Article:

Jafari, R orcid.org/0000-0001-7298-2363 and Yu, W (2015) Fuzzy control for uncertainty nonlinear systems with dual fuzzy equations. Journal of Intelligent & Fuzzy Systems, 29 (3). pp. 1229-1240. ISSN 1064-1246

https://doi.org/10.3233/ifs-151731

Fuzzy control for uncertainty nonlinear systems with dual fuzzy equations

Raheleh Jafari and Wen Yu*
Departamento de Control Automático, CINVESTAV-IPN (National Polytechnic Institute), Mexico City, Mexico

Abstract. Many uncertain nonlinear systems can be modeled by the linear-in-parameter model, and the parameters are uncertain in the sense of fuzzy numbers. Fuzzy equations can be used to model these nonlinear systems. The solutions of the fuzzy equations are the controllers. In this paper, we give the controllability condition for the fuzzy control via dual fuzzy equations. Two types of neural networks are applied to approximate the solutions of the fuzzy equations. These solutions are then transformed into the fuzzy controllers. The novel methods are validated with five benchmark examples.

Keywords: Fuzzy equation, fuzzy number, fuzzy control

1. Introduction

Fuzzy control can be divided into direct and indirect methods [15]. The direct fuzzy uses a fuzzy system as a controller, while the indirect fuzzy control uses a fuzzy model to approximate the nonlinear system first, then a controller is designed based on the fuzzy model. The indirect fuzzy controller utilizes the simple topological structure and universal approximation ability of fuzzy model. It has been widely used in uncertain nonlinear system control. We use indirect fuzzy control in this paper.

The fuzzy model usually comes from several fuzzy rules [36]. These fuzzy rules represent the controlled nonlinear system. Since any nonlinear system can be approximated by several piecewise linear systems (Takagi-Sugeno fuzzy model) or known nonlinear systems (Mamdani fuzzy model) [23], fuzzy models can approximate a large class of nonlinear systems, while keep the simplicity of the linear models. In this paper, we discuss another type of fuzzy model. The basic idea is: many nonlinear systems can be expressed by linear-in-parameter models, such as Lagrangian mechanical systems [31]. The parameters of these models are uncertain, and the uncertainties satisfy the fuzzy set theory [41]. In this way, the inconvenience problems in nonlinear modeling, complexity and uncertainty, are solved by linear-in-parameter structure and fuzzy logic theory. The linear-in-parameter model with fuzzy parameters is called fuzzy equation [10, 14, 30].

Fuzzy equations are very simple compared with the normal system systems [23, 36]. They can applied directly for nonlinear control. The nonlinear system modeling corresponds to find the fuzzy parameters of the fuzzy equation, and the fuzzy control is to design suitable nonlinear functions in the fuzzy equation. Both fuzzy modeling and fuzzy control via fuzzy equations need solution of the fuzzy equation. There are various approaches [16] uses the parametric form of fuzzy numbers and replaced the original fuzzy equations by crisp linear systems. In [8], the extension principle is applied. The coefficients can be real or complex fuzzy numbers. However, the existence of the solution is not guaranteed [1] suggests the homeotypic analysis method [2] applies the Newton's method. In [4], the solution of fuzzy equations are obtained by the fixed point method. One of

*Corresponding author. Wen Yu, Departamento de Control Automático, CINVESTAV-IPN (National Polytechnic Institute), Mexico City, Mexico. Tel.: +52 55 57473734; Fax: +52 55 57473982; E-mail: yuw@ctrl.cinvestav.mx.

the well-known methods is the method of α-level [17]. By using the method of superimposition of sets, fuzzy numbers can be solved. Recently, fuzzy fractional differential and integral equations have been extensively studied in [3, 5, 24, 29, 37]. However, the above methods are very complex.

The numerical solution of the fuzzy equation can be obtained by iterative method [21], interpolation method [39], and Runge-Kutta method [28]. It can also applied to fuzzy differential equations [22]. These methods are also difficult to be applied. Both neural networks and fuzzy logic are universal estimators, they can approximate any nonlinear function to any prescribed accuracy, provided that sufficient hidden neurons and fuzzy rules are available [13]. Resent results show that the fusion procedure of these two different technologies seems to be very effective for nonlinear systems identification [40]. Neural networks can also be used to solve the fuzzy equation. In [9], the simple fuzzy quadratic equation is solved by a three neurons networks [19] and [20] extend the results of [9] into fuzzy polynomial equation. [26] gives a matrix form of the neural learning. However, these methods are very special, they cannot solve general fuzzy equations with neural networks.

In this paper, we discuss more general fuzzy equations: dual fuzzy equations [39]. Normal fuzzy equations have fuzzy numbers only on one side of the equation. However, dual fuzzy equations have fuzzy numbers on both sides of the equation. Since the fuzzy numbers cannot be moved between the sides of the equation [21], dual fuzzy equations are more general and difficult. We first discuss the existence of the solutions of the dual fuzzy equations. It corresponds to controllability problem of the fuzzy control [12]. Then we provide two methods to approximate the solutions of the dual fuzzy equations. They are controller design process. Finally, we use five real examples to show the effectiveness of our fuzzy control design methods with neural networks.

2. Uncertain nonlinear system modeling with dual fuzzy equations

Consider the following unknown discrete-time nonlinear system

$$\bar{x}_{k+1} = \bar{f}\left[\bar{x}_k, u_k\right], \quad y_k = \bar{g}\left[\bar{x}_k\right] \tag{1}$$

where $u_k \in \Re^u$ is the input vector, $\bar{x}_k \in \Re^l$ is an internal state vector, and $y_k \in \Re^m$ is the output vector. $\bar{f}$ and $\bar{g}$ are general nonlinear smooth functions $\bar{f}, \bar{g} \in C^\infty$. Denoting $Y_k = \left[y_{k+1}^T, y_k^T, \cdots\right]^T$, $U_k = \left[u_{k+1}^T, u_k^T, \cdots\right]^T$. If $\frac{\partial Y}{\partial \bar{x}}$ is non-singular at $\bar{x} = 0$, $U = 0$, this leads to the following model

$$y_k = \Phi[y_{k-1}^T, y_{k-2}^T, \cdots u_k^T, u_{k-1}^T, \cdots] \tag{2}$$

where $\Phi(\cdot)$ is an unknown nonlinear difference equation representing the plant dynamics, u_k and y_k are measurable scalar input and output. The nonlinear system (2) is a NARMA model. We can also regard the input of the nonlinear system as

$$x_k = [y_{k-1}^T, y_{k-2}^T, \cdots u_k^T, u_{k-1}^T, \cdots]^T \tag{3}$$

the output as y_k.

Many nonlinear systems as in (2) can be rewritten as the following linear-in-parameter model,

$$y_k = \sum_{i=1}^{n} a_i f_i(x_k) \tag{4}$$

or

$$y_k + \sum_{i=1}^{m} b_i g_i(x_k) = \sum_{i=1}^{n} a_i f_i(x_k) \tag{5}$$

where a_i and b_i are linear parameters, $f_i(x_k)$ and $g_i(x_k)$ are nonlinear functions. The variables of these functions are measurable input and output.

A famous example of this kind of model is the robot manipulator [31]

$$M(q)\ddot{q} + C(q, \dot{q})\dot{q} + B\dot{q} + g(q) = \tau \tag{6}$$

(6) can be rewritten as

$$\sum_{i=1}^{n} Y_i(q, \dot{q}, \ddot{q})\theta_i = \tau \tag{7}$$

To identify or control the linear-in-parameter systems (4), (5) or (7), the normal least square or adaptive methods can be applied directly.

In this paper, we consider the uncertain nonlinear systems, *i.e.*, the parameters a_i, b_i or θ_i are not fixed (not crisp). They are uncertain in the sense of fuzzy logic. The uncertain nonlinear systems are modeled by linear-in-parameter models with fuzzy parameters. These models are called fuzzy equations. Before introduce fuzzy equations, we need the following definitions. The explanations for these definitions can be found in [42].

Definition 1. [fuzzy number] A fuzzy number u is a function $u \in E : \Re \to [0, 1]$, such that, 1) u is normal, (there exists $x_0 \in \Re$ such that $u(x_0) = 1$; 2) u is convex,

$u(\lambda x + (1-\lambda)y) \geq \min\{u(x), u(y)\}$, $\forall x, y \in \Re, \forall \lambda \in [0, 1]$; 3) u is upper semi-continuous on $\Re$, i.e., $u(x) \leq u(x_0) + \varepsilon$, $\forall x \in N(x_0)$, $\forall x_0 \in \Re,$, $\forall \varepsilon > 0$, $N(x_0)$ is a neighborhood; 4) The set $u^+ = \{x \in \Re, u(x) > 0\}$ is compact.

We use so called membership functions to express the fuzzy number. The most popular membership functions are the triangular function

$$u(x) = F(a, b, c) = \begin{cases} \frac{x-a}{b-a} & a \leq x \leq b \\ \frac{c-x}{c-b} & b \leq x \leq c \end{cases} \quad (8)$$

otherwise $u(x) = 0$, and trapezoidal function

$$u(x) = F(a, b, c, d) = \begin{cases} \frac{x-a}{b-a} & a \leq x \leq b \\ \frac{d-x}{d-c} & c \leq x \leq d \\ 1 & b \leq x \leq c \end{cases} \quad (9)$$

otherwise $u(x) = 0$.

Similar with crisp number, the fuzzy number u has also four basic operations: $\oplus$, $\ominus$, $\odot$, and $\oslash$. They represent the operations: sum, subtract, multiply, and "multiplied by a crisp number".

The dimension of x in the fuzzy number u depends on the membership function, for example (8) has three variables (9), has four variables. In order to define consistency operations, we first apply $\alpha-$level operation to the fuzzy number.

Definition 2. [α-level] The α-level of fuzzy number u is defined as

$$[u]^\alpha = \{x \in \Re : u(x) \geq \alpha\} \quad (10)$$

where $0 < \alpha \leq 1, u \in E$.

So $[u]^0 = u^+ =\{x \in \Re, u(x) > 0\}$. Because $\alpha \in [0, 1]$, $[u]^\alpha$ is a bounded as $\underline{u}^\alpha \leq [u]^\alpha \leq \overline{u}^\alpha$. The α-level of u between $\underline{u}^\alpha$ and $\overline{u}^\alpha$ is defined as

$$[u]^\alpha = A\left(\underline{u}^\alpha, \overline{u}^\alpha\right) \quad (11)$$

Let $u, v \in E$, $\lambda \in \Re$, we define the following fuzzy operations. $\underline{u}^\alpha$ and $\overline{u}^\alpha$ are the function of α. We define $\underline{u}^\alpha = d_M(\alpha)$, $\overline{u}^\alpha = d_U(\alpha)$, $\alpha \in [0, 1]$.

Definition 3. [Lipchitz constant] [27] The Lipschitz constant H of a fuzzy number $u \in E$ is

$$\begin{array}{c} |d_M(\alpha_1) - d_M(\alpha_2)| \leq H\,|\alpha_1 - \alpha_2| \\ \text{or } |d_U(\alpha_1) - d_U(\alpha_2)| \leq H\,|\alpha_1 - \alpha_2| \end{array} \quad (12)$$

Definition 4. [fuzzy operations] [38] Sum,

$$[u \oplus v]^\alpha = [u]^\alpha + [v]^\alpha = [\underline{u}^\alpha + \underline{v}^\alpha, \overline{u}^\alpha + \overline{v}^\alpha] \quad (13)$$

Subtract,

$$[u \ominus v]^\alpha = [u]^\alpha - [v]^\alpha = [\underline{u}^\alpha - \underline{v}^\alpha, \overline{u}^\alpha - \overline{v}^\alpha] \quad (14)$$

Multiply,

$$\underline{w}^\alpha \leq [u \odot v]^\alpha \leq \overline{w}^\alpha \text{ or } [u \odot v]^\alpha = A\left(\underline{w}^\alpha, \overline{w}^\alpha\right) \quad (15)$$

where $\underline{w}^\alpha = \underline{u}^\alpha \underline{v}^1 + \underline{u}^1 \underline{v}^\alpha - \underline{u}^1 \underline{v}^1$, $\overline{w}^\alpha = \overline{u}^\alpha \overline{v}^1 + \overline{u}^1 \overline{v}^\alpha - \overline{u}^1 \overline{v}^1$, $\alpha \in [0, 1]$. It is a cross product of twp fuzzy numbers.

Multiplied by a crisp number: For arbitrary crisp real positive number τ,

$$\begin{array}{l} -\tau \underline{u}^\alpha \leq [u]^\alpha \oslash \tau \leq -\tau \overline{u}^\alpha \\ \text{or } [u]^\alpha \oslash \tau = A(-\tau \underline{u}^\alpha, -\tau \overline{u}^\alpha) \end{array}$$

Obviously, we have the following properties: the scalar multiplication: $\alpha \in [0, 1]$

$$[\lambda u]^\alpha = \lambda [u]^\alpha = \begin{cases} A(\lambda \underline{u}^\alpha, \lambda \overline{u}^\alpha) & \lambda \geq 0 \\ A(\lambda \overline{u}^\alpha, \lambda \underline{u}^\alpha) & \lambda < 0 \end{cases} \quad (16)$$

$$\ominus u = (-1)u, \quad u \in E$$

Definition 5. [dot product] [6]The dot product of two fuzzy variables u and v is

$$(u.v)^\alpha = A\begin{pmatrix} \min\{\underline{u}^\alpha \underline{v}^\alpha, \underline{u}^\alpha \overline{v}^\alpha, \overline{u}^\alpha \underline{v}^\alpha, \overline{u}^\alpha \overline{v}^\alpha\} \\ \max\{\underline{u}^\alpha \underline{v}^\alpha, \underline{u}^\alpha \overline{v}^\alpha, \overline{u}^\alpha \underline{v}^\alpha, \overline{u}^\alpha \overline{v}^\alpha\} \end{pmatrix}$$

Definition 6. [distance] The distance between the fuzzy numbers u and v is

$$d(u, v) = \sup_{0 \leq \alpha \leq 1} \{\max\left(|\underline{u}^\alpha - \underline{v}^\alpha|, |\overline{u}^\alpha - \overline{v}^\alpha|\right)\} \quad (17)$$

Definition 7. [absolute value] [2] Absolute value of a triangular fuzzy number $u(x) = F(a, b, c)$ is

$$|u(x)| = |a| + |b| + |c| \quad (18)$$

Definition 8. [positive] A fuzzy number $u \in E$ is said to be positive if $\underline{u}^1 \geq 0$ and negative if $\overline{u}^1 \leq 0$.

Clearly, If u is positive and v is negative then $u \odot v = \ominus(u \odot (\ominus v))$ is a negative fuzzy number. If u is negative and v is positive then $u \odot v = \ominus((\ominus u) \odot v)$ is a negative fuzzy number. If u and v are negative then $u \odot v = (\ominus u) \odot (\ominus v)$ is a positive fuzzy number.

If u is positive and v is negative:

$$(u \odot v)^{\alpha} = A \begin{pmatrix} \overline{u}^{\alpha}\underline{v}^{1} + \overline{u}^{1}\underline{v}^{\alpha} \\ -\overline{u}^{1}\underline{v}^{1}, \underline{u}^{\alpha}\overline{v}^{1} + \underline{u}^{1}\overline{v}^{\alpha} - \underline{u}^{1}\overline{v}^{1} \end{pmatrix}$$

If u is negative and v is positive:

$$(u \odot v)^{\alpha} = A \begin{pmatrix} \underline{u}^{\alpha}\overline{v}^{1} + \underline{u}^{1}\overline{v}^{\alpha} \\ -\underline{u}^{1}\overline{v}^{1}, \overline{u}^{\alpha}\underline{v}^{1} + \overline{u}^{1}\underline{v}^{\alpha} - \overline{u}^{1}\underline{v}^{1} \end{pmatrix}$$

If u and v are negative:

$$(u \odot v)^{\alpha} = A \begin{pmatrix} \overline{u}^{\alpha}\overline{v}^{1} + \overline{u}^{1}\overline{v}^{\alpha} \\ -\overline{u}^{1}\overline{v}^{1}, \underline{u}^{\alpha}\underline{v}^{1} + \underline{u}^{1}\underline{v}^{\alpha} - \underline{u}^{1}\underline{v}^{1} \end{pmatrix}$$

When the parameters in the linear-in-parameter model (4) or (5) are fuzzy number, (4) and (5) become fuzzy equations. For the uncertain nonlinear system (1), we use the following two types of fuzzy equations to model it

$$y_k = a_1 f_1(x_k) \oplus a_2 f_2(x_k) \oplus \ldots \oplus a_n f_n(x_k) \quad (19)$$

or

$$\begin{aligned} & a_1 f_1(x_k) \oplus a_2 f_2(x_k) \oplus \ldots \oplus a_n f_n(x_k) \\ & = b_1 g_1(x_k) \oplus b_2 g_2(x_k) \oplus \ldots \oplus b_m g_m(x_k) \oplus y_k \end{aligned} \quad (20)$$

Because a_i and b_i are fuzzy numbers, we use the fuzzy operation $\oplus$. (20) has more general form than (19), it is called dual fuzzy equation.

In a special case, $f_i(x_k)$ has polynomial form,

$$a_1 x_k \oplus \ldots \oplus a_n x_k^n = b_1 x_k \oplus \ldots \oplus b_n x_k^n \oplus y_k \quad (21)$$

(21) is called dual fuzzy polynomial.

If we use the dual polynomial fuzzy Equation (21) to model a nonlinear function

$$z_k = f(x_k) \quad (22)$$

The object is to minimize error between the two output y_k and z_k. Since y_k is a fuzzy number and z_k is a crisp number, we use the maximum of all points as the modeling error

$$\max_k |y_k - z_k| = \max_k |y_k - f(x_k)| = \max_k |\beta_k| \quad (23)$$

where $y_k = F(a(k), b(k), c(k))$, $\beta_k = F(\beta_1, \beta_2, \beta_3)$, which are defined in (8). From the definition of the absolute value of a triangular fuzzy number (18),

$$\begin{aligned} & \max\nolimits_k |\beta_k| = \max\nolimits_k \begin{bmatrix} |a(k) - f(x_k)| \\ +|b(k) - f(x_k)| + |c(k) - f(x_k)| \end{bmatrix} \\ & \beta_1 = \max\nolimits_k |a(k) - f(x_k)| \\ & \beta_2 = \max\nolimits_k \{b(k) + f(x_k)\} \\ & \beta_3 = \max\nolimits_k \{c(k) + f(x_k)\} \end{aligned} \quad (24)$$

The modelling problem (23) is to find $a(k)$, $b(k)$, and $c(k)$, such that

$$\min_{a_k, b_k, c_k} \left\{ \max_k |\beta_k| \right\} = \min_{a_k, b_k, c_k} \left\{ \max_k |y_k - f(x_k)| \right\} \quad (25)$$

From (24)

$$\begin{aligned} & \beta_1 \geq |a(k) - f(x_k)|, \quad \beta_2 \geq b(k) + f(x_k) \\ & \beta_3 \geq c(k) + f(x_k) \end{aligned}$$

(25) can be solved by the linear programming method,

$$\begin{cases} \min \beta_1 \\ \text{subject: } \begin{array}{l} \beta_1 + \sum_{j=0}^{n} a_j x_k^j \ominus \sum_{j=0}^{n} b_j x_k^j \geq f(x_k) \\ \beta_1 - \{\sum_{j=0}^{n} a_j x_k^j \ominus \sum_{j=0}^{n} b_j x_k^j\} \geq -f(x_k) \end{array} \end{cases} \quad (26)$$

$$\begin{cases} \min \beta_2 \\ \text{subject: } \begin{array}{l} \beta_2 - \left[\sum_{j=0}^{n} \underline{a}_j x_k^j \ominus \sum_{j=0}^{n} \underline{b}_j x_k^j\right] \geq f(x_k) \\ \beta_2 \geq 0 \end{array} \end{cases} \quad (27)$$

$$\begin{cases} \min \beta_3 \\ \text{subject: } \begin{array}{l} \beta_3 - \left[\sum_{j=0}^{n} \bar{a}_j x_k^j \ominus \sum_{j=0}^{n} \bar{b}_j x_k^j\right] \geq f(x_k) \\ \beta_3 \geq 0 \end{array} \end{cases} \quad (28)$$

where $\underline{a}_j$, $\underline{b}_j$, $\bar{a}_j$ and $\bar{b}_j$ are defined as in (11). In this way, the best approximation of $f(x_k)$ at point x_k is $y_k = F(a_k, b_k, c_k)$. The approximation error of this approximation β_k is minimized.

In this paper, we use the dual fuzzy Equation (20) to model the uncertain nonlinear system (1). The controller design process is to find u_k, such that the output of the plant y_k can follow desired output y_k^*, or the trajectory tracking error is minimized

$$\min_{u_k} \left\| y_k - y_k^* \right\| \quad (29)$$

This control object can be considered as: finding a solution u_k for the following dual fuzzy equation

$$\begin{aligned} & a_1 f_1(x_k) \oplus a_2 f_2(x_k) \oplus \ldots \oplus a_n f_n(x_k) \\ & = b_1 g_1(x_k) \oplus b_2 g_2(x_k) \oplus \ldots \oplus b_m g_m(x_k) \oplus y_k^* \end{aligned} \quad (30)$$

where $x_k = [y_{k-1}^T, y_{k-2}^T, \cdots u_k^T, u_{k-1}^T, \cdots]^T$

It is impossible to obtain an analytical solution for (30). In this paper, we use neural networks to approximate the solution.

2.1. Controllability of uncertain nonlinear systems via dual fuzzy equations

Since the control object is to find a u_k for the dual fuzzy Equation (30), the controllability problem becomes if the dual fuzzy equation has solution. In order to show the existence of the solution of (30). We need the following lemmas

Lemma 1. *If the dual fuzzy Equation (30) has a crisp solution u_k, then*

$$\left\{\cap_{j=1}^{n}\text{domain}\left[f_j(x)\right]\right\} \cap \left\{\cap_{j=1}^{m}\text{domain}\left[g_j(x)\right]\right\} \neq \phi \tag{31}$$

Proof. Let $u_0 \in \Re$ be a solution of (30), the dual fuzzy equation becomes

$$\begin{aligned} & a_1 f_1(u_0) \oplus \ldots \oplus a_n f_n(u_0) \\ & = b_1 g_1(u_0) \oplus \ldots \oplus b_m g_m(u_0) \oplus y_k^* \end{aligned}$$

Since $f_j(u_0)$ and $g_j(u_0)$ exist, $u_0 \in \text{domain}\left[f_j(x)\right]$, $u_0 \in \text{domain}\left[g_j(x)\right]$. Consequently, it can be concluded that $u_0 \in \cap_{j=1}^{n}\text{domain}\left[f_j(x)\right] = D_1$, and $u_0 \in \cap_{j=1}^{m}\text{domain}\left[g_j(x)\right] = D_2$. So there exists u_0, such that $u_0 \in D_1 \cap D_2 \neq \phi$.

Obviously, the necessary condition for the existence of the solution of (30) is (31).

Assume two fuzzy numbers m_0, $n_0 \in E$, $m_0 < n_0$. Define a set $K(x) = \{x \in E, m_0 \leq x \leq n_0\}$, and an operator $S : K \to K$, such that

$$S(m_0) \geq m_0, \quad S(n_0) \leq n_0 \tag{32}$$

here S is condensing and continuous, it is bounded as $S(z) < r(z)$, $z \subset K$ and $r(z) > 0$. $r(Z)$ can be regarded as the measure of z.

Lemma 2. *If we define $n_i = S(n_{i-1})$ and $m_i = S(m_{i-1})$, $i = 1, 2, \ldots$, the upper and lower bounds of S are $\bar{s}$ and $\underline{s}$, then*

$$\bar{s} = \lim_{i \to +\infty} n_i, \quad \underline{s} = \lim_{i \to +\infty} m_i, \tag{33}$$

and

$$m_0 \leq m_1 \leq \ldots \leq m_n \leq \ldots \leq n_n \leq \ldots \leq n_1 \leq n_0. \tag{34}$$

The proof of this lemma is directly, see [11].

If there exists a fixed point x_0 in K, the successive iterates $x_i = S(x_{i-1})$, $i = 1, 2, \ldots$ will converge to x_0, *i.e.*, the distance (17) $\lim_{i\to\infty} d(x_i, x_0) = 0$.

Theorem 1. *If the fuzzy numbers a_i and b_j ($i = 1 \cdots n$, $j = 1 \cdots m$) in (30) satisfy the Lipschitz condition (12)*

$$\begin{aligned} |d_M(a_i) - d_M(a_k)| &\leq H\,|a_i - a_k| \\ |d_U(a_i) - d_U(a_k)| &\leq H\,|a_i - a_k| \\ |d_M(b_i) - d_M(b_k)| &\leq H\,|b_i - b_k| \\ |d_U(b_i) - d_U(b_k)| &\leq H\,|b_i - b_k| \end{aligned} \tag{35}$$

where $k = 1 \cdots n$, d_M and d_U are defined in (12), the upper bounds of f_i and g_j are $|f_i| \leq \bar{f}$, $\left|g_j\right| \leq \bar{g}$, then the dual fuzzy Equation (30) has a solution u which is in the following set

$$K_H = \left\{ \begin{array}{c} u \in E, |\overline{u}^{\alpha_1} - \underline{u}^{\alpha_2}| \\ \leq \left(n\bar{f} + m\bar{g}\right) H\,|\alpha_1 - \alpha_2| \end{array} \right\} \tag{36}$$

Proof. Because the fuzzy numbers a_i and b_j in (30) are linear-in-parameter, from the definition (12) and the property (16)

$$\begin{aligned} d_M(\alpha) = a_{1M}(\alpha) f_1(x) \oplus \ldots \oplus a_{nM}(\alpha) f_n(x) \\ \ominus b_{1M}(\alpha) g_1(x) \ominus \ldots \ominus b_{mM}(\alpha) g_m(x) \end{aligned}$$

So

$$\begin{aligned} & |d_M(\alpha) - d_M(\varphi)| = |f_1(x)|\,|\,a_{1M}(\alpha) \ominus a_{1M}(\varphi)\,| \\ & + \cdots + |f_n(x)|\,|a_{nM}(\alpha) \ominus a_{nM}(\varphi)| \\ & + |g_1(x)|\,|b_{1M}(\alpha) \ominus b_{1M}(\varphi)| \\ & + \cdots + |g_m(x)|\,|b_{mM}(\alpha) \ominus b_{mM}(\varphi)| \end{aligned} \tag{37}$$

By the Lipschitz condition (12), (37) is

$$\begin{aligned} |d_M(\alpha) - d_M(\varphi)| \leq \bar{f} H \textstyle\sum_{i=1}^{n} |\alpha - \varphi| \\ + \bar{g} H \textstyle\sum_{i=1}^{n} |\alpha - \varphi| = \left(n\bar{f} + m\bar{g}\right) H\,|\alpha - \varphi| \end{aligned}$$

Similarly, the upper bounds satisfy

$$|d_U(\alpha) - d_U(\varphi)| \leq \left(n\bar{f} + m\bar{g}\right) H\,|\alpha - \varphi|$$

Since the lower bound $|d_M(\alpha) - d_M(\varphi)| \geq 0$, by Lemma 2 the solution is in K_H which is defined in (36).

The following theorem uses linear the programming conditions (26)–(28) to show the controllability conditions of the dual polynomial fuzzy Equation (21).

Lemma 3. *If the data number m and the order the polynomial n in (21) satisfy*

$$m \geq 2n + 1 \tag{38}$$

where $k = 1 \cdots m$, then the solutions of (27) and (28) are $\beta_2 = \beta_3 = 0$.

Proof. Because

$$\sum_{j=0}^{n} \underline{a}_j x_k^j \ominus \sum_{j=0}^{n} \underline{b}_j x_k^j \leq -f(x_k) \tag{39}$$

$i = 1, 2, ..., m$. We choose $2n + 1$ points for x_k, and the interpolating the dual polynomial

$$b(k) = \sum_{j=0}^{n} \underline{a}_j x_k^j \ominus \sum_{j=0}^{n} \underline{b}_j x_k^j \tag{40}$$

If $h = \max_k\{b(k) + f(x_k)\}$ and $h > 0$, then we can change the dual polynomial (21) into a new dual polynomial $b(k) - h$. This new dual polynomial satisfies (39). Because the feasible point of (27) $\beta_2 \geq 0$, it must be zero. Similar result can be obtained for (28).

Both $f(x_k)$ and x_k are crisp. If the data number is $k = 1 \cdots n$, there exists solution for the polynomial approximation [25]. Because $b(k)$ and $c(k)$, (26) has a solution.

Theorem 2. *If the data number is big enough as (38), and the dual polynomial fuzzy Equation (21) satisfies*

$$D[h(x_{k1}, u_{k1}), h(x_{k2}, u_{k2})] \leq lD[u_{k1}, u_{k2}] \tag{41}$$

where $0 < l < 1$, $h(\cdot)$ represents a dual polynomial fuzzy equation,

$$\begin{aligned} h(x_{k1}, u_{k1}) &: a_1 x_{k1} \oplus ... \oplus a_n x_{k1}^n \\ &= b_1 x_{k1} \oplus ... \oplus b_n x_{k1}^n \oplus y_{k1} \end{aligned} \tag{42}$$

$D[u, v]$ is the Hausdorff distance [32],

$$D[u, v] = \max\left\{\sup_{x\in u}\inf_{y\in v} d(x, y), \sup_{x\in v}\inf_{y\in u} d(x, y)\right\}$$

$d(x, y)$ is the distance defined in (17), then (21) has a unique solution u.

Proof. From Lemma 2 we know, there are solutions for (26)–(28), if there are many data which satisfy (38). Without loss of generality, we assume the solutions for (26)–(28) are at $x_k = 0$, which corresponds to u_0 (41) means $h(\cdot)$ in (42) is continuous. If we choose a $\delta > 0$ such that $D\left[y_k, u_0\right] \leq \delta$, then

$$D[h(x_k, u_0), u_0] \leq (1 - l)\delta$$

Here $h(0, u_0) = u_0$. Now we select x near 0, $x_k \in [0, c]$, $c > 0$, and define

$$\mathcal{C}_0 : \rho = \sup_{x_k\in[0,c]} D\left[y_{k_1}, y_{k_2}\right]$$

Let $\{y_{k_m}\}$ be a sequence in $\mathcal{C}_0$, for any $\varepsilon > 0$, we can find $N_0(\varepsilon)$ such that $\rho < \varepsilon, m, n \geq N_0$. So $y_{k_m} \longrightarrow y_k$ for $x_k \in [0, c]$. Furthermore

$$D\left[y_k, u_0\right] \leq D\left[y_k, y_{k_m}\right] + D\left[y_{k_m}, u_0\right] < \varepsilon + \delta \tag{43}$$

for all $x \in [0, c]$, $m \geq N_0(\varepsilon)$. Since $\varepsilon > 0$ is arbitrary small,

$$D\left[y_k, u_0\right] \leq \delta \tag{44}$$

for all $x \in [0, c]$. We now show that y_k is continuous at $x_0 = 0$. Given $\delta > 0$, there exists $\delta_1 > 0$ such that

$$D\left[y_k, u_0\right] \leq D\left[y_k, y_{k_m}\right] + D\left[y_{k_m}, u_0\right] \leq \varepsilon + \delta_1$$

for every $m \geq N_0(\varepsilon)$, by (44), whenever $|x - x_0| < \delta_1$, y_k is continuous at $x_0 = 0$. So (21) has a unique solution u_0.

The necessary condition for the controllability (existence of solution) of the dual fuzzy Equation (30) is (31), the sufficient condition of the controllability is (35). For most of membership functions such as the triangular function (8) and the trapezoidal function (9), the Lipschitz condition (35) is satisfied. They are controllable.

3. Fuzzy controller design with neural networks approximation

There are not analytical solution for the dual fuzzy Equation (30). In this paper, we use neural networks to approximate the solution (control). In order to use neural networks to approximate the solution of the dual fuzzy Equation (30), we first need to transform it into normal fuzzy Equation as (19).

Generally, the inverse element for an arbitrary fuzzy number $u \in E$ does not exist , *i.e.*, there is not $v \in E$, such that

$$u \oplus v = 0$$

In other word,

$$u \oplus (\ominus u) \neq 0$$

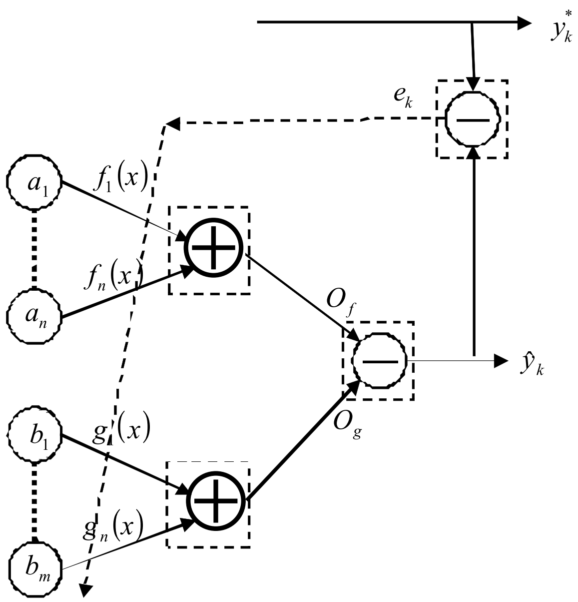

Fig. 1. Dual fuzzy equation in the form of neural network (NN).

So (30) cannot be

$$\begin{gathered} a_1 f_1(x_k) \oplus ... \oplus a_n f_n(x_k) \ominus b_1 g_1(x_k) \\ \ominus ... \ominus b_m g_m(x_k) = y_k^* \\ [a_1 \ominus b_1]\, f_1(x_k) \oplus [a_2 \ominus b_2]\, f_2(x_k) \oplus ... = y_k^* \end{gathered}$$

In this paper we use the $\oslash$ operation. We add $\oplus b_i g_i(x)$, and apply $\oslash\tau$ on the both sides of (30)

$$\begin{aligned} & a_1 f_1(x_k) \oplus ... \oplus a_n f_n(x_k) \\ & \oplus \left\{ \left[b_1 g_1(x) \oplus ... \oplus b_m g_m(x) \right] \oslash \tau \right\} \\ & = b_1 g_1(x_k) \oplus ... \oplus b_m g_m(x_k) \\ & \oplus \left\{ \left[b_1 g_1(x) \oplus ... \oplus b_m g_m(x) \right] \oslash \tau \right\} \oplus y_k^* \end{aligned} \tag{45}$$

When $\tau = 1$, be the definition of $\oslash$, (45) is

$$a_1 f_1(x) \oplus ... \oplus a_n f_n(x) \ominus b_1 g_1(x) \ominus ... \ominus b_m g_m(x) = y_k^* \tag{46}$$

We design a neural network to represent the fuzzy equation (46), see Fig. 1. The input to the neural network is the fuzzy numbers a_i and b_i, the output of the fuzzy number y_k. The weights are $f_i(x)$ and $g_j(x)$.

The objective is to find suitable weight x (solution) such that the output of the neural network $\hat{y}_k$ converges to the desired output y_k^*. In the control point of view, we want to find a controller u_k which is a function of x, such that the output of the plant (1) y_k (crisp value) approximate the fuzzy number y_k^*.

In order to simplify the operation of the neural network as in Fig. 1, we use the triangular fuzzy number (8) in this paper. The input fuzzy numbers a_i and b_i are first applied to α-level as in (10)

$$\begin{aligned} [a_i]^\alpha &= A\left(\underline{a}_i^\alpha, \overline{a}_i^\alpha\right) \quad i = 1 \cdots n \\ [b_j]^\alpha &= A\left(\underline{b}_i^\alpha, \overline{b}_i^\alpha\right) \quad j = 1 \cdots m \end{aligned} \tag{47}$$

Then they are multiplied by the weights $f_i(x)$ and $g_j(x)$, and summarized according to (13)

$$\begin{aligned} \left[O_f\right]^\alpha &= A\begin{pmatrix} \sum_{i\epsilon M_f} f_i(x)\,\underline{a_i}^\alpha + \sum_{i\epsilon C_f} f_i(x)\,\overline{a_i}^\alpha, \\ \sum_{i\epsilon C_f} f_i(x)\,\overline{a_i}^\alpha, \sum_{i\epsilon M_f} f_i(x)\,\underline{a_i}^\alpha \end{pmatrix} \\ \left[O_g\right]^\alpha &= A\begin{pmatrix} \sum_{j\epsilon M_g} g_j(x)\,\underline{b_j}^\alpha + \sum_{j\epsilon C_g} g_j(x)\,\overline{b_j}^\alpha, \\ \sum_{j\epsilon C_g} g_j(x)\,\overline{b_j}^\alpha, \sum_{j\epsilon M_g} g_j(x)\,\underline{b_j}^\alpha \end{pmatrix} \end{aligned} \tag{48}$$

where $M_f = \{i | f_i(x) \geq 0\}$, $C_f = \{i | f_i(x) < 0\}$, $M_g = \{j | g_j(x) \geq 0\}$, $C_g = \{j | g_j(x) < 0\}$.

The output of the neural network is

$$\left[\hat{y}_k\right]^\alpha = A\left(\underline{O_f}^\alpha - \underline{O_g}^\alpha, \overline{O_f}^\alpha - \overline{O_g}^\alpha\right) \tag{49}$$

In order to train the weights, we need to define a cost function for the fuzzy numbers. The training error is

$$e_k = y_k^* \ominus \hat{y}_k$$

where $\left[y_k^*\right]^\alpha = A\left(\underline{y_k^*}^\alpha, \overline{y_k^*}^\alpha\right)$, $\left[\hat{y}_k\right]^\alpha = A\left(\underline{\hat{y}_k}^\alpha, \overline{\hat{y}_k}^\alpha\right)$, $[e_k]^\alpha = A\left(\underline{e_k}^\alpha, \overline{e_k}^\alpha\right)$. The cost function is defined as

$$\begin{aligned} J_k &= \underline{J}^\alpha + \overline{J}^\alpha \\ \underline{J}^\alpha &= \tfrac{1}{2}\left(\underline{y_k^*}^\alpha - \underline{\hat{y}_k}^\alpha\right)^2 \\ \overline{J}^\alpha &= \tfrac{1}{2}\left(\overline{y_k^*}^\alpha - \overline{\hat{y}_k}^\alpha\right)^2 \end{aligned} \tag{50}$$

Obviously, $J_k \to 0$ means $\left[\hat{y}_k\right]^\alpha \to \left[y_k^*\right]^\alpha$.

Remark 1. A main advantage of the least mean square index (50) is that it has a self-correcting feature which permits to operate for arbitrarily long period without deviating from its constraints. The corresponding gradient algorithm is susceptible to cumulative round off errors and is suitable for long runs without an additional error-correction procedure. It is more robust in statistics, identification and signal processing [33].

Now we use gradient method to train the weights $f_i(x)$ and $g_j(x)$. The solution x_0 is the functions of $f_i(x)$ and $g_j(x)$. We calculate $\frac{\partial J_k}{\partial x_0}$ as

$$\frac{\partial J_k}{\partial x_0} = \frac{\partial \underline{J}^\alpha}{\partial x_0} + \frac{\partial \overline{J}^\alpha}{\partial x_0}$$

By the chain rule

$$\frac{\partial \underline{J}^{\alpha}}{\partial x_0} = \frac{\partial \underline{J}^{\alpha}}{\partial \underline{\hat{y}_k}^{\alpha}} \frac{\partial \underline{\hat{y}_k}^{\alpha}}{\partial \underline{O_f}^{\alpha}} \sum \frac{\partial \underline{O_f}^{\alpha}}{\partial f_i(x)} \frac{\partial f_i(x)}{\partial x_0}$$
$$+ \frac{\partial \underline{e}^{\alpha}}{\partial \underline{\hat{y}_k}^{\alpha}} \frac{\partial \underline{\hat{y}_k}^{\alpha}}{\partial \underline{O_g}^{\alpha}} \sum \frac{\partial \underline{O_g}^{\alpha}}{\partial g_j(x)} \frac{\partial g_j(x)}{\partial x_0}$$
$$\frac{\partial \overline{J}^{\alpha}}{\partial x_0} = \frac{\partial \underline{J}^{\alpha}}{\partial \underline{\overline{\hat{y}_k}}^{\alpha}} \frac{\partial \underline{\overline{\hat{y}_k}}^{\alpha}}{\partial \overline{O_f}^{\alpha}} \sum \frac{\partial \overline{O_f}^{\alpha}}{\partial f_i(x)} \frac{\partial f_i(x)}{\partial x_0}$$
$$+ \frac{\partial \overline{e_k}^{\alpha}}{\partial \overline{\hat{y}_k}^{\alpha}} \frac{\partial \overline{\hat{y}_k}^{\alpha}}{\partial \overline{O_f}^{\alpha}} \sum \frac{\partial \overline{O_f}^{\alpha}}{\partial g_j(x)} \frac{\partial g_j(x)}{\partial x_0}$$

If f_i' and g_j' are positive

$$\frac{\partial \underline{J}^{\alpha}}{\partial x_0} = \sum_{i=1}^{n} - \left(\underline{y_k^{*}}^{\alpha} - \underline{\hat{y}_k}^{\alpha}\right) \underline{a}_i^{\alpha} f_i'$$
$$+ \sum_{j=1}^{m} \left(\underline{y_k^{*}}^{\alpha} - \underline{\hat{y}_k}^{\alpha}\right) \underline{b}_j^{\alpha} g_j'$$
$$\frac{\partial \overline{J}^{\alpha}}{\partial x_0} = \sum_{i=1}^{n} - \left(\overline{y_k^{*}}^{\alpha} - \overline{\hat{y}_k}^{\alpha}\right) \overline{a_i}^{\alpha} f_i'$$
$$+ \sum_{j=1}^{m} \left(\overline{y_k^{*}}^{\alpha} - \overline{\hat{y}_k}^{\alpha}\right) \overline{b_j}^{\alpha} g_j'$$

Otherwise

$$\frac{\partial \underline{J}^{\alpha}}{\partial x_0} = \sum_{i=1}^{n} - \left(\underline{y_k^{*}}^{\alpha} - \underline{\hat{y}_k}^{\alpha}\right) \overline{a_i}^{\alpha} f_i'$$
$$+ \sum_{j=1}^{m} \left(\underline{y_k^{*}}^{\alpha} - \underline{\hat{y}_k}^{\alpha}\right) \overline{b_j}^{\alpha} g_j'$$
$$\frac{\partial \overline{J}^{\alpha}}{\partial x_0} = \sum_{i=1}^{n} - \left(\overline{y_k^{*}}^{\alpha} - \overline{\hat{y}_k}^{\alpha}\right) \underline{a}_i^{\alpha} f_i'$$
$$+ \sum_{j=1}^{m} \left(\overline{y_k^{*}}^{\alpha} - \overline{\hat{y}_k}^{\alpha}\right) \underline{b}_j^{\alpha} g_j'$$

The solution x_0 is updated as

$$x_0(k+1) = x_0(k) - \eta \frac{\partial J_k}{\partial x_0}$$

where η is the training rate $\eta > 0$. In order to increase training process, we add a momentum term as

$$x_0(k+1) = x_0(k) - \eta \frac{\partial J_k}{\partial x_0} + \gamma \left[x_0(k) - x_0(k-1)\right]$$

where $\gamma > 0$

After x_0 is updated, it should be substitute to the weights $f_i(x_0)$ and $g_j(x_0)$.

The solution of the dual fuzzy equation (30) can be also approximated by another type of neural network, see Fig. 2. Here the inputs are the nonlinear functions $f_i(x)$ and $g_j(x)$, the weights are the fuzzy number a_i and b_j. We use the training error e_k to update x.

The input is a crisp number $x(k)$. After the nonlinear operations $f_i(x)$ and $g_j(x)$, O_f and O_g are the same as (48). The output of this neural network is the same as (49).

The different between the networks of Fig. 1 (NN) and Fig. 2 (FNN) are: FNN does not change weights, it is an autonomous system. NN is a standard neural network. FNN is more robust than NN, and we can use bigger training rate η in FNN.

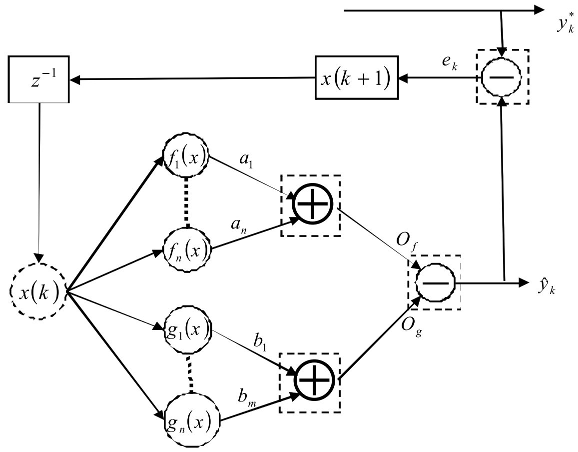

Fig. 2. Dual fuzzy equation in the form of feedback neural network (FNN).

4. Applications

In this section, we use several real applications to show how to use the dual fuzzy equation to design fuzzy controller.

Example 1. [A chemistry process] A chemical reaction is to use the poly ethylene (PE) and poly propylene (PP) to generate a desired substance (DS). If the cost of the material is defined as x, the cost PE is x and the cost of PP is x^2. The weights of PE and PP are uncertain, which satisfy the triangle function (8). We want to product two types DS. If we wish the cost between them are $F(3.5, 4, 5) = y^*$, what is the cost x ? The weights of PE are $F(2.5, 3, 3.25) = a_1$ and $F(0.75, 1, 1.25) = b_1$. The weights of PP are $F(1.75, 2, 2.5) = a_2$ and $(1.75, 2, 2.5) = b_2$. The above relation can be modeled by the following dual fuzzy equation

$$(2.5, 3, 3.25)x \oplus (1.75, 2, 2.5)x^2$$
$$= (0.75, 1, 1.25)x \oplus (1.75, 2, 2.5)x^2 \oplus (3.5, 4, 5)$$

Here $f_1(x) = g_1(x) = x$, $f_2(x) = g_2(x) = x^2$. We use NN and FNN shown in Fig. 1 and Fig. 2 to approximate the solution x. The learning rates for them are the same $\eta = 0.02$. The results are shown in Table 1. The exact solution is $x_0 = 2$. The neural networks start from $x(0) = 4$. Both neural networks converge to the real solution.

Table 1
Comparison results of two types of neural networks

k	$x(k)$ with NN	k	$x(k)$ with FNN
1	3.8377	1	3.7970
2	3.6105	2	3.3090
3	3.3435	3	2.9567
⋮	⋮	⋮	⋮
38	2.0053	26	2.0080
39	2.0044	27	2.0053
40	2.0036	28	2.0034

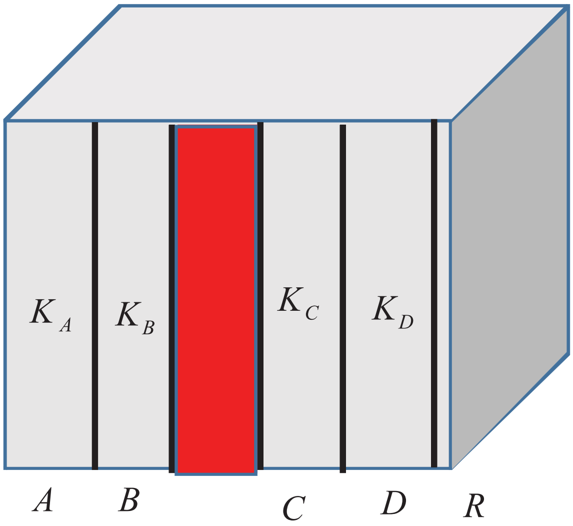

Fig. 3. Heat source by insulating materials.

Example 2. [Heat source by insulating materials]Heat source is in the center of the insulating materials. The thickness of the materials are not exact, which satisfy the trapezoidal function (9),

$$A = F(0.12, 0.14, 0.15, 0.18) = a_1$$
$$B = F(0.08, 0.1, 0.2, 0.5) = a_2$$
$$C = F(0.09, 0.1, 0.2, 0.4) = b_1$$
$$D = F(0.02, 0.03, 0.05, 0.08) = b_2$$

see Fig. 3. The conductivity coefficient of these materials are $K_A = e^x = f_1$, $K_B = x\sqrt{x} = f_2$, $K_C = x^2 = g_1$, $K_D = x sin(\frac{\Pi x}{8}) = g_2$, here x is the elapsed time. The object of the example is to find the time when the thermal resistance at the right side arrives $R = F(0.00415, 0.00428, 0.00569, 0.03187) = y^*$.

Table 2
Comparison results of two types of neural networks

k	$x(k)$ with NN	k	$x(k)$ with FNN
1	0.6251	1	0.7250
2	1.0542	2	1.1060
3	1.3321	3	1.5042
⋮	⋮	⋮	⋮
39	2.9899	10	2.9931
40	2.9922	11	2.9959
41	2.9940	12	2.9974

The thermal balance is [18]:

$$\frac{A}{K_A} \oplus \frac{B}{K_B} = \frac{C}{K_C} \oplus \frac{D}{K_D} \oplus R$$

The exact solution is $x = 3$ [18]. The maximum learning rate of NN as Fig. 1 is $\eta = 0.005$. The maximum learning rate of FNN as Fig. 2 is $\eta = 0.1$. The approximation results are shown in Table 2. FNN is faster and more robust than NN.

Example 3. [Water tank system] The water tank system has two inlet valves q_1, q_2, and two outlet valves q_3, q_4, see Fig. 4. The areas of the valves are uncertain as the triangle function (8), $A_1 = F(0.023, 0.025, 0.026)$, $A_2 = F(0.01, 0.02, 0.04)$, $A_3 = F(0.014, 0.015, 0.017)$, $A_4 = F(0.04, 0.06, 0.07)$. The velocities of the flow (controlled by the valves) are $f_1 = (\frac{x}{10})e^x$, $f_2 = x cos(\Pi x)$, $f_3 = cos(\frac{\Pi x}{8})$, $f_4 = \frac{x}{2}$. If we hope the outlet flow is $q = (4.090, 6.338, 36.402) = y^*$, what is the control variable x. The mass balance of the tank is [34]:

$$\rho A_1 f_1 \oplus \rho A_2 f_2 = \rho A_3 f_3 \oplus \rho A_4 f_4 \oplus q$$

where ρ is the density of the water. The exact solution is $x_0 = 2$ [34]. We use $x(0) = 5$, $\eta = 0.001$, $\gamma = 0.001$ for both NN and FNN. The error $|\hat{x} - x_0|$ between the approximate solution $\hat{x}$ and the exact solution x_0 is shown in Fig. 5. For this example, both NN and FNN work well.

Example 4. [Solid cylindrical rod] The deformation of a solid cylindrical rod depend on the stiffness E, the forces on it F, the positions of the forces L, and the diameter of the rod d [35], see Fig. 6. The positions are not exact, they satisfy the trapezoidal function (9). $L_1 = F(0.3, 0.4, 0.6, 0.7)$, $L_2 = F(0.5, 0.7, 0.8, 0.9)$, $L_3 = F(0.5, 0.7, 0.8, 0.9)$. The are of the rod is $A = \frac{\pi}{4}d^2$. The external forces are the function of x, $F_1 = x^7$, $F_2 = x^6\sqrt{x}$, $F_3 = e^{2x}$ [7].

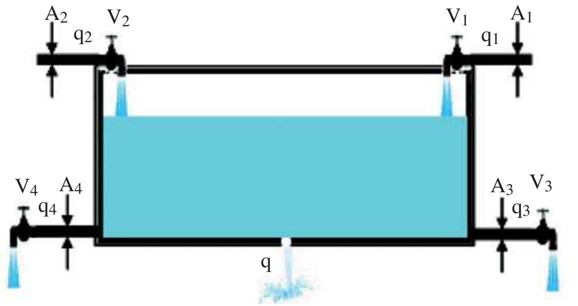

Fig. 4. Water tank system.

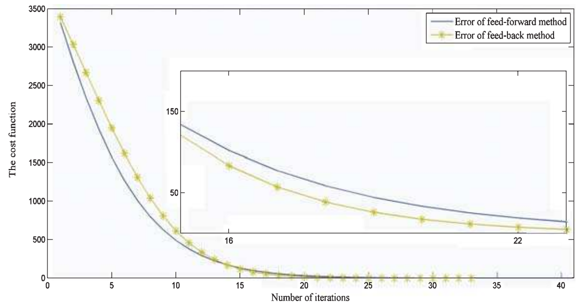

Fig. 5. The error between the approximate solution and the exact solution.

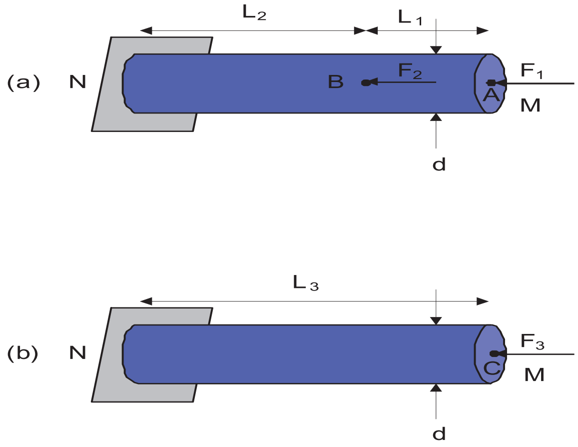

Fig. 6. Two solid cylindrical rods.

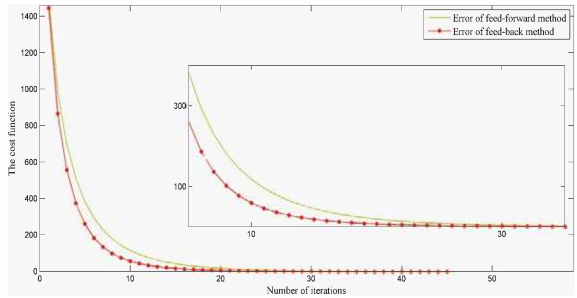

Fig. 7. The error between the approximate solution and the exact solution.

We want the desired deformation at the point N is $N^* = F(0.000673, 0.000931, 0.001164, 0.001310)$ as in (9), what is control force should be applied. According to the tension relations [7]

$$\frac{L_1 F_1}{AE} \oplus \frac{L_2(F_1 + F_2)}{AE} = \frac{L_3 F_3}{AE} \oplus N^*$$

where $d = 0.02,\ E = 70 \times 10^9$. The exact solution is $x = 4$.

We use $x(0) = 7, \eta = 0.002, \gamma = 0.002$ for both NN and FNN. The error $|\hat{x} - x_0|$ between the approximate solution $\hat{x}$ and the exact solution x_0 is shown in Fig. 7. For this example, both NN and FNN work well. FNN is little better than NN.

Example 5. [Water Channel system] The water in the pipe d_1 is divided into three pipes d_2, d_3, d_4, see Fig. 8. The areas of the pipes are uncertain, they satisfy the trapezoidal function (9). $A_1 = F(0.4, 0.6, 0.7, 0.8)$, $A_2 = F(0.05, 0.1, 0.2, 0.4)$, $A_3 = F(0.03, 0.08, 0.1, 0.2)$. The water velocities in the pipes are controlled by the valves parameter x, $v_1 = x^3$, $v_2 = \frac{e^x}{2}$, $v_3 = x$ [34]. The control object is to let the flow in pipe d_4 is

Table 3
Comparison results of two types of neural networks

k	$x(k)$ with NN	k	$x(k)$ with FNN
1	5.9024	1	5.9226
2	5.7361	2	5.5341
3	5.5321	3	5.1234
⋮	⋮	⋮	⋮
77	3.0599	21	3.0162
78	3.0322	22	3.0131
79	3.0110	23	3.0086

$$Q = F(10.207861, 14.955723, 16.591446, 16.982892)$$

what is the valve control parameter x. By mass balance

$$A_1 v_1 = A_2 v_2 \oplus A_3 v_3 \oplus Q$$

The exact solution is $x = 3$ [34]. The maximum learning rate of NN as Fig. 1 is $\eta = 0.001$. The maximum learning rate of FNN as Fig. 2 is $\eta = 0.08$. The approximation results are shown in Table 3. FNN is faster and more robust than NN.

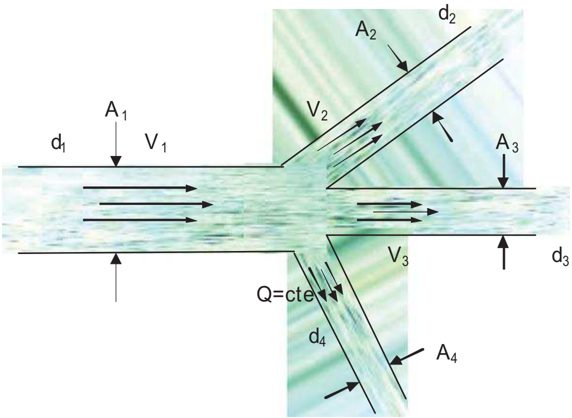

Fig. 8. Water Channel system.

5. Conclusions

In order to model uncertain nonlinear system, we use dual fuzzy equations, which are in the form of linear-in-parameter. We first prove that these fuzzy models have solutions under certain conditions. These conditions are controllability of the fuzzy control algorithms. By some special fuzzy operations, we transform the dual fuzzy equations into two types of neural networks. We design modified gradient descent algorithms to train the neural networks, such that the solutions (fuzzy controllers) are estimated by the neural networks he novel methods are validated with Five benchmark examples are proposed to validate our methods. Further work is to study the stability of training algorithms.

References

[1] S. Abbasbandy, The application of homotopy analysis method to nonlinear equations arising in heat transfer, *Phys Lett A* **360** (2006), 109–113.

[2] S. Abbasbandy and R. Ezzati, Newton's method for solving a system of fuzzy nonlinear equations, *Appl Math Comput* **175** (2006), 1189–1199.

[3] R.P. Agarwal, V. Lakshmikantham and J.J. Nieto, On the concept of solution for fractional differential equations with uncertainty, *Nonlinear Anal* **72** (2010), 2859–2862.

[4] T. Allahviranloo, M. Otadi and M. Mosleh, Iterative method for fuzzy equations, *Soft Computing* **12** (2007), 935–939.

[5] S. Arshad and V. Lupulescu, On the fractional differential equations with uncertainty, *Nonlinear Anal* **74** (2011), 3685–3693.

[6] A. Ban and B. Bede, Properties of the cross product of fuzzy numbers, *Journal of Fuzzy Mathematics* **14** (2006), 513–531.

[7] F.P. Beer and E.R. Johnston, Mechanics of materials, 2nd Edition, mcgraw-Hill, 1992.

[8] J. Buckley and Y. Qu, Solving linear and quadratic fuzzy equations, *Fuzzy Sets Systems* **35** (1990), 43–59.

[9] J. Buckley and E. Eslami, Neural net solutions to fuzzy problems: The quadratic equation, *Fuzzy Sets Syst* **86** (1997), 289–298.

[10] J. Buckley and Y. Hayashi, Can fuzzy neural nets approximate continuous fuzzy functions, *Fuzzy Sets Syst* **61** (1994), 43–51.

[11] J.-C. Chang, H. Chen, S.-M. Shyu and W.-C. Lian, Fixed-point theorems in fuzzy real line, *Comput Math Appl* **47** (2004), 845–851.

[12] S.M. Chen, Fuzzy system reliability analysis using fuzzy number arithmetic operations, *Fuzzy Sets Syst* **64** (1994), 31–38.

[13] G. Cybenko, Approximation by superposition of sigmoidal activation function, *Math Control Sig Syst* **2** (1989), 303–314.

[14] D. Dubois and H. Prade, Fuzzy set theoretic differences and inclusions and their use in the analysis of fuzzy equations, *Control Cybern* **13** (1984), 129–146.

[15] G. Feng, A survey on analysis and design of model-based fuzzy control systems, *IEEE Trans Fuzzy Syst* **14**(5) (2006), 676–697.

[16] M. Friedman, M. Ming and A. Kandel, Fuzzy linear systems, *Fuzzy Sets and Systems* **96** (1998), 201–209.

[17] R. Goetschel and W. Voxman, Elementary calculus, *Fuzzy Sets Syst* **18** (1986), 31–43.

[18] J.P. Holman, Heat transfer, 8th Ed, McGraw-Hill, New York, 1997.

[19] A. Jafarian, R. Jafari, A. Khalili and D. Baleanud, Solving fully fuzzy polynomials using feed-back neural networks, *International Journal of Computer Mathematics* **92**(4) (2015), 742–755.

[20] A. Jafarian and S. Measoomynia, Solving fuzzy polynomials using neural nets with a new learning algorithm, *Appl Math Sci* **5** (2011), 2295–2301.

[21] M. Kajani, B. Asady and A. Vencheh, An iterative method for solving dual fuzzy nonlinear equations, *Appl Math Comput* **167** (2005), 316–323.

[22] V. Lupulescu, On a class of fuzzy functional differential equations, *Fuzzy Sets Syst* **160** (2009), 1547–1562.

[23] E.H. Mamdani, Application of fuzzy algorithms for control of simple dynamic plant, *IEE Proceedings - Control Theory and Applications* **121**(12) (1976), 1585–1588.

[24] M. Mazandarani and A.V. Kamyad, Modified fractional Euler method for solving fuzzy fractional initial value problem, *Commun Nonlinear Sci Numer Simul* **18** (2013), 12–21.

[25] H.N. Mhaskar and D.V. Pai, Fundamentals of Approximation Theory, CRC Press, Boca Raton. FL, 2000.

[26] M. Mosleh, Evaluation of fully fuzzy matrix equations by fuzzy neural network, *Appl Math Model* **37** (2013), 6364–6376.

[27] H.T. Nguyen, A note on the extension principle for fuzzy sets, *J Math Anal Appl* **64** (1978), 369–380.

[28] S. Pederson and M. Sambandham, The Runge-Kutta method forhybrid fuzzy differential equation, *Nonlinear Anal Hybrid Syst* **2** (2008), 626–634.

[29] S. Salahshour, T. Allahviranloo and S. Abbasbandy, Solving fuzzy fractional differential equations by fuzzy Laplace transforms, *Commun Non-linear Sci Numer Simul* **17** (2012), 1372–1381.

[30] E. Sanchez, Solution of fuzzy equations with extended operations, *Fuzzy Sets and Systems* **12** (1984), 273–248.

[31] M.W. Spong and M. Vidyasagar, *Robot Dynamics and Control*, John Wiley and Sons, 1989.

[32] P.V. Subrahmanyam and S.K. Sudarsanam, On some fuzzy functional equations, *Fuzzy Sets and Systems* **64** (1994), 333–338.

[33] J.A.K. Suykens, J. De Brabanter, L. Lukas and J. Vandewalle, Weighted least squares support vector machines: Robustness and sparse approximation, *Neurocomputing* **48** (2002), 85–105.

[34] V.L. Streeter, E.B. Wylie and E. Benjamin, Fluid mechanics, 4th Ed, Mc. Graw-Hill Book Company, 1999.
[35] A. Tahavvor and M. Yaghoubi, Analysis of natural convection from a column of cold horizontal cylinders using artificial neural network, *Appl Math Model* **36** (2012), 3176–3188.
[36] T. Takagi and M. Sugeno, Fuzzy identification of systems and its applications to modeling and control, *IEEE Trans Syst, Man, Cybern* **15** (1985), 116–132.
[37] H. Wang and Y. Liu, Existence results for fuzzy integral equations of fractional order, *Int J Math Anal* **5** (2011), 811–818.
[38] M. Wagenknecht, R. Hampel and V. Schneider, Computational aspects of fuzzy arithmetics based on archimedean t-norms, *Fuzzy Sets Syst* **123** (2001), 49–62.
[39] M. Waziri and Z. Majid, A new approach for solving dual fuzzy nonlinear equations using Broyden's and Newton's methods, Advances in Fuzzy Systems, 2012, 5. Article 682087
[40] W. Yu and X. Li, Fuzzy identification using fuzzy neural networks with stable learning algorithms, *IEEE Transactions on Fuzzy Systems* **12**(3) (2004), 411–420.
[41] L.A. Zadeh, The concept of a liguistic variable and its application to approximate reasoning, *Information Sciences* **8** (1975), 199–249.
[42] L.A. Zadeh, Toward a generalized theory of uncertainty (GTU) an outline, *Information Sciences* **172** (2005), 1–40.