IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received November 15, 2016, accepted December 23, 2016, date of publication January 5, 2017, date of current version January 27, 2017.

Digital Object Identifier 10.1109/ACCESS.2017.2647920

Fuzzy Differential Equations for Nonlinear System Modeling With Bernstein Neural Networks

RAHELEH JAFARI[1], WEN YU[1], (Senior Member, IEEE), AND XIAOOU LI[2]
[1]Departamento de Control Automático, CINVESTAV-IPN, National Polytechnic Institute, Mexico City 07360, Mexico
[2]Departamento de Computación, CINVESTAV-IPN, National Polytechnic Institute, Mexico City 07360, Mexico

Corresponding author: W. Yu (yuw@ctrl.cinvestav.mx)

ABSTRACT With the fuzzy set theory, the uncertainty of nonlinear systems can be modeled using fuzzy differential equations. The solutions of these equations are the model output, but they are very difficult to obtain. In this paper, we first transform fuzzy differential equations into four ordinary differential equations. Then, we construct neural models with the structure of these ordinary differential equations. Theory analysis and simulation results show that these new models are effective for modeling uncertain nonlinear systems.

INDEX TERMS Fuzzy equation, nonlinear system modeling, neural networks.

I. INTRODUCTION

Since the uncertainty in parameters can be transformed into fuzzy set theory [56], fuzzy set and fuzzy system theory are good tools to address uncertainty systems. Fuzzy models are applied to a large class of uncertain nonlinear systems, for example Takagi-Sugeno fuzzy model [52]. When the parameters of an equation are changeable in the manner of a fuzzy set, this equation becomes a fuzzy equation [13]. When the parameters or the states of the differential equations are uncertain, they can be modeled with fuzzy differential equations (FDEs).

Many FDEs use fuzzy numbers as the coefficients of the differential equations to describe the uncertainties [21]. The applications of these FDEs are connected with nonlinear modeling and control [29]–[32]. Another type of FDE uses fuzzy variables to express the uncertainties. Studies on the solutions of FDEs are applied to chaotic analysis, quantum systems, and engineering problems, such as those in civil engineering. The basic idea of the fuzzy derivative was first introduced in [16] and was extended in [19]. The linear first-order equation is the simplest FDE. By generalizing the differentiability, [8] gave an analytical solution. In [35], the first-order FDE with periodic boundary conditions was analyzed. Then, higher order linear FDEs were studied. In [6], the analytical solutions of the second-order FDE were obtained. The analytical solutions of third-order linear FDEs are found in [26], while [12] and [5] proposed analytical approaches to resolve nth-order linear FDEs.

It is more difficult to solve nonlinear FDEs. Using the interval-valued method, [50] examined the basis of solutions for nonlinear FDEs with generalized differentiability. [44] used periodic boundary and Hukuhara differentiability for impulsive FDEs. [22] suggested some suitable criteria to fuzzify the crisp solutions. [38] used two-point fuzzy boundary values for FDEs. [25] used homeotypic analysis technique for FDEs. However, all of above analytical methods for the solutions of FDEs are very difficult, especially for nonlinear FDEs.

Numerical solutions of FDEs have been discussed recently by many scientists. The numerical solutions of first-order FDEs were proposed in [49] with an iterative technique. [3] used Laplace transformation for second-order FDEs. By extending classical fuzzy set theory, [28] obtained a numerical solution for an FDE. The predictor-corrector approach was applied in [4]. The Euler numerical technique was used in [42], [53], and [43] to solve FDEs. Other numerical techniques, such as the Nystrom approach [34], Taylor method [1] and Runge-Kutta approach [45], can also be applied to solve FDEs. However, the approximation accuracy of these numerical calculations are normally lower [39].

The solutions of FDEs are uniformly continuous and inside compact sets [11]. Neural networks can provide good estimations for the solutions of FDEs. [2] showed that the solution of an ordinary differential equation (ODE) can be approximated by a neural network. [24] discussed the differences between the exact solution and approximation solutions of ODEs. [40] and [55] applied neural approximations of ODEs to dynamic systems. [41] used the B-splines neural network to estimate the solutions of nonlinear ODEs. [37] applied dynamic neural networks to approximate first-order ODEs.

However, there are few studies that use neural networks to solve FDEs. [20] suggested a static neural network to solve FDEs. Since the structure of these neural networks in the above mentioned works are not suitable for FDEs, the approximation accuracy is poor.

In this paper, we apply a new model, the Bernstein neural network, which uses the properties of the Bernstein polynomial, to FDEs. The Bernstein polynomial has good uniform approximation abilities for continuous functions [18]. It also has innumerable drawing properties, homogeneous shape-sustaining approximation, bona fide estimation, and low boundary bias. A very important property of the Bernstein polynomial is that it generates a smooth estimation for equal distance knots [17]. This property is suitable for FDE approximation.

We use two types of neural networks, static and dynamic models, to approximate the solutions of the FDEs. These numerical methods use the generalized differentiability of FDEs. The solutions of the FDEs are then substituted into four ODEs, and the corresponding Bernstein neural networks are applied. Finally, we use some real examples to show the effectiveness of our approximation methods with the Bernstein neural networks.

II. FUZZY DIFFERENTIAL EQUATION FOR UNCERTAIN NONLINEAR SYSTEM MODELING

Consider the following controlled unknown nonlinear system

$$\dot{x} = f_1(x_1, u, t) \tag{1}$$

where $f_1(x_1, u)$ is the unknown vector function, $x_1 \in \Re^n$ is an internal state vector, and $u \in \Re^m$ is the input vector.

In this paper, we use the following FDE to model the uncertain nonlinear system (1),

$$\frac{d}{dt}x = f(x, u) \tag{2}$$

where $x \in \Re^n$ is the fuzzy variable that corresponds to the state x_1 in (1), $f(t, x)$ is a fuzzy vector function that relates to $f_1(x_1, u)$, and $\frac{d}{dt}x$ is the fuzzy derivative. Here, the uncertainties of the nonlinear system (1) are in the sense of fuzzy logic. They are defined as follows.

Definition 1: If x is: 1) normal, there exists $\zeta_0 \in \mathbb{R}$ in such a manner that $x(\zeta_0) = 1$; 2) convex, $x(\lambda\zeta + (1-\lambda)\zeta) \geq \min\{x(\zeta), x(\xi)\}$, $\forall\zeta, \xi \in \mathbb{R}, \forall\lambda \in [0, 1]$; 3) upper semi-continuous on $\mathbb{R}$, $x(\zeta) \leq x(\zeta_0) + \varepsilon$, $\forall\zeta \in N(\zeta_0)$, $\forall\zeta_0 \in \mathbb{R}$, $\forall\varepsilon > 0$, $N(\zeta_0)$ is a neighborhood; or 4) $x^+ = \{\zeta \in \mathbb{R}, x(\zeta) > 0\}$ is compact, then x is a fuzzy variable, and the fuzzy set is defined as E, $x \in E : \mathbb{R} \to [0, 1]$.

The fuzzy variable x can also be represented as

$$x = A\left(\underline{x}, \bar{x}\right) \tag{3}$$

where $\underline{x}$ is the lower-bound variable, $\bar{x}$ is the upper-bound variable, and A is a continuous function. The membership functions are utilized to implicate the fuzzy variable x. The best known membership functions are the triangular function

$$x(\zeta) = F(a, b, c) = \begin{cases} \dfrac{\zeta - a}{b - a} & a \leq \zeta \leq b \\ \dfrac{c - \zeta}{c - b} & b \leq \zeta \leq c \\ 0 & \text{otherwise} \end{cases} \tag{4}$$

and trapezoidal function

$$x(\zeta) = F(a, b, c, d) = \begin{cases} \dfrac{\zeta - a}{b - a} & a \leq \zeta \leq b \\ \dfrac{d - \zeta}{d - c} & c \leq \zeta \leq d \\ 1 & b \leq \zeta \leq c \\ 0 & \text{otherwise} \end{cases} \tag{5}$$

For the crisp variable, the fuzzy variable x possess three essential operations: $\oplus$, $\ominus$ and $\odot$, which signify sum, subtract, and multiply, respectively.

The fuzzy variable x that contains the dimension of ζ is dependent on the membership functions, where (4) includes three variables and (5) includes four variables. To demonstrate the consistency of operations, the application initially lies within the $\alpha-$level operation of the fuzzy number.

A fuzzy number x associates with a real value with α-level as

$$[x]^\alpha = \{a \in \mathbb{R} : x(a) \geq \alpha\} \tag{6}$$

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

If $x, y \in E$, $\lambda \in \mathbb{R}$, the fuzzy operations are as follows:
Sum,

$$[x \oplus y]^\alpha = [x]^\alpha + [y]^\alpha = [\underline{x}^\alpha + \underline{y}^\alpha, \bar{x}^\alpha + \bar{y}^\alpha] \tag{7}$$

subtract,

$$[x \ominus y]^\alpha = [x]^\alpha - [y]^\alpha = [\underline{x}^\alpha - \underline{y}^\alpha, \bar{x}^\alpha - \bar{y}^\alpha] \tag{8}$$

or multiply,

$$\underline{z}^\alpha \leq [x \odot y]^\alpha \leq \bar{z}^\alpha \text{ or } [x \odot y]^\alpha = A\left(\underline{z}^\alpha, \bar{z}^\alpha\right) \tag{9}$$

where $\underline{z}^\alpha = \underline{x}^\alpha \underline{v}^1 + \underline{x}^1\underline{y}^\alpha - \underline{x}^1\underline{y}^1$, $\bar{z}^\alpha = \bar{x}^\alpha\bar{y}^1 + \bar{x}^1\bar{y}^\alpha - \bar{x}^1\bar{y}^1$, and $\alpha \in [0, 1]$.

Therefore, $[x]^0 = x^+ = \{\zeta \in \mathbb{R}, x(\zeta) > 0\}$. Since $\alpha \in [0, 1]$, $[x]^\alpha$ is a bounded interval such that $\underline{x}^\alpha \leq [x]^\alpha \leq \bar{x}^\alpha$. The α-level of x in the midst of $\underline{x}^\alpha$ and $\bar{x}^\alpha$ is given as

$$[x]^\alpha = A\left(\underline{x}^\alpha, \bar{x}^\alpha\right) \tag{10}$$

Definition 2: The fuzzy derivative of f at x_0 is expressed as

$$\frac{d}{dt}f(x_0) = \lim_{h\to 0}\frac{1}{h}[f(x_0 + h) \ominus_{gH} f(x_0)] \tag{11}$$

where $\ominus_{gH}$ is the Hukuhara difference [9], defined by

$$x \ominus_{gH} y = z \Longleftrightarrow \begin{cases} 1)\ x = y \oplus z \\ 2)\ y = x \oplus (-1)z \end{cases} \tag{12}$$

The $\alpha-$level of the fuzzy derivative is

$$f(x, \alpha) = [\underline{f}(x, \alpha), \bar{f}(x, \alpha)]$$

where $x \in E$ for each $\alpha \in [0, 1]$.

If we apply the $\alpha-$level (10) to $f(x,u)$ in (2)

$$[x \ominus_{gH} y]^\alpha = [\min\{\underline{x}^\alpha - \underline{y}^\alpha, \bar{x}^\alpha - \bar{y}^\alpha\}, \\ \max\{\underline{x}^\alpha - \underline{y}^\alpha, \bar{x}^\alpha - \bar{y}^\alpha\}]$$

Then, we obtain two functions: $\underline{f}\left[u, \underline{x}(\zeta,\alpha), \bar{x}(\zeta,\alpha)\right]$ and $\bar{f}\left[u, \underline{x}(\zeta,\alpha), \bar{x}(\zeta,\alpha)\right]$. Thus, the fuzzy differential equation (2) can be equivalent to the following four ordinary differential equations (ODE)

$$1)\begin{cases} \frac{d}{dt}\underline{x} = \underline{f}\left[u, \underline{x}(\zeta,\alpha), \bar{x}(\zeta,\alpha)\right] \\ \frac{d}{dt}\bar{x} = \bar{f}\left[u, \underline{x}(\zeta,\alpha), \bar{x}(\zeta,\alpha)\right] \end{cases}$$
$$2)\begin{cases} \frac{d}{dt}\underline{x} = \bar{f}\left[u, \underline{x}(\zeta,\alpha), \bar{x}(\zeta,\alpha)\right] \\ \frac{d}{dt}\bar{x} = \underline{f}\left[u, \underline{x}(\zeta,\alpha), \bar{x}(\zeta,\alpha)\right] \end{cases} \quad (13)$$

The fuzzy model of (1) can be regarded as four ordinary differential equations (13).

In this paper, we use the fuzzy differential equation (2) to model the uncertain nonlinear system (1), such that the output of the plant x can follow the plant output x_1,

$$\min_f \|x - x_1\| \quad (14)$$

This modeling object can be considered as finding $\bar{f}$ and $\underline{f}$ in the fuzzy equations of (13), or as fining the solutions of these models. It is impossible to obtain analytical solutions, but in this paper, we use neural networks to approximate them, as shown in Figure 1.

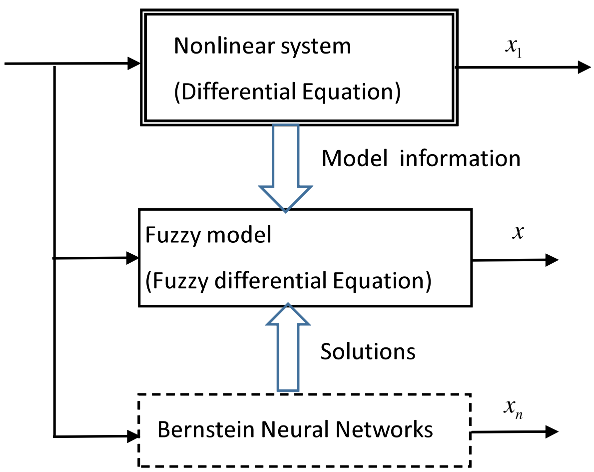


FIGURE 1. Nonlinear system modeling with fuzzy differential equations.

The following theorems give theoretical support to nonlinear system modeling via fuzzy differential equations.

Theorem 1: If the fuzzy function f and its derivative $\frac{\partial f}{\partial x}$ are on the rectangle $[-p,p] \times [-q,q]$, where $p,q \in E$, E is a fuzzy set and there exists an unique fuzzy solution for the following fuzzy differential equation

$$\frac{d}{dt}x = f(t,x), \quad x(t_0) = x_0 \quad (15)$$

for all $t \in (-b,b)$, $b \le p$

Proof: We utilize Picard's iteration technique [10] to develop a sequence of fuzzy functions $\varphi_n(t)$ as

$$\varphi_{n+1}(t) = \varphi_0 \oplus \int_0^t f(s,\varphi_n(s))ds$$
$$= \varphi_0 \ominus_H (-1)\int_0^t f(s,\varphi_n(s))ds$$

We first validate that $\varphi_n(t)$ is continuous and prevails for all n. Obviously, if $\varphi_n(t)$ prevails, then $\varphi_{n+1}(t)$ also prevails as

$$\varphi_{n+1}(t) = \varphi_0 \oplus \int_0^t f(s,\varphi_n(s))ds$$
$$= \varphi_0 \ominus_H (-1)\int_0^t f(s,\varphi_n(s))ds$$

Since f is continuous, there exists $N \in E$ such that $|f(t,x)| \le N$ for all $t \in [-p,p]$, as well as all $x \in [-q,q]$. If we set $t \in [-b,b]$ for $b \le \min(q/N,p)$, then it is possible

$$\|\varphi_{n+1} \ominus \varphi_0\| = \|\int_0^t f(s,\varphi_n(s))ds\| \le N|t| \le Nb \le q$$

This validates that $\varphi_{n+1}(t)$ acquires values in $[-q,q]$. Because

$$\varphi_n(t) = \sum_{k=1}^{n}(\varphi_n(t) \ominus \varphi_{n-1}(t))$$

for any $\gamma < 1$, we select $t \in (-b,b)$ such that $|\ \varphi_k(t) \ominus \varphi_{k-1}(t)\ | \le \gamma^k$ for all k. This signifies that there exists $\gamma < 1$ [33]

$$|\ \varphi_k(t) \ominus \varphi_{k-1}(t)\ | \le \gamma^k$$

From the mean value theorem [48],

$$\varphi_k(t) \ominus \varphi_{k-1}(t) = \int_0^t [f(s,\varphi_{k-1}(s)) \ominus f(s,\varphi_{k-2}(s))]ds$$

Applying the mean value theorem to the fuzzy function $h(x) = f(s,x)$ at the two points $\varphi_{k-1}(s)$ and $\varphi_{k-2}(s)$,

$$h(\varphi_{k-1}(s)) \ominus h(\varphi_{k-2}(s)) = h'(\psi_k(s))(\varphi_{k-1}(s)) \ominus \varphi_{k-2}(s))$$

Taking into consideration $h'(x) = \frac{\partial f}{\partial x}$, we obtain

$$\varphi_k(t) \ominus \varphi_{k-1}(t) = \int_0^t \frac{\partial f}{\partial x}(s,\psi_k(s))(\varphi_{k-1}(s) \ominus \varphi_{k-2}(s))ds \quad (16)$$

Because $|\ \varphi_{k-1}(s) \ominus \varphi_{k-2}(s)\ | \le \gamma^{k-1}$ for $s \le t$, and $b < \gamma/N$, by substituting the above relation in (16) and bounding $\frac{\partial f}{\partial x}$ by N we have ,

$$|\ \varphi_k(t)) \ominus \varphi_{k-1}(t)\ | \le \int_0^t N\gamma^{k-1}ds = Nt\gamma^{k-1} \le Nb\gamma^{k-1}$$

To validate that x is continuous, it is necessary to show that for any given $\epsilon > 0$ there exists a $\delta > 0$ such that $|\ t_2 - t_1\ | < \delta$

implies $| \varphi(t_2) \ominus \varphi(t_1) | < \epsilon$. For notation convenience, we suppose that $t_1 < t_2$. It follows that

$$\begin{aligned}\varphi(t_2) \ominus \varphi(t_1) &= \lim_{n\to\infty} \varphi_n(t_2) \ominus \lim_{n\to\infty} \varphi_n(t_1) \\ &= \lim_{n\to\infty}(\varphi_n(t_2) \ominus \varphi_n(t_1)) \\ &= \lim_{n\to\infty} \int_{t_1}^{t_2} f(s, \varphi_n(s))ds\end{aligned}$$

There exists N such that $| f(s, x) | \leq N$. Hence

$$| \varphi(t_2) \ominus \varphi(t_1) | \leq \int_{t_1}^{t_2} Nds = N | t_2 - t_1 | \leq N\delta$$

Thus, by selecting $\delta < \epsilon/N$ it is observed that $| \varphi(t_2) \ominus \varphi(t_1) | < \epsilon$. So $\lim_{n\to\infty} \varphi_n(t)$ prevails for all t.

Now we demonstrate that $\lim_{n\to\infty} \varphi_n(t)$ is continuous. Since

$$\begin{aligned}\varphi(t) = \lim_{n\to\infty} \varphi_n(t) &= \lim_{n\to\infty} \int_0^t f(s, \varphi_{n-1}(s))ds \\ &= \int_0^t \lim_{n\to\infty} f(s, \varphi_{n-1}(s))ds \\ &= \int_0^t f(s, \lim_{n\to\infty} \varphi_{n-1}(s))ds\end{aligned}$$

where the last step (moving the limit inside the function) follows from the fact that f is continuous in each variable. Hence it is clear that

$$\varphi(t) = \int_0^t f(s, \varphi(s))ds$$

because all functions are continuous,

$$\frac{d}{dt}\varphi = f(s, \varphi(t))$$

If there exists another solution $\phi(t)$,

$$\varphi(t) \ominus \phi(t) = \int_0^t (f(s, \varphi(t)) \ominus f(s, \phi(t)))ds$$

Since the two functions are different, there exists $\epsilon > 0$ and $| \varphi(t) \ominus \phi(t) | > \epsilon$. We define

$$m = \max_{0\leq t\leq b} | \varphi(t) \ominus \phi(t) |$$

N is the bound for $\frac{\partial f}{\partial x}$. Utilizing the mean value theorem,

$$| \varphi(t) \ominus \phi(t) | \leq \int_0^t N | \varphi(t) - \phi(t) | ds \leq N | t | m \leq Nbm$$

If we select $b < \epsilon/2mN$, it signifies that for all $t < b$, $| \varphi(t) - \phi(t) | < \epsilon/2$, indicating that the least difference is ϵ. Therefore, there exists a unique fuzzy solution. ∎

Theorem 2: If the following fuzzy differential equation

$$\frac{d}{dt}x = f(t, x) \tag{17}$$

where $f \in \bar{J}_{ab}$ and $\bar{J}_{ab}$ is the set of linear strongly bounded operators, for every operators f there exists a function $\tau \in L([a, b]; E_+)$ such that $|f(\nu)(t)| \leq \tau(t)\|\nu\|_G$, $t \in [a, b]$ and $\nu \in G([a, b]; E)$, and there prevail $f_0, f_1 \in \varphi_{ab}$, where φ_{ab} is a set of linear operators $f \in \bar{J}_{ab}$ from the set $G([a, b]; E_+)$ to the set $L([a, b]; E_+)$, such that

$$\begin{aligned}&|\underline{f}(t, \underline{\nu}, \overline{\nu}) + \underline{f}_1(t, \underline{\nu}, \overline{\nu})| \leq \underline{f}_0(t, |\underline{\nu}|, |\overline{\nu}|), \quad t \in [a, b] \\ &|\overline{f}(t, \underline{\nu}, \overline{\nu}) + \overline{f}_1(t, \underline{\nu}, \overline{\nu})| \leq \overline{f}_0(t, |\underline{\nu}|, |\overline{\nu}|), \quad t \in [a, b]\end{aligned} \tag{18}$$

then (15) has an unique solution.

Proof: If x is a solution of (17) and $-\frac{1}{2}f_1 \in J_{ab}(a)$,

$$\frac{d}{dt}\beta = -\frac{1}{2}f_1(t, \beta) \oplus f_0(t, |x|) \oplus \frac{1}{2}f_1(t, |x|) \tag{19}$$

contains a unique solution β. Moreover, as $f_0, f_1 \in \varphi_{ab}$

$$\begin{aligned}&\underline{\beta}(t) \geq 0, \quad t \in [a, b] \\ &\overline{\beta}(t) \geq 0, \quad t \in [a, b]\end{aligned} \tag{20}$$

According to (18) and the condition $f_1 \in \varphi_{ab}$, from (19) we have

$$\begin{aligned}&\frac{d}{dt}\underline{\beta} \geq -\frac{1}{2}\underline{f}_1(t, \underline{\beta}, \overline{\beta}) + \underline{f}(t, \underline{x}, \overline{x}) + \frac{1}{2}\underline{f}_1(t, \underline{x}, \overline{x}) \\ &\frac{d}{dt}\overline{\beta} \geq -\frac{1}{2}\overline{f}_1(t, \underline{\beta}, \overline{\beta}) + \overline{f}(t, \underline{x}, \overline{x}) + \frac{1}{2}\overline{f}_1(t, \underline{x}, \overline{x})\end{aligned}$$

thus $t \in [a, b]$

$$\begin{aligned}&\frac{d}{dt}(-\underline{\beta}) \leq -\frac{1}{2}\underline{f}_1(t, -\underline{\beta}, -\overline{\beta}) + \underline{k}(t, \underline{x}, \overline{x}) + \frac{1}{2}\underline{k}_1(t, \underline{x}, \overline{x}) \\ &\frac{d}{dt}(-\overline{\beta}) \leq -\frac{1}{2}\overline{f}_1(t, -\underline{\beta}, -\overline{\beta}) + \overline{f}(t, \underline{x}, \overline{x}) + \frac{1}{2}\overline{f}_1(t, \underline{x}, \overline{x})\end{aligned}$$

The last two inequalities are due to the presumption $-\frac{1}{2}f_1 \in J_{ab}(a)$

$$\begin{aligned}&|\underline{x}(t)| \leq \underline{\beta}(t) \quad t \in [a, b] \\ &|\overline{x}(t)| \leq \overline{\beta}(t) \quad t \in [a, b]\end{aligned} \tag{21}$$

According to (21) and the conditions $f_0, f_1 \in \varphi_{ab}$, (19) results in

$$\begin{aligned}&\frac{d}{dt}\underline{\beta} \leq \underline{f}_0(t, \underline{\beta}, \overline{\beta}), \quad t \in [a, b] \\ &\frac{d}{dt}\overline{\beta} \leq f_0(t, \underline{\beta}, \overline{\beta}), \quad t \in [a, b]\end{aligned}$$

As $f_0 \in J_{ab}(a)$, the last inequality with $\beta(a) = 0$ yields $\underline{\beta}(t) \leq 0$ and $\overline{\beta}(t) \leq 0$ for $t \in [a, b]$. (20) implies $\beta \equiv 0$. Thus, based on (21) we have $x \equiv 0$. ∎

In fact, the nonlinear system can be modeled by the neural network directly. However, this data-driven black box identification method does not use the model information. Conversely, the fuzzy differential equation uses the model information of the nonlinear system, such as the brief form of the differential equation.

III. SOLVING FUZZY DIFFERENTIAL EQUATION WITH NEURAL NETWORKS

In general, it is difficult to solve the four equations (13) or (2). In this paper, we use a special neural network, the Bernstein neural network, to approximate the solutions of the fuzzy differential equation (2).

The Bernstein neural network use the following Bernstein polynomial,

$$B(x_1, x_2) = \sum_{i=0}^{N} \sum_{j=0}^{M} (Ni) \binom{M}{j} W_{i,j} x_{1i} (T - x_{1i})^{N-i} x_{2j} (1 - x_{2j})^{M-j} \quad (22)$$

where $\binom{N}{i} = \frac{N!}{i!(N-i)!}$, $\binom{M}{j} = \frac{M!}{j!(M-j)!}$, $W_{i,j}$ is the coefficient.

This two-variables polynomial can be regarded as a neural network, which has two inputs x_{1i} and x_{2j}, and one output y,

$$y = \sum_{i=0}^{N} \sum_{j=0}^{M} \lambda_i \gamma_j W_{i,j} x_{1i} (T - x_{1i})^{N-i} x_{2j} (1 - x_{2j})^{M-j} \quad (23)$$

where $\lambda_i = \binom{N}{i}$, $\gamma_j = \binom{M}{j}$.

Because the Bernstein neural network (23) has similar forms as (13), we use the Bernstein neural network (23) to approximate the solutions of four ODEs in (13).

If x_1 and x_2 in the Bernstein polynomial are defined as the time interval t and the α-level, respectively, the solution of (2) in the form of the Bernstein neural network is

$$x_m(t, \alpha) = x_m(0, \alpha) \oplus t \sum_{i=0}^{N} \sum_{j=0}^{M} \lambda_i \gamma_j W_{i,j} t_i (T - t_i)^{N-i} \times \alpha_j (1 - \alpha_j)^{M-j} \quad (24)$$

where $x_m(0, \alpha)$ is the initial condition of the solution.

Thus, the derivative of (23) is

$$\begin{aligned} &1) \begin{cases} \frac{d}{dt} \underline{x}_m = C_1 + C_2 \\ \frac{d}{dt} \bar{x}_m = D_1 + D_2 \end{cases} \\ &2) \begin{cases} \frac{d}{dt} \underline{x}_m = C_1 + C_2 \\ \frac{d}{dt} \bar{x}_m = D_1 + D_2 \end{cases} \end{aligned} \quad (25)$$

where $t \in [0, T]$, $\alpha \in [0, 1]$, $t_k = kh$, $h = \frac{T}{k}$, $k = 1, \ldots, N$, $\alpha_j = jh_1$, $h_1 = \frac{1}{M}$, $j = 1, \ldots, M$,

$$C_1 = \sum_{i=0}^{N} \sum_{j=0}^{M} \lambda_i \gamma_j \underline{W}_{i,j} t_i (T - t_i)^{N-i} \alpha_j (1 - \alpha_j)^{M-j}$$

$$D_1 = \sum_{i=0}^{N} \sum_{j=0}^{M} \lambda_i \gamma_j \overline{W}_{i,j} t_i (T - t_i)^{N-i} \alpha_j (1 - \alpha_j)^{M-j}$$

$$C_2 = t_k \sum_{i=0}^{N} \sum_{j=0}^{M} \lambda_i \gamma_j \underline{W}_{i,j} [i t_{i-1,j} (T - t_i)^{N-i} - (N - i)\, t_{i,j} (T - t_i)^{N-i-1}] \alpha_j^i (1 - \alpha_j)^{M-j}$$

$$D_2 = t_k \sum_{i=0}^{N} \sum_{j=0}^{M} \lambda_i \gamma_j \overline{W}_{i,j} [i t_{i-1,j} (T - t_i)^{N-i} - (N - i)\, t_{i,j} (T - t_i)^{N-i-1}] \alpha_j^i (1 - \alpha_j)^{M-j}$$

The above equations can be regarded as the neural network form, as shown in Figure 2. The output is

$$N(t, \alpha) = \sum_{i=0}^{N} \sum_{j=0}^{M} (a_{i,j} \lambda_i f_i^1(t) f_i^2(t) \gamma_j g_j^1(\alpha) g_j^2(\alpha)))$$

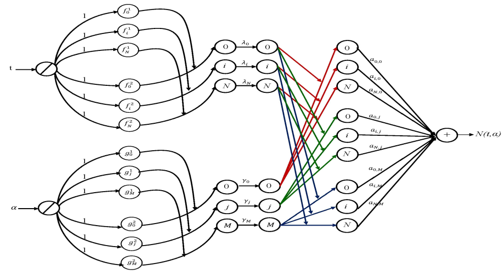


FIGURE 2. Bernstein neural network.

where $f_i^1 = t^i$, $f_i^2 = (T - t)^{N-i}$, $\lambda_i = \frac{1}{T^N} \binom{N}{i}$, $g_j^1 = \alpha^j$, $g_j^2 = (1 - \alpha)^{M-j}$, and $\gamma_j = \binom{M}{j}$.

We define the training errors between (25) and (13) as

$$\begin{aligned} &1) \begin{cases} \underline{e}_1 = C_1 + C_2 - \underline{f} \\ \bar{e}_1 = D_1 + D_2 - \bar{f} \end{cases} \\ &2) \begin{cases} \underline{e}_2 = C_1 + C_2 - \bar{f} \\ \bar{e}_2 = D_1 + D_2 - \underline{f} \end{cases} \end{aligned} \quad (26)$$

The standard back-propagation learning algorithm is utilized to update the weights with the above training errors

$$\underline{W}_{i,j}(k+1) = \underline{W}_{i,j}(k) - \eta_1 \left(\frac{\partial \underline{e}_1^2}{\partial \underline{W}_{i,j}} + \frac{\partial \bar{e}_1^2}{\partial \underline{W}_{i,j}} \right)$$

$$\overline{W}_{i,j}(k+1) = \overline{W}_{i,j}(k) - \eta_2 \left(\frac{\partial \underline{e}_2^2}{\partial \overline{W}_{i,j}} + \frac{\partial \bar{e}_2^2}{\partial \overline{W}_{i,j}} \right) \quad (27)$$

where η_1 and η_2 are positive learning rates.

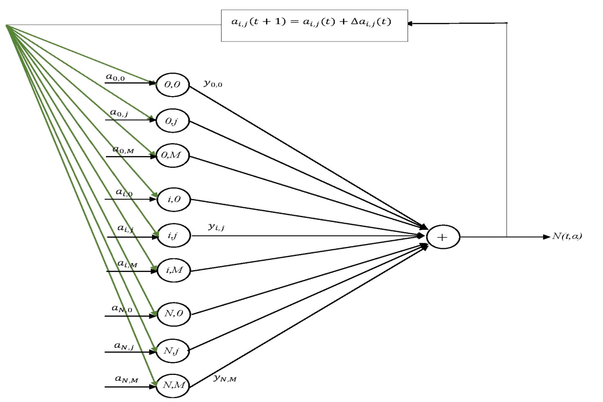


FIGURE 3. A dynamic Bernstein neural network.

The momentum terms, $\gamma \Delta \underline{W}_{i,j}(k-1)$ and $\gamma \Delta \bar{W}_{i,j}(k-1)$, can be added to stabilize the training process. The above Bernstein neural network can be converted to a recurrent (dynamic) form, as shown in Figure 3. The dynamic Bernstein neural network is

$$\begin{cases} \frac{d}{dt} \underline{x}_m(t, \alpha) = \underline{P}(t, \alpha) A(\underline{x}_m(t, \alpha), \bar{x}_m(t, \alpha)) + \underline{Q}(t, \alpha) \\ \frac{d}{dt} \bar{x}_m(t, \alpha) = \overline{P}(t, \alpha) A(\underline{x}_m(t, \alpha), \bar{x}_m(t, \alpha)) + \overline{Q}(t, \alpha) \end{cases} \quad (28)$$

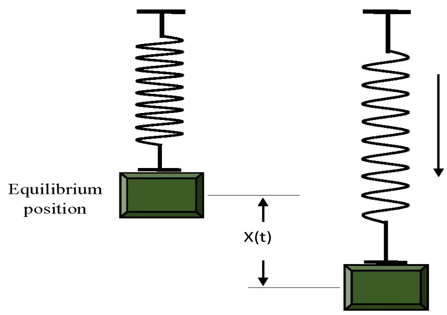


FIGURE 4. Vibration mass.

Obviously, this dynamic network has the form of

$$f(t,x) = P(t)x + Q(t)$$

and it is closed to (2). The training algorithm is similar to (27) and only the training errors are changed as

$$1)\begin{cases}\underline{e}_1 = C_1 + C_2 - \underline{P}A(\underline{x}_m, \bar{x}_m) - \underline{Q} \\ \bar{e}_1 = D_1 + D_2 - \bar{P}A(\underline{x}_m, \bar{x}_m) - \bar{Q}\end{cases}$$

$$2)\begin{cases}\underline{e}_2 = C_1 + C_2 - \bar{P}A(\underline{x}_m, \bar{x}_m) - \bar{Q} \\ \bar{e}_2 = D_1 + D_2 - \underline{P}A(\underline{x}_m, \bar{x}_m) - \underline{Q}\end{cases} \quad (29)$$

IV. APPLICATIONS

In this section, we use several real applications to show how to use fuzzy differential equations (FDEs) and Bernstein neural networks (BNNs) to model the nonlinear systems.

Example 1: The vibration mass system shown in Figure 4 can be modeled by the ordinary differential equation (ODE),

$$\frac{d}{dt}v(t) = \frac{k}{m}x(t), \quad v(t) = \frac{d}{dt}x(t) \quad (30)$$

where the spring constant $k = 1$, and the mass m is changeable in fuzzy number (0.75, 1.125). The ODE (30) becomes the FDE, and $x(t)$ becomes a fuzzy variable. If the initial position is $x(0) = (0.75+0.25\alpha, 1.125-0.125\alpha)$, $\alpha \in [0, 1]$, then the exact solutions of (30) are [27]

$$x(t,\alpha) = \left[(0.75 + 0.25\alpha)e^t, (1.125 - 0.125\alpha)e^t\right] \quad (31)$$

where $t \in [0, 1]$. We use the static Bernstein neural network (24), SNN, to approximate the solution (31)

$$\begin{cases}\underline{x}_m(t,\alpha) = (0.75 + 0.25\alpha) \\ \quad +t\sum_{i=0}^{N}\sum_{j=0}^{M}\lambda_i\gamma_j\underline{W}_{i,j}t_i(T-t_i)^{N-i}\alpha_j(1-\alpha_j)^{M-j} \\ \bar{x}_m(t,\alpha) = (1.125 - 0.125\alpha) \\ \quad +t\sum_{i=0}^{N}\sum_{j=0}^{M}\lambda_i\gamma_j\bar{W}_{i,j}t_i(T-t_i)^{N-i}\alpha_j(1-\alpha_j)^{M-j}\end{cases}$$

We also use the dynamic Bernstein neural network (28), DNN, to approximate the solutions. The learning rates are $\eta = 0.01$ and $\gamma = 0.01$. To compare our results, we use the other two popular methods: Max-Min Euler method and Average Euler method [53]. The results are compared in Table 1. Corresponding solution plots are shown in Figure 5.

TABLE 1. Approximation errors.

α	SNN	DNN	Max-Min Euler	Average Euler
0	[0.0601,0.1098]	[0.0207,0.0601]	[0.0934,0.1401]	[0.2054,0.4390]
0.2	[0.0658,0.1067]	[0.0241,0.0612]	[0.0996,0.1370]	[0.1394,0.3761]
0.8	[0.0791,0.0891]	[0.0328,0.0499]	[0.1183,0.1276]	[0.0586,0.1874]
1	[0.0921,0.0921]	[0.0534,0.0534]	[0.1246,0.1246]	[0.1246,0.1246]

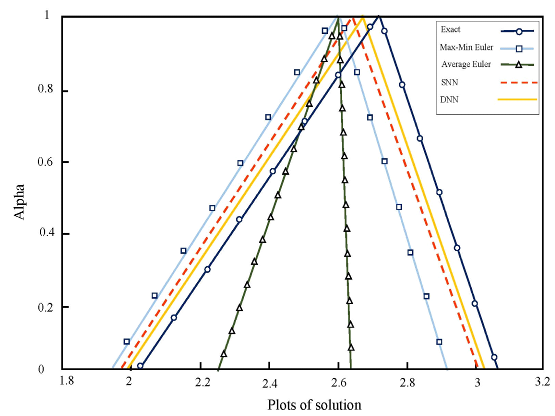


FIGURE 5. Comparison plot of SNN, DNN, Max-Min Euler, Average Euler and the exact solution.

Example 2: The heat treatment system in welding can be modeled as [14]:

$$\frac{d}{dt}x(t) = 3Ax^2(t) \quad (32)$$

where the transfer area A is uncertain as $A = (1+\alpha, 3-\alpha)$, $\alpha \in [0, 1]$. Therefore, (32) is a fuzzy differential equation. The initial condition is $x(0) = (0.5\sqrt{\alpha}, 0.2\sqrt{1-\alpha}+0.5)$. The static Bernstein neural network (24) has the form of

$$\begin{cases}\underline{x}_m(t,\alpha) = 0.5\sqrt{\alpha} \\ \quad +t\sum_{i=0}^{N}\sum_{j=0}^{M}\lambda_i\gamma_j\underline{W}_{i,j}t_i(T-t_i)^{N-i}\alpha_j(1-\alpha_j)^{M-j} \\ \bar{x}_m(t,\alpha) = 0.2\sqrt{1-\alpha} + 0.5 \\ \quad +t\sum_{i=0}^{N}\sum_{j=0}^{M}\lambda_i\gamma_j\bar{W}_{i,j}t_i(T-t_i)^{N-i}\alpha_j(1-\alpha_j)^{M-j}\end{cases}$$

With the learning rates $\eta = 0.002$ and $\gamma = 0.002$. The approximation results are shown in Table 2.

TABLE 2. Approximation errors of BNN.

α	SNN	DNN
0	[0.0511,0.0754]	[0.0224,0.0381]
0.1	[0.0402,0.0623]	[0.0203,0.0362]
0.8	[0.0373,0.0509]	[0.0157,0.0362]
0.9	[0.0401,0.0635]	[0.0202,0.0408]
1	[0.0394,0.0394]	[0.0167,0.0167]

Example 3: A tank system is shown in Figure 6. Assume $I = t + 1$ to be inflow disturbances of the tank, which generates vibration in liquid level x, where $R = 1$ is the flow

TABLE 3. Solutions of different method.

α	SNN	DNN	Neural network
0	[0.0387, 0.0884]	[0.0101, 0.0398]	[0.0701, 0.1012]
0.2	[0.0451, 0.0841]	[0.0225, 0.0575]	[0.0771, 0.1207]
0.8	[0.0544, 0.0635]	[0.0144, 0.0289]	[0.0649, 0.0812]
1	[0.0554, 0.0554]	[0.0311, 0.0311]	[0.0901, 0.0901]

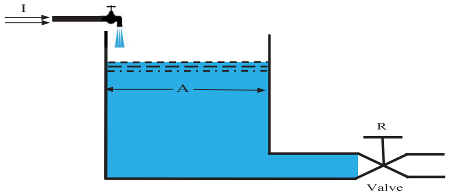


FIGURE 6. Liquid tank system.

obstruction that can be curbed using the valve. $A = 1$ is the cross section of the tank. The liquid level can be described as [51],

$$\frac{d}{dt}x(t) = -\frac{1}{AR}x(t) + \frac{I}{A} \tag{33}$$

The initial condition is $x(0) = (0.96 + 0.04\alpha, 1.01 - 0.01\alpha)$. The static Bernstein neural network (24) has the form of

$$\begin{cases} \underline{x}_m(t,\alpha) = (0.96 + 0.04\alpha) \\ \quad +t\sum_{i=0}^{N}\sum_{j=0}^{M}\lambda_i\gamma_j\underline{W}_{i,j}t_i(T-t_i)^{N-i}\alpha_j(1-\alpha_j)^{M-j} \\ \overline{x}_m(t,\alpha) = (1.01 - 0.01\alpha) \\ \quad +t\sum_{i=0}^{N}\sum_{j=0}^{M}\lambda_i\gamma_j\bar{W}_{i,j}t_i(T-t_i)^{N-i}\alpha_j(1-\alpha_j)^{M-j} \end{cases}$$

where $t \in [0, 1]$. We also use the dynamic Bernstein neural network (28) to approximate the solutions. To compare our results, we use the other generalization of the neural network method [20]. The comparison results are shown in Table 4. The specifications quoted here are $\eta = 0.001$ and $\gamma = 0.001$. Corresponding errors are shown in Figure 7.

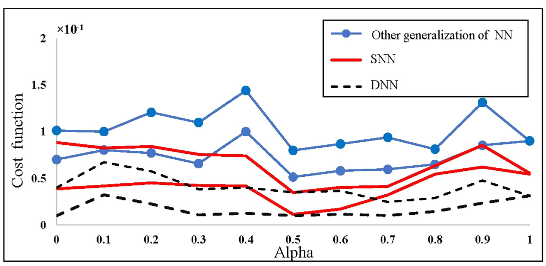


FIGURE 7. Errors between the exact solution and the approximations.

Example 4: A tank with a heating system is shown in Figure 8, where $R = 0.5$ and the thermal capacitance is considered to be $C = 2$. The temperature is x. The model is [46],

$$\frac{d}{dt}x(t) = -\frac{1}{RC}x(t) \tag{34}$$

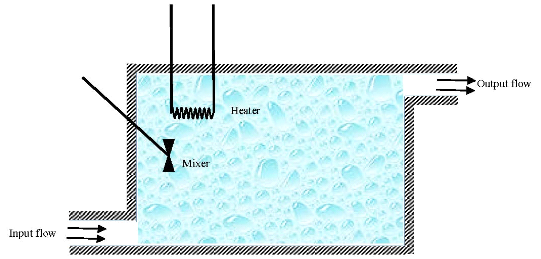


FIGURE 8. Thermal system.

where $t \in [0, 1]$ and x is the amount of sinking in each moment. If the initial position is $u(0) = (\alpha - 1, 1 - \alpha)$ and $\alpha \in [0, 1]$, then the exact solutions of the fuzzy differential equation (34) are

$$x(t,\alpha) = [(\alpha - 1)e^t, (1-\alpha)e^t] \tag{35}$$

We use the static Bernstein neural network (24) to approximate the solution (35)

$$\begin{cases} \underline{x}_m(t,\alpha) = (\alpha - 1) \\ \quad +t\sum_{i=0}^{N}\sum_{j=0}^{M}\lambda_i\gamma_j\underline{W}_{i,j}t_i(T-t_i)^{N-i}\alpha_j(1-\alpha_j)^{M-j} \\ \overline{x}_m(t,\alpha) = (1 - \alpha) \\ \quad +t\sum_{i=0}^{N}\sum_{j=0}^{M}\lambda_i\gamma_j\bar{W}_{i,j}t_i(T-t_i)^{N-i}\alpha_j(1-\alpha_j)^{M-j} \end{cases}$$

where $\eta = 0.001$ and $\gamma = 0.001$. We also use the dynamic Bernstein neural network (28) to approximated the solutions. The errors related to SNN and DNN are illustrated in Table 4.

TABLE 4. NN approximation errors.

α	SNN	DNN
0	[0.0407, 0.0604]	[0.0184, 0.0317]
0.1	[0.0351, 0.0578]	[0.0151, 0.0305]
0.2	[0.0334, 0.0523]	[0.0111, 0.0284]
0.8	[0.0282, 0.0417]	[0.0104, 0.0301]
0.9	[0.0253, 0.0501]	[0.0102, 0.0313]
1	[0.0323, 0.0323]	[0.0112, 0.0112]

TABLE 5. Different NNs.

τ	α=0.2,n=10	α=0.2,n=15	α=0.2,n=20
100	[0.0687, 0.1087]	[0.0585, 0.0901]	[0.0487, 0.0831]
300	[0.0404, 0.0814]	[0.0392, 0.0789]	[0.0334, 0.0613]
τ	α=0.5,n=10	α=0.5,n=15	α=0.5,n=20
100	[0.0545, 0.0957]	[0.0416, 0.0852]	[0.0352, 0.0683]
300	[0.0390, 0.0611]	[0.0291, 0.0581]	[0.0267, 0.0411]
τ	α=0.8,n=10	α=0.8,n=15	α=0.8,n=20
100	[0.0389, 0.0855]	[0.0311, 0.0748]	[0.0219, 0.0533]
300	[0.0308, 0.0552]	[0.0206, 0.0498]	[0.0192, 0.0317]

For different number of learning steps $\tau = 100$, $\tau = 200$, and $\tau = 300$, and hidden neurons $n = 10$, $n = 15$, and $n = 20$, the results are shown in Table 5 and Figure 9.

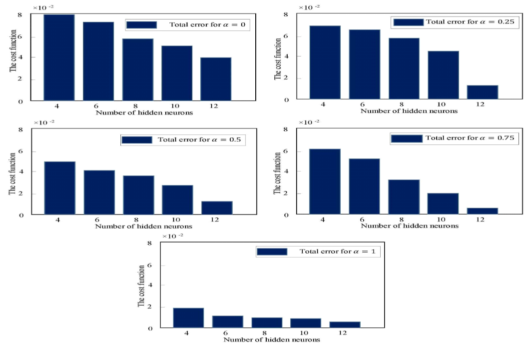


FIGURE 9. Different neural elements for $\tau = 300$.

Both static neural network and dynamic neural network are suitable for solving the fuzzy differential equation. The leaning process of the dynamic Bernstein neural network (28) is faster than the static Bernstein neural network (24). The robustness of (24) is better than (28), because the weights of the dynamic Bernstein neural network are difficult to converge.

V. CONCLUSIONS

In this paper, we use fuzzy differential equations (FDEs) to model unknown nonlinear systems. The existence conditions of FDEs are given. Since the solutions of the fuzzy differential equations are difficult to obtain, we use static and dynamic Bernstein neural networks to approximate the solutions. We first transform the FDEs into four ODEs with Hukuhara differentiability. Then, we construct neural models with the structure of the ODEs. With a modified backpropagation method for the fuzzy variables, the neural networks are trained. Some real examples are given to show the effectiveness of our methods. Future works will involve the application of these methods to fuzzy partial differential equations.

REFERENCES

[1] S. Abbasbandy and T. A. Viranloo, "Numerical solutions of fuzzy differential equations by Taylor method," *J. Comput. Meth. Appl. Math.*, vol. 2, no. 2, pp. 113–124, 2002.

[2] S. Agatonovic-Kustrin and R. Beresford, "Basic concepts of artificial neural network (ANN) modeling and its application in pharmaceutical research," *J. Pharmaceutical Biomed. Anal.*, vol. 22, no. 5, pp. 717–727, Jun. 2000.

[3] M. B. Ahmadi, N. A. Kiani, and N. Mikaeilvand, "Laplace transform formula on fuzzy nth-order derivative and its application in fuzzy ordinary differential equations," *Soft Comput.*, vol. 18, no. 12, pp. 2461–2469, Dec. 2014.

[4] T. Allahviranloo, N. Ahmady, and E. Ahmady, "Numerical solution of fuzzy differential equations by predictor–corrector method," *Inf. Sci.*, vol. 177, no. 7, pp. 1633–1647, Apr. 2007.

[5] T. Allahviranloo, N. Ahmady, and E. Ahmady, "A method for solving nth order fuzzy linear differential equations," *Comput. Math. Appl.*, vol. 86, no. 4, pp. 730–742, 2009.

[6] T. Allahviranloo, N. A. Kiani, and M. Barkhordari, "Toward the existence and uniqueness of solutions of second-order fuzzy differential equations," *Inf. Sci.*, vol. 179, no. 8, pp. 1207–1215, Mar. 2009.

[7] B. Bede, "Note on 'Numerical solutions of fuzzy differential equations by predictor–corrector method,'" *Inf. Sci.*, vol. 178, no. 7, pp. 1917–1922, Apr. 2008.

[8] B. Bede, I. J. Rudas, and A. L. Bencsik, "First order linear fuzzy differential equations under generalized differentiability," *Inf. Sci.*, vol. 177, no. 7, pp. 1648–1662, Apr. 2007.

[9] B. Bede and L. Stefanini, "Generalized differentiability of fuzzy-valued functions," *Fuzzy Sets Syst.*, vol. 230, pp. 119–141, Nov. 2013.

[10] S. S. Behzadi and T. Allahviranloo, "Solving fuzzy differential equations by using picard method," *Iranian J. Fuzzy Syst.*, vol. 13, no. 3, pp. 71–81, 2016.

[11] J. J. Buckley and T. Feuring, "Fuzzy differential equations," *Fuzzy Sets Syst.*, London, UK, vol. 110, no. 1, pp. 43–54, Feb. 2000.

[12] J. J. Buckley and T. Feuring, "Fuzzy initial value problem for Nth-order linear differential equations," *Fuzzy Sets Syst.*, vol. 121, no. 2, pp. 247–255, Jul. 2001.

[13] J. J. Buckley and Y. Hayashi, "Can fuzzy neural nets approximate continuous fuzzy functions?" *Fuzzy Sets Syst.*, vol. 61, no. 1, pp. 43–51, Jan. 1994.

[14] H. B. Cary and S. C. Helzer, *Modern Welding Technology*. Upper Saddle River, NJ, USA: Pearson Education, 2005.

[15] Y. Chalco-Cano and H. Román-Flores, "On new solutions of fuzzy differential equations," *Chaos, Solitons Fractals*, vol. 38, no. 1, pp. 112–119, Oct. 2008.

[16] S. L. Chang and L. A. Zadeh, "On fuzzy mapping and control," *IEEE Trans. Syst., Man, Cybern.*, vol. SMC-2, no. 1, pp. 30–34, Jan. 1972.

[17] S. M. Curtis and S. K. Ghosh, "A variable selection approach to monotonic regression with Bernstein polynomials," *J. Appl. Statist.*, vol. 38, no. 5, pp. 961–976, 2011.

[18] P. J. Davis, *Interpolation and Approximation*. New York, NY, USA: Dover, 1975.

[19] D. Dubois and H. Prade, "Towards fuzzy differential calculus part 3: Differentiation," *Fuzzy Sets Syst.*, vol. 8, no. 3, pp. 225–233, Sep. 1982.

[20] S. Effati and M. Pakdaman, "Artificial neural network approach for solving fuzzy differential equations," *Inf. Sci.*, vol. 180, no. 8, pp. 1434–1457, Apr. 2010.

[21] S. S. Esmaili and A. M. Nasrabadi, "Different initial conditions in fuzzy Tumor model," *J. Biomed. Sci. Eng.*, vol. 3, no. 10, pp. 1001–1005, Oct. 2010.

[22] O. S. Fard, A. Esfahani, and A. V. Kamyad, "On solution of a class of fuzzy BVPs," *Iran. J. Fuzzy Syst.*, vol. 9, no. 1, pp. 49–60, 2012.

[23] G. Feng, "A survey on analysis and design of model-based fuzzy control systems," *IEEE Trans. Fuzzy Syst.*, vol. 14, no. 5, pp. 676–697, Oct. 2006.

[24] C. Filici, "On a neural approximator to ODEs," *IEEE Trans. Neural Netw.*, vol. 19, no. 3, pp. 539–543, Mar. 2008.

[25] M. S. Hashemi and J. Malekinagad, "Series solution of fuzzy wave-like equations with variable coefficients," *J. Intell. Fuzzy Syst.*, vol. 25, no. 2, pp. 415–428, 2013.

[26] F. Hawrra and K. H. Amal, "On fuzzy Laplace transforms for fuzzy differential equations of the third order," *J. Kerbala Univ.*, vol. 11, no. 3, pp. 251–256, 2013.

[27] M. Hazewinkel, *Oscillator Harmonic*. London, UK: Springer, 2001.

[28] E. Hüllermeier, "An approach to modelling and simulation of uncertain dynamical systems," *Int. J. Uncertainty Fuzzyness Knowl.-Based Syst.*, vol. 5, no. 2, pp. 117–137, 1997.

[29] R. Jafari and W. Yu, "Fuzzy control for uncertainty nonlinear systems with dual fuzzy equations," *J. Intell. Fuzzy. Syst.*, vol. 29, no. 3, pp. 1229–1240, 2015.

[30] A. Jafarian and R. Jafari, "Approximate solutions of dual fuzzy polynomials by feed-back neural networks," *J. Soft Comput. Appl.*, vol. 2012, 2012, Art. no. 00005, doi: 10.5899/2012/jsca-00005.

[31] A. Jafarian, R. Jafari, A. K. Golmankhaneh, and D. Baleanu, "Solving fully fuzzy polynomials using feed-back neural networks," *Int. J. Comput. Math.*, vol. 92, no. 4, pp. 742–755, 2015.

[32] A. Jafarian, R. Jafari, M. M. Al Qurashi, and D. Baleanu, "A novel computational approach to approximate fuzzy interpolation polynomials," *SpringerPlus*, vol. 5, no. 1, p. 1428, 2016, doi: 10.1186/s40064-016-3077-5.

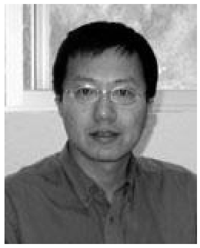
[33] U. Kadak and F. Başar, "On some sets of fuzzy-valued sequences with the level sets," *Contemp. Anal. Appl. Math.*, vol. 1, no. 2, pp. 70–90, 2013.

[34] A. Khastan and K. Ivaz, "Numerical solution of fuzzy differential equations by Nyström method," *Chaos, Solitons. Fractals.*, vol. 41, no. 2, pp. 859–868, Jul. 2009.

[35] A. Khastan, J. J. Nieto, and R. Rodríguez-López, "Periodic boundary value problems for first-order linear differential equations with uncertainty under generalized differentiability," *Inf. Sci.*, vol. 222, pp. 544–558, Feb. 2013.

[36] V. Lakshmikantham and R. N. Mohapatra, *Theory of Fuzzy Differential Equations and Inclusions*. New York, NY, USA: Taylor & Francis, 2003.
[37] H. Lee and I. S. Kang, ''Neural algorithm for solving differential equations,'' *J. Comput. Phys.*, vol. 91, no. 1, pp. 110–131, Nov. 1990.
[38] H.-K. Liu, ''Comparison results of two-point fuzzy boundary value problems,'' *Int. J. Comput. Math. Sci.*, vol. 5, no. 1, pp. 1–7, 2011.
[39] W. A. Lodwick and K. D. Jamison, ''Special issue: Interfaces between fuzzy set theory and interval analysis,'' *Fuzzy Sets Syst.*, vol. 135, no. 1, pp. 1–3, 2003.
[40] A. J. Meade, Jr., and A. A. Fernandez, ''The numerical solution of linear ordinary differential equations by feedforward neural networks,'' *Math. Comput. Model.*, vol. 19, no. 12, pp. 1–25, Jun. 1994.
[41] A. J. Meade, Jr., and A. A. Fernandez, ''Solution of nonlinear ordinary differential equations by feedforward neural networks,'' *Math. Comput. Model.*, vol. 20, no. 9, pp. 19–44, Nov. 1994.
[42] M. Ma, M. Friedman, and A. Kandel, ''Numerical solutions of fuzzy differential equations,'' *Fuzzy Sets Syst.*, vol. 105, no. 1, pp. 133–138, Jul. 1999.
[43] J. J. Nieto, A. Khastan, and K. Ivaz, ''Numerical solution of fuzzy differential equations under generalized differentiability,'' *Nonlinear Anal. Hybrid Syst.*, vol. 3, no. 4, pp. 700–707, Nov. 2009.
[44] J. J. Nieto and R. Rodríguez-López, M. Villanueva-Pesqueira, ''Exact solution to the periodic boundary value problem for a first-order linear fuzzy differential equation with impulses,'' *Fuzzy Optim. Decision Making*, vol. 10, pp. 323–339, Dec. 2011.
[45] S. C. Palligkinis, G. Papageorgiou, and I. T. Famelis, ''Runge–Kutta methods for fuzzy differential equations,'' *Appl. Mathe. Comput.*, vol. 209, no. 1, pp. 97–105, Mar. 2008.
[46] R. H. Pletcher, J. C. Tannehill, and D. Anderson, *Computational Fluid Mechanics and Heat Transfer*. New York, NY, USA: Taylor & Francis, 1997.
[47] M. L. Puri and D. A. Ralescu, ''Differentials of fuzzy functions,'' *J. Math. Anal. Appl.*, vol. 91, no. 2, pp. 552–558, Feb. 1983.
[48] D. Ralescu, ''Toward a general theory of fuzzy variables,'' *J. Math. Anal. Appl.*, vol. 86, no. 1, pp. 176–193, Mar. 1982.
[49] O. S. Fard and N. Ghal-Eh, ''Numerical solutions for linear system of first-order fuzzy differential equations with fuzzy constant coefficients,'' *Inf. Sci.*, vol. 181, no. 20, pp. 4765–4779, Oct. 2011.
[50] L. Stefanini and B. Bede, ''Generalized Hukuhara differentiability of interval-valued functions and interval differential equations,'' *Nonlinear Anal. Theory, Methods Appl.*, vol. 71, nos. 3–4, pp. 1311–1328, Aug. 2009.
[51] V. L. Streeter, E. B. Wylie, and K. W. Bedford, *Fluid Mechanics*. New York, NY, USA: McGraw-Hill, 1998.
[52] T. Takagi and M. Sugeno, ''Fuzzy identification of systems and its applications to modeling and control,'' *IEEE Trans. Syst., Man, Cybern.*, vol. SMC-15, no. 1, pp. 116–132, Jan./Feb. 1985.
[53] S. Tapaswini and S. Chakraverty, ''Euler-based new solution method for fuzzy initial value problems,'' *Int. J. Artif. Intell. Soft Comput.*, vol. 4, no. 1, pp. 58–79, 2014.
[54] M. Wagenknecht, R. Hampel, and V. Schneider, ''Computational aspects of fuzzy arithmetics based on archimedean t-norms,'' *Fuzzy Sets Syst.*, vol. 123, no. 1, pp. 49–62, Oct. 2001.
[55] H. S. Yazdi and R. Pourreza, ''Unsupervised adaptive neural-fuzzy inference system for solving differential equations,'' *Appl. Soft Comput.*, vol. 10, no. 1, pp. 267–275, Jan. 2010.
[56] L. A. Zadeh, ''Toward a generalized theory of uncertainty (GTU)—An outline,'' *Inf. Sci.*, vol. 172, nos. 1–2, pp. 1–40, Jun. 2005.

RAHELEH JAFARI received the B.S. degree from the Shabestari of Islamic Azad University, Iran, in 2003, and the M.S. degree in applied mathematics from the Islamic Azad University of Arak, in 2010. She is currently pursuing the Ph.D. degree with the Departamento de Control Automático, CINVESTAV-IPN (National Polytechnic Institute). Her current research interests include fuzzy logic, fuzzy equation, neural networks, and nonlinear system modeling.



WEN YU (M'97–SM'04) received the B.S. degree from Tsinghua University, Beijing, China, in 1990, and the M.S. and Ph.D. degrees in electrical engineering from Northeastern University, Shenyang, China, in 1992 and 1995, respectively. From 2002 to 2003, he held research positions with the Mexican Institute of Petroleum. He was a Senior Visiting Research Fellow with Queen's University Belfast, Belfast, U.K., from 2006 to 2007, and a Visiting Associate Professor with the University of California at Santa Cruz, Santa Cruz, from 2009 to 2010. Since 1996, he has been with the CINVESTAV-IPN (National Polytechnic Institute), Mexico City, Mexico, where he is currently a Professor and the Department Chair of the Automatic Control Department. He has authored over 100 research papers in reputed journals. His Google Scholar h-index is 33 and the citation number is 4100. He serves as an Associate Editor of the IEEE Transactions on Cybernetics, the *Neurocomputing*, and the *Journal of Intelligent and Fuzzy Systems*. He is a member of the Mexican Academy of Sciences.

Xiaoou Li received the B.S. and the Ph.D. degree in applied mathematics and electrical engineering from Northeastern University, China, in 1991 and 1995, respectively. From 1995 to 1997, she was a Lecturer of Electrical Engineering with the Department of Automatic Control, Northeastern University. From 1998 to 1999, she was an Associate Professor of Computer Science with the Centro de Instrumentos-UNAM. Since 2000, she has been a Professor of Computer Science with the SecciÃşn de ComputaciÃşn, Departamento de IngenierÃa ElÃľctrica, CINVESTAV-IPN, Mexico. Her current research interests include Petri net theory and application, neural networks, advanced database systems, computer integrated manufacturing, and discrete event systems.