Neurocomputing 283 (2018) 107–119

Contents lists available at ScienceDirect

Neurocomputing

journal homepage: www.elsevier.com/locate/neucom

Self-learning interval type-2 fuzzy neural network controllers for trajectory control of a Delta parallel robot

Xingguo Lu[a], Yue Zhao[b,*], Ming Liu[a]

[a] State Key Laboratory of Robotics and System, Harbin Institute of Technology, Harbin 150001, PR China
[b] Control theory and Engineering, School of Astronautics, Harbin Institute of Technology, Harbin 150001, PR China

ARTICLE INFO

Article history:
Received 6 June 2017
Revised 31 October 2017
Accepted 15 December 2017
Available online 27 December 2017

Communicated by A.M. Alimi

Keywords:
Interval type-2 fuzzy logic systems
Neural network
Intelligent control
Delta parallel robot
Self-learning algorithm

ABSTRACT

This paper presents a self-learning interval type-2 fuzzy neural network (SLIT2FNN) control scheme for trajectory tracking problem of a Delta robot. This intelligent control scheme is computationally efficient and can be easily applied to existing equipment. The controller has a parallel structure, which combines an interval type-2 fuzzy neural network (IT2FNN) controller and a traditional proportional-derivative (PD) controller. We use the PD controller to compensate the transient performance, and use the IT2FNN to learn the dynamic characteristics of the system. A novel arrangement of trapezoidal interval type-2 fuzzy membership functions (IT2MF) is proposed, the arrangement enables the adaptation laws to have an analytical form. A learning algorithm based on sliding mode control (SMC) theory is proposed for the parameter training of the IT2FNN system. The control algorithm learns from the feedback error online and tunes the parameters of the IT2FNN, and will become the main source of the control signal after several learning iterations. Unlike model-based control, this control method has no requirement of prior information and constraint conditions from the robot plant. Lyapunov stability method is employed to prove asymptotic stability of the proposed approach. The structure of the SLIT2FNN and the operations in each layer are introduced in detail. The performance and robustness of the proposed controller is demonstrated on the Delta robot trajectory tracking problems in the presence of structured and unstructured uncertainties. Simulation results illustrate that the proposed SLIT2FNN control approach produces higher trajectory tracking accuracy and more robust to uncertainties as compared to its counterparts.

1. Introduction

Because the driving motors of parallel robots are installed on the base platform, the inertia of the moving parts is reduced. Parallel robots can obtain not only higher speed and acceleration than their serial counterparts but also higher structure stiffness and payload capacity. All these advantages make parallel robots widely used in many industry applications [1–4]. One of the most encountered problems when using parallel robots in engineering applications is the trajectory tracking control problem of the parallel manipulators [5]. However, before the desired control performance is obtained, there are many challenges needed to be addressed. The nonlinearities of the parallel robots, structured and unstructured dynamic uncertainties and various operating conditions are among the typical challenges to be faced.

A lot of research has been done with the tracking control problem of parallel robot in the recent decade. Some of the works were based on traditional control method: [6,7] were based on feedback linearization, [8–10] on adaptive theory, and [11–14] on variable structure system theory. However, such type of control algorithms could provide a satisfactory performance should base on the assumption that the dynamic model of the plant is perfectly known. But to obtain the perfect mathematical dynamic model of a parallel robot is very difficult and sometimes impossible. Moreover, even if the explicit parameters of the uncertain dynamic exist, the dynamic model of the robot is very hard to derive especially for the parallel robot with complex kinematic structure.

On another aspect, computational intelligence control methods, such as fuzzy logic systems (FLS) and artificial neural networks (ANN), have been used in the trajectory tracking control problems recently. Intelligent controller can learn the dynamics of the robots with structured and unstructured uncertainties online, and tune the parameters and structure of the controller according to the operating dynamics and conditions without requiring the mathematical model of the control plant. Therefore, the overall optimal control performance can be obtained. With the current trend of

* Corresponding author.
E-mail addresses: luxingguo@hit.edu.cn (X. Lu), yue.zhao@hit.edu.cn, zhaoyuelj@163.com (Y. Zhao), liuming@hit.edu.cn (M. Liu).

https://doi.org/10.1016/j.neucom.2017.12.043
0925-2312/

"Industry 4.0", more and more intelligent controllers will be applied to robots to meet the requirement of increasing control performance and adaptability. However, most of the robots currently used in industry and laboratories are with non-intelligent controllers. For what concerns applications, it is surely desirable to replace the existing controller with intelligent controller while involving little changes and not too much investments. Within this scenario, the motivation of this paper is to present an intelligent controller for trajectory control of a parallel Delta robot, and the controller can learn the robot dynamics online and adjust its operational parameters according to the working conditions. Also, the controller should be easy to apply to the existing equipment. In this study, a SLIT2FNN controller with a SMC theory based learning algorithm is proposed. We design the controller using a parallel structure, which combines a conventional PD controller and an intelligent controller (SLIT2FNN). We use the PD controller to compensate the transient performance of the intelligent control system. During the learning procedure of the SLIT2FNN, the control signal is mainly comes from the PD controller. After several learning iterations, the SLIT2FNN controller takes over the control, and become the main source of the control signal. Thus the uncertainties and nonlinearities of the robot system are handled by the SLIT2FNN, and the applicability is ensured by the parallel structure.

When the fuzzy logic controller (FLC) is applied to the nonlinear control systems, the performance is quit robust. This is due to the FLC is designed based on human experience and knowledge, without requiring of the precise plant mathematical model and can directly deal with the uncertainties of the systems [15–19]. As a kind of FLC, type-1 fuzzy logic controllers (T1FLC) have been applied to many engineering fields successfully [20–23]. However, the control performance of the T1FLC will decrease when higher degrees of uncertainties are encountered [24,25]. Type-2 fuzzy logic controllers (T2FLC) have been credited to be more powerful to cope with system variations and higher degrees of uncertainties [26–28]. There is at least one type-2 fuzzy set (T2FS) in a typical T2FLC, and the T2FS has a primary membership and a corresponding secondary grade. The increasing of the fuzzy set dimensions makes it has more design degrees of freedom to handle uncertainty information directly. Due to the type-reduction procedure, the general T2FLCs are computationally intensive [29–31], so the interval type-2 fuzzy logic controllers (IT2FLC) are more commonly adopted in researches [32–36].

ANN is another popular computational intelligence approach, which has an inherent learning ability and theoretically can uniformly approximating a nonlinear function to any degree of accuracy [37]. ANNs are used in robot control systems to model the dynamic process or to compensate for the uncertainties without requiring the information of the robot systems [38–41]. Despite the success applications of the ANN based control systems, there are three main problems have to be faced when applying ANNs to robot system control: I) the learning process degrades the transient response; II) the internal structure can't be explained in human language (which often be treated as a black box); III) incapable of using the already exist human expert experience. FNNs are the combination of FLS and ANN, and have the advantages of both systems (reasoning ability and learning ability). So, in this work, by using FNN, we introduce the learning and representation capability of ANNs and the seasoning ability of the FLSs into one system. Besides, interval type-2 fuzzy sets (IT2FS) are adopted for a trade-off between the computational burden and the ability of dealing with uncertainties.

Parameter tuning is an important feature of intelligence for FNNs. The adaptive, self-organizing and self-learning ability is achieved by the tuning algorithm of a FNN. There are two kinds of tuning method widely used for FNN: gradient descent based method and evolutionary algorithm based method. For the gradient descent method, partial derivative calculation is needed to determine the value of the FNN parameters. So the learning speed of this method is slow, especially when the searching space is complex. Also, because of the partial derivative calculation, it is very difficult to obtain an analytical expression of the learning algorithm for convergence and stability analysis [42]. For the evolutionary algorithm based method, random search calculations are required when the parameter tuning of a FNN is performed. Since it is a stochastic algorithm, the stability is questionable and to obtain optimized parameters usually need to repeat the computations for several times. In addition, the computational burden is much higher than gradient descent based method [43,44]. Considering the drawbacks of these two methods, an alternative parameter tuning method with higher computational efficiency is needed in this work.

SMC techniques have been widely used in the robust control of nonlinear systems [45–49]. Using the robust character of the SMC techniques, parameter learning rules for ANNs based on SMC theory has been studied by many researchers [50–52]. The SMC theory based parameter tuning method has the advantages of robustness and high computational efficiency as compared with gradient descent and evolutionary algorithm based parameter tuning methods. Moreover, the parameter learning rules based on SMC theory make it possible to analyze the stability of the system by using the well-established mathematical theories. So the SMC theory based parameter tuning method for FNN is considered in this paper. But there are still some disadvantages in SMC, one of the most occurs is the chattering phenomenon when the system dynamics is close to the sliding surface. In order to eliminate this problem we add a boundary layer in the control system. When control signal is in this layer, the original control signal will be replaced by an equivalent continuous function.

The major contributions of this paper are as follows. The first is the proposal of a SLIT2FNN controller. The controller has a parallel structure including a traditional PD controller and an intelligent controller, and this kind of structure makes it possible to apply intelligent controllers to the existing equipment with little changes and not too much investments. Second, trapezoidal IT2MFs are adopted in the design of the FNN system. Their advantages over the other kinds of MFs are mentioned in [53]. Accordingly, we proposed a novel arrangement of the trapezoidal IT2MFs. This kind of MF arrangement is easy to design and enables the parameter adaptation laws to have an analytical form. To the best of the authors' knowledge, this is the first time such kind of trapezoidal IT2MFs arrangement is proposed. A parameter learning algorithm based on SMC theory is proposed to adjust the parameters of the IT2FNN, and the stability of the parallel intelligent control scheme is proved. This approach is computational efficient and robust and has the potential to be applied to real-time systems. Finally, the benefits of the proposed SLIT2FNN control system are tested on the trajectory tracking problem of a Delta parallel robot in the presence of various nonlinearities and uncertainties.

The rest of this paper is organized as follows. The Dynamic model of the Delta robot is provided in Section 2. Section 3 presents the design of the IT2FNN. The SMC theory based learning laws and the stability analysis of the proposed SLIT2FNN control scheme is addressed in Section 4. Simulation results are shown and discussed in Section 5. The conclusions are drawn in Section 6.

2. Dynamics of the Delta parallel robot

The Delta parallel robot designed in Harbin Institute of Technology is illustrated in Fig. 1, which will be used to evaluate the effectiveness of the controllers. This robot is driven by the external revolute joint in the actuating arms and the passive arms are with

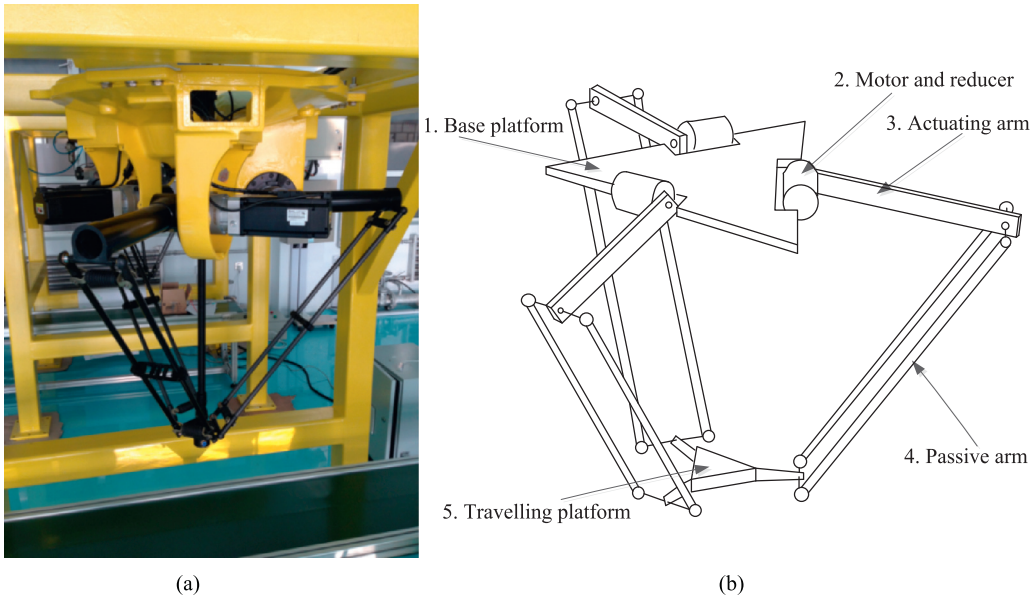


Fig. 1. (a) Delta robot. (b) Scheme of the Delta robot.

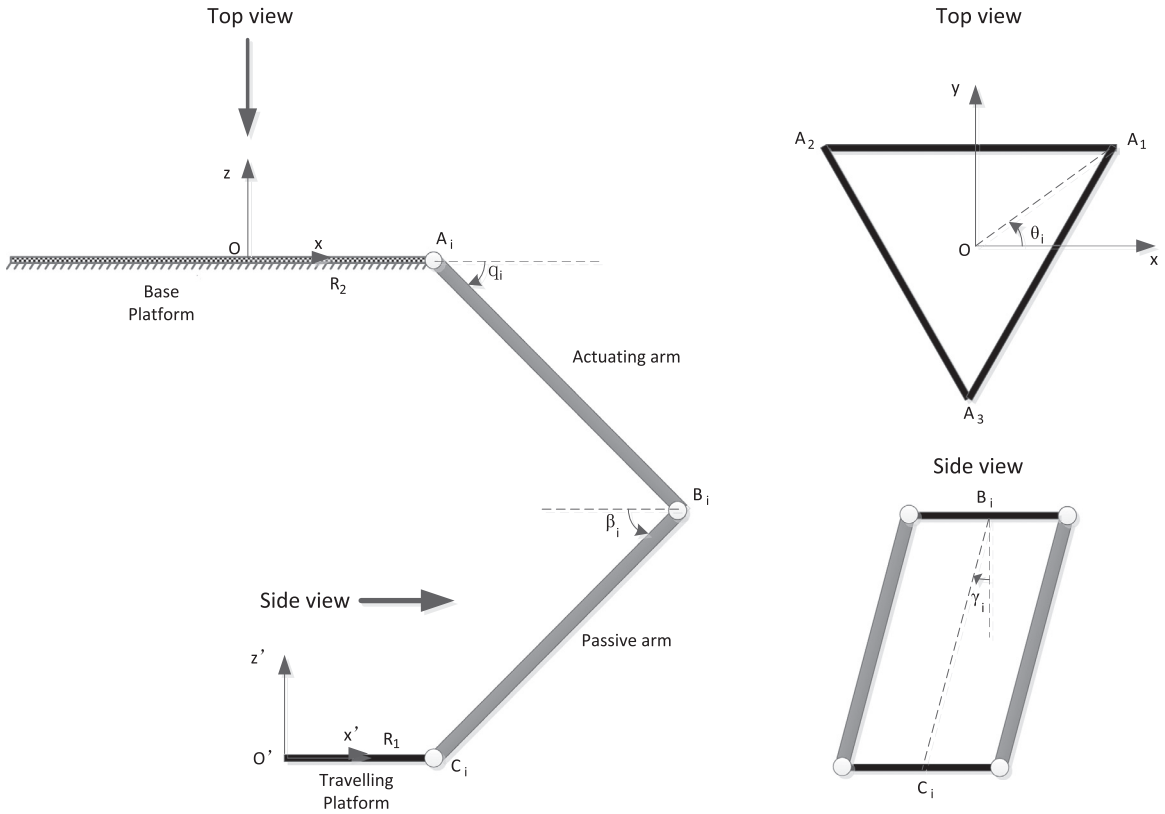


Fig. 2. Structure parameters of the robot.

the parallelogram structure, which ensures the high speed translational motion of the travelling platform. The motors are installed on the base platform to reduce the inertia of the moving parts, so this robot can obtain higher speed and acceleration than its serial counterparts.

The structure parameters are depicted in Fig. 2. Using the virtual work principle, the rigid body dynamic model can be expressed as:

$$\boldsymbol{\tau}_{m_n} + \sum_{i=1}^{3} \boldsymbol{\tau}_{L,i} + \sum_{i=1}^{3} \boldsymbol{\tau}_{M,i} = 0 \tag{1}$$

where τ_{m_n}, $\tau_{L,i}$ and $\tau_{M,i}$ are the generalized forces of the travelling platform, the actuating and passive arm i, respectively.

Eq. (1) can be further simplified as:

$$\boldsymbol{\tau} = \boldsymbol{I}_L\ddot{\boldsymbol{q}} + \boldsymbol{J}^T m_n \ddot{\boldsymbol{X}}_n - \boldsymbol{G}_L - \boldsymbol{J}^T \boldsymbol{G}_n + \sum_{i=1}^{3} \boldsymbol{\tau}_{M,i} \tag{2}$$

where $\boldsymbol{\tau}$ is the driving torque, $\boldsymbol{I}_L$ is the inertia of the actuating arms and the motors, $\boldsymbol{q}$ is the angle matrix of the driving arms, $\boldsymbol{J}$ is the Jacobian matrix, m_n is the sum of the mass of the payload and the travelling platform, $\boldsymbol{X}_n$ is the tool center point of the robot, $\boldsymbol{G}_L$ and $\boldsymbol{G}_n$ are the gravity forces of the active arms and the travelling

platforms, respectively. Ref. [54] gives a more detailed derivation of the dynamic model.

Eq. (2) can be rewritten as:

$$\boldsymbol{\tau} = \boldsymbol{A}\ddot{\boldsymbol{q}} + \boldsymbol{T}_{cc}\dot{\boldsymbol{q}} + \boldsymbol{T}_g + \sum_{i=1}^{3} \boldsymbol{\tau}_{M,i} \tag{3}$$

where $\boldsymbol{A} = \boldsymbol{I}_L + \boldsymbol{J}^T m_n \boldsymbol{J}$ is the inertia matrix, $\boldsymbol{T}_{cc} = \boldsymbol{J}^T m_n \dot{\boldsymbol{J}}$ is the Coriolis/centripetal matrix and $\boldsymbol{T}_g = -\boldsymbol{J}^T \boldsymbol{G}_n - \boldsymbol{G}_L$ is the gravity vector.

Considering the frictions at the actuating joints and the gear reducers, the dynamic equation of the Delta parallel robot can be written as follows:

$$\boldsymbol{\tau} = \boldsymbol{A}\ddot{\boldsymbol{q}} + \boldsymbol{T}_{cc}\dot{\boldsymbol{q}} + \boldsymbol{T}_g + \sum_{i=1}^{3} \boldsymbol{\tau}_{M,i} + F_v\dot{\boldsymbol{q}} + F_c \mathrm{sgn}(\dot{\boldsymbol{q}}) \tag{4}$$

where F_v is the viscous friction coefficient and F_c is the Coulomb friction.

Because the Jacobian matrix is determined by the position of the tool center point, from Eqs. (3) and (4), one can see that the inertia matrix $\boldsymbol{A}$, the Coriolis/centripetal matrix $\boldsymbol{T}_{cc}$ and the gravity matrix $\boldsymbol{T}_g$ are all varying with the position of the tool center point. The motion between the kinematic chains is interactional because of the dynamics coupling effect between the arms. Thus, even assuming there is no external disturbance, the Delta parallel robot is still a time-varying, coupling, and nonlinear with inherent uncertainty dynamic system.

3. Interval type-2 fuzzy neural network

3.1. Backgrounds of type-2 fuzzy sets

Here, we will discuss about the theoretic backgrounds of T2FSs and IT2FSs [24,55]. A typical T2FS denoted as $\tilde{A}$ can be represented by a type-2 membership function $\mu_{\tilde{A}}(x,u)$, $x \in X$ and $u \in J_x \subseteq [0, 1]$:

$$\tilde{A} = \int_{x\in X} \int_{u\in J_x} \mu_{\tilde{A}}(x,u)/(x,u) \tag{5}$$

where $0 \le \mu_{\tilde{A}}(x,u) \le 1$, $\iint$ denotes the union over all admissible x and u for continuous universes of discourse. For discrete universes of discourse, $\int$ is replaced by Σ.

An IT2FS $\tilde{A}$ can be expressed as:

$$\tilde{A} = \int_{x\in X} \int_{u\in J_x} 1/(x,u) \tag{6}$$

All the secondary membership grades of $\tilde{A}$ are set to 1, which makes a great computational simplification when calculating the type-reduced sets.

The Footprint-Of-Uncertainty (FOU) of the IT2FS is defined by the union of all primary memberships:

$$\mathrm{FOU}(\tilde{A}) = \bigcup_{x\in X} J_x \tag{7}$$

The FOU of an IT2FS is shown in Fig. 3. We can see that the FOU is bounded by two T1 fuzzy sets, which are called the upper membership function (UMF) denoted as $\bar{\mu}_{\tilde{A}}(x)$ and the lower membership function (LMF) denoted as $\underline{\mu}_{\tilde{A}}(x)$, i.e.:

$$\bar{\mu}_{\tilde{A}}(x) \equiv \overline{\mathrm{FOU}(\tilde{A})} \quad \forall x \in X \tag{8}$$

$$\underline{\mu}_{\tilde{A}}(x) \equiv \underline{\mathrm{FOU}(\tilde{A})} \quad \forall x \in X \tag{9}$$

Note that for a IT2FS $J_x = [\underline{\mu}_{\tilde{A}}, \bar{\mu}_{\tilde{A}}]$, $\forall x \in X$.

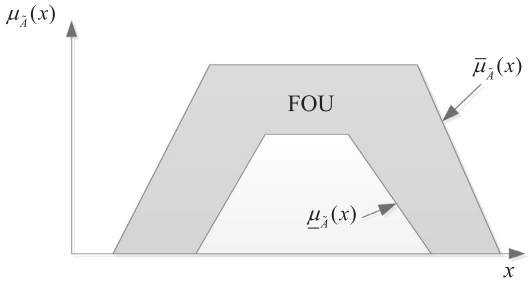


Fig. 3. FOU, LMF and UMF of IT2FS.

3.2. Interval type-2 fuzzy neural network structure

In general, one type of the most commonly used interval type-2 fuzzy logic systems has I inputs $x_1 \in X_1, \ldots, x_I \in X_I$, and one output $y \in Y$ with N rules, has the form (for the nth rule) [56]:

R^n: IF x_1 is $\tilde{A}_1^n$ and...and x_I is $\tilde{A}_I^n$, THEN $y^n = c_0^n + c_1^n x_1 + c_2^n x_2 + \cdots + c_I^n x_I$, $n = 1, \ldots, N$.

where $\tilde{A}_i^n$ $(i = 1, 2, \ldots, I)$ is the corresponding IT2MF for the input x_i, c_j^n $(j = 0, 1, \ldots, I)$ denotes the consequent parameters of the nth rule, y^n is the output of the nth rule.

In this paper, we choose the trapezoidal IT2MFs as the antecedent part of the IT2FNN and variable crisp numbers as the consequent part. This kind of structure is called IT2FNN-0 architecture, where the parameters $c_1^n, \ldots c_I^n$ are all equal to 0 [57,58]. In order to conveniently describe the seven-layered IT2FNN, we have made some adjustments to the original IT2FNN-0 architecture, and the structure is shown in Fig. 4. This network is composed of two kinds of nodes: adaptive nodes and fixed nodes. The adaptive nodes which are represented by squares have modifiable parameters while the fixed nodes represented by circles only perform predetermined math operations. Parameters in the adaptive nodes are determined by the learning algorithm, which will be discussed in detail in next section.

The detailed operations in each layer of the proposed seven-layered IT2FNN shown in Fig. 4 will be introduced hereinafter.

Layer 1 (input layer): There are two input signals feed the system in layer 1 including the joint error (x_1) and the time derivative of the joint error (x_2). The joint error is defined by the difference between the actual and the demand angle of the actuating arm. No mathematical operations will be performed in the nodes of this layer.

Layer 2 (fuzzification layer): The input signals are fuzzified in this layer, and the outputs of this layer are membership degrees $\underline{\mu}$ and $\bar{\mu}$. The antecedent part of the IT2FNN is composed of two series of trapezoidal IT2MFs which are symmetric with respect to the y-axis. The arrangement of these fuzzy sets is illustrated in Fig. 5. Advantages of this kind of trapezoidal IT2MF over other MFs are mentioned in [53]. The novel arrangement of the trapezoidal IT2MFs makes it possible to have a closed form parameter adaption laws. The mathematical expression for UMF and LMF can be written as:

$$\underline{\mu}_{1i} = \begin{cases} 1 & x_1 < -\underline{c}_{1i} - \underline{\alpha}_{1i} \\ \frac{\underline{c}_{1i} - \underline{\alpha}_{1i} - x_1}{2\underline{c}_{1i}} & -\underline{c}_{1i} - \underline{\alpha}_{1i} \le x_1 \le \underline{c}_{1i} - \underline{\alpha}_{1i} \\ 0 & x_1 > \underline{c}_{1i} - \underline{\alpha}_{1i} \end{cases} \tag{10}$$

when $i = 1, 3, 5 \ldots$

$$\underline{\mu}_{1i} = \begin{cases} 0 & x_1 < -\underline{c}_{1i} + \underline{\alpha}_{1i} \\ \frac{\underline{c}_{1i} - \underline{\alpha}_{1i} + x_1}{2\underline{c}_{1i}} & -\underline{c}_{1i} + \underline{\alpha}_{1i} \le x_1 \le \underline{c}_{1i} + \underline{\alpha}_{1i} \\ 1 & x_1 > \underline{c}_{1i} + \underline{\alpha}_{1i} \end{cases} \tag{11}$$

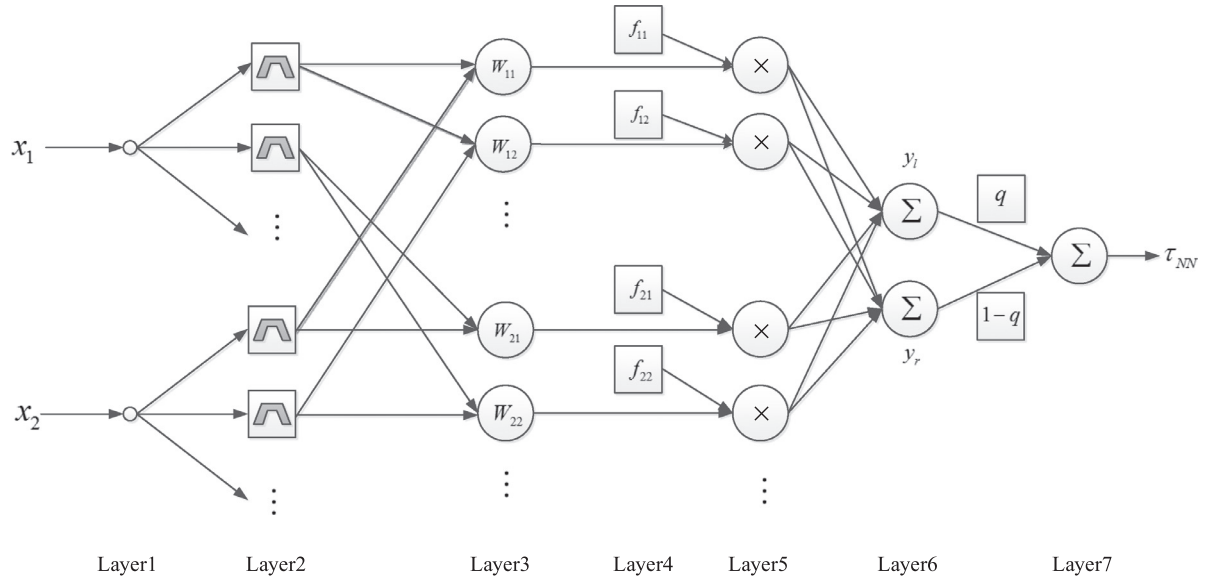


Fig. 4. Structure of the proposed seven-layered IT2FNN.

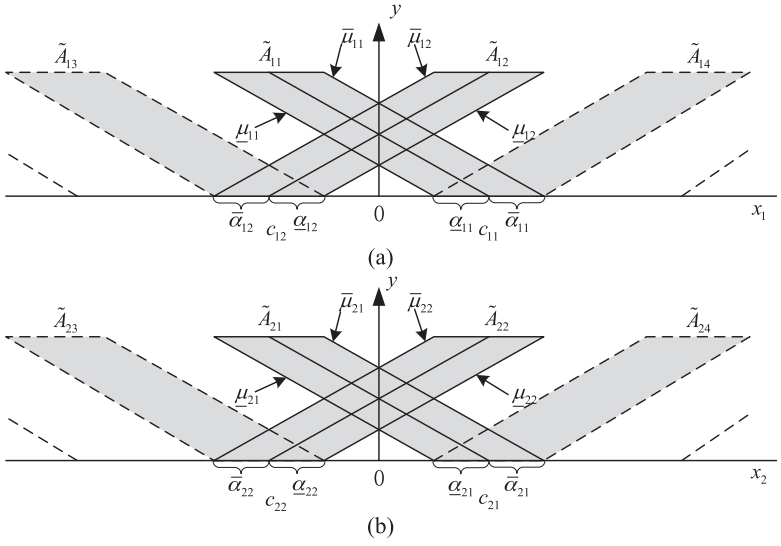


Fig. 5. The trapezoidal MFs and their arrangement used in the neural network.

when $i = 2, 4, 6\ldots$

$$\bar{\mu}_{1i} = \begin{cases} 1 & x_1 < -\bar{c}_{1i} + \bar{\alpha}_{1i} \\ \frac{\bar{c}_{1i}+\bar{\alpha}_{1i}-x_1}{2\bar{c}_{1i}} & -\bar{c}_{1i} + \bar{\alpha}_{1i} \le x_1 \le \bar{c}_{1i} + \bar{\alpha}_{1i} \\ 0 & x_1 > \bar{c}_{1i} + \bar{\alpha}_{1i} \end{cases} \tag{12}$$

when $i = 1, 3, 5\ldots$

$$\bar{\mu}_{1i} = \begin{cases} 0 & x_1 < -\bar{c}_{1i} - \bar{\alpha}_{1i} \\ \frac{\bar{c}_{1i}+\bar{\alpha}_{1i}+x_1}{2\bar{c}_{1i}} & -\bar{c}_{1i} - \bar{\alpha}_{1i} \le x_1 \le \bar{c}_{1i} - \bar{\alpha}_{1i} \\ 1 & x_1 > \bar{c}_{1i} - \bar{\alpha}_{1i} \end{cases} \tag{13}$$

when $i = 2, 4, 6\ldots$

$$\underline{\mu}_{2j} = \begin{cases} 1 & x_2 < -\underline{c}_{2j} - \underline{\alpha}_{2j} \\ \frac{\underline{c}_{2j}-\underline{\alpha}_{2j}-x_2}{2\underline{c}_{2j}} & -\underline{c}_{2j} - \underline{\alpha}_{2j} \le x_2 \le \underline{c}_{2j} - \underline{\alpha}_{2j} \\ 0 & x_2 > \underline{c}_{2j} - \underline{\alpha}_{2j} \end{cases} \tag{14}$$

when $i = 1, 3, 5\ldots$

$$\underline{\mu}_{2j} = \begin{cases} 0 & x_2 < -\underline{c}_{2j} + \underline{\alpha}_{2j} \\ \frac{\underline{c}_{2j}-\underline{\alpha}_{2j}+x_2}{2\underline{c}_{2j}} & -\underline{c}_{2j} + \underline{\alpha}_{2j} \le x_2 \le \underline{c}_{2j} + \underline{\alpha}_{2j} \\ 1 & x_2 > \underline{c}_{2j} + \underline{\alpha}_{2j} \end{cases} \tag{15}$$

when $i = 2, 4, 6\ldots$

$$\bar{\mu}_{2j} = \begin{cases} 1 & x_2 < -\bar{c}_{2j} + \bar{\alpha}_{2j} \\ \frac{\bar{c}_{2j}+\bar{\alpha}_{2j}-x_2}{2\bar{c}_{2j}} & -\bar{c}_{2j} + \bar{\alpha}_{2j} \le x_2 \le \bar{c}_{2j} + \bar{\alpha}_{2j} \\ 0 & x_2 > \bar{c}_{2j} + \bar{\alpha}_{2j} \end{cases} \tag{16}$$

when $i = 1, 3, 5\ldots$

$$\bar{\mu}_{2j} = \begin{cases} 0 & x_2 < -\bar{c}_{2j} - \bar{\alpha}_{2j} \\ \frac{\bar{c}_{2j}+\bar{\alpha}_{2j}+x_2}{2\bar{c}_{2j}} & -\bar{c}_{2j} - \bar{\alpha}_{2j} \le x_2 \le \bar{c}_{2j} - \bar{\alpha}_{2j} \\ 1 & x_2 > \bar{c}_{2j} - \bar{\alpha}_{2j} \end{cases} \tag{17}$$

when $i = 2, 4, 6\ldots$

where $\underline{\mu}_{1i}$ and $\bar{\mu}_{1i}$ are the LMF and UMF values of the input x_1, $\underline{\mu}_{2j}$ and $\bar{\mu}_{2j}$ are the LMF and UMF values of the input x_2, c_{1i} and

c_{2j} are the base point of the MFs for the inputs x_1 and x_2, a_{1i} and a_{2j} are the blurring degrees of the MFs for the inputs x_1 and x_2.The output of each node in this layer is an interval $[\underline{\mu}, \bar{\mu}]$.

Nodes in this layer are all adaptive nodes, which mean that, the parameters: $\underline{c}_{1i}$, $\bar{c}_{1i}$, $\underline{c}_{2j}$, $\bar{c}_{2j}$, $\underline{\alpha}_{1i}$, $\bar{\alpha}_{1i}$, $\underline{\alpha}_{2j}$, $\bar{\alpha}_{2j}$ are modifiable. The learning algorithm for these parameters will be discussed in the next section.

Layer 3 (firing layer): The firing strength W_{ij} is calculated in this layer. Each node uses the output values of the second layer to calculate the firing strength. The product operator is adopted in the fuzzy meet operations. The output of this layer W_{ij} is an interval:

$$W_{ij} = \left[\underline{W}_{ij}, \bar{W}_{ij}\right] \tag{18}$$

$$\underline{W}_{ij} = \underline{\mu}_{1i}(x_1)\underline{\mu}_{2j}(x_2) \tag{19}$$

$$\bar{W}_{ij} = \bar{\mu}_{1i}(x_1)\bar{\mu}_{2j}(x_2) \tag{20}$$

Nodes in this layer are fixed nodes, so only the predetermined math operations shown in Eqs. (19 and 20) are performed in these nodes.

Layer 4 (consequent layer): The consequent part of each fuzzy rule R^n is determined in this layer. For each output of layer 3, there is a corresponding consequent node in layer 4. Here, crisp numbers are used in the consequent parts of the proposed IT2FNN:

$$y^{ij} = f_{ij} \tag{21}$$

The values of f_{ij} are calculated by the SMC theory based learning laws. The functions will be given in the next section. Thus, all the nodes are adaptive nodes in this layer.

Layer 5 (product layer): This layer computes the product of the firing strength and the values of f_{ij} (obtained from layer 3 and layer 4). The math operations of the product layer are denoted by:

$$y_l^{ij} = f_{ij}\underline{W}_{ij} \tag{22}$$

$$y_r^{ij} = f_{ij}\bar{W}_{ij} \tag{23}$$

where y_l^{ij} and y_r^{ij} are the outputs of each node.

Nodes in this layer only perform the predetermined math operations, so all the nodes are fixed nodes.

Layer 6 (left-right point layer): The left and right end points of the inference engine used in this paper are calculated in this layer, which is given as:

$$y_l = \sum_{i=1}^{I}\sum_{j=1}^{J}\tilde{\underline{W}}_{ij}f_{ij} \tag{24}$$

$$y_r = \sum_{i=1}^{I}\sum_{j=1}^{J}\tilde{\bar{W}}_{ij}f_{ij} \tag{25}$$

where y_l and y_r are the left and right end point values, respectively. $\tilde{\underline{W}}_{ij}$ and $\tilde{\bar{W}}_{ij}$ are defined as:

$$\tilde{\underline{W}}_{ij} = \frac{\underline{W}_{ij}}{\sum_{i=1}^{I}\sum_{j=1}^{J}\underline{W}_{ij}} \tag{26}$$

$$\tilde{\bar{W}}_{ij} = \frac{\bar{W}_{ij}}{\sum_{i=1}^{I}\sum_{j=1}^{J}\bar{W}_{ij}} \tag{27}$$

We can observe from the Eqs. (24–27), there are no variable parameters in this layer, so the nodes are fixed nodes.

Layer 7 (output layer): This layer calculates the output of the IT2FNN. The output processing part of the IT2FNN uses the following expression instead of KM algorithm to perform the type-reduction calculations:

$$\begin{aligned}\tau_{NN} &= qy_l + (1-q)y_r \\ &= \frac{q\sum_{i=1}^{I}\sum_{j=1}^{J}\underline{W}_{ij}f_{ij}}{\sum_{i=1}^{I}\sum_{j=1}^{J}\underline{W}_{ij}} + \frac{(1-q)\sum_{i=1}^{I}\sum_{j=1}^{J}\bar{W}_{ij}f_{ij}}{\sum_{i=1}^{I}\sum_{j=1}^{J}\bar{W}_{ij}}\end{aligned} \tag{28}$$

where the design parameter $q \in [0, 1]$ enables us to share the left and right end point values in the final output [59].

The function of this layer combines the result of layer 6 and the design parameter q, where q is a time-varying modifiable parameter obtained by the learning algorithm given in the next section.

4. SMC theory based learning algorithm and stability analysis

In this section, the IT2FNN parameter learning algorithm is proposed, and then the stability analysis of the control system is given.

4.1. IT2FNN parameter learning rules

In order to realize the trajectory tracking control of the Delta robot, three identical SLIT2FNN controllers are used to control three active joints of the robot, respectively. The architecture of the control system is presented in Fig. 6. The parallel self-learning controller consists of a conventional PD controller and an IT2FNN controller. The PD control law is defined as:

$$\tau_{PD} = K_P e + K_D \dot{e} \tag{29}$$

where K_p is the proportional gain, K_D is the derivative gain. The PD controller is provided both as a feedback controller to guarantee the transient performance of the control system and as a reference signal to train the IT2FNN.

The IT2FNN uses both the feedback error of the Delta robot system and the output of the PD controller to update the antecedent parameters (trapezoidal IT2MFs) and consequent parameters (crisp values of the consequent part and the design parameter q).

Using the SMC theory principles, the sliding surface $s(e,t) =$ 0for the robot system can be defined as:

$$\begin{aligned}s(e,t) = \chi e(t) + \dot{e}(t) &= \frac{K_P}{K_D}e(t) + \dot{e}(t) = \frac{1}{K_D}\tau_{PD}(t) \\ &= \frac{1}{K_D}(\tau(t) + \tau_{NN}(t))\end{aligned} \tag{30}$$

where $\chi = \frac{K_P}{K_D}$ is a strictly positive real constant, since it is a coefficient of the Hurwitz polynomial. If the condition $s(e,t)\dot{s}(e,t) < 0$ is satisfied for all t in $[t, t_f) \subset (0, t_f)$, maintaining the system states on $s(e, t)$ will result in $e(t)$ approaches zero with the initial condition $e(0) \neq 0$ within an finite time t_f. Consequently, the tracking control is fulfilled.

A suitable online learning algorithm for the parameters of the IT2FNN has to be found so as to retain the error $e(t)$ on the sliding surface $s(e,t) = 0$. To achieve this purpose, the parameter adaptive laws are defined as:

$$\dot{\underline{\alpha}}_{1i} = \dot{\underline{c}}_{1i} + (-1)^i\dot{x}_1 \tag{31}$$

$$\dot{\bar{\alpha}}_{1i} = -\dot{\bar{c}}_{1i} - (-1)^i\dot{x}_1 \tag{32}$$

$$\dot{\underline{\alpha}}_{2j} = \dot{\underline{c}}_{2j} + (-1)^j\dot{x}_2 \tag{33}$$

$$\dot{\bar{\alpha}}_{2j} = -\dot{\bar{c}}_{2j} - (-1)^j\dot{x}_2 \tag{34}$$

$$\dot{\underline{c}}_{1i} = -\frac{2\beta\underline{c}_{1i}^2\mathrm{sgn}(\tau_{PD})}{\underline{c}_{1i} - \underline{\alpha}_{1i} + (-1)^i x_1}\underline{\mu}_{1i} \tag{35}$$

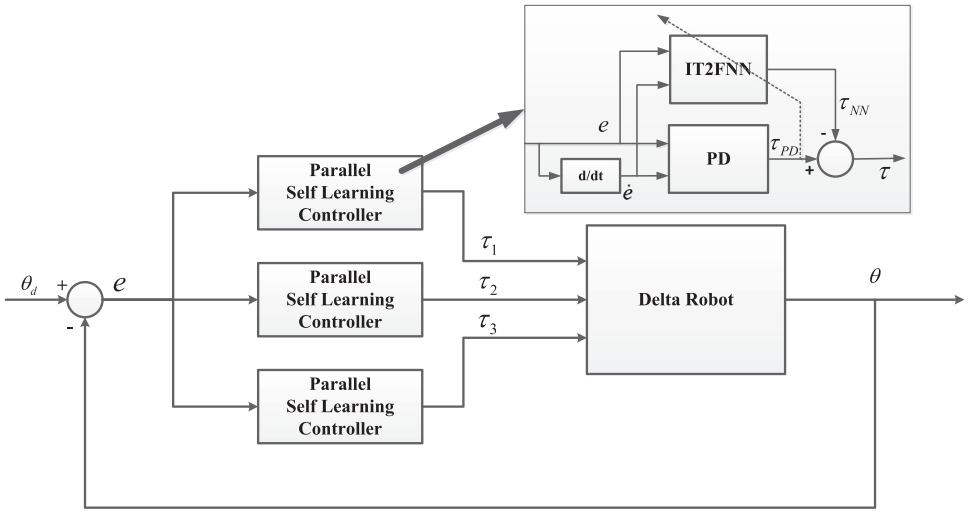


Fig. 6. Structure of the proposed SLIT2FNN control scheme for Delta robot trajectory control.

$$\dot{\bar{c}}_{1i} = -\frac{2\beta \bar{c}_{1i}^2 \mathrm{sgn}(\tau_{PD})}{\bar{c}_{1i} + \bar{\alpha}_{1i} + (-1)^i x_1} \bar{\mu}_{1i} \tag{36}$$

$$\dot{\underline{c}}_{2j} = -\frac{2\beta \underline{c}_{2j}^2 \mathrm{sgn}(\tau_{PD})}{\underline{c}_{2j} - \underline{\alpha}_{2j} + (-1)^j x_2} \underline{\mu}_{2j} \tag{37}$$

$$\dot{\bar{c}}_{2j} = -\frac{2\beta \bar{c}_{2j}^2 \mathrm{sgn}(\tau_{PD})}{\bar{c}_{2j} + \bar{\alpha}_{2j} + (-1)^j x_2} \bar{\mu}_{2j} \tag{38}$$

$$\dot{f}_{ij} = -\frac{\beta \mathrm{sgn}(\tau_{PD}) \left(q\underline{\tilde{W}}_{ij} + (1-q)\bar{\tilde{W}}_{ij} \right)}{D_f} \tag{39}$$

$$\dot{q} = -\frac{\beta \mathrm{sgn}(\tau_{PD})}{\sum_{i=1}^{I}\sum_{j=1}^{J} f_{ij}\left(\underline{\tilde{W}}_{ij} - \bar{\tilde{W}}_{ij} \right)} \tag{40}$$

where $D_f = \sum_{i=1}^{I}\sum_{j=1}^{J} \left(q\underline{\tilde{W}}_{ij} + (1-q)\bar{\tilde{W}}_{ij} \right)^2$, β is a positive design constant.

These parameters determine the structure of the IT2FNN, and then influence the control performance of the proposed approach. The parameters $\underline{\alpha}_{1i}$, $\bar{\alpha}_{1i}$, $\underline{\alpha}_{2j}$ and $\bar{\alpha}_{2j}$ are blurring degrees of the trapezoidal MFs, and the shape of the IT2FS control surface is defined by them, the detailed discussion on the influences of the blurring degrees on the control performance is given in our previous works [53,60]. The position of each trapezoidal MF on the x-axis is determined by $\underline{c}_{1i}$, $\bar{c}_{1i}$, $\underline{c}_{2j}$ and $\bar{c}_{2j}$. f_{ij} equals to the consequent part of each fuzzy rule, and q is the scaling factor to share the left and right end point values of the output of the inference engine.

4.2. Stability analysis

In order to investigate the stability of the system, we have the following assumptions:

Assumption 1. The inputs of the SLIT2FNN system $x_1(t)$ and $x_2(t)$, and their time derivatives are bounded:

$$|x_i(t)| \le X, |\dot{x}_i(t)| \le \dot{X}, i = 1, 2, \forall t \tag{41}$$

where X and $\dot{X}$ are assumed to be some known positive constants.

Assumption 2. The outputs of the controller are bounded by:

$$|\tau_{NN}(t)| \le T, |\dot{\tau}_{NN}(t)| \le \dot{T}, \forall t \tag{42}$$

$$|\tau_{PD}(t)| \le T, |\dot{\tau}_{PD}(t)| \le \dot{T}, \forall t \tag{43}$$

$$|\tau(t)| \le T, |\dot{\tau}(t)| \le \dot{T}, \forall t \tag{44}$$

where T and $\dot{T}$ are assumed to be some known positive constants.

Assumption 3. The positive design parameter β is defined by the following inequality:

$$\beta > \dot{T} \tag{45}$$

Consider the following Lyapunov function candidate:

$$V = \frac{1}{2}s^2 = \frac{1}{2K_D^2}\tau_{PD}^2 \tag{46}$$

Differentiating V yields:

$$\dot{V} = \frac{1}{K_D^2}\tau_{PD}\dot{\tau}_{PD} = \frac{1}{K_D^2}\tau_{PD}(\dot{\tau} + \dot{\tau}_{NN}) \tag{47}$$

From Eq. (28), τ_{NN} can be written as:

$$\begin{aligned}\tau_{NN} &= qy_l + (1-q)y_r \\ &= \frac{q\sum_{i=1}^{I}\sum_{j=1}^{J}\underline{W}_{ij}f_{ij}}{\sum_{i=1}^{I}\sum_{j=1}^{J}\underline{W}_{ij}} + \frac{(1-q)\sum_{i=1}^{I}\sum_{j=1}^{J}\bar{W}_{ij}f_{ij}}{\sum_{i=1}^{I}\sum_{j=1}^{J}\bar{W}_{ij}} \\ &= q\sum_{i=1}^{I}\sum_{j=1}^{J} f_{ij}\underline{\tilde{W}}_{ij} + (1-q)\sum_{i=1}^{I}\sum_{j=1}^{J} f_{ij}\bar{\tilde{W}}_{ij}\end{aligned} \tag{48}$$

The time derivative of τ_{NN} is as follows:

$$\begin{aligned}\dot{\tau}_{NN} = {} & \dot{q}\sum_{i=1}^{I}\sum_{j=1}^{J} f_{ij}\underline{\tilde{W}}_{ij} + q\sum_{i=1}^{I}\sum_{j=1}^{J}\left(\dot{f}_{ij}\underline{\tilde{W}}_{ij} + f_{ij}\dot{\underline{\tilde{W}}}_{ij} \right) \\ & -\dot{q}\sum_{i=1}^{I}\sum_{j=1}^{J} f_{ij}\bar{\tilde{W}}_{ij} \\ & +(1-q)\sum_{i=1}^{I}\sum_{j=1}^{J}\left(\dot{f}_{ij}\bar{\tilde{W}}_{ij} + f_{ij}\dot{\bar{\tilde{W}}}_{ij} \right)\end{aligned} \tag{49}$$

Taking the derivative of Eq. (26), we have:

$$\begin{aligned}\dot{\underline{\tilde{W}}}_{ij} &= \frac{\dot{\underline{W}}_{ij}}{D_1} - \frac{\underline{W}_{ij}\dot{D}_1}{D_1^2} = \frac{\dot{\underline{W}}_{ij}}{D_1} - \frac{\underline{\tilde{W}}_{ij}\dot{D}_1}{D_1} \\ &= \frac{\dot{\underline{\mu}}_{1i}(x_1)\underline{\mu}_{2j}(x_2) + \underline{\mu}_{1i}(x_1)\dot{\underline{\mu}}_{2j}(x_2)}{D_1} \\ &\quad - \frac{\underline{\tilde{W}}_{ij}\left(\sum_{i=1}^{I}\sum_{j=1}^{J}\left(\dot{\underline{\mu}}_{1i}(x_1)\underline{\mu}_{2j}(x_2) + \underline{\mu}_{1i}(x_1)\dot{\underline{\mu}}_{2j}(x_2) \right) \right)}{D_1}\end{aligned} \tag{50}$$

where $D_1 = \sum_{i=1}^{I}\sum_{j=1}^{J}\underline{W}_{ij}$.

From Eqs. (10)–(17), the time derivative of these equations can be calculated as:

$$\dot{\underline{\mu}}_{1i} = \frac{\dot{\underline{c}}_{1i} - \dot{\underline{\alpha}}_{1i} + (-1)^i\dot{x}_1}{2\underline{c}_{1i}} - \frac{\underline{c}_{1i} - \underline{\alpha}_{1i} + (-1)^i x_1}{2\underline{c}_{1i}^2}\dot{\underline{c}}_{1i} \tag{51}$$

$$\dot{\bar{\mu}}_{1i} = \frac{\dot{\bar{c}}_{1i} + \dot{\bar{\alpha}}_{1i} + (-1)^i\dot{x}_1}{2\bar{c}_{1i}} - \frac{c_{1i} + \alpha_{1i} + (-1)^i x_1}{2\bar{c}_{1i}^2}\dot{\bar{c}}_{1i} \tag{52}$$

$$\dot{\underline{\mu}}_{2j} = \frac{\dot{\underline{c}}_{2j} - \dot{\underline{\alpha}}_{2j} + (-1)^j\dot{x}_2}{2\underline{c}_{2j}} - \frac{\underline{c}_{2j} - \underline{\alpha}_{2j} + (-1)^j x_2}{2\underline{c}_{2j}^2}\dot{\underline{c}}_{2j} \tag{53}$$

$$\dot{\bar{\mu}}_{2j} = \frac{\dot{\bar{c}}_{2j} + \dot{\bar{\alpha}}_{2j} + (-1)^j\dot{x}_2}{2\bar{c}_{2j}} - \frac{\bar{c}_{2j} + \bar{\alpha}_{2j} + (-1)^j x_2}{2\bar{c}_{2j}^2}\dot{\bar{c}}_{2j} \tag{54}$$

From Eqs. (31), (33), (35), (37), (51) and Eq. (53), we can get:

$$\begin{aligned}
&\dot{\underline{\mu}}_{1i}(x_1)\underline{\mu}_{2j}(x_2) + \underline{\mu}_{1i}(x_1)\dot{\underline{\mu}}_{2j}(x_2)\\
&= \left(\frac{\dot{\underline{c}}_{1i} - \dot{\underline{\alpha}}_{1i} + (-1)^i\dot{x}_1}{2\underline{c}_{1i}} - \frac{\underline{c}_{1i} - \underline{\alpha}_{1i} + (-1)^i x_1}{2\underline{c}_{1i}^2}\dot{\underline{c}}_{1i}\right)\underline{\mu}_{2j}(x_2)\\
&\quad + \underline{\mu}_{1i}\left(\frac{\dot{\underline{c}}_{2j} - \dot{\underline{\alpha}}_{2j} + (-1)^j\dot{x}_2}{2\underline{c}_{2j}} - \frac{\underline{c}_{2j} - \underline{\alpha}_{2j} + (-1)^j x_2}{2\underline{c}_{2j}^2}\dot{\underline{c}}_{2j}\right)\\
&= 2\beta\underline{\mu}_{1i}(x_1)\underline{\mu}_{2j}(x_2)\mathrm{sgn}(\tau_{PD})\\
&= 2\beta\underline{W}_{ij}\mathrm{sgn}(\tau_{PD})
\end{aligned} \tag{55}$$

Substituting Eq. (55) into Eq. (50), we have:

$$\begin{aligned}
\dot{\underline{W}}_{ij} &= \frac{2\beta\underline{W}_{ij}\mathrm{sgn}(\tau_{PD})}{D_1} - \frac{\tilde{W}_{ij}\left(\sum_{i=1}^{I}\sum_{j=1}^{J}2\beta\underline{W}_{ij}\mathrm{sgn}(\tau_{PD})\right)}{D_1}\\
&= 2\beta\tilde{\underline{W}}_{ij}\mathrm{sgn}(\tau_{PD}) - 2\beta\tilde{\underline{W}}_{ij}\mathrm{sgn}(\tau_{PD})\frac{D_1}{D_1}\\
&= 0
\end{aligned} \tag{56}$$

Similarly, it is clear to know that:

$$\ddot{\tilde{W}}_{ij} = 0 \tag{57}$$

So, the time derivative of τ_{NN} can be written as:

$$\begin{aligned}
\dot{\tau}_{NN} &= \dot{q}\sum_{i=1}^{I}\sum_{j=1}^{J}f_{ij}\tilde{\underline{W}}_{ij} + q\sum_{i=1}^{I}\sum_{j=1}^{J}\left(\dot{f}_{ij}\tilde{\underline{W}}_{ij} + f_{ij}\dot{\tilde{\underline{W}}}_{ij}\right)\\
&\quad -\dot{q}\sum_{i=1}^{I}\sum_{j=1}^{J}f_{ij}\tilde{\bar{W}}_{ij} + (1-q)\sum_{i=1}^{I}\sum_{j=1}^{J}\left(\dot{f}_{ij}\tilde{\bar{W}}_{ij} + f_{ij}\dot{\tilde{\bar{W}}}_{ij}\right)\\
&= \dot{q}\sum_{i=1}^{I}\sum_{j=1}^{J}f_{ij}\tilde{\underline{W}}_{ij} + q\sum_{i=1}^{I}\sum_{j=1}^{J}\dot{f}_{ij}\tilde{\underline{W}}_{ij}\\
&\quad -\dot{q}\sum_{i=1}^{I}\sum_{j=1}^{J}f_{ij}\tilde{\bar{W}}_{ij} + (1-q)\sum_{i=1}^{I}\sum_{j=1}^{J}\dot{f}_{ij}\tilde{\bar{W}}_{ij}
\end{aligned} \tag{58}$$

If Eq. (58) is inserted into Eq. (47), the following is obtained:

$$\begin{aligned}
\dot{V} &= \frac{1}{K_D^2}\tau_{PD}\left(\sum_{i=1}^{I}\sum_{j=1}^{J}\left(q\dot{f}_{ij}\tilde{\underline{W}}_{ij} + (1-q)\dot{f}_{ij}\tilde{\bar{W}}_{ij}\right)\right.\\
&\quad \left.+\dot{q}\sum_{i=1}^{I}\sum_{j=1}^{J}f_{ij}\left(\tilde{\underline{W}}_{ij} - \tilde{\bar{W}}_{ij}\right) + \dot{\tau}\right)\\
&= \frac{1}{K_D^2}\tau_{PD}\left(\sum_{i=1}^{I}\sum_{j=1}^{J}\dot{f}_{ij}\left(q\tilde{\underline{W}}_{ij} + (1-q)\tilde{\bar{W}}_{ij}\right)\right.\\
&\quad \left.+\dot{q}\sum_{i=1}^{I}\sum_{j=1}^{J}f_{ij}\left(\tilde{\underline{W}}_{ij} - \tilde{\bar{W}}_{ij}\right) + \dot{\tau}\right)
\end{aligned} \tag{59}$$

Substituting Eq. (39) and Eq. (40) into Eq. (59), we can get:

$$\begin{aligned}
\dot{V} &= \frac{1}{K_D^2}\tau_{PD}(-2\beta\mathrm{sgn}(\tau_{PD}) + \dot{\tau})\\
&= \frac{1}{K_D^2}(-2\beta|\tau_{PD}| + \tau_{PD}\dot{\tau}) < \frac{1}{K_D^2}|\tau_{PD}|(\dot{T} - 2\beta) < 0
\end{aligned} \tag{60}$$

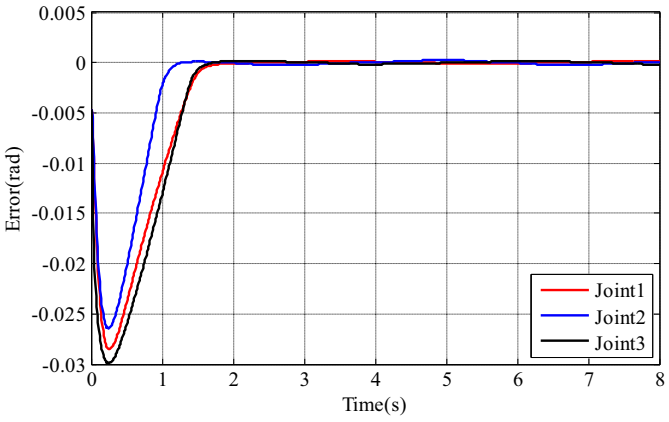


Fig. 7. Joint errors of the robot using SLIT2FNN controller.

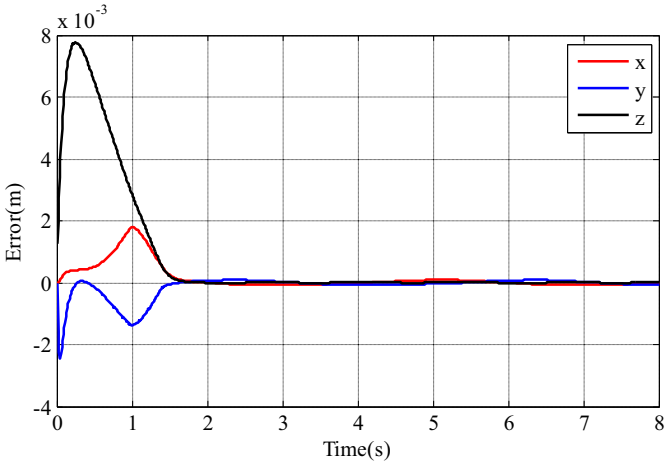


Fig. 8. Position errors of the robot using SLIT2FNN controller.

Thus, $V > 0$ and $\dot{V} < 0$ are guaranteed. It can be concluded that given an arbitrary initial condition $e(0) \neq 0$, the error $e(t)$ and $\tau_{PD}(t)$ will approach zero within a finite time t_f. Therefore, according to Lyapunov stability theory in [61], the controlled system is globally stable and the tracking error of the joint velocity and position convergence are guaranteed by the learning algorithm driving the system to closely track the desired motion trajectories.

In order to alleviate the chattering effect, we inserted the following function instead of the signum function in Eqs. (35)–(40):

$$\mathrm{sgn}(\tau_{PD}) := \frac{\tau_{PD}}{|\tau_{PD}| + \delta} \tag{61}$$

where $\delta = 0.05$.

5. Simulation results and discussion

In this section, the performance of the proposed SLIT2FNN controller is compared with the conventional PD controller by three simulations. For this purpose, these two controllers are connected to the same Delta robot. The control objective is to let the tool center point of the robot follow a given trajectory so that the errors of the position and the velocity are minimized.

In the first simulation, the reference trajectory is chosen as follows:

$$\begin{cases} x = 0.1\cos\left(\frac{\pi}{2}t + \pi\right) + 0.1\\ y = 0.1\sin\left(\frac{\pi}{2}t + \pi\right)\\ z = -0.6769 \end{cases} \tag{62}$$

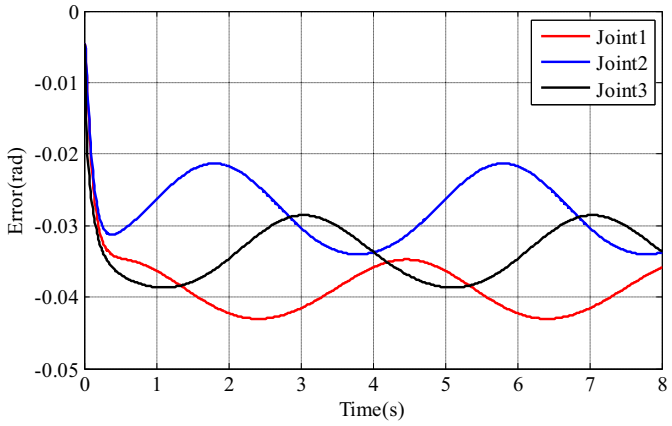


Fig. 9. Joint errors of the robot using PD controller.

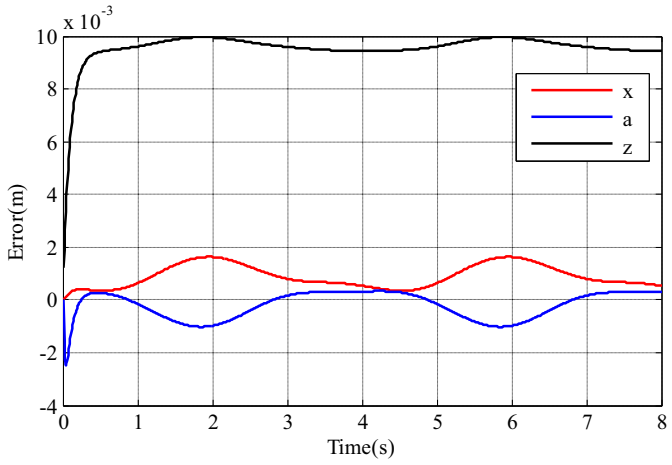


Fig. 10. Position errors of the robot using PD controller.

where (x, y, z) is the position of the tool center point in the workspace, and t is the time. Initial position of the tool center point is $(0, 0, -0.6769)$.

The RMSE values of the three active joints and the position of tool center point are selected as the performance index of the two controllers when the Delta robot finishes a working cycle along the given trajectory in 8 s.

Table 1
RMSE values for simulation 1.

	PD	SLIT2FNN	Improvement (%)
Joint 1	0.0388	0.0077	80
Joint 2	0.0280	0.0064	77
Joint 3	0.0338	0.0086	75
x	9.9132×10^{-4}	4.2494×10^{-4}	57
y	5.6100×10^{-4}	3.8542×10^{-4}	31
z	9.5713×10^{-3}	2.1068×10^{-3}	78

The tracking errors of the SLIT2FNN controller and the PD controller in joint space and workspace are illustrated in Figs. 7–10. The RMSE values of the two controllers are tabulated in Table 1.

We can observe from Figs. 7–10, at the beginning (about 1.7 s), the IT2FNN was tuning its parameters using the learning algorithm (learning procedure), and the control signals of the parallel self-learning controller were mainly comes from the PD controller, because the tracking errors of the two controllers were similar to each other. After the parameters of the IT2FNN were tuned from the default values to the desired values (starting from 1.7 s), the IT2FNN took over the control, and the tracking errors of the robot were significantly decreased. The summarized RMSE values in Table 1 validate the performance improvement of the SLIT2FNN controller.

Fig. 11 represents the variations of the online tuning parameter q of the three SLIT2FNN controllers. We can see the effect of the learning algorithm by observing the variations of q_1, q_2 and q_3 when the robot was tracking a reference trajectory. It also can be concluded that, when the learning procedure was finished (about 1.7 s), the values of q_1, q_2 and q_3 were changing periodically. The main reasons for the periodic changing of the values are: I) the structure of the Delta robot is symmetrical, so the distribution of the inertia matrix is symmetrical in the working space, II) the trajectory given in Eq. (62) is a circle in the working space, which makes the driving torques of the three joints changing periodically.

In order to show the learning process of the SLIT2FNN more clearly, another simulation experiment was carried out. The trajectory keeps unchanged in the second simulation, but the Delta robot needs to run two working cycles. For the first working cycle (8 s), only the PD controller was used to control the robot, and in the second working cycle (starting from 8 s), the SLIT2FNN was applied and working in parallel with the PD controller. Fig. 12 shows the control signals in the second simulation. We can see that, when the SLIT2FNN was connected, the learning process started, the SLIT2FNN became the leading controller after about 1.7 s the same as the result in the last simulation. The desired and actual

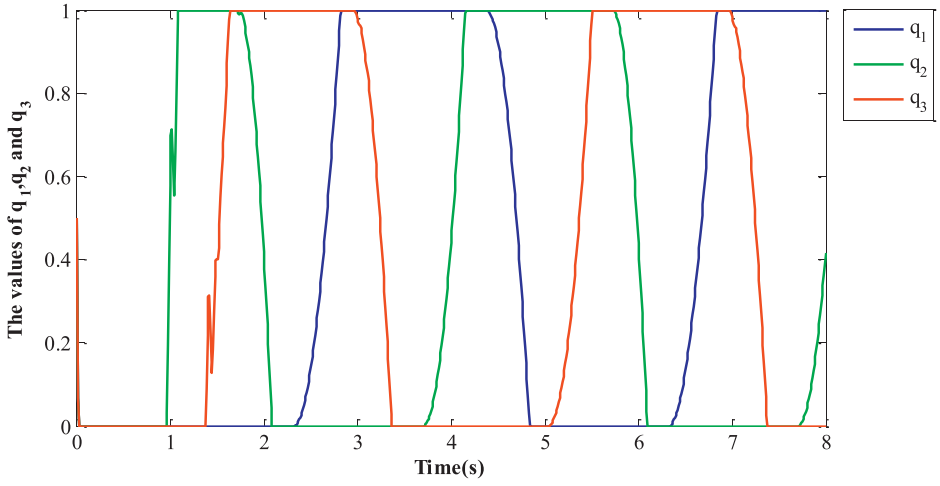
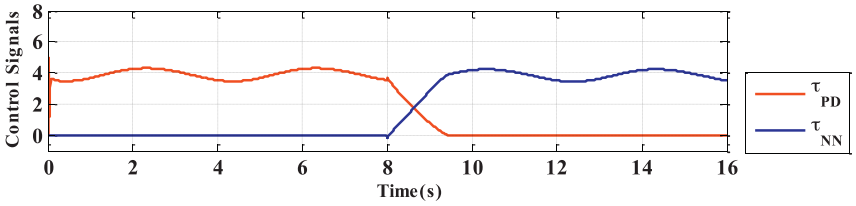
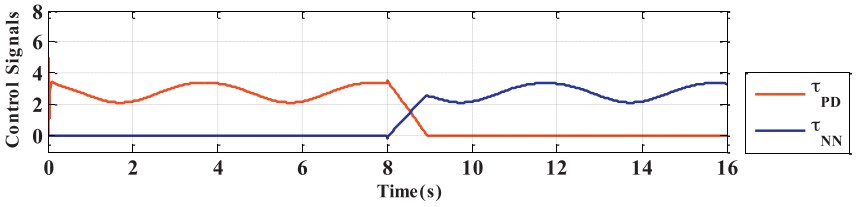


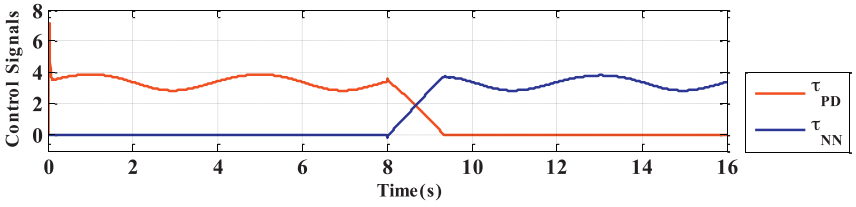
Fig. 11. The values of parameter q.



(a) Joint 1



(b) Joint 2



(c) Joint 3

Fig. 12. The control signals of the two controllers.

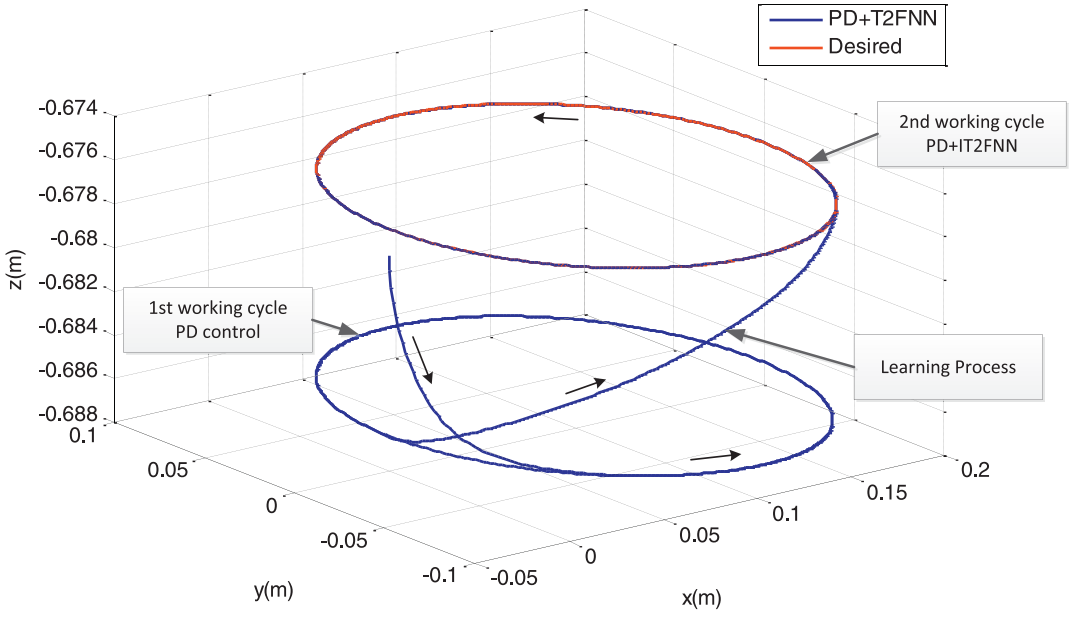


Fig. 13. Trajectory of the travelling platform.

trajectories are shown in Fig. 13, and the tracking errors are shown in Figs. 14 and 15. It can be clearly seen from these figures: I) the learning process of the SLIT2FNN, II) and the trajectory tracking accuracy improvement in the second working cycle.

In the third simulation, the Delta robot was applied to track a new trajectory, which is given in Eq. (63). A 2 kg payload was added to the travelling platform in the second working cycle (starting from 4 s) in order to verify the uncertainty robustness of the aforementioned two controllers. The tracking errors of the two controllers are shown in Figs. 16–19. It is clear that the control performance of the SLIT2FNN can be found to feature to increased precision and robustness in the presence of structured and unstructured uncertainties. The statistical summary of the RMSE values of the two controllers provided in Table 2 validates the previous observations.

$$\begin{cases} x = 0.1\cos\left(\frac{\pi}{2}t + \pi\right) + 0.1 \\ y = 0.1\sin\left(\frac{\pi}{2}t + \pi\right) \\ z = -0.025\sin(2\pi t) - 0.6769 \end{cases} \tag{63}$$

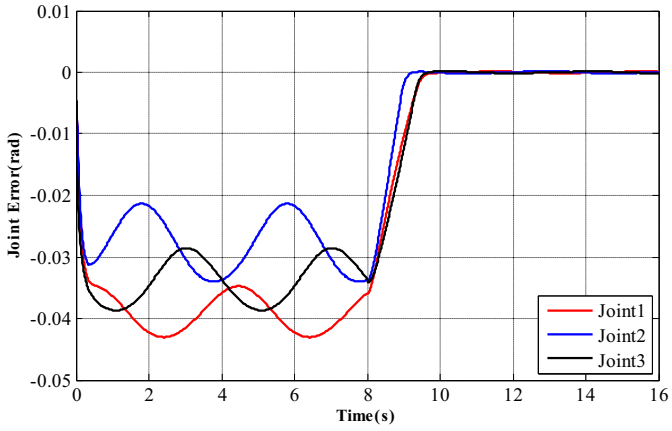


Fig. 14. Joint errors in the second simulation.

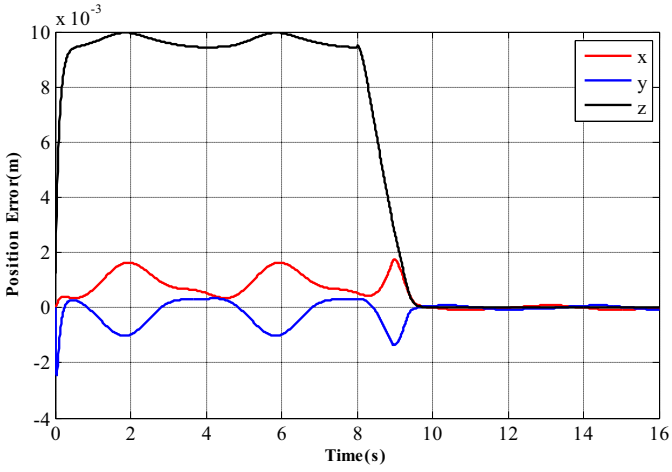


Fig. 15. Position errors in the second simulation.

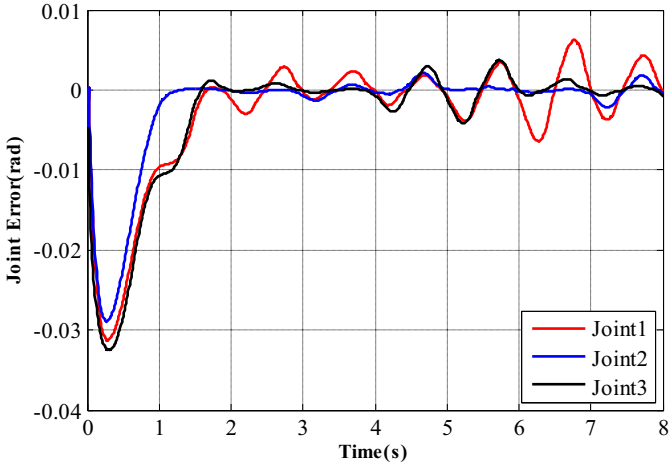


Fig. 16. Joint errors using SLIT2FNN controller with uncertainties.

Table 2
RMSE values for simulation 3.

	PD	SLIT2FNN	Improvement (%)
Joint 1	0.0390	0.0084	78
Joint 2	0.0280	0.0068	76
Joint 3	0.0339	0.0088	74
x	1.1715×10^{-3}	6.4289×10^{-4}	45
y	6.7610×10^{-4}	4.6359×10^{-4}	31
z	9.5980×10^{-3}	2.2643×10^{-3}	76

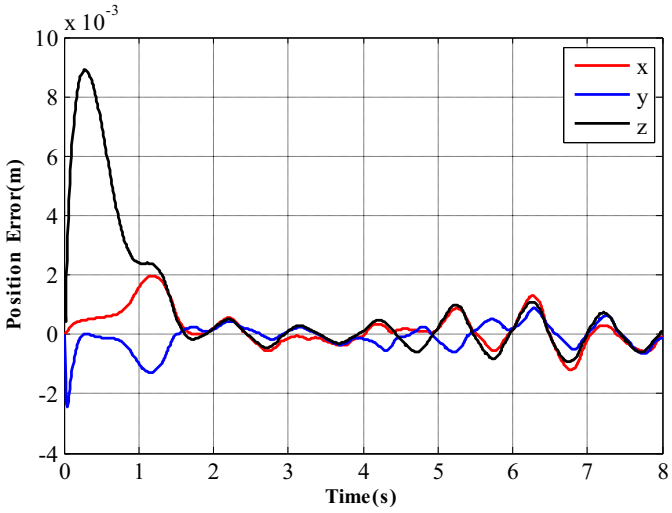


Fig. 17. Position errors using SLIT2FNN controller with uncertainties.

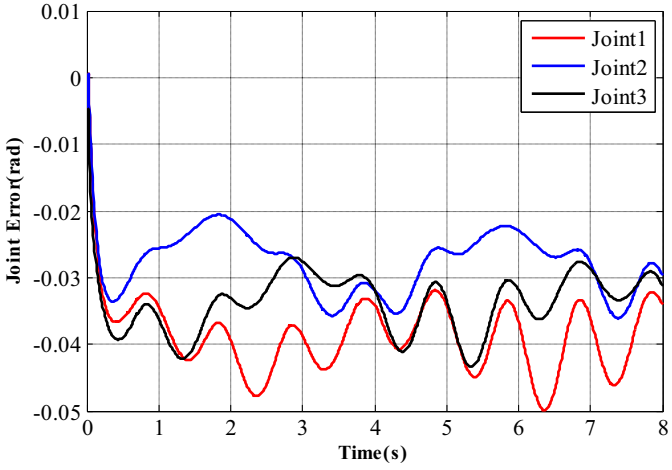


Fig. 18. Joint errors using PD controller with uncertainties.

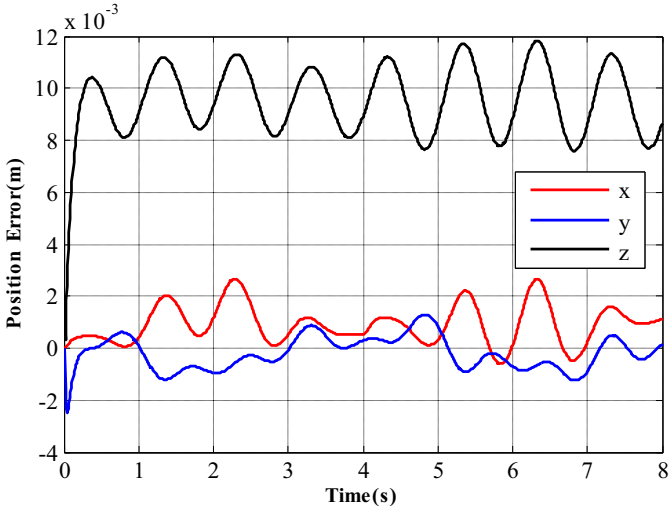


Fig. 19. Position errors using PD controller with uncertainties.

It should be pointed out that the proposed controller requires no prior information about the mathematical model of the controlled object. Here, it can be argued that the performance of PD controller can be improved by carefully tuning. But in real life, there exist un-modeled internal and external uncertainties, if the parameters of the PD controller are better tuned in one circumstances, it doesn't mean that the controller can work well in all other circumstances. Thus, the proposed control structure is more preferable in engineering applications. Because this kind of control structure not only can provide a PD controller's transient performance, but also can get better control accuracy after the learning procedure is done.

6. Conclusion

In this study, we proposed an SLIT2FNN controller for the trajectory tracking control problem of the Delta parallel robot system. The SLIT2FNN controller has a parallel structure, which is a combination of an IT2FNN and a PD controller. We use the PD controller to compensate the transient performance of the intelligent control system during the unavoidable learning procedure of the IT2FNN. A learning algorithm based on SMC theory was proposed to tune the parameters of the IT2FNN. To alleviate the high computational complexity of general T2FLCs, IT2FSs were adopted in the antecedent part of the proposed IT2FNN and crisp numbers were adopted in the consequent part. Unlike other traditional neural network control methodologies, the proposed controller does not need the designer to provide optimized values for the structure parameters of the neural network. Instead, the SLIT2FNN system will tune these parameters online using the proposed SMC theory based learning laws. The stability of the proposed SLIT2FNN control system was demonstrated using Lyapunov stability theorem. Numerical simulation results showed that the SLIT2FNN controller achieved better control performance than the traditional PD controller for robotic trajectory tracking control, in terms of improving the tracking accuracy and the robustness under unknown system uncertainties. Moreover, one of the most important advantages of the control method is that this kind of intelligent control scheme can be easily applied to the existing equipment while involving little changes on traditional controller and not too much investments. This work is one of the first attempts in implementing SLIT2FNN control architecture for the parallel manipulator systems. Our future research will focus on applying this control scheme to a real-time robot system.

References

[1] J. Hirano, D. Tanaka, T. Watanabe, and T. Nakamura, Development of delta robot driven by pneumatic artificial muscles, 2014.
[2] S.B. Park, H.S. Kim, C. Song, et al., Dynamics modeling of a delta-type parallel robot (ISR 2013), 2013 44th International Symposium on Robotics (ISR), IEEE, 2013, pp. 1–5.
[3] O. Linda, M. Manic, Evaluating uncertainty resiliency of Type-2 Fuzzy Logic Controllers for parallel delta robot, in: International Conference on Human System Interaction, 2011, pp. 91–97.
[4] M. Afroun, T. Chettibi, S. Hanchi, Planning optimal motions for a DELTA parallel Robot, 2006. MED'06. 14th Mediterranean Conference on Control and Automation, IEEE, 2006, pp. 1–6.
[5] B.K. Yoo, W.C. Ham, Adaptive control of robot manipulator using fuzzy compensator, IEEE Trans. Fuzzy Syst. 8 (2) (2000) 186–199.
[6] Z. Liu, C. Jiang, J. Liu, et al., Cascaded feedback linearization tracking control of nonholonomic mobile robot, 2013 32nd Chinese Control Conference (CCC), IEEE, 2013, pp. 4232–4237.
[7] D. Chwa, Tracking control of differential-drive wheeled mobile robots using a backstepping-like feedback linearization, IEEE Trans. Syst. Man Cybern. Part A Syst. Hum. 40 (6) (2010) 1285–1295.
[8] B. Achili, B. Daachi, Y. Amirat, A. Ali-cherif, A robust adaptive control of a parallel robot, Int. J. Control 83 (10) (2010) 2107–2119.
[9] J. Zou, Adaptive backstepping control for parallel robot with uncertainties in dynamics and kinematics, Robotica, pp. 1–24, 2014.
[10] J. Cazalilla, M. Vallés, V. Mata, M. Díaz-Rodríguez, A. Valera, Adaptive control of a 3-DOF parallel manipulator considering payload handling and relevant parameter models, Rob. CIM-Int. Manuf. 30 (30) (2014) 468–477.
[11] M. Ertugrul, O. Kaynak, Neuro sliding mode control of robotic manipulators, Mechatronics 10 (1–2) (2000) 239–263.
[12] M. Jin, J. Lee, P.H. Chang, C. Choi, Practical nonsingular terminal sliding-mode control of robot manipulators for high-accuracy tracking control, IEEE Trans. Ind. Electron 56 (9) (2009) 3593–3601.
[13] Y.B. Liu, H.M. Zhang, X.H. Zhao, Robust adaptive fuzzy sliding-mode control method of 3-RRRT parallel robot, Electr. Mach. Control 21 (11) (2009) 3398-3386.
[14] S.A. Moezi, M. Rafeeyan, S. Ebrahimi, Sliding mode control of 3-RPR parallel robot on the optimal path using Cuckoo optimization algorithm, Modares Mech. Eng. 15 (2015) 147–158.
[15] S.R.S. Abdullah, M.M. Mustafa, R.A. Rahman, et al., A fuzzy logic controller of two-position pump with time-delay in heavy metal precipitation process, 2011 International Conference on Pattern Analysis and Intelligent Robotics (ICPAIR), 1, IEEE, 2011, pp. 171–176.
[16] R.M. Hilloowala, A. Sharaf, A rule-based fuzzy logic controller for a PWM inverter in a standalone wind energy conversion scheme, IEEE Trans. Ind. Appl. 32 (1) (1996) 57–65 1996-01-01.
[17] N. Kehtarnavaz, E. Nakamura, N. Griswold, et al., Autonomous vehicle following by a fuzzy logic controller, 1994. Industrial Fuzzy Control and Intelligent Systems Conference, and the NASA Joint Technology Workshop on Neural Networks and Fuzzy Logic, Fuzzy Information Processing Society Biannual Conference, IEEE, 1994, pp. 333–337.
[18] E. Pathmanathan, R. Ibrahim, Development and implementation of fuzzy logic controller for flow control application, 2010 International Conference on Intelligent and Advanced Systems (ICIAS), IEEE, 2010, pp. 1–6.
[19] O.A. Arqub, Adaptation of reproducing kernel algorithm for solving fuzzy Fredholm–Volterra integrodifferential equations, Neural Comput. Appl. 28 (7) (2017) 1591–1610 2017-01-01.
[20] M. Yahyaei, J.E. Jam, R. Hosnavi, Controlling the navigation of automatic guided vehicle (AGV) using integrated fuzzy logic controller with programmable logic controller (IFLPLC)—stage 1, Int. J. Adv. Manuf. Technol. 47 (5-8) (2010) 795–807.
[21] Z. Xia, J. Li, J. Li, Delay-dependent non-fragile H∞ filtering for uncertain fuzzy systems based on switching fuzzy model and piecewise Lyapunov function, Int. J. Autom. Comput. 7 (4) (2010) 428–437.
[22] K.H. Su, S.J. Huang, C.Y. Yang, Development of robotic grasping gripper based on smart fuzzy controller, Int. J. Fuzzy Syst. 17 (4) (2015) 595–608.
[23] V.B. Nguyen, A.S. Morris, Genetic algorithm tuned fuzzy logic controller for a robot arm with two-link flexibility and two-joint elasticity, J. Intell. Rob. Syst. 49 (1) (2007) 3–18.
[24] J.M. Mendel, Uncertain rule-based fuzzy logic system: introduction and new directions, 2001.
[25] O. Castillo, L. Amador-Angulo, J.R. Castro, M. Garcia-Valdez, A comparative study of type-1 fuzzy logic systems, interval type-2 fuzzy logic systems and generalized type-2 fuzzy logic systems in control problems, Inf. Sci. 354 (2016) 257–274 2016-01-01.
[26] H.A. Hagras, A hierarchical type-2 fuzzy logic control architecture for autonomous mobile robots, IEEE Trans Fuzzy Syst. 12 (4) (2004) 524–539.
[27] J.M. Mendel, R.I.B. John, Type-2 fuzzy sets made simple, IEEE Trans. Fuzzy Syst. 10 (2) (2002) 117–127.
[28] K. Tai, A. El-Sayed, M. Biglarbegian, C.I. Gonzalez, O. Castillo, S. Mahmud, Review of recent type-2 fuzzy controller applications, Algorithms 9 (2) (2016) 39 2016-01-01.
[29] O. Castillo, L. Cervantes, J. Soria, M. Sanchez, J.R. Castro, A generalized type-2 fuzzy granular approach with applications to aerospace, Inf. Sci. 354 (2016) 165–177 2016-01-01.
[30] L. Cervantes, O. Castillo, Type-2 fuzzy logic aggregation of multiple fuzzy controllers for airplane flight control, Inf. Sci. 324 (2015) 247–256 2015-01-01.
[31] M.A. Sanchez, O. Castillo, J.R. Castro, Generalized type-2 fuzzy systems for controlling a mobile robot and a performance comparison with interval type-2 and type-1 fuzzy systems, Expert Syst. Appl. 42 (14) (2015) 5904–5914 2015-01-01.
[32] O. Castillo, P. Melin, A review on interval type-2 fuzzy logic applications in intelligent control, Inf. Sci. 279 (2014) 615–631.
[33] G.M. Mendez, O. Castillo, R. Colas, H. Moreno, Finishing mill strip gage setup and control by interval type-1 non-singleton type-2 fuzzy logic systems, Appl. Soft Comput. 24 (2014) 900–911.
[34] H.G. Lee, H.T. Jeon, Interval type-2 fuzzy logic control system of flight longitudinal motion, J. Korean Inst. Intell. Syst. 25 (2) (2015) 168–173.
[35] G.M. Mendez, L. Leduc-Lezama, R. Colas, G. Murillo-Perez, J. Ramirez-Cuellar, J.J. Lopez, Modelling and control of coiling entry temperature using interval type-2 fuzzy logic systems, Ironmak. Steelmak. 37 (2) (2010) 126–134.
[36] S. Barkat, A. Tlemcani, H. Nouri, Noninteracting adaptive control of PMSM using interval type-2 fuzzy logic systems, IEEE Trans. Fuzzy Syst. 19 (5) (2011) 925–936.
[37] H. Chaoui, W. Gueaieb, Type-2 fuzzy logic control of a flexible-joint manipulator, J. Intell. Rob. Syst. 51 (2) (2008) 159–186.
[38] W. Gao, R.R. Selmic, Neural network control of a class of nonlinear systems with actuator saturation, IEEE Trans. Neural Netw. 17 (1) (2006) 147–156.
[39] C.P. Bechlioulis, Z. Doulgeri, G.A. Rovithakis, Neuro-adaptive force/position control with prescribed performance and guaranteed contact maintenance, IEEE Trans. Neural Netw. 21 (12) (2010) 1857–1868.
[40] S. Cong, Y. Liang, PID-like neural network nonlinear adaptive control for uncertain multivariable motion control systems, IEEE Trans. Ind. Electron. 56 (10) (2009) 3872–3879.

[41] J.I. Mulero-martinez, Robust GRBF static neurocontroller with switch logic for control of robot manipulators, IEEE Trans. Neural Netw. Learn. Syst. 23 (23) (2012) 1053–1064.
[42] S. Beyhan, M. Alci, Extended fuzzy function model with stable learning methods for online system identification, Int. J. Adapt. Control 25 (2) (2011) 168–182 2011-01-01.
[43] O.A. Arqub, Z. Abo-Hammour, Numerical solution of systems of second-order boundary value problems using continuous genetic algorithm, Inf. Sci. 279 (2014) 396–415 2014-01-01.
[44] S.P. Panigrahi, S.K. Nayak, S.K. Padhy, A genetic-based neuro-fuzzy controller for blind equalization of time-varying channels, Int. J. Adapt. Control 22 (7) (2008) 705–716 2008-01-01.
[45] T. Niknam, M.H. Khooban, A. Kavousifard, M.R. Soltanpour, An optimal type II fuzzy sliding mode control design for a class of nonlinear systems, Nonlinear Dyn. 75 (75) (2014) 73–83.
[46] T.C. Lin, Based on interval type-2 fuzzy-neural network direct adaptive sliding mode control for SISO nonlinear systems, Commun. Nonlinear Sci. Numer. Simul. 15 (12) (2010) 4084–4099.
[47] M.R. Akbarzadeh-T, R. Shahnazi, Direct adaptive fuzzy PI sliding mode control of systems with unknown but bounded disturbances, Iran. J. Fuzzy Syst. 3 (2) (2006) 5–6.
[48] H. Chaoui, W. Gueaieb, M.C.E. Yagoub, et al., Hybrid neural fuzzy sliding mode control of flexible-joint manipulators with unknown dynamics, IECON 2006-32nd Annual Conference on IEEE Industrial Electronics, IEEE, 2006, pp. 4082–4087.
[49] S. Zeghlache, T. Benslimane, N. Amardjia, A. Bouguerra, Interval type-2 fuzzy sliding mode controller based on nonlinear observer for a 3-DOF helicopter with uncertainties, Int. J. Fuzzy Syst. (2016) 1–20.
[50] A.V. Topalov, E. Kayacan, Y. Oniz, et al., Adaptive neuro-fuzzy control with sliding mode learning algorithm: Application to antilock braking system, Proceedings of the ASCC 2009 7th Asian Control Conference, IEEE, 2009, pp. 784–789.
[51] G.G. Parma, B.R. Menezes, A.P. Braga, Sliding mode algorithm for training multilayer artificial neural networks, Electron. LETT 34 (1) (1998) 97–98.
[52] E. Kayacan, E. Kayacan, H. Ramon, O. Kaynak, Towards agrobots: trajectory control of an autonomous tractor using type-2 fuzzy logic controllers, IEEE/ASME Trans. Mechatronics 20 (1) (2015) 287–298.
[53] X.G. Lu, M. Liu, J.X. Liu, Design and optimization of interval type-2 fuzzy logic controller for delta parallel robot trajectory control, Int. J. Fuzzy Syst. 19 (1) (2017) 190–206.
[54] A. Codourey, Dynamic modeling of parallel robots for computed-torque control implementation, Int. J. Rob. Res. 17 (12) (1998) 1325–1336 1998-12-01.
[55] D. Wu, J.M. Mendel, Designing practical interval type-2 fuzzy logic systems made simple, 2014 IEEE International Conference on Fuzzy Systems (FUZZ-IEEE), IEEE, 2014, pp. 800–807.
[56] M.B. Begian, W.W. Melek, J.M. Mendel, Parametric design of stable type-2 TSK fuzzy systems, NAFIPS 2008. Annual Meeting of the North American Fuzzy Information Processing Society, IEEE, 2008, pp. 1–6.
[57] O. Castillo, J.R. Castro, P. Melin, A. Rodriguez-Diaz, Universal approximation of a class of interval type-2 fuzzy neural networks in nonlinear identification, Adv. Fuzzy Syst. (2013) 7 vol2013-01-01 2013.
[58] O. Castillo, J.R. Castro, P. Melin, A. Rodriguez-Diaz, Application of interval type-2 fuzzy neural networks in non-linear identification and time series prediction, Soft Comput. 18 (6) (2014) 1213–1224 2014-06-01.
[59] D. Wu, Approaches for reducing the computational cost of interval type-2 fuzzy logic systems: overview and comparisons, IEEE Trans. Fuzzy Syst. 21 (1) (2013) 80–99.
[60] X. Lu, M. Liu, J. Liu, et al., Derivation and analysis of a self-tuning interval type-2 fuzzy PI controller, 2016 IEEE 25th International Symposium on Industrial Electronics (ISIE), IEEE, 2016, pp. 362–368.
[61] H. Khalil, Nonlinear Systems, 3rd ed., 2001.

Xingguo Lu received the B.S. degree in Mechanical Engineering in 2004, the M.E. degree in Mechatronics Engineering in 2010, both from Harbin Institute of Technology, Harbin, China. Since 2012, he is a Ph.D. candidate in Mechatronics Engineering at State Key Laboratory of Robotics and System, Harbin Institute of Technology, Harbin, China.

His current research interests include interval type-2 fuzzy logic control, intelligent control of robot, optimization algorithms and design of parallel robots.

Yue Zhao received the B.S. degree in electronic information engineering, and M.S. degree in Information and Communication Engineering from Harbin Institute of Technology, Harbin, China, in 2008 and 2010, respectively. She received the Ph.D. degree in image and system from INSA de Lyon, Lyon, France, in 2014. Since 2014, she has been an Assistant Professor with the School of Astronautics of Harbin Institute of Technology.

Her main research interests include pattern recognition, feature detection, deep learning and medical image analysis.

Ming Liu received his Ph.D. degree in mechanical engineering from Ruhr-Universität, Bochum, Germany, in 2001. Since 2010, he is a professor with Harbin Institute of Technology, China.

His current research interests include dynamic modelling and simulation, mechanical-electro-hydraulic hybrid drive systems.