Research Article

INTERNATIONAL JOURNAL OF ADVANCED ROBOTIC SYSTEMS

International Journal of Advanced Robotic Systems
March-April 2018: 1–13

DOI: 10.1177/1729881418767307
journals.sagepub.com/home/arx
SAGE

A neural network–based synchronized computed torque controller for three degree-of-freedom planar parallel manipulators with uncertainties compensation

Quang Vinh Doan[1], Tien Dung Le[1], Quang Dan Le[2] and Hee-Jun Kang[3]

Abstract

This study presents a new adaptive synchronized computed torque control algorithm based on neural networks for three degree-of-freedom planar parallel manipulators. The basic idea of the proposed control algorithm is to use the incorporation of cross-coupling errors of active joints with the adaptive computed torque control algorithm, online self-tuned neural networks, and error compensators. The key to the success of the proposed approach is to improve the trajectory tracking accuracy of the parallel manipulator's end-effector while driving the synchronization errors among active joints to zero. The uncertainties of the control system such as modeling errors, frictional terms, and external disturbances are adaptively compensated online during the trajectory tracking of the parallel manipulator. Using the Lyapunov theory, it is proved that the tracking errors and error rates of the overall system asymptotically converge to zero. To demonstrate the effectiveness of the proposed control algorithm, compared simulations are conducted using MATLAB/Simulink [version 2013a] combined with Solidworks 2014.

Keywords

Parallel manipulator, synchronized control, computed torque control, uncertainties compensation, neural networks, online tuning

Date received: 25 December 2017; accepted: 23 February 2018

Topic: Robot Manipulation and Control
Topic Editor: Andrey V Savkin
Associate Editor: Yongping Pan

Introduction

In the last recent decades, research on the control of parallel manipulators has had drawn a lot of interest in the robotics community. This is because parallel manipulators have potential advantages in terms of high rigidity, high speed, high accuracy, and high load-carrying capacity over serial robotic manipulators. They are widely used in numerous applications such as simulators, humanoid robots, medical robots, micro-robots, and high-accuracy pick-and-place positioning and are becoming increasingly popular in the

[1] The University of Danang – University of Science and Technology, Danang, Vietnam
[2] Graduate School of Electrical Engineering, University of Ulsan, Ulsan, South Korea
[3] School of Electrical Engineering, University of Ulsan, Ulsan, South Korea

Corresponding author:
Tien Dung Le, Faculty of Electrical Engineering, The University of Danang – University of Science and Technology, 54 Nguyen Luong Bang street, Danang 550000, Vietnam.
Email: ltdung@dut.udn.vn

strict requirements such as small product size and shorter assembly time.

In the literature on tracking control of parallel manipulators, numerous control methods have been proposed, which could be classified as two kinds of approaches. The first one is model-free control approach, such as proportional–derivative (PD) controller,[1,2] nonlinear PD controller,[3,4] and adaptive switching learning PD control method.[5] The other one is model-based control approach, such as computed torque controllers,[6–10] the model-based iterative learning controller,[11] sliding mode controllers,[12–16] and the adaptive controllers.[17–21] The common characteristic of these approaches is that only local feedback information of the controlled joint is fed to the control loop of each individual actuator. Feedback signal from other actuated joints cannot be received. As a result, errors caused by disturbances in the control loop of one actuator are corrected by this loop only, while others do not respond. In parallel manipulators, the trajectory of the end-effector is led by all actuator motions. Therefore, all actuated joints in parallel manipulators should be synchronously controlled to increase the tracking accuracy.

Many works on synchronized control of parallel manipulators have been conducted in recent years. This control approach involves kinematic coupling among the active joints of the parallel manipulators, which results in higher tracking accuracy of the end-effector. The synchronized control approach was first proposed by Koren.[22] In his study, the cross-coupling control was used for a multi-axis machining tool to achieve the multi-axis tracking synchronization. It is the employment of the synchronization error that makes the synchronized control different from traditional proportional–integral–derivative control. This useful approach has become more popular in position synchronization control of multiple motion axes,[23,24] mobile robot control,[25,26] and tracking control of parallel manipulators.[27–32]

For tracking control of single parallel manipulators, Sun et al.[27] presented a cross-coupled control approach to the tracking control of parallel manipulators in a synchronous manner. On the other hand, Ren et al.[28] have proposed a convex synchronized control algorithm for a three degree-of-freedom (3-DOF) planar parallel manipulator. The control algorithm is based on the combination of the advantages of the convex combination method and the synchronized control method. In another approach, a synchronized control algorithm, which does not need dynamic model of parallel manipulators, has been presented such as integration of saturated proportional–integral synchronous control and PD feedback control algorithm.[29] In addition, Ren et al.[30] presented an adaptive synchronized control method for parallel manipulator based on the combination of synchronized control and adaptive control. Shang et al.[31] proposed an active joint-synchronization controller to solve control problem of redundantly actuated parallel manipulators. Ren et al.,[32] in an experimental comparison study of the synchronized control approaches, have shown that the synchronized control methods based on a dynamic model of the robot can achieve better performance than the model-free ones. However, the model-based synchronized control algorithms are much more complex than the model-free ones due to the complexity of the dynamic model of parallel manipulators. In addition, the aforementioned methods are still complex and require numerous computation to be applied.

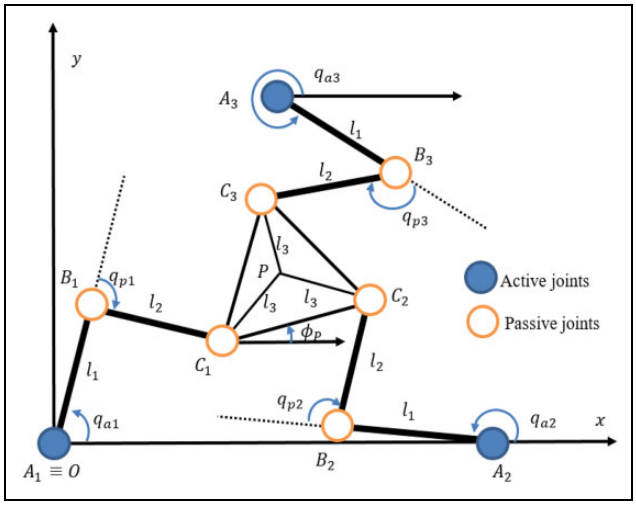

Figure 1. Three-DOF planar parallel manipulator. DOF: degree of freedom.

In this article, we propose a new synchronized control algorithm for 3-DOF planar parallel manipulators. The control algorithm is based on the combination of synchronization error and cross-coupling error with the computed torque control algorithm and uncertainties compensation method. The uncertainties of the control system such as modeling errors, frictional terms, and external disturbances are adaptively compensated by radial basis function neural networks (RBFNNs) and error compensators. The online weight tuning algorithms of the neural networks and error compensators are derived with the strict theoretical stability proof of the Lyapunov theorem.

The rest of this article is organized as follows. In the second section, the kinematic modeling and dynamic modeling of 3-DOF planar parallel manipulators are formulated in the active joint space. The proposed synchronized computed torque controller (SCTC) is presented in the third section. The comparative simulations are carried out in fourth section in order to verify the effectiveness of the proposed controller. Finally, a conclusion is reached in the fifth section.

Kinematic modeling and dynamic modeling of 3-DOF planar parallel manipulators

The 3-DOF planar parallel robot manipulator is shown in Figure 1. It works on a horizontal plane and in a reference frame *Oxy*. The manipulator consists of three active joints

and six passive joints. The active joints are actuated by actuators, while the passive ones can move freely. The end-effector of the robot is a triangle $C_1C_2C_3$, which connects the ending points of three 2-DOF serial robot manipulators. P is the center point of the end-effector. The 3-DOF planar parallel robot manipulator is controlled by actuating three active joints A_1, A_2, and A_3.

We denote the vectors of variable as follows

- $\boldsymbol{q_a} = [q_{a1},\ q_{a2},\ q_{a3}]^T$ is the vector of active joint angulars.
- $\boldsymbol{q_p} = [q_{p1},\ q_{p2},\ q_{p3}]^T$ is the vector of passive joint angulars.
- $\boldsymbol{X} = [x_P,\ y_P,\ \phi_P]^T$ is the vector of Cartesian coordinates and rotate angle of the end-effector.

In the control system, for the 3-DOF planar parallel robot manipulator, the control signal will be sent to drive three active joints of the robot. Therefore, we need to develop kinematic model and dynamic model in active joint space.

Kinematic modeling

Forward kinematics. In the forward kinematics problem, we compute the end-effector's coordinate $\boldsymbol{X}(x_p, y_p, \phi_p)$ in the reference A_1xy from the given active joint angular vector $\boldsymbol{q_a}(q_{a1},\ q_{a2},\ q_{a3})$.

From Figure 1, we have the following equation system

$$\begin{aligned}&[x_P - x_{Ai} - l_1 \cos q_{ai} - l_3 \cos(\phi_P + \psi_i)]^2 \\ &\quad + [y_P - y_{Ai} - l_1 \sin q_{ai} - l_3 \sin(\phi_P + \psi_i)]^2 \\ &= l_2^2, \quad i = 1, 2, 3\end{aligned} \tag{1}$$

in which ψ_i ($i = 1, 2, 3$) are, respectively, equal to $7\pi/6$, $11\pi/6$, and $\pi/2$.

Solving the equation system (1) by applying the numerical method, we obtain coordination x_P, y_P, and the value of ϕ_P with given input $\boldsymbol{q_a}$.

In the control system, we need to know the value of passive joint angles. After having the output $\boldsymbol{X}(x_P, y_P, \phi_P)$, we can compute the value of passive joint angles using the following equation:

$$q_{pi} = \pi - \cos^{-1}\left(\frac{l_1^2 + l_2^2 - x_{Ci}^2 - y_{Ci}^2}{2l_1 l_2}\right),\ i = 1, 2, 3 \tag{2}$$

where the coordinates of the points C_i ($i = 1, 2, 3$) are computed as follows

$$x_{Ci} = x_P + l_3 \cos(\phi_P + \psi_i) - x_{Ai}, \quad i = 1, 2, 3 \tag{3}$$

$$y_{Ci} = x_P + l_3 \sin(\phi_P + \psi_i) - y_{Ai}, \quad i = 1, 2, 3 \tag{4}$$

Inverse kinematics. The inverse kinematics problem is to obtain the active joint angular vector $\boldsymbol{q_a}(q_{a1},\ q_{a2},\ q_{a3})$ from the given coordinates $\boldsymbol{X}(x_P,\ y_P,\ \phi_P)$ of the end-effector.

From Figure 1, we have the following equations of the inverse kinematics

$$q_{ai} = \tan^{-1}\left(\frac{y_{Ci}}{x_{Ci}}\right) + \cos^{-1}\left(\frac{l_1^2 - l_2^2 + x_{Ci}^2 + y_{Ci}^2}{2l_1\sqrt{x_{Ci}^2 + y_{Ci}^2}}\right), \quad i = 1, 2, 3 \tag{5}$$

Jacobian matrices. We have the following equations obtained from Figure 1

$$\begin{bmatrix} x_P \\ y_P \end{bmatrix} = \begin{bmatrix} x_{oi} + l_1 \cos q_{ai} + l_2 \cos(q_{ai} + q_{pi}) & + l_3 \cos(\psi_i + \phi_P) \\ y_{oi} + l_1 \sin q_{ai} + l_2 \sin(q_{ai} + q_{pi}) & + l_3 \sin(\psi_i + \phi_P) \end{bmatrix}, \quad i = 1, 2, 3 \tag{6}$$

By differentiating equation (6) with respect to time, we obtain the following

$$\begin{bmatrix} \dot{x}_P \\ \dot{y}_P \end{bmatrix} = \begin{bmatrix} -l_1\dot{q}_{ai}\sin q_{ai} - l_2(\dot{q}_{ai} + \dot{q}_{pi})\sin(q_{ai} + q_{pi}) & -l_3\dot{\phi}_P \sin(\psi_i + \phi_P) \\ l_1\dot{q}_{ai}\cos q_{ai} + l_2(\dot{q}_{ai} + \dot{q}_{pi})\cos(q_{ai} + q_{pi}) & +l_3\dot{\phi}_P \cos(\psi_i + \phi_P) \end{bmatrix}, \quad i = 1, 2, 3 \tag{7}$$

Rearranging equation (7) leads to an equation under the matrix form

$$\boldsymbol{J}_{z1}\dot{\boldsymbol{X}} = \boldsymbol{J}_o\dot{\boldsymbol{q}}_a \tag{8}$$

where $\dot{\boldsymbol{X}} = [\dot{x}_P,\ \dot{y}_P,\ \dot{\phi}_P]^T$ corresponds to the velocity of end-effector in the Cartesian coordinate, and $\dot{\boldsymbol{q}}_a = [\dot{q}_{a1},\ \dot{q}_{a2},\ \dot{q}_{a3}]^T$ corresponds to the angular velocity vector of active joints.

Jacobian matrices are expressed by the following equations

$$\boldsymbol{J}_{z1} = \begin{bmatrix} az1_1 & bz1_1 & cz1_1 \\ az1_2 & bz1_2 & cz1_2 \\ az1_3 & bz1_3 & cz1_3 \end{bmatrix} \tag{9}$$

$$\boldsymbol{J}_o = \begin{bmatrix} dz1_1 & 0 & 0 \\ 0 & dz1_2 & 0 \\ 0 & 0 & dz1_3 \end{bmatrix} \tag{10}$$

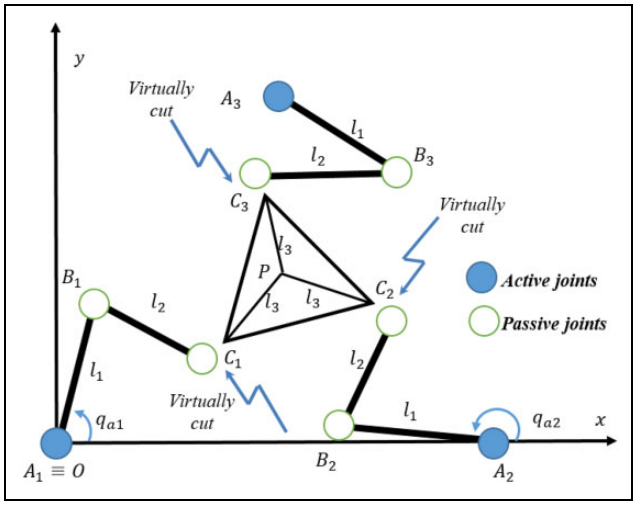

Figure 2. The equivalent open-chain system obtained by virtually cut at C_1, C_2, and C_3.

in which

$$\begin{cases} az1_i = \cos(q_{ai} + q_{pi}) \\ bz1_i = \sin(q_{ai} + q_{pi}) \\ cz1_i = -l_3 \sin(q_{ai} + q_{pi} - \psi_i - \phi_P) \\ dz1_i = l_1 \sin q_{pi} \end{cases} ; \quad i = 1, 2, 3 \tag{11}$$

Finally, equation (8) could be rewritten as follows

$$\dot{\boldsymbol{q}}_{\boldsymbol{a}} = \boldsymbol{J}_{\boldsymbol{oz}} \dot{\boldsymbol{X}} \tag{12}$$

where $\boldsymbol{J}_{\boldsymbol{oz}}$ is the Jacobian matrix

$$\boldsymbol{J}_{\boldsymbol{oz}} = \boldsymbol{J}_o^{-1} \boldsymbol{J}_{z1} \tag{13}$$

Equation (13) could be used for singularity analysis. It could be seen that the singularities of the 3-DOF planar parallel manipulator occur whenever $\boldsymbol{J}_o^{-1}$, $\boldsymbol{J}_{z1}$, or both become singular.

Dynamic modeling

In this section, the dynamic modeling of the 3-DOF planar parallel manipulator is presented in active joint space. For deriving the dynamic model, we follow the method presented by Le and Kang.[33] First, three passive joints C_1, C_2, and C_3 are virtually cut to form an equivalent open-chain system as shown in Figure 2. Second, using the Lagrangian approach, the dynamic model of the equivalent open-chain system can be computed and obtained as follows[34]

$$\boldsymbol{M_O}(\boldsymbol{q})\,\ddot{\boldsymbol{q}} + \boldsymbol{C_O}(\boldsymbol{q}, \dot{\boldsymbol{q}})\dot{\boldsymbol{q}} + \boldsymbol{F_O} = \boldsymbol{\tau_O} \tag{14}$$

in which $\boldsymbol{M_O} \in \mathbb{R}^{9\times 9}$ is the inertia matrix, $\boldsymbol{C_O} \in \mathbb{R}^{9\times 9}$ is the Coriolis and centrifugal force matrix, $\boldsymbol{F_O}$ is the vector of friction forces, $\boldsymbol{\tau_O} = [\boldsymbol{\tau}_a, \boldsymbol{\tau}_p]^T \in \mathbb{R}^{9\times 1}$ is the joint torque vector of the equivalent open-chain system, $\boldsymbol{\tau}_a \in \mathbb{R}^{3\times 1}$ is the active joint torque vector, and $\boldsymbol{\tau}_p \in \mathbb{R}^{6\times 1}$ is the passive joint torque vector. The matrices $\boldsymbol{M}_O$ and $\boldsymbol{C}_O$ are computed by the following equations

$$\boldsymbol{M_O} = \begin{bmatrix} \delta_1 & 0 & 0 & \varepsilon_1 & 0 & 0 & 0 & 0 & 0 \\ 0 & \delta_2 & 0 & 0 & \varepsilon_2 & 0 & 0 & 0 & 0 \\ 0 & 0 & \delta_3 & 0 & 0 & \varepsilon_3 & 0 & 0 & 0 \\ \eta_1 & 0 & 0 & \beta_1 & 0 & 0 & 0 & 0 & 0 \\ 0 & \eta_2 & 0 & 0 & \beta_2 & 0 & 0 & 0 & 0 \\ 0 & 0 & \eta_3 & 0 & 0 & \beta_3 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & \boldsymbol{m_P} & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & \boldsymbol{m_P} & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \boldsymbol{I_P} \end{bmatrix} \tag{15}$$

$$\boldsymbol{C_O} = \begin{bmatrix} \vartheta_1 & 0 & 0 & \mu_1 & 0 & 0 & 0 & 0 & 0 \\ 0 & \vartheta_2 & 0 & 0 & \mu_2 & 0 & 0 & 0 & 0 \\ 0 & 0 & \vartheta_3 & 0 & 0 & \mu_3 & 0 & 0 & 0 \\ \rho_1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & \rho_2 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & \rho_3 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \tag{16}$$

$$\boldsymbol{F_O} = [f_{a1}, f_{a2}, f_{a3}, f_{b1}, f_{b2}, f_{b3}, f_{c1}, f_{c2}, f_{c3}]^T \tag{17}$$

where $\delta_i = \alpha_i + \beta_i + 2\gamma_i \cos\boldsymbol{q}_{pi}$, $\varepsilon_i = \beta_i + \gamma_i \cos\boldsymbol{q}_{pi}$, $\vartheta_i = -2\gamma_i \dot{\boldsymbol{q}}_{pi} \sin\boldsymbol{q}_{pi}$, $\eta_i = \beta_i + \gamma_i \cos\boldsymbol{q}_{pi}$, $\alpha_i = m_{i1} l_{c1}^2 + m_{i1} l_2^2 + I_{i1}$, $\beta_i = m_{i2} l_{c2}^2 + I_{i2}$, $\gamma_i = m_{i2} l_{c1} l_{c2}$, and f_{ai}, f_{bi}, and f_{ci} are the friction forces at joints of the robot ($i = 1, 2, 3$).

Since the parallel manipulator is controlled by actuating three active joints A_1, A_2, and A_3, we need to develop the dynamic model in the active joint space with the input torque $\boldsymbol{\tau}_a = (\boldsymbol{\tau}_{a1}, \boldsymbol{\tau}_{a2}, \boldsymbol{\tau}_{a3})^T \in \mathbb{R}^{3\times 1}$. Following the D'Alembert principle and principle of virtual work, we have the following relationship[35]

$$\boldsymbol{W}^T \tau_O = \tau_a \tag{18}$$

where $\boldsymbol{W}$ is the Jacobian matrix

$$\boldsymbol{W} = \begin{bmatrix} \boldsymbol{I} \\ \dfrac{\partial \boldsymbol{q}_p}{\partial \boldsymbol{q}_a} \\ \dfrac{\partial \boldsymbol{X}_p}{\partial \boldsymbol{q}_a} \end{bmatrix} = \begin{bmatrix} \boldsymbol{I} \\ \boldsymbol{J}_1 \\ \boldsymbol{J}_2 \end{bmatrix} = \begin{bmatrix} \boldsymbol{I} \\ \boldsymbol{J}_{oz}^{-1} \boldsymbol{J}_{pz} \\ \boldsymbol{J}_{oz}^{-1} \end{bmatrix} \tag{19}$$

where $\boldsymbol{I} \in \mathbb{R}^{3\times 3}$ is the identity matrix. The matrices $\boldsymbol{J}_{\boldsymbol{pz}}$ is computed as follows

$$\boldsymbol{J}_{\boldsymbol{pz}} = \boldsymbol{J}_p^{-1} \boldsymbol{J}_{z2} \tag{20}$$

where

$$\boldsymbol{J}_p = \begin{bmatrix} dz2_1 & 0 & 0 \\ 0 & dz2_2 & 0 \\ 0 & 0 & dz2_3 \end{bmatrix} \tag{21}$$

$$\boldsymbol{J}_{z2} = \begin{bmatrix} az2_1 & bz2_1 & cz2_1 \\ az2_2 & bz2_2 & cz2_2 \\ az2_3 & bz2_3 & cz2_3 \end{bmatrix} \tag{22}$$

$$\begin{cases} az2_i = l_1\ \cos q_{ai} + l_{c2}\ \cos(q_{ai} + q_{pi}) \\ bz2_i = l_1\ \sin q_{ai} + l_{c2}\ \sin(q_{ai} + q_{pi}) \\ cz2_i = l_2 l_3\ \sin(\psi_i + \phi_P - q_{ai} - q_{pi}) + l_1 l_3\ \sin(\psi_i + \phi_P - q_{ai}) \\ dz2_i = -l_{c1} l_{c2}\ \sin q_{pi} \end{cases}, \quad i = 1, 2, 3 \tag{23}$$

By multiplying both sides of equation (14) by $\boldsymbol{W}^{\boldsymbol{T}}$ and using the relationship in equation (15), we obtain the following

$$\boldsymbol{W}^T \left(\boldsymbol{M_O}(\boldsymbol{q})\, \ddot{\boldsymbol{q}} + \boldsymbol{C_O}(\boldsymbol{q}, \dot{\boldsymbol{q}}) \dot{\boldsymbol{q}} + \boldsymbol{F_O} \right) = \boldsymbol{\tau}_a \tag{24}$$

In addition, we have the following relationships

$$\dot{\boldsymbol{q}} = [\boldsymbol{I}, \partial \boldsymbol{q}_p / \partial \boldsymbol{q}_a, \partial \boldsymbol{X_P} / \partial \boldsymbol{q}_a]^T \dot{\boldsymbol{q}}_a = \boldsymbol{W} \dot{\boldsymbol{q}}_a \tag{25}$$

$$\ddot{\boldsymbol{q}} = \dot{\boldsymbol{W}} \dot{\boldsymbol{q}}_a + \boldsymbol{W} \ddot{\boldsymbol{q}}_a \tag{26}$$

By substituting equations (25) and (26) into equation (24) and rearranging the new equation, we obtain the following

$$\widehat{\boldsymbol{M}}_a \ddot{\boldsymbol{q}}_a + \widehat{\boldsymbol{C}}_a \dot{\boldsymbol{q}}_a + \boldsymbol{F}_a = \boldsymbol{\tau}_a \tag{27}$$

where $\widehat{\boldsymbol{M}}_a = \boldsymbol{W}^T \boldsymbol{M}_O \boldsymbol{W} \in \mathbb{R}^{3\times 3}$ is the estimated inertia matrix, $\widehat{\boldsymbol{C}}_a = \boldsymbol{W}^T \boldsymbol{M}_O \dot{\boldsymbol{W}} + \boldsymbol{W}^T \boldsymbol{C}_O \boldsymbol{W} \in \mathbb{R}^{3\times 3}$ is the estimated Coriolis and centrifugal force matrix, and $\boldsymbol{F}_a = \boldsymbol{W}^T \boldsymbol{F}_O \in \mathbb{R}^{3\times 1}$ is the vector of friction torques in active joints.

The dynamic model in equation (27) satisfies the following properties, which were proved by Cheng et al.[36]

1. $\widehat{\boldsymbol{M}}_a$ is a symmetric and positive definite matrix.
2. $\dot{\widehat{\boldsymbol{M}}}_a - 2\widehat{\boldsymbol{C}}_a$ is a skew-symmetric matrix.

In the presence of bounded uncertainties such as errors of dynamic model and external disturbances, we can express the actual dynamic model of the parallel manipulator by the following equation

$$\boldsymbol{M}_a \ddot{\boldsymbol{q}}_a + \boldsymbol{C}_a \dot{\boldsymbol{q}}_a + \boldsymbol{F}_a + \boldsymbol{d}_a = \boldsymbol{\tau}_a \tag{28}$$

where $\boldsymbol{M}_a = \widehat{\boldsymbol{M}}_a + \Delta \boldsymbol{M}_a$ and $\boldsymbol{C}_a = \widehat{\boldsymbol{C}}_a + \Delta \boldsymbol{C}_a$ are the actual dynamic parameters, $\boldsymbol{d}_a$ is the vector of external disturbances on active joints, and $\Delta \boldsymbol{M}_a$ and $\Delta \boldsymbol{C}_a$ are dynamic modeling errors.

Finally, we obtain the actual dynamic equation of the 3-DOF planar parallel manipulator as follows

$$\widehat{\boldsymbol{M}}_a \ddot{\boldsymbol{q}}_a + \widehat{\boldsymbol{C}}_a \dot{\boldsymbol{q}}_a + \Delta \boldsymbol{\tau}_a = \boldsymbol{\tau}_a \tag{29}$$

in which $\Delta \boldsymbol{\tau}_a$ is the vector of modeling errors, friction forces, and external disturbances

$$\Delta \boldsymbol{\tau}_a = \Delta \boldsymbol{M}_a \ddot{\boldsymbol{q}}_a + \Delta \boldsymbol{C}_a \dot{\boldsymbol{q}}_a + \boldsymbol{F}_a + \boldsymbol{d}_a \tag{30}$$

Proposed adaptive SCTC

Define the tracking error

$$e = \boldsymbol{q}_{da}(t) - \boldsymbol{q}_a(t) \tag{31}$$

in which $\boldsymbol{q}_{da}(t) \in \mathbb{R}^{3\times 1}$ is the desired trajectory of the active joint angles.

In the synchronized control methods, not only the tracking error of each individual active joint must come to zero ($e_{ai}(t) \to 0$, $i = 1, 2, 3$) but also the tracking errors of all active joints must be equal during the trajectory tracking control

$$e_{a1}(t) = e_{a2}(t) = e_{a3}(t) \tag{32}$$

The synchronization errors are defined as follows

$$\begin{aligned} \mathcal{E}_1(t) &= e_{a1}(t) - e_{a2}(t) \\ \mathcal{E}_2(t) &= e_{a2}(t) - e_{a3}(t) \\ \mathcal{E}_3(t) &= e_{a3}(t) - e_{a1}(t) \end{aligned} \tag{33}$$

The vector of synchronization error is $\mathcal{E} = [\mathcal{E}_1(t), \mathcal{E}_2(t), \mathcal{E}_3(t)]^T$. Under the synchronization error definition, control torques $\boldsymbol{\tau}_{ai}$ are to control position errors $e_{ai}(t) \to 0$

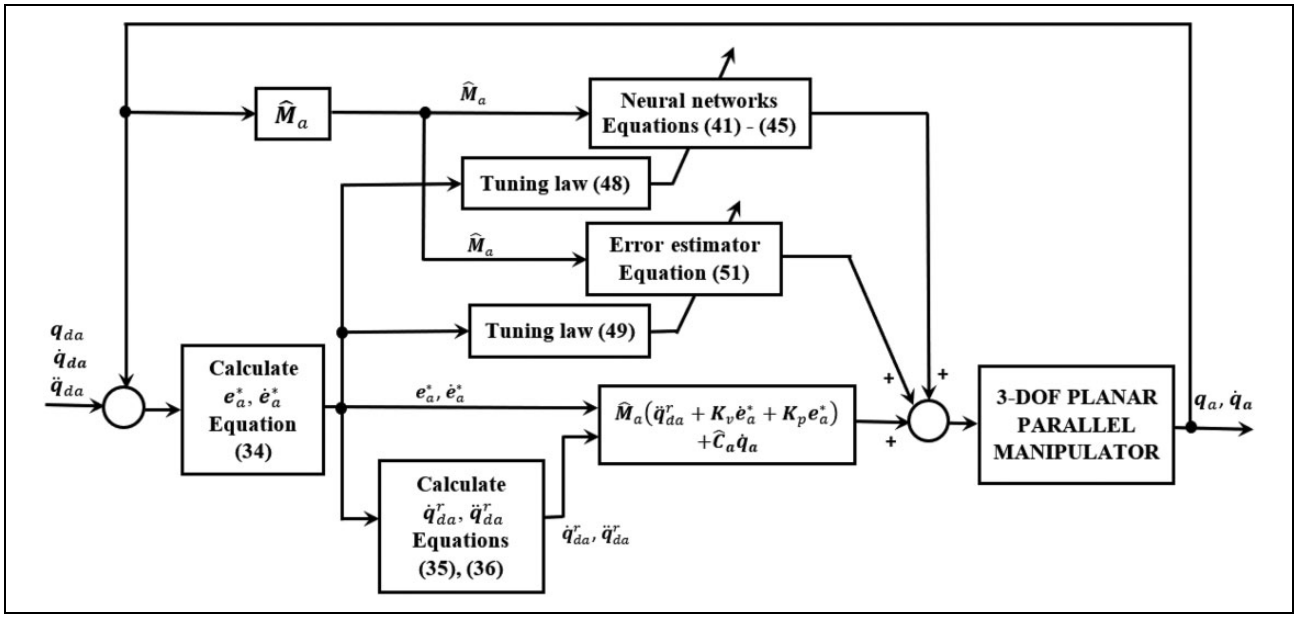

Figure 3. Flow diagram of the proposed synchronized computed torque controller.

and, at the same time, to synchronize the motions of active joints ($i = 1, 2, 3$).

To accommodate both position error and synchronization error, cross-coupling error insight has been introduced to provide an effective method to eliminate the interconnections among multiple motion axes systems. The cross-coupling error is defined by the following

$$\begin{cases} e_{a1}^* = e_{a1} + \sigma \int_0^t (\varepsilon_1 - \varepsilon_3)\, \mathrm{d}w \\ e_{a2}^* = e_{a2} + \sigma \int_0^t (\varepsilon_2 - \varepsilon_1)\, \mathrm{d}w \\ e_{a3}^* = e_{a3} + \sigma \int_0^t (\varepsilon_3 - \varepsilon_2)\, \mathrm{d}w \end{cases} \tag{34}$$

in which σ is the positive parameter and w is the time variable.

The reference tracking velocity and the reference tracking acceleration are defined as follows

$$\dot{\boldsymbol{q}}_{da}^r = \dot{\boldsymbol{q}}_{da} + \sigma\Delta\varepsilon \tag{35}$$

$$\ddot{\boldsymbol{q}}_{da}^r = \ddot{\boldsymbol{q}}_{da} + \sigma\Delta\dot{\varepsilon} \tag{36}$$

in which $\Delta\varepsilon = [\varepsilon_1 - \varepsilon_3,\ \varepsilon_2 - \varepsilon_1, \varepsilon_3 - \varepsilon_2]^T$ and $\Delta\dot{\varepsilon} = [\dot{\varepsilon}_1 - \dot{\varepsilon}_3, \dot{\varepsilon}_2 - \dot{\varepsilon}_1, \dot{\varepsilon}_3 - \dot{\varepsilon}_2]^T$.

The proposed controller is expressed by the following equation

$$\boldsymbol{\tau}_a = \widehat{\boldsymbol{M}}_a(\ddot{\boldsymbol{q}}_{da}^r + \boldsymbol{K}_v\dot{\boldsymbol{e}}_a^* + \boldsymbol{K}_p\boldsymbol{e}_a^*) + \widehat{\boldsymbol{C}}_a\dot{\boldsymbol{q}}_a + \boldsymbol{\tau}_{cp} + \boldsymbol{\tau}_{er} \tag{37}$$

where $\boldsymbol{e}_a^* = [e_{a1}^*, e_{a2}^*,\ e_{a3}^*]^T$ is the vector of cross-coupling errors, $\boldsymbol{K}_v = \mathrm{diag}\{k_{v1}, k_{v2}, k_{v3}\}$ and $\boldsymbol{K}_p = \mathrm{diag}\{k_{p1}, k_{p2}, k_{p3}\}$ are the positive constant parameter matrices, and $\boldsymbol{\tau}_{cp} \in \Re^{3\times 1}$ is the uncertainty compensation controller which contains a bank of neural network inside for estimating vector $\Delta\boldsymbol{\tau}_a$. Since the term $\boldsymbol{\tau}_{cp}$ cannot perfectly estimate $\Delta\boldsymbol{\tau}_a$, the term $\boldsymbol{\tau}_{er}$ is added to compensate the error of the estimation. The block diagram of the proposed controller is shown in Figure 3.

The proposed controller has some major contributions. First, the combination of synchronization error and cross-coupling error with computed torque control algorithm brings about the advantages of the two methods such as the high accuracy and low online computational burden. The second contribution is the error estimators with online adaptation law, which helps to compensate the estimation errors of the neural network.

From equations (37) and (30), we obtain the following

$$\ddot{\boldsymbol{e}}_a^* + \boldsymbol{K}_v\dot{\boldsymbol{e}}_a^* + \boldsymbol{K}_p\boldsymbol{e}_a^* = \widehat{\boldsymbol{M}}_a^{-1}(\Delta\boldsymbol{\tau}_a - \boldsymbol{\tau}_{cp} - \boldsymbol{\tau}_{er}) \tag{38}$$

We define

$$\rho = \widehat{\boldsymbol{M}}_a^{-1}\Delta\boldsymbol{\tau}_a \tag{39}$$

and then the compensative control law $\boldsymbol{\tau}_{cp}$ is defined as follows

$$\boldsymbol{\tau}_{cp} = \widehat{\boldsymbol{M}}_a\widehat{\rho} \tag{40}$$

in which $\widehat{\rho}$ is the vector of a bank of RBFNNs for online approximation of $\Delta\boldsymbol{\tau}_a$. The structure of each RBFNN used in this study is shown in Figure 4. It is a fixed three-layer network whose output is determined by specified hidden units in certain local receptive fields.[37–39]

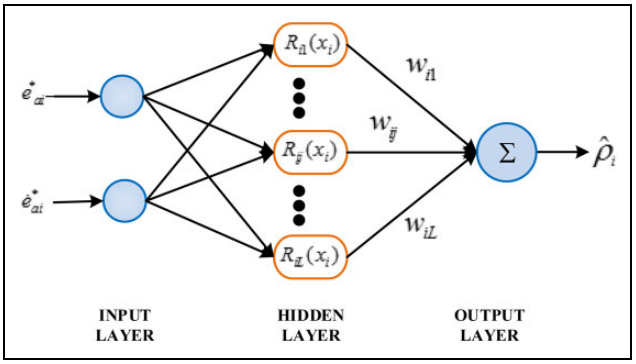

Figure 4. Structure of an RBFNN with L hidden units. RBFNN: radial basis function neural network.

The basis functions in the hidden layer are chosen as Gaussian function and can be expressed as follows

$$R_{ij}(\boldsymbol{x}_i) = \exp\left(-\frac{\| \boldsymbol{x}_i - \boldsymbol{c}_{ij}\|^2}{2m_{ij}^2}\right),\ j = 1, 2, \ldots, L;\ i = 1, 2, 3 \tag{41}$$

where $\boldsymbol{x}$ is an N-dimensional input vector, $\boldsymbol{c}_{ij}$ is a center vector with the same dimension as $\boldsymbol{x}$, m_{ij} is the variance of the basic function, L is the number of hidden units in the hidden layer, and $||.||$ denotes the Euclidean norm.

The output of the RBFNN is computed by the weighted sum method[40,41]

$$\widehat{\rho}_i = \sum_{j=1}^{L} \widehat{w}_{ij} R_{ij}(\boldsymbol{x}_i) = \widehat{\boldsymbol{W}}_i^T \boldsymbol{R}_i(\boldsymbol{x}_i) \tag{42}$$

$$\widehat{\boldsymbol{W}}_i = [\widehat{w}_{i1}, \widehat{w}_{i2}, \ \ldots, \widehat{w}_{iL}]^T,\ \ i = 1, 2, 3 \tag{43}$$

$$\boldsymbol{R}_i(\boldsymbol{x}_i) = [R_{i1}(\boldsymbol{x}_i), R_{i2}(\boldsymbol{x}_i), \ \ldots, R_{iL}(\boldsymbol{x}_i)]^T,\ \ i = 1, 2, 3 \tag{44}$$

The input vector $\boldsymbol{x}_i$ of each RBFNN is used as follows

$$\boldsymbol{x}_i = [e_{ai}^*, \dot{e}_{ai}^*]^T,\ \ i = 1, 2, 3 \tag{45}$$

where e_{ai}^* and $\dot{e}_{ai}^*$ are, respectively, the tracking cross-coupling error and derivative of cross-coupling error of each active joint of the parallel manipulator.

Then, the vector $\widehat{\rho} \in \Re^3$ can be written as follows

$$\widehat{\rho} = \begin{bmatrix} \widehat{\rho}_1 \\ \widehat{\rho}_2 \\ \widehat{\rho}_3 \end{bmatrix} = \begin{bmatrix} \widehat{\boldsymbol{W}}_1^T \boldsymbol{R}_1(\boldsymbol{x}_1) \\ \widehat{\boldsymbol{W}}_2^T \boldsymbol{R}_2(\boldsymbol{x}_2) \\ \widehat{\boldsymbol{W}}_3^T \boldsymbol{R}_3(\boldsymbol{x}_3) \end{bmatrix} \tag{46}$$

It has been shown that for some sufficiently large number of hidden layer neurons, there are ideal weights and thresholds such that ρ can be represented by the following

$$\rho = \begin{bmatrix} \boldsymbol{W}_1^{*T} \boldsymbol{R}_1(\boldsymbol{x}_1) \\ \boldsymbol{W}_2^{*T} \boldsymbol{R}_2(\boldsymbol{x}_2) \\ \boldsymbol{W}_3^{*T} \boldsymbol{R}_3(\boldsymbol{x}_3) \end{bmatrix} + \begin{bmatrix} \vartheta_1 \\ \vartheta_2 \\ \vartheta_3 \end{bmatrix} \tag{47}$$

where $\boldsymbol{W}_i^*$ is the bounded ideal weight vector of the output layer, and ϑ_i is the bounded error ($i = 1, 2, 3$).

Theorem. Consider the dynamic model in equation (29) of the 3-DOF planar parallel manipulator together with the proposed controller in equation (37). If we choose the following adaptation tuning laws for the RBFNN and the error estimators

$$\dot{\widehat{\boldsymbol{W}}}_i = \mu_i \boldsymbol{X}_i^T \boldsymbol{P}_i \boldsymbol{B}_i \boldsymbol{R}_i(\boldsymbol{x}_i) \tag{48}$$

$$\dot{\widehat{\vartheta}}_i = \eta_i \boldsymbol{X}_i^T \boldsymbol{P}_i \boldsymbol{B}_i \tag{49}$$

then the closed-loop system is asymptotically stable.

Proof. By substituting equations (46), (47), (40), and (39) into equation (38), we obtain the following

$$\ddot{\boldsymbol{e}}_a^* + \boldsymbol{K}_v \dot{\boldsymbol{e}}_a^* + \boldsymbol{K}_p \boldsymbol{e}_a^* = \begin{bmatrix} \widetilde{\boldsymbol{W}}_1^T \boldsymbol{R}_1(\boldsymbol{x}_1) \\ \widetilde{\boldsymbol{W}}_2^T \boldsymbol{R}_2(\boldsymbol{x}_2) \\ \widetilde{\boldsymbol{W}}_3^T \boldsymbol{R}_3(\boldsymbol{x}_3) \end{bmatrix} + \vartheta - \widehat{\boldsymbol{M}}_a^{-1} \boldsymbol{\tau}_{er} \tag{50}$$

where $\widetilde{\boldsymbol{W}}_i = \boldsymbol{W}_i^* - \widehat{\boldsymbol{W}}$ is the weight estimation error and $\boldsymbol{\vartheta} = [\vartheta_1, \vartheta_2, \vartheta_3]^T$ is the vector of errors.

Next, we define the error estimator as follows

$$\boldsymbol{\tau}_{er} = \widehat{\boldsymbol{M}}_a \widehat{\vartheta} \tag{51}$$

where $\widehat{\boldsymbol{\vartheta}}$ is the estimated value of $\boldsymbol{\vartheta}$.

Now, by substituting equation (51) into equation (50), we obtain the following

$$\ddot{\boldsymbol{e}}_a^* + \boldsymbol{K}_v \dot{\boldsymbol{e}}_a^* + \boldsymbol{K}_p \boldsymbol{e}_a^* = \begin{bmatrix} \widetilde{\boldsymbol{W}}_1^T \boldsymbol{R}_1(\boldsymbol{x}_1) \\ \widetilde{\boldsymbol{W}}_2^T \boldsymbol{R}_2(\boldsymbol{x}_2) \\ \widetilde{\boldsymbol{W}}_3^T \boldsymbol{R}_3(\boldsymbol{x}_3) \end{bmatrix} + \widetilde{\boldsymbol{\vartheta}} \tag{52}$$

where $\widetilde{\boldsymbol{\vartheta}} = \boldsymbol{\vartheta} - \widehat{\boldsymbol{\vartheta}}$.

Equation (52) is equivalent to the following equation

$$\begin{cases} \ddot{e}_{a1}^* + k_{v1} \dot{e}_{a1}^* + k_{p1} e_{a1}^* = \widetilde{\boldsymbol{W}}_1^T \boldsymbol{R}_1(\boldsymbol{x}_1) + \widetilde{\vartheta}_1 \\ \ddot{e}_{a2}^* + k_{v2} \dot{e}_{a2}^* + k_{p2} e_{a2}^* = \widetilde{\boldsymbol{W}}_2^T \boldsymbol{R}_2(\boldsymbol{x}_2) + \widetilde{\vartheta}_2 \\ \ddot{e}_{a3}^* + k_{v3} \dot{e}_{a3}^* + k_{p3} e_{a3}^* = \widetilde{\boldsymbol{W}}_3^T \boldsymbol{R}_3(\boldsymbol{x}_3) + \widetilde{\vartheta}_3 \end{cases} \tag{53}$$

For each active joint, supposed that state vector is defined as $\boldsymbol{X}_i = [e_{ai}^*, \dot{e}_{ai}^*]^T,\ i = 1, 2, 3$, we obtain the following state equations

$$\dot{\boldsymbol{X}}_i = \boldsymbol{A}_i \boldsymbol{X}_i + \boldsymbol{B}_i \left(\widetilde{\boldsymbol{W}}_i^T \boldsymbol{R}_i(\boldsymbol{x}_i) + \widetilde{\vartheta}_i\right),\ \ i = 1, 2, 3 \tag{54}$$

in which

$$\boldsymbol{A}_i = \begin{bmatrix} 0 & 1 \\ -k_{pi} & -k_{vi} \end{bmatrix};\ \boldsymbol{B}_i = \begin{bmatrix} 0 \\ 1 \end{bmatrix}$$

Then, the Lyapunov equation for each active joint can be formulated as follows

$$\boldsymbol{A}_i^T \boldsymbol{P}_i + \boldsymbol{P}_i \boldsymbol{A}_i = -\boldsymbol{Q}_i \tag{55}$$

where $\boldsymbol{P}_i$ and $\boldsymbol{Q}_i$ are positive definite matrices ($i = 1, 2, 3$). We can choose a positive definite matrix $\boldsymbol{Q}_i$, and then the positive definite matrix $\boldsymbol{P}_i$ is solved from the Lyapunov equation (55).

For analyzing the stability of the control system, a positive definite Lyapunov function candidate is chosen as follows

$$V = \sum_{i=1}^{3} \boldsymbol{X}_i^T \boldsymbol{P}_i \boldsymbol{X}_i + \frac{1}{2}\sum_{i=1}^{3} \mu_i^{-1} \tilde{\boldsymbol{W}}_i^T \tilde{\boldsymbol{W}}_i + \frac{1}{2}\sum_{i=1}^{3} \eta_i^{-1} \tilde{\vartheta}_i^2 \tag{56}$$

where $\tilde{\vartheta}_i = \vartheta_i - \hat{\vartheta}_i$, and μ_i and η_i are the learning rates with positive values ($i = 1, 2, 3$).

Differentiating V respect to time, we get the following

$$\dot{V} = \sum_{i=1}^{3} \left(\dot{\boldsymbol{X}}_i^T \boldsymbol{P}_i \boldsymbol{X}_i + \boldsymbol{X}_i^T \boldsymbol{P}_i \dot{\boldsymbol{X}}_i \right) - \sum_{i=1}^{3} \mu_i^{-1} \tilde{\boldsymbol{W}}_i^T \hat{\boldsymbol{W}}_i - \sum_{i=1}^{3} \eta_i^{-1} \tilde{\vartheta}_i \dot{\hat{\vartheta}}_i \tag{57}$$

By substituting equations (53) and (55) into equation (57), we obtain the following

$$\begin{aligned} \dot{V} = &- \sum_{i=1}^{3} \boldsymbol{X}_i^T \boldsymbol{Q}_i \boldsymbol{X}_i + \sum_{i=1}^{3} \boldsymbol{X}_i^T \boldsymbol{P}_i \boldsymbol{B}_i \tilde{\boldsymbol{W}}_i^T \boldsymbol{R}_i(\boldsymbol{x}_i) \\ &+ \sum_{i=1}^{3} \boldsymbol{X}_i^T \boldsymbol{P}_i \boldsymbol{B}_i \tilde{\vartheta}_i - \sum_{i=1}^{3} \mu_i^{-1} \tilde{\boldsymbol{W}}_i^T \hat{\boldsymbol{W}}_i - \sum_{i=1}^{3} \eta_i^{-1} \tilde{\vartheta}_i \dot{\hat{\vartheta}}_i \end{aligned} \tag{58}$$

Now, by replacing the adaptation tuning laws in equations (48) and (49) into equation (58), we obtain the following

$$\dot{V} = -\sum_{i=1}^{3} \boldsymbol{X}_i^T \boldsymbol{Q}_i \boldsymbol{X}_i \ \leq 0 \tag{59}$$

Since Q_i is a positive definite matrix, $\dot{V} = 0$ only when $\boldsymbol{X}_i = \boldsymbol{0}$, $i = 1, 2, 3$. Therefore, we can see that the control system is asymptotically stable with respect to $\boldsymbol{X}_i$. It means that

$$\lim_{t \to \infty} \boldsymbol{X}_i = 0,\ i \ = \ 1, 2, 3$$

Or equivalent to ($i = 1,2,3$)

$$\begin{cases} \lim\limits_{t \to \infty} e_i = 0 \Rightarrow \lim\limits_{t \to \infty} q_{ai} = q_{dai} \\ \lim\limits_{t \to \infty} \dot{e}_i = 0 \Rightarrow \lim\limits_{t \to \infty} \dot{q}_{ai} = \dot{q}_{dai} \end{cases}$$

Table 1. Parameters of the parallel manipulator.

Variable	Description	Value	Unit
l_1	Length of the lower part of each leg	0.4	m
l_2	Length of the upper part of each leg	0.6	m
l_3	Dimension of the moving platform	0.2	m
l_{c1}	Distance from the joint to the mass center of the lower part of each leg	0.2	m
l_{c2}	Distance from the joint to the mass center of the upper part of each leg	0.3	m
m_{i1}	Mass of the the lower part of each leg	5.12	kg
m_{i2}	Mass of the upper part of each leg	7.39	kg
m_p	Mass of the moving platform	3.84	kg
I_{i1}	Moment of inertia of the lower of *i*th leg	91×10^{-3}	kg·m^2
I_{i2}	Moment of inertia of the upper of *i*th leg	267×10^{-3}	kg·m^2
I_p	Moment of inertia of the moving platform	65×10^{-3}	kg·m^2

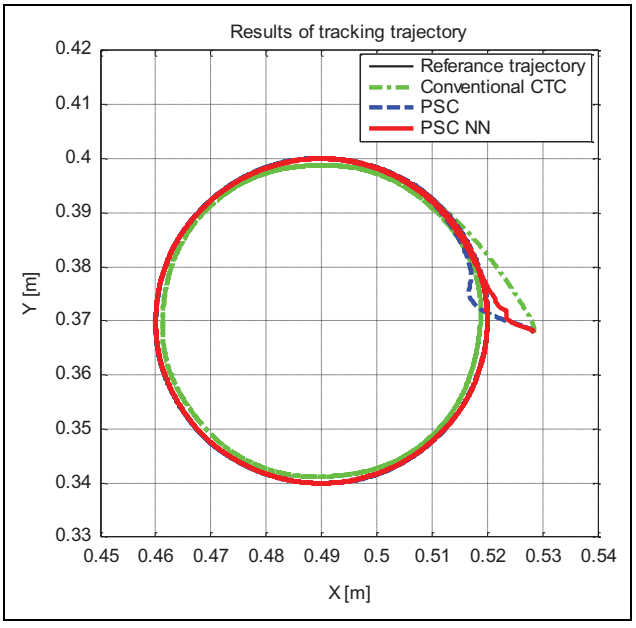

Figure 5. Results of tracking circular trajectory of conventional CTC (green line), SCTC (blue line), and the proposed ASCTC (red line). CTC: computed torque controller; SCTC: synchronized computed torque controller.

Thus, it is proved that, with the proposed controller, the actual active joint positions converge to their desired values.

Simulation

To illustrate the effectiveness of the proposed controller in this article, the simulations are performed for a 3-DOF

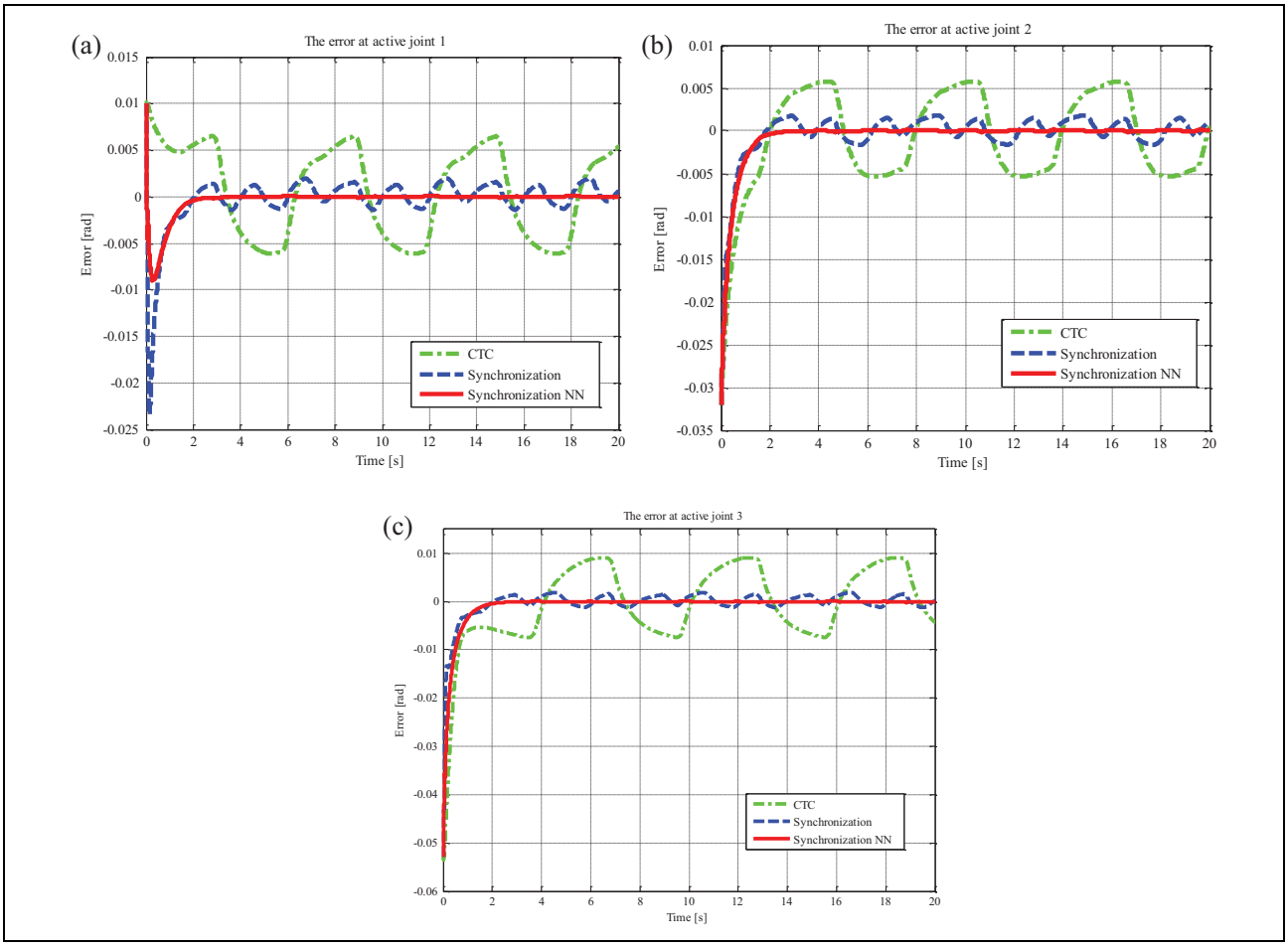

Figure 6. Comparison of tracking errors in active joint space: (a) error of active joint 1, (b) error of active joint 2, and (c) error of active joint 3.

planar parallel manipulator. Simulations are conducted by using the combination of Solidworks 2014 and SimMechanics of MATLAB [version 2013a]. First, the 3-D computer-aided design (CAD) model of the parallel manipulator is built in Solidworks. Each mechanical part of the parallel manipulator is designed separately and assembled using the suitable joints. Second, the 3-D CAD model is exported to an XML file by using the SimMechanics link plug-in. This link plug-in is downloaded from Mathworks official website. The XML file is then imported into Simulink environment. By this way, the geometry from the CAD assembly is saved as geometry files and associated with the proper body in SimMechanics. In the next step, joint actuators, sensors, and friction forces are added to the mechanical model. This mechanical model is then connected to the control algorithm block, which is written in Simulink.

In Solidworks, there is a special tool to verify the dynamic parameters of each component and the whole manipulator platform. The dynamic parameters are presented in Table 1.

To evaluate the effectiveness of the proposed adaptive SCTC, the following control algorithms for the 3-DOF planar parallel manipulator are simulated and compared:

1. The traditional computed torque controller for parallel manipulator was presented by Le et al.[10]

$$\boldsymbol{\tau}_a = \widehat{\boldsymbol{M}}_a(\ddot{\boldsymbol{q}}_{da} + \boldsymbol{K}_v\dot{\boldsymbol{e}}_a + \boldsymbol{K}_p\boldsymbol{e}_a) + \widehat{\boldsymbol{C}}_a\dot{\boldsymbol{q}}_a \tag{60}$$

where $\boldsymbol{K}_v$ and $\boldsymbol{K}_p$ are chosen as $\boldsymbol{K}_v = \text{diag}\{25, 25, 25\}$ and $\boldsymbol{K}_p = \text{diag}\{50, 50, 50\}$.

2. The SCTC without uncertainties and external disturbances compensation

$$\boldsymbol{\tau}_a = \widehat{\boldsymbol{M}}_a(\ddot{\boldsymbol{q}}^r_{da} + \boldsymbol{K}_v\dot{\boldsymbol{e}}^*_a + \boldsymbol{K}_p\boldsymbol{e}^*_a) + \widehat{\boldsymbol{C}}_a\dot{\boldsymbol{q}}_a \tag{61}$$

in which $\boldsymbol{K}_v$ and $\boldsymbol{K}_p$ are also chosen as $\boldsymbol{K}_v = \text{diag}\{25, 25, 25\}$ and $\boldsymbol{K}_p = \text{diag}\{50, 50, 50\}$.

3. The proposed control algorithm is described by equation (37). The matrices $\boldsymbol{K}_v$ and $\boldsymbol{K}_p$ have the same value to cases 1 and 2. Other parameters are chosen as $\sigma = 0.1$, the number of neuron in the hidden layer $L = 25$, and the center vector of

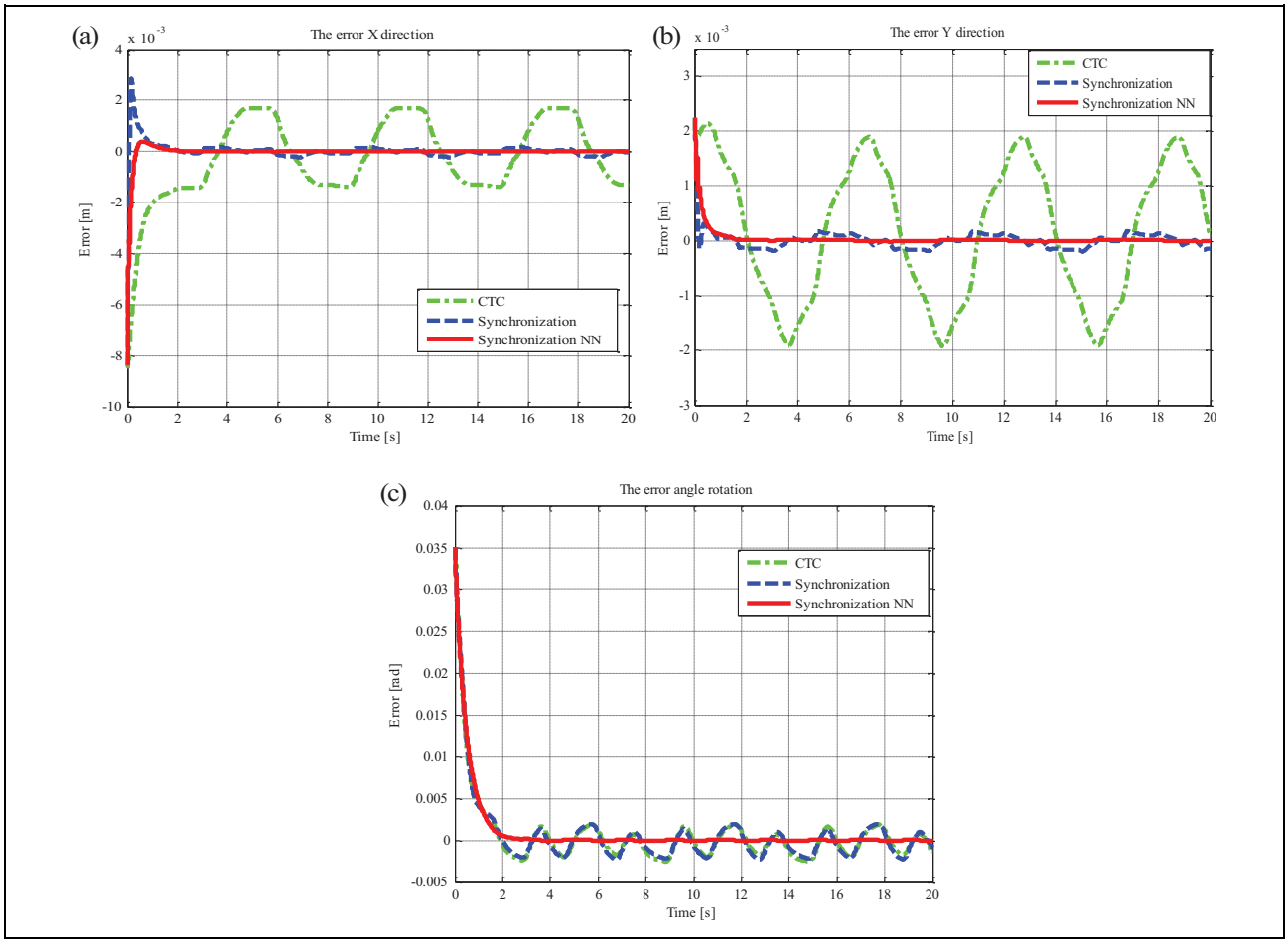

Figure 7. Comparison of tracking errors of the end-effector: (a) error in the *X*-direction, (b) error in the *Y*-direction, and (c) error of rotary angle.

each basis function neural network is $c_{jk} = [-0.5, -0.25, 0, 0.25, 0.5]^T (j = 1, 2, 3, 4, 5; k = 1, 2, 3, 4, 5)$. These values of parameters are obtained by trial and error method.

The desired trajectory of the end-effector of the 3-DOF planar parallel manipulator is

$$\begin{cases} x = 0.49 + 0.03 \cos\left(\dfrac{\pi t}{3}\right) \\ y = 0.37 + 0.03 \sin\left(\dfrac{\pi t}{3}\right) \\ \phi = \pi/2 \end{cases} \quad (62)$$

Friction forces at each active joint of the parallel manipulator are included as follows

$$F_{ai} = 0.5 \text{ sign}(\dot{q}_{ai}) + 0.4\dot{q}_{ai} \, i = \quad 1, 2, 3 \quad (63)$$

The results of tracking trajectory are shown in Figure 5. The end-effector is controlled to track a circular in 20 s. The initial point of the end-effector of the manipulator is $A_0(0.528, 0.368)$. From Figure 5, we can see that the tracking trajectory caused by the conventional computed torque control algorithm has the biggest difference from the reference trajectory. The synchronized computed torque control algorithm (without uncertainties compensation) and the proposed adaptive SCTC produce better results than the conventional CTC. When focusing on the starting point A_0, it can be seen that the proposed SCTC has the fastest convergence speed to the reference trajectory among the three controllers.

Figure 6 shows the tracking errors of active joints 1, 2, and 3 of the parallel manipulator in active joint space. It is observed that the errors caused by the SCTC are much smaller than the errors associated with the conventional CTC. Especially the proposed adaptive synchronized computed torque controller (ASCTC) brings about the smallest tracking errors compared with the SCTC without uncertainty compensation and the conventional CTC. Additionally, it can be seen that the errors caused by the proposed ASCTC are very small, almost equal to zero.

The tracking errors of the end-effector in the *X*-direction, in the *Y*-direction, and the error of rotary angle are shown in Figure 7. As can be seen from the figure, the

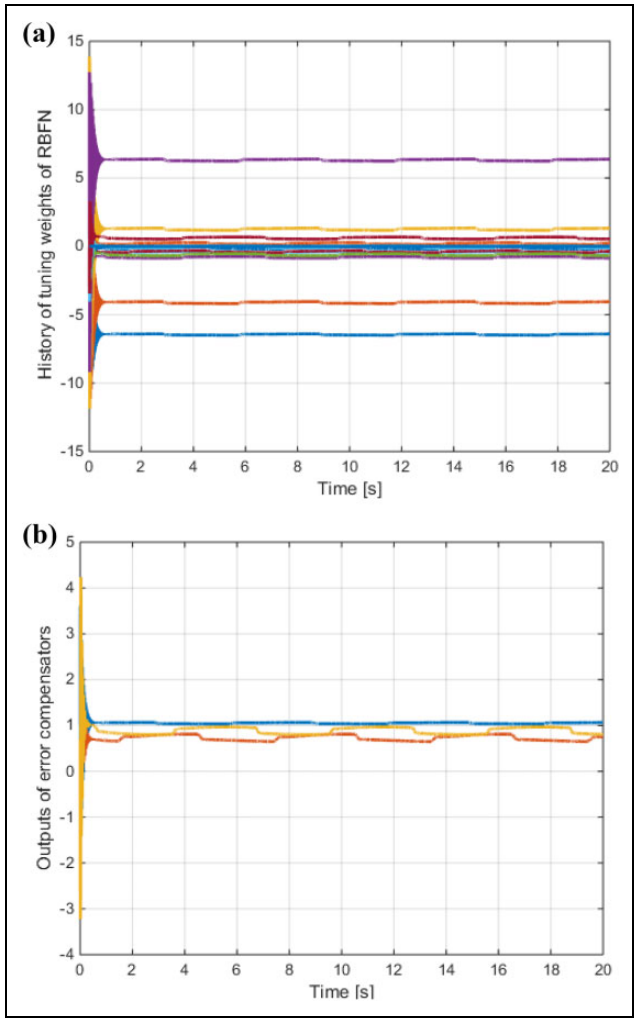

Figure 8. (a) The weight tuning histories of the radial basis function networks and (b) the results of online tuning, the output of error compensators.

SCTC produces smaller tracking errors compared to the conventional CTC. Interestingly, among the three controllers, the proposed ASCTC has the smallest tracking errors. Hence, we can conclude that the proposed ASCTC is highly efficient in control of a 3-DOF planar parallel manipulator. Additionally, it is concluded that the model uncertainties and external disturbances could be greatly tolerated using the proposed ASCTC.

The weight tuning histories of the radial basis function network in each active joint are shown in Figure 8(a). The initial output weight matrices are $\boldsymbol{W}_i(0) = 0.01 \times \boldsymbol{I}$. It can be observed that the weights remain bounded. Figure 8(b) shows the results of online tuning, the output of error compensators. The learning rate matrices are $\mu_i = 0.0015$ and $\eta_i = 0.01$ ($i = 1, 2, 3$). The results showed that the outputs of error compensators are also bounded.

From the simulations, we found that the major obstacle in the development of the control system with the proposed controller is the parameters of the RBFNN such as the centers and variances. It is quite difficult to choose the suitable values for these parameters. In future works, we are going to overcome this difficulty by applying the fully online tuning method for RBFNN, which was proposed by Le and Kang.[33]

Conclusion

An adaptive synchronized computed torque control algorithm based on neural networks and error compensators has been proposed in this article. By integrating the definitions of synchronization error, and cross-coupling error of active joints with an adaptive computed torque control algorithm, the results inherit the advantages of both methods, such as the high accuracy and low online computational burden. The proposed control algorithm handles the uncertainties and external disturbances using a bank of neural networks and error compensators. The weights of neural networks and error compensators are adaptively tuned online during the tracking trajectory of the parallel manipulator. The stability of the closed-loop control system is theoretically proved by the Lyapunov method. The results of computer simulations verified the effectiveness of the proposed control algorithm. In the future research direction, a real implementation on the experimental system will be conducted.

Declaration of conflicting interests

The author(s) declared no potential conflicts of interest with respect to the research, authorship, and/or publication of this article.

Funding

The author(s) disclosed receipt of the following financial support for the research, authorship, and/or publication of this article: This research is funded by the Ministry of Education and Training (MOET) of the Socialist Republic of Vietnam, under grant number KYTH-17.

References

1. Ghorbel FH, Chetelat O, Gunawardana R, et al. Modeling and set point control of closed-chain mechanisms: theory and experiment. *IEEE Trans Control Syst Technol* 2000; 8: 801–815.
2. Yang C, Huang Q, Jiang H, et al. PD control with gravity compensation for hydraulic 6-DOF parallel manipulator. *Mech Mach Theor* 2010; 45: 666–677.
3. Ouyang PR, Zhang WJ, and Wu FX. Nonlinear PD control for trajectory tracking with consideration of the design for control methodology. In: *Proceedings. ICRA'02. IEEE international conference on robotics and automation*, Washington, DC, USA, 11–15 May 2002, Vol. 4, pp. 4126–4131. IEEE.
4. Su YX, Dong S, Lu R, et al. Nonlinear PD synchronized control for parallel manipulators. In: *ICRA. Proceedings of the 2005 IEEE international conference on robotics and automation*, Barcelona, Spain, 18–22 April 2005, pp. 1374–1379. IEEE.

5. Ouyang PR, Zhang WJ, and Gupta MM. An adaptive switching learning control method for trajectory tracking of robot manipulators. *Mechatronics* 2006; 16: 51–61.
6. Yang Z, Wu J, and Mei J. Motor-mechanism dynamic model based neural network optimized computed torque control of a high speed parallel manipulator. *Mechatronics* 2007; 17: 381–390.
7. Paccot F, Lemoine P, Andreff N, et al. A vision-based computed torque control for parallel kinematic machines. In: *ICRA. IEEE international conference on robotics and automation*, Pasadena, CA, USA, 19–23 May 2008, pp. 1556–1561. IEEE.
8. Shang W and Cong S. Nonlinear computed torque control for a high-speed planar parallel manipulator. *Mechatronics* 2009; 19: 987–992.
9. Yang Z, Wu J, Mei J, et al. Mechatronic model based computed torque control of a parallel manipulator. *Int J Adv Robotic Syst* 2008; 5(1): 123–128.
10. Le TD, Kang HJ, Suh YS, et al. An online self-gain tuning method using neural networks for nonlinear PD computed torque controller of a 2-dof parallel manipulator. *Neurocomputing* 2012; 116: 53–61.
11. Abdellatif H and Heimann B. Advanced model-based control of a 6-DOF hexapod robot: a case study. *IEEE/ASME Trans Mech* 2010; 15: 269–279.
12. Kim HS, Cho YM, and Lee KII. Robust nonlinear task space control for 6 DOF parallel manipulator. *Automatica* 2005; 41(9): 1591–1600.
13. Guo H, Liu Y, Liu G, et al. Cascade control of a hydraulically driven 6-DOF parallel robot manipulator based on a sliding mode. *Control Eng Pract* 2008; 16: 1055–1068.
14. Begon P, Pierrot F, and Dauchez P. Fuzzy sliding mode control of a fast parallel robot: *Proceedings., IEEE international conference on robotics and automation*, Nagoya, Japan, 21–27 May 1995, Vol. 1, pp. 1178–1183. IEEE.
15. Le TD, Kang HJ, and Suh YS. Chattering-free neuro-sliding mode control of 2-DOF planar parallel manipulators. *Int J Adv Robotic Syst* 2013; 10(1): 22.
16. Qi Z, McInroy JE, and Jafari F. Trajectory tracking with parallel robots using low chattering, fuzzy sliding mode controller. *J Int Robotic Syst* 2007; 48: 333–356.
17. Ryu JH, Song J, and Kwon DS. A nonlinear friction compensation method using adaptive control and its practical application to an in-parallel actuated 6-DOF manipulator. *Control Eng Pract* 2001; 9: 159–167.
18. Zhu X, Tao G, Yao B, et al. Adaptive robust posture control of parallel manipulator driven by pneumatic muscles with redundancy. *IEEE/ASME Trans Mech* 2008; 13: 441–450.
19. Zhu X, Tao G, Yao B, et al. Adaptive robust posture control of a parallel manipulator driven by pneumatic muscles. *Automatica* 2008; 44: 2248–2257.
20. Honegger M, Brega R, and Schweiter G. Application of a nonlinear adaptive controller to a 6 DOF parallel manipulator. In: *Proceedings. ICRA. IEEE international conference on robotics and automation*, San Francisco, CA, USA, 24–28 April 2000, Vol. 2, pp. 1930–1935. IEEE.
21. Zhu X, Tao G, Yao B, et al. Integrated direct/indirect adaptive robust posture trajectory tracking control of a parallel manipulator driven by pneumatic muscles. *IEEE Trans Control Syst Technol* 2009; 17: 576–588.
22. Koren Y. Cross-coupled biaxial computer control for manufacturing systems. *J Dyn Systs Meas Control Trans ASME* 1980; 102(4): 265–272.
23. Cheng MH, Li YJ, and Bakhoum EG. Controller synthesis of tracking and synchronization for multiaxis motion system. *IEEE Trans Control Syst Technol* 2014; 22(1): 378–386.
24. Li Y, Zheng Q, and Yang L. Design of robust sliding mode control with disturbance observer for multi-axis coordinated traveling system. *Comput Math Appl* 2012; 64(5): 759–765.
25. Al-Ayasrah O, Alukaidey T, Salman R, et al. Dual feed-back and feed-forward synchronized cross-coupled motion control for two-wheel mobile robot. In: *IEEE*, *International workshop on robotic sensors: robotic and sensor environments*, Ottawa, Ont, Canada, 30 September–1 October 2005, pp. 55–60. IEEE.
26. Feng L, Koren Y and Borenstein J. Cross-coupling motion controller for mobile robots. *IEEE Control Syst* 1993; 13(6): 35–43.
27. Sun D, Lu R, Mills JK, et al. Synchronous tracking control of parallel manipulators using cross-coupling approach. *Int J Robotics Res* 2006; 25(11): 1137–1147.
28. Ren L, Mills JK, and Sun D. Trajectory tracking control for a 3-DOF planar parallel manipulator using the convex synchronized control method. *IEEE Trans Control Syst Technol* 2008; 16(4): 613–623.
29. Su Y, Sun D, Ren L, et al. Integration of saturated PI synchronous control and PD feedback for control of parallel manipulators. *IEEE Trans Robotics* 2006; 22(1): 202–207.
30. Ren L, Mills JK, and Sun D. Adaptive synchronized control for a planar parallel manipulator: theory and experiments. *J Dyn Syst Meas Control* 2006; 128(4): 976–979.
31. Shang W, Cong S, Zhang Y, et al. Active joint synchronization control for a 2-DOF redundantly actuated parallel manipulator. *IEEE Trans Control Syst Technol* 2009; 17(2): 416–423.
32. Ren L, Mills JK, and Sun D. Experimental comparison of control approaches on trajectory tracking control of a 3-DOF parallel robot. *IEEE Trans Control Syst Technol* 2007; 15(5): 982–988.
33. Le TD and Kang HJ. An adaptive tracking controller for parallel robotic manipulators based on fully tuned radial basic function networks. *Neurocomputing* 2014; 137: 12–23.
34. Le QD, Kang HJ, and Le TD. An adaptive position synchronization controller using orthogonal neural network for 3-DOF planar parallel manipulators. In: *International conference on intelligent computing*, Liverpool, UK, 7–10 August 2017, pp. 3–14. Cham: Springer.
35. Nakamura Y and Yamane K. Dynamics computation of structure-varying kinematic chains and its application to human figures. *IEEE Trans Robotics Autom* 2000; 16(2): 124–134.

36. Cheng H, Yiu YK, and Li Z. Dynamics and control of redundantly actuated parallel manipulators. *IEEE/ASME Trans Mech* 2003; 8(4): 483–491.
37. Rubio JJ. Discrete time control based in neural networks for pendulums. *Appl Soft Comput* 2017. DOI: 10.1016/j.asoc.2017.04.056.
38. Pan Y, Liu Y, Xu B, et al. Hybrid feedback feedforward: an efficient design of adaptive neural network control. *Neural Netw* 2016; 76: 122–134.
39. Pan Y and Yu H. Biomimetic hybrid feedback feedforward neural-network learning control. *IEEE Trans Neural Netw Learn Syst* 2017; 28(6): 1481–1487.
40. Rubio JJ, Elias I, Cruz DR, et al. Uniform stable radial basis function neural network for the prediction in two mechatronic processes. *Neurocomputing* 2017; 227: 122–130.
41. Paramo-Carranza LA, Meda-Campaña JA, Rubio JJ, et al. Discrete-time Kalman filter for Takagi–Sugeno fuzzy models. *Evolv Syst* 2017; 8(3): 211–219.