Designing Flexible Neuro-Fuzzy System Based on Sliding Mode Controller for Magnetic Levitation Systems

Zahra Mohammadi[1], Mohammad Teshnehlab[2] and Mahdi Aliyari Shoorehdeli [3]

[1] **Electrical Engineering Department, Science and Research Branch of Islamic Azad University
Tehran, Iran**

[2] **Faculty of Electrical Engineering, Department of Control Engineering, K. N. Toosi University of Technology,
Tehran, Iran**

[3] **Faculty of Electrical Engineering, Department of Mechatronics Engineering, K. N. Toosi University of Technology,
Tehran, Iran**

Abstract

This study presents a novel controller of magnetic levitation system by using new neuro-fuzzy structures which called flexible neuro-fuzzy systems. In this type of controller we use sliding mode control with neuro-fuzzy to eliminate the Jacobian of plant. At first, we control magnetic levitation system with Mamdani-type neuro-fuzzy systems and logical-type neuro-fuzzy systems separately and then we use two types of flexible neuro-fuzzy systems as controllers. Basic flexible OR-type neuro-fuzzy inference system and basic compromise AND-type neuro-fuzzy inference system are two new flexible neuro-fuzzy controllers which structure of fuzzy inference system (Mamdani or logical) is determined in the learning process. We can investigate with these two types of controllers which of the Mamdani or logical type systems has better performance for control of this plant. Finally we compare performance of these controllers with sliding mode controller and RBF sliding mode controller.

Keywords: *Flexible neuro-fuzzy inference systems, Sliding mode, Mamdani approach, Logical approach and Magnetic levitation system.*

1. Introduction

The objective of this study is to keep a metal ball suspended in mid-air via magnetic suspension. This system is magnetic levitation which has great practical importance in many engineering fields and industrial systems. Introducing nonlinearity and instability of magnetic levitation systems (MLS) make them quite complex to control. So we proposed a hybrid controller which composed of a flexible neuro-fuzzy inference system (FLEXNFIS) based on sliding mode to overcome the difficulties of magnetic levitation system.

In recent years, various control strategies have been proposed in the literatures for MLS, Such as: feedback linearization technique [1, 2] and sliding mode control [3]. The input-output, input-state, and exact linearization techniques have been used [4, 5]. Intelligent control such as neural network techniques [6] and fuzzy control design [7] has also been used to control magnetic levitation systems.

On the other hand, various neuro-fuzzy structures have been proposed so far in literatures [8-12]. They combine the natural language description of fuzzy systems and learning properties of neural networks for different applications. Most of neuro-fuzzy structures can be divided into two approaches based on the connection between the antecedents and consequents in the individual rules [13]. The first approach is Mamdani type reasoning that consequents and antecedents are connected by a t-norm, e.g. min or product operator. The second is logical type approach that consequents and antecedents are connected by fuzzy implication, e.g. an S-implication (see, e.g. [14, 15]).

The idea of flexible neuro-fuzzy inference systems (FLEXNFIS) has been developed by Leszek Rutkowski and Krzysztof Cpalka [16-20]. FLEXNFIS is a type of new developed neuro-fuzzy systems. The connectives in the structure of such systems are flexible that is a major improvement in importance. It combines the logical approach and Mamdani type reasoning to construct a neuro-fuzzy system which exhibit simultaneous appearance of Mamdani and logical type inferences. The basic compromise AND-type neuro-fuzzy inference system is an example of such systems that we use them to control the MLS. Another important quality of FLEXNFIS is the automatic determination of fuzzy inference (Mamdani or logical) in the process of learning. The basic OR-type neuro-fuzzy inference system is another example of such systems that type of the fuzzy inference system is determined at the end of learning process and we use it to control the MLS. Learning algorithm is gradient optimization with constraint that learns the extra parameters applied in the structure of systems.

We use from sliding mode control strategies [21] to eliminate the Jacobian of plant. First of all we control magnetic levitation system with Mamdani and logical type neuro-fuzzy based on sliding mode separately and then two types of FLEXNFIS are used to control the MLS.

The aim of this study is designing FLEXNFIS as a novel controller for MLS and researching its particulars. Then we compare performance of this controller with Mamdani and logical neuro-fuzzy systems and also investigate by using OR-type neuro-fuzzy inference system which of the Mamdani reasoning or logical approaches can be a better controller for MLS. Besides, we compare the performance of FLEXNFIS with sliding mode controller and RBF sliding mode controller [21]. Simulation results introduce FLEXNFIS comparatively powerful, feasible and effective for MLS.

2. Dynamics of Magnetic Levitation System

The dynamic equations of magnetic levitation system can be written as [2]:

$$\frac{dp}{dt} = v$$
$$m\frac{dv}{dt} = mg_c - C\left(\frac{i}{p}\right)^2$$
$$Ri + \frac{d(L(p)i)}{dt} = e \qquad (1)$$

Where, p denotes the position of ball, v is the ball's velocity, R is the coil's resistance, i is the current through the electromagnet, e is the applied voltage, m is the mass of the levitated object, g_c denotes the gravity and C is the magnetic force constant. L is the coil's inductance that is a nonlinear function of ball's position (p) and L_1 is a parameter of system. It can be written as follow:

$$L(p) = L_1 + \frac{2C}{p} \qquad (2)$$

We chose the states and control input as: $x_1 = p$, $x_2 =$ v, $x_3=i$ and $u=e$. Thus, the state space equations of magnetic levitation system are as follows:

$$\dot{x}_1 = x_2$$
$$\dot{x}_2 = g_c - \frac{C}{m}\left(\frac{x_3}{x_1}\right)^2$$
$$\dot{x}_3 = \frac{-R}{L}x_3 + \frac{2C}{L}\left(\frac{x_2 x_3}{x_1^2}\right) + \frac{u}{L} \qquad (3)$$

We use sliding mode control [21, 22] which is defined for structures with uncertainties. This technique tries to control system by using a sliding surface definition based on state-space.

Magnetic ball should be settled in the desired distance, x_{1d} and only the vertical motion p is considered. The output error is defined as:

$$e_1 = x_1 - x_{1d} \qquad (4)$$

In this study the sliding surface on the phase plane can be defined as:

$$S = \left(\frac{d}{dt} + \lambda\right)^{n-1} e_1 \qquad (5)$$

In 3^{th} -order systems, n=3:

$$S = \left(\frac{d}{dt} + \lambda\right)^2 e_1 = \ddot{e}_1 + \lambda_1 \dot{e}_1 + \lambda_2 e_1$$
$$S = g_c - \frac{C}{m}\left(\frac{x_3}{x_1}\right)^2 + \lambda_1 x_2 + \lambda_2 (x_1 - x_{1d}) \qquad (6)$$

In this way, the switching surface $\dot{S}$ will be defined as:

$$\dot{S} = \dddot{x}_1 + \lambda_1 \ddot{x}_1 + \lambda_2 \dot{x}_1 \qquad (7)$$

$$\dddot{x}_1 = \frac{\partial \dot{x}_2}{\partial t} = \frac{2C}{m}\left((1-\frac{2C}{Lx_1})\frac{x_2 x_3^2}{x_1^3} + \frac{R}{L}\left(\frac{x_3}{x_1}\right)^2\right) - \frac{2Cx_3}{Lmx_1^2}u \qquad (8)$$

where, $\dot{x}_1 = x_2$ and $\ddot{x}_1 = \dot{x}_2$. These equations will be used in section 4.3 and are very important for Adaptive law. Derivation of Lyapunov function is used to learn the neuro-fuzzy system based on sliding mode. The description of our generalized learning method will be discussed in section 4.3.

The overall block diagram of the system under control is shown in Fig. 1. The neuro-fuzzy systems will be two types of FLEXNFIS, Mamdani and logical type.

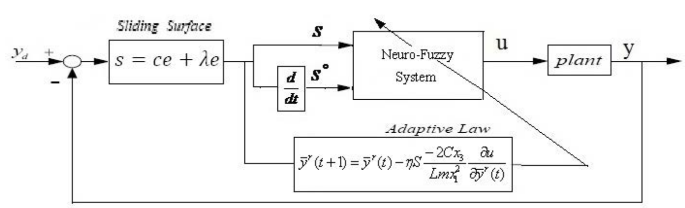

Fig. 1 The overall block diagram of system.

3. General Architecture of Neuro-Fuzzy Inference System

This section defines a neuro-fuzzy system, which can presents two approaches of Mamdani-type and logical-type and calls neuro-fuzzy inference system (NFIS). It is, from the topology point of view, a fuzzy inference system uses learning techniques that is similar to standard back propagation in feed forward networks. So, we review fuzzy inference structure of NFIS briefly [20].

In this study the fuzzy system of NFIS is like a multi-input single-output mapping $U \rightarrow V$, which $U \subset R^n$ is the input space and $V \subset R$ is output space. The canonical form of fuzzy rule base can be defined as:

$Ru^{(l)}$: If x_1 is $A_1{}^l$ and x_2 is $A_2{}^l$ and ... x_n is $A_n{}^l$ then y is B^l. (9)

where $\boldsymbol{x}=[x_1,\ldots, x_n] \in U$, $y \in V$. membership functions of fuzzy sets $A_1^l, A_2^l, \ldots, A_n^l$ are determined by $\mu_{A_i^l}(x_i)$ that $i=1,\ldots,n$ are the number of inputs, $l=1,\ldots,M$ are the number of rules and membership functions of B^l are determined by $\mu_B{}^l(y)$. The firing strength of rules is defined by:

$$\tau_l(\mathbf{x}) = \overset{n}{\underset{i=1}{T}}\{\mu_{A_i^l}(\mathbf{x}_i)\} \tag{10}$$

Each of M rules in fuzzy rule base determines a fuzzy set $\overline{B}^l \subset U$ given by compositional rule of inference:

$$\overline{B}^l = A' \circ (\mathbf{A}^l \rightarrow B^l) \tag{11}$$

where $\mathbf{A}^l = A_1^l \times A_2^l \times \ldots \times A_n^l$. Membership functions of $\overline{B}^l$ can be determined by sup-star composition:

$$\mu_{\overline{B}^l}(y) = \sup_{x \in U}\{T\{\mu_{A'}(\mathbf{x}), \mu_{A_1^l \times .. \times A_n^l \rightarrow B^l}(\mathbf{x}, y)\}\} \tag{12}$$

that T comes from class of t-norms. So, for a singleton fuzzifier, i.e., a crisp input $x \in U$, formula (12) will be defined as:

$$\mu_{\overline{B}^l}(y) = \mu_{A_1^l \times .. \times A_n^l \rightarrow B^l}(\mathbf{x}, y) = I(\mu_{A^l}(\mathbf{x}), \mu_{B^l}(y)) \tag{13}$$

I(.) can be a t-norm (engineering implication [23]) in Mamdani approach or fuzzy implication [14, 15] in logical approach. So, we can write generally:

$$I(\mu_{A^l}(x), \mu_{B^l}(y)) = \begin{cases} I_{eng}(\mu_{A^l}(x), \mu_{B^l}(y)) \\ for \quad mamdani \quad approach \\ I_{fuzzy}(\mu_{A^l}(x), \mu_{B^l}(y)) \\ for \quad \mathbf{log}ical \quad approach \end{cases} \tag{14}$$

Output of the fuzzy inference engine is the fuzzy set B' that is aggregation of M individual fuzzy sets $\overline{B}^l \subset U$. Type of the aggregation operator is different in two approaches too. In Mamdani approach, the aggregation can be any operator in class of s-norms and in logical approach it can be any operator in class of t-norms:

Mamdani approach uses s-norms:

$$B' = \bigcup_{l=1}^{M} \overline{B}^l$$

$$\mu_{B'}(y) = \overset{M}{\underset{l=1}{S}} \mu_{\overline{B}^l}(y) \tag{15}$$

Logical approach uses t-norms:

$$B' = \bigcap_{l=1}^{M} \overline{B}^l$$

$$\mu_{B'}(y) = \overset{M}{\underset{l=1}{T}}\{\mu_{\overline{B}^l}(y)\} \tag{16}$$

The defuzzification technique is the centre of area (COA) [18] to define the output. So the discrete form of output is defined by:

$$\boldsymbol{y} = \frac{\sum_{r=1}^{M} \bar{\boldsymbol{y}}^r . \mu_{B'}(\bar{\boldsymbol{y}}^r)}{\sum_{r=1}^{M} \mu_{B'}(\bar{\boldsymbol{y}}^r)} \tag{17}$$

where $\bar{y}^r$ are the canters of membership functions $\mu_B{}^r(y)$ and $r=1,\ldots,M$.

$$\mu_{B'}(\bar{y}^r) = \max_{y \in V}\{\mu_{B'}(y)\} \tag{18}$$

There are two different models of NFIS with different definitions for implication operators Eq. (14) and aggregation operators (t-norms and s-norms). Consequently we will have two models for Mamdani and logical approaches (for details see e.g. [18-20]):

Mamdani neuro-fuzzy system:

$$y = \frac{\sum_{r=1}^{M} \bar{y}^r \overset{M}{\underset{l=1}{S}}\{I_{eng}\{\overset{n}{\underset{i=1}{T}}\{\mu_{A_i^l}(x_i)\}, \mu_{B^l}(\bar{y}^r)\}\}}{\sum_{r=1}^{M} \overset{M}{\underset{l=1}{S}}\{I_{eng}\{\overset{n}{\underset{i=1}{T}}\{\mu_{A_i^l}(x_i)\}, \mu_{B^l}(\bar{y}^r)\}\}} \tag{19}$$

Logical neuro-fuzzy system:

$$\boldsymbol{y} = \frac{\sum_{r=1}^{M} \bar{\boldsymbol{y}}^r \overset{M}{\underset{l=1}{T}}\{I_{fuzzy}\{\overset{n}{\underset{i=1}{T}}\{\mu_{A_i^l}(\mathbf{x}_i)\}, \mu_{B^l}(\bar{\mathbf{y}}^r)\}\}}{\sum_{r=1}^{M} \overset{M}{\underset{l=1}{T}}\{I_{fuzzy}\{\overset{n}{\underset{i=1}{T}}\{\mu_{A_i^l}(\mathbf{x}_i)\}, \mu_{B^l}(\bar{\mathbf{y}}^r)\}\}} \tag{20}$$

A generalized architecture of NFIS that supports both Mamdani and logical approaches is proposed by Rutkowski [18-20] and described by:

$$y = f(x) = \frac{\sum_{r=1}^{M} \bar{y}^r . agr_r(x, \bar{y}^r)}{\sum_{r=1}^{M} agr_r(x, \bar{y}^r)} \quad (21)$$

4. Designing of Flexible NFIS

Eq. (21) in previous section apply in both flexible and nonflexible systems with different definitions for $agr_r(x,\bar{y}^r)$, $I_{l,r}(x,\bar{y}^r)$ and $\tau_l(x)$ [18, 20]. In nonflexible systems $agr_r(x,\bar{y}^r)$, $I_{l,r}(x,\bar{y}^r)$ and $\tau_l(x)$ are defined with traditional triangular norms [24-26] or fuzzy implications. But in flexible systems we use some important definitions to construct $agr_r(x,\bar{y}^r)$, $I_{l,r}(x,\bar{y}^r)$ and $\tau_l(x)$ in Eq. (21) (for details see e.g. [18, 20]). These definitions are called adjustable triangular norms that contain compromise operator, H-function and adjustable quasi-implication which play an important role in construction of flexible NFIS [18, 20]. In the next section we define two types of flexible NFIS using these definitions.

4.1. Basic flexible OR-type NFIS

The basic neuro-fuzzy system of an OR-type is given as follows [20]:

$$\tau_1(x) = H\begin{pmatrix} \mu_{A_1^l}(x_1),\ldots,\mu_{A_n^l}(x_n) \\ v = 0 \end{pmatrix} \quad (22)$$

$$I_{l,r}(x, \bar{y}^r) = H\begin{pmatrix} \widetilde{N}_{1-v}(\tau_l(x)), \mu_{B^l}(\bar{y}^r); \\ v \end{pmatrix} \quad (23)$$

$$agr_r(x, \bar{y}^r) = H\begin{pmatrix} I_{1,r}(x, \bar{y}^r),\ldots, I_{M,r}(x, \bar{y}^r) \\ 1-v \end{pmatrix} \quad (24)$$

Parameter v is the type of Basic flexible OR-type NFIS which described by Eq. (22)-(24). According to (22)-(24) equations system is Mamdani-type for v=0. Its behaviour is "more Mamdani" for $v \in (0,0.5)$, undetermined for v=0.5 and "more logical" for $v \in (0.5,1)$. It is logical-type for v=1.
It should be noticed that parameter v can be learned and consequently, type of the system can be determined in the process of learning.

4.2. Basic compromise AND-type NFIS

Flexible compromise AND-type NFIS presents a combination of two basic fuzzy inference systems (Mamdani and logic) and uses "engineering implication" [23] and fuzzy implications [14] together [20]. The firing strength of rules is defined similar to Eq. (22). The implication and aggregation operators are defined as:

$$I_{l,r}(x, \bar{y}^r) = \begin{pmatrix} (1-\lambda)H\begin{pmatrix} \widetilde{N}_1(\tau_l(x)), \mu_{B^l}(\bar{y}^r); \\ v = 0 \end{pmatrix} + \\ + \lambda H\begin{pmatrix} \widetilde{N}_0(\tau_l(x)), \mu_{B^l}(\bar{y}^r); \\ v = 1 \end{pmatrix} \end{pmatrix} \quad (25)$$

$$agr_r(x, \bar{y}^r) = \begin{pmatrix} (1-\lambda)H\begin{pmatrix} I_{1,r}(x, \bar{y}^r),\ldots, I_{M,r}(x, \bar{y}^r) \\ 1 \end{pmatrix} \\ + \lambda H\begin{pmatrix} I_{1,r}(x, \bar{y}^r),\ldots, I_{M,r}(x, \bar{y}^r) \\ 0 \end{pmatrix} \end{pmatrix} \quad (26)$$

In this structure we learn compromise parameter λ. The value of it shows which of the fuzzy inference systems (Mamdani or logic) is more dominant in the learning process.

4.3. Learning procedure

Adaptive neuro-fuzzy controller needs plant's Jacobian. To solve the problem of FLEXNFIS on the subject of plant's Jacobian, we use sliding mode controllers [21]. Block diagram of controller is shown in Fig. 1. So learning process of FLEXNFIS is based on derivation of Lyapunov function and called generalized learning. Based on the Lyapunov theorem, the sliding surface reaching condition is:

$$V = \frac{1}{2}S^2 \Rightarrow \dot{V} = S\dot{S} < 0 \quad (27)$$

The steepest descent rule is used to minimize the value of $S\dot{S} < 0$ by respecting to input-output membership function parameters. $\bar{y}^r$ are centers of output membership functions that $r = 1,\ldots M$ and are trained by iterative procedure:

$$\bar{y}^r(t+1) = \bar{y}^r(t) - \eta \frac{\partial S\dot{S}}{\partial \bar{y}^r(t)} = \bar{y}^r(t) - \eta S \frac{\partial \dot{S}}{\partial \bar{y}^r(t)} \quad (28)$$

$$\frac{\partial \dot{S}}{\partial \bar{y}^r(t)} = \frac{\partial \dot{S}}{\partial u} \times \frac{\partial u}{\partial \bar{y}^r(t)} \quad (29)$$

By using Eq. (3), (7), (8) we can say:

$$\frac{\partial \dot{S}}{\partial u}=\frac{\partial \ddot{x}_1}{\partial u}=\frac{-2Cx_3}{Lmx_1^2} \tag{30}$$

So, the recursive equation can be defined as:

$$\bar{y}^r(t+1)=\bar{y}^r(t)-\eta S\frac{-2Cx_3}{Lmx_1^2}\frac{\partial u}{\partial \bar{y}^r(t)} \tag{31}$$

To calculate $\frac{\partial u}{\partial \bar{y}^r}$, we recall layered architecture of NFIS [20] and apply the back propagation method to train the system. By the use of chain rule we can say:

$$\frac{\partial u}{\partial \bar{y}^r}=\frac{\partial u}{\partial \bar{y}^r}+\sum_{l=1}^{M}\frac{\partial u}{\partial agr_r(x,\bar{y}^r)}\times\frac{\partial agr_r(x,\bar{y}^r)}{\partial \bar{y}^r} \tag{32}$$

On the other hand:

$$\frac{\partial agr_r(x,\bar{y}^r)}{\partial \bar{y}^r}=\frac{\partial agr_r(x,\bar{y}^r)}{\partial I_{l,r}}\times\frac{\partial I_{l,r}}{\partial \mu_{B^l}(\bar{y}^r)}\times\frac{\partial \mu_{B^l}(\bar{y}^r)}{\partial \bar{y}^r} \tag{33}$$

Two other important parameters that we update are ν and λ. We apply the gradient optimization with constraint to optimize the ν parameter, which determines type of the system for OR-type NFIS, and also compromise parameter λ for AND-type NFIS.

Learning process of ν parameter for OR-type NFIS is given by:

$$\nu(t+1)=\nu(t)-\eta S\frac{\partial \dot{S}}{\partial \nu(t)} \tag{34}$$

$$\frac{\partial \dot{S}}{\partial \nu}=\frac{\partial \dot{S}}{\partial u}\times\frac{\partial u}{\partial \nu}=\frac{-2Cx_3}{Lmx_1^2}\times\frac{\partial u}{\partial \nu} \tag{35}$$

According to (22)-(24) equations we will say:

$$\begin{aligned}\frac{\partial u}{\partial \nu}&=\sum_{l=1}^{M}\sum_{r=1}^{M}\frac{\partial u}{\partial agr_r(x,\bar{y}^r)}\times\frac{\partial agr_r(x,\bar{y}^r)}{\partial I_{l,r}(x,\bar{y}^r)}\times\\&\left(\frac{\partial I_{l,r}(x,\bar{y}^r)}{\partial \nu}+\left(\frac{\partial I_{l,r}(x,\bar{y}^r)}{\partial \tilde{N}_{1-\nu}(\tau_l(x))}\right)\frac{\partial \tilde{N}_{1-\nu}(\tau_l(x))}{\partial(1-\nu)}\frac{\partial N(\nu)}{\partial \nu}\right)\\&+\sum_{l=1}^{M}\frac{\partial u}{\partial agr_r(x,\bar{y}^r)}\times\frac{\partial agr_r(x,\bar{y}^r)}{\partial(1-\nu)}\times\frac{\partial N(\nu)}{\partial \nu}\end{aligned} \tag{36}$$

Learning process of λ parameter for AND-type NFIS is given by:

$$\lambda(t+1)=\lambda(t)-\eta S\frac{\partial \dot{S}}{\partial \lambda(t)} \tag{37}$$

$$\frac{\partial \dot{S}}{\partial \lambda}=\frac{\partial \dot{S}}{\partial u}\times\frac{\partial u}{\partial \lambda}=\frac{-2Cx_3}{Lmx_1^2}\times\frac{\partial u}{\partial \lambda} \tag{38}$$

According to Equations (25), (26) we can say:

$$\begin{aligned}\frac{\partial u}{\partial \lambda}&=\sum_{l=1}^{M}\sum_{r=1}^{M}\frac{\partial u}{\partial agr_r(x,\bar{y}^r)}\times\frac{\partial agr_r(x,\bar{y}^r)}{\partial I_{l,r}(x,\bar{y}^r)}\times\frac{\partial I_{l,r}(x,\bar{y}^r)}{\partial \lambda}\\&+\sum_{r=1}^{M}\frac{\partial u}{\partial agr_r(x,\bar{y}^r)}\times\frac{\partial agr_r(x,\bar{y}^r)}{\partial \lambda}\end{aligned} \tag{39}$$

4.4. Designing Parallel OR-type FLEXNFIS Based Controller Design

In basic flexible OR-type NFIS, parameter ν represents the type of system (Mamdani or logical) in the process of learning. The best value of parameter ν in the process of learning is equal to zero or one. If parameter $\nu=0$, the fuzzy inference system will be Mamdani reasoning and for $\nu=1$ the fuzzy inference system desires to logical type.

We use this property to construct a n ew structure for control of magnetic levitation system. Because updating the parameter ν requires large computations which is time consuming. The overall block diagram of the system under control is shown in Fig. 2. Mamdani is an OR-type FLEXNFIS with $\nu=0$ and logic is an OR-type FLEXNFIS with $\nu=1$. U_m and U_L are the outputs of Mamdani and logic respectively that multiple in the proper coefficients Q_m and Q_L that are selected respect to better performance. From Fig. 2, the input to the MLS (plant) is given by $u=Q_mU_m+Q_LU_L$.

The neuro-fuzzy systems are trained on-line during the control process to give the controller ability of adapting with the changes. In this structure we don't need to learn the parameter ν and we learn centers of the output membership functions $\mu_{B^r}(y)$. The adaptive law for Mamdani and logic are different as shown in Fig. 2 and they use Eq. (32) for learning process separately. The plant's Jacobian is eliminated in both of the controllers.

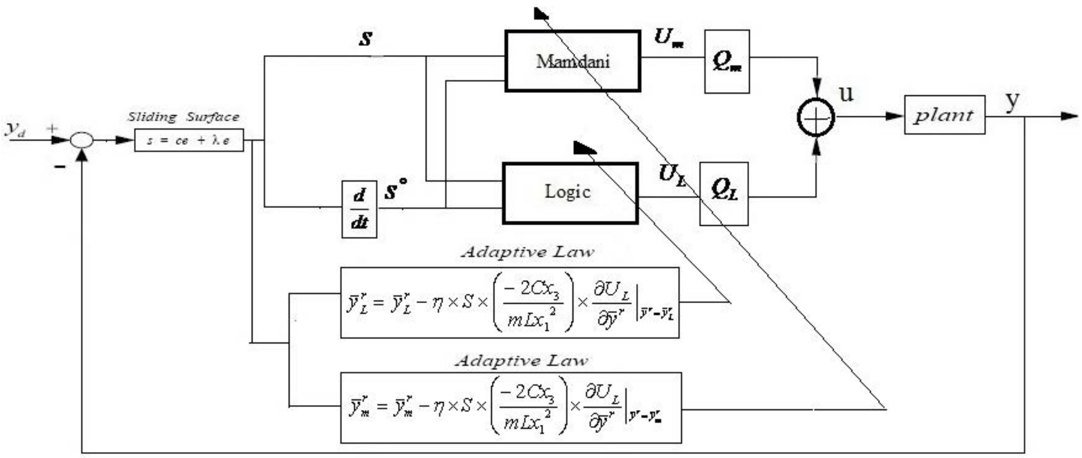

Fig. 2 The overall block diagram of Parallel OR-type FLEXNFIS.

5. Experiments and Results

In this section, simulation results show the position versus time (millisecond) and the signal control (the applied voltage) versus time for system. After that we will discuss about results.

5.1. Experiment Setup

We can divide our simulation into five categories. In all experiments there are two inputs for controller: switching surface (S) and change of switching surface ($\dot{S}$). U is the output of controller which is control variable of the MLS or applied voltage. We define two membership functions type Gaussian for each of inputs and four membership functions for output. So, the sliding mode inference rules are designed as:

If S is A_1^1 and $\dot{S}$ is A_2^1 then u is B_1

If S is A_1^1 and $\dot{S}$ is A_2^2 then u is B_2

If S is A_1^2 and $\dot{S}$ is A_2^1 then u is B_3

If S is A_1^2 and $\dot{S}$ is A_2^2 then u is B_4

The parameters of the MLS are as follows [2, 21]. The coil's resistance R=28.7Ω, mass of the ball m=11.87g, the gravity g_c=9.81$msec^{-2}$, the magnetic force constant C=1.24×10^{-4}, the inductance L_1=0.65H, and x_{1d}=0.01 is the desired value of x_1. The parameters of sliding surface are chosen such that λ_1 in sliding surface is set as 61, λ_2=930.

- In the first experiment, we learn the parameters of the membership functions of the Mamdani-type system. Eq. (19) describes the appearance of such systems. We apply the product operator for engineering implication or t-norm [23]. The simulation results are shown in Fig. 3.
- In the second experiment we learn the parameters of the membership functions of the logical-type system. Eq. (20) determines this type of neuro-fuzzy system. We apply the Reichenbach operator for fuzzy implication [14, 15 and 20]. We also use product operator for aggregation. Fig. 1 shows the overall block diagram of control system. The simulation results are represented in Fig. 4.
- In the third experiment, we design Basic flexible OR-type NFIS, and learn the parameters of the membership functions and parameter $\nu \in [0,1]$. The learning parameters formulas are described by Eq. (28), (34). In this experiment adjustable Quasi-Implication is applied to define implication operator. H-function is generated by the product t-norm [18, 20]. The simulation results are depicted in Fig. 5.

We considered a constraint for parameter ν in Eq. (23), (24) to satisfy the range $0 \leq \nu \leq 1$. (For details see e.g. [18]). The learning of parameter ν is replaced by:

$$f_z(\nu) = \frac{1}{1+\exp(-(p_1\nu - p_2))} \tag{40}$$

We suppose that p_1=5, p_2=2.5 and the initial value of ν=0.5. The results are depicted in Fig. 6 which shows the learning of function with Eq. (40) versus time. As figure shows, the neuro-fuzzy system with Eq. (21) becomes Mamdani-type for ν=0 at the end of learning process.

- In the fourth experiment, we design basic compromise AND-type NFIS described by Eq. (25), (26) as controller. We learn the parameters of the membership functions and the parameter $\lambda\epsilon[0,1]$. Learning of the parameter λ was described by Eq. (37). The implication and aggregation operators are generated similar to previous experiment. The simulation results are shown in Fig. 7. We also consider a constraint for parameter λ similar to Eq. (40). Learning of this function versus time is described in Fig. 8.

- In the final experiment, we construct the Parallel OR-type FLEXNFIS based controller design and learn the parameters of the membership functions of Mamdani and logic systems separately. According to Fig. 2 Q_m and Q_L, coefficients of Mamdani and logical systems respectively, are obtained considering better performance. Q_m is equal to 0.9 and Q_L is equal to 0.1. These coefficients show that Mamdani has better performance compare with logical system. Fig. 9 represents the simulation results.

Finally we compare our results with sliding mode controller (without chattering) [3] and RBF sliding mode controller which are applied for control of MLS [21]. Fig. 10 and Fig. 11 show the output and signal control of these controller. In the next section we'll compare and analyze these Figures.

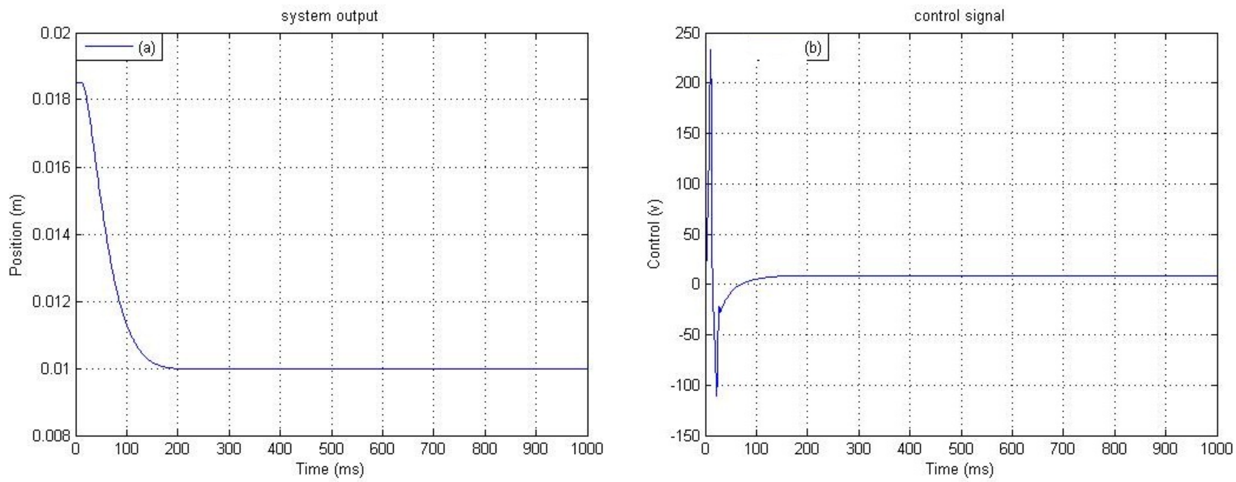

Fig. 3 Mamdani neuro-fuzzy controller based on sliding mode (a) position of ball (b) control signal

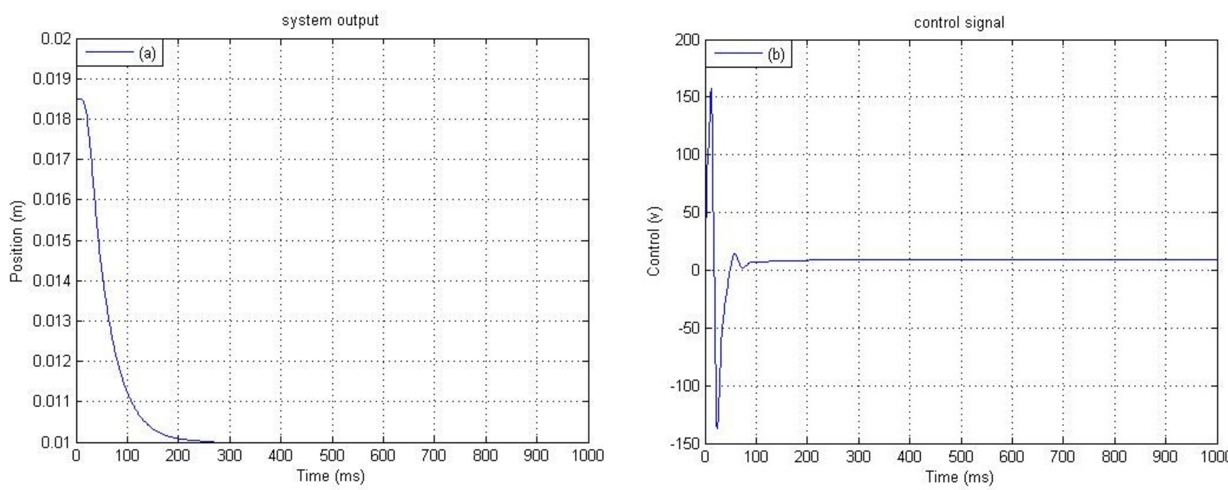

Fig. 4 Logical-type neuro-fuzzy controller based on sliding mode (a) position of ball (b) control signal

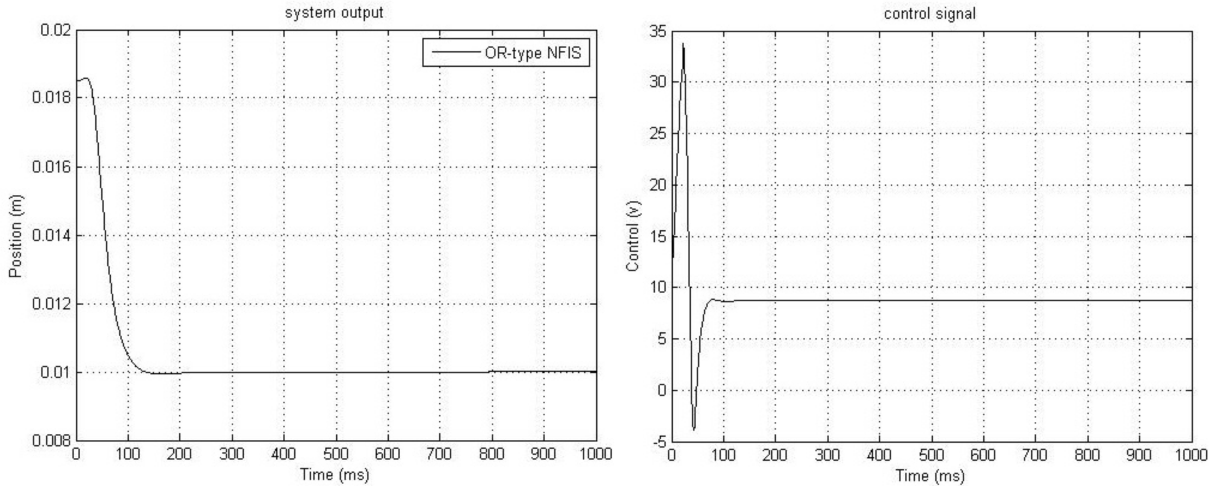

Fig. 5 Basic flexible OR-type NFIS based on sliding mode (a) position of ball (b) control signal

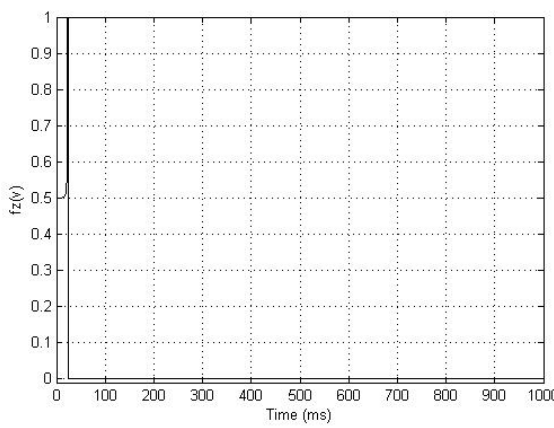

Fig. 6 Depiction of the learning of parameter v

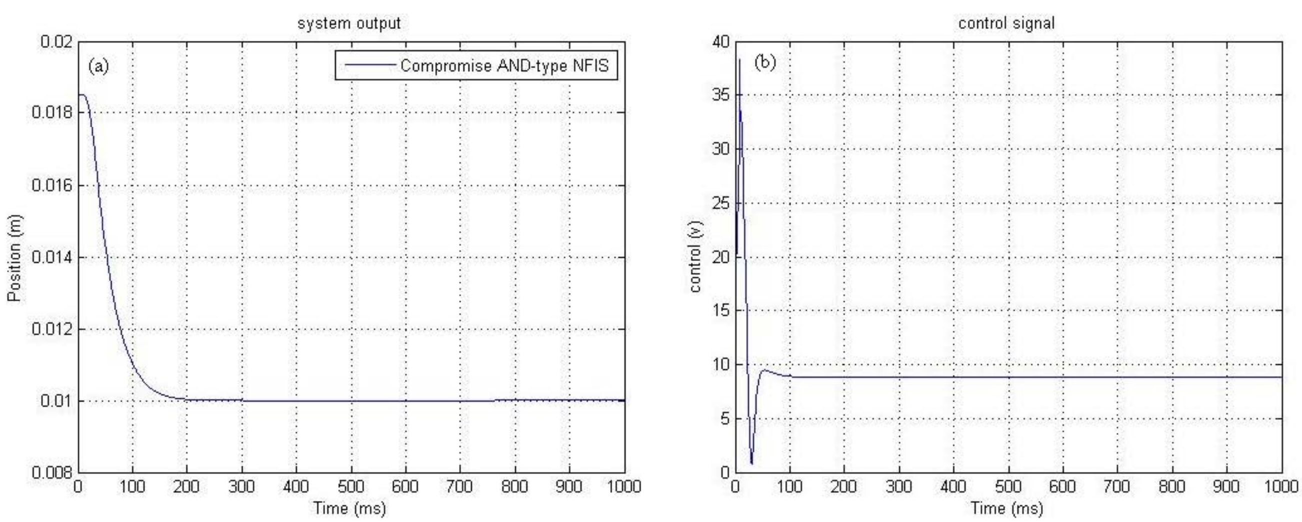

Fig. 7 Basic compromise AND-type NFIS based on sliding mode (a) position of ball (b) control signal

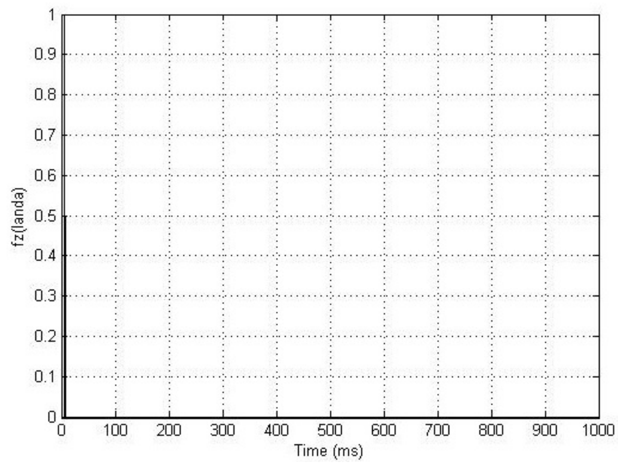

Fig. 8 Description of the learning of parameter λ

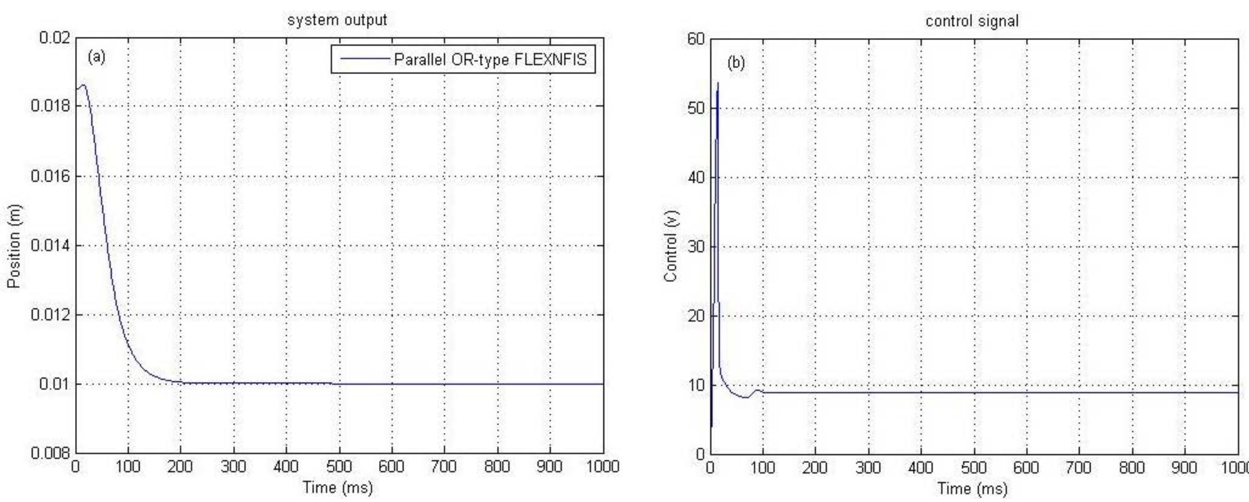

Fig. 9 Parallel OR-type FLEXNFIS Based sliding mode (a) position of ball (b) control signal

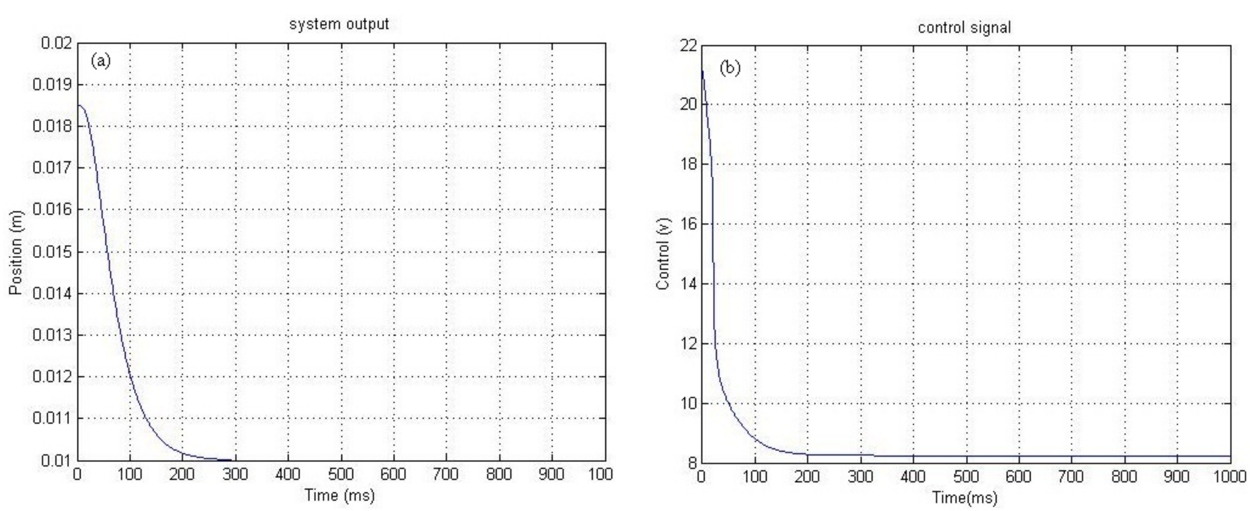

Fig. 10 Sliding mode without chattering (a) Position of ball and (b) control signal [3]

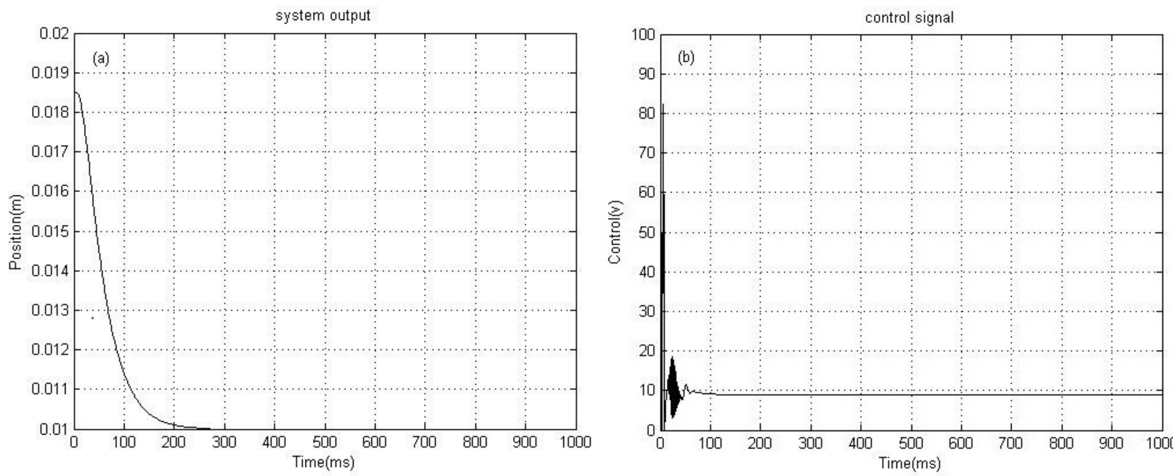

Fig. 11 RBF-Sliding mode controller (a) Position of ball and (b) control signal [21]

5.2. Analysis and Comparison of All Simulation Results

In this section we comprise the results of simulations considering cost function and settling time (ts). The first criterion is minimum control effort [27] which depends on control signal (applied voltage). To minimum the control signal we consider this cost function:

$$J = \int_0^t |u(t)| dt \tag{41}$$

The other criterion is tracking problem:

$$IAE = \int_0^t |x_1(t) - r(t)| dt \tag{42}$$

The final performance index is summation of above criterions and settling time considering some coefficients (K_1=45, K_2=6.25e-004, K_3=10) to coordinate these criterions.

$$PE = K_1 t_s + K_2 \int_0^t |u(t)| dt + K_3 \int_0^t |x_1(t) - r(t)| dt \tag{43}$$

Table 1: Comparison Results Summary

Neuro-fuzzy Controller Based-on Sliding Mode	Settling Time	$J = \int_0^\infty \lvert u(t) \rvert dt$	IAE	PE
Mamdani-type	0.128 sec	1.0603e+004	0.5398	17.74
Logical-type	0.135 sec	1.1563e+004	0.5235	18.5371
Basic flexible OR-type NFIS	0.100 sec	9.1253e+003	0.5051	15.25
Parallel OR-type FLEXNFIS	0.125 sec	9.1282e+003	0.5506	16.83
Basic compromise AND-type NFIS	0.123 sec	9.0844e+003	0.5014	16.2269
Sliding mode control [3]	0.159 sec	8.6118e+003	0.6420	18.9576
RBF sliding mode control [21]	0.141 sec	9.1146e+003	0.5352	17.3937

According to table 1 and Fig. 3 to Fig. 11 we can say:

- By respect to PE in table 1, Mamdani method as a controller is better than the logical method for magnetic levitation system. On the other hand, Fig. 3 and Fig. 4 show the voltage peaks of these two controllers are very high so these two controllers can't be proper for MLS in practice applications.

- As we said, Flexible OR-type NFIS is a system that type of fuzzy inference (Mamdani or logic) is determined in learning process [20]. So parameter ν determines the type of the system. As Fig. 6 shows, parameter ν has reached to zero at the end of learning process. It means that type of fuzzy inference system is Mamdani. In this way flexible OR-type NFIS shows if we control MLS with Mamdani neuro-fuzzy system given by Eq. (19) we will have better results as compare with logical method with Eq. (20) (see, e.g. [18]).
- Fig. 6 and Fig. 8 show that parameter ν and λ at the end of learning process takes the value of zero and doesn't take a value between (0, 1). Because in learning process the best value of them are determined (zero or one) and for these values system is well defined.
- With due attention to table 1 and Fig. 9, Parallel OR-type FLEXNFIS has reasonable results knowing that we don't learn the parameter ν. But because of better results in table 1 for flexible OR-type NFIS it is better to learn the parameter ν in spite of the more computations.
- Basic compromise AND-type NFIS has proper results comparing with Mamdani and logical neuro-fuzzy controllers separately. Fig. 8 shows that the value of parameter λ at the end of the learning is equal to zero and it means that Mamdani fuzzy model is dominated in system. It represents the same result as OR-type NFIS.
- As table 1 show, flexible neuro-fuzzy systems have better performance index among of the other controllers. Besides, among of flexible systems, OR-type NFIS has the best performance.
- Finally we compare our results with RBF-Sliding mode controller. The control signal in Fig. 11 (b) has chattering and is not proper for voltage source. So it should be modified. But control signals of figures 5, 7 and 9 (flexible systems) are suitable and have no chattering.

6. Conclusions

This paper introduced structure of flexible neuro-fuzzy systems briefly and proposed two new structures of flexible neuro-fuzzy systems as novel controllers for magnetic levitation system.
In this research basic OR-type NFIS and basic compromise AND-type NFIS was designed based on sliding mode for stabilization and control the magnetic levitation system to the desired point in the state space. At last Parallel OR-type FLEXNFIS was proposed to overcome the computations of learning the parameter ν. Simulation results indicated flexible systems based controller design worked well when applied to the magnetic levitation and had high performance. They also showed Mamdani-type neuro-fuzzy systems can be a better controller for magnetic levitation system as compared with logical type for magnetic levitation system.

References

[1] T. Mizutani, H. Katayama, A. Ichikawa, "Tracking Control of a Magnetic Levitation System by Feedback Linearization", SICE Annual Conference in Sapporo, August 6th, 2004. Hokkaido Institude of Technology, Japan.
[2] W. Barie and J.Chiasson, "Linear and nonlinear state-space controllers for magnetic levitation", International Journal of Systems Science 27 (1996, No)11, 1153-1163.
[3] N. F. AL-Muthairi and M. ZriBi, "Sliding mode control of a magnetic levitation system", Received 18 October 2003 and in revised from 23 February 2004. Hindawi Publishing Corporation Math Problem Eng, 2: 93-107.
[4] F. Zhang and K. Suyama, "Nonlinear feedback control of magnetic levitation system by exact linearization approach", Proc. 4th IEEE Conferences on Control Application, (New York), 1995, pp. 267-268.
[5] A. Charara, J. De Miras, and B. Carson, "Nonlinear Control of a Magnetic Levitation System without Paramagnetization", IEEE Transactions on Control Systems Technology 4 (1996, No) 5, 513-523.
[6] M. Liari and G. Bloch, "A Neural Network with Minimal Structure for Maglev System Modeling and Control", IEEE International Symposium on Intelligent Control/Intelligent Systems and Semiotics, (Massachusetts), 1999, pp. 40-45.
[7] C-A. Dragos, S. Pretil, R-E. Precup, R-G. Bulzan, C. Pozna and J.K. Tar, "Takagi-Sugeno fuzzy controller for a magnetic levitation system laboratory equipment", IEEE International joint Conference on Computational Cybernetics and Technical Informatics, (ICCC-CONTI 2010). Timisora, Romania.
[8] E. Czogala, J. Leski, "Fuzzy and Neuro-Fuzzy Intelligent Systems, Physica Verlag", A Springer Verlag Company, Heidelberg, New York, 2000.
[9] M. Gorzalczany, "Computational Intelligence Systems and Applications: Neuro-Fuzzy and Fuzzy Neural Synergisms", Springer Verlag, 2002.
[10] D. Nauck and R. Kruse, "Neuro-fuzzy systems for function approximation, Fuzzy Sets and Systems", Vol. 101, 1999, pp. 261-271.
[11] J. S. Wang and C. S. G. Lee, "Self Adaptive Neuro-fuzzy inference Systems for Classification Applications", IEEE Trans on Fuzzy Systems, 2002, Vol. 10, pp. 790-802.
[12] R. R. Yager and D. P. Filev, Essentials of fuzzy modelling and control, John Wiley and sons, 1994.
[13] K. Cpalka, L. Rutkowski, "A Method for Designing Flexible Neuro-fuzzy Systems", Springer Verlag, Berlin Heidelberg, LNAI 4029, 2006, pp. 212-219.
[14] J. C. Fodor, "On fuzzy implication operators, fuzzy sets and systems", Vol. 42, 1991, pp. 293-300.
[15] E. P. Klement, R. Mesiar and E. Pap, Triangular Norms, Kluwer Academic Publishers, Netherlands, 2000.

[16] K. Cpalka, L. Rutkowski, “Neuro-Fuzzy Structures for Pattern Classification”, WSEAS Trans. on Computers, Issue 7, Vol. 4, 2005, pp. 697-688.
[17] K. Cpalka, L. Rutkowski, “Flexible Takagi-Sugeno Neuro-Fuzzy Structures for Nonlinear Approximation”, WSEAS Trans. on Computers, Issue 9, Vol. 4, 2005, pp. 1450-1458.
[18] L. Rutkowski and K. Cpalka, “Designing and learning of adjustable quasi triangular norms with applications to neuro-fuzzy systems”, WSEAS Trans. on Computers, Issue 9, Vol. 4, 2005, pp. 1450-1458.
[19] L. Rutkowski, K. Cpalka, “Flexible neuro-fuzzy systems”, IEEE Transactions on Neural Networks, 2003, pp. 554-574.
[20] L. Rutkowski, Flexible Neuro-Fuzzy Systems (structures, Learning and Performance Evaluation), eBook ISBN: 1-4020-8043-3, Kluwer Academic Publishers, New York Boston London, 2004.
[21] M. Aliasghary, M. Teshnehlab, A. Jalilvand, M. Aliyari Shoorehdeli, “Hybrid Control of Magnetic Levitation System Based-on New Intelligent Sliding Mode control”, Journal of Applied Science 8(14): ISSN 1812-5654, 2008, 2561-2568.
[22] J. J. E Slotin and W. Li, Applied nonlinear control, 1st Edn. Prentice Hall, New York, 1991.
[23] J. M. Mendel, “Fuzzy logic systems for engineering: a tutorial”, Proc. Of the IEEE, March 1995, 83(3): 345-377.
[24] E. P. Klement, R. Mesiar and E. Pap, Triangular Norms, Kluwer Academic Publishers, Netherlands, 2000.
[25] T. Calvo, G. Mayor and R. Mesiar, Aggregation Operators, New Trans and Applications, Physica Verlag, New York, 2002.
[26] O. Cordon, F. Herrera and A. Perigrin 1997, “Applicability of Fuzzy Operators in the Design of Fuzzy Logic Controllers”, Fuzzy Sets and Systems, Vol. 86, No. 1, 1997, pp. 15-41.
[27] Donald E. Kirk, Optimal control theory: an introduction, 0-486-43484-2 (pbk), Originally Published, Englewood Cliffs, NJ. : Prentice Hall, 1937.

Zahra Mohammadi received the M.Sc. degree in Electrical Engineering from Science and Research Branch of Islamic Azad University, Tehran, Iran, in 2011; she is currently working toward the Ph.D. degree. Her current research interest includes Fuzzy, Neural Network and Soft Computing.

Mohammad Teshnehlab is professor at Department of Control, Faculty of Electrical Engineering, K.N.T.U University. His current research interests include Fuzzy, Neural Network, Soft Computing, Evolutionary Filtering, and Simultaneous Localization and Mapping.

Mahdi Aliyari Shoorehdeli is assistant professor at Department of Mechatronics Control, Faculty of Electrical Engineering K.N.T.U University. His research interest includes Mechatronics Systems, Fault Diagnosis, Neural Networks, Fuzzy systems, Nero-Fuzzy control, Identification and Predication, Pattern Recognition and Swarm Intelligence.