INTELLIGENT CONTROL OF MAGNETIC SUSPENSION SYSTEMS

Gamal Abd-El-Raheem ; Abd-El-fattah Mahmoud ;
Amer Abd-El-fattah
Electrical Engineering Department, Faculty of Engineering, Assiut University, Assiut, Egypt.

Hosam Ismail Hasan
Petroleum Pipelines Company, Egypt.

(Received January 28, 2006 Accepted April 8, 2006)

This paper proposes the application of fuzzy and neuro-fuzzy techniques to the control of an open loop unstable magnetic suspension system using genetic algorithms (GA's) as a tuning algorithm. A Mamdani type fuzzy PID controller is investigated. The proposed controller employs a two-stage control structure, where the fuzzy PI controller reasonably aims to suppressing steady state errors and the fuzzy PD controller provides stabilizing action. The membership functions' parameters for both controllers are determined by the GA's. A Takagi-Sugeno-Kang (TSK) controller using parallel distributed compensation (PDC) approach is presented. The proposed controller, which is nonlinear, is fuzzy blending of individual piecewise linear controllers designed using pole placement technique. A neuro-fuzzy controller is proposed. Reinforcement learning along with GA's was used to train the neuro-fuzzy controller.

KEYWORDS: *Fuzzy control, Two-stage fuzzy PID controller, Parallel distributed compensation, Genetic algorithms, neuro-fuzzy.*

1. INTRODUCTION

Magnetic suspension techniques are being utilized in various fields such as, industrial, transportation and aerospace projects. Magnetic bearings, which are the most important industrial application of magnetic suspension, are used in applications where ordinary bearings meet difficulties. Contact free bearings are an attractive technology for special applications which imply high speed rotation, minimum friction or clean environments such turbo compressors and high speed milling spindles. However, the design of a control system for magnetic suspension systems is difficult and constitutes a challenging task due to the highly nonlinear and inherently open-loop unstable electromagnetic dynamics in the relationship between the lift force and the air gap distance. Until now in industrial applications, magnetic bearings' control has been realized with classic (PID) or modern (Q-parameterization, H_∞) controllers [1, 2]. Such linear control techniques depend on linearizing the dynamics of the magnetic suspension system about a nominal operating point. These control approaches provide very good design solutions by linear terms only in a limited small region of the nominal design condition. To avoid such deficiency, most industrial magnetic suspension systems, which adopt conventional linear control techniques, use only small portion of the available air gap during operation. However, in some industrial applications, it is desirable to use a larger portion of the available air gap. Nonlinear control techniques, such as sliding mode, are quite complicated compared to linear controller design. It takes great numerical effort to solve nonlinear equations and such controllers are sensitive to modeling errors. Even when a system can be described exactly it is not always possible to find a nonlinear solution so that the closed loop system is stable. On the other hand, fuzzy logic has been considered as a potentially useful means of designing controllers for nonlinear systems, which shown better performance and easier implementation in controlling these nonlinear systems than conventional methods [3].

This paper presents three architectures for nonlinear fuzzy controller. The first architecture is a Mamdani type fuzzy PID controller. Similar in construction to the conventional PID controllers, there have been several types of fuzzy PID controllers proposed for different situations. A two-stage fuzzy PID controller, which is composed of fuzzy PD and fuzzy PI controllers in parallel, is investigated in this paper. The fuzzy PD section presents the derivative action (D) that provides stabilizing action which is essential in the case of open loop unstable system as in magnetic suspension to achieve stability. And the fuzzy PI section presents the integral action (I) which aims to suppressing steady state errors. However, one of the early steps of designing a fuzzy controller is to represent system knowledge in terms of fuzzy linguistic terms and hence assigning membership functions to system variables. In most cases, the designer has to make two design decisions; the first is on the rule base of the fuzzy system. In most applications, this problem is solved by exploiting the knowledge of an expert and forming the rule-base by a trial-and-error approach which depends on the simplicity of the system or by using another optimization algorithm, like GA's, to form the rule base [4]. For the two-stage fuzzy PID controller proposed here, the separation between the two sections of the controller makes it much easier interpret the rules as it makes it very logical. Secondly, the geometrical forms of the membership functions such as triangular, trapezoidal, etc. and the parameters of those membership functions. The problem of tuning the membership functions has attracted so much attention [5].

The integration of fuzzy logic control (FLC) and genetic algorithms (GA's) has generated a lot of interest since it is widely believed that the two methods complement each other. GA's are used to optimize the design of FLCs, whereas the well established area of fuzzy control has provided a main stream application area for the GA's. GA's are general purpose optimization algorithms with a probabilistic component that provides a mean to search poorly understood and irregular spaces. GA's can be used in various ways to optimize a fuzzy controller depending on which design parameters are to be optimized. They can be used to tune the membership functions' parameters. GA's also can be applied to tune the scaling factors of the controller [6]. Tuning the rule-base is another important application of GA's in the design of fuzzy controllers. Finally, some or all the design parameters can be tuned simultaneously by the GA's. In this paper GA-based reinforcement learning scheme along with an external reinforcement signal from the controlled system has been used to tune the membership functions' parameters in the fuzzy PID controller. The external reinforcement signal will be available at the end of an evaluation cycle to play the role of the fitness function, and the reciprocal of the performance index (ITAE) has been used as the fitness function.

The second architecture is a Takagi-Sugeno-Kang (TSK) controller using parallel distributed compensation (PDC) approach. This controller achieves nonlinearity through piece-wise linear approximation. The main idea of this controller design is to divide the operating air gap into a limited number of local piecewise linear models and derive a control rule for each model by linear control techniques, feedback control via pole placement was used here, and fuzzy logic is used to smoothly schedule each control rule over the entire air gap so as to compensate for operating point dependence nonlinearity.

Neuro-fuzzy systems combine the learning features of neural networks with the linguistic rule interpretation of fuzzy inference systems. Among several existing neuro-fuzzy systems, few have been widely used: ANFIS (Adaptive Network based Fuzzy Inference System) [7], NEFCON (NEural Fuzzy CONtroller) [8, 9], this one well suited to control applications, and NN-FLC (Neural Network based Fuzzy Logic Controller) [10]. This paper employs a hybrid neuro-fuzzy-genetic system based on a modified version of NN-FLC model as a basis and genetic algorithm for reinforcement learning. The GA's learning starts with the definition of the chromosome's representation, which is formed by the parameters of each membership function of the input and output variables. The membership functions' parameters are adjusted so that a suitable performance may be obtained. The fitness function of each individual is calculated as the reciprocal of the performance index (ITAE).

2. MAGNETIC SUSPENSION SYSTEM

The single axis magnetic suspension system, which components are shown in **Figure 1**, consists of a ball bearing of mass (*m*) placed underneath an electromagnet at distance (*x*). When a current flows through the electromagnetic coil, it will generate an attractive electromagnetic force in the vertical direction. The difference between the electromagnetic force and the gravitational force will induce an up or down motion on the ball bearing.

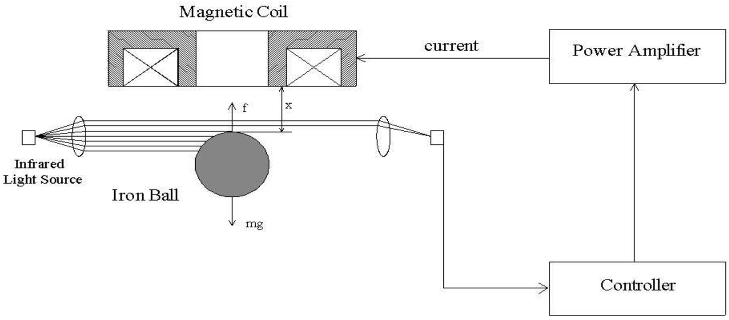


Figure 1: Magnetic Suspension System.

Equations of Motion

The magnetic circuit, as shown in **Figure 1**, consists of the ball bearing and the electromagnet. The electromagnet coil has N turns, across which there is an instantaneous voltage *e* and through which a current *i* is flowing. A decrease of *dx* in the air gap length gives an instantaneous increase in energy input of *eidt* and stored magnetic energy of ($0.5\, i^2 dL$) where *L* is the inductance of the coil. If *f* is the force at the air gap faces, then the mechanical work exerted on the ball is (*-fdx*).

The differential equation describing the motion of the mass *m* is;

$$m\frac{d^2x}{dt^2} = mg - f \tag{1}$$

where $$f = c\left(\frac{i}{x}\right)^2 \tag{2}$$

and *x* is measured from the magnetic ball face [1]. The other equation of motion is obtained from the voltage balance in the electromagnetic circuit. The total voltage is balanced by the voltage drop due to the resistance and the inductive reactance of the coil. Therefore,

$$e = Ri + \frac{d(Li)}{dt} \tag{3}$$

where

g	acceleration of gravity.
f	attractive force of the electromagnet.
x	distance from the electromagnet.
e	voltage across the coil.
i	coil current.
R	coil resistance.
L	coil inductance.

Since is always kept around the close neighborhood of $x_\circ$, *L* can be considered as a constant. This will greatly simplify the system model without losing much accuracy. Therefore, equation (3) can be rewritten as;

$$e = Ri + L\frac{di}{dt} \tag{4}$$

Equations (1, 4) represent the differential equations that describe the dynamics of the magnetic suspension system [1]. A Matlab/Simulink magnetic suspension nonlinear model based on HUMUSOFT CE 152 Magnetic Suspension System was used in this paper.

3. MAMDANI TYPE FUZZY PID CONTROLLER

Fuzzy controllers can be classified according to the control action and the controller architecture into three categories, fuzzy PD, fuzzy PI and fuzzy PID. The realization of fuzzy PID controller is usually proposed in the form of three inputs controller. The inputs are usually $e, \sum e, \Delta e$ or $e, \Delta e, \Delta_2 e$. This implementation is difficult for parameters' tuning and in most cases it will need optimization for the rule base, besides it loses the physical point of view. Rather, let us take in account that the proportional action P is the natural control action, the derivative action D provides stabilizing action and integral I aims to suppressing steady state errors. Still, it is necessary to stress the fact that the I-portion of the controller in itself causes instability and slows the

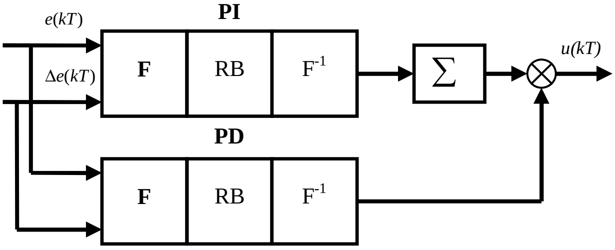


Figure 2: Structure of a fuzzy PID controller.

controlled process down, while the D-portion speeds the controlling action up. Thus the classical PID controller compared to the PI one accelerates the transient response. Knowing this information, we would rather build the fuzzy PID controller in such a way, in which we could separately influence the changes of P, I and D components in some natural way. That is why we propose to build the fuzzy PID controller as a parallel combination of two fuzzy controllers: I and PD or PI and PD which can be defined as;

$$u(kT) = u_{PI}((k-1)T) + \Delta u_{PI}(kT) + u_{PD}(kT) \tag{5}$$

where $$\Delta u_{PI}(kT) = k_{PI} e(kT) + k_I \Delta e(kT) \tag{6}$$

and $$u_{PD}(kT) = k_{PD} e(kT) + k_D \Delta e(kT) \tag{7}$$

see [11]. The structure of this controller is shown in **figure 2**. Besides the simplicity of this architecture, it will keep the most important feature of the fuzzy control which the informal nature of the control design process.

A. Controller Description

As illustrated previously, the proposed controller consists of two parallel sections PD and PI. First inputs to the controller will be the error signal;

$$e(kT) = x_d(kT) - x(kT) \tag{8}$$

where $x_d(kT)$ is the desired position and $x(kT)$ is the sphere position at time kT. The second input is the derivative of the error which will be defined as;

$$ce(kT) = (e(kT) - e((k-1)T))/T \tag{9}$$

The controller output is the sum of $u_{PD}(kT)$, the PD section output, and the integration of $du_{PI}(kT)$, the PI section output. And will be written as;

$$u(kT) = u_{PD}(kT) + u_{PI}((k-1)T) + du_{PI}(kT) \tag{10}$$

Two sets of fuzzy rules are to be used to map the error $e(kT)$ and rate of change of the error $ce(kT)$ into changes in the controller action with both its section, i.e. changes in $u_{PD}(kT)$ and $du_{PI}(kT)$. These fuzzy rules will be in the form of;

If $e(kT)$ is LP and $ce(kT)$ is ZE then u_{PD} is LP

where LP, ZE are short notations for large positive and zero respectively. **Table 1** illustrates the rule base used for the PD section. This rule base consists of 49 rules. **Table 2** illustrates the rule base for the PI section. This rule base consists of 5 rules.

Table 1: FIS rule base for the PD section.

e *ce*	**LP**	**MP**	**SP**	**ZE**	**SN**	**MN**	**LN**
LP	LP	LP	LP	LP	MP	MP	SP
MP	LP	LP	MP	SP	ZE	SN	MN
SP	MP	MP	SP	SP	ZE	SN	SN
ZE	MP	MP	SP	ZE	SN	MN	MN
SN	SP	SP	ZE	SN	SN	MN	MN
MN	MP	SP	ZE	SN	MN	MN	MN
LN	SN	MN	MN	LN	LN	LN	LN

Table 2: FIS rule base for the PI section.

e *ce*	**MP**	**SP**	**ZE**	**SN**	**MN**
ZE	LP	MP	ZE	MN	LN

For fuzzy conjunction, implication, aggregation and defuzzification certain design decisions have been made to achieve the best possible performance. The *min* operator was used as a fuzzy conjunction, the *clipping* method for fuzzy implication, the *max* operator for fuzzy aggregation and the *centroid* method for defuzzification.

The next step in the controller design is to tune the membership function so as to achieve our certain goals; in this paper the GA's were used to accomplish this task. The following sections describe in details the steps for using GA's to optimize a fuzzy control system.

B. Genetic Algorithms

Genetic algorithms are probabilistic optimization routines modeled after Darwinian's theory of natural evolution. Genetic algorithms emulate nature performing genetic operations over a finite set (population) of potential solutions (individuals). Genetic algorithms operate by function evaluation as compared to other optimization routines based on gradient evaluation. As a result, GA's have shown the ability to optimize systems with multi-modal objective functions and with nonlinear parameters. Here, the GA's are used to tune the membership functions' parameters for both the input and output spaces by using the reinforcement signal from the model as the fitness function, while in other researches GA's have been used to tune the rule base itself [12]. Roughly speaking, GA’s manipulate strings of binary digits, 1's and 0's, called chromosomes which represent multiple points in the search space through proper encoding mechanism. GA’s carry out simulated evolution on populations of such chromosomes. Bit string encoding is the most common encoding technique used by GA's researchers because of its ease of creating and manipulating. However, with a binary string encoding the resulting search time is much longer than that of using real-value string encoding used here [13]. **Figure 3** illustrates the behavior of standard genetic algorithm during one tuning cycle. Next, we will briefly describe the basic operators in a GA.

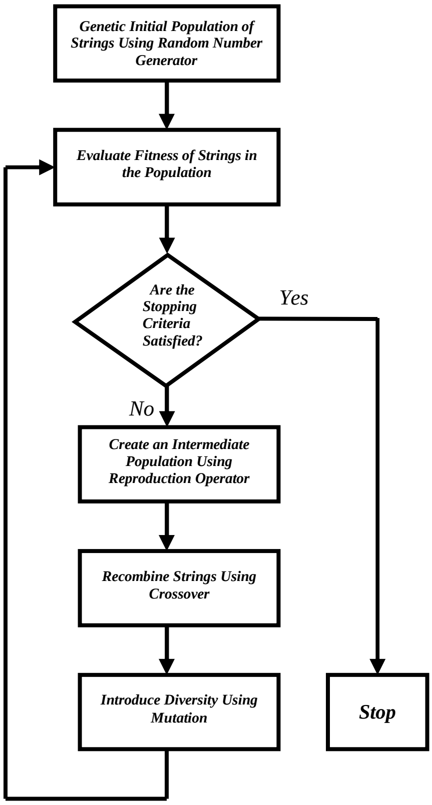

Genetic Initial Population of Strings Using Random Number Generator
Evaluate Fitness of Strings in the Population
Are the Stopping Criteria Satisfied?
Yes
No
Create an Intermediate Population Using Reproduction Operator
Recombine Strings Using Crossover
Introduce Diversity Using Mutation
Stop

Figure 3: The Standard Genetic Algorithm.

Mutation is the random alteration of one or more genes with a probability P_m. In a real value string, mutation is the random selection and alteration of *i*th gene by the addition of random value. Mutation is usually considered the innovative genetic operator which allows GA's to search new patterns. Mutation probability is nominally small within 0.0001 < P_m< 1. In some applications, mutation is set to a larger value at the beginning (to explore new patterns) and is reduced later (to help reach a converging solution). And even in more complex studies, different strategies have been used to specify mutation rates during the learning process [14]. As the mutation rate increases, the algorithm behaves more like a random search routine.

Reproduction is the act of generating exact replicas of an individual I of a given population and placing the replicas among an intermediate population according to a certain algorithm. Following the theme of survival of the fittest, the highest fit individual is assigned higher probability P_{ri} to reproduce, where

$$P_{ri} = \frac{f_i}{\sum_{i=1}^{n} f_i} \tag{11}$$

and n is the size of the population. In some elitist strategies, the highest fit set of individuals in the population is directly placed in the next population, bypassing the intermediate population. Roulette wheel is one of the most popular election methods, and it will be used in this dissertation.

Crossover is the recombination of strings among two randomly selected individuals in the intermediate population to create new offspring from the next generation. Every pair of individuals will result in two individuals in the next population. The crossover rate p_c determines where crossovers occur and is usually much higher than p_m, i.e. $0.1 < p_c < 0.9$ as it encourages recombination among already found patterns. Crossover can occur in one or more points.

Interpretation Function is the transformation function between the representation (genotype) space and the evaluation (phenotype) space. Most of the current researches, as in this paper, concentrate on optimizing parameters of the domain knowledge. The domain knowledge consists of membership function and rule base. Membership function parameters include the general membership function shape (Triangular, Trapezoidal, Sigmoidal, Gaussian, etc.) and the defining points (Center, Max Right, Min Left, etc.). Rule base parameters like fuzzy associative memory, fuzzy operators like disjunctive (OR) and conjunctive (AND).

In this paper only defining point of the membership functions are chosen to be optimized. This was a design decision which was made considering the fact that in control of most physical systems, including the magnetic suspension, rules can often be derived either intuitively as in our system or through operator experience as in process control systems, and for this controller the two parallel sections made it much easier to derive rules for each section separately.

Fitness Function Design is a very important aspect of design of GA's since GA's depend on fitness function to guide the direction of its search. Obviously, a fitness function needs to include all the pertinent parameters which need to be minimized/maximized. Due to the similarities which GA's share with nature, the fitness function is often designed such that the more desirable solutions correspond to higher fitness. So, the GA's optimization is usually regarded as a search for parameters which maximize the fitness function. In this paper the reciprocal of the performance index (*ITAE*), which is described as;

$$ITAE = \int_0^t \left| e(t) \right| t \, dt \tag{12}$$

will be used as the fitness function. Where (t) is a period of time during the simulation is run with the current string parameters' values are applied to the fuzzy controller. An additional bonus will be added to the fitness value in case the controller could stabilize the sphere till the end of the test period (t), this helps in distinguishing desirable and undesirable strings (specially at the beginning of the tuning process). The value of the ITAE is not to be available for calculations except after 0.2 sec. to avoid the divide by zero error.

C. Optimization of Fuzzy Systems Using GA

Table 3: Parameterized Membership Functions.

Membershi p Function	Parameters		
LN	-	b_0	c_0
MN	a_1	b_1	c_1
SN	a_2	a_2	c_2
ZE	a_3	b_3	c_3
SP	a_4	b_4	c_4
MP	a_5	b_5	c_5
LP	a_6	b_6	-

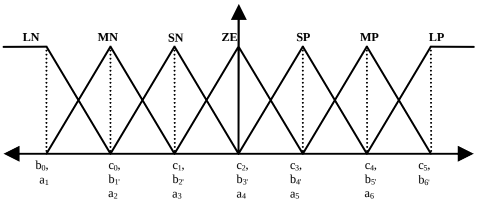


Figure 4: Representation of a Fuzzy Set with 7 Membership Functions.

Each of the fuzzy variables is represented by a fuzzy set. Fuzzy sets consist of a number of membership functions which are often can be represented by mathematical functions. Triangular membership function has been widely used in fuzzy control problems and will be used in this paper. **Figure 4** illustrates the representation of a fuzzy set with 7 membership functions. **Table 3** shows the membership function representation of one fuzzy variable with parameterized membership functions. Each membership function is represented by three parameters (*a*, *b*, and *c*). Triangular membership function is specified by three parameters as follows:

$$triangular(x:a,b,c)=\begin{cases} 0 & x<a \\ (x-a)/(b-a) & a\le x\le b \\ (c-x)/(c-b) & b\le x\le c \\ 0 & x>c \end{cases} \tag{13}$$

In the same manner a fuzzy set with *n* membership functions can be represented by (3*n* – 2) parametric genes, where the parameter (*a*) of the leftmost membership function and the parameter (*c*) of the rightmost membership function are constants with large negative and positive values respectively. The chromosome, which represents the potential solution in the problem space will consist of the parametric representations of the fuzzy sets specified for each input and output variable.

Membership function integrity is very important for the GA's, and it must be checked and enforced at the beginning of each cycle. There are two types of integrity errors, out of boundary errors and non-definition errors. Such errors violate the two basic membership functions' principles. First, each membership function overlaps only with the closest neighboring membership function. Secondly, for every input variable, any possible input value must have a membership value more than zero.

These two kinds of errors could occur either in the initial population, during the crossover process, or during the mutation process. Membership function integrity must be checked in every stage. The integrity requirements for the triangular membership function are the following:

1. $a_i < b_i < c_i$

where i is the membership function number in the fuzzy set, and a ,b , and c are the triangular membership function's parameters.

2. For each membership function (i) in the fuzzy set where i-2 > 0 and i+2 < n-1:
   $c_{i\text{-}2} < a_i$ and $c_i < a_{i+2}$
3. $a_0 < a_1 < \ldots.. < a_{\text{n-1}}$
4. $b_0 < b_1 < \ldots.. < b_{n\text{-}1}$
5. $c_0 < c_1 < \ldots.. < c_{n\text{-}1}$
6. Lower limit of the fuzzy set < b_0.
7. Upper limit of the fuzzy set > $b_{n\text{-}1}$.

where n is the number of the membership functions in the fuzzy set, and where fuzzy membership function number (0) is the leftmost membership function and membership function number (n) is the rightmost membership function.

Parameters encoding is the process of formulating the membership functions' parameters into the chromosomes. As stated earlier, the controller system has two inputs, the error (e) and the error derivative (ce), and two outputs, the PD section output (u_{PD}) and the PI section output (du_{PI}) which is to be integrated and summed with the first output.

Each input and output in this system will have a fuzzy set of 7 membership functions, except the PD section output (u_{PD}) will have a fuzzy set of 5 membership functions. To decrease the number of parameters we used asymmetric membership functions, except for the function associated with the zero input, which must be centered in the zero value, **figure 5** illustrates this parameterization scheme. So the 7 membership functions fuzzy set will be described with 7 parameters.

For the output (u_{PD}) fuzzy set, another scheme was used to parameterize the fuzzy set, in this method each membership function will be represented with both c (its center) and w (its width). This method will give the system more flexibility in order to represent the nonlinearity of the system besides it decreases the number of parameters associated with each membership function. For this output the center of the membership function labeled ZE will be at the value of the nominal voltage of the system at that specific operating point (c_4). **Figure 6** illustrates this parameterization scheme. For the output (du_{PI}), the same parameterization scheme will be used except that the value of the membership function labeled ZE will be at zero value.

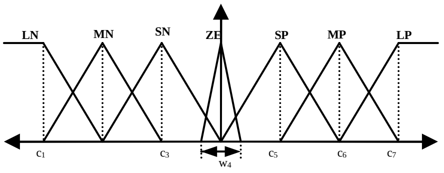


Figure 5: Parameterization scheme for the inputs *e, ce.*

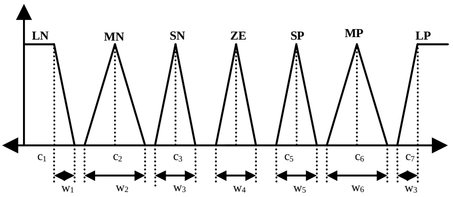


Figure 6: Parameterization scheme for the output (u_{PD}).

D. Simulation Results

A Matlab/Simulink magnetic levitation model was used during this study and the parameters of that model are as the following:

Ball mass (m) =	0.00837	kg
Coil resistance (R) =	1.0	Ω
Coil inductance (L) =	3	mH
Constant (c) =	2.16e-6	N.m^2/A^2
Power Amplifier Gain =	0.3	A/V
D/A gain =	5	
Position sensor gain =	1000	V/m
A/D gain =	0.2	
Operating air gap =	1 – 6	mm

The fuzzy control architecture and GA's optimization for fuzzy controllers were discussed in previous sections. In this section, the above mentioned controller is simulated for the control of the magnetic suspension system. Through simulation, we will show the utility of fuzzy control architecture and GA's optimization in this application.

An initial population of 100 real-valued chromosomes will be randomly initialized, the size of population is a critical design decision because it affects the evaluation time, some researches uses fuzzy reasoning to tune the population size [15]. Then each chromosome will be checked for membership function integrity, and if any one violates the integrity conditions that chromosome will be randomly regenerated till the conditions are met. Then an interpreter takes each real-valued string and uses it to set the parameters in the fuzzy controller, ad run the simulation for a specified time period (TIME = 2 sec.) during which the performance of the controller is to be evaluated. Fitness function, used in this paper, which is defined as 1 / *ITAE* (see equation 12) and $0 \leq t \leq$ TIME. To avoid the divide-by-zero problem at the very beginning the value of the *ITAE* is not to be available for calculations except after 0.2 sec. An additional bonus will be added to the fitness value in case the controller could stabilize the sphere till the end of the test period TIME, this helps in distinguishing desirable and undesirable strings.

When each string in the population has been evaluated and given fitness value the reproduction process takes place. The roulette wheel will be used here for reproduction process. Multi-point crossover will be used here because of its capability of solving one major problem of the simple crossover; this problem is that one-point crossover cannot combine certain combinations of features encoded on chromosomes. In the proposed GA algorithm Nc = 2. And the mutation probability will be set to 0.9. Since we use real-value encoding scheme, we use a relatively high mutation probability = 0.1 in our algorithm. This is different from the traditional GA's that use binary encoding scheme. The latter are largely driven by recombination, not mutation. And the mutation variable will be randomly generated factor of the value of gene to be mutated. The above learning process continues to new generations until the number of generations meets a predetermined stop criterion. The stop criterion used here is the maximum number of generations generated. After the whole evaluation process is stopped the chromosome with the largest fitness value in the last generation is selected and encoded in the final fuzzy controller. **Figure 7** illustrates the fitness of the system during the learning process. **Figure 8** illustrates the step response of the system. **Figure 9** illustrates the response of the system with a step noise equals 150% of the ball weight. **Figure 10.a** illustrates the system response to a sinusoidal trajectory with 1 Hz frequency and 0.5 mm amplitude. **Figure 10.b** illustrates the system response to a square trajectory with 1 Hz frequency and 0.5 mm amplitude. Testing the controller for robustness has been performed by investigating the system's response to an increase in the ball mass with 30% of the ball mass and checking the step response, **figure 11.a**, and system response to a step noise with the equals 150% of ball weight, **figure 11.b**. The system step response with 30% increase in the coil inductance value, see **figure 11.c**. **Figure 12.a** illustrates the fuzzy set for the input variable (*e*), **figure 12.b** illustrates the fuzzy set for the input variable (*ce*), **figure 12.c** illustrates the fuzzy set for the output variable (u_{PD}), **figure 12.d** illustrates the fuzzy set for the output variable (du_{PI}) after tuning.

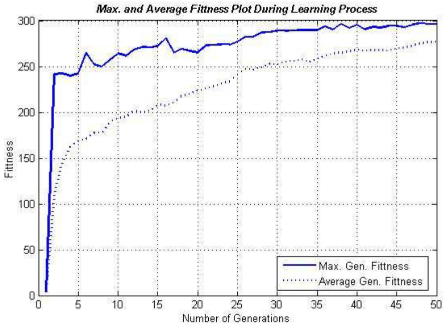


Figure 7.

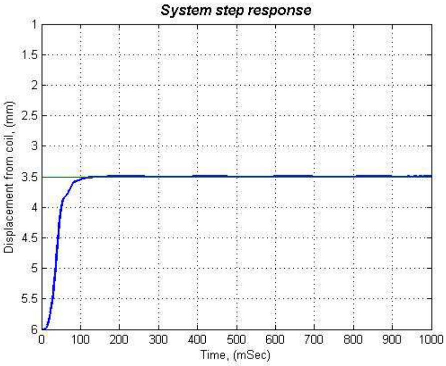


Figure 8.

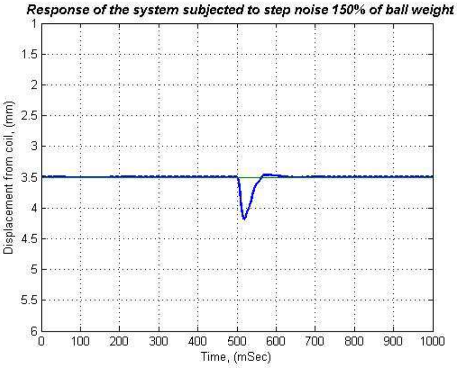


Figure 9.

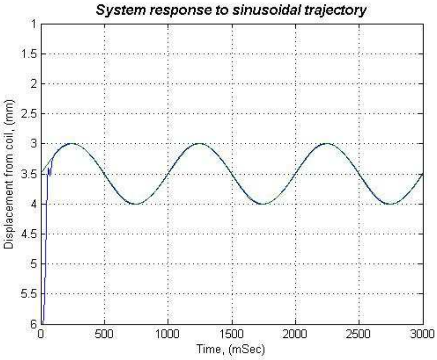


Figure 10.a

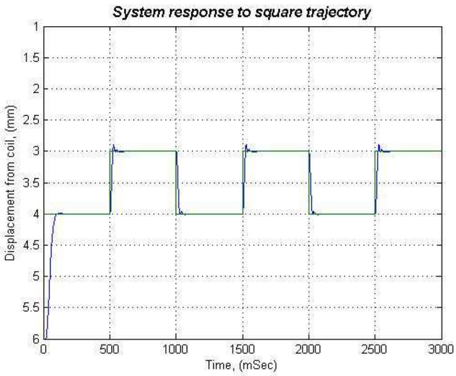


Figure 10.b.

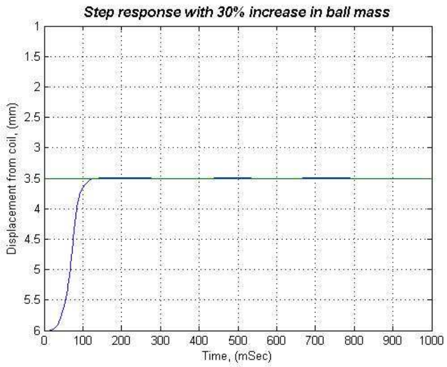


Figure 11.a

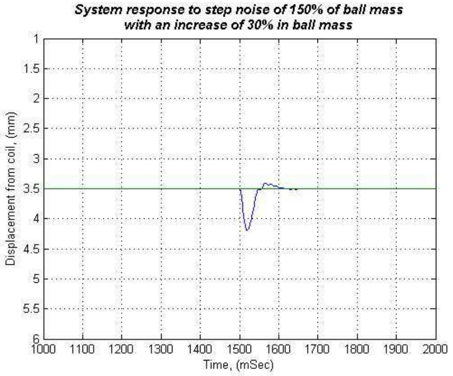


Figure 11.b

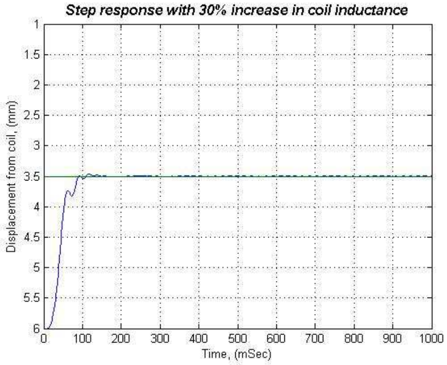


Figure 11.c

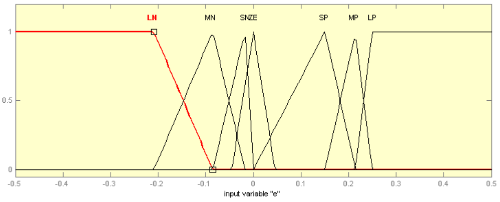


Figure 12.a: Fuzzy Set for the Input Variable (*e*).

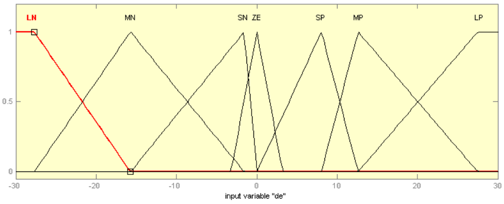


Figure 12.b: Fuzzy Set for the Input Variable (*ce*).

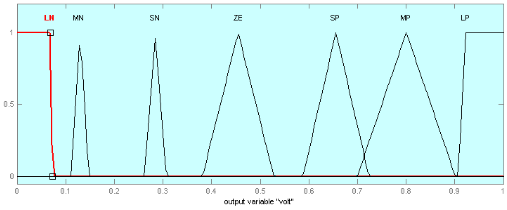


Figure 12.c: Fuzzy Set for the Output Variable (u_{PD}).

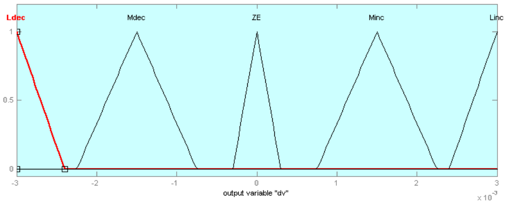


Figure 12.d: Fuzzy Set for the Output Variable (du_{PI}).

4. TAKAGI-SUGENO-KANG (TSK) CONTROLLER

The main motivation for developing TSK model is to reduce the number of rules that would otherwise be required if the Mamdani Model is to be used, especially for complex and nonlinear problems. Toward this goal, the TSK model replaces the fuzzy sets in the consequent of Mamdani rule with a linear equation of the input variables. The concept of PDC approach is to design a compensator using linear control design techniques for each operating range. The resulting overall fuzzy controller, which is nonlinear, is a fuzzy blending of each individual linear controller. In this sense, the TSK model-based control approach akin to the notion of gain scheduling technique in conventional control engineering. **Figure 13** illustrates the block diagram of the PDC architecture.

A. Parallel Distributed Compensation

The most useful way of dealing with nonlinearity of the plant is to linearize it about a single nominal equilibrium point (namely local model) in order to use linear control techniques. This approach, however, is generally effective only in a limited region of the nominal design condition, because the variable may vary in a wide range far away from the equilibrium point. In contrast, a global model describes the system's input-output relationship for the entire input space. In particular, it is well known that nonlinear global model can often be approximated by a set of linear local models. The linear local models which can be described by;

$$\dot{x} = Ax + B\tilde{u} \tag{14}$$

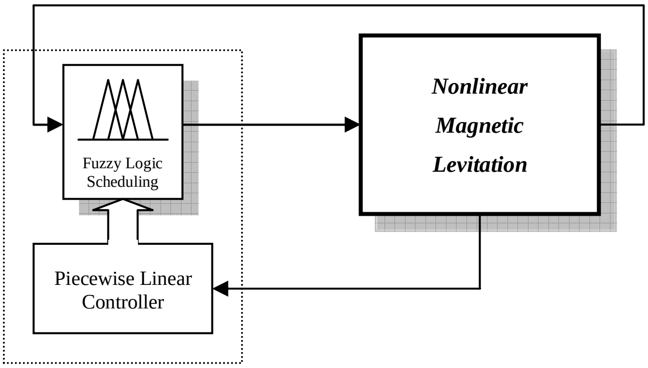


Figure 13: Block Diagram of PDC approach.

In this paper, state feedback control design via pole placement was used for each local piecewise linear system. Although being open-loop unstable in nature, magnetic suspension system is completely controllable since the controllability matrix $Wc = [B \quad AB]$ has a full rank of 2. Thus linear state feedback is possible and we shall choose the control signal to be;

$$\underline{U} = K\underline{X} \tag{15}$$

where $K=[k_1 \quad k_2 \quad k_3]$, and the control parameters k_1, k_2, and k_3 are selected so that all design requirements are satisfied. The *ith* fuzzy control rule of the controller takes the form;

$$IF\ \ x\ \ is\ \ (ith\ region)\ \ THEN\ \ u\ =\ -K_i\,x \tag{16}$$

Each of the rules can be viewed as describing a "local" state-feedback controller associated with the corresponding "local" linear sub-model of the systems to be controlled.

B. System Linearization

The first step in our approach is to linearize the system at different operating points to cover all the operating range. Three operating points at 2, 3.5 and 5mm were defined here.

The linearized equations describing the variation from the Taylor series expansion; if the variables of the operating point are expressed with the subscript "0" and the variables at the neighborhood of the operating point are represented with the subscript "1" then the linearized equations are;

$$f_1 = k_i i_1 + k_x x_1 \tag{17}$$

where

$$k_i = \left.\frac{\partial f}{\partial i}\right|_{\substack{i=i_{\circ\circ}\\ x=x_{\circ\circ}}} = \frac{2ci_\circ}{x_\circ} \tag{18}$$

$$k_x = \left.\frac{\partial f}{\partial x}\right|_{\substack{i=i_{\circ\circ}\\ x=x_{\circ\circ}}} = \frac{-2ci_\circ{}^2}{x_\circ{}^3} \tag{19}$$

Then we get,

$$f_1 = \frac{2ci_\circ}{x_\circ{}^2}\,i_1 - \frac{2ci_\circ{}^2}{x_\circ{}^3}\,x_1 \tag{20}$$

$$e_1 = Ri_1 + L\frac{di_1}{dt} \tag{21}$$

$$m\frac{d^2x_1}{dt^2} = -f_1 \tag{22}$$

where

$$f_\circ = mg = c\left[\frac{i_\circ}{x_\circ}\right]^2 \tag{23}$$

C. Augmented State Equation with Integral Control

There have been numerous ways proposed to deal with disturbances rejection problems and to achieve robustness such as classical and adaptive control. Most of these techniques lead to difficulties when there are uncertainties in the systems model. On the other hand, fuzzy logic has been considered as a potentially useful means for designing nonlinear controllers. So, we will aim to merge the two advantages of nonlinearity and robustness to achieve better performance.

For each sub-system, now have we a single input single output system. Being a linearized third order system, the three physical variables are;

$x(t)$ displacement, and denoted by $x1(t)$.

$v(t)$ velocity, and denoted by $x2(t)$.

$i(t)$ current in the coil and denoted by $x3(t)$.

which are selected as state variable, i.e.;

$\underline{X}^{T}(t) = [x(t) \quad v(t) \quad i(t)]$

And the open loop system for each linear sub-system will be described by;

$$\begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \\ \dot{x}_3 \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 \\ \frac{ki_{\circ}}{mx_{\circ}} & 0 & \frac{-k}{m} \\ 0 & 0 & \frac{-R}{L} \end{bmatrix} \underline{X} + \begin{bmatrix} 0 \\ 0 \\ \frac{1}{L} \end{bmatrix} \underline{U} \tag{24}$$

$$[y] = [1 \quad 0 \quad 0]\underline{X} \tag{25}$$

In vector form,

$$\dot{\underline{X}} = A\underline{X} + B\underline{U} \tag{26}$$

where

$$A = \begin{bmatrix} 0 & 1 & 0 \\ \frac{ki_{\circ}}{mx_{\circ}} & 0 & \frac{-k}{m} \\ 0 & 0 & \frac{-R}{L} \end{bmatrix}$$

and

$$B^{T} = \begin{bmatrix} 0 & 0 & \frac{1}{L} \end{bmatrix}$$

$$y = C\underline{X} \tag{27}$$

where

$$C = [1 \quad 0 \quad 0]$$

To get the augmented equation, we start with an ad hoc solution to integral control by augmenting the state vector with the desired dynamics. For the system given, we can take the integral of the error, ($e = r - y$) where r is the desired operating point, as well as the state of the plant x, by augmenting the plant state with extra (integral) state x_I, which obeys the differential equation $x_I = r - C\underline{X}$.

The augmented state equations become;

$$\begin{bmatrix} \dot{\underline{X}} \\ \dot{x}_I \end{bmatrix} = \begin{bmatrix} A & 0 \\ -C & 0 \end{bmatrix} \begin{bmatrix} \underline{X} \\ x_I \end{bmatrix} + \begin{bmatrix} B \\ 0 \end{bmatrix} \underline{U} \tag{28}$$

and the feedback law is;

$$\underline{U} = K \begin{bmatrix} \underline{X} \\ x_I \end{bmatrix} = KX_{aug} \tag{29}$$

where $K = \begin{bmatrix} k_1 & k_2 & k_3 & k_4 \end{bmatrix}$ (30)

With this revised definition of the system, we can apply full state feedback control design techniques. In this paper, damping ratio ($\eta = 0.7$) and natural frequency ($\omega_n = 90$) were set as design requirements.

$$A_{aug} = \begin{vmatrix} 0 & 1 & 0 & 0 \\ \frac{ki_\circ}{mx_\circ} & 0 & \frac{-k}{m} & 0 \\ 0 & 0 & \frac{-R}{L} & 0 \\ -1 & 0 & 0 & 0 \end{vmatrix} \tag{31}$$

$$B_{aug}^T = \begin{vmatrix} 0 & 0 & \frac{1}{L} & 0 \end{vmatrix} \tag{32}$$

Let us assume that certain specifications are to be met for its transient response, i.e., the desired closed loop dominant poles are located at $(-\alpha \pm j\beta)$, the third and the fourth closed loop pole should lie at $(-\gamma)$. The characteristic equation of the desired closed loop system is;

$$\lambda^4 + (2\alpha + 2\gamma)\lambda^3 + (4\alpha\gamma + \gamma^2 + \alpha^2 + \beta^2)\lambda^2 + \gamma(2\alpha^2 + 2\beta^2 + 2\alpha\gamma)\lambda + \gamma^2(\alpha^2 + \beta^2) = 0 \tag{33}$$

If we apply full state feedback, i.e. the control signal is;

$$\underline{U} = K\underline{X}_{aug} \tag{34}$$

where $K = \begin{bmatrix} k_1 & k_2 & k_3 & k_4 \end{bmatrix}$

then we get the state equation for the feedback system as;

$$\dot{\underline{X}}_{aug} = (A_{aug} + B_{aug}K)\underline{X}_{aug} \tag{35}$$

The characteristic equation of the closed loop system in terms of system parameters and state feedback gain is;

$$\left|\lambda I - (A_{aug} + B_{aug}K)\right| = \lambda^4 + \left(\frac{R - k_3}{L}\right)\lambda^3 + \left(\frac{kk_2}{mL} - \frac{ki_\circ}{mx_\circ}\right)\lambda^2 + \left(\frac{ki_\circ(k_3 - R)}{mx_\circ L} + \frac{kk_1}{mL}\right)\lambda - \frac{kk_4}{mL} = 0 \tag{36}$$

From the characteristic equation of the desired closed loop system and the characteristic equation of the closed loop system in terms of system parameters and state feedback gain we can obtain the feedback gains k_1, k_2, k_3 and k_4. In the same manner we can get the state feedback gains for each subsystem associated with each operating point.

The fuzzy rules then will take the following form:

Rule #1:

IF x is near THEN

$v = 7386.34 * x + 34.2 * x' - 3.158 * i - 191804.9 * e_I$

Rule #2:

IF x is medium THEN

$v = 12319.6 * x + 59.5 * x' - 3.158 * i - 335695.6\ e_I$

Rule #3:

IF x is far THEN

$v = 17251.5 * x + 84.7 * x' - 3.158 * i - 479541.8\ e_I$

D. Simulation Results

The step response of the system is shown in **figure 14**. The response of the system to a step noise equals 200% of ball weight is shown in **figure 15**. System response to a sinusoidal trajectory with frequency 1 Hz and amplitude 0.5 mm is illustrated in **figure 16.a**. System response to a square trajectory with frequency 1 Hz and amplitude 0.5 mm is illustrated in **figure 16.b**. Testing the controller for robustness has been performed by investigating the system's response to an increase in the ball mass with 30% of the ball mass and checking the step response, **figure 17.a**, and system response to a step noise with the value of 150% of ball weight, **figure 17.b**. The system step response with 50% increase in the coil inductance value, see **figure 17.c**.

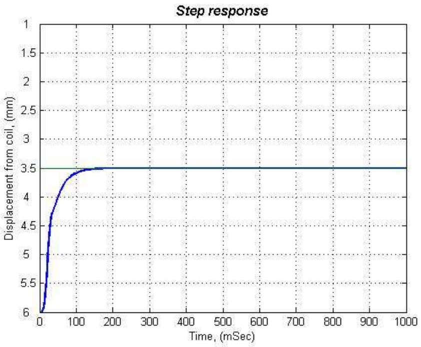


Figure 14

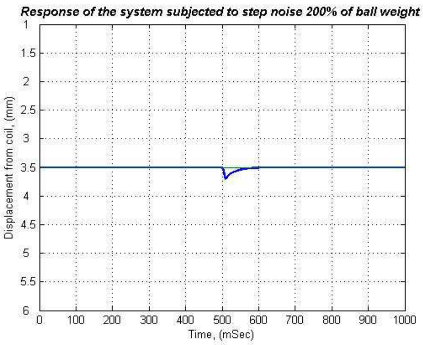


Figure 15

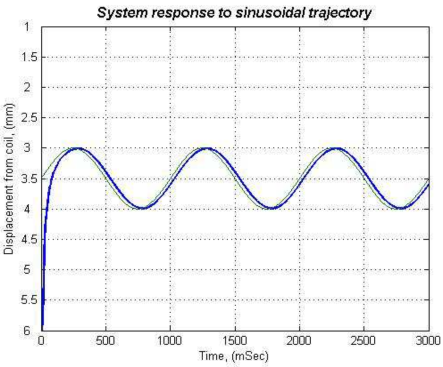


Figure 16.a

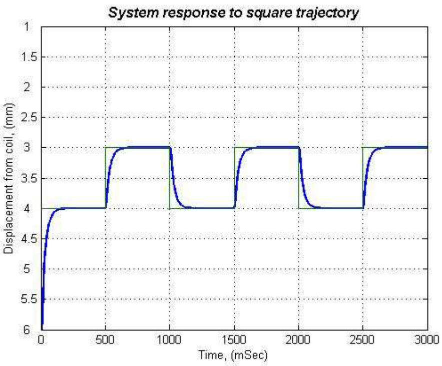


Figure 16.b

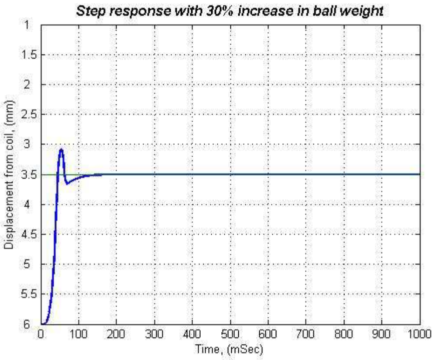


Figure 17.a

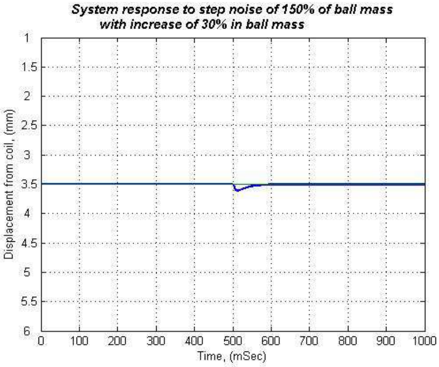


Figure 17.b

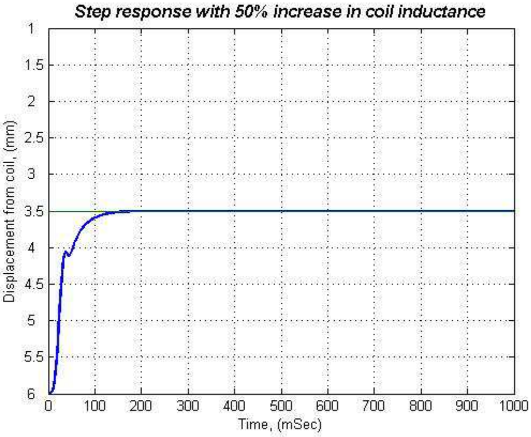


Figure 17.c

5. NEURO-FUZZY CONTROLLER

Most of the supervised learning algorithms for neuro-fuzzy systems require precise training data sets for setting the weights and the connectivity of the links for various applications. For some real-world applications, precise data for training/learning are usually difficult, if not impossible, to obtain. For this reason, there has been a growing interest in reinforcement learning algorithms for neuro-fuzzy systems.

One of the approaches to reinforcement learning is by the use of GA’s. This paper attempts to use GA's to tune a neuro-fuzzy controller for control of the magnetic suspension system. All of the parameters of the controller are simultaneously tuned by GA's.

A. Controller Description

The proposed is a feedforward five layered system with the structure 2 × 14 × 49 × 14 × 2 as shown in **figure 18**. The controller will have two inputs, error (e) and error derivative (ce) as described by equations (8, 9), respectively. And outputs of the controller are $u_{PD}(kT)$ providing the PD action and $du_{PI}(kT)$ which provides the PI action of the controller. We shall next describe the functions of the nodes in each of the following five layers of the proposed system. In the following, *f* is an integration function of a node, which combines activation from other nodes to provide net input for this node, and o is an activation function of a node. In the following equations, superscript is used to indicate the layer number. The nodes in layer one are the input nodes, which represent the error e and the change of error ce.

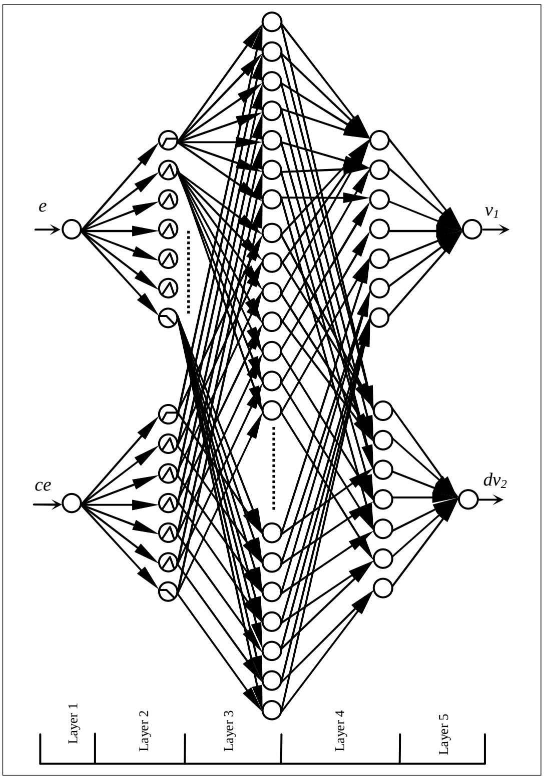


Figure 18: Proposed Neural-Network-Based Fuzzy Logic Controller.

Layer 1:

The nodes in this layer transmit input values directly to the next layer. That is;

$$O_i^1 = I_i^1 = x_i^1, x_i = e \quad or \quad ce \tag{37}$$

where (i) is the number of inputs to the controller and it will be 2 in this case. From (37), the link weight (w_i^1) is unity.

Layer 2:

If we use a single node to perform a simple membership function, then the output function of this node should be this membership function.

$$I_j^2 = w_{ij}^2.O_j^1 \quad or \quad I_j^2 = O_j^1 \tag{38}$$

For example, for a triangular function defined as;

$$f = T_{x_i}^{j}(a_{ij}, b_{ij}, c_{ij}) = \begin{cases} 0 & x_i < a_{ij} \\ (x_i - a_{ij})/(b_{ij} - a_{ij}) & a_{ij} \le x_i \le b_{ij} \\ (c_{ij} - x_i)/(c_{ij} - b_{ij}) & b_{ij} \le x_{ij} \le c_{ij} \\ 0 & x_i > c_{ij} \end{cases} \tag{39}$$

and $O_j^2 = f(I_j^2)$ (40)

where a_{ij}, b_{ij} and c_{ij} are respectively, the left point, the center and the right point of the triangular function of the *j*th linguistic value of the *i*th linguistic variable. Weights of this layer will be unity.

Layer 3:

Nodes in this layer are used to perform the precondition matching of fuzzy logic rules. Hence, the rule nodes should perform the fuzzy AND operation.

$$I_l^3 = \min{}_j (w_{jl}^3 . O_j^2) \tag{41}$$

and $O_l^3 = f_l^3(I_l^3) = I_l^3$ (42)

where (l) is the number of fuzzy rules. The link weight at layer three is (w_l^3).

Layer 4:

The links at layer four should perform the fuzzy OR operation to integrate the fired rules which have the same consequence.

$$I_k^4 = \max{}_l (w_{lk}^4 . O_l^3) \tag{43}$$

and $O_k^4 = f_k^4(I_k^4) = I_k^4$ (44)

where (k) is the number of the consequent values for the two outputs. The link weight at layer four (w_k^4) is unity.

Layer 5:

The nodes in this layer transmit the decision signals to out of the network. The nodes and the layer-five links attached to them act as the defuzzifier. If b_k's are the centers of the triangular membership functions then the following function can be used to simulate the *Height* defuzzification method.

$$I_m^5 = \sum_k w_{km}^5 . O_k^4 \tag{45}$$

and $O_m^5 = \dfrac{I_m^5}{\sum_k O_k^4}$ (46)

Here the link at layer five (w_{km}^5) is b_k , and (m) is the number of output nodes[7, 16].

B. Optimization of Neuro-Fuzzy Controllers Using GA

As stated earlier, the proposed neuro-fuzzy system consists of five layers. Layer 1 only transmits the input values and has nothing to be tuned as the links' weights of this layer are unity. Layer 2 contains nodes representing the triangular membership functions describing the fuzzy set of the input variables, and the seven fuzzy sets describing each input will be distributed evenly in the universes of discourse and constant during optimization process, **figure 19**, and also no parameters to be tuned in this layer as the links' weights are unity for this layer. Layer 3 represents the rule nodes, the link of this layer has weight matrix [$w_1^3, w_2^3, ..., w_{49}^3$], where the links coming from the same node in the previous layer will have the same weights, i.e.

$$\begin{aligned}
&w_1^3 = w_2^3 = w_3^3 = w_4^3 = w_5^3 = w_6^3 = w_7^3 = W_1^3 \\
&w_8^3 = w_9^3 = w_{10}^3 = w_{11}^3 = w_{12}^3 = w_{13}^3 = w_{14}^3 = W_2^3 \\
&\vdots \\
&w_{92}^3 = w_{93}^3 = w_{94}^3 = w_{95}^3 = w_{96}^3 = w_{97}^3 = w_{98}^3 = W_{14}^3
\end{aligned}$$

In this approach, the matrix of parameters to be optimized for this layer is $W^3 = [W_1^3\ W_2^3\ W_3^3\ ....W_{14}^3]$. Layer 4 which perform the fuzzy *OR* operation has unity links' weights and nothing for tuning. Layer 5 which contains the output nodes performs the defuzzification function. This layer has links' weights matrix $W^5 = [W_1^5\ \ W_2^5\ \ W_3^5 ....\ W_{14}^5]$ which simulates the centers of the output seven fuzzy sets associated with each of the two outputs.

In this manner, the total number of parameters to be optimized will be 28 parameters. The chromosome, which represents the potential solution in the problem space will consist of the parametric representations of the fuzzy sets specified for each input and output variable, see **figure 20**.

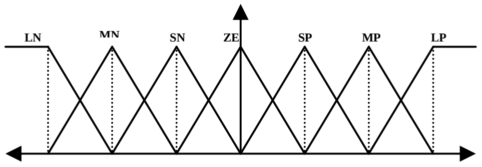


Figure 19: Representation of a Fuzzy Set with 7 Membership Functions Distributed Evenly in The Universe of Discourse.

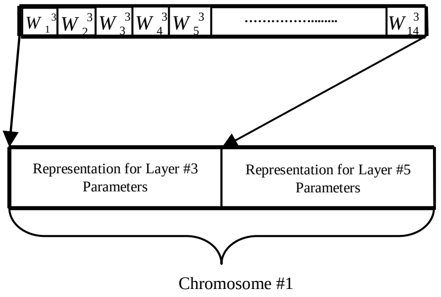


Figure 20: Encoding Neuro-fuzzy System Parameters into the Chromosome

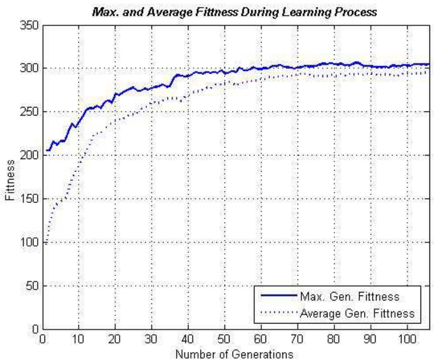
C. Simulation Results

In this section, the above mentioned neuro-fuzzy controller is simulated for the control of the magnetic suspension system. Through simulation, we will show the utility of neuro-fuzzy control architecture and GA-optimization.

Parameters Initialization of layer 3 link weight matrix W^3 parameters starts with initializing them as a random value between 0.25 and 1.5, while parameters of layer 5 link weight matrix associated with the first output will be a random value between 0 and 1 taking into account the logic order of the fuzzy set membership function centers, i.e. the center of LP membership function must be to the right of (or greater than) the center of MP membership function. Parameters associated with the second output in the same layer will be initialized to be between -0.03 and 0.03, taking into account the logic order of the fuzzy set membership functions as described earlier. An initial population of 100 real-valued chromosomes will be randomly initialized. Then each of the string will be checked for membership function integrity, and if any string violates the integrity conditions that string will be randomly regenerated till the conditions are met. Then an interpreter takes each real-valued string and uses it to set the parameters in the fuzzy controller, and run the simulation for a pre-specified time period (TIME = 2.5 sec.) during which the performance of the controller is to be evaluated.

The fitness function, used in this paper, which is defined as 1 / *ITAE*. Again to avoid the divide-by-zero problem at the very beginning, the value of the *ITAE* is not to be available for calculations except after 0.2 sec. An additional bonus will be added to the fitness value in case the controller could stabilize the sphere till the end of the test period (TIME); this helps in distinguishing desirable and undesirable strings especially at the beginning of the tuning process.

When each string in the population has been evaluated and given fitness value the reproduction process takes place. The roulette wheel will be used here for reproduction process. Multi-point crossover will be used here because of its capability of solving one major problem of the simple crossover; this problem is that one-point crossover cannot combine certain combinations of features encoded on chromosomes. In the proposed GA algorithm Nc = 2. And the mutation probability will be set to 0.9. Since we use real-value encoding scheme, we use a relatively high mutation probability = 0.1 in our algorithm. This is different from the traditional GA's that use binary encoding scheme. The latter are largely driven by recombination, not mutation. And the mutation variable will be randomly generated factor of the value of gene to be mutated. The above learning process continues to new generations until the number of generations meets a predetermined stop criterion. The stop criterion used here is the maximum number of generations generated. Finally, after the whole evaluation process is stopped the string with the largest fitness value in the last generation is selected and encoded in the final fuzzy controller. **Figure 21** illustrates the fitness of the system during the learning process. **Figures 22** illustrate the step response of the system for a 2.5mm unit step at the operating point. System responses to sinusoidal and square trajectory with amplitude 0.5 mm and frequency 1 Hz respectively are illustrated in **figures 23 (a, b) Figure 24** illustrates the response of the system to a step noise equals 150% of the ball weight.



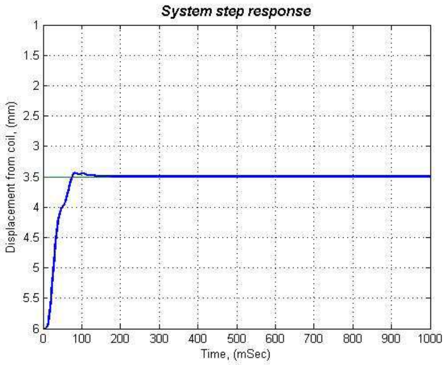
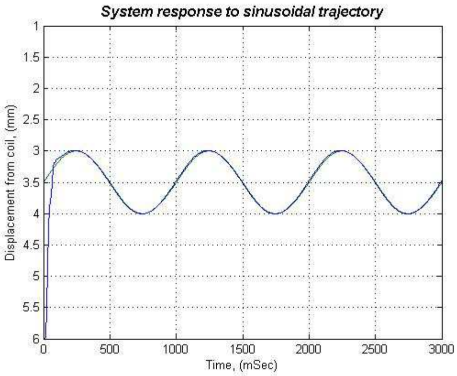


Figure 21

Figure 22



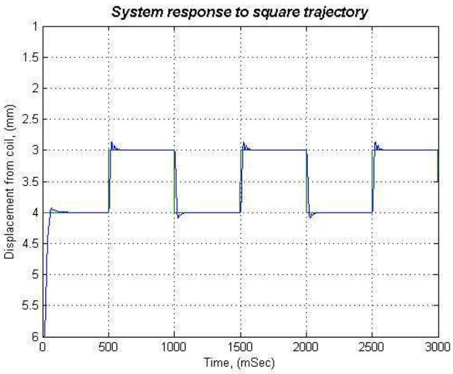
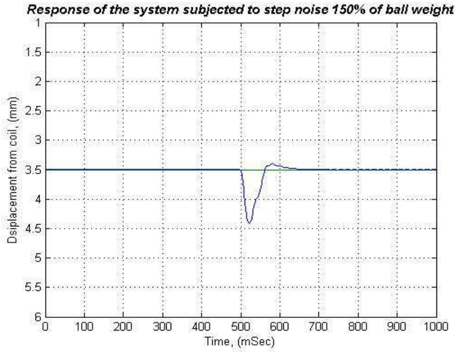


Figure 23.a

Figure 23.b



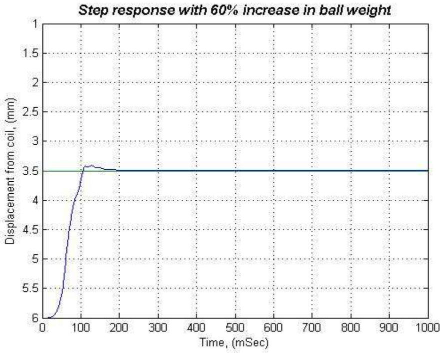
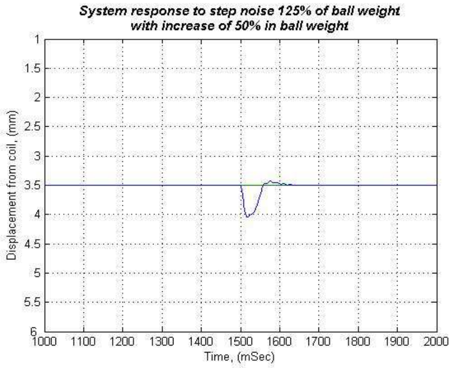


Figure 24

Figure 25.a



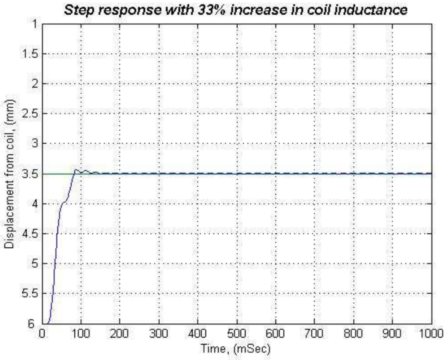


Figure 25.b

Figure 25.c

Testing the controller for robustness has been performed by investigating the system's response to an increase in the ball mass with 60% of the ball mass and checking the step response, **figure 25.a**, and system response to a step noise with a value equals 125% of ball weight, **figure 25.b**. The system step response with 33% increase in the coil inductance value, see **figure 25.c**.

6. CONCLUSIONS

For the Mamdani type fuzzy PID controller, we have proposed an alternative method to build the fuzzy PID controller which keeps the physical point of view, and we used GA's for parameters selection. Simulation results of step response of the magnetic suspension system showed a short rise and zero steady state error. The controller also showed fast tracking performance and very small error in case of sinusoidal and square position trajectories. Also, the controller is capable of rejecting disturbances and compensating the effect of sudden change in the suspended object's mass. The controller has been tested for robustness, and it showed good step response in case of increased ball mass and coil inductance with zero steady state error. We can conclude that the fuzzy logic is effective for nonlinear systems, with unknown parameters and disturbances. This methods turned fuzzy controllers design for this system into a systematic method, by using the same PI and PD rule base for different systems and accomplish parameters tuning by the standard GA's.

We also proposed TSK fuzzy controller, with the so called parallel distributed compensation (PDC) technique to facilitate the design of gain scheduling controllers and also to provide a systematic approach to blending rules. Simulation results of step response of the magnetic suspension system showed a good step response with a short rise and settling time. Using different values of η and ω_n, it will be obvious that the dynamic response of the system depends totally on the design constrains set for each subsystem. The controller showed delayed tracking performance. The controller also showed a very good capability to compensate for modeling errors and uncertainties and could reject step noises under these conditions.

The third and the last method proposed here is a modified version of NN-FLC structure. This approach had a reduced number of parameters (weights) to be tuned online by GA's, which means shorter training cycle, and also employs fuzzy if-then rules. The controller also showed fast tracking performance and almost zero steady state error in case of sinusoidal and square position trajectories. Also, the controller is capable of rejecting disturbances and compensating the effect of sudden change in the suspended object's mass. The controller has been tested for robustness also, and it showed good step response with zero steady state error in case of increased ball mass and coil inductance.

Figure 26 indicates the integral time absolute error (ITAE) value for the step response and the square and sinusoidal position trajectories for the three proposed approaches. The time span chosen for comparison is 3 seconds corresponding to 3000 samples.

It is clear that neuro-fuzzy controller which uses GA's to accomplish parameters tuning gave to some extent better results than the two other controllers for step response. And both the fuzzy PID and neuro-fuzzy controllers showed very close tracking performance for the tested trajectories. While the TSK fuzzy controller showed a relatively slow tracking performance, yet it could achieve a good step response.

From the tests accomplished to check the three controllers' behavior against noise rejection, the TSK fuzzy controller gave the best results hence it could compensate for

200% of ball mass step noise. The three controllers gave good results when tested for robustness.

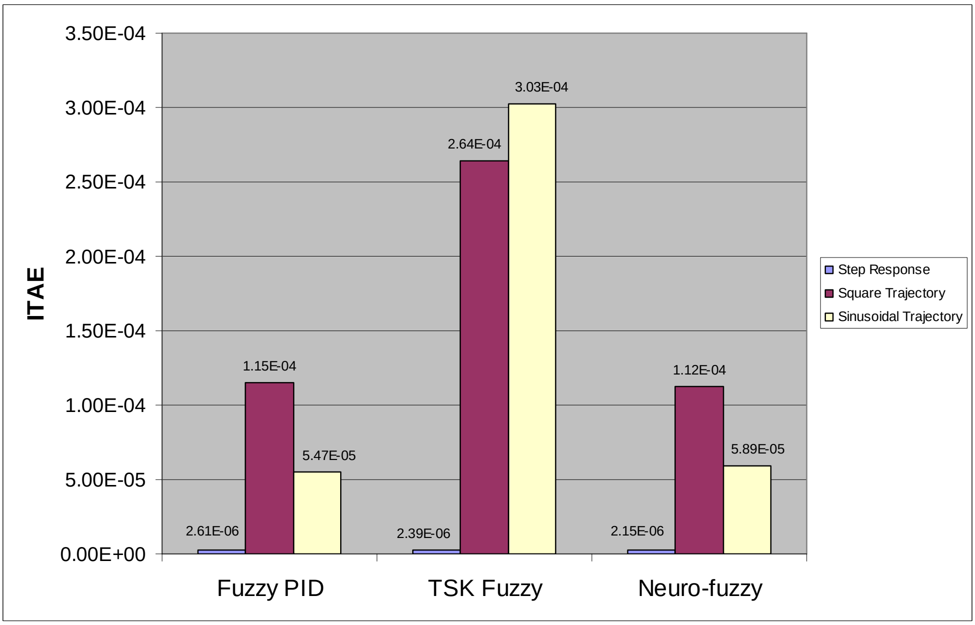


Figure 26: ITAE of step response and two different trajectories.

REFERENCES

[1] Amer Abd El-Fattah "Analysis and design of magnetic levitation control systems", Master of science thesis, Assiut University, 1994.

[2] Abd El-Fattah M. Mohamed "Optimal control design and nonlinear dynamics of magnetic bearings systems", Doctor of philosophy dissertation, University of Maryland, USA, 1990.

[3] Basil Hamed "Comparison of fuzzy logic and classical controller designs for nonlinear systems", Doctor of philosophy dissertation, New Mexico State University, USA, 1999.

[4] P.T. Chan, W.F. Xie, A.B. Rad "Tuning of fuzzy controller for an open-loop unstable system: a genetic approach", Elsevier Science Ltd., Fuzzy Sets and Systems, Volume 111, Issue 2, Pages 137-152, 2000.

[5] Mohamed-Reza Akbarzadeh-Totonchi "Fuzzy control and evolutionary optimization of complex systems", Doctor of philosophy dissertation, New Mexico State University, USA, 1999.

[6] Guo-Min Li "Robust control strategies for motor servo systems", Doctor of philosophy dissertation, The Hong Kong Polytechnic University, Hong Kong, 1999.

[7] John Yen and Reza Langari "Fuzzy logic: intelligence, control, and information", Prentice Hall, 1999.

[8] Andreas Nürenberger and Rudolf Kruse "Neuro-fuzzy techniques under Matlab/Simulink applied to a real plant", Fuzzy Systems Proceedings, 1998.

IEEE World Congress on Computational Intelligence., The 1998 IEEE International Conference on Volume 1, May 1998 Page(s):572 - 576 vol.1.

[9] J. F. M. Amaral, M. M. Vellasco and M. A. C. Pacheco "A neuro-fuzzy-genetic system for automatic setting of control strategies", IFSA World Congress and 20th NAFIPS International Conference, 2001. Joint 9thVolume 3, 25-28 July 2001 Page(s):1553 - 1558 vol.3

[10] C. T. Lin and C. S. George Lee "Reinforcement structure/parameter learning for neural-network-based fuzzy logic controllers", Fuzzy Systems, IEEE Transactions, Volume 2, Issue 1, Feb. 1994 Page(s):46 - 63.

[11] Petr Pivoňka "Comparative Analysis of Fuzzy PI/PD/PID Controller Based on Classical PID Controller Approach", In Proc. of the 2002 IEEE World Congress on Computational Intelligence, USA, 2002, pp. 541-546.

[12] M.H. Lim*, S. Rahardja, B.H. Gwee "A GA paradigm for learning fuzzy rules", Fuzzy Sets and Systems, Volume 82, Issue 2, September 1996, Pages 177-186, Elsevier Science, 1996.

[13] D. Whitley and T. Starkweather, "Optimizing small neural networks using a distributed genetic algorithm" Int. Joint Conf. Neural Networks, 1990.

[14] D T Pham and D Karaboga. "Genetic algorithms with variable mutation rates: application to fuzzy logic controller design", IMechE, 1997, Proc. Instn. Mech. Engrs. Vol. 211 Part I.

[15] Chin-Chih Hsu, Shin-Ichi Yamada, Hideji Fujikawa, and Koichiro Chida "A fuzzy self-tuning parallel genetic algorithm for optimization" Computers & Industrial Engineering, Vol. 30, pp. 883-893, Elsevier Science Ltd., 1996.

[16] S. Rajasekaran, G. A. Vijayalakshmi Pai "Neural networks, fuzzy logic, and Genetic algorithms synthesis and applications", Prentice Hall of India, 2003.

التحكم الذكي لنظام التعليق المغناطيسي

هذا البحث يتناول تطبيق التقنية الغيمية و العصبية الغيمية الغيمية على التحكم نظام التعليق المغناطيسي الغير متزن مع استخدام اللوغاريتمات الجينية كلوغاريتم توليف. يتم بحث متحكم تناسبي – تكاملي – تفاضلي غيمي من نوعية **Mamdani** والذي يعتمد بنية تحكم ثنائية المرحلة؛ حيث يهدف المتحكم الفرعي التناسبي – التكاملي الغيمي إلى إلغاء خطأ الوضع النهائي بينما يقوم المتحكم الفرعي التناسبي – التفاضلي الغيمي بعمل التوازن للنظام. يتم تعيين متغيرات دوال العضوية الخاصة بكل متحكم فرعي باستخدام اللوغاريتمات الجينية. كما يتم تناول متحكم غيمي من نوعية **Takagi-Sugeno-Kang** باستخدام طريقة التكافؤ المتوازي المقسم. المتحكم المقدم (وهو متحكم لا خطي) هو عبارة عن خلط غيمي لمتحكمات خطية منفردة مصممة باستخدام تقنية تعيين موضع القطب. أيضاً يتم تناول متحكم عصبي – غيمي حيث يتم التعليم الدعمي بالإضافة إلى اللوغاريتمات الجينية لتدريب المتحكم.