Special Issue Article

Advances in Mechanical Engineering

Advances in Mechanical Engineering
2017, Vol. 9(9) 1–9

DOI: 10.1177/1687814017717429
journals.sagepub.com/home/ade
SAGE

Approximate solutions by artificial neural network of hybrid fuzzy differential equations

Mahmoud Paripour[1], Massimiliano Ferrara[2] and Mehdi Salimi[3]

Abstract
In this article, we propose a new approach to solve the hybrid fuzzy differential equations based on the feed-forward neural networks. We first replace it by a system of ordinary differential equations. A trial solution of this system involves two parts. The first part satisfies the initial condition and contains no adjustable parameters; however, the second part involves a feed-forward neural network containing adjustable parameters (the weights). This method shows that using neural networks provides solutions with good generalization and the high accuracy.

Keywords
Hybrid systems, fuzzy differential equations, neural network, feed-forward artificial neural networks, approximate solution

Date received: 7 February 2017; accepted: 1 June 2017

Academic Editor: Dumitru Baleanu

Introduction

Fuzzy differential equations (FDEs) are significant for studying and solving large proportions of problems in many topics of applied mathematics, particularly in relation to physics, geography, medicine, biology, control chaotic systems, bioinformatics and computational biology, synchronize hyperchaotic systems, economics and finance, and so on.[1–3] In lots of applications, some of the parameters are represented by fuzzy numbers rather than the crisp numbers, and hence, it is essential to develop mathematical models and numerical procedures which would have appropriately treated to general FDEs. The knowledge about differential equations is often incomplete or vague. The FDEs were formulated by Kaleva[4] and Seikkala[5] initially. Hybrid systems received too much attention in the recent literatures as well. Hybrid systems evolve in continuous time, like differential systems, but undergoing a fundamental change in their governing equations at a sequence of discrete times. When a continuous time dynamics of a hybrid system is given by a FDE, the system is called a hybrid fuzzy differential system. For analytical results on hybrid fuzzy differential equations (HFDEs), see Lakshmikantham and Liu[6] and Lakshmikantham and Mohapatra.[7] Hybrid systems are devoted for modeling, designing, and validating interactive systems of computer programs and continuous systems as well. These control systems which are capable of controlling complex systems have discrete event dynamics as well as continuous time dynamics that can be modeled by hybrid systems. The differential systems containing

[1]Department of Computer Engineering and Information Technology, Hamedan University of Technology, Hamedan, Iran
[2]Department of Law and Economics, Mediterranea University of Reggio Calabria, Reggio Calabria, Italy
[3]Department of Mathematics, Tuyserkan Branch, Islamic Azad University, Tuyserkan, Iran

Corresponding author:
Mehdi Salimi, Department of Mathematics, Tuyserkan Branch, Islamic Azad University, Tuyserkan, Iran.
Email: msalimi1@yahoo.com

fuzzy-valued functions and interaction with a discrete time controller are named as hybrid fuzzy differential systems (HFDEs). Several numerical techniques have been applied for solving HFDEs. For instance, Pederson and Sambandham[8,9] investigated the numerical solution of these equations using the Euler and Runge–Kutta methods. Prakash and Kalaiselvi[10] studied the predictor-corrector method for HFDEs. In addition, Fard and Bidgoli[11] solved HFDEs by the Nystrom method. Similarly, Kim and Sakthivel[12] studied the predictor–corrector method for HFDEs, and Allahviranloo and Salahshour[13] investigated the numerical solution of HFDEs, using the Euler method under characterization theorem and Bede's differentiability. Ahmadian et al.[14–16] applied Runge–Kutta method with lower function evaluations and the reduced the number of function evaluations for solving HFDEs. Paripour et al.[17] applied Adomian decomposition method for solving HFDEs.

In this study, we develop numerical methods for HFDEs by an application of artificial neural network. Lagaris et al.[18] designed artificial neural networks for solving ordinary differential equations (ODEs) and partial differential equations (PDEs). They used multilayer perceptron to estimate the solution of differential equations. In 2010, Effati and Pakdaman[19] used artificial neural networks for solving FDEs. Similarly, Pakdaman et al.[20] solved differential equations of fractional order using an optimization technique based on training artificial neural network. Their neural network model was trained over an interval (over which the differential equation must be solved), so the inputs of the neural network model were the training points. Bede[21] proved a characterization theorem which states that under certain conditions, an FDE under the Hukuhara differentiability is equivalent to a system of ODEs. Moreover, Bede also noticed that this characterization theorem can aid to solve FDEs numerically through converting the FDEs to a system of ODEs, which later could be solved by numerical methods. In this article, using the characterization theorem, we generalize a fourth-order Runge–Kutta method that originally presented to solve the HFDEs. That is, we substitute the original initial value problem with two parametric hybrid ordinary differential systems. Then, the extension of Bede's characterization theorem for HFDEs, which was investigated by Pederson and Sambandham,[22] is employed to generalize the derivatives. Finally, these results are applied to solve the HFDEs numerically by the fourth-order reduced Runge–Kutta (RRK) method.

Preliminaries

Definition 1. A fuzzy number is a function $\tilde{u} : \mathbb{R} \to [0, 1]$ with the following properties:[23]

1. $\tilde{u}$ is normal, that is, $\exists t_0 \in \mathbb{R}$ such that $\tilde{u}(t_0) = 1$;
2. $\tilde{u}$ is a fuzzy convex set, that is

$$\tilde{u}(\alpha t_1 + (1-\alpha)t_2) \geq min\{\tilde{u}(t_1), \tilde{u}(t_2)\}$$
$$\forall t_1, t_2 \in \mathbb{R}, \quad \alpha \in [0, 1]$$

3. $\tilde{u}$ is the upper semi-continuous on $\mathbb{R}$;
4. $\overline{\{t \in R : \tilde{u}(t) > 0\}}$ is a compact set, where $\overline{S}$ denotes the closure of S.

The set of all fuzzy numbers is denoted by E. Kaleva[4] gave an alternative definition which yields the same E.

Definition 2. An ordered pair of functions $(\underline{u_1}(r), \overline{u_1}(r))$, $0 \leq r \leq 1$ presents an arbitrary fuzzy number, in parametric form, which satisfies the following requirements:[23]

1. $\underline{u_1}(r)$ is a bounded left continuous non-decreasing on [0, 1];
2. $\overline{u_1}(r)$ is a bounded left continuous non-increasing function on [0, 1];
3. $\underline{u_1}(r) \leq \overline{u_1}(r)$, $0 \leq r \leq 1$.

The following equations define the addition and the scalar multiplication of fuzzy numbers in E:

1. $\tilde{u}_1 \oplus \tilde{u}_2 = \left(\underline{u_1}(r) + \underline{u_2}(r), \overline{u_1}(r) + \overline{u_2}(r)\right)$;
2. $(\lambda \odot \tilde{u}_1) = \begin{cases} (\lambda \underline{u_1}(r), \lambda \overline{u_1}(r)), & \lambda \geq 0 \\ (\lambda \overline{u_1}(r), \lambda \underline{u_1}(r)), & \lambda < 0 \end{cases}$.

Definition 3. The Hausdorff distance between $\tilde{u}_1$ and $\tilde{u}_2$ for arbitrary fuzzy numbers, $\tilde{u}_1 = (\underline{u_1}(r), \overline{u_1}(r))$ and $\tilde{u}_2 = (\underline{u_2}(r), \overline{u_2}(r))$, is[24]

$$D(\tilde{u}_1, \tilde{u}_2) = \sup_{r \in [0,1]} max\left\{|\underline{u_1}(r) - \underline{u_2}(r)|, |\overline{u_1}(r) - \overline{u_2}(r)|\right\}$$

The following properties hold:

1. (E, D) is a complete metric space;
2. $D(\tilde{u}_1 \oplus \tilde{u}_2, \tilde{u}_3 \oplus \tilde{u}_2) = D(\tilde{u}_1, \tilde{u}_3)$, $\forall \tilde{u}_1, \tilde{u}_2, \tilde{u}_3 \in E$;
3. $D(\lambda \odot \tilde{u}_1, \lambda \odot \tilde{u}_2) = |\lambda| D(\tilde{u}_1, \tilde{u}_2)$, $\forall \tilde{u}_1, \tilde{u}_2 \in E$, $\forall \lambda \in \mathbb{R}$;
4. $D(\tilde{u}_1 \oplus \tilde{u}_2, \tilde{u}_3 \oplus \tilde{u}_4) \leq D(\tilde{u}_1, \tilde{u}_3) + D(\tilde{u}_2, \tilde{u}_4)$, $\forall \tilde{u}_1, \tilde{u}_2, \tilde{u}_3, \tilde{u}_4 \in E$.

Definition 4. Let $f : \mathbb{R} \to E$ is a fuzzy-valued function. If for arbitrary fixed $t_0 \in \mathbb{R}$ and $\varepsilon > 0, \delta < 0$ such that[25]

$$|t - t_0| < \delta \Rightarrow D(f(t), f(t_0)) < \varepsilon$$

f is said to be continuous.

Definition 5. Suppose $x, y \in E$. If there exists $z \in E$ such that $x = y \oplus z$, then z is called the Hukuhara difference (H-difference) of x and y, and it is denoted by $x \ominus y$.[26] In this article, the sign $\ominus$ stands for H-difference, and also note that $x \ominus y \neq x + (-1)y$.

Definition 6. Let $f : T \to E$ is a fuzzy function and $T \subseteq \mathbb{R}$. We say f is differentiable at $t_0 \in T$, if there exists an element $f'(t_0) \in E$ such that limits[26]

$$\lim_{\Delta t \to 0^+} \frac{f(t_0 + \Delta t) \ominus f(t_0)}{\Delta t}$$

and

$$\lim_{\Delta t \to 0^+} \frac{f(t_0) \ominus f(t_0 - \Delta t)}{\Delta t}$$

exist and are equal to $f'(t_0)$. Here, the limits are taken in the metric space (E, D).

The above definition is a generalization of the H-differentiability of a set-valued function. From Bhaskar et al.,[27] it follows that a H-differentiable function has increasing length of support, so this definition of a derivative is very restrictive. In this regard, Bede and Gal[28] introduced a more generalized definition of H-differentiability which is our interest in this article.

Definition 7. Let $f : (a, b) \to E$ and $x_0 \in (a, b)$. We say that f is strongly generalized H-differentiable at t_0 if there exists an element $f'(t_0) \in E$, such that[28]

1. For all $\Delta t > 0$ sufficiently small, $\exists f(t_0 + \Delta t) \ominus f(t_0)$, $\exists f(t_0) \ominus f(t_0 - \Delta t)$ and limits (in the metric D)

$$\lim_{\Delta t \to 0} \frac{f(t_0 + \Delta t) \ominus f(t_0)}{\Delta t} = \lim_{\Delta t \to 0} \frac{f(t_0) \ominus f(t_0 - \Delta t)}{\Delta t} = f'(t_0)$$

or

2. For all $\Delta t > 0$ sufficiently small, $\exists f(t_0) \ominus f(t_0 + \Delta t)$, $\exists f(t_0 - \Delta t) \ominus f(t_0)$ and limits (in the metric D)

$$\lim_{\Delta t \to 0} \frac{f(t_0) \ominus f(t_0 + \Delta t)}{-\Delta t} = \lim_{\Delta t \to 0} \frac{f(t_0 - \Delta t) \ominus f(t_0)}{-\Delta t} = f'(t_0)$$

or

3. For all $\Delta t > 0$ sufficiently small, $\exists f(t_0 + \Delta t) \ominus f(t_0)$, $\exists f(t_0 - \Delta t) \ominus f(t_0)$ and limits (in the metric D)

$$\lim_{\Delta t \to 0} \frac{f(t_0 + \Delta t) \ominus f(t_0)}{\Delta t} = \lim_{\Delta t \to 0} \frac{f(t_0 - \Delta t) \ominus f(t_0)}{-\Delta t} = f'(t_0)$$

or

4. For all $\Delta t > 0$ sufficiently small, $\exists f(t_0) \ominus f(t_0 + \Delta t)$, $\exists f(t_0) \ominus f(t_0 - \Delta t)$ and limits (in the metric D)

$$\lim_{\Delta t \to 0} \frac{f(t_0) \ominus f(t_0 + \Delta t)}{-\Delta t} = \lim_{\Delta t \to 0} \frac{\exists f(t_0) \ominus f(t_0 - \Delta t)}{\Delta t} = f'(t_0)$$

Theorem 1 (Bede's characterization theorem[21]). Let us consider the fuzzy initial value problem (FIVP)

$$\begin{cases} y'(x) = f(x, y(x)) \\ y(x_0) = y_0 \end{cases} \tag{1}$$

where $f : [x_0, x_0 + a] \times E \to E$ is such that

1. $[f(x, y)]^r = [\underline{f}^r(x, \underline{y}, \overline{y}), \overline{f}^r(x, \underline{y}, \overline{y})]$;
2. $\underline{f}^r$ and $\overline{f}^r$ are equicontinuous (i.e. for any $\varepsilon > 0$ there is a $\delta > 0$ such that $|\underline{f}^r(x, y, z) - \underline{f}^r(x_1, y_1, z_1)| < \varepsilon$ and $|\overline{f}^r(x, y, z) - \overline{f}^r(x_1, y_1, z_1)| < \varepsilon$ for all $r \in [0, 1]$, whenever $(x, y, z), (x_1, y_1, z_1) \in [x_0, x_0 + a] \times \mathbb{R}^2$ and $\| (x, y, z) - (x_1, y_1, z_1) \| < \delta$) and uniformly bounded on any bounded set;
3. There exists on an $L > 0$ such that

$$|\underline{f}^r(x, y, z) - \underline{f}^r(x, y_1, z_1)| \leq L max\{|y_1 - y|, |z_1 - z|\} \text{ for all } r \in [0, 1]$$
$$|\overline{f}^r(x, y, z) - \overline{f}^r(x, y_1, z_1)| \leq L max\{|y_1 - y|, |z_1 - z|\} \text{ for all } r \in [0, 1]$$

Then, the FIVP (equation (1)) and system of ODEs

$$\begin{cases} (\underline{y}^r(x))' = \underline{f}^r(x, \underline{y}^r, \overline{y}^r) \\ (\overline{y}^r(x))' = \overline{f}^r(x, \underline{y}^r, \overline{y}^r) \\ \underline{y}^r(x_0) = \underline{y}^r_0 \\ \overline{y}^r(x_0) = \overline{y}^r_0 \end{cases} \tag{2}$$

are equivalent.

The hybrid fuzzy differential system

Consider the following hybrid fuzzy differential system

$$\begin{cases} y'(x) = f(x, y(x), \lambda_k(y_k)), \quad x \in [x_k, x_{k+1}] \\ y(x_k) = y_k \end{cases} \tag{3}$$

where $0 \le x_0 < x_1 < \cdots < x_k < \cdots$, $x_k \to \infty$, $f \in C[\mathbb{R}^+ \times E \times E, E]$, $\lambda_k \in C[E, E]$.

To be specific, the system would look like

$$y'(x) = \begin{cases} y'_0(x) = f(x, y_0(x), \lambda_0(y_0)), & y_0(x_0) = y_0, \quad x_0 \le x \le x_1 \\ y'_1(x) = f(x, y_1(x), \lambda_1(y_1)), & y_1(x_1) = y_1, \quad x_1 \le x \le x_2 \\ \vdots & \vdots \\ y'_k(x) = f(x, y_k(x), \lambda_k(y_k)), & y_k(x_k) = y_k, \quad x_k \le x \le x_{k+1} \\ \vdots & \vdots \end{cases} \tag{4}$$

Assuming that the existence and uniqueness of solutions of equation (3) hold for each $[x_k, x_{k+1}]$, by the solution of equation (3), we obtain the following function

$$y(x) = y(x, x_0, y_0) = \begin{cases} y_0(x), & x_0 \le x \le x_1 \\ y_1(x), & x_1 \le x \le x_2 \\ \vdots & \vdots \\ y_k(x), & x_k \le x \le x_{k+1} \\ \vdots & \vdots \end{cases} \tag{5}$$

We note that the solutions of equation (3) are piecewise differentiable in each interval for $x \in [x_k, x_{k+1}]$, for a fixed $y_k \in E$ and $k = 0, 1, 2, \ldots$. We replace equation (3) by the following equivalent system

$$\begin{cases} \underline{y}'(x) = \underline{f}(x, y(x), \lambda_k(y_k)), & \underline{y}(x_k) = \underline{y_k} \\ \overline{y}'(x) = \overline{f}(x, y(x), \lambda_k(y_k)), & \overline{y}(x_k) = \overline{y_k} \end{cases} \tag{6}$$

The parametric form of the above system is given by

$$\begin{cases} \underline{y}'(x,r) = F\Big(x, \underline{y}(x,r), \overline{y}(x,r), \underline{\lambda_k}(y_k)(r), \overline{\lambda_k}(y_k)(r)\Big), & \underline{y}(x_k, r) = \underline{y_k}(r) \\ \overline{y}'(x,r) = G\Big(x, \underline{y}(x,r), \overline{y}(x,r), \underline{\lambda_k}(y_k)(r), \overline{\lambda_k}(y_k)(r)\Big), & \overline{y}(x_k, r) = \overline{y_k}(r) \end{cases} \tag{7}$$

where $x \in [x_k, x_{k+1}]$ and $r \in [0, 1]$. Using Bede's characterization theorem proposed by Bede,[21] Pederson and Sambandham[22] generalized the following characterization theorem for HFDEs.

$$\min_{\overrightarrow{v}} \sum_{i=1}^{m} \left\{ \begin{array}{l} \Big(\underline{y}'_T(x_i, r, \underline{v}) - F[x_i, \underline{y}_T(x_i, r, \underline{v}), \overline{y}_T(x_i, r, \overline{v}), \underline{\lambda_k}(y_k)(r), \overline{\lambda_k}(y_k)(r)]\Big)^2 \\ + \Big(\overline{y}'_T(x_i, r, \overline{v}) - G[x_i, \underline{y}_T(x_i, r, \underline{v}), \overline{y}_T(x_i, r, \overline{v}), \underline{\lambda_k}(y_k)(r), \overline{\lambda_k}(y_k)(r)]\Big)^2 \end{array} \right\} \tag{8}$$

Theorem 2. Consider the HFDE (3) expanded as equation (4) where for $k = 0, 1, 2, \ldots$, and each $f_k : [x_k, x_{k+1}] \times E \to E$, we have:[22]

1. $[f_k(x, y)]^r = \Big[\underline{f}_k^r(x, \underline{y}, \overline{y}), \overline{f}_k^r(x, \underline{y}, \overline{y})\Big]$;
2. $\underline{f}_k^r$ and $\overline{f}_k^r$ are equicontinuous and uniformly bounded on any bounded set;
3. There exists a $L_k > 0$ such that

$|\underline{f}_k^r(x, y, z) - \underline{f}_k^r(x, y_1, z_1)| \le L_k max\{|y_1 - y|,$
$|z_1 - z|\}$ for all $r \in [0, 1]$
$|\overline{f}_k^r(x, y, z) - \overline{f}_k^r(x, y_1, z_1)| \le L_k max\{|y_1 - y|,$
$|z_1 - z|\}$ for all $r \in [0, 1]$

Then, the FIVP (equation (3)) and system of ODEs

$$\begin{cases} (\underline{y}_k^r(x))' = {}^r_k(x, \underline{y}_k^r, \overline{y}_k^r) \\ (\overline{y}_k^r(x))' = \overline{f}_k^r(x, \underline{y}_k^r, \overline{y}_k^r) \\ \underline{y}_k^r(x_k) = \underline{y}_{k-1}^r(x_k), \quad \text{if } k > 0, \ \underline{y}_0^r(x_0) = \underline{y}_0^r \\ \overline{y}_k^r(x_k) = \overline{y}_{k-1}^r(x_k), \quad \text{if } k > 0, \ \overline{y}_0^r(x_0) = \overline{y}_0^r \end{cases}$$

are equivalent.

Neural networks

Using neural networks provides solutions with very good generalizability (such as differentiability). However, an important feature of multilayer perceptrons is their utility to approximate functions, which leads to a wide applicability in most problems. In this article, the function approximation capability of feed-forward neural networks is used by expressing the trial solutions for system (7) as the sum of two terms (see equation (9)). The first term satisfies the initial conditions and does not contain adjustable parameters. The second term involves a feed-forward neural network to be trained, so satisfies the differential equations. Since it is known as a multilayer perceptron with one hidden layer which can approximate any function to arbitrary accuracy, the multilayer perceptron is used as the type of the network architecture.

If $\underline{y}_T(x, r, \underline{p})$ is a trial solution for the first equation in system (3) and $\overline{y}_T(x, r, \overline{p})$ is a trial solution for the second equation in system (3) where $\underline{p}$ and $\overline{p}$ are adjustable parameters (indeed $\underline{y}_T(x, r, \underline{p})$ and $\overline{y}_T(x, r, \overline{p})$ are approximations of $\underline{y}_T(x, r)$ and $\overline{y}_T(x, r)$, respectively), then a discretized issue of system (3) might be converted to the optimization problem

where $\overrightarrow{v} = (\underline{v}, \overline{v})$ includes all adjustable parameters with the initial conditions

$$\underline{y}_T(x_0, r, \underline{v}) = \underline{y}_0(r), \quad \overline{y}_T(x_0, r, \overline{v}) = \overline{y}_0(r)$$

Each trial solution $\underline{y}_T$ and $\overline{y}_T$ employs one feed-forward neural network for which the corresponding networks are denoted by $\underline{N}$ and $\overline{N}$, with adjustable parameters $\underline{v}$ and $\overline{v}$, respectively. Thus, $\underline{y}_T$ and $\overline{y}_T$ can be selected as follows

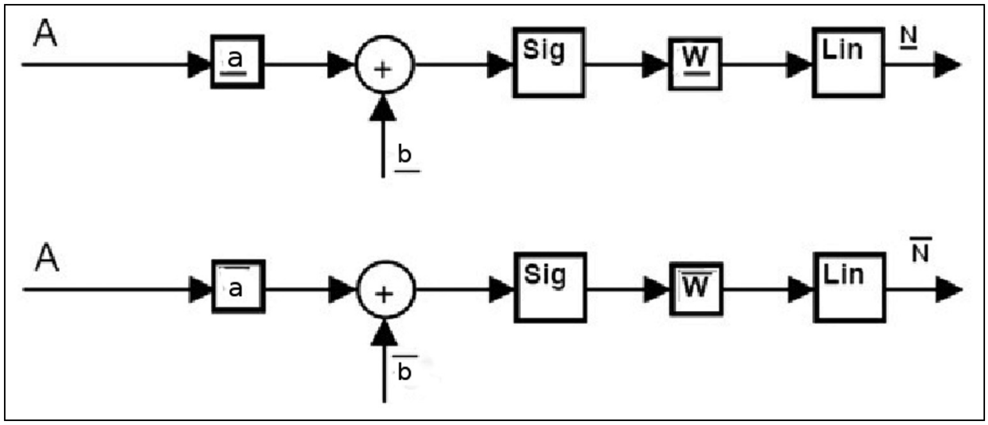

Figure 1. Architecture of the perceptron.

$$\begin{cases} \underline{y}'_T(x,r,\underline{v}) = \underline{y}(x_0,r) + (x-x_0)\underline{N}(x,r,\underline{v}) \\ \overline{y}_T(x,r,\overline{v}) = \overline{y}(x_0,r) + (x-x_0)\overline{N}(x,r,\overline{v}) \end{cases} \tag{9}$$

where $\underline{N}$ and $\overline{N}$ are single-output feed-forward neural networks with adjustable parameters $\underline{v}$ and $\overline{v}$, respectively. Here, x and r are the network inputs. It is easy to see that in equation (9), $\underline{y}_T$ and $\overline{y}_T$ satisfy the initial conditions. From equation (9), it is easy to show that

$$\begin{cases} \underline{y}'_T(x,r,\underline{v}) = \underline{N}(x,r,\underline{v}) + (x-x_0)\dfrac{\partial \underline{N}}{\partial x} \\ \overline{y}'_T(x,r,\overline{v}) = \overline{N}(x,r,\overline{v}) + (x-x_0)\dfrac{\partial \overline{N}}{\partial x} \end{cases} \tag{10}$$

$$\min_{\vec{v}} \sum_{i=1}^{n} \left\{ \begin{array}{l} \left(\underline{N}(x_i,r,\underline{v}) + (x_i-x_0)\dfrac{\partial \underline{N}}{\partial x} - F[x_i, \underline{y}_T(x_i,r,\underline{v}), \overline{y}_T(x_i,r,\overline{v}), \underline{\lambda_k}(y_k)(r), \overline{\lambda_k}(y_k)(r)]\right)^2 \\ + \left(\overline{N}_T(x_i,r,\overline{v}) + (x_i-x_0)\dfrac{\partial \overline{N}}{\partial x} - G[x_i, \underline{y}_T(x_i,r,\underline{v}), \overline{y}_T(x_i,r,\overline{v}), \underline{\lambda_k}(y_k)(r), \overline{\lambda_k}(y_k)(r)]\right)^2 \end{array} \right\} \tag{13}$$

Now suppose a multilayer perceptron has a hidden layer with H sigmoid units and a linear output unit (Figure 1). Therefore, we have

$$\begin{cases} \underline{N} = \sum_{i=1}^{m} \underline{w}_i \sigma(\underline{t}_i), \quad \underline{t}_i = \underline{a}_{i1}x + \underline{a}_{i2}r + \underline{b}_i \\ \overline{N} = \sum_{i=1}^{m} \overline{w}_i \sigma(\overline{t}_i), \quad \overline{t}_i = \overline{a}_{i1}x + \overline{a}_{i2}r + \overline{b}_i \end{cases} \tag{11}$$

where $\sigma(t)$ is the sigmoid transfer function, $\underline{a}$ and $\overline{a}$ ($m \times 2$ matrices) are the weights of input layers, and $\underline{b}$ and $\overline{b}$ ($m \times 1$ matrices) are the bias vectors of input units. $\underline{w}$ and $\overline{w}$ ($m \times 1$ matrices) are the weight vectors of output units, and $\sigma(t) = 1/(1+e^{-t})$ is the sigmoid transfer function. The following is obtained

$$\begin{cases} \dfrac{\partial \underline{N}}{\partial x} = \sum_{i=1}^{m} \underline{w}_i \underline{a}_{i1} \sigma'(\underline{t}_i) \\ \dfrac{\partial \overline{N}}{\partial x} = \sum_{i=1}^{m} \overline{w}_i \overline{a}_{i1} \sigma'(\overline{t}_i) \end{cases} \tag{12}$$

where $\sigma'(\overline{t}_i)$ is the first derivative of the sigmoid function. Now, if we substitute equation (10) in (8), the constrained optimization problem (8) might be changed with the unconstrained optimization problem as follow

Numerical example

We use the Broyden–Fletcher–Goldfarb–Shanno (BFGS) quasi-Newton method to minimize the objective function in the MATLAB optimization toolbox. We take $\underline{e}(x,r) = \underline{y}_T(x,r) - \underline{y}_a(x,r)$ and $\overline{e}(x,r) = \overline{y}_T(x,r) - \overline{y}_a(x,r)$, where $y_T = (\underline{y}_T, \overline{y}_T)$ is the approximated solution and $y_a = (\underline{y}_a, \overline{y}_a)$ is the known exact solution. We use multilayer perceptron consisting of 1 hidden layer with 10 hidden units and 1 linear output unit.

Example. Assume the hybrid FIVP

$$\begin{cases} y'(x) = y(x) + m(x)\lambda_k(y_{x_k}), \quad x \in [x_k, x_{k+1}], \quad x_k = k, \quad k = 0, 1, \ldots \\ y(0) = [r - 0.25, 2 - 1.25r] \end{cases} \tag{14}$$

where

Table 1. Comparison of the approximated solutions and the exact solutions on [0, 1].

r	Exact solution $(\underline{y}_a(1,r), \overline{y}_a(1,r))$	Approximated solution $(\underline{y}_T(1,r), \overline{y}_T(1,r))$	Absolute error
0.0	(–0.67957045, 5.0967784)	(–0.67957036, 5.0967784)	(6.215545e−07, 1.875812e−06)
0.1	(–0.40774227, 4.7909717)	(–0.40774282, 4.7909717)	(2.279674e−06, 8.964055e−07)
0.2	(–0.13591409, 4.4851650)	(–0.13591370, 4.4851650)	(1.967348e−07, 2.704311e−07)
0.3	(0.13591409, 4.1793583)	(0.13591389, 4.1793583)	(1.503723e−06, 2.986754e−07)
0.4	(0.40774227, 3.8735516)	(0.40774239, 3.8735516)	(1.733932e−06, 1.385048e−06)
0.5	(0.67957045, 3.5677448)	(0.67957043, 3.5677448)	(7.522059e−07, 1.104163e−06)
0.6	(0.95139863, 3.2619381)	(0.95139853, 3.2619382)	(2.363650e−07, 5.354381e−07)
0.7	(1.22322682, 2.9561314)	(1.12232269, 2.9561314)	(1.381800e−07, 5.435571e−07)
0.8	(1.49505500, 2.6503247)	(1.4950549, 2.6503247)	(2.570600e−07, 2.209716e−06)
0.9	(1.76688318, 2.3445180)	(1.7668831, 2.3445180)	(6.189354e−07, 7.521389e−08)
1.0	(2.03871137, 2.03871137)	(2.0387113, 2.0387113)	(4.528218e−06, 4.528218e−06)

$$m(x) = \begin{cases} 2(t(mod1)), & x(mod1) \leq 0.5 \\ 2(1 - t(mod1)), & x(mod1) > 0.5 \end{cases} \quad (15)$$

and

$$\lambda_k(\mu) = \begin{cases} \widehat{0}, & k = 0 \\ \mu, & k \in \{1, 2, \ldots\} \end{cases} \quad (16)$$

for which $\widehat{0} \in E^n$ is defined as $\widehat{0}(y) = 1$ if $y = 0$ and $\widehat{0}(y) = 0$ if $y \neq 0$.

The hybrid fuzzy initial problem (equation (14)) is equivalent to the following system of FIVPs

$$\begin{cases} y'_0(x) = y_0(x), & x \in [0, 1] \\ y(0) = [r - 0.25, -1.125r + 1.875] \\ y'_i(x) = x_i(t) + m(x) y_i(x_i), & x \in [x_i, x_{i+1}] \end{cases}$$

In equation (14), $y(x) + m(x)\lambda_k(y_{x_k})$ is a continuous function of x, y, and $\lambda_k(y_{x_k})$. Therefore, referring example by Kaleva,[4] for each $k = 0, 1, 2, \ldots$ the FIVP

$$\begin{cases} y'(x) = f(x, y(x), \lambda_k(y_k)), & x \in [x_k, x_{k+1}] \\ y(x_k) = y_k \end{cases}$$

has a unique solution on $[x_k, x_{k+1}]$.

Exact solution for $x = 1$ is

$$y(1, r) = [(r - 0.25)e, (-1.125r + 1.875)e], \quad r \in [0, 1]$$

The trial solutions in the neural form are as follows for $x \in [0, 1]$

$$\begin{cases} \underline{y}_T(x) = (r - 0.25) + x\underline{N}(x, r, \underline{v}) \\ \overline{y}_T(x) = (-1.125r + 1.875) + x\overline{N}(x, r, \overline{v}) \end{cases}$$

The exact solution for $x = 1.5$ is

$$y(1.5, r) = [(5.291r - 1.3227), (-5.9523r + 9.9202)], \quad r \in [0, 1]$$

The trial solutions for $x \in [1, 1.5]$ are

$$\begin{cases} \underline{y}_T(x) = (5.291r - 1.3227) + (x - 1)\underline{N}(x, r, \underline{v}) \\ \overline{y}_T(x) = (-5.9523r + 9.9202) + (x - 1)\overline{N}(x, r, \overline{v}) \end{cases}$$

The exact solution for $x = 2$ is

$$y(2, r) = [(9.5992r - 2.3415), (-10.4312r + 17.6889)], \quad r \in [1.5, 2]$$

The trial solutions for $x \in [1.5, 2]$ are

$$\begin{cases} \underline{y}_T(x) = (9.5992r - 2.3415) + (x - 1.5)\underline{N}(x, r, \underline{v}) \\ \overline{y}_T(x) = (-10.4312r + 17.6889) + (x - 1.5)\overline{N}(x, r, \overline{v}) \end{cases}$$

The numerical results of example are given in Tables 1–3. The exact and approximate solutions by artificial neural network are compared and plotted at $x = 1$, $x = 1.5$, and $x = 2$ in Figures 2–4, respectively. It is deduced that the results are very close to the exact solutions which confirm the validity and feasibility of this method.

Conclusion

In summary, we defined a new method for solving HFDEs. We demonstrated the ability of neural networks for approximating the solutions of FDEs. By comparing our achievements with the results obtained using numerical methods, it is clear that our proposed method gives more accurate approximations. Also better results (specially in nonlinear cases) might be possible if we use more neurons or training points. In addition, after solving a FDE, we obtained the solution at any arbitrary point in the training interval (even between training points). Applicability in function approximations of neutral networks is the main reason for using neural networks. More research is in progress for applying and extending this new approach for solving nth-order FDEs as well as a system of FDEs. The numerical results showed that the method has good accuracy and it is efficient.

Table 2. Comparison of the approximated solutions and the exact solutions on [1, 1.5].

r	Exact solution $(\underline{y}_a(1.5, r), \overline{y}_a(1.5, r))$	Approximated solution $(\underline{y}_T(1.5, r), \overline{y}_T(1.5, r))$	Absolute error
0.0	(–1.3227000, 9.9202000)	(–1.3226999, 9.9202001)	(5.689375e−07, 1.803075e−07)
0.1	(–0.79360000, 9.93249699)	(–0.79359992, 9.3249704)	(5.323851e−07, 1.929814e−07)
0.2	(–0.26450000, 8.7297400	(–0.26450009, 8.7297400)	(8.652878e−08, 3.078635e−07)
0.3	(0.26460000, 8.1345099)	(0.26459997, 8.1345101)	(1.503723e−06, 2.907790e−07)
0.4	(0.79370000, 7.5392800)	(0.79369997, 7.5392796)	(3.160333e−08, 4.194941e−07)
0.5	(1.32280000, 6.9440499)	(1.3227999, 6.9440496)	(1.548584e−07, 1.045301e−06)
0.6	(1.85190000, 6.3488200)	(1.8519000, 6.3488203)	(1.394677e−07, 7.646418e−07)
0.7	(2.38100000, 5.7535900)	(2.3809999, 5.7535900)	(4.624456e−08, 3.001244e−07)
0.8	(2.91010000, 5.1583599)	(2.9101000, 5.1583601)	(7.806130e−08, 4.992620e−07)
0.9	(3.43920000, 4.5631299)	(3.4392000, 4.5631302)	(2.682584e−07, 1.344480e−07)
1.0	(3.96830000, 3.9678999)	(3.9682999, 3.9678999)	(2.320238e−07, 0.256288e−08)

Table 3. Comparison of the approximated solutions and the exact solutions on [1.5, 2].

r	Exact solution $(\underline{y}_a(2, r), \overline{y}_a(2, r))$	Approximated solution $(\underline{y}_T(2, r), \overline{y}_T(2, r))$	Absolute error
0.0	(–2.341500, 17.688900)	(–2.341499, 17.688902)	(6.215545e−07, 1.875812e−06)
0.1	(–1.381580, 16.645780)	(–1.381582, 16.645789)	(2.279674e−06, 8.584055e−07)
0.2	(–0.421660, 15.602660	(–0.421661, 15.602662)	(1.967348e−07, 2.350311e−07)
0.3	(0.538260, 14.559540)	(0.538262, 14.559541)	(1.503723e−06, 0.211054e−07)
0.4	(1.498180, 13.516420)	(1.498185, 13.516421)	(1.733932e−06, 1.315011e−06)
0.5	(2.458100, 12.473300)	(2.458099, 12.473301)	(1.531159e−07, 1.104163e−06)
0.6	(3.418019, 11.430180)	(3.418017, 11.430185)	(2.301750e−07, 5.354311e−07)
0.7	(4.377940, 10.387060)	(4.377947, 10.387059)	(7.801521e−07, 8.412501e−07)
0.8	(5.337860, 9.343940)	(5.337862, 9.343942)	(1.301100e−07, 2.201216e−06)
0.9	(6.297780, 8.300820)	(6.297786, 8.300821)	(6.189354e−07, 1.521389e−08)
1.0	(7.257731, 7.257731)	(7.257732, 7.257732)	(1.719323e−07, 1.719218e−07)

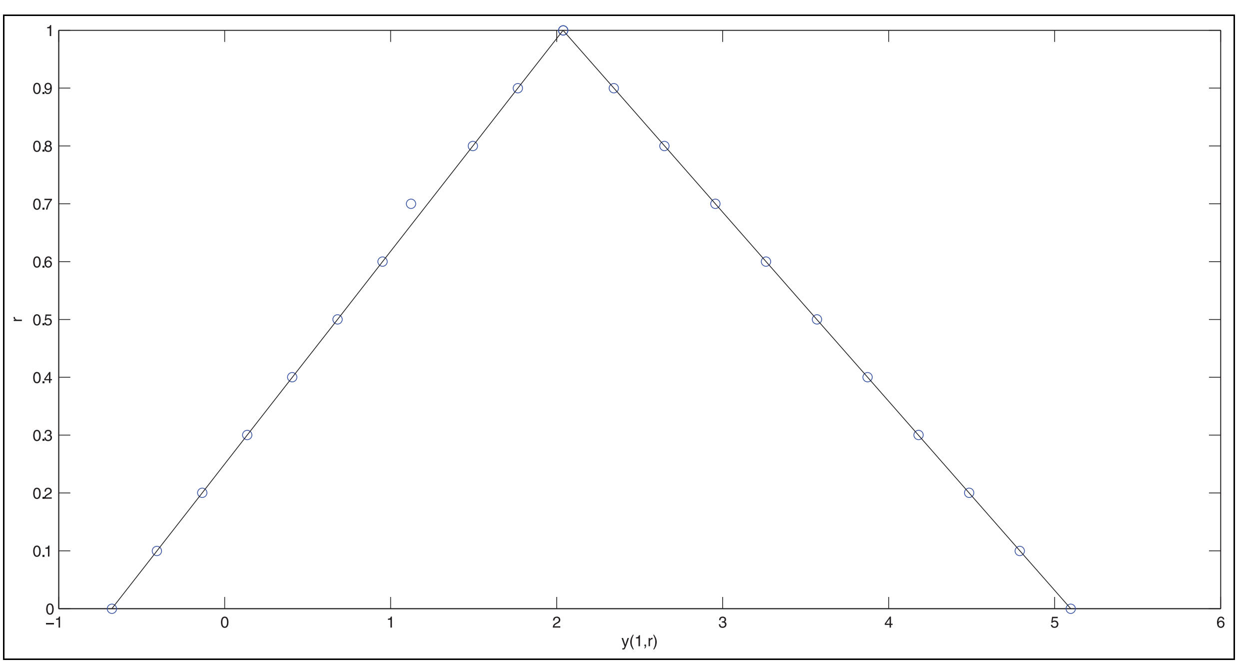

Figure 2. Comparison exact solution and approximated solution ◦ in $x = 1$.

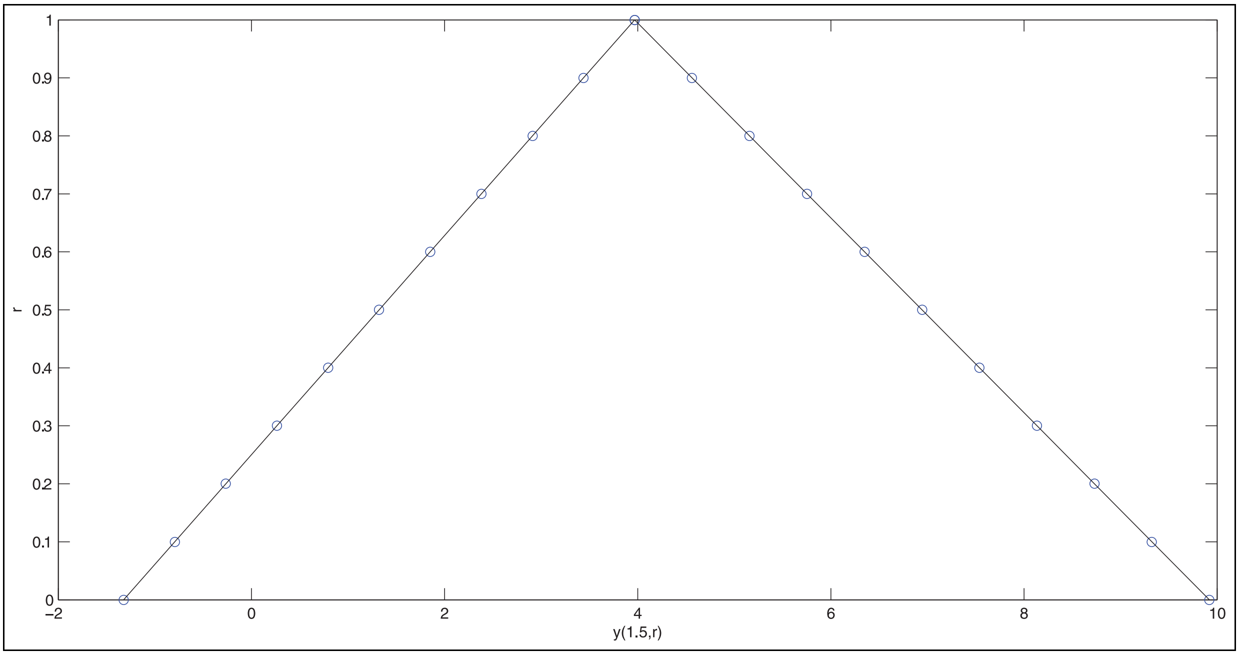

Figure 3. Comparison exact solution and approximated solution ∘ in $x = 1.5$.

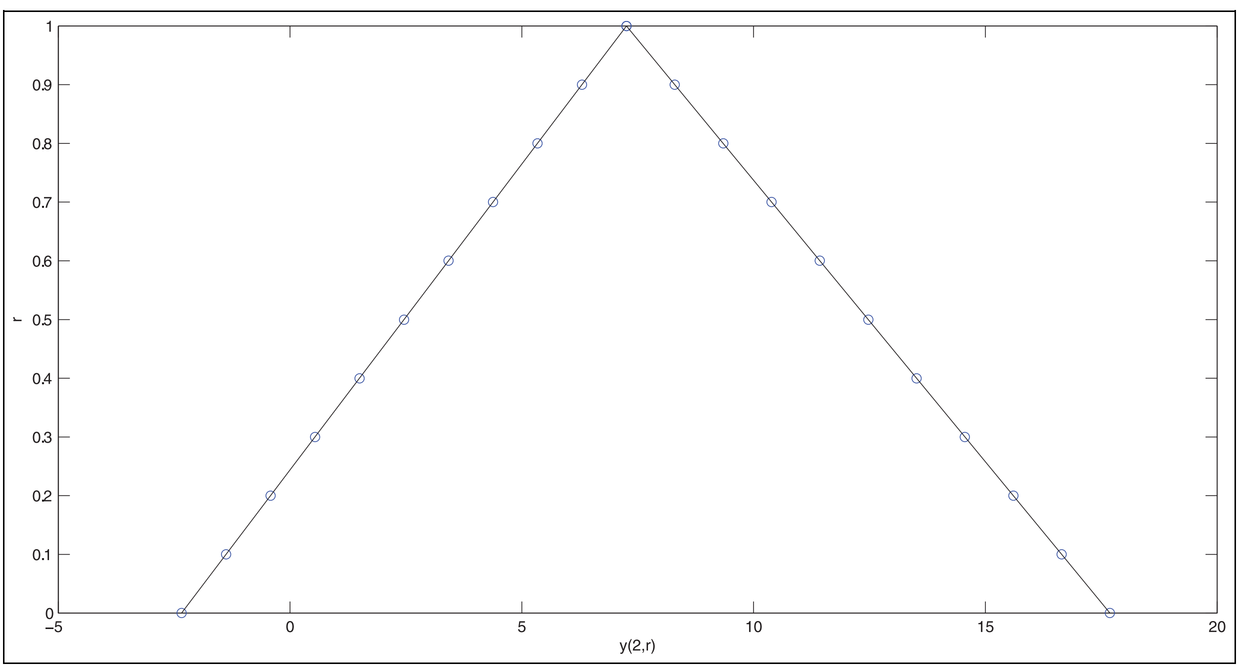

Figure 4. Comparison exact solution and approximated solution ∘ in $x = 2$.

Declaration of conflicting interests

The author(s) declared no potential conflicts of interest with respect to the research, authorship, and/or publication of this article.

Funding

The author(s) disclosed receipt of the following financial support for the research, authorship, and/or publication of this article: The second author, Massimiliano Ferrara, received a financial support from the iritMED, Istituto di Ricerca per l Innovazione e la Tecnologia nel Mediterraneo, for publication of this article.

References

1. Casasnovas J and Rossello F. Averaging fuzzy biopolymers. *Fuzzy Set Syst* 2005; 152: 139–158.

2. Feng G and Chen G. Adaptive control of discrete-time chaotic systems: a fuzzy control approach. *Chaos Soliton Fract* 2005; 23: 459–467.
3. Zhang H, Liao X and Yu J. Fuzzy modeling and synchronization of hyperchaotic systems. *Chaos Soliton Fract* 2005; 26: 835–843.
4. Kaleva O. Fuzzy differential equations. *Fuzzy Sets Syst* 1987; 24: 301–317.
5. Seikkala S. On the fuzzy initial value problem. *Fuzzy Sets Syst* 1987; 159: 319–330.
6. Lakshmikantham V and Liu XZ. Impulsive hybrid systems and stability theory. *Dyn Syst Appl* 1998; 7: 1–10.
7. Lakshmikantham V and Mohapatra RN. *Theory of fuzzy differential equations and inclusions*. London: Taylor and Francis, 2003.
8. Pederson S and Sambandham M. Numerical solution to hybrid fuzzy systems. *Math Comput Model* 2007; 45: 1133–1144.
9. Pederson S and Sambandham M. The Runge-Kutta method for hybrid fuzzy differential equation. *Nonlin Anal Hybrid Syst* 2008; 2: 626–634.
10. Prakash P and Kalaiselvi V. Numerical solution of hybrid fuzzy differential equations by predictor-corrector method. *Int J Comp Math* 2009; 86: 121–134.
11. Fard OS and Bidgoli TA. The Nystrom method for hybrid fuzzy differential equation IVPs. *J King Soud Univ* 2010; 23: 371–379.
12. Kim H and Sakthivel R. Numerical solution of hybrid fuzzy differential equations using improved predictor–corrector method. *Commun Nonlinear Sci Numer Simulat* 2012; 17: 3788–3794.
13. Allahviranloo T and Salahshour S. Euler method for solving hybrid fuzzy differential equation. *Soft Comput* 2011; 15: 1247–1253.
14. Ahmadian A, Suleiman M, Ismail F, et al. A Runge-Kutta method with lower function evaluations for solving hybrid fuzzy differential equations. *Intell Inform Database Syst* 2013; 7802: 265–274.
15. Ahmadian A, Chang CS and Salahshour A. Runge-Kutta method with the reduced number of function evaluations for solving hybrid fuzzy differential equations. *Soft Comput* 2015; 19: 1051–1062.
16. Ahmadian A, Salahshour S, Chan CS, et al. Numerical solutions of fuzzy differential equations by an efficient Runge-Kutta method with generalized differentiability. *Fuzzy Sets Syst*. Epub ahead of print 1 December 2016. DOI: 10.1016/j.fss.2016.11.013.
17. Paripour M, Hajilou E, Hajilou A, et al. Application of Adomian decomposition method to solve hybrid fuzzy differential equations. *J Taibah Univ* 2015; 9: 95–103.
18. Lagaris IE, Likas A and Fotiadis DI. Artificial neural networks for solving ordinary and partial differential equations. *IEEE T Neural Netw* 1998; 9: 987–1000.
19. Effati S and Pakdaman M. Artificial neural network approach for solving fuzzy differential equations. *Inform Sci* 2010; 180: 1434–1457.
20. Pakdaman M, Ahmadian A, Effati S, et al. Solving differential equations of fractional order using an optimization technique based on training artificial neural network. *Appl Math Comput* 2017; 293: 81–95.
21. Bede B. Note on numerical solutions of fuzzy differential equations by predictor corrector method. *Inform Sci* 2008; 178: 1917–1922.
22. Pederson S and Sambandham M. Numerical solution of hybrid fuzzy differential equation IVPs by a characterization theorem. *Inform Sci* 2009; 179: 319–328.
23. Anastassiou GA and Gal SG. On a fuzzy trigonometric approximation theorem of weierstrass-type. *J Fuzzy Math* 2001; 9: 701–708.
24. Anastassiou GA. *Fuzzy mathematics: approximation theory*. Berlin, Heidelberg: Springer, 2010.
25. Guang-Quan Z. Fuzzy continuous function and its properties. *Fuzzy Sets Syst* 1991; 43: 159–171.
26. Chalco-Cano Y and Roman-Flores H. On new solutions of fuzzy differential equations. *Chaos Soliton Fract* 2008; 38: 112–119.
27. Bhaskar TG, Lakshmikantham V and Devi V. Revisiting fuzzy differential equations. *Nonlinear Anal* 2004; 58: 351–359.
28. Bede B and Gal SG. Generalizations of the differentiability of fuzzy number value functions with applications to fuzzy differential equations. *Fuzzy Sets Syst* 2005; 151: 581–590.