Hybrid Neural Network Control for Uncertain Nonlinear Discrete-Time Systems with Bounded Disturbance

Rahul Kumar[1] · Uday Pratap Singh[1] · Arun Bali[1] · Kuldip Raj[1]

Abstract

This paper has proposed an adaptive control scheme based on a hybrid neural network (HNN) to address the problem of uncertain nonlinear systems with discrete-time having bounded disturbances. This proposed control scheme is composed of a neural network (NN) and differential evolution (DE) technique which is used to initialize the weights of the NN and the controller is designed in such a manner so that the stability can be ensured and the desired trajectory can be achieved. The designed HNN is employed to approximate unknown functions present in the system. By using the concept of system transformation, the adaptive law and controller are designed and the whole system is proved to be stable in the sense of semi-globally uniformly ultimately boundedness (SGUUB) with the assistance of Lyapunov theory. Finally, the validity and effectiveness of the results are proved through two simulation examples.

Keywords Hybrid neural network · Uncertain nonlinear systems · Lyapunov method · Bounded disturbances · Differential evolution

1 Introduction

In the past two decades, a great deal of attention is devoted by the researchers to the area of uncertain nonlinear discrete-time systems such as robot manipulators, servo motor, and underwater vehicle systems [1–3]. Due to the development of technology control devices and objects are becoming more complex and not possible to control via a linear control model [4, 5]. Therefore the control of nonlinear systems is becoming much more challengeable since not all states are measurable and it becomes more difficult in the presence of disturbances and hence drawing a lot of attention from researchers [6–9]. Due to uncertainty and disturbances, it has been acknowledged that the methodologies to control uncertain nonlinear systems are comparatively more difficult than the linear control [10]. Bounded disturbances often exist in many practical problems and become the cause of the instability of control systems. Therefore, designing an efficient control scheme and its stability attract the attention of researchers. However, due to limitations

✉ Uday Pratap Singh
  usinghiitg@gmail.com

1 School of Mathematics, Shri Mata Vaishno Devi University, Katra 182320, JK, India

in modeling, measurement and observations many physical systems are affected by parameter uncertainties.

Uncertain systems appear very often in the numerous practical plants, in this connection the adaptive neural network (ANN) control and the control based on fuzzy logic systems (FLS) have achieved extensive accomplishments and improvements to approximate nonlinear functions present in such systems [11–14]. Many methods were developed like back-stepping, sliding mode control, observer-based system, and robust control which are used to tackle such nonlinear systems [15–18]. An adaptive tracking control methodology was discussed for uncertain nonlinear single-input and single-output (SISO) systems [19] and multi-input and multi-output (MIMO) systems in [17, 20, 21]. Tracking control of uncertain systems is more difficult in presence of diverse forces which may interfere with the systems [22]. Among many diverse forces, the dead zone is one force that is symmetric and asymmetric in nature [23]. Adaptive tracking control is developed for switching systems with non-symmetric dead-zone [24], in addition to this, observer-based adaptive control and some fuzzy based adaptive control techniques addressing nonlinear systems with discrete-time and bounded disturbances with dead-zone are presented in [25–28] while nonlinear time-delay problem with unknown nonlinear dead-zone is presented in [27, 29, 30].

In the assessment with continuous systems, discrete-time systems have more practical consequences. Although a large number of the real-time processes are continuous in nature, they are demonstrated mostly utilizing discrete-time as they are effectively feasible [31]. Keeping into view this advantage, the discrete-time systems are taken into consideration. Many adaptive control techniques were developed to approximate the unknown functions for nonlinear discrete-time systems [23, 30, 32, 33]. The time-changing parameters in the systems may cause major challenges in tracking performance. Another important issue is time-delay input for nonlinear systems which usually makes a system complicated. Back-stepping based adaptive control design employing intelligent systems like neural networks or fuzzy logic systems (FLS) gives more fruitful results for nonlinear systems [17, 34–36].

Motivated by the aforementioned discussion, it has been noticed that an adaptive control strategy based on NN or FLS, ensured the basic system stability, but how to attain the preferred trajectory remains to be studied additionally. To enhance the theory of adaptive control, especially for nonlinear systems with discrete-time, we propose an adaptive controller for uncertain nonlinear systems with discrete-time on the basis of a hybrid neural network. In the proposed hybrid neural network controller, we use the evolutionary differential evolution method to initialize the weight and biases of the neural network with mean square error (MSE) as the fitness function. In this work, we address HNN adaptive controller to attain the reference signal for uncertain nonlinear systems with discrete-time having bounded disturbances and immersion of utterly unknown functions.

The primary endowments of this work are summed up as follows:

(1) An evolutionary method named differential evolution is used for the initialization of the weight vector of the neural network. Employing the HNN and feedback approach, the proposed control scheme is capable of successfully tracking the reference signal. The considered adaptive HNN is used to approximate unknown functions involving dead-zone.

(2) The proposed adaptive control scheme surmounts the requirements of affine form and linear parametric situation, i.e. it is not required to be assumed that the nonlinear uncertainties are needed to be linear with unknown parameters.

(3) By the use of the Lyapunov method, the semi-global uniformly ultimate boundedness (SGUUB) of the system is verified. The unknown functions are be approximated by the proposed HNN controller and the error signal converges in a compact set containing zero, which is shown and verified by the using simulation examples.

The remaining part of this paper has been structured in the following manner. The formulation of the problem for uncertain nonlinear systems with discrete-time having dead-zone and preliminaries is presented in Sect. 2. In Sect. 3, the HNN controller is discussed, and Sect. 4 deals with the stability analysis of the system using the Lyapunov method. Section 5 provides the simulation results and description, and the conclusion of this paper carries out in Sect. 6.

Notation: Following are the notations that will be used throughout this paper.

In the whole of the article, all work is done using standard notations. $\mathfrak{R}^m$ denotes an m-dimensional Euclidean space; $\overline{\zeta}_i(k)$ stands for the ith state of the system and $\overline{\zeta}_i(k) = \left[\zeta_1(k), \zeta_2(k), \cdots, \zeta_i(k)\right]^T \in \mathfrak{R}^i$ represents state vector where T is the transpose of vector/matrix and $_s(k)$ be the supremum of $\zeta(\bullet)$; $\varepsilon(k)$ represents tracking error; $\varepsilon()$ represents approximation error for NN and ε is a constant. $y(k)$ and $y_d(k)$ represents obtained and the desired trajectory respectively.

2 Problem Formulation and Preliminaries

Let us take into consideration the under mentioned nonlinear stochastic pure-feedback systems as follows:

$$
\begin{cases}
\zeta_1(k + 1) = \psi_1\big(\zeta_1(k)\big) + \phi_1\big(\zeta_1(k)\big)\zeta_2(k) \\
\qquad\qquad \vdots \\
\zeta_{n-1}(k + 1) = \psi_{n-1}\big(\zeta_1(k), \cdots, \zeta_{n-1}(k)\big) + \phi_{n-1}\big(\zeta_1(k), \cdots, \zeta_{n-1}(k)\big)\zeta_n(k) \\
\zeta_n(k + 1) = \psi_n\big(\zeta_1(k), \cdots, \zeta_n(k)\big) + \phi_n\big(\zeta_1(k), \cdots, \zeta_n(k)\big)\tau(k) + d(k) \\
y(k) = \zeta_1(k)
\end{cases}
\tag{1}
$$

where, $\overline{\zeta}_i(k) = \left[\zeta_1(k), \zeta_2(k), \cdots, \zeta_i(k)\right]^T \in \mathfrak{R}^i$, where $i = 1, 2, \cdots, n, \tau(k) \in \mathfrak{R}$ and $y(k) \in \mathfrak{R}$ be the state vector, input and output of the system respectively; $d(k) \in \mathfrak{R}^m$ stand for external bounded disturbances such that $\|d(k)\| \leq d_M$, and $\psi_i(\zeta(k)); \forall i = 1, 2, \cdots, n$ and $\phi_i(\zeta(k)); \forall i = 1, 2, \cdots, n$ are unknown smooth and bounded functions with all its partial derivatives are also continuous. The dead zone function $\tau(k)$ in Eq. (1) is defined as:

$$
\tau(k) = \begin{cases}
\varphi_r(v(k)) : & v(k) \geq b_r \\
0 & : \quad b_l < v(k) < b_r \\
\varphi_l(v(k)) : & v(k) \leq -b_l
\end{cases}
\tag{2}
$$

where $v(k) \in \mathfrak{R}$ is known as dead zone for the input, $b_l < 0$ and $b_r > 0$ be the unknown dead zone parameters while φ_r and φ_l are the unknown smooth nonlinear function. The functions $\varphi_r(v(k))$ and $\varphi_l(v(k))$ are smooth functions.

$$
\tau(k) = \left[C_r(v(k)), C_l(v(k))\right] \left[\hbar_r(v(k)), \hbar_l(v(k))\right]^T v(k) + p(v(k))
\tag{3}
$$

where

$$C_r(\upsilon(k)) = \begin{cases} 0, & \upsilon(k) \le b_l \\ \varphi_r^{'}(b_r), & b_l < \upsilon(k) < b_r \\ \varphi_r^{'}(\eta_r(\upsilon(k))), & b_r \le \upsilon(k) < \infty \end{cases} , \quad C_l(\upsilon(k)) = \begin{cases} \varphi_l^{'}(\eta_l(\upsilon(k))), & -\infty < \upsilon(k) \le b_l \\ \varphi_l^{'}(b_l), & b_l < \upsilon(k) < b_r \\ 0, & \upsilon(k) \ge b_r \end{cases}$$

$$\hbar_r(\upsilon(k)) = \begin{cases} 1, & \upsilon(k) > b_l \\ 0, & \upsilon(k) \le b_l \end{cases} , \quad \hbar_l(\upsilon(k)) = \begin{cases} 1, & \upsilon(k) < b_r \\ 0, & \upsilon(k) \ge b_r \end{cases} ,$$

$$\eta_r(\upsilon(k) \in (b_r, \infty) \text{ and } \eta_l(\upsilon(k) \in (-\infty, b_l)$$

Let $y_d(k)$ be the reference signal for the system (1) and $y(k)$ is the calculated output of the system (1). Then our objective is to minimize the usual quadratic cost function given in (4):

$$E = \frac{1}{2} \sum_{k=1}^{n} \left(y(k) - y_d(k) \right)^2 \tag{4}$$

Definition [37]

The Nussbaum gain $N(\zeta(k))$ of a discrete sequence $\{\zeta(k)\}$ for discrete time satisfy following properties:

$$\zeta(0) = 0, \ \zeta(k) \ge 0 \text{ and } \Delta\zeta(k) = |\zeta(k+1) - \zeta(k)| < \varepsilon; \forall k$$

where, ε is constant.

Then $N(\zeta(k))$ defined as follows:

$$N(\zeta(k)) = \zeta_s(k)H(\zeta(k)) \tag{5}$$

where, $\zeta_s(k) = \sup_{k' \le k} \{\zeta(k')\}$ and $H(\zeta(k))$ is defined as:

$$H(\zeta(0)) = 1 \text{ for } k = k_1$$

If $H(\zeta(k_1)) = 1$, then if $\sum_{k'=0}^{k_1} N(\zeta(k')) \Delta\zeta(k') > \zeta_s^{3/2}(k_1)$

In this case, $H(\zeta(k_1 + 1)) = -1$ otherwise $H(\zeta(k_1 + 1)) = 1$.

If $H(\zeta(k_1)) = -1$, then if $\sum_{k'=0}^{k_1} N(\zeta(k')) \Delta\zeta(k') < -\zeta_s^{3/2}(k_1)$.

In this case, $H(\zeta(k_1 + 1)) = 1$ otherwise $H(\zeta(k_1 + 1)) = -1$.

Result [37] If $\zeta_s(k)$ holds uniform boundedness, then same is true for $|H(\zeta(k))|$ and if $\zeta_s(k)$ is unbounded then $|H(\zeta(k))|$ oscillates between $-\infty$ and $+\infty$. Using this result we establish lemma 1.

Lemma 1 [37] *If the function $V(\zeta)$ is such that it is positive definite and $N(\zeta)$ is Nussbaum gain defined in Eq.* (5) *and if inequality* (6) *holds*:

$$V(k) \le \sum_{k'=0}^{k_1} \left(c_1 + \lambda N(\zeta(k')) \right) \Delta\zeta(k') + c_2\zeta(k) + c_3 \tag{6}$$

where c_1, c_2 and c_3 are constants then $V(k), \zeta(k)$ and RHS of Eq. (6) *must be bounded.*

NN Approximation

In the control theory, there are several well-formulated methods for unknown function approximations namely the polynomial method, fuzzy logic, and neural network, etc. In this work we use a neural network as an unknown function approximator. The basic universal approximation result says that any smooth function $\psi(\zeta)$ can be approximated arbitrarily on a closed compact set with appropriate weight adjustment using NN. If $\psi(\zeta)$ be an unknown smooth function can be written as:

$$\psi(\xi) = W^T \sigma(\xi) + \varepsilon(\xi) \tag{7}$$

where, $\varepsilon(\xi)$ is known as NN function approximation error and bounded by $\bar{\varepsilon}$ i.e. $\|\varepsilon(\xi)\| \le \bar{\varepsilon}$. The error function defined in Eq. (3) can be minimized via choosing optimal weight vector W, for this let us consider Lyapunov function as:

$$V = \frac{1}{2} e(k) e^T(k) \tag{8}$$

where $e(k) = y(k) - y_d(k)$ and Lyapunov function defined above is same as the usual quadratic cost function. The derivative of Eq. (8) is given by:

$$\dot{V} = -e^T \frac{\partial e}{\partial W} \dot{W} \quad \text{or} \quad \dot{V} = -e^T J \dot{W} \tag{9}$$

where $J = \frac{\partial e}{\partial W}$. If arbitrary initial weight $W(0)$ is updated by the rule

$$W(t_1) = W(0) + \int_0^{t_1} \dot{W} dt \dot{W} = \frac{\|e\|^2}{\|J^T e\|^2} J^T e \tag{10}$$

where error e converges to zero under the condition that $\dot{W}$ exist along the convergence trajectory. Using Eq. (10), Eq. (6) yields: $\dot{V} = -\|e\|^2 \le 0$ and $\dot{V} < 0$ if $e \ne 0$. The difference equation representation of weight updating algorithm based on Eq. (10) is given by:

$$W(t+1) = W(t) + \alpha \dot{W}(t) \tag{11}$$

where α is known as learning rate.

The structure of the hybrid neural network proposed in this work can be seen in Fig. 1, where $[x_1, x_2, ..., x_n]$ is the input state vector and $[y_1, y_2, ..., y_m]$ is the initial weight matrix defined by using differential evolution, σ is the activation function and V_{jk} is the weight matrix from hidden to output layer and $[y_1, y_2, ..., y_m]$ is the output vector.

3 Adaptive Controller Design

In this part of paper, we will use pure feedback technique to design an adaptive controller $\tau(k)$ in Eq. (1) or $v(k)$ which is defined in Eq. (3). The detail system transformation, design procedure and stability analysis are described below:

Step 1: Let first design the virtual controller $v(k)$ as defined in Eq. (3), start the virtual control design to stabilize the first part of Eq. (1) as follows:

$$\zeta_{2\psi}(k) = -\frac{1}{\phi_1(\zeta_1(k))} \left[\psi_1(\zeta_1(k)) - y_r(k+1) \right] \tag{12}$$
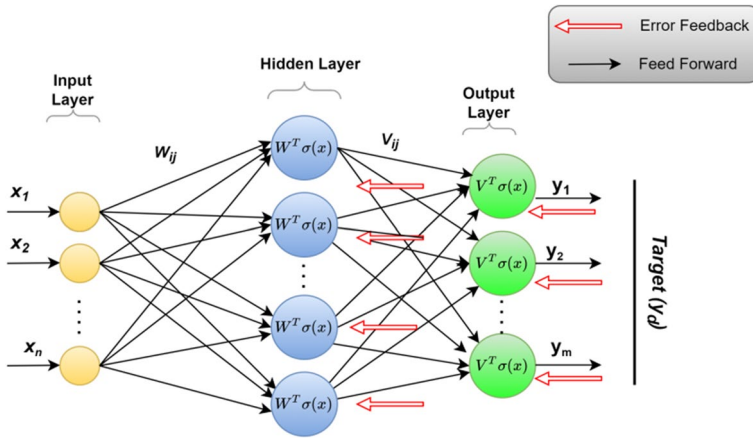
Fig. 1 Neural network

Similarly the virtual control design to stabilize the second part of Eq. (1) as follows:

$$\zeta_{3\psi}(k) = -\frac{1}{\phi_2\big(\zeta_1(k), \zeta_2(k)\big)}\big[\psi_2\big(\zeta_1(k), \zeta_2(k)\big) - \zeta_{2\psi}(k+1)\big] \tag{13}$$

Repeat the above procedure to get the final control $\tau(k)$, by examine the first $(n\text{-}1)$ equation of system (1). For this let us denote:

$$\zeta_i(k+1) = \rho_i\big(\overline{\zeta}_{i+1}(k)\big); \forall i = 1, \cdots, n-2 \tag{14}$$

where$\rho_i\big(\overline{\zeta}_{i+1}(k)\big) = \psi_i\big(\zeta_1(k+1), \cdots, \zeta_i(k+1)\big) + \phi_i\big(\zeta_1(k+1), \cdots, \zeta_i(k+1)\big)\zeta_{i+1}(k+1)$

Therefore:

$$\overline{\zeta}_i(k+1) = \begin{bmatrix} \zeta_1(k+1) \\ \vdots \\ \zeta_i(k+1) \end{bmatrix} = \begin{bmatrix} \rho_1\big(\overline{\zeta}_{1+1}(k)\big) \\ \vdots \\ \rho_i\big(\overline{\zeta}_{i+1}(k)\big) \end{bmatrix} = \Psi_i\big(\overline{\zeta}_{i+1}(k)\big) \in \mathfrak{R}^i; \forall i = 1, 2, \cdots, n-1$$

After moving one step ahead system (1) can be expressed as:

$$\begin{aligned} \zeta_i(k+2) = {}& \psi_i\big(\zeta_1(k+1), \cdots, \zeta_i(k+1)\big) \\ & +\phi_i\big(\zeta_1(k+1), \ldots, \zeta_i(k+1)\big) \times \zeta_{i+1}(k+1); \forall i = 1, \ldots, n-2 \end{aligned} \tag{15}$$

$$\zeta_{n-1}(k+2) = \psi_{n-1}\big(\zeta_1(k+1), \ldots, \zeta_{n-1}(k+1)\big) + \phi_{n-1}\big(\zeta_1(k+1), \ldots, \zeta_{n-1}(k+1)\big) \times \zeta_n(k+1) \tag{16}$$

Equations (15, 16) can be rewritten as:

$$\begin{aligned} \zeta_i(k+2) = {}& \rho_i\big(\overline{\zeta}_{i+2}(k)\big); \text{ for all } i = 1, \cdot, n-2 \\ \zeta_{n-1}(k+2) = {}& \Psi_{n-1}\big(\overline{\zeta}_n(k)\big) + \Phi_{n-1}\big(\overline{\zeta}_n(k)\big) \times \zeta_n(k+1) \end{aligned} \tag{17}$$

where

$$\rho_i\left(\overline{\zeta}_{i+2}(k)\right) = \psi_i\left(\Psi_i\left(\overline{\zeta}_{i+1}(k)\right)\right) + \varphi_i\left(\Psi_i\left(\overline{\zeta}_{i+1}(k)\right)\right) \times \rho_{i+1}\left(\overline{\zeta}_{i+2}(k)\right),$$

$$\Psi_{n-1}\left(\overline{\zeta}_n(k)\right) = \psi_{n-1}\left(\Psi_{n-1}\left(\overline{\zeta}_n(k)\right)\right), \quad \Phi_{n-1}\left(\overline{\zeta}_n(k)\right) = \varphi_{n-1}\left(\Psi_{n-1}\left(\overline{\zeta}_n(k)\right)\right)$$

Using Eq. (15) and moving one step ahead the first (n-2) terms at the ($k+2$)th step in (17) we have:

$$\overline{\zeta}_i(k+2) = \begin{bmatrix} \zeta_1(k+2) \\ \vdots \\ \zeta_i(k+2) \end{bmatrix} = \begin{bmatrix} \rho_1\left(\overline{\zeta}_{1+1}(k+1)\right) \\ \vdots \\ \rho_{i+1}\left(\overline{\zeta}_{i+1}(k+1)\right) \end{bmatrix} = \begin{bmatrix} \rho_1\left(\Psi_2\left(\overline{\zeta}_3(k)\right)\right) \\ \vdots \\ \rho_i\left(\Psi_{i+1}\left(\overline{\zeta}_{i+2}(k)\right)\right) \end{bmatrix}$$

From Eq. (15) it is clear that

$$\overline{\zeta}_i(k+2) = \Psi_i\left(\overline{\zeta}_{i+2}(k)\right); \text{for all } i = 1, 2, \cdots, n-2 \tag{18}$$

Similar way to above process, we have

$$\zeta_1(k+n-1) = \rho_1\left(\overline{\zeta}_n(k)\right)$$
$$\zeta_2(k+n-1) = \Psi_2\left(\overline{\zeta}_n(k)\right) + \Phi_2\left(\overline{\zeta}_n(k)\right) \times \zeta_3(k+n-2) \tag{19}$$

where $\rho_1\left(\overline{\zeta}_n(k)\right) = \psi_1\left(\Psi_1\left(\overline{\zeta}_{n-1}(k)\right)\right) + \varphi_1\left(\Psi_1\left(\overline{\zeta}_{n-1}(k)\right)\right) \times \psi_2\left(\overline{\zeta}_n(k)\right)$ and

$$\Psi_2\left(\overline{\zeta}_n(k)\right) = \psi_2\left(\Psi_2\left(\overline{\zeta}_n(k)\right)\right)$$

$$\Phi_2\left(\overline{\zeta}_n(k)\right) = \varphi_2\left(\Psi_2\left(\overline{\zeta}_n(k)\right)\right)$$

Now at the nth step, we have

$$\zeta_1(k+n) = \Psi_1\left(\overline{\zeta}_n(k)\right) + \Phi_1\left(\overline{\zeta}_n(k)\right)\zeta_2(k+n-1) \tag{20}$$

where $\Psi_1\left(\overline{\zeta}_n(k)\right) = \psi_1\left(\rho_1\left(\overline{\zeta}_n(k)\right)\right)$ and $\Phi_1\left(\overline{\zeta}_n(k)\right) = \varphi_1\left(\rho_1\left(\overline{\zeta}_n(k)\right)\right)$.

From Eqs. (16) to (20), Eq. (1) can be re-casted as:

$$\begin{cases} \zeta_1(k+n) = \Psi_1\left(\overline{\zeta}_n(k)\right) + \Phi_1\left(\overline{\zeta}_n(k)\right) \times \zeta_2(k+n-1) \\ \quad\quad \vdots \\ \zeta_{n-1}(k+2) = \Psi_{n-1}\left(\overline{\zeta}_n(k)\right) + \Phi_{n-1}\left(\overline{\zeta}_n(k)\right) \times \zeta_n(k+1) \\ \zeta_n(k+1) = \psi_n\left(\overline{\zeta}_n(k)\right) + \phi_n\left(\overline{\zeta}_n(k)\right) \times \tau(k) + d(k) \\ y(k+n) = \zeta_1(k+n) \end{cases} \tag{21}$$

From Eq. (19) to (20), we have

$$\zeta_1(k+n) = \Psi_1(k) + \Psi_2(k) \times \Phi_1(k) + \Phi_1(k) \times \Phi_2(k) \times \zeta_3(k+n-2) \tag{22}$$

Repeating above process, until $\tau(k)$ appears

$$\zeta_1(k+n) = \Psi\left(\overline{\zeta}_m(k)\right) + \Phi\left(\overline{\zeta}_m(k)\right) \times \tau(k) + d_1(k) \tag{23}$$

where $\Psi\left(\overline{\zeta}_m(k)\right) = \Psi_1(k) + \Psi_2(k) \times \Phi_1(k) + \cdots + \Psi_{n-1}(k) \times \prod_{i=1}^{n-2} \Phi_j(k) + \psi_n(k) \times \prod_{i=1}^{n-2} \Phi_j(k)$

$$\Phi\left(\overline{\zeta}_m(k)\right) = \varphi_n(k) \times \prod_{i=1}^{n-1} \Phi_j(k)$$

$$d_1(k) = d(k) \times \prod_{i=1}^{n-1} \Phi_j(k)$$

Then the output of the re-casted system is given as:

$$y(k+n) = \Psi\left(\overline{\zeta}_m(k)\right) + \Phi\left(\overline{\zeta}_m(k)\right) \times \tau(k) + d_1(k) \tag{24}$$

Let us define tracking error $\varepsilon(k) = y(k) - y_d(k)$, from Eq. (24) error is:

$$\varepsilon(k+n) = \Psi\left(\overline{\zeta}_m(k)\right) + \Phi\left(\overline{\zeta}_m(k)\right)\tau(k) + d_1(k) - y_d(k+n) \tag{25}$$

Since, the dead zone $\tau(k)$ defined in Eq. (2) is containing unknown variables, therefore $\tau(k)$ can be used for controller in terms of Eq. (3):

$$\varepsilon(k+n) = \Psi\left(\overline{\zeta}_m(k)\right) + \Phi\left(\overline{\zeta}_m(k)\right)\left[C^T(\upsilon(k))\hbar_r(\upsilon(k))\upsilon(k) + p(\upsilon(k))\right] + d_1(k) - y_d(k+n) \tag{26}$$

where, $C^T(\upsilon(k)) = \left[C_r(\upsilon(k)), \; C_l(\upsilon(k))\right].$
For simplification let us denote:

$$\Gamma\left(\overline{\zeta}_m(k), \upsilon(k)\right) = \Psi\left(\overline{\zeta}_m(k)\right) + \Phi\left(\overline{\zeta}_m(k)\right) \times C^T(\upsilon(k)) \times \hbar_r(\upsilon(k))\upsilon(k)$$

Then Eq. (26) becomes

$$\varepsilon(k+n) = \Gamma\left(\overline{\zeta}_m(k), \upsilon(k)\right) - y_d(k+n) + \Phi\left(\overline{\zeta}_m(k)\right) \times p(\upsilon(k)) + d_1(k) \tag{27}$$

By using implicit theorem, find the required control input $\upsilon^*(k)$ such that following holds

$$\Gamma\left(\upsilon^*(k), \overline{\zeta}_m(k)\right) - y_d(k+n) = 0 \tag{28}$$

With the neural network approximation as in Eq. (7), we get

$$\upsilon^*(k) = W^{*T}\sigma(\upsilon(k)) + \varepsilon^*(\upsilon(k)) \tag{29}$$

where, $\sigma(\upsilon(k))$ is the bounded input to the hidden layer and $\varepsilon^*(\upsilon(k))$ error after approximation which is to be minimized, for this let us assume $\hat{W}$ is the approximation of W^* and $\overline{W} = \hat{W} - W^*$. Using this construct HNN controller for $\upsilon(k)$ as

$$v(k) = \hat{W}^T \sigma(v(k)) + \hat{\varepsilon}(v(k)) \tag{30}$$

where, $\hat{\varepsilon}$ is the estimation of ε^* and $\overline{\varepsilon} = \hat{\varepsilon} - \varepsilon^*$. Using mean value theorem and Eq. (28), (27) can be further written as:

$$
\begin{aligned}
\varepsilon(k+n) &= \Gamma\left(\overline{\zeta}_m(k), v(k)\right) - \Gamma\left(\overline{\zeta}_m(k), v^*(k)\right) + d_1(k) + \Phi\left(\overline{\zeta}_m(k)\right)p(v(k)) \\
&= g\left(\overline{\zeta}_m(k), v^\#(k)\right)\left(v(k) - v^\#(k)\right) + d_1(k) + \Phi\left(\overline{\zeta}_m(k)\right)p(v(k))
\end{aligned}
\tag{31}
$$

where $v^\#(k) \in [\min\{v^*(k), v(k)\}, \max\{v^*(k), v(k)\}]$

Using Eqs. (29, 30, 31) becomes.

$$
\begin{aligned}
\varepsilon(k+n) &= g\left(\overline{\zeta}_m(k), v^\#(k)\right) \times \left(\overline{W}^T \sigma(v(k)) + \hat{\varepsilon}(v(k)) - \overline{\varepsilon}(v(k))\right) + \tau_1(k) + \Phi\left(\overline{\zeta}_m(k)\right) \times p(v(k)) \\
&= g\left(\overline{\zeta}_m(k), v^\#(k)\right) \times (\pi(k) + \varepsilon^*(v(k))) + d_0(k)
\end{aligned}
$$

where $\pi(k) = \overline{W}^T \sigma(v(k))$ and $d_0(k) = g\left(v^\#(k), \overline{\zeta}_m(k)\right) \times \overline{\varepsilon}(v(k)) + d_1(k) + \Phi\left(\overline{\zeta}_m(k)\right)p(v(k))$.

Now, the parameter adjustment regulation are designed as

$$\hat{W}(k+1) = \hat{W}(k_1) - \kappa N(\zeta(k))\sigma(v(k_1))\frac{\alpha(k)\chi(k+1)}{B(k)} \tag{32}$$

$$\hat{\varepsilon}(k+1) = \hat{\varepsilon}(k_1) - (\chi(k+1)) \times \kappa\frac{N(\zeta(k)) \times \alpha(k)}{B(k)} \tag{33}$$

where, κ is positive adaptation gain constant, $\alpha(k)$ is augmented error and $k_1 = k - n + 1$.

$$\chi(k+1) = \kappa\left(\frac{\varepsilon(k+1)}{1 + |N(\zeta(k))|}\right)$$

$$B(k) = 1 + \chi^2(k+1) + \left\|\sigma(v(k_1))\right\| + N(\zeta(k))$$

$$\alpha(k) = \begin{cases} 1, & |\chi(k+1)| > \kappa \\ 0, & otherwise \end{cases}$$

$\Delta\zeta(k) = \zeta(k+1) - \zeta(k) = \frac{\alpha(k) \times (1 + |N(\zeta(k))|) \times \chi^2(k+1)}{B(k)}$. For the design of controller, the sequence $\{\zeta(k)\}$ must satisfy $\zeta(0) = 0, \zeta(k) \geq 0$ and $\Delta\zeta(k) = |\zeta(k+1) - \zeta(k)| < \varepsilon$; for all k.

The overall working of the control process can be well understood through Figs. 2 and 3.

Algorithm A step-by-step procedure of detailed implementation of the controller is as follows:

1. Select initial state and parameters.
2. Construct HNN and evaluate error, $y(k) - y_d(k)$.
3. Update the weight. Calculate the controller and construct equation
4. Use the controller $v(k)$ for the given system.
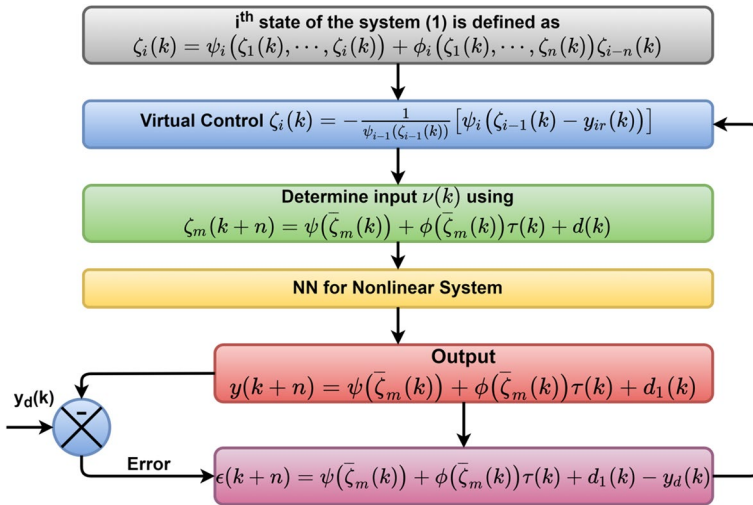5. Go to step 3 for next updation.

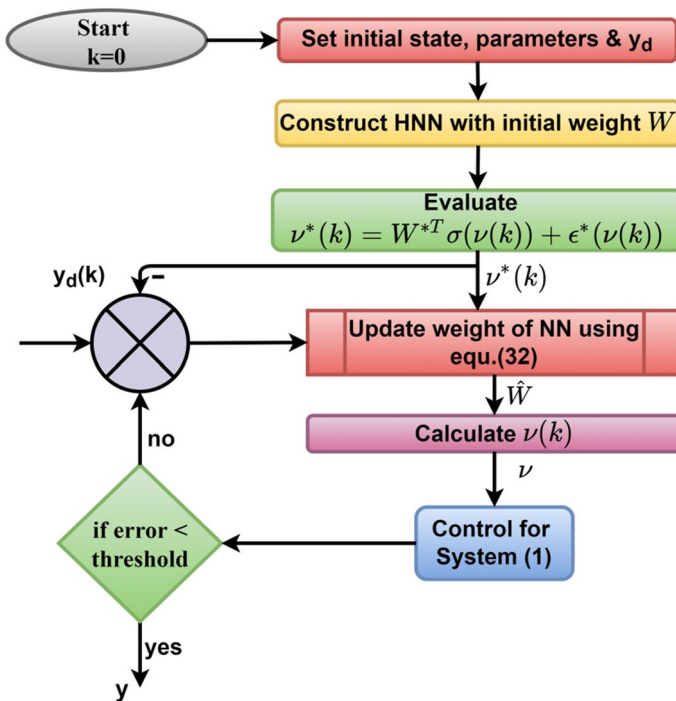**Fig. 2** Working process of the Controller

Fig. 3 Flow Chart for the Algorithm

4 Analysis of Stability

Theorem: If a nonlinear system of the type (1) with dead zone. To construct the HNN controller with parameter adaption law defined in Eqs. (32, 33) with appropriate design parameters, the feedback systems guarantee that system possess semi-global uniformly ultimate boundedness.

Proof: Let us consider the Lyapunov function:

$$V(k) = V_1(k) + V_2(k)$$

where, $V_1(k) = \sum_{j=0}^{n-1} \overline{W}^T(k_1+j)\,\overline{W}(k_1+j)$ and $V_2(k) = \sum_{j=0}^{n-1} \overline{\varepsilon}^2(k_1+j)$

$$\overline{W}(k+1) = \overline{W}(k_1) - \kappa \times N(\zeta(k)) \times \left(\frac{\alpha(k)}{B(k)}\right) \times \sigma(\upsilon(k_1)) \times \chi(k+1)$$

$$\Delta V_1(k) = V_1(k+1) - V_1(k)$$
$$= \overline{W}^T(k+1) \times \overline{W}(k+1) - \overline{W}^T(k_1) \times \overline{W}(k_1)$$
$$= \left(\overline{W}^T(k+1) - \overline{W}^T(k_1)\right)^T \times \left(\overline{W}^T(k+1) - \overline{W}^T(k_1)\right) + 2\overline{W}^T(k_1) \times \left(\overline{W}(k+1) - \overline{W}(k_1)\right)$$

$$(34)$$

From Eq. (34), we have

$$\Delta V_1(k) = \left(-\kappa N(\zeta(k))\sigma(\upsilon(k_1))\frac{\alpha(k)\chi(k+1)}{B(k)}\right)^T$$
$$\times \left(-\kappa N(\zeta(k))\sigma(\upsilon(k_1))\frac{\alpha(k)\chi(k+1)}{B(k)}\right)$$
$$+ 2\overline{W}^T(k_1)\left(-\kappa N(\zeta(k))\sigma(\upsilon(k_1))\frac{\alpha(k)\chi(k+1)}{B(k)}\right)$$

$$\Delta V_1(k) = \left(\kappa^2 N^2(\zeta(k)) \times \left\|\sigma(\upsilon(k_1))\right\|^2 \times \left(\frac{\alpha^2(k)}{B^2(k)}\right) \times \chi^2(k+1)\right)$$
$$+ \left(-2\kappa N(\zeta(k))\overline{W}^T(k_1)\sigma(\upsilon(k_1)) \times \left(\frac{\alpha^2(k)}{B^2(k)}\right) \times \chi(k+1)\right)$$

$$(35)$$

$$\overline{\varepsilon}(k+1) = \overline{\varepsilon}(k_1) - \kappa \times \left(\frac{\alpha(k)}{B(k)}\right) \times N(\zeta(k)) \times \chi(k+1)$$

$$\Delta V_2(k) = V_2(k+1) - V_2(k) = \overline{\varepsilon}^2(k+1) - \overline{\varepsilon}^2(k_1)$$

$$\Delta V_2(k) = \left(\overline{\varepsilon}(k+1) - \overline{\varepsilon}(k_1)\right)^2 + 2\overline{\varepsilon}(k_1)\left(\overline{\varepsilon}(k+1) - \overline{\varepsilon}(k_1)\right)$$

$$\Delta V_2(k) = \left(-\kappa N(\zeta(k)) \times \left(\frac{\alpha(k)}{B(k)}\right) \times \chi(k+1)\right)^2 + 2\overline{\varepsilon}(k_1)\left(-\kappa N(\zeta(k)) \times \left(\frac{\alpha(k)}{B(k)}\right) \times \chi(k+1)\right)$$

$$= \kappa^2 \frac{N^2(\zeta(k))\alpha^2(k)}{B^2(k)} \chi^2(k+1) - 2\overline{\varepsilon}(k_1)\kappa \frac{N(\zeta(k))\alpha(k)}{B(k)} \chi(k+1) \qquad (36)$$

$$\Delta V(k) = \left(\kappa^2 N^2(\zeta(k)) \left(\left\| \sigma(\upsilon(k_1)) \right\|^2 + 1 \right) \frac{\alpha^2(k)\chi^2(k+1)}{B^2(k)} \right)$$

$$- 2\kappa N(\zeta(k))\overline{W}^T(k_1)\sigma(\upsilon(k_1)) \frac{\alpha(k)\chi(k+1)}{B(k)}$$

$$- 2\overline{\varepsilon}(k_1)\kappa \frac{N(\zeta(k))\alpha(k)}{B(k)} \chi(k+1)$$

$$\Delta V(k) = \left(\kappa^2 \times \chi^2(k+1) \times N^2(\zeta(k)) \times \left(\left\| \sigma(\upsilon(k_1)) \right\|^2 + 1 \right) \times \left(\frac{\alpha^2(k)}{B^2(k)} \right) \right)$$

$$- 2\kappa N(\zeta(k)) \times \chi(k+1) \times \left(\frac{\alpha(k)}{B(k)} \right) \times \left(\pi(k) + \overline{\varepsilon}(k_1) \right) \qquad (37)$$

From Eq. (37) we get, $\varepsilon(k+1) = g\left(\overline{\zeta}_m(k_1), \upsilon^\#(k_1) \right)\left(\pi(k_1) + \overline{\varepsilon}(k_1) \right) + d_0(k_1)$ where, $k_1 = k - n + 1$ and $g\left(\overline{\zeta}_m(k_1), \upsilon^\#(k_1) \right)$ is not zero therefore multiplying both side by $g^{-1}\left(\overline{\zeta}_m(k_1), \upsilon^\#(k_1) \right)$, we get

$$\pi(k) + \overline{\varepsilon}(k_1) = \frac{\varepsilon(k+1) - \tau_0(k)}{g\left(\overline{\zeta}_m(k_1), \upsilon^\#(k_1) \right)} \qquad (38)$$

Using Eqs. (38, 37) yields

$$\Delta V(k) = \left(\kappa^2 N^2(\zeta(k)) \times \chi^2(k+1) \times \left(\left\| \sigma(\upsilon(k_1)) \right\|^2 + 1 \right) \left(\frac{\alpha^2(k)}{B^2(k)} \right) \right)$$

$$- 2\kappa N(\zeta(k)) \times \left(\frac{\alpha(k)}{B(k)} \right) \times \left(\frac{\varepsilon(k+1) - \tau_0(k)}{g\left(\overline{\zeta}_m(k_1), \upsilon^\#(k_1) \right)} \right) \times \chi(k+1)$$

Since, $\chi(k+1)1 + |N(\zeta(k))| = \kappa\varepsilon(k+1)$ we have

$$\Delta V(k) = \left(\kappa^2 N^2(\zeta(k)) \times \chi^2(k+1) \times \left(\left\| \sigma(\upsilon(k_1)) \right\|^2 + 1 \right) \times \left(\frac{\alpha^2(k)}{B^2(k)} \right) \right)$$

$$+ \frac{2N(\zeta(k))}{g\left(\overline{\zeta}_m(k_1), \upsilon^\#(k_1) \right)} \left[-\frac{\alpha(k)\chi(k+1)\chi(k+1)(1 + |N(\zeta(k))|)}{B(k)} + 2\kappa \frac{\alpha(k)\chi(k+1)\tau_0(k)}{B(k)} \right] \qquad (39)$$

By the use of Young's inequality $xy \leq \frac{1}{2}(x^2 + y^2)$, we get

$$\frac{2\kappa N(\zeta(k))\alpha(k)\chi(k+1)d_0(k)}{B(k)g\left(\overline{\zeta}_m(k_1), v^\#(k_1)\right)} \leq \frac{\kappa^2 N^2(\zeta(k))\alpha^2(k)\chi^2(k+1)}{B^2(k)} + \frac{\left(d_0(k)\right)^2}{\left(g\left(\overline{\zeta}_m(k_1), v^\#(k_1)\right)\right)^2}$$

$$\leq \frac{\kappa^2 N(\zeta(k))}{B(k)}\Delta\zeta(k) + \frac{\left(d_0(k)\right)^2}{\left(g\left(\overline{\zeta}_m(k_1), v^\#(k_1)\right)\right)^2}$$

$$\leq \kappa^2 \Delta\zeta(k) + \frac{\left(d_0(k)\right)^2}{\left(g\left(\overline{\zeta}_m(k_1), v^\#(k_1)\right)\right)^2}$$

and $\kappa^2 N^2(\zeta(k))\left(\left\|\sigma\left(v(k_1)\right)\right\|^2 + 1\right)\frac{\alpha^2(k)\chi^2(k+1)}{B^2(k)} \leq \frac{\kappa^2\alpha(k)(1+|N(\zeta(k))|)\chi^2(k+1)}{B(k)}$

$$\leq \kappa^2 \Delta\zeta(k)$$

and hence we have from Eq. (39)

$$\Delta V(k) \leq -\frac{2N(\zeta(k))}{g\left(\overline{\zeta}_m(k_1), v^\#(k_1)\right)}\Delta\zeta(k) + 2\kappa^2\Delta\zeta(k) + \frac{\left(d_0(k)\right)^2}{\left(g\left(\overline{\zeta}_m(k_1), v^\#(k_1)\right)\right)^2} \quad (40)$$

Taking, summation on of Eq. (40), we get

$$V(k) \leq -2\sum_{k'=0}^{k} \frac{N\left(\zeta(k')\right)}{g\left(\overline{\zeta}_m(k_1), v^\#(k_1)\right)}\Delta\zeta(k') + 2\kappa^2\zeta(k') + \frac{\left(d_0(k')\right)^2}{\left(g\left(\overline{\zeta}_m(k_1), v^\#(k_1)\right)\right)^2}$$

Since, $\hat{W}(k)$, $\hat{\varepsilon}(k)$, $\zeta(k)$ and $N(\zeta(k))$ are bounded and hence $1 + |N(\zeta(k))|$ is also bounded. Since, $\Delta\zeta(k) = \frac{\alpha(k)(1+|N(\zeta(k))|)\chi^2(k+1)}{B(k)} \geq 0$, and bounded which concludes that $\alpha(k) = \frac{\kappa\varepsilon(k)}{(1+|N(\zeta(k))|)}$ is bounded and hence $\varepsilon(k) = \frac{1}{\kappa}\alpha(k)(1 + |N(\zeta(k))|)$ is also bounded and since $\varepsilon(k) = y(k) - y_d(k)$ where $y_d(k)$ is bounded and hence output of the HNN $y(k)$ is also bounded.

5 Simulation Examples

In this part of paper, two examples with simulation are provided to demonstrate the validity of the designed controller.

Example 1: Consider the below mentioned stochastic nonlinear system
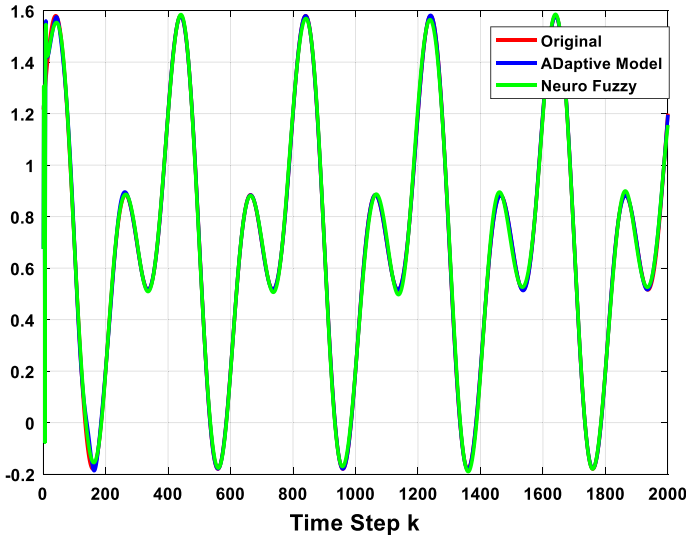
Fig. 4 Tracking of desired trajectory y_d (red line) using Hybrid NN (blue line)

$$\begin{cases} \zeta_1(k+1) = \dfrac{1.4\zeta_1^2(k)}{1+\zeta_1^2(k)} + 0.3\zeta_2(k) \\[4mm] \zeta_2(k+1) = \dfrac{\zeta_1^2(k)}{1+\zeta_1^2(k)+\zeta_1^2(k)} + \tau(k) + 0.1 \times \cos(0.05k) \times \cos(\zeta_1 k) \\[4mm] y(k) = \zeta_1(k) \end{cases}$$

where $\tau(k) = \begin{cases} 0.3\left(v^2(k)-0.5\right), & v(k) \geq 0.5 \\ 0, & -0.6 < v^2(k) < 0.5 \\ 0.2\left(v^2(k)+0.6\right), & v(k) < -0.6 \end{cases}$ and reference trajectory is given

by: $y_d(k) = 0.7 + 0.5\sin\left(kT\pi/5\right) + 0.5\cos\left(kT\pi/10\right)$ where, $T = 0.05$.

By employing the proposed HNN controller, the system is capable of successfully tracking the reference trajectory with error quickly converging to a small neighborhood of zero. The tracking performance of the controller can be seen in Figs. 4 and 5 where Fig. 4 shows the tracking performance for tracking the reference trajectory and Fig. 5 shows the tracking error. The performance of the proposed controller is compared with existing methods as shown in the Figs. 4 and 5. From Fig. 5, it can be clearly observed that the tracking error for the proposed controller is smaller than the other ones, and hence better results are obtained.

Example 2: Consider the below mentioned stochastic nonlinear system

$$\begin{cases} \zeta_1(k+1) = 0.2\zeta_1(k)cos\zeta_1(k) + 0.1\zeta_1(k)sin\zeta_1(k) + 3\zeta_2(k) \\ \zeta_2(k+1) = 0.3\zeta_2(k)\dfrac{\zeta_1(k)}{1+\zeta_1^2(k)} - 0.6\dfrac{\zeta_2^3(k)}{2+\zeta_2^2(k)} - 0.1\tau(k) \\ y(k) = \zeta_1(k) \end{cases}$$

and reference trajectory is given as $y_d(k) = 1.5sin\left(\frac{kT\pi}{5}\right) + 1.5cos\left(\frac{kT\pi}{10}\right)$

where $T = 0.05$

By choosing initial states as [1, 1], the tracking performance of the proposed controller can be seen in Figs. 6 and 7. From Fig. 6, it can be clearly observed that the controller has successfully tracked the reference trajectory with a very small error rapidly converging to zero. Fig. 7 shows the tracking error of the proposed controller in comparison with the existing controllers and it can be clearly seen that the proposed controller has achieved a better performance than the existing ones.
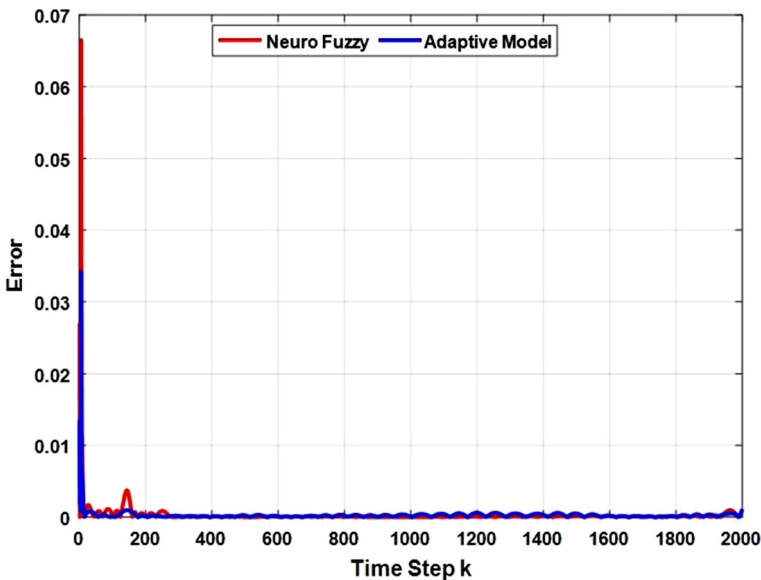

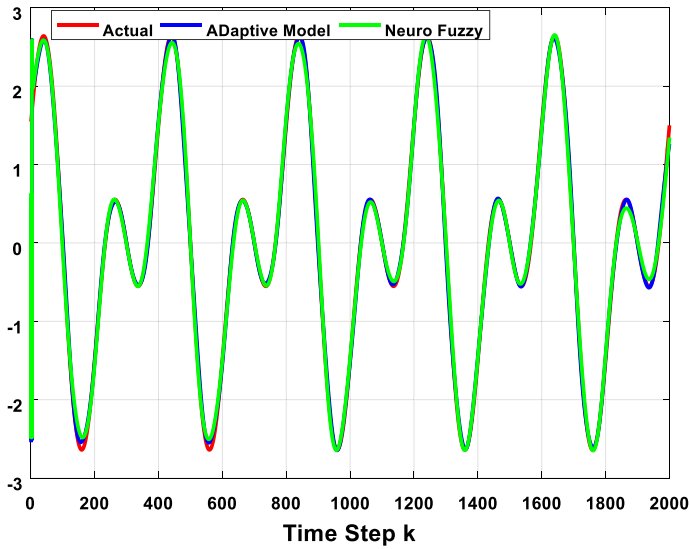**Fig. 5** Error measured by hybrid model (blue line) and Neuro-fuzzy model (red line)

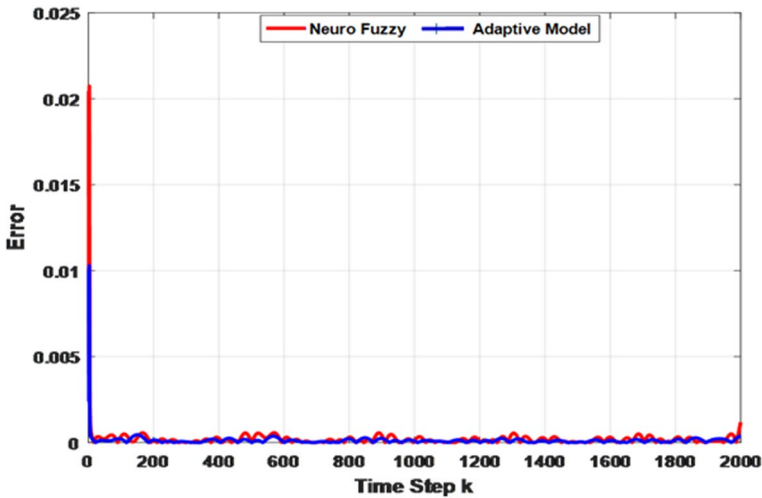**Fig. 6** Tracking of desired trajectory y_d (red line) using Hybrid NN y (blue line)

Fig. 7 Error measured by hybrid model (blue line) and Neuro-fuzzy model (red line)

6 Conclusion

In this work, an HNN is used as a controller for nonlinear discrete-time systems in presence of a nonlinear dead zone. This HNN is designed on the basis of one-step advanced state prediction of higher-order nonlinear discrete-time systems. The discrete Nussbaum function is used in order to counter the knowledge of control directions. Also, it is verified that the feedback systems possess semi-globally ultimate uniform boundedness and output

tracking errors converge in the closed neighborhood of zero. The performance of the HNN model is envisaged by the two simulation examples. Results show that the HNN control model by the use of Nussbaum gain works as predicted in the analysis.

Declarations

Conflict of interest On behalf of the all authors, We declare that we have no conflict of interest in any terms like financial support from any organizations/institutions, data, code used in this paper etc.

References

1. Lian, Y., Zhou, Y., Zhang, J., Ma, S., & Wu, S. (2022). An Intelligent Nonlinear Control Method for the Multistage Electromechanical Servo System. *Applied Sciences, 12*(10), 5053.
2. Cruz-Zavala, E., Nuño, E., & Moreno, J. A. (2021). Robust trajectory-tracking in finite-time for robot manipulators using nonlinear proportional-derivative control plus feed-forward compensation. *International Journal of Robust and Nonlinear Control, 31*(9), 3878–3907.
3. Gong, P., Yan, Z., Zhang, W., & Tang, J. (2021). Lyapunov-based model predictive control trajectory tracking for an autonomous underwater vehicle with external disturbances. *Ocean Engineering, 232*, 109010.
4. Bacciotti, A., & Rosier, L. (2005). *Liapunov functions and stability in control theory*. Heidelberg: Springer Science & Business Media.
5. Esfandiari, F., & Khalil, H. K. (1992). Output feedback stabilization of fully linearizable systems. *International Journal of control, 56*(5), 1007–1037.
6. Jiang, B., Karimi, H. R., Kao, Y., & Gao, C. (2019). Reduced-order adaptive sliding mode control for nonlinear switching semi-Markovian jump delayed systems. *Information Sciences, 477*, 334–348.
7. Lei, H., & Lin, W. (2005). Universal output feedback control of nonlinear systems with unknown growth rate. *IFAC Proceedings Volumes, 38*(1), 1073–1078.
8. Li, F., & Liu, Y. (2017). Global finite-time stabilization via time-varying output-feedback for uncertain nonlinear systems with unknown growth rate. *International Journal of Robust and Nonlinear Control, 27*(17), 4050–4070.
9. Li, Y., Li, K., & Tong, S. (2018). Finite-time adaptive fuzzy output feedback dynamic surface control for MIMO nonstrict feedback systems. *IEEE Transactions on Fuzzy Systems, 27*(1), 96–110.
10. Miao, P., Shen, Y., Li, Y., & Bao, L. (2016). Finite-time recurrent neural networks for solving nonlinear optimization problems and their application. *Neurocomputing, 177*, 120–129.
11. Singh, U. P., & Jain, S. (2016). Modified chaotic bat algorithm based counter propagation neural network for uncertain nonlinear discrete time system. *International Journal of Computational Intelligence and Applications, 15*(03), 1650016.
12. Chen, B., Liu, X., Liu, K., & Lin, C. (2009). Direct adaptive fuzzy control of nonlinear strict-feedback systems. *Automatica, 45*(6), 1530–1535.
13. Chen, B., Liu, X., Liu, K., & Lin, C. (2010). Fuzzy-approximation-based adaptive control of strict-feedback nonlinear systems with time delays. *IEEE Transactions on Fuzzy Systems, 18*(5), 883–892.
14. Singh, U. P., Jain, S., Gupta, R. K., & Tiwari, A. (2019). AFMBC for a class of nonlinear discrete-time systems with dead zone. *International Journal of Fuzzy Systems, 21*(4), 1073–1084.
15. Tong, S., Min, X., & Li, Y. (2020). Observer-based adaptive fuzzy tracking control for strict-feedback nonlinear systems with unknown control gain functions. *IEEE Transactions on Cybernetics, 50*(9), 3903–3913.
16. Das, M., & Mahanta, C. (2014). Optimal second order sliding mode control for linear uncertain systems. *ISA transactions, 53*(6), 1807–1815.

17. Shaocheng, T., Changying, L., & Yongming, L. (2009). Fuzzy adaptive observer backstepping control for MIMO nonlinear systems. *Fuzzy sets and systems, 160*(19), 2755–2775.

18. Mobayen, S., & Majd, V. J. (2012). Robust tracking control method based on composite nonlinear feedback technique for linear systems with time-varying uncertain parameters and disturbances. *Nonlinear Dynamics, 70*(1), 171–180.

19. Ho, H. F., Wong, Y. K., & Rad, A. B. (2009). Adaptive fuzzy sliding mode control with chattering elimination for nonlinear SISO systems. *Simulation Modelling Practice and Theory, 17*(7), 1199–1210.

20. Lee, H. (2010). Robust adaptive fuzzy control by backstepping for a class of MIMO nonlinear systems. *IEEE Transactions on Fuzzy Systems, 19*(2), 265–275.

21. Chen, B., Liu, X., Liu, K., & Lin, C. (2009). Novel adaptive neural control design for nonlinear MIMO time-delay systems. *Automatica, 45*(6), 1554–1560.

22. Zhou, Q., Zhao, S., Li, H., Lu, R., & Wu, C. (2018). Adaptive neural network tracking control for robotic manipulators with dead zone. *IEEE Transactions on Neural Networks and Learning Systems, 30*(12), 3611–3620.

23. Chen, M., & Tao, G. (2015). Adaptive fault-tolerant control of uncertain nonlinear large-scale systems with unknown dead zone. *IEEE Transactions on Cybernetics, 46*(8), 1851–1862.

24. Zhao, X., Shi, P., Zheng, X., & Zhang, L. (2015). Adaptive tracking control for switched stochastic nonlinear systems with unknown actuator dead-zone. *Automatica, 60*, 193–200.

25. Cao, L., Zhou, Q., Dong, G., & Li, H. (2019). Observer-based adaptive event-triggered control for nonstrict-feedback nonlinear systems with output constraint and actuator failures. *IEEE Transactions on Systems, Man, and Cybernetics: Systems, 51*(3), 1380–1391.

26. Xu, B., Sun, F., Pan, Y., & Chen, B. (2016). Disturbance observer based composite learning fuzzy control of nonlinear systems with unknown dead zone. *IEEE Transactions on Systems, Man, and Cybernetics: Systems, 47*(8), 1854–1862.

27. Chen, B., Liu, X., Liu, K., & Lin, C. (2013). Fuzzy approximation-based adaptive control of nonlinear delayed systems with unknown dead zone. *IEEE Transactions on Fuzzy Systems, 22*(2), 237–248.

28. Liu, Y. J., Tong, S., Li, D. J., & Gao, Y. (2015). Fuzzy adaptive control with state observer for a class of nonlinear discrete-time systems with input constraint. *IEEE Transactions on Fuzzy Systems, 24*(5), 1147–1158.

29. Liu, Y. J., Li, S., Tong, S., & Chen, C. P. (2018). Adaptive reinforcement learning control based on neural approximation for nonlinear discrete-time systems with unknown nonaffine dead-zone input. *IEEE transactions on neural networks and learning systems, 30*(1), 295–305.

30. Liu, Y. J., Gao, Y., Tong, S., & Chen, C. P. (2015). A unified approach to adaptive neural control for nonlinear discrete-time systems with nonlinear dead-zone input. *IEEE Transactions on Neural Networks and Learning Systems, 27*(1), 139–150.

31. Singh, U. P., & Jain, S. (2018). Optimization of neural network for nonlinear discrete time system using modified quaternion firefly algorithm: case study of Indian currency exchange rate prediction. *Soft Computing, 22*(8), 2667–2681.

32. Wei, Q., & Liu, D. (2015). Neural-network-based adaptive optimal tracking control scheme for discrete-time nonlinear systems with approximation errors. *Neurocomputing, 149*, 106–115.

33. Na, J., Lv, Y., Wu, X., Guo, Y., & Chen, Q. (2014). Approximate optimal tracking control for continuous-time unknown nonlinear systems. In *Proceedings of the 33rd chinese control conference* (pp. 8990–8995). IEEE

34. Zhou, Q., Shi, P., Lu, J., & Xu, S. (2011). Adaptive output-feedback fuzzy tracking control for a class of nonlinear systems. *IEEE Transactions on Fuzzy Systems, 19*(5), 972–982.

35. Mehraeen, S., Jagannathan, S., & Crow, M. L. (2011). Decentralized dynamic surface control of large-scale interconnected systems in strict-feedback form using neural networks with asymptotic stabilization. *IEEE Transactions on Neural Networks, 22*(11), 1709–1722.

36. Chen, W., & Li, J. (2008). Decentralized output-feedback neural control for systems with unknown interconnections. *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics), 38*(1), 258–266.

37. Lee, T. H., & Narendra, K. (1986). Stable discrete adaptive control with unknown high-frequency gain. *IEEE Transactions on Automatic Control, 31*(5), 477–479.

Rahul Kumar received M.Sc. degree in 2019 in Mathematics from Shri Mata Vaishno Devi University, Katra, India. He is currently pursuing Ph.D. Degree from Department of Mathematics at Shri Mata Vaishno Devi University Katra, Jammu and Kashmir, India. His area of interest is Nonlinear Discrete Time Systems, Control Theory and Soft Computing. He is currently working on control of nonlinear dynamical systems using soft computing approaches.

Uday Pratap Singh obtained B.Sc. Degree from Dr. R.M.L. Awadh University, Ayodhya, (U.P.), India, M.Sc. Degree from Indian Institute of Technology, Guwahati, India, and received Ph.D. in Computer Science from Barkatullah University, Bhopal. He is currently working as an Associate Professor in the School of Mathematics at Shri Mata Vaishno Devi University Katra, Jammu and Kashmir, India. His area of research work is Nonlinear Systems, Soft Computing, and Theoretical Computer Science. He authored many research papers, published at reputed journals and conferences. He has guided 03 Ph.D. Students and has undertaken a number of Research and Development projects sponsored by the GoI. He is a life member of IAENG, and Computer Society of India.

Arun Bali received M.Sc. degree in 2017 in Mathematics from Shri Mata Vaishno Devi University, Katra, India. He is currently pursuing Ph.D. Degree from Department of Mathematics at Shri MataVaishno Devi University Katra, Jammu and Kashmir, India. His area of interest is Adaptive Control Systems and Soft Computing. He is currently working on adaptive control theory of nonlinear dynamical systems.

Kuldip Raj , born in Jammu, in 1968. He received the master's and PhD degree in Mathematics from Jammu University, Jammu, India. He has over 20 years' experience in teaching and research. He has the credit of making significant contribution to research and development in the area of Functional Analysis, Opearator Theory and Sequence Spaces etc. He authored many research papers, published at reputed journals and conferences. He has guided 07 Ph.D. Scholars and has undertaken a number of major Research and Development projects sponsored by the GoI. He is life member of IMS, JMS etc.