac{\kappa}{\gamma}\dot{\mathbf{D}}(\mathbf{q}) & \mathbf{0}_n & \mathbf{0}_n \end{bmatrix} \begin{bmatrix} \dot{\mathbf{e}}_q \\ \boldsymbol{\psi} \\ \mathbf{e}_q \end{bmatrix} \\ & - \begin{bmatrix} \dot{\mathbf{e}}_q^T & \boldsymbol{\psi}^T & \mathbf{e}_q^T \end{bmatrix} \begin{bmatrix} \mathbf{D}(\mathbf{q}) & \frac{-1}{\gamma}\mathbf{D}(\mathbf{q}) & \frac{\kappa}{\gamma}\mathbf{D}(\mathbf{q}) \\ \frac{-1}{\gamma}\mathbf{D}(\mathbf{q}) & \boldsymbol{\Gamma} & \mathbf{0}_n \\ \frac{\kappa}{\gamma}\mathbf{D}(\mathbf{q}) & \mathbf{0}_n & \kappa\gamma^2\boldsymbol{\Gamma} \end{bmatrix} \begin{bmatrix} \mathbf{D}^{-1}(\mathbf{q}) \\ \mathbf{0}_n \\ \mathbf{0}_n \end{bmatrix} \mathbf{C}(\mathbf{q},\dot{\mathbf{q}})\dot{\mathbf{e}}_q \end{aligned} \tag{B3}$$

Further simplifications of the right-hand side of (B3) lead to

$$\frac{1}{2}\mathbf{x}^T\dot{\mathbf{P}}\mathbf{x} - \mathbf{x}^T\mathbf{PBC}(\mathbf{q},\dot{\mathbf{q}})\dot{\mathbf{e}}_q = \frac{1}{2}\dot{\mathbf{e}}_q^T\big(\dot{\mathbf{D}}(\mathbf{q}) - 2\mathbf{C}(\mathbf{q},\dot{\mathbf{q}})\big)\dot{\mathbf{e}}_q + \frac{1}{\gamma}\big(\kappa\mathbf{e}_q^T - \boldsymbol{\psi}^T\big)\big(\dot{\mathbf{D}}(\mathbf{q}) - \mathbf{C}(\mathbf{q},\dot{\mathbf{q}})\big)\dot{\mathbf{e}}_q \tag{B4}$$

Based on property 3 (i.e., the skew-symmetric property) and some simplification, (B5) is derived.

$$\frac{1}{2}\mathbf{x}^T\dot{\mathbf{P}}\mathbf{x} - \mathbf{x}^T\mathbf{PBC}(\mathbf{q},\dot{\mathbf{q}})\dot{\mathbf{e}}_q = \frac{1}{\gamma}\big(\kappa\mathbf{e}_q^T - \boldsymbol{\psi}^T\big)\,\mathbf{C}(\mathbf{q},\dot{\mathbf{q}})\dot{\mathbf{e}}_q \tag{B5}$$

Now, based on property 2, it is concluded that

$$\frac{1}{2}\mathbf{x}^T\dot{\mathbf{P}}\mathbf{x} - \mathbf{x}^T\mathbf{PBC}(\mathbf{q},\dot{\mathbf{q}})\dot{\mathbf{e}}_q \leq \frac{\xi_c}{\gamma}\sqrt{1+\kappa^2}\|\dot{\mathbf{q}}\|\|\mathbf{x}\|\,\dot{\mathbf{e}}_q \tag{B6}$$

https://doi.org/10.1017/S0263574722000303 Published online by Cambridge University Press

Using the fact that $\left\|\dot{\mathbf{e}}_q\right\| \leq \|\mathbf{x}\|$ and $\left\| \boldsymbol{\psi}^T \quad \mathbf{e}_q^T \right\| \leq \|\mathbf{x}\|$, (B7) results.

$$\frac{1}{2}\mathbf{x}^T\dot{\mathbf{P}}\mathbf{x} - \mathbf{x}^T\mathbf{PBC}(\mathbf{q}, \dot{\mathbf{q}})\dot{\mathbf{e}}_q \leq \frac{\wp_1}{\gamma}\|\dot{\mathbf{q}}\|\|\mathbf{x}\|^2 \tag{B7}$$

where $\wp_1 = \xi_c\sqrt{1+\kappa^2}$. Then, based on (B7), the inequality (40) is held.

3. Derivation of (41)

Using property 2 and based on the same simplification as before, it is clear that

$$\begin{aligned} -\mathbf{x}^T\mathbf{PBC}\left(\mathbf{q}, \dot{\mathbf{e}}_q\right)\dot{\mathbf{q}}_d &= -\begin{bmatrix} \dot{\mathbf{e}}_q^T & \boldsymbol{\psi}^T & \mathbf{e}_q^T \end{bmatrix} \begin{bmatrix} \mathbf{I}_n \\ \frac{-1}{\gamma}\mathbf{I}_n \\ \frac{\kappa}{\gamma}\mathbf{I}_n \end{bmatrix} \mathbf{C}\left(\mathbf{q}, \dot{\mathbf{e}}_q\right)\dot{\mathbf{q}}_d \\ &\leq \wp_2\|\mathbf{x}\|\left\|\dot{\mathbf{e}}_q\right\| \leq \wp_2\|\mathbf{x}\|^2 \end{aligned} \tag{B8}$$

where $\wp_2 = \sqrt{1+\frac{1}{\gamma^2}(1+\kappa^2)}\xi_c V_M$ and we have used the fact that $\left\|\dot{\mathbf{e}}_q\right\| \leq \|\mathbf{x}\|$.

Cite this article: A. Izadbakhsh, A. A. Kalat and N. Nikdel (2022). "FAT-based robust adaptive controller design for electrically direct-driven robots using Phillips q-Bernstein operators", Robotica **40**, 3415–3434. https://doi.org/10.1017/S0263574722000303

https://doi.org/10.1017/S0263574722000303 Published online by Cambridge University Press