Modeling and Control of Uncertain Nonlinear Systems

Raheleh Jafari
Centre for Artificial Intelligence Research (CAIR)
University of Agder
Grimstad, Norway
Email: jafari3339@yahoo.com

Sina Razvarz
Departamento de Control Automático
CINVESTAV-IPN
(National Polytechnic Institute)
Mexico City, Mexico
Email: srazvarz@yahoo.com

Alexander Gegov
School of Computing
University of Portsmouth
Buckingham Building
Portsmouth PO1 3HE, UK
Email: alexander.gegov@port.ac.uk

Satyam Paul
School of Engineering and Sciences
Tecnologico de Monterrey
Monterrey, Nuevo Leon
Mexico, 64849
Email: satyam.controlsystems@gmail.com

***Abstract*—A survey of the methodologies associated with the modeling and control of uncertain nonlinear systems has been given due importance in this paper. The basic criteria that highlights the work is relied on the various patterns of techniques incorporated for the solutions of fuzzy equations that corresponds to fuzzy controllability subject. The solutions which are generated by these equations are considered to be the controllers. Currently, numerical techniques have come out as superior techniques in order to solve these types of problems. The implementation of neural networks technique is contributed in the complex way of dealing the appropriate coefficients and solutions of the fuzzy systems.**

I. Introduction

In recent days, many methods involving uncertainties have used fuzzy numbers [20][21][22][34][35][36], where the uncertainties of the system are represented by fuzzy coefficients. Fuzzy method is a highly favorable tool for uncertain nonlinear system modeling. The fuzzy models approximate uncertain nonlinear systems with several linear piecewise systems (Takagi-Sugeno method) [40]. Mamdani models use fuzzy rules to achieve a good level of approximation of uncertainties [26]. When the parameter of an equation are changeable in the manner of fuzzy set, this equation becomes a fuzzy equation [12].

In comparison with the normal systems, fuzzy equations are considered to be very noncomplex. It is feasible for them to apply directly for nonlinear control. The approach of fuzzy control is associated with the design of appropriate nonlinear functions in the fuzzy equation. Fuzzy control through fuzzy equations requires solution of the fuzzy equation. Several approaches are incorporated. [18] utilized the parametric form of fuzzy numbers and restored the original fuzzy equations using crisp linear systems. In [13], the extension principle is implemented and it suggests that the coefficients can be either real or complex fuzzy numbers. However, the validation of the solution is not assured. [2] inducted the Newton's technique. In [5], the solution of fuzzy equations are extracted using the fixed point methodology. The numerical solution associated with fuzzy equation can be fetched using the iterative technique [25], interpolation technique [42] and Runge-Kutta technique [33].

Neural networks can be implemented for resolving the fuzzy equation. A generalized fuzzy quadratic equation is resolved by utilizing neural networks which has been mentioned in [9]. Neural networks have been utilized in order to extract the solution of dual fuzzy equations which has been illustrated in [19]. A matrix pattern associated to the neural learning has been quoted in [27]. However, these techniques are not general as they cannot resolve general fuzzy equations with neural networks. Also, they cannot generate the fuzzy coefficients directly with neural networks [39].

In this paper, a survey on the numerical solutions of the fuzzy equations and dual fuzzy equations is given. The solutions which are generated by these equations are considered to be the controllers. Here, it has been presented that the roots of the mentioned equations can be extracted with different methods. The needs in mathematical modelling of efficient numerical algorithms as an alternative to classical methods of applied mathematics makes enormous progress for obtaining efficient numerical methods. In this paper, the advantages of the numerical methods in terms of accuracy is discussed. Studying of previous works by other researchers shows that no study has been done as a survey for the solutions of these equations, so that this survey will be a good beginning for those showing interest in the field of these kinds of equations.

II. Mathematical preliminaries

The following definitions are used in this paper.

Definition 1: If v is: 1) normal, there exists $\vartheta_0 \in \Re$ in such a manner $v(\vartheta_0) = 1$, 2) convex, $v(\gamma\vartheta + (1-\gamma)\vartheta) \geq \min\{v(\vartheta), v(\theta)\}$, $\forall\vartheta, \theta \in \Re, \forall\gamma \in [0,1]$, 3) upper semi-continuous on $\Re$, $v(\vartheta) \leq v(\vartheta_0) + \varepsilon$, $\forall\vartheta \in N(\vartheta_0)$, $\forall\vartheta_0 \in \Re$, $\forall\varepsilon > 0$, $N(\vartheta_0)$ is a neighborhood, 4) $v^+ = \{\vartheta \in \Re, v(\vartheta) > 0\}$ is compact, then v is a fuzzy variable, $v \in E : \Re \to [0, 1]$.

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

$$\mu_v = G(a, b, c) = \begin{cases} \frac{\vartheta - a}{b-a} & a \leq \vartheta \leq b \\ \frac{c-\vartheta}{c-b} & b \leq \vartheta \leq c \end{cases} \quad otherwise\ \mu_v = 0 \tag{1}$$

and trapezoidal function

$$\mu_v = G(a, b, c, d) = \begin{cases} \frac{\vartheta - a}{b-a} & a \leq \vartheta \leq b \\ \frac{d-\vartheta}{d-c} & c \leq \vartheta \leq d \\ 1 & b \leq \vartheta \leq c \end{cases} \quad otherwise\ \mu_v = 0 \tag{2}$$

Definition 2: The fuzzy number v in association to the α-level is illustrated as

$$[v]^\alpha = \{\vartheta \in \Re : v(\vartheta) \geq \alpha\} \tag{3}$$

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

Therefore $[v]^0 = v^+ = \{\vartheta \in \Re, v(\vartheta) > 0\}$. Since $\alpha \in [0, 1]$, $[v]^\alpha$ is a bounded mentioned as $\underline{v}^\alpha \leq [v]^\alpha \leq \overline{v}^\alpha$. The α-level of v in midst of $\underline{v}^\alpha$ and $\overline{v}^\alpha$ is explained as

$$[v]^\alpha = (\underline{v}^\alpha, \overline{v}^\alpha) \tag{4}$$

$\underline{v}^\alpha$ and $\overline{v}^\alpha$ signify the function of α. We state $\underline{v}^\alpha = d_A(\alpha)$, $\overline{v}^\alpha = d_B(\alpha)$, $\alpha \in [0, 1]$.

If $v_1, v_2 \in E$, the fuzzy operations are as follows

Sum,

$$[v_1 \oplus v_2]^\alpha = [v_1]^\alpha + [v_2]^\alpha = [\underline{v_1}^\alpha + \underline{v_2}^\alpha, \overline{v_1}^\alpha + \overline{v_2}^\alpha] \tag{5}$$

subtract,

$$[v_1 \ominus v_2]^\alpha = [v_1]^\alpha - [v_2]^\alpha = [\underline{v_1}^\alpha - \underline{v_2}^\alpha, \overline{v_1}^\alpha - \overline{v_2}^\alpha] \tag{6}$$

or multiply,

$$\underline{\omega}^\alpha \leq [v_1 \odot v_2]^\alpha \leq \overline{\omega}^\alpha or [v_1 \odot v_2]^\alpha = (\underline{\omega}^\alpha, \overline{\omega}^\alpha) \tag{7}$$

where $\underline{\omega}^\alpha = \underline{v_1}^\alpha \underline{v_2}^1 + \underline{v_1}^1 \underline{v_2}^\alpha - \underline{v_1}^1 \underline{v_2}^1$, $\overline{\omega}^\alpha = \overline{v_1}^\alpha \overline{v_2}^1 + \overline{v_1}^1 \overline{v_2}^\alpha - \overline{v_1}^1 \overline{v_2}^1$, and $\alpha \in [0, 1]$.

Definition 3: A general discrete-time nonlinear system can be described as

$$\bar{x}_{k+1} = \bar{p}[\bar{x}_k, w_k], \quad s_k = \bar{q}[\bar{x}_k] \tag{8}$$

Here we consider $w_k \in \Re^u$ as the input vector, $\bar{x}_k \in \Re^l$ is regarded as an internal state vector and $s_k \in \Re^m$ is the output vector. $\bar{p}$ and $\bar{q}$ are noted as generalized nonlinear smooth functions $\bar{p}, \bar{q} \in C^\infty$. Define $S_k = \left[s_{k+1}^T, s_k^T, \cdots\right]^T$ and $W_k = \left[w_{k+1}^T, w_k^T, \cdots\right]^T$. Suppose $\frac{\partial S}{\partial \bar{x}}$ is non-singular at the instance $\bar{x}_k = 0$, $W_k = 0$, this will create a path towards the following model

$$s_k = \Omega[s_{k-1}^T, s_{k-2}^T, \cdots w_k^T, w_{k-1}^T, \cdots] \tag{9}$$

where $\Omega(\cdot)$ is an nonlinear difference equation exhibiting the plant dynamics, w_k and s_k are computable scalar input and output respectively.

Definition 4: Taking into consideration the nonlinear systems as mentioned in (9), it can be simplified as the following linear-in-parameter model

$$s_k = \sum_{i=1}^{n} b_i p_i(x_k) \tag{10}$$

or

$$s_k + \sum_{i=1}^{m} c_i q_i(x_k) = \sum_{i=1}^{n} b_i p_i(x_k) \tag{11}$$

where b_i and c_i are considered to be the linear parameters, $p_i(x_k)$ and $q_i(x_k)$ are nonlinear functions. The variables related to these functions are quantifying input and output.

When the parameters in the linear-in-parameter model (10) or (11) are fuzzy number, (10) and (11) become fuzzy equations and dual fuzzy equations.

III. Numerical methods for solving fuzzy and dual fuzzy equations

A. Steepest descent method

In [3], a numerical solution associated with fuzzy nonlinear equation $G(x) = 0$ is suggested using steepest descent technique, where the fuzzy quantities are demonstrated in parametric form. The equation is represented by parametric form as mentioned below

$$\begin{cases} \underline{G}(\underline{x}^\alpha, \overline{x}^\alpha) = 0 \\ \overline{G}(\underline{x}^\alpha, \overline{x}^\alpha) = 0 \end{cases}$$

The function $K : \Re^2 \to \Re$ is stated by

$$K(\underline{x}, \overline{x}) = [\underline{G}(\underline{x}^\alpha, \overline{x}^\alpha), \overline{G}(\underline{x}^\alpha, \overline{x}^\alpha)]^2$$

The technique of steepest descent characterizes a local minimum for two-variable function K. The technique of steepest descent is stated as:

1. Finding out K at an initial approximation $X_0^\alpha = (\underline{x}_0^\alpha, \overline{x}_0^\alpha)$.

2. Determine a direction from $X_0^\alpha = (\underline{x}_0^\alpha, \overline{x}_0^\alpha)$ which causes a decrease in the value of K.

3. Shift a suitable amount in this direction and consider the new value $X_1^\alpha = (\underline{x}_1^\alpha, \overline{x}_1^\alpha)$.

4. Repeat sequence 1 via 3 with X_0^α replaced by X_1^α.

The steepest descent technique converges only linearly to the solution, but in general it converges even for weak initial approximations [17]. Even though steepest descent technique does not need a superior initial value, its drawback is due to its low convergency speed.

B. Adomian decomposition method

The Adomian decomposition method was primarily laid down by George Adomian in [4]. In [1] the standard Adomian decomposition is implemented on simple iteration technique in order to resolve the equation $g(x) = 0$, where $g(x)$ is a nonlinear function. The convergency related to the series solution is proved. Initially the nonlinear equation is transformed into canonical form, after that the Adomian technique computes the solution which is in the series form. As practically all the terms associated with the series are not possible to determine, hence the estimation of the solution from the truncated series has been accomplished. Also, the convergency related to the truncated series is usually very rapid.

Babolian et al. [6] altered the standard Adomian technique mentioned in [1] in order to solve nonlinear equation $g(x) = 0$ for acquiring a sequence of approximations to the solution, with approximately superlinear convergency. They have employed Cherruault's definition [15] and took into consideration the order of convergency related to the technique [7].

In [32] a potential numerical algorithm is demonstrated in order to solve fuzzy polynomial equations $\sum_{i=1}^{n} b_i x^i = d$ on the basis of Newton's technique, where x and d are considered to be fuzzy numbers, also all coefficients are taken to be fuzzy numbers. The modified Adomian decomposition methodology is implemented for the construction of the numerical algorithm. Primarily the fuzzy polynomials are illustrated in a parametric form and finally they have been resolved using Adomian decomposition technique.

In [41] the Shanks transformation is employed on the Adomian decomposition technique in order to resolve nonlinear equations so as to improvise the preciseness of the approximate solutions. The numerical results demonstrate that the the implementation of this technique in the same conditions generates more appropriate solutions in concerned to the nonlinear equations when compared with those extracted from the Adomian decomposition technique. The Shanks transform is an effective approach which can speed up the convergency rate of the series.

The advantage of the Adomian decomposition technique is that it can provide analytical approximations to solutions of nonlinear equations without supposing that the system has weak nonlinearities. The major drawback of the Adomian decomposition technique is the complex and difficult procedure in order to compute the Adomian polynomials. The ranking method is simple and inexpensive.

C. Ranking method

The ranking methodology was primarily laid down by Delgado et al [16]. In [37] the researcher used a ranking methodology of fuzzy numbers for obtaining the real roots of polynomial equation which has been demonstrated as follows

$$D_1 x + D_2 x^2 + ... + D_n x^n = D_0$$

where $x \in \Re$ as well as $D_0, D_1, ..., D_n$ are taken to be fuzzy numbers. In [37] the fuzzy polynomial equation is converted to system of crisp polynomial equations. This conversion performs with ranking methodology on the basis of three parameters value, ambiguity as well as fuzziness. The obtained system of crisp polynomial equations is resolved numerically.

In [28] the ranking methodology is suggested in order to extract the real roots associated to a dual fuzzy polynomial equation which has been displayed as follows

$$C_1 x + C_2 x^2 + ... + C_n x^n = D_1 x + D_2 x^2 + ... + D_n x^n + q$$

where $x \in \Re$ as well as $C_1, ..., C_n, D_1, ..., D_n, q$ are denoted as fuzzy numbers. The dual fuzzy polynomial equation is converted to the system associated to the crisp dual polynomial equations. This conversion is carried out by utilizing ranking methodology on the basis of three parameters namely value, ambiguity and fuzziness.

In [29] the real roots of the polynomial equation in the form of $C_1 x + C_2 x^2 + ... + C_n x^n = C_0$ is obtained by utilizing the ranking method of fuzzy numbers, where $x \in \Re$ as well as $C_0, C_1, ..., C_n$ are denoted as fuzzy numbers. Also in the quoted paper, the ranking methodology is utilized for real roots associated with dual polynomial equation which is mentioned as below

$$C_1 x + C_2 x^2 + ... + C_n x^n = A_1 x + A_2 x^2 + ... + A_n x^n + p$$

where $x \in \Re$, $C_1, ..., C_n, A_1, ..., A_n$ as well as p are considered to be fuzzy numbers.

In [30] the ranking technique is implemented in order to obtain the real roots of an interval type-2 dual fuzzy polynomial equation $B_1 x + B_2 x^2 + ... + B_n x^n = D_1 x + D_2 x^2 + ... + D_n x^n + k$, where $x \in \Re$, the coefficients $B_1, ..., B_n, D_1, ..., D_n$ as well as k are termed as interval type-2 fuzzy numbers. Type-2 dual fuzzy polynomial equation is converted into a system of crisp type-2 dual fuzzy polynomial equation. The mentioned conversion is performed by ranking method associated with fuzzy numbers on the basis of three parameters viz value, ambiguity and fuzziness.

It was revealed that solutions on the basis of three parameters such as value, ambiguity and fuzziness are not efficient to generate solutions. Henceforth in [31], a novel ranking methodology is suggested in order to eradicate the intrinsic weakness. The novel ranking methodology which is incorporated with four parameters has been implemented in the interval type-2 fuzzy polynomials. It covers the interval type-2 of fuzzy polynomial equation, dual fuzzy polynomial equations as well as system of fuzzy polynomials. The effectiveness of the novel ranking methodology is numerically considered in the triangular fuzzy numbers as well as the trapezoidal fuzzy numbers.

The main disadvantage of ranking method is that it can be applied only when membership functions are known. Approximation methods such as fuzzy neural networks are also effective tools to overcome the limitations of the other numerical methods. The major advantage of using fuzzy neural networks is training large amount of data sets, quick convergence and high accuracy.

D. Neural network method

In [10] neural network has been employed for solving fuzzy linear equation in the form mentioned below

$$CX = D \tag{12}$$

where C, D and X are considered to be triangular fuzzy numbers. For certain values of C and D, (12) generates no solution for X [13]. The training of neural network in order to solve (12) is mentioned by the researchers in [10], considering that zero does not belong to the support of C. The investigation is carried out considering neural network solutions termed to be Y and X^*. When there is no restrictions in concerned to the weights of the network, then the neural network output will be Y. The non existence of relationship between Y and X is validated by utilizing computer analysis. X^* is the solution of the neural network, taking into consideration that the certain sign restrictions are set on the weights. X^* is illustrated to be an approximation which is named as a new solution of fuzzy equations. It has been displayed by using $X \leq X^*$.

The evolutionary algorithm as well as neural network in combination have been utilized for solving the following fuzzy equation which has been mentioned in [11]

$$GX + F = D \tag{13}$$

where G, F, D and X are termed as triangular fuzzy numbers. The first solution type (X_c) related to (13) is stated to be the classical solution that utilizes α-cut and interval arithmetic for obtaining X_c.

Example 1. Assume $[G] = (1, 2, 3)$, $[F] = (-3, -2, -1)$ and $[D] = (3, 4, 5)$. Employing the intervals into the fuzzy equation generates

$$\begin{array}{c} (1+\alpha)\underline{X}_c^\alpha + (-3+\alpha) = (3+\alpha) \\ (3-\alpha)\overline{X}_c^\alpha + (-1-\alpha) = (5-\alpha) \end{array}$$

here $[X_c]^\alpha = (\underline{X_c}^\alpha, \overline{X_c}^\alpha)$. It can be extracted

$$\begin{array}{c} \underline{X}_c^\alpha = \frac{6}{1+\alpha} \\ \overline{X}_c^\alpha = \frac{6}{3-\alpha} \end{array}$$

However $[\underline{X}_c^\alpha, \overline{X}_c^\alpha]$ does not state a fuzzy number as because $\underline{X}_c^\alpha(\overline{X}_c^\alpha)$ is a decreasing (increasing) function of α. Occasionally X_c prevails and sometimes wont exist.

By the fuzzification of the crisp solution $(d - f)/g, g \neq 0$, the other solution is extracted. $(D - F)/G$ represents the fuzzified solution, taking into assumption that zero does not belong to the support of G. For the evaluation of the fuzzified solution, two approaches have been suggested. The primary approach generates the solution X_e by utilizing the extension principle as well as the secondary approach generates the solution X_I by means of α-cut and interval arithmetic. X_e can be achieved as mentioned below

$$X_e = \min\{\Pi(g, f, d) | (d - f)/g = x\}$$

where $\Pi(g, f, d) = \min\{G(g), F(f), D(d)\}$. For obtaining α-cut of X_e the process is described as follows

$$\begin{array}{c} \underline{X}_e^\alpha = \min\{\frac{d-f}{g} | g \in [G]^\alpha, f \in [F]^\alpha, d \in [D]^\alpha\} \\ \overline{X}_e^\alpha = \max\{\frac{d-f}{g} | g \in [G]^\alpha, f \in [F]^\alpha, d \in [D]^\alpha\} \end{array}$$

where $[X_e]^\alpha = (\underline{X_e}^\alpha, \overline{X_e}^\alpha)$. The solution X_I can be calculated as follows

$$[X_I]^\alpha = ([D]^\alpha - [F]^\alpha)/[G]^\alpha$$

The original fuzzy equation maybe or may not be solved by $X_e(X_I)$. Taking into account some fuzzy equations X_e is computationally complex to extract, so in [11] an evolutionary algorithm has been implemented for estimating its α-cuts. [11] can be generalized to the interactions of fuzzy problems, evolutionary algorithms as well as neural networks. There have been disadvantages incorporated in the method which has been mentioned in [11]. The method is exclusively meant for the symmetric fuzzy numbers, in addition it computes just the upper bound as well as the lower bound of the fuzzy numbers avoiding the center part.

In [24] the researchers obtained the approximate solution related to the following fuzzy polynomial having degree n

$$B_1 x + ... + B_n x^n = B_0 \tag{14}$$

where $B_0, B_1, ..., B_n$ and x belong to fuzzy set. They laid down two types of neural networks for approximating the solution related to (14), namely feedforward (static) as well as recurrent (dynamic) models. The corresponding algorithms related to both neural networks are based on the least mean square. The difference between two neural networks suggest that dynamic neural network is superiorly robust than static neural network. The technique which is illustrated in [24] is just able to find approximate solution of the special case of fuzzy equation, not generalized case.

The general fuzzy equations to be mentioned as dual fuzzy equations [42] has been illustrated in [19]. Normal fuzzy equations posses fuzzy numbers solely on one side of the equation. However, dual fuzzy equations posses fuzzy numbers on both sides of the equation. As because it is not possible to move the fuzzy numbers in between the sides of the equation [25], dual fuzzy equations are superiorly generalized and complex. In [19] the existence of the solutions related to the dual fuzzy equations is analyzed that is incorporated with the controllability problem associated to fuzzy control [14]. Two kinds of neural networks for approximation of the solutions related to dual fuzzy equations have been demonstrated namely static and dynamic models.

In [23] a dynamic neural network is proposed for resolving a dual fuzzy polynomial which is demonstrated as follows

$$a_1 x + ... + a_n x^n = b_1 x + ... + b_n x^n + d \tag{15}$$

where $a_1, ..., a_n$, $b_1, ..., b_n$ and d belong to fuzzy set. The neural network is trained by back-propagation-type learning algorithm which has five layer where connection weights are crisp numbers. The important advantage of this methodology is that, it can greatly reduce the size of calculations and generate a high accuracy of the numerical solution.

IV. Comparison of numerical methods

In this section, two examples of application have been laid down in order to compare the efficiency of the numerical methods to approximate the solution of dual fuzzy equations.

Example 1. The water tank system contains two inlet valves k_1, k_2, as well as two outlet valves k_3, k_4, see Figure 1. The areas of the valves are uncertain as the triangle function (1), $C_1 = G(0.021, 0.023, 0.024)$, $C_2 = G(0.008, 0.018, 0.038)$, $C_3 = G(0.012, 0.013, 0.015)$, $C_4 = G(0.038, 0.058, 0.068)$.

The velocities of the flow (controlled by the valves) are $g_1 = (\frac{\vartheta}{10})e^{\vartheta}$, $g_2 = \vartheta cos(\Pi\vartheta)$, $g_3 = cos(\frac{\Pi\vartheta}{8})$, $g_4 = \frac{\vartheta}{2}$. If the outlet flow is aimed to be $k = (4.088, 6.336, 36.399)$, what is the quantity of the control variable ϑ.

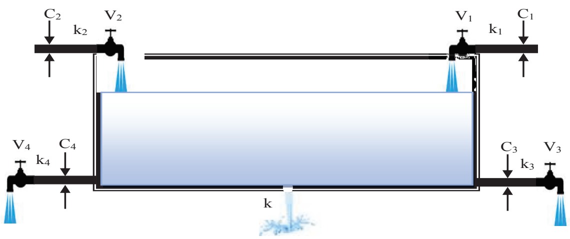


Fig. 1. Water tank system

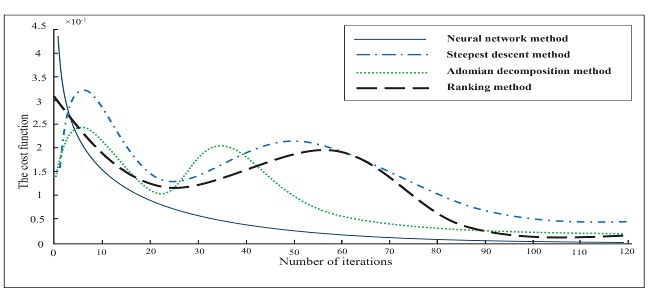
The mass balance of the tank is [38]:

$$\rho C_1 g_1 \oplus \rho C_2 g_2 = \rho C_3 g_3 \oplus \rho C_4 g_4 \oplus k$$

where ρ is considered to be the density of the water. The exact solution is taken to be $\vartheta_0 = 2$ [38]. To approximate the solution, we use four popular methods: Steepest descent method, Adomian decomposition method, Ranking method, and Neural network method. The errors of these methods are shown in Table 1. In this table k is the number of iterations. It can be seen that all four methods can approximate the solutions of the dual fuzzy equations. Neural network method is more suitable for solving these kind of equations. By increasing the number of iterations the estimated errors of the neural networks based algorithm are less than the other methods. Neural network method is more robust when compared with the other methods. Corresponding error plots are demonstrated in Figure 2.

Table1. Approximation errors

k	Steepest descent	Adomian decomposition	Ranking	Neural network
1	0.16812	0.14078	0.31006	0.43967
2	0.26096	0.22364	0.23801	0.32375
3	0.32651	0.18008	0.11952	0.21763
⋮	⋮	⋮	⋮	⋮
119	0.05234	0.03456	0.02986	0.00316
120	0.04935	0.03025	0.02563	0.00286



Approximation errors of six popular methods

Fig. 2. Approximation error of four popular methods

Example 2. The deformation of a solid cylindrical rod depends on the stiffness E, the forces on it g, the positions of the forces L, and the diameter of the rod d, see Figure 3. The positions are not exact, they satisfy the trapezoidal function (2), $L_1 = G(0.2, 0.3, 0.5, 0.6)$, $L_2 = G(0.4, 0.6, 0.7, 0.8)$, $L_3 = G(0.4, 0.6, 0.7, 0.8)$. The are of the rod is $A = \frac{\pi}{4}d^2$. The external forces are the function of ϑ, $g_1 = \vartheta^7$, $g_2 = \vartheta^6\sqrt{\vartheta}$, $g_3 = e^{2\vartheta}$. If the the desired deformation at the point M is aimed to be $M^* = G(0.000563, 0.000822, 0.001003, 0.001211)$, what is the quantity of the control force ϑ.

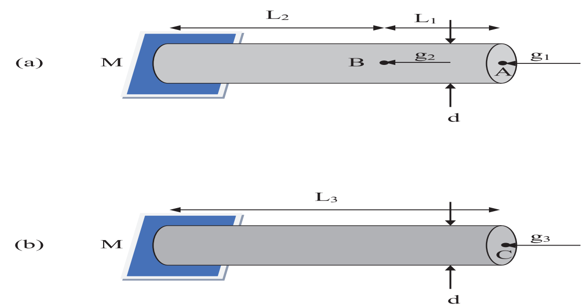


Fig. 3. Two solid cylindrical rods

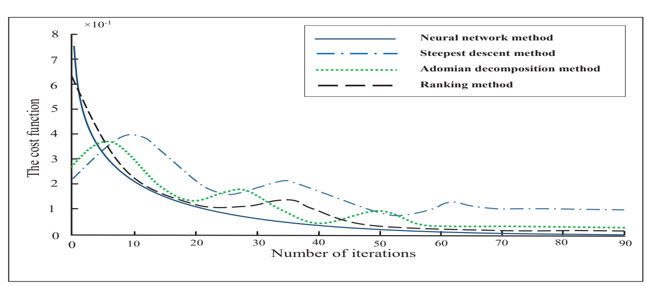
According to the tension relations we have [8]

$$\frac{L_1 g_1}{AE} \oplus \frac{L_2(g_1 + g_2)}{AE} = \frac{L_3 g_3}{AE} \oplus M^*$$

where $d = 0.02$, $E = 70 \times 10^9$. The exact solution is $\vartheta = 4$. To approximate the solution, we use four popular methods: Steepest descent method, Adomian decomposition method, Ranking method, and Neural network method. The errors of these methods are shown in Table 2. Neural network method is more robust than the other methods. Furthermore, the estimated error of the neural network is less when compared with other methods. Corresponding error plots are demonstrated in Figure 4.

Table2. Approximation errors

k	Steepest descent	Adomian decomposition	Ranking	Neural network
1	0.2013	0.2678	0.6004	0.7883
2	0.2996	0.3301	0.4987	0.5002
3	0.1844	0.2394	0.3791	0.3101
⋮	⋮	⋮	⋮	⋮
89	0.08014	0.05943	0.05001	0.00985
90	0.07201	0.04902	0.04112	0.00711



Approximation errors of six popular methods

Fig. 4. Approximation error of four popular methods

V. CONCLUSIONS

In this paper, some of numerical methodologies are demonstrated as a solution of fuzzy equations and dual fuzzy equations. This survey illustrates that the roots of the fuzzy equation can be extracted with different algorithms. However, in few cases there exist no roots in fuzzy equation. Solution of fuzzy polynomial by ranking methodology is proposed for solving fuzzy polynomial equation which convert to a crisp system of polynomial equations, therefore the system is easily solvable. For obtaining the roots of system in a case that there is no exact solution, iteration methodologies can be utilized for estimating the solution. By modified Adomian decomposition methodology, the roots can be extracted by laying down fuzzy polynomial in parametric form and solving it by Adomian decomposition methodology. Differently with fuzzy neural network, the root of fuzzy equation can be obtained by laying down a learning algorithm. This survey supplies an input for those showing interest in the field of fuzzy equations.

REFERENCES

[1] K. Abbaoui, Y. Cherruault, Convergence of Adomian's method applied to nonlinear equations, *Math. Comput. Model.* Vol.20, pp.69-73, 1994.

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

[3] S. Abbasbandy, A. Jafarian, Steepest descent method for solving fuzzy nonlinear equations, *Appl. Math. Comput.* Vol.174, pp.669-675, 2006.

[4] G. Adomian, A review of the Decomposition method in Applied Mathematics, *Journal of Mathematical Analysis and Applicatins,* Vol.135, pp.501-544, 1988.

[5] T. Allahviranloo, M. Otadi, M. Mosleh, Iterative method for fuzzy equations, *Soft Computing,* Vol.12, pp.935-939, 2007.

[6] E. Babolian, J. Biazar, Solution of nonlinear equations by modified Adomian decomposition method, *Appl. Math. Comput.* Vol.132, pp.167-172, 2002.

[7] E. Babolian, J. Biazar, On the order of convergence of Adomian method, *Appl. Math. Comput.* Vol.130, pp.383-387, 2002.

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

[9] J.J. Buckley, E. Eslami, Neural net solutions to fuzzy problems: The quadratic equation, *Fuzzy Sets Syst.* Vol.86, pp.289-298, 1997.

[10] J.J. Buckley, E. Eslami, Y. Hayashi, Solving fuzzy equations using neural nets, *Fuzzy Sets Syst.* Vol.86, pp.271-278, 1997.

[11] J.J. Buckley, T. Feuring, Y. Hayashi, Solving fuzzy equations using evolutionary algorithms and neural nets, *Soft comput.* Vol.6, pp.116-123, 2002.

[12] J.J. Buckley, Y. Hayashi, Can fuzzy neural nets approximate continuous fuzzy functions, *Fuzzy Sets Syst.* Vol.61, pp.43-51, 1994.

[13] J.J. Buckley, Y. Qu, Solving linear and quadratic fuzzy equations, *Fuzzy Sets Syst.* Vol.35, pp.43-59, 1990.

[14] S.M. Chen, Fuzzy system reliability analysis using fuzzy number arithmetic operations, *Fuzzy Sets Syst.* Vol.64, pp.31-38, 1994.

[15] Y. Cherruault, Convergence of Adomian's method, *Kybernetes,* Vol.9, pp.31-38, 1988.

[16] M. Delgado, M.A. Vila, W. Voxman, On a canonical representation of fuzzy Numbers, *Fuzzy Sets Syst.* Vol.93, pp.125-135, 1998.

[17] J.D. Faires, Numerical Analysis, *seventh ed., PWS-Kent, Boston,* 2001.

[18] M. Friedman, M. Ming, A. Kandel, Fuzzy linear systems,*Fuzzy Sets Syst.* Vol.96, pp.201-209, 1998.

[19] R. Jafari, W. Yu, Fuzzy Control for Uncertainty Nonlinear Systems with Dual Fuzzy Equations, *Journal of Intelligent and Fuzzy Systems.* Vol.29, pp.1229-1240, 2015.

[20] R. Jafari, W. Yu, Uncertainty nonlinear systems modeling with fuzzy equations, *In Proceedings of the 16th IEEE International Conference on Information Reuse and Integration (IRI '15),* pp.182-188, San Francisco, Calif, USA, August, 2015.

[21] R. Jafari, W. Yu, X. Li, Numerical solution of fuzzy equations with Z-numbers using neural networks, *Intelligent automation and Soft Computing,* pp.1-7, 2017.

[22] R. Jafari, W. Yu, X. Li, S. Razvarz, Numerical Solution of Fuzzy Differential Equations with Z-numbers Using Bernstein Neural Networks, *International Journal of Computational Intelligence Systems,* Vol.10, pp.1226-1237, 2017.

[23] A. Jafarian, R. Jafari, Approximate solutions of dual fuzzy polynomials by feed-back neural networks, *J. Soft Comput. Appl.* 2012, doi:10.5899/2012/jsca-00005.

[24] A. Jafarian, R. Jafari, A. Khalili, D. Baleanud, Solving fully fuzzy polynomials using feed-back neural networks. *International Journal of Computer Mathematics*, Vol.92, pp.742-755, 2015.

[25] M. Kajani, B. Asady, A. Vencheh, An iterative method for solving dual fuzzy nonlinear equations. *Appl. Math. Comput.* Vol.167, pp.316-323, 2005.

[26] E.H. Mamdani, Application of fuzzy algorithms for control of simple dynamic plant, *IEE Proceedings Control Theory and Applications*, Vol.121, No.12, 1585-1588, 1976.

[27] M. Mosleh, Evaluation of fully fuzzy matrix equations by fuzzy neural network, *Appl. Math. Model.* Vol.37, pp.6364-6376, 2013.

[28] A. Noorani, J. Kavikumar, M. Mustafa, and S. Nor, Solving dual fuzzy polynomial equation by ranking method, *Far East J. Math. Sci.* Vol.15, pp.151-163, 2011.

[29] R. Nurhakimah Ab, A. Lazim, Solving Fuzzy Polynomial Equation and The Dual Fuzzy Polynomial Equation Using The Ranking Method, *The 3th International conference on Mathematics Scinces* pp. 798-804, 2014.

[30] R. Nurhakimah Ab, A. Lazim, An Interval Type-2 Dual Fuzzy Polynomial Equations and Ranking Method of Fuzzy Numbers, *International Journal of Mathematical, Computational, Physical, Electrical and Computer Engineering,* Vol.8, 2014.

[31] R. Nurhakimah Ab, A. Lazim, Ab.G. Ahmad Termimi, A. Noorani, Solutions of Interval Type-2 Fuzzy Polynomials Using A New Ranking Method, *AIP Conference,* Vol.1682, 2015.

[32] M. Otadi, M. Mosleh, Solution of fuzzy polynomial equations by modified Adomian decomposition method, *Soft. Comput.* Vol.15, pp.187-192, 2011.

[33] S. Pederson, M. Sambandham, The Runge-Kutta method forhybrid fuzzy differential equation, *Nonlinear Anal. Hybrid Syst.* Vol.2, pp.626-634, 2008.

[34] S. Razvarz, R. Jafari, O-C. Granmo, A. Gegov, Solution of dual fuzzy equations using a new iterative method, *In Proceedings of the 10th Asian Conference on Intelligent Information and Database Systems, Lecture Notes in Artificial Intelligence (subseries of LNCS). Springer.* pp.245-255, 2018.

[35] S. Razvarz, R. Jafari, W. Yu, Numerical Solution of Fuzzy Differential Equations with Z-numbers using Fuzzy Sumudu Transforms, *Advances in Science, Technology and Engineering Systems Journal (ASTESJ),* Vol.3, pp.66-75, 2018.

[36] S. Razvarz, R. Jafari, W. Yu, A. Khalili, PSO and NN Modeling for Photocatalytic Removal of Pollution in Wastewater, *14th International Conference on Electrical Engineering, Computing Science and Automatic Control (CCE) Electrical Engineering,* pp.1-6, 2017.

[37] H. Rouhparvar, Solving fuzzy polynomial equation by ranking method, *First Joint Congress on Fuzzy and Intelligent Systems, Ferdowsi University of Mashhad, Iran,* 2007.

[38] V.L. Streeter, E.B. Wylie, E. Benjamin, *Fluid mechanics*, 4th Ed, Mc. Graw-Hill Book Company. 1999.

[39] A. Tahavvor, M. Yaghoubi, Analysis of natural convection from a column of cold horizontal cylinders using artificial neural network, *Appl. Math. Model.* Vol.36, pp.3176-3188, 2012.

[40] T. Takagi, M. Sugeno, Fuzzy identification of systems and its applications to modeling and control, *IEEE Trans. Syst. Man. Cybern.* Vol.15, 116-132, 1985.

[41] A.R. Vahidi, B. Jalalvand, Improving the Accuracy of the Adomian Decomposition Method for Solving Nonlinear Equations, *Appl. Math. Sci.* Vol.6, pp.487-497, 2012.

[42] M. Waziri, Z. Majid, A new approach for solving dual fuzzy nonlinear equations using Broyden's and Newton's methods, *Advances in Fuzzy Systems*, Vol.2012, Article 682087, 5 pages, 2012.