Binary Channels Fuzzy Self-Adjusted Neural Network for Solving Time-Changing QP Problems

Yamei Luo, *Member, IEEE*, Qingyi Ren, Zihao Zheng, Siyuan Chen, Yu Liu, Zhijun Zhang*

***Abstract*—A novel binary channels fuzzy self-adjusted neural network (BCF-SANN) is proposed and researched for solving time-changing quadratic programming (QP) problems in this paper. Unlike the fixed parameters of typical zeroing neural network, the main parameters of the proposed BCF-SANN are time-changing, and its errors are adaptively quickly convergent. The biggest advantage of the novel neural network is that it combines fuzzy self-adjusted controller which takes the errors and derivatives of errors as fuzzy inputs and neural networks, further improves the convergence and robustness of the neural networks. To design the novel neural network, a time-changing QP problem is firstly established, then using Lagrange's law, the time-changing QP problem is transformed into a time-changing matrix equation, finally, based on the time-changing parameter neural dynamics method, a novel BCF-SANN is proposed. The detailed design process is given in this paper, and the convergence and robustness of the proposed BCF-SANN are proved by theoretical analysis. Through comparative experiments, it is demonstrated that the proposed BCF-SANN has faster convergence rate and stronger robustness than the traditional zeroing neural network and one-dimensional fuzzy recurrent neural network.**

***Index Terms*—Recurrent neural network, zeroing neural network, fuzzy controller, time-changing QP problem, convergence, robustness**

I. Introduction

TIME changing quadratic programming (QP) problem is broadly used in many areas, for example, in the case of robot kinematics [1]–[4], signal processing [5], nonlinear programming [6], [7] and design for controller [8], [9].

Neural networks, as parallel computing methods, are powerful approaches for solving time-changing QP problems. Most current neural networks use gradient descent techniques as their learning strategy. For instance, Wang proposed using the gradient descent method to create a neural network (termed as GNN), which is an implicit dynamics approach [10]. The mode of GNN is a collection of n independent subnetworks, and the order of their coefficient matrices is n [11]. By utilizing the time derivatives of the time-changing parameter matrix and constructing matrix valued error function, the mode of GNN is capable of solving the complex matrix equation problems in real time. However, GNN does not exactly converges to the time-changing problems because the GNN method is a implicit dynamics by constructing an energy function for handling static problems [12]. Then zeroing neural network (ZNN) with explicit dynamics was designed to solve time-changing problems [13]–[15]. ZNN was effective in solving time-changing problems when various types of activation functions are used [16]. By constructing an matrix/vector error function and using the time derivative of error function, the residual errors of ZNN can achieve exponential convergence, which is used to solve time-changing problems. However, ZNN is unable to obtain the precise solutions in a finite amount of time while solving fast time-changing problems or time-changing problems with external disturbance.

In order to obtain precise solutions of fast time-changing problems with external disturbance, Zhang *et al.* proposed a varying-parameter convergent differential neural network (VPCDNN), of which the residual errors can achieve super-exponential convergence [17]. In contrast with ZNN with fixed design parameters, VPCDNN with monotonically increasing time-varying design-parameters has better convergence performance [18]–[20]. To increase the rate of convergence, Zhang *et al.* further exploited an exponential-enhanced-type VPCDNN (E-VPCDNN) [21]. Since the E-VPCDNN possessed super-exponential convergence performance, it has been utilized in solving disturbed time-changing inversion systems [22], mutual collision avoidance of redundant robots' arm [23], unmanned aerial vehicles [24], motion planning of redundant robots [25], and so on. Additionally, researchers also presented a circadian rhythms neural network to solve time-changing problems with periodic noises [26].

Although both the VPCDNN and circadian rhythms neural networks have fast convergent rate and strong noise suppression ability, the expert experience could not be considered and integrated into the design process of the recurrent neural networks. In addition, the parameter saturation problems may exist in VPCDNN because its time-changing parameters amplify the coefficients to a very large number even infinity. To overcome the deficiencies mentioned above, inspired by the thought of VPCDNN, researchers try to combine the recurrent neural network with the adaptive fuzzy controller to further improve the convergence and robustness of recurrent neural networks. Fuzzy theory based on the fuzzy set and fuzzy logic was originally proposed by Zadeh [27]–[29] and was widely used in pattern recognition, artificial intelligence [30]–[36], etc. Fuzzy logic controllers, which consist of three steps of a fuzzification, fuzzy logic, and defuzzification, are typical-ly used with highly nonlinear systems that have significant parameter variations [37]. Generally, designing a fuzzy logic controller is equivalent to constructing an algorithm that can transform the control method which are founded on experience without accurate mathematical strategies into an automated

Yamei Luo, Qingyi Ren, Zihao Zheng, Siyuan Chen, Yu Liu and Zhijun Zhang are with the School of Automation Science and Engineering, South China University of Technology, Guangzhou 510640, China. (Corresponding author*, e-mail: auzjzhang@scut.edu.cn)

control method [38].

In 1994, H. A. Malki proposed the fuzzy proportional-derivative (termed as fuzzy PD) controller, which has the same linear structure as a traditional digital PD controller. Although the proportional-derivative controller's inputs are linear, due to the nonlinear elements in the functions of the controller, the gains are nonlinear [39]. Zamani *et al.* developed an adaptive fuzzy proportional-derivative controller, whose output gain can be adaptively adjusted according to the type of input function, and the control effect is better than other existed control strategies [40]. Afterwards, an adaptive fuzzy recurrent neural network (termed as AFRNN) is proposed in order to reduce the end-effector position accumulation errors and joint-angular drifts efficiently of a redundant robot, which increases computational accuracy and enhances network robustness [41]. In order to achieve strong robustness, the convergent parameters of AFRNN are designed as the fuzzy controller's outputs. Compared with the GNN, ZNN, and VPCDNN models, the convergent parameters of the AFRNN are self-adaptive, and they can adjust themselves to adapt to the changes of the magnitude's error.

Inspired by the design method of AFRNN and fuzzy PD controller, a binary channels fuzzy self-adjusted neural network (termed as BCF-SANN) is proposed to improve the accuracy and robustness of the control system. In AFRNN model, the error e is the only decisive factor of convergent parameters, while in BCF-SANN model both the predictive error (difference between expected and actual values) and its rate of change are taken to the fuzzy controller as two input variables, which determine the output as convergence coefficients. That is to say, in BCF-SANN, neural-dynamic convergent parameters can adjust themselves precisely when the error or its derivative exists. It is important to note that the BCF-SANN will be useful to resolve time-changing optimum control problems, including the redundant robot manipulators' motion planning.

The following is the paper's organization structure. In Section II, a novel BCF-SANN inspired by the design of AFRNN is proposed. In Section III, the convergence and robustness of the proposed BCF-SANN model are analyzed and discussed under various assumptions (using different activation functions). In Section IV, a simulative time-changing QP problem is used to demonstrate the convergence and robustness of the proposed BCF-SANN model. In Section V, the final conclusions are presented. The significant contributions of this paper are as below:

- A novel binary channels fuzzy self-adjusted neural network (BCF-SANN) is proposed and used to solving time-changing quadratic programming (QP) problems. This is the first time to combine binary channels self-adjusted fuzzy controller with recurrent neural networks.
- Theoretical analysis and simulations with different activation functions are conducted to illustrate the faster convergent rate and stronger robustness of the proposed BCF-SANN.
- Compared with ZNN, the proposed BCF-SANN has self-adjusted convergent parameters which adapt to the changes of the magnitude of error and the time derivative of the error. BCF-SANN with self-adjusted convergent parameters can realize stronger robustness against external interference.
- Compared with AFRNN, the BCF-SANN model shows better self adaptive capability and faster responses to error as time varies.

II. Problem Formulation

In this section, a time-changing QP problem is formulated firstly. Secondly, based on neural dynamical design methods and fuzzy control theory, the BCF-SANN model is exploited and analyzed.

A. Time-Changing Convex QP Problem

Generally, a time-changing convex QP problem can be described as

$$\begin{aligned} \text{min.} \quad & y(t)^{\mathrm{T}}B(t)y(t)/2 + c^{\mathrm{T}}(t)y(t) && (1) \\ \text{s. t.} \quad & J(t)y(t) = b(t) && (2) \end{aligned}$$

where $t \geq 0$ and $B(t) \in \mathbb{R}^{\varrho\times\varrho}$ denote the continuous time and a positive-definite matrix, respectively; $b(t) \in \mathbb{R}^{\varrho}$ and $J(t) \in \mathbb{R}^{\sigma\times\varrho}(\varrho \geq \sigma)$ denote a coefficient vector and a coefficient matrix; $c(t) \in \mathbb{R}^{\sigma}$ denotes a coefficient vector. In order to simplify and avoid duplication, the independent variable t is ignored in the derivation of the formula or equation.

The Lagrange approach is used to resolve QP problems, and the Lagrange functions for QP problems (1)-(2) are defined as follows, i.e.,

$$\begin{aligned} \mathcal{L}(y,\eta,t) =& y^{\mathrm{T}}By/2 + c^{\mathrm{T}}y \\ &+ \eta^{\mathrm{T}}(Jy-b), t \in [0,+\infty) \end{aligned} \quad (3)$$

where $\eta \in \mathbb{R}^{\sigma}$ denotes a Lagrange multiplier vector. According to the Lagrange condition, the partial derivatives of equation (3) should satisfy the following conditions, i.e.,

$$\begin{cases} \dfrac{\partial\mathcal{L}(y,\eta,t)}{\partial y} & = By + c + J^{\mathrm{T}}\eta = 0, \\ \dfrac{\partial\mathcal{L}(y,\eta,t)}{\partial\eta} & = Jy - b = 0. \end{cases}$$

The equations above can be rewritten as a matrix equation form, i.e.,

$$Px = u \quad (4)$$

where

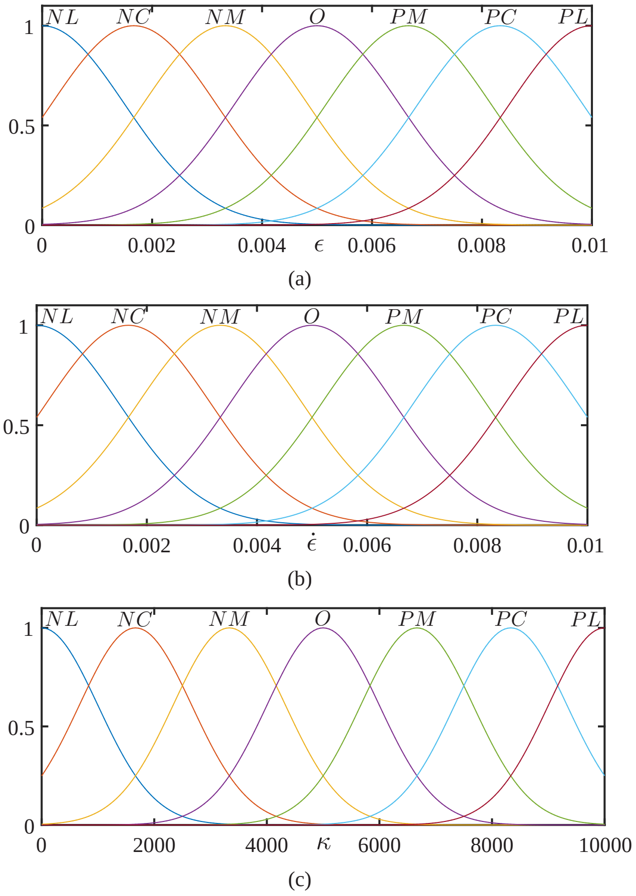
$$P := \begin{bmatrix} B & J^{\mathrm{T}} \\ J & 0 \end{bmatrix} \in \mathbb{R}^{(\varrho+\sigma)\times(\varrho+\sigma)}$$

$$x := \begin{bmatrix} y \\ \eta \end{bmatrix} \in \mathbb{R}^{\varrho+\sigma} \quad , \; u := \begin{bmatrix} -c \\ b \end{bmatrix} \in \mathbb{R}^{\varrho+\sigma}.$$

Defined the error as follow

$$\epsilon = Px - u \in \mathbb{R}^{\varrho+\sigma}. \quad (5)$$

When Equation (4) is met and the error ϵ is anticipated to converge to zero, the solution x is found.



(a)

(b)

(c)

Fig. 1. (a) Membership function of the input ϵ. (b) Membership function of the input $\dot{\epsilon}$. (c) Membership function of the output κ.

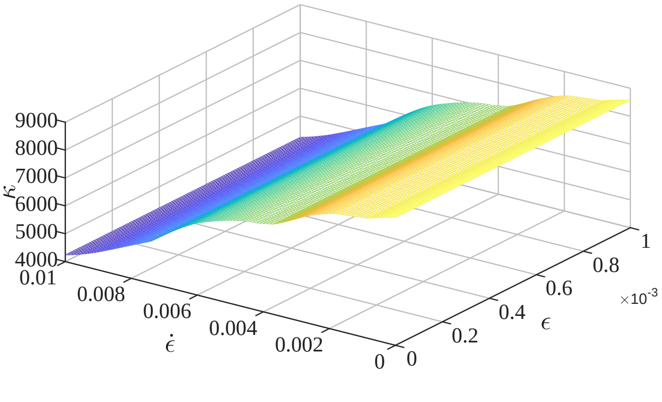


Fig. 2. The relationship between the output κ and two inputs ϵ and $\dot{\epsilon}$ of binary channels fuzzy self-adjusted controller. (ϵ as x-axis, $\dot{\epsilon}$ as y-axis, κ as z-axis.)

B. Binary Channels Fuzzy Self-Adjusted Neural Network

In practical applications, due to the complexity of a real-world, the control system often exists large system error and external interference. It is expected that the system error converges to zero quickly and the system has strong robustness, so a fuzzy processing rule is necessary [41].

Based on the neural dynamics design method and the fuzzy strategy design method, the following fuzzy varying-parameter design formula can be obtained as

$$\frac{\mathrm{d}\epsilon}{\mathrm{d}} = -(\nu + \kappa)\boldsymbol{\Theta}(\epsilon) \tag{6}$$

where $\nu + \kappa > 0$ determines the range of convergence coefficient with the constant $\nu > 0$ and κ denotes the output of the binary channels fuzzy self-adjusted controller; $\boldsymbol{\Theta}(\cdot) : \mathbb{R}^{\varrho+\sigma} \to \mathbb{R}^{\varrho+\sigma}$ represents an activation-function. It is noteworthy that any monotonically growing odd function may serve as the activation-function.

Substitute (5) into (6), we can obtain

$$P\dot{x} = -\dot{P}x - (\nu + \kappa)\boldsymbol{\Theta}(Px - u) + \dot{u} \tag{7}$$

where $\dot{P} = \mathrm{d}P/\mathrm{d}t$, and $\dot{x} = \mathrm{d}x/\mathrm{d}t$. The convergent coefficient $\nu + \kappa$ consists of constant value and output of the binary channels fuzzy self-adjusted controller, it is designed by using fuzzy control techniques and the differential equations theorem. The implicit dynamical equation (7) is referred to as the BCF-SANN model, which can be used to solve equation (4).

For further improving the convergence rate and computational precision, the output of the binary channels fuzzy self-adjusted controller with the error function (5) and the derivatives of error function(5) as inputs is expressed as κ. This is a novel method, which does not require an accurate mathematical model, and shows good tolerance and adaptability for systems which are too complex or difficult to describe accurately. Specifically, there are three steps need to be carried out in designing the binary channels fuzzy self-adjusted controller.

1) **Fuzzification.** Fuzzification is a procedure that converts the fuzzy controller's input and output values into the appropriate fuzzy language variable values. Fuzzification involves three steps:
   - **Describe the fuzzy controller's structure.** The binary channels fuzzy self-adjusted controller is composed of two inputs (i.e., the error ϵ and the derivative of the error $\dot{\epsilon}$) and one output (i.e., κ).
   - **Define fuzzy input subsets and fuzzy output subsets.** The input subsets E and DE and output subset K contains all input values ϵ, $\dot{\epsilon}$ and output value κ. In this paper, the input subsets and output subset are all defined as

     $$\{NL,\ NC,\ NM,\ O,\ PM,\ PC,\ PL\}$$

     where NL denotes negative large, NC denotes negative center, NM denotes negative mini, O denotes zero, PM denotes positive mini, PC denotes positive center, and PL denotes positive large.
   - **Membership function.** $\mathcal{A}$ is a fuzzy set on $\mathcal{U}$ if for any element χ in domain $\mathcal{U}$, there is $\mathcal{A}(\chi) \in [0, 1]$ corresponding to it, and $\mathcal{A}(\chi)$ is called the degree of membership of χ to $\mathcal{A}$. When χ varies in $\mathcal{U}$, $\mathcal{A}(\chi)$ is a function called the membership function of $\mathcal{A}$. The closer the degree of membership $\mathcal{A}(\chi)$ is to 1, the higher the degree to χ belongs to $\mathcal{A}$. The closer $\mathcal{A}(\chi)$ is to 0 the lower the degree to which

χ belongs to $\mathcal{A}$. The membership function $\mathcal{A}(\chi)$, which takes values in $[0, 1]$.

There are usually four commonly used membership functions, they are triangular membership function, bell membership function, Gaussian membership function and sigmoid membership function [41]. In this paper, Gaussian membership function are used as binary channels inputs membership functions and output membership function, the binary channels inputs membership functions are represented by $\mu_E(\epsilon)$ and $\mu_{DE}(\dot{\epsilon})$, where $\mu_E(\epsilon)$ denotes the degree to which ϵ belongs to the fuzzy subset E, $\dot{\epsilon}$ denotes the degree to which $\dot{\epsilon}$ belongs to the fuzzy subset DE, and the output membership function is represented by $\mu_K(\kappa)$ which denotes the degree of κ belongs to the fuzzy subset K. Fig. 1(a), Fig. 1(b) and Fig. 1(c) respectively shows the binary channels inputs membership functions and output membership function.

2) **Fuzzy inference engine.** The fuzzy control rules of the binary channels input fuzzy subsets E and DE and the single output fuzzy subsets K use "IF-THEN" rules, for example,

- $\mathcal{R}^{(1)}$: if $E = PL$ and $DE = PL$, then $K = PL$;
  ...
- $\mathcal{R}^{(8)}$: if $E = PC$ and $DE = PC$, then $K = PC$;
  ...
- $\mathcal{R}^{(15)}$: if $E = PM$ and $DE = PM$, then $K = PM$;
  ...
- $\mathcal{R}^{(22)}$: if $E = O$ and $DE = O$, then $K = O$;
  ...
- $\mathcal{R}^{(36)}$: if $E = NM$ and $= NM$, then $K = NM$;
  ...
- $\mathcal{R}^{(43)}$: if $E = NC$ and $DE = NC$, then $K = NC$;
  ...
- $\mathcal{R}^{(49)}$: if $E = NL$ and $DE = NL$, then $K = NL$.

The complete rules of the binary channels fuzzy self-adjusted controller are shown in Table I. On account of fuzzy rules $\mathcal{R}^{(l)}, l = 1, 2, 3...49$, for the input fuzzy subsets E and DE, we combine elements ϵ from E and $\dot{\epsilon}$ from DE into an ordered pair (ϵ, $\dot{\epsilon}$). All of the ordered pairs form a new set called the direct product of E and DE, which is shown in Fig. 2 and can be described as

$$M = E \times DE = \{(\epsilon, \dot{\epsilon}) \mid \epsilon \in E, \dot{\epsilon} \in DE\} \tag{8}$$

Then the following relation is obtained

$$K = M \circ \mathcal{R} \tag{9}$$

where $\circ$ representing fuzzy synthesis operation, and $\mathcal{R} = \mathcal{R}^{(1)} \cup \mathcal{R}^{(2)} \cup \mathcal{R}^{(3)} \cup ... \mathcal{R}^{(49)}$. Further a fuzzy implication function (fuzzy relation) can be obtained

$$\mu_{M\circ\mathcal{R}}(\kappa) = \mu_{M\circ\mathcal{R}^{(1)}} \vee \mu_{M\circ\mathcal{R}^{(2)}} \vee \mu_{M\circ\mathcal{R}^{(3)}} \vee ... \mu_{M\circ\mathcal{R}^{(49)}} \tag{10}$$

where

$$\mu_{M\circ\mathcal{R}^{(l)}} = sup(\mu_{E^{(l)}} \wedge \mu_{DE^{(l)}} \wedge \mu_{K^{(l)}}), l = 1, 2, 3...49 \tag{11}$$

with $\vee$ and $\wedge$ representing the maximum or minimum value operation, i.e.,

$$\begin{aligned} \mu_X \vee \mu_Y &= max\{\mu_X, \mu_Y\} \\ \mu_X \wedge \mu_Y &= min\{\mu_X, \mu_Y\} \end{aligned} \tag{12}$$

sup denotes the supremum of a set.

TABLE I
FUZZY RULES WITH THE FUZZY DOMAIN
$(NL, NC, NM, O, PM, PC, PL)$.

Rule No.	E	DE	K	Rule No.	E	DE	K
1	NL	NL	PL	26	O	PM	NM
2	NL	NC	PL	27	O	PC	NC
3	NL	NM	PC	28	O	PL	NC
4	NL	O	PC	29	PM	NL	PM
5	NL	PM	PM	30	PM	NC	PM
6	NL	PC	O	31	PM	NM	O
7	NL	PL	O	32	PM	O	NM
8	NC	NL	PL	33	PM	PM	NM
9	NC	NC	PL	34	PM	PC	NC
10	NC	NM	PC	35	PM	PL	NC
11	NC	O	PM	36	PC	NL	O
12	NC	PM	PM	37	PC	NC	O
13	NC	PC	O	38	PC	NM	NM
14	NC	PL	O	39	PC	O	NM
15	NM	NL	PC	40	PC	PM	NC
16	NM	NC	PC	41	PC	PC	NC
17	NM	NM	PM	42	PC	PL	NL
18	NM	O	PM	43	PL	NL	O
19	NM	PM	O	44	PL	NC	NM
20	NM	PC	NM	45	PL	NM	NM
21	NM	PL	NC	46	PL	O	NC
22	O	NL	PC	47	PL	PM	NC
23	O	NC	PM	48	PL	PC	NL
24	O	NM	PM	49	PL	PL	NL
25	O	O	O				

3) **Defuzzification.** In order to obtain the binary channels fuzzy self-adjusted controller's output κ, the defuzzification need to be operated. The following are some common methods of defuzzification.

a) Maximal membership method, i.e.,

$$\kappa_0 = \text{arg max}\mu_{E\circ\mathcal{R}}(\kappa). \tag{13}$$

b) Center-of-gravity method, i.e.,

$$\kappa_0 = \frac{\int_K \kappa\mu_{E\circ\mathcal{R}}(\kappa)\mathrm{d}\kappa}{\int_K \mu_{E\circ\mathcal{R}}(\kappa)\mathrm{d}\kappa}. \tag{14}$$

c) Weighted average method, i.e.,

$$\kappa_0 = \frac{\sum_{i=1}^{\sigma} \kappa_i a_i}{\sum_{i=1}^{\sigma} a_i}. \tag{15}$$

After applying the binary channels fuzzy self-adjusted controller to the time-changing neural network, a BCF-SANN is designed to solve time-changing QP problems.

III. CONVERGENCE AND ROBUSTNESS ANALYSIS

In this section, several activation functions are used to analyze the properties of the proposed BCF-SANN model.

A. Convergence Analysis

To ensure that BCF-SANN (7) can solve the time-changing QP problem (1)-(2), the BCF-SANN must be shown to be convergent. The following is the convergence proof of BCF-SANN.

Theorem 1. *Considering the smoothly time-changing strictly-convex QP problem (1)-(2), when a monotonically-increasing odd function is used as a activation function array $\boldsymbol{\Theta}(\cdot)$, state vectors $x = [y^T, \eta^T] \in \mathbb{R}^{\varrho+\sigma}$ of the BCFT-SANN (7) will globally converge to the theory resolution $x^* = [y^{*T}, \eta^{*T}] \in \mathbb{R}^{\varrho+\sigma}$ from any initial state $x(0) = [y^T(0), \eta^T(0)]$. It is worth pointing out that the solution to (1)-(2) is found in the first ϱ elements of x^*.*

Proof: According to Lyapunov theorem of stability, a Lyapunov function candidate is defined as

$$d = \frac{\|\epsilon\|_2^2}{2} = \frac{\epsilon^{\mathrm{T}}\epsilon}{2} \geq 0 \tag{16}$$

where ϵ represents the error vector $\epsilon = Px - u$, $\|\cdot\|_2$ represents a vector's two norm. According to the formula (6), the time derivative of formula (16) is written as

$$\begin{aligned} \dot{d}_{BC} &= \epsilon^{\mathrm{T}} \frac{\mathrm{d}\epsilon}{\mathrm{d}\tau} \\ &= -(\nu+\kappa)\epsilon^{\mathrm{T}}\boldsymbol{\Theta}(\epsilon) \\ &= -(\nu+\kappa)\sum_{i=1}^{\varrho+\sigma} \epsilon_i \mathit{\Theta}(\epsilon_i) \end{aligned} \tag{17}$$

where ϵ_i represents the ith element of ϵ. $\boldsymbol{\Theta}(\cdot) : \mathbb{R}^{\varrho+\sigma} \to \mathbb{R}^{\varrho+\sigma}$ represents a monotonically-increasing odd activation-function, and $\mathit{\Theta}(\cdot)$ represents a processing-unit of array $\boldsymbol{\Theta}(\cdot)$. $\dot{d}_{BC}$ represents the differential of d when BCF-SANN is applied. By using function $\mathit{\Theta}(\epsilon_i)$ as follow, i.e.,

$$\epsilon_i \mathit{\Theta}(\epsilon_i) = \begin{cases} > 0, & \text{if} \quad \epsilon_i \neq 0 \\ = 0, & \text{if} \quad \epsilon_i = 0. \end{cases}$$

Since the convergence coefficient consists of the constant parameter $\nu > 0$ and the binary channels fuzzy self-adjusted controller's output $\kappa > 0$. It is obtained that $\dot{d}_{BC}$ is nonpositive when $t \in [0, +\infty)$. The Lyapunov stability theorem states that if a dynamic system has a tendency toward stability, the state vector x will eventually converge to the theory resolution x^*, and the error vector ϵ will progressively converge to zero from any starting state $x(0)$. Therefore, $x - x^*$ converges to 0 globally, and correspondingly, $y - y^*$ converges to 0. The proof is finished. ■

Theorem 2. *Take the time-changing linear system (4) into consideration. The state vectors $x = [y^T, \eta^T] \in \mathbb{R}^{\varrho+\sigma}$ of the BCF-SANN (7) converge to the theory resolution $x^* = [y^{*T}, \eta^{*T}] \in \mathbb{R}^{\varrho+\sigma}$ with the the following monotonically growing odd function array $\boldsymbol{\Theta}(\cdot)$ as the activation functions, i.e.,*

1) *Linear type*

$$\boldsymbol{\Theta}(\epsilon_i) = \epsilon_i. \tag{18}$$

2) *Power type*

$$\boldsymbol{\Theta}(\epsilon_i) = \epsilon_i^{\rho} \tag{19}$$

where $\rho > 1$, and ρ is odd, $|\ \epsilon_i\ | \geq 1$.

3) *Bipolar sigmoid type*

$$\boldsymbol{\Theta}(\epsilon_i) = \frac{(1+e^{-\omega})(1-e^{-\omega\epsilon_i})}{(1-e^{-\omega})(1+e^{-\omega\epsilon_i})} \tag{20}$$

where $\omega \geq 2$, and $0 \leq |\ \epsilon_i\ | \leq 1$.

4) *Sinh type*

$$\boldsymbol{\Theta}(\epsilon_i) = \frac{e^{\epsilon_i} - e^{-\epsilon_i}}{2}. \tag{21}$$

Proof:

1) When linear type activation function (18) is used as the activation function , the ith scalar-type formula of BCF-SANN is expressed as

$$\frac{\mathrm{d}\epsilon_i}{\mathrm{d}t} = -(\nu+\kappa)\,\boldsymbol{\Theta}(\epsilon_i) \tag{22}$$

This coefficient κ of the equation (22) has a certain range, with the minimum value and the maximum value which are set by designers according to specific situations. Using the method for solving differential equation, we can get the error function

$$\epsilon_i(t) = \epsilon_i(0)\, e^{-(\nu+\kappa)t} \tag{23}$$

where $i = 1, 2, ..., (\varrho + \sigma)$. The solution in vector form is

$$\epsilon(t) = \epsilon(0)\, e^{-(\nu+\kappa)t} \tag{24}$$

In the case of $\nu > 0$, κ denotes the output of the binary channels fuzzy self-adjusted controller and $\kappa \geq 0$, when $t \to +\infty$, $\epsilon(t) \to 0$.

2) When power type activation function (19) is used as the activation function, where $\rho > 1$ and ρ is odd, according to the modes of BCF-SANN, we have

$$\frac{\mathrm{d}\epsilon_i}{\mathrm{d}t} = -(\nu+\kappa)\,\epsilon_i^{\rho} \tag{25}$$

By using the theory of differential equation, we get

$$\epsilon_i(t) = \left[-(\nu+\kappa)(1-\rho)\,t + \epsilon_i^{1-\rho}(0)\right]^{\frac{1}{1-\rho}} \tag{26}$$

Since $\rho > 1$, we have $1/(1-\rho) < 0$, $\nu > 0$, κ denotes the output of the binary channels fuzzy self-adjusted controller and $\kappa \geq 0$. When $t \to +\infty$, $-(\nu+\kappa)(1-\rho)\,t \to +\infty$. Thus with $1/(1-\rho) < 0$, each element of $\epsilon_i(t)$ will converge to 0, which makes $\epsilon(t) \to 0$.

3) When bipolar sigmoid type activation function (20) is used as the activation function, according to the model of BCF-SANN, we have

$$\frac{\mathrm{d}\epsilon_i}{\mathrm{d}t} = -(\nu+\kappa)\,\frac{(1+e^{-\omega})(1-e^{-\omega\epsilon_i})}{(1-e^{-\omega})(1+e^{-\omega\epsilon_i})} \tag{27}$$

To simplify, we get

$$\frac{\mathrm{d}\epsilon_i}{\mathrm{d}t} = -(\nu+\kappa)\varpi\frac{1-e^{-\omega\epsilon_i}}{1+e^{-\omega\epsilon_i}} \tag{28}$$

where $\varpi = \frac{1+e^{-\omega}}{1-e^{-\omega}}$. According to function (20), $\omega \geq 2$, and $0 \leq |\ \epsilon_i\ | \leq 1$, by using the theory of differential equation, we get

$$\epsilon_i = -\frac{2}{\omega}\times \ln\left(\frac{-e^{-\omega((\nu+\kappa)\varpi t+c)/2}+\sqrt{4+e^{-\omega((\nu+\kappa)\varpi t+c)}}}{2}\right) \tag{29}$$

where

$$c = -\epsilon_i\left(0\right) - \frac{2}{\omega}(1-e^{-\omega\epsilon_i(0)})$$

We set $p_i = e^{-\omega((\nu+\kappa)\varpi t+c)/2}$, the function ϵ_i can be simplified to

$$\epsilon_i = -\frac{2}{\omega}\ln\left(\frac{-p_i+\sqrt{4+p_i^2}}{2}\right) \tag{30}$$

When $t \to \infty$, $p_i = e^{-\omega((\nu+\kappa)\varpi t+c)/2} \to 0$, each element of ϵ_i will converge to 0, which makes $\epsilon \to 0$.

4) When sinh type function (21) is used as the activation function, according to the modes of BCF-SANN, we have

$$\frac{\mathrm{d}\epsilon_i}{\mathrm{d}t} = -(\nu+\kappa)\frac{e^{\epsilon_i}-e^{-\epsilon_i}}{2} \tag{31}$$

Based on the differential equation theory, in the case of $\epsilon_i > 0$, the solution is

$$\begin{aligned}\epsilon_i &= \ln\frac{1+e^{-(\nu+\kappa)t+\gamma}}{1-e^{-(\nu+\kappa)t+\gamma}}\\ &= \ln(1+\frac{2}{e^{(\nu+\kappa)t+\gamma}-1})\end{aligned} \tag{32}$$

where $\gamma = \ln\frac{\exp\epsilon_i(0)-1}{\exp\epsilon_i(0)+1}$. When $t \to \infty$, $\epsilon_i \to 0$.

■

B. Robustness Analysis

The existence of hardware implementation errors and external perturbations in real-world applications is well acknowledged. The disturbed BCF-SANN model can be expressed mathematically as follows, in accordance with (7).

$$\begin{aligned}P\dot{x} = &-(\dot{P}+\Delta\mathcal{D})x-(\nu+\kappa)\mathbf{\Theta}(Px\\ &-u)+\dot{u}+\Delta\mathcal{S}\end{aligned} \tag{33}$$

where $\Delta\mathcal{D} \in \mathbb{R}^{(\varrho+\sigma)\times(\varrho+\sigma)}$ represents different hardware implementation defects of matrix P, $\Delta\mathcal{S} \in \mathbb{R}^{(\varrho+\sigma)}$ represents external perturbation errors of vector u.

Theorem 3. *If $\|\triangle\mathcal{D}\|_F \leqslant \theta_{\mathcal{D}} \in R$, $\|\triangle\mathcal{S}\|_2 \leqslant \theta_{\mathcal{S}} \in R$, $\left\|P^{-1}\right\|_F \leqslant \varepsilon_P \in R$, $\|u\|_2 \leqslant \varepsilon_u \in R$, for $t \in [0,+\infty)$ and $0 < \theta_{\mathcal{D}},\theta_{\mathcal{S}},\varepsilon_P,\varepsilon_u < +\infty$, then the upper bound of the absolute value of the error $\epsilon = Px-u$ in the perturbed BCF-SANN (33) with linear type activation function is $\frac{\varphi}{2}\left(1+\sqrt{\varrho+\sigma}\right)$, where $\varphi = (\theta_{\mathcal{D}}\varepsilon_P\varepsilon_u+\theta_{\mathcal{S}})/(\nu+\kappa-\theta_{\mathcal{D}}\varepsilon_P)$ and $\nu+\kappa > \theta_{\mathcal{D}}\varepsilon_P$, it will converge to 0 as $t \to \infty$ if the value κ that is the output of the binary channels fuzzy self-adjusted controller is large enough.*

Proof: As defined previously, the time derivative of the error function $\epsilon = Px-u$ can be written as $\dot{\epsilon} = \dot{P}x + P\dot{x} - \dot{u}$. Combined with (33), the perturbed error function can be rewritten as

$$\begin{aligned}\dot{\epsilon} = &-\left(\nu+\kappa\right)\mathbf{\Theta}\left(\epsilon\right)+\Delta\mathcal{S}-\Delta\mathcal{D}P^{-1}\\ &\times\epsilon-\Delta\mathcal{D}P^{-1}u.\end{aligned} \tag{34}$$

To verify the proposed BCF-SANN's robustness, a Lyapunov function candidate is defined as follow, ie.,

$$d = \frac{\|\epsilon\|_2^2}{2} = \frac{\epsilon^{\mathrm{T}}\epsilon}{2} = \sum_{i=1}^{\varrho+\sigma}\epsilon_i^2/2 \geq 0 \tag{35}$$

where $d > 0$ if $\epsilon \neq 0$, and $d = 0$ only if $\epsilon = 0$.

Considering (34), the time derivative of the Lyapunov function (35) is obtained as

$$\begin{aligned}\dot{d} &= \epsilon^{\mathrm{T}}\dot{\epsilon}\\ &= -(\nu+\kappa)\epsilon^{\mathrm{T}}\mathbf{\Theta}\left(\epsilon\right)+\epsilon^{\mathrm{T}}\Delta\mathcal{S}+\\ &\epsilon^{\mathrm{T}}\mathcal{G}\epsilon-\epsilon^{\mathrm{T}}\Delta\mathcal{D}P^{-1}u\\ &= -\left(\nu+\kappa\right)\epsilon^{\mathrm{T}}\mathbf{\Theta}\left(\epsilon\right)+\epsilon^{\mathrm{T}}\Delta\mathcal{S}+\\ &\epsilon^{\mathrm{T}}\frac{\mathcal{G}+\mathcal{G}^{\mathrm{T}}}{2}\epsilon-\epsilon^{\mathrm{T}}\Delta\mathcal{D}P^{-1}u\end{aligned} \tag{36}$$

where $\mathcal{G} = -\Delta\mathcal{D}P^{-1}$ and $\mathcal{G}$ is a positive definite symmetric matrix.

As is known to all, given that every element of a vector is either less than or equal to its maximum, and combined with the inequality $\underset{1\leqslant i\leqslant\varrho+\sigma}{max}|\Delta\mathcal{S}_i| \leqslant \|\Delta\mathcal{S}\|_2$, the second term of the equation (36) $\epsilon^{\mathrm{T}}\Delta\mathcal{S}$ can be rewritten as

$$\begin{aligned}\epsilon^{\mathrm{T}}\Delta\mathcal{S} &\leqslant \sum_{i=1}^{\varrho+\sigma}|\epsilon_i|\underset{1\leqslant i\leqslant\varrho+\sigma}{max}|\Delta\mathcal{S}_i|\\ &\leqslant \sum_{i=1}^{\varrho+\sigma}|\epsilon_i|\,\|\Delta\mathcal{S}\|_2\\ &\leqslant \sum_{i=1}^{\varrho+\sigma}|\epsilon_i|\,\theta_{\mathcal{S}}.\end{aligned} \tag{37}$$

Considering the third term of (36), it follows from the inequality $\max_{1\leqslant i\leqslant\varrho+\sigma}|\lambda_i(P)| \leqslant \|P\|_F$ and $\|\beta\chi\|_F \leqslant \|\beta\|_F\,\|\chi\|_F$, the following inequality is obtained, i.e.,

$$\begin{aligned}&\epsilon^{\mathrm{T}}\frac{\mathcal{G}+\mathcal{G}^{\mathrm{T}}}{2}\epsilon\\ &\leqslant \epsilon^{\mathrm{T}}\epsilon\underset{1\leqslant i\leqslant\varrho+\sigma}{max}\left|\lambda_i\left(\frac{\mathcal{G}+\mathcal{G}^{\mathrm{T}}}{2}\right)\right|\\ &= \epsilon^{\mathrm{T}}\epsilon\underset{1\leqslant i\leqslant\varrho+\sigma}{max}|\lambda_i(\frac{\Delta\mathcal{D}P^{-1}+(\Delta\mathcal{D}P^{-1})^{\mathrm{T}}}{2})|\\ &\leqslant \epsilon^{\mathrm{T}}\epsilon\left\|\frac{\Delta\mathcal{D}P^{-1}+(\Delta\mathcal{D}P^{-1})^{\mathrm{T}}}{2}\right\|_F\\ &\leqslant \epsilon^{\mathrm{T}}\epsilon\left\|\Delta\mathcal{D}P^{-1}\right\|_F\\ &\leqslant \epsilon^{\mathrm{T}}\epsilon\,\|\Delta\mathcal{D}\|_F\left\|P^{-1}\right\|_F\\ &\leqslant \epsilon^{\mathrm{T}}\epsilon\theta_{\mathcal{D}}\varepsilon_P.\end{aligned} \tag{38}$$

Similarly, based on the inequality $\|\beta\chi\|_2 \leqslant \|\beta\|_F\,\|\chi\|_2$, the fourth term of the equation (36) $\epsilon^{\mathrm{T}}\left(-\Delta\mathcal{D}P^{-1}u\right)$ can be

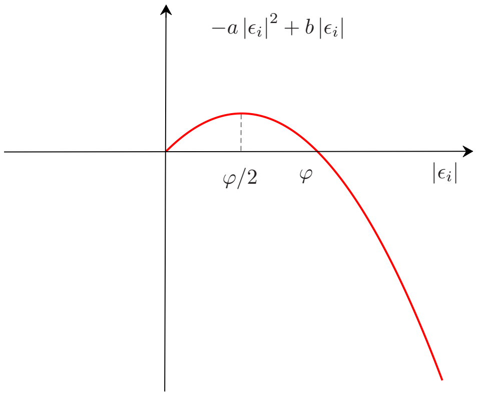


Fig. 3. The function curve of a dependent variable $a\left|\epsilon_i\right|^2-b\left|\epsilon_i\right|$ with respect to an independent variable $\left|\epsilon_i\right|$.

rewritten as

$$\begin{aligned}
&\epsilon^{\mathrm{T}}\left(-\Delta\mathcal{D}P^{-1}u\right)\\
&\leqslant\sum_{i=1}^{\varrho+\sigma}\left|\epsilon_i\right|\underset{1\leqslant i\leqslant\varrho+\sigma}{max}\left|\left[\Delta\mathcal{D}P^{-1}u\right]_i\right|\\
&\leqslant\sum_{i=1}^{\varrho+\sigma}\left|\epsilon_i\right|\cdot\left\|\Delta\mathcal{D}P^{-1}u\right\|_2\\
&\leqslant\sum_{i=1}^{\varrho+\sigma}\left|\epsilon_i\right|\cdot\left\|\Delta\mathcal{D}P^{-1}\right\|_F\left\|u\right\|_2\\
&\leqslant\sum_{i=1}^{\varrho+\sigma}\left|\epsilon_i\right|\cdot\left\|\Delta\mathcal{D}\right\|_F\left\|P^{-1}\right\|_F\left\|u\right\|_2\\
&\leqslant\sum_{i=1}^{\varrho+\sigma}\left|\epsilon_i\right|\cdot\theta_{\mathcal{D}\varepsilon_P\varepsilon_u}.
\end{aligned}\tag{39}$$

Substitute (38), (37), and(39) into (36), we can get

$$\begin{aligned}
&\dot{d}\leqslant-\left(\nu+\kappa\right)\epsilon^{\mathrm{T}}\boldsymbol{\Theta}\left(\epsilon\right)+\epsilon^{\mathrm{T}}\epsilon\theta_{\mathcal{D}\varepsilon_P}+\\
&\sum_{i=1}^{\varrho+\sigma}\left|\epsilon_i\right|\theta_{\mathcal{S}}+\sum_{i=1}^{\varrho+\sigma}\left|\epsilon_i\right|\theta_{\mathcal{D}\varepsilon_P\varepsilon_u}\\
&=-\sum_{i=1}^{\varrho+\sigma}\left|\epsilon_i\right|\times\\
&\left(\left(\nu+\kappa\right)\varTheta\left(\left|\epsilon_i\right|\right)-\theta_{\mathcal{D}\varepsilon_P}\left|\epsilon_i\right|-\theta_{\mathcal{S}}-\theta_{\mathcal{D}\varepsilon_P\varepsilon_u}\right).
\end{aligned}\tag{40}$$

To better analyses the function $\dot{d}$, we set the function

$$\Omega\left(\epsilon_i\right)=\left(\nu+\kappa\right)\varTheta\left(\left|\epsilon_i\right|\right)-\theta_{\mathcal{D}\varepsilon_P}\left|\epsilon_i\right|-\theta_{\mathcal{S}}-\theta_{\mathcal{D}\varepsilon_P\varepsilon_u}$$

There are two situations should be analyzed, i.e.,

1) If $\Omega\left(\epsilon_i\right)\geqslant 0$ for $\forall i=1,2,...,\varrho+\sigma$, then $d\geqslant 0$ and $\dot{d}\leqslant 0$. According to Lyapunov's stability theorem, the error function ϵ can converge to zero, which means the state solution x can converge to the theory resolution x^*.
2) If $\Omega\left(\epsilon_i\right)<0$ for $\exists i=1,2,...,\varrho+\sigma$, the right side of (40) is positive, thus $\dot{d}$ has a positive upper bound. There exist two conditions, i.e., $\dot{d}\leqslant 0$ or $\dot{d}>0$.
   a) If $\dot{d}\leqslant 0$, as $d\geqslant 0$, then according to the Lyapunov theorem, the error function ϵ can converge to zero, and the system can reach the steady state.
   b) If $\dot{d}>0$, by using the linear type activation function $\varTheta\left(\left|\epsilon_i\right|\right)=\left|\epsilon_i\right|$, (40) is rewritten as

$$\begin{aligned}
&\dot{d}\leqslant-\sum_{i=1}^{\varrho+\sigma}\left|\epsilon_i\right|\times\\
&\left(\left(\nu+\kappa\right)\left|\epsilon_i\right|-\theta_{\mathcal{D}\varepsilon_P}\left|\epsilon_i\right|-\theta_{\mathcal{S}}-\theta_{\mathcal{D}\varepsilon_P\varepsilon_u}\right)\\
&=-\left(\left(\nu+\kappa\right)-\theta_{\mathcal{D}\varepsilon_P}\right)\sum_{i=1}^{\varrho+\sigma}\left|\epsilon_i\right|\times\\
&(\left|\epsilon_i\right|-\frac{\theta_{\mathcal{D}\varepsilon_P\varepsilon_u}+\theta_{\mathcal{S}}}{\nu+\kappa-\theta_{\mathcal{D}\varepsilon_P}})\\
&=-\left(\left(\nu+\kappa\right)-\theta_{\mathcal{D}\varepsilon_P}\right)\sum_{i=1}^{\varrho+\sigma}\left|\epsilon_i\right|^2+\\
&\left(\theta_{\mathcal{D}\varepsilon_P\varepsilon_u}+\theta_{\mathcal{S}}\right)\sum_{i=1}^{\varrho+\sigma}\left|\epsilon_i\right|\\
&=\sum_{i=1}^{\varrho+\sigma}-((\nu+\kappa)-\theta_{\mathcal{D}\varepsilon_P})|\epsilon_i|^2+\\
&(\theta_{\mathcal{D}\varepsilon_P\varepsilon_u}+\theta_{\mathcal{S}})|\epsilon_i|\\
&=\sum_{i=1}^{\varrho+\sigma}\left(-a\left|\epsilon_i\right|^2+b\left|\epsilon_i\right|\right)
\end{aligned}\tag{41}$$

where $a=\nu+\kappa-\theta_{\mathcal{D}\varepsilon_P}$, $b=\theta_{\mathcal{D}\varepsilon_P\varepsilon_u}+\theta_{\mathcal{S}}$. In the case of $a>0$, which means $\nu+\kappa>\theta_{\mathcal{D}\varepsilon_P}$. For the convenience of discussion, we draw the function model of ith element of the right side of (41) in Fig. 3. In (41), considering $\dot{d}>0$, d will increase, which result in an increase in $|\epsilon_i|$. As $|\epsilon_i|$ increases, the upper bound of $\dot{d}$ first increase to the maximum value and then keep decreasing until it is less than 0. So it can be inferred that d first monotonically increases to a maximum value and then monotonically decreases. Therefore, there must exists a certain moment when $\dot{d}<0$, the system is stable again. Thus, there must have an upper bound of $|\epsilon_i|$, such that $\dot{d}=0$. Assuming $\forall i=1,2,...,(\varrho+\sigma-1)$,

$$|\epsilon_i|=\frac{\theta_{\mathcal{D}\varepsilon_P\varepsilon_u}+\theta_{\mathcal{S}}}{\nu+\kappa-\theta_{\mathcal{D}\varepsilon_P}}\tag{42}$$

in the case of $\nu+\kappa>\theta_{\mathcal{D}\varepsilon_P}$, the summation of the former $\varrho+\sigma-1$ elements of the (41) can achieve the maximum value. When $\dot{d}=0$, we have

$$\begin{aligned}
&\sum_{i=1}^{\varrho+\sigma}|\epsilon_i|(|\epsilon_i|-\frac{\theta_{\mathcal{D}\varepsilon_P\varepsilon_u}+\theta_{\mathcal{S}}}{\nu+\kappa-\theta_{\mathcal{D}\varepsilon_P}})\\
&=|\epsilon_j|(|\epsilon_j|-\frac{\theta_{\mathcal{D}\varepsilon_P\varepsilon_u}+\theta_{\mathcal{S}}}{\nu+\kappa-\theta_{\mathcal{D}\varepsilon_P}})+\\
&\sum_{i=1,i\neq j}^{\varrho+\sigma}|\epsilon_i|(|\epsilon_i|-\frac{\theta_{\mathcal{D}\varepsilon_P\varepsilon_u}+\theta_{\mathcal{S}}}{\nu+\kappa-\theta_{\mathcal{D}\varepsilon_P}})\\
&=|\epsilon_j|^2-|\epsilon_j|\frac{\theta_{\mathcal{D}\varepsilon_P\varepsilon_u}+\theta_{\mathcal{S}}}{\nu+\kappa-\theta_{\mathcal{D}\varepsilon_P}}-\\
&\frac{\varrho+\sigma-1}{4}(\frac{\theta_{\mathcal{D}\varepsilon_P\varepsilon_u}+\theta_{\mathcal{S}}}{\nu+\kappa-\theta_{\mathcal{D}\varepsilon_P}})^2=0
\end{aligned}\tag{43}$$

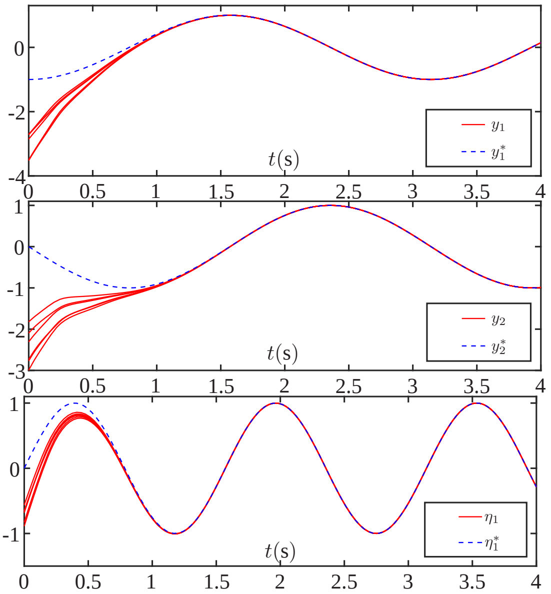


Fig. 4. Simulations of state vector $x = [y_1, y_2, \eta_1]^{\mathrm{T}}$ from six random initial states and theoretical solutions $x^* = \left[y_1^*, y_2^*, \eta_1^*\right]^{\mathrm{T}}$ when BCF-SANN is used to solve the time-changing QP (44) problems.

By solving (43), $|\epsilon_j| = \frac{\theta_{\mathcal{D}}\varepsilon_P\varepsilon_u+\theta_{\mathcal{S}}}{2(\nu+\kappa-\theta_{\mathcal{D}}\varepsilon_P)}(1+\sqrt{\varrho+\sigma})$. Thus the upper bound of $|\epsilon_j|$ can be reached when $i = j$. If the value κ that is the output of the binary channels fuzzy self-adjusted controller is large enough, $|\epsilon_j|$ will converge to 0. In other words, when $\dot{d} > 0$, the upper bound of the error converges to 0 with $t \to \infty$, the system is stable, the state vector x can converge to the theory resolution x^*.

■

IV. Illustrative Examples

This section presents the solution of the proposed BCF-SANN to a realistic time-changing QP problem. The simulation results of the ZNN (Zeroing Neural Network) and AFRNN (Adaptive Fuzzy Recurrent Neural Network) solutions to the identical time-changing QP problem are also provided for comparisons.

A. Convergence Verification

We use the following time-changing QP problem as an example to verify the convergence of the proposed BCF-SANN, i.e.,

$$\begin{aligned}\min.\ &(\cos t/2+1)\,y_1^2+(\cos t/2+1)\,y_2^2+\\&\sin 2ty_1y_2+\cos 2ty_1+\sin 2ty_2\\ \mathrm{s.\,t.}\ &\cos 3ty_1+\sin 3ty_2=\sin 4t\end{aligned} \tag{44}$$

The time-changing QP problem is rewritten as the standard tima-changing QP problem like (1) and (2) and whose parameters are defined as

$$B=\begin{bmatrix}\cos t+2 & \sin 2t\\ \sin 2t & \cos t+2\end{bmatrix},\quad c=\begin{bmatrix}\cos 2t\\ \sin 2t\end{bmatrix},$$

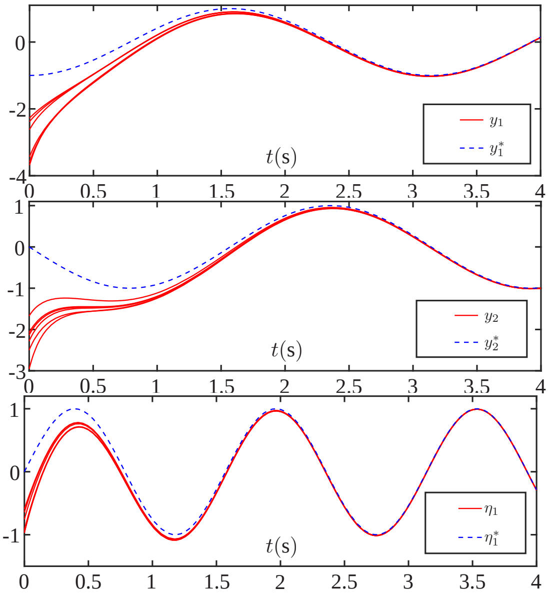


Fig. 5. Simulations of state vector $x = [y_1, y_2, \eta_1]^{\mathrm{T}}$ from six random initial states and theoretical solutions $x^* = \left[y_1^*, y_2^*, \eta_1^*\right]^{\mathrm{T}}$ when AFRNN is used to solve the time-changing QP (44) problems.

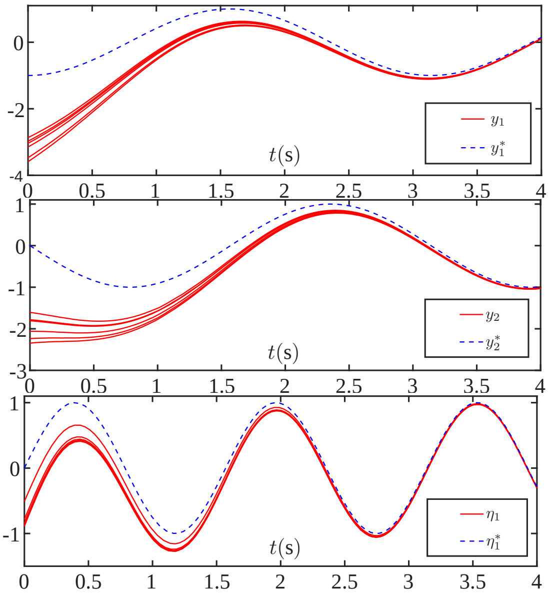


Fig. 6. Simulations of state vector $x = [y_1, y_2, \eta_1]^{\mathrm{T}}$ from six random initial states and theoretical solutions $x^* = \left[y_1^*, y_2^*, \eta_1^*\right]^{\mathrm{T}}$ when ZNN is used to solve the tima-changing QP (44) problems.

$$J=\left[\cos 3t\quad \sin 3t\right],\quad b=\sin 4t,$$

$$y=[y_1,y_2]^{\mathrm{T}}. \tag{45}$$

According to Section II, the time-changing QP problem can be rewritten as

$$Px=u \tag{46}$$

where

$$P=\begin{bmatrix}\cos t+2 & \sin 2t & \cos 3t\\ \sin 2t & \cos t+2 & \sin 3t\\ \cos 3t & \sin 3t & 0\end{bmatrix},$$

$$u = [-\cos 2t, -\sin 2t, \sin 4t]^{\mathrm{T}},$$

$$x = [y_1, y_2, \eta_1]^{\mathrm{T}}.$$

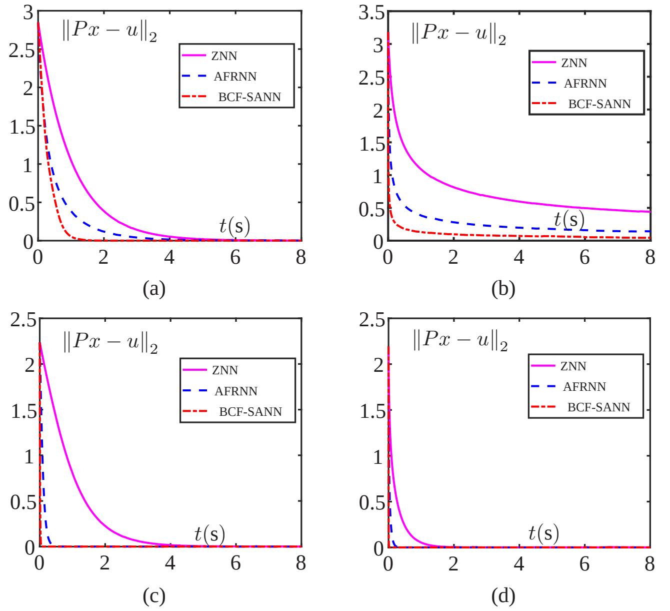


Fig. 7. The residual error $\|Px-u\|_2$ of BCF-SANN, ZNN and AFRNN. (a) Using linear type activation. (b) Using power type activation. (c) Using bipolar function. (d) Using sinh function.

First of all, using the same linear type activation function, the time-changing QP (44) is solved by BCF-SANN, AFRNN and ZNN respectively. The simulative curves of the state vector $x = [y_1, y_2, \eta_1]^{\mathrm{T}}$ form BCF-SANN, AFRNN and ZNN are respectively shown in Figs. 4, 5 and 6. In Fig. 4, all the state variables y_1, y_2 and the Lagrange multiplier η_1 with six random initial states converge to the theory resolution $x^* = [y_1^*, y_2^*, \eta_1^*]^{\mathrm{T}}$ in less than 1.2 s. Conversely, Fig. 5 and Fig. 6 respectively shows the state variables y_1, y_2, and the vector of Lagrange-multiplier η_1 for AFRNN and ZNN costs more time to convergent to theory solution. In another word, the BCF-SANN has a higher convergence performance than ZNN and AFRNN for solving time-changing QP problems. The convergence performance is further demonstrated by tracking the residual errors $\|Px-u\|_2$ of BCF-SANN, AFRNN and ZNN when they solve the time-changing QP problem using a linear type activation function. We can observe in Fig. 7(a) that the BCF-SANN's residual errors reach zero earlier than the AFRNN's residual errors. Once again, BCF-SANN is shown to be superior in terms of convergence. Moreover, the BCF-SANN's residual error reaches zero in 1.6 seconds, but the ZNN's residual error takes 4 seconds to converge to zero. By using different activation functions, the simulation results about convergence of BCF-SANN are shown in Fig. 7. The preceding simulations lead to the conclusion that BCF-SANN has superior convergence efficiency and precision compared to AFRNN and ZNN.

B. Robustness Verification

This section presents a perturbed example to verify the robustness of the proposed BCF-SANN while solving time-changing QP problems. We design the systematic errors $\Delta\mathcal{D}$ and $\Delta\mathcal{S}$ in the mode of ZNN, AFRNN and the proposed BCF-SANN.

$$\Delta\mathcal{D} = \begin{bmatrix} 5\sin t + 2 & 4\cos t & -3\sin t \\ \cos 2t & 2\cos 2t & -3\cos 2t \\ \sin 3t & -\sin 4t & \cos 5t \end{bmatrix},$$

$$\Delta\mathcal{S} = \begin{bmatrix} 2\sin t \\ 4\cos 2t \\ -5\sin 3t \end{bmatrix}.$$

The simulations of the perturbed ZNN, AFRNN and BCF-SANN are shown in Fig. 8(a) with linear activation function, Fig. 8(b) with power activation function, Fig. 8(c) with bipolar sigmoid activation function and Fig. 8(d) with sinh activation function. From the simulative results, the robustness of BCF-SANN is stronger than that of AFRNN and ZNN. It is worth pointing out that the residual error $\|Px-u\|_2$ of disturbed BCF-SANN is always smaller than ZNN and AFRNN.

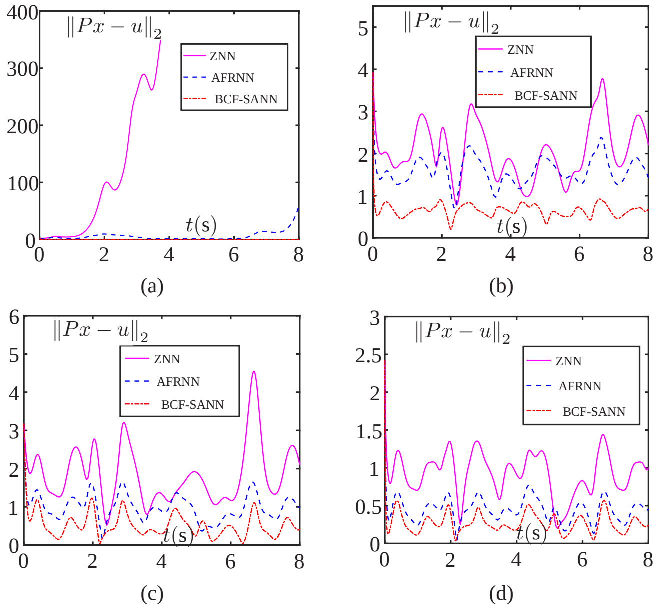


Fig. 8. The residual error $\|Px-u\|_2$ of disturbed ZNN, AFRNN and BCF-SANN with different type activation functions. (a) Using linear type activation function. (b) Using power type activation function. (c) Using bipolar sigmoid type activation function. (d) Using sinh type activation function.

V. Conclusions

In this paper, a novel binary channels fuzzy self-adjusted neural network (BCF-SANN) has been proposed and analyzed to solve time-changing QP problems. The design process of the BCF-SANN has been given in detail. In addition, the convergence and robustness of the proposed BCF-SANN with four different type activation functions have been proved in detail. With the self-adjustment ability of convergence coefficient, the BCF-SANN has faster convergent rate and stronger robustness than the traditional ZNN and AFRNN. Finally, comparative simulation results have further verified the fast convergence and stronger robustness of the proposed BCF-SANN.

REFERENCES

[1] Z. Zhang, Z. Li, Y. Zhang, Y. Luo, and Y. Li, "Neural-dynamic-method-based dual-arm cmg scheme with time-varying constraints applied to humanoid robots," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 26, no. 12, pp. 3251–3262, 2015.

[2] Y. Qi, L. Jin, Y. Wang, L. Xiao, and J. Zhang, "Complex-valued discrete-time neural dynamics for perturbed time-dependent complex quadratic programming with applications," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 31, no. 9, pp. 3555–3569, 2020.

[3] Y. Zhang, Z. Li, M. Yang, L. Ming, and J. Guo, "Jerk-level zhang neurodynamics equivalency of bound constraints, equation constraints, and objective indices for cyclic motion of robot-arm systems," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 34, no. 6, pp. 3005–3018, 2023.

[4] W. Li, P. W. Y. Chiu, and Z. Li, "A novel neural approach to infinity-norm joint-velocity minimization of kinematically redundant robots under joint limits," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 34, no. 1, pp. 409–420, 2023.

[5] Y. Xia and J. Wang, "Robust regression estimation based on low-dimensional recurrent neural networks," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 29, no. 12, pp. 5935–5946, 2018.

[6] J. Wang and J. Wang, "Two-timescale multilayer recurrent neural networks for nonlinear programming," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 33, no. 1, pp. 37–47, 2022.

[7] J. Wang and J. Wang, "Two-timescale multilayer recurrent neural networks for nonlinear programming," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 33, no. 1, pp. 37–47, 2022.

[8] Z. Cui, J. Li, W. Li, X. Zhang, P. W. Y. Chiu, and Z. Li, "Fast convergent antinoise dual neural network controller with adaptive gain for flexible endoscope robots," *IEEE Transactions on Neural Networks and Learning Systems*, pp. 1–14, 2022.

[9] Y. Qi, L. Jin, X. Luo, and M. Zhou, "Recurrent neural dynamics models for perturbed nonstationary quadratic programs: A control-theoretical perspective," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 33, no. 3, pp. 1216–1227, 2022.

[10] J. Wang, "Recurrent neural network for solving quadratic programming problems with equality constraints," *Electronics Letters*, vol. 28, no. 14, pp. 1345–1347, 1992.

[11] J. Wang, "A recurrent neural network for real-time matrix inversion," *Applied Mathematics and Computation*, vol. 55, no. 1, pp. 89–100, 1993.

[12] L. Xiao and Y. Zhang, "Zhang neural network versus gradient neural network for solving time-varying linear inequalities," *IEEE Transactions on Neural Networks*, vol. 22, no. 10, pp. 1676–1684, 2011.

[13] Yunong Zhang and S. S. Ge, "Design and analysis of a general recurrent neural network model for time-varying matrix inversion," *IEEE Transactions on Neural Networks*, vol. 16, no. 6, pp. 1477–1490, 2005.

[14] Y. Zhang, K. Chen, X. Li, C. Yi, and H. Zhu, "Simulink modeling and comparison of zhang neural networks and gradient neural networks for time-varying lyapunov equation solving," in *2008 Fourth International Conference on Natural Computation*, vol. 3, pp. 521–525, 2008.

[15] Y. Zhang and K. Chen, "Comparison on zhang neural network and gradient neural network for time-varying linear matrix equation axb = c solving," in *2008 IEEE International Conference on Industrial Technology*, pp. 1–6, 2008.

[16] Y. Zhang and Y. Yang, "Simulation and comparison of zhang neural network and gradient neural network solving for time-varying matrix square roots," in *2008 Second International Symposium on Intelligent Information Technology Application*, vol. 2, pp. 966–970, 2008.

[17] Z. Zhang, S. Li, and X. Zhang, "Simulink comparison of varying-parameter convergent-differential neural-network and gradient neural network for solving online linear time-varying equations," in *2016 12th World Congress on Intelligent Control and Automation (WCICA)*, pp. 887–894, 2016.

[18] Z. Zhang, Y. Lu, L. Zheng, S. Li, Z. Yu, and Y. Li, "A new varying-parameter convergent-differential neural-network for solving time-varying convex qp problem constrained by linear-equality," *IEEE Transactions on Automatic Control*, vol. 63, no. 12, pp. 4110–4125, 2018.

[19] Z. Zhang, L. Kong, and L. Zheng, "Power-type varying-parameter rnn for solving tvqp problems: Design, analysis, and applications," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 30, no. 8, pp. 2419–2433, 2019.

[20] Z. Zhang, L. Kong, L. Zheng, P. Zhang, X. Qu, B. Liao, and Z. Yu, "Robustness analysis of a power-type varying-parameter recurrent neural network for solving time-varying qm and qp problems and applications," *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, pp. 1–14, 2018.

[21] Z. Zhang, L. Zheng, T. Qiu, and F. Deng, "Varying-parameter convergent-differential neural solution to time-varying overdetermined system of linear equations," *IEEE Transactions on Automatic Control*, vol. 65, no. 2, pp. 874–881, 2020.

[22] Z. Zhang, T. Chen, M. Wang, and L. Zheng, "An exponential-type anti-noise varying-gain network for solving disturbed time-varying inversion systems," *IEEE Transactions on Neural Networks and Learning Systems*, pp. 1–14, 2019.

[23] Z. Zhang, L. Zheng, Z. Chen, L. Kong, and H. R. Karimi, "Mutual-collision-avoidance scheme synthesized by neural networks for dual redundant robot manipulators executing cooperative tasks," *IEEE Transactions on Neural Networks and Learning Systems*, pp. 1–15, 2020.

[24] Z. Zhang, B. Zhou, L. Zheng, Z. Zhang, C. Song, and H. Pei, "A varying-parameter adaptive multi-layer neural dynamic method for designing controllers and application to unmanned aerial vehicles," *IEEE Transactions on Intelligent Transportation Systems*, pp. 1–13, 2020.

[25] Z. Zhang and Z. Yan, "A varying parameter recurrent neural network for solving nonrepetitive motion problems of redundant robot manipulators," *IEEE Transactions on Control Systems Technology*, vol. 27, no. 6, pp. 2680–2687, 2019.

[26] Y. Luo, X. Deng, J. Wu, Y. Liu, and Z. Zhang, "A new finite-time circadian rhythms learning network for solving nonlinear and nonconvex optimization problems with periodic noises," *IEEE Transactions on Cybernetics*, vol. 52, no. 11, pp. 12514–12524, 2022.

[27] L. A. Zadeh, "Fuzzy sets," *Information and control*, vol. 8, no. 3, pp. 338–353, 1965.

[28] L. A. Zadeh, "Quantitative fuzzy semantics," in *Fuzzy Sets, Fuzzy Logic, And Fuzzy Systems: Selected Papers by Lotfi A Zadeh*, pp. 105–122, World Scientific, 1996.

[29] L. A. Zadeh, "Similarity relations and fuzzy orderings," *Information sciences*, vol. 3, no. 2, pp. 177–200, 1971.

[30] C. Y. Teh, Y. W. Kerk, K. M. Tay, and C. P. Lim, "On modeling of data-driven monotone zero-order tsk fuzzy inference systems using a system identification framework," *IEEE Transactions on Fuzzy Systems*, vol. 26, no. 6, pp. 3860–3874, 2018.

[31] J. Li, Q. Zhang, X. Yan, and S. K. Spurgeon, "Observer-based fuzzy integral sliding mode control for nonlinear descriptor systems," *IEEE Transactions on Fuzzy Systems*, vol. 26, no. 5, pp. 2818–2832, 2018.

[32] Y. Zhao, H. Gao, and J. Qiu, "Fuzzy observer based control for nonlinear coupled hyperbolic pde-ode systems," *IEEE Transactions on Fuzzy Systems*, vol. 27, no. 7, pp. 1332–1346, 2019.

[33] A. Wang, L. Liu, J. Qiu, and G. Feng, "Event-triggered robust adaptive fuzzy control for a class of nonlinear systems," *IEEE Transactions on Fuzzy Systems*, vol. 27, no. 8, pp. 1648–1658, 2019.

[34] Y. Zeng, H. Lam, and L. Wu, "Model reduction of discrete-time interval type-2 tcs fuzzy systems," *IEEE Transactions on Fuzzy Systems*, vol. 26, no. 6, pp. 3545–3554, 2018.

[35] H. Sun, R. Yu, Y. Chen, and H. Zhao, "Optimal design of robust control for fuzzy mechanical systems: Performance-based leakage and confidence-index measure," *IEEE Transactions on Fuzzy Systems*, vol. 27, no. 7, pp. 1441–1455, 2019.

[36] J. Li, S. Zhou, and S. Xu, "Fuzzy control system design via fuzzy lyapunov functions," *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, vol. 38, no. 6, pp. 1657–1661, 2008.

[37] R. Palm, "Sliding mode fuzzy control," in *[1992 Proceedings] IEEE International Conference on Fuzzy Systems*, pp. 519–526, 1992.

[38] C. C. Lee, "Fuzzy logic in control systems: fuzzy logic controller. i," *IEEE Transactions on Systems, Man, and Cybernetics*, vol. 20, no. 2, pp. 884–899, 1990.

[39] H. A. Malki, Huaidong Li, and Guanrong Chen, "New design and stability analysis of fuzzy proportional-derivative control systems," *IEEE Transactions on Fuzzy Systems*, vol. 2, no. 4, pp. 245–254, 1994.

[40] A.-A. Zamani, S. Tavakoli, and S. Etedali, "Control of piezoelectric friction dampers in smart base-isolated structures using self-tuning and adaptive fuzzy proportional–derivative controllers," *Journal of Intelligent Material Systems and Structures*, vol. 28, no. 10, pp. 1287–1302, 2017.

[41] Z. Zhang and Z. Yan, "An adaptive fuzzy recurrent neural network for solving non-repetitive motion problem of redundant robot manipulators," *IEEE Transactions on Fuzzy Systems*, pp. 1–1, 2019.