Applying Modular Networks and Fuzzy-Logic controllers to nonlinear Flexible Structures

Hava T. Siegelmann*, Azmon Ofri**, Hugo Guterman+

*Faculty of Industrial Engineering and Management Technion, Haifa 32000 Israel
**Dept. of Mechanical Engineering Technion ,Haifa 32000 Israel
+Dept. of Electrical & Computer Engineering Ben-Gurion University of The Negev, Beer-Sheva, 84105 Israel

Abstract - This paper describes a computer simulation analysis of modular networks and fuzzy-logic controllers for the motion control of 2-D nonlinear flexible structures .The Stochastic-Gradient Learning Algorithm using modular networks is presented and applied as two inputs one output controller. The self learning process of the Modular Networks consists of a set of LQR. Each one of the LQR was optimized to control a different zone, a state space. The trained controller is used to actually control the system. The two methods are compared quantitatively though the work done, performance index (settling time, energy consumption and over shoot). The comparison shows the advantage of the MN control method having better performance index results and requires less effort in the tuning stage.

1. Introduction

Flexible structures such as solar panels, large antennas and huge frame truss structures mounted on space stations are designated as large space structures (LSS). With the increase in space activity the use of light flexible structures has gained importance as a mean to reduce the high cost of launching masses into space. LSS are characterized by low natural frequencies and inherent low damping. The decrease in the weight of a LSS affects its flexibility, which in turn demands tighter active control of the system. Non-linearities are from the satellites which cannot be determined with the precision and certainty provided by traditional formulations. Furthermore, structures undergo changes during their lifetime which contribute to the uncertainties and make the control mission even harder. For simplicity and reliability, LSS employ passive vibration absorbers, e.g. mass-dampers[1-2], these devices suppress vibration and maintain displacements of the structure within an admissible operation envelope. In the last decade, combinations of active and passive controls were introduced [3] and the use of active controllers to achieve accuracy in the motions of flexible structures became a major factor in their construction because of their ability to increase the performance of the system and suppress it's vibrations[4-6]. The non-linearities, introduced by excitation inputs and the large displacements of the structure, can be controlled only by a complex non-linear active control.

In this paper we present and compare numerical simulations of two methods for nonlinear control of flexible solar panel mechanics, based on the 'mass-parameters' method. The control law strategy for the flexible solar panel model was to reach steady state $\dot{x} = \ddot{x} = 0$ by fast suppression of the structure's vibrations. In the first method, the non-linear control law was specified by Fuzzy-Set theory and Fuzzy Logic, which have the ability to handle the non-linearities and uncertainties in the model by using a fuzzy logic controller (**FLC**). Fuzzy sets are sets with non distinct boundaries that map members of the universe of discourse to their grades of membership by membership functions. The association between the inputs and outputs are modeled by fuzzy relations called fuzzy rules. In the second algorithm, the specified non-linear control law was formulated using **Modular-Networks (MN)**, which were first described by Jacobes and et. al. [7-8]. They suggested a modular network that learns to perform nonlinear control tasks using a piecewise control strategy called "gain scheduling", e.g. fitting a linear gain to different work spaces. In the MN algorithm, as in any other neural network algorithm, a self learning adaptive process of a network is based on a simple mathematical rule (**Stochastic-Gradient Learning Algorithm**) as will be shown in section 3.

The two nonlinear algorithms, MN and FLC, are based on similar 'divide and conquer' philosophies. The transfer function between the input and the output signals of these algorithms is nonlinear allowing the algorithms to control nonlinear complex systems. The basic idea in both systems is to split the nonlinear complex problem, i.e. the nonlinear motion equation, into simpler sub-systems. By dividing the work-space of the motion of the structure into sectors, different linear controllers can be handled separately. This enables the construction of simple linear relations (gain) for each different region. An overlap is accepted in both algorithms: more than one module can generate an output from a single input. The final output is composed of a weighted sum of the outputs from a few modules. The main differences between the methods are in the constructing of both the modules and the weights. In FLC, the output is the sum of each weighted rule. The ability of the FLC to construct its rules manually from expert knowledge is one of its advantages: by describing the system using accuracy rules, only minor changes in the final structure of the FLC are typically required. However, no optimal method to construct the final weights (of different rules) in fuzzy logic controller exists. Most of the "fine tuning" is done by trial and error with no logical or systematic method to obtain the best solution. The construction of the MN, as opposed to FLC, does not use pre-knowledge of expert both for different modules and for weights. However, the weight factor for each module is determined by the gating networks using I/O data.

This paper provides a quantitative comparison between two non-linear active control systems. Its concludes that the modular network controller is superior to FLC both in its performance indexes i.e., settling time, energy consumption and over shoot as well as in its shorter design time due to simpler and more systematic way of constructing the control strategy.

This paper is organized as follows: Section 2 describes the mathematical model of the structure and the control objectives. In section 3, the main stochastic-gradient learning algorithm is presented, using MN (modular network), and our learning and training methods (phase I and II) are described. Finally in section 5 we compare the simulation results of the two methods.

2. Structural modeling and control objectives.

There are various ways to model the dynamics of a large space structure (LSS). Rao, Pen & Venkayya [10], and Chidamparam & Leissa [11] survey some of the main approaches of modeling flexible structures, from discrete coordinates like: 'lumped masses', finite elements and Newton-Euler methods, to continuum approaches. However, as mentioned in the review of modeling techniques by Bainum in [Chap.7 of 6], structures that are required to undergo substantial relative motions such as LSS, can be described by a collection of interconnected rigid bodies. The ability to describe large vibrations with no limitations on the relative motions of the substructures makes this method applicable for modeling systems with geometric non-linearities.

The flexible beam-structure of a solar panel, as shown in Fig.1, can be described by a two dimensional Mass-Damper-Spring approach(including non-linear effects).

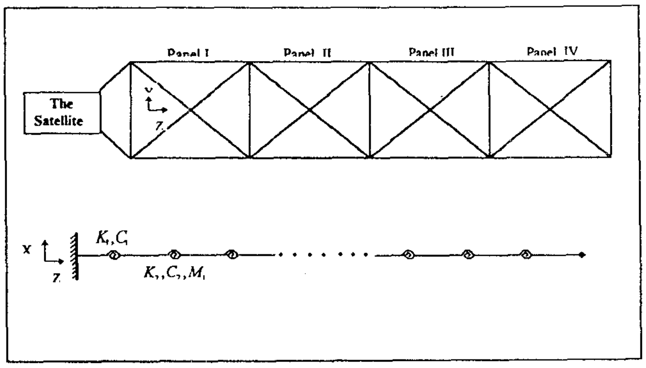

Fig.1: The 2-D model of the solar structure

The second order differential nonlinear equations that describe the flexural vibrations of the structure are presented by desecrate nonlinear equations with n nodes (DOF). Fig.2. shows the configuration around mass M_i.

The non-linear equations of motion are expressed as follows; the angle between two element is

$$\sin\theta_i = (X_i - X_{i-1}) / L_i \qquad (1)$$

where L is the panel length. The moment τ_i needed to generate a relative rotation of the rigid rods relatively to M_i, including rotational stiffness of the non-linear elements and rotational damping, is :

$$\tau_i = K_i(\Delta\theta_i + \varepsilon\,\Delta\theta_i^{\,3}) + C_i\Delta\dot{\theta}_i \qquad (2)$$

For the rod i, the equilibrium equation yields the shearing forces:

$$\phi_i = (\tau_i - \tau_{i+1}) / L_{i+1} \qquad (3)$$

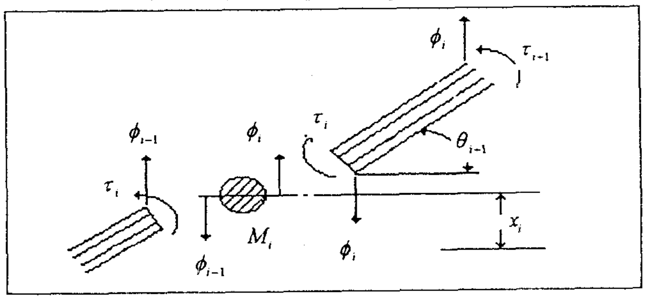

Fig.2 : 2-D model configuration around M_i

Finally, Newton's equation for the mass M_i is:

$$M_i\ddot{x}_i = \phi_i - \phi_{i-1} \qquad (4)$$

The aim of the vibration controller is to suppress the vibration of the system and to reach the steady state $\dot{x} = \ddot{x} = 0$. Using a feedback control of the form

$$u = -KX \qquad (5)$$

where **X** is the system state vector

$$X = [x\ \dot{x}]^T \qquad (6)$$

a controller is constructed such that active damping is introduced into the system. Therefore, the close loop response should exhibit a relatively high damping ratio. This ratio is related to the amount of the control effort (energy) invested to stabilize the system. To control the nonlinear dynamics we need to use a nonlinear controller for a full state adjustment. An alternative way, that we will take, is the use of 'Gain Scheduling' strategy, fitting a different linear gain to each small work space, small enough to treat it as a linear work-space. Hence, after linearization the state space formulation becomes

$$\dot{X} = AX + Bu \qquad (7)$$

where u is the force vector and

$$A = \begin{pmatrix} 0 & I \\ -M^{-1}k & M^{-1}C \end{pmatrix} \qquad (8)$$

The motion of the beam is divided into five sectors which describe different range of displacements from the desired stable motion. Each sector described by linear equations, as in Eq.(7) and a different control Gain is being constructed to meet the required performances. The quality of the control algorithm for vibration suppressing is measured by three requirements: the suppressing should be fast, it should require minimum effort, and the number of the oscillations in the structure reaching the steady state should be consider as well.

Using Eq.(8) we apply an optimum control law that minimizes the quadratic performance function:

$$J = \int_0^\infty (X^T Q X + u^T R u)dt \qquad (9)$$

where Q is a positive-semidefinite state weighting matrix and R is a positive-definite control weighting matrix. The solution is obtained by:

$$u = -R^{-1} B^T P X \qquad (10)$$

where P is found from the Riccati equation:

$$PA + A^T P - PBR^{-1} B^T P + Q = 0 \quad (11)$$

The linear Gain of the close loop system is :

$$K = -R^{-1} B^T P \qquad (12)$$

By choosing the weighting matrixes Q and R we defined the control strategy along the structure maneuver and achieved fine improvement of the controller performances.

3. Modular Network (MN)

For nonlinear systems such as LSS, the controller effort is strongly based on the state space location. Hence, by designing and combining a set of controllers, which addresses the specific requirements of different operating points, tighter control can be achieved. This strategy, similar to the conventional 'gain scheduling' method, is the basic philosophy of the **"Modular Networks"**. Modular networks attack a complex problem by dividing it into less complex ones. It consists of many simple processing units (modules) each of them has inputs/output relationship.

Two different learning algorithms are usually used to train the MN. In the first, which is based on individual learning, each module learns independently and the gating module will combine them together. Different input/output learning sets are used for each module.

The second algorithm is based on a global learning algorithm where the input/output learning sets are used to train the modules and the gating between them in parallel. As a result, each module approximates a simple function (linear gain) and if a general function is enriched we can add an additional module that will take care of the learning of the new part, without affecting too much the modules that have already been tuned. In this study we use the latter approach ; it consists of two phases:

Phase 1- By assuming linear motion in each sector we build a different linear controller (LQR) for each sector. Our goal is that each module learns the system's behavior of a specific sector .

In this phase we generate our knowledge base (learning examples).The system has two inputs and one output. The controller phase state will be the displacement (x) and the velocity ($\dot{x}$) of the tip of the beam. The output (**u**) of the controller will be the force at the actuator. For every "linear system" with a linear controller (LQR) we simulate the action of the controller for a different sector. The result of the simulation will be the learning set.

Phase 2 - Training the modules network to the desired behavior as generated in phase 1 and use the modular network as a controller of the system (beam).

In the next sections we shall describe how the network is built and the two phases of the learning and testing algorithms.

3.1 Modular network configuration

The particular modular network we use, as described in Fig 3, consists of **K** - Modules, which symbolize the linear controllers of each space, and one integrating unit called **Gating network**. Each module consists of **q** neurons in a single layer and the Gating network consists of **K** neurons in a single layer. There are **P** inputs (the size of inputs vector) for each module, $X = [x\ \dot{x}]$.

Vectors X and **d** ,the training examples are constructed by phase 1 and applied to the network simultaneously, see Fig. 3.

3.1.1 The algorithm [8]

The algorithm which is based on the **"Stochastic-Gradient Learning Algorithm"** is presented here shortly, giving only the final structure from [8].

1. **Initialization** - For both the synaptic weights of the modular expert and these of the gating network, initial values between 0 and 1 are assigned using a standard random function.

2. **Adapting the expert and gating networks** - For each epoch, inputs vector-**x** and desired response vector-**d** are presented to the network. Simultaneously, the following computation are being applied: for epochs **n** = 0,1,2,.., outputs **i** = 1,2,..,**K** and neurons **m** = 1,2,..,**q**

First, the weighted sum of the inputs applied to the ith output neuron of the gating network $\mu_i(n)$ is defined by:

$$\mu_i(n) = x^T a_i(n) \quad (13)$$

Let $g_i(n)$ be the activation of the ith output neuron of the gating network, where it is presented by:

$$g_i(n) = \frac{\exp(\mu_i(n))}{\sum_{j=1}^{K} \exp(\mu_j(n))} \quad (14)$$

For $y_i^{(m)}$, the output of the **m** neuron from the **ith** module:

$$y_i^{(m)}(n) = x^T w_i^{(m)}(n) \quad (15)$$

Therefore, the error of the output is calculated by $e_i^{(m)}$:

$$e_i^{(m)}(n) = d^{(m)} - y_i^{(m)}(n) \quad (16)$$

Adapting the Expert Networks.

The next step is to assign the new values for the weights by adding the error to them while adhering to the rule "winner takes all", namely that the module with the smallest error will get most of the update. Using a small learning rate parmeter η we modify the synaptic vector w_i as followed:

$$\Delta w_i^m (n+1) = w_i^m (n) + \eta h_i e_i^m (n) X \quad (17)$$

Where :

$$h_i(n) = \frac{g_i \exp(-0.5\|d - y_i(n)\|^2)}{\sum_{j=1}^{K} g_j \exp(-0.5\|d - y_j(n)\|^2)} \quad (18)$$

Adapting the Gating Network

Similary, the synaptic vector a_i is adjusted to the gating network, defined by:

$$\Delta a_i = \eta(h_i - g_i)X \quad (19)$$

Equations 13-19 will be used for all the available training examples in the learning set.

3. The network is considered to be at a steady state if the change in the sum of Δw and Δa is smaller than the converging parameter denoted by ε. Computations in steps-2 are to repeat until the network is stabilized.

4. Result and Discussion

For the purpose of our investigation, we selected a mathematical flexible cantilever beam model which has tight and small natural frequencies. The reduced model described by three degrees of freedom (3 elements), each one is characterized by a lenght L, mass M and nonlinear spring K (without damping C=0). The relationship between each pair of element considers the geometric and material nonlinearity (Eq.4). This model assumes only bending moments in a plane (2-D). Table 2 shows the three modes of the beam structure.

MODE	First	Second	Third
Frequency(rad/sec)	3.6278	22.223	50.539

Table 2: Natural frequencies of the cantilever beam.

A collocated single sensor (accelerometer) and actuator (vertical force) pair was located at the tip of the beam. The selected parameters for the model are:

$\mathbf{L}_i = 0.28$ [m] $\mathbf{M}_i = 0.5$ [Kg] $\mathbf{K}_i = 10$ [Kg/sec^2]

Initial conditions for the simulations were selected as follows: $[x\ \dot{x}] = [0, 1.3]$. The free response of the system is shown in Fig.4.

Two different controllers have been built based on FLC and MN respectively. The controllers have the state phase inputs, $[x\ \dot{x}]$ and one output $\boldsymbol{u}$, at the tip of the beam. Succeeding the description of the MN in section 3, the two phases were constructed in the following manner:

Phase 1- The learning set.

To generate samples from which the system has to learn, the work space of the motion of the beam was divided into five areas. Each area represents different location (displacement) of the steady state (steady state is defined as zero displacement in every element in the beam). By using the angle between the 'zero line' and the beam line we define **20** degree as '**Positive Big** ' ,10 degree as '**Positive**' and **around zero** degree as '**Zero**'; similarly, we defined the '**Negative** 'and '**negative Big** '(**-10, -20**).

The linear systems are constructed from the nonlinear structure by linearization around each work space and a linear controller, using LQR algorithm, have been built. We fitted a different gain controller K, as in Eq.(12), to every linear system to accomplish the following demands: 1) In high displacement ('Positive Big', 'negative Big') suppressing vibration is fast. Hence high energy is required in the controller. 2) In small displacement ('Positive', Negative') only small amount of energy is required. 3) When the system is almost stable (around 'zero') there is no need to take any action and the energy can be saved. These particular demands are specific to our system.

The learning and testing sets consist of the pair **X** and **u** ; these are the measurements taken from the five linear controllers.

Phase 2 - Learning and training the network.

The goal of the learning algorithm is to model these five linear systems and to combine them by the gating network. Simulation of the modular network has been conducted. We set **k**=5, **q**=1 and **p**=2. The learning ends when weights ($\Delta\, w$ and Δ) do not increases in more then $\varepsilon = 0.005$.

Figures **5-10** show the closed loop response of the system by using MN and FLC (with and without fine tuning) for displacement of the tip and the actuator output. Three criteria were employed in the comparison of the controllers.

1) ***Settling Time*** : The time that the system reaches a stable state. By comparing the two system results , Fig **5-6**, MN controller reaches the final state approximately 4 second after initialization while the FLC without fine tuning stabiles after 7 seconds only. Fine tuning in the membership functions of the FLC (in both output and input states) decreases the settling time as much as in the MN system (Fig. 7). The remaining run time in both simulations demonstrates the ability of the controllers to hold the desired position without spillover .

2) ***Over Shoot***: The maximum displacement of the beam from the stable state. Because of the fact that the structure of the MN controller consists of a combination of linear systems, the decay of the displacement of the beam is gradual and smooth. In FLC the decay of the response is more rapid and has jumping decline. This response is due to the nature of the FLC structure ,especially the way the membership functions divided the work space of the output controller (***u***). This way of division makes the controller response nonlinear. The symmetric behavior of the FLC around the zero line is a result of the symmetric rules and the membership functions.

In both cases , MN controller and FLC (without fine tuning), the overshoot is the same while in the fine tuning FLC we see a bigger overshoot in the first oscillate. However, the vibrations are rapidly damped. Both systems, MN and FLC (with fine tuning), reach a stable state after 10 oscillations.

3)***Control effort***: Another major difference between the two controllers is due to the total energy required to suspend the vibrations in the beam, supplied by the actuator placed at the tip of the beam. Fig **8-10** describe the controller outputs in the MN, FLC without and with fine tuning. The FLC without tuning requires relatively much higher power than the MN. Fine tuning of the membership functions of the FLC reduces the energy required but still the MN controller is more efficient.

Summary

This study shows that MN based controller is effective in stabilizing the movements of the flexible structure. While for the FLC much effort it is required to obtain an 'optimum' solution (fine tuning of the membership function and the rules), the MN utilize self learning adaptive process using a learning set and required no outside intervention. Both algorithms reach the stable state at the same settling time. However a major advantage is found in the MN controller that reduced the control effort, supplied by the actuator.

Further research should be done to combine the advantages of NN to tune the membership function of the FLC.

REFERENCES

[1] P.C.Hughes, "Passive dissipation of energy in large space structures" . J.Guidance and control vol. 3 No.4 pp. 380-382 1980.

[2] J.Prucz, "Analysis and design tradeoffs for passively damped structural joints". J.Spacecraft vol. 23 No.6 pp. 576-584 1986.

[3] Dean W.Sparks Jr & Jer-nan Juany, "Survey of Experiments and Experimental Facilities for control of flexible structures". J.Guidance and control No.4 pp. 801-816 1992.

[4] Eunsup Sim & Sung Lee, " Active vibration control of flexible structures with acceleration or combined feedback". AIAA 91-1015 pp.744-753.

[5] D.C Hyand , J.L Tunkins & R.W. Longman, "Active control technology for large space structures". J.Guidance and control vol.16 No.5 pp. 801-820 1993.

[6] L. P.J. Lynch & Siva S.Banda, "Large space structures: Dynamics and control". Springer-Verlay 1988.

[7] R.Jacobs,M.I.Jordan,S.J.Nowlan & G.E.Hinton ,"Adaptive Mixtures of Local Experts". Neural Computation 3,p 79-87 1991.

[8] Haykin S. "Neural Networks" Chapter 12- 'Modular Networks' pp 473-496 1994.

[9] Yasuyuki S. , Masao T. & Kazuyuki H. " Criteria-oriented configuration control of adaptive structure and its modular neural network representation". First joint U.S/JAPAN conference on adaptive structures pp. 402-421 Nov. 13-15 1990.

[10] S.S .Rao, T.S. Pan & V.B.Venkayya "Modeling, control and design of flexible structures". Applied Mechanics Reviews ,vol. 43 No.5 Part 1,May 1990.

[11] P.Chidamparam & A.W.Leissa. "Vibrations of planar curved beams, rings and arches". App Mech Rev Vol. 46 No.9 September 1993.

[12] E.H. Mamdani & P.J.King," The application of fuzzy control systems to industrial processes". Automatica vol.13 pp.235-242 1977.

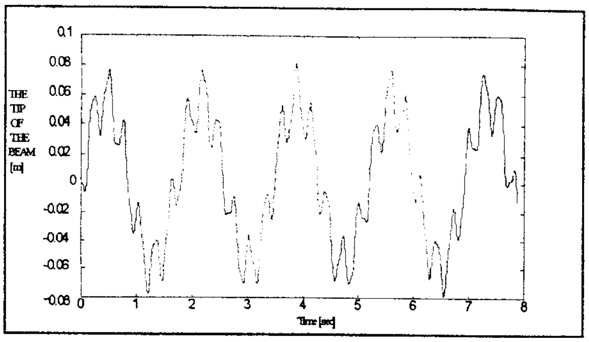

Fig4: The free response of the structure

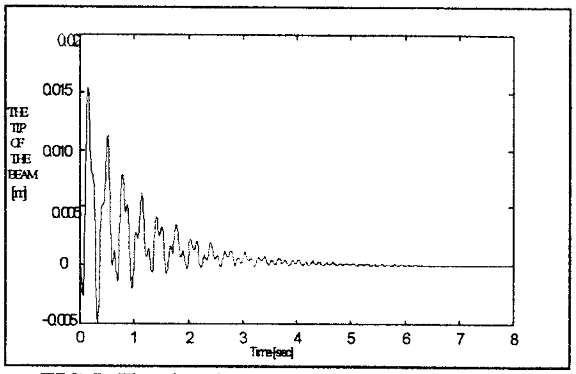

FIG 5: The close loop response of the MN system.

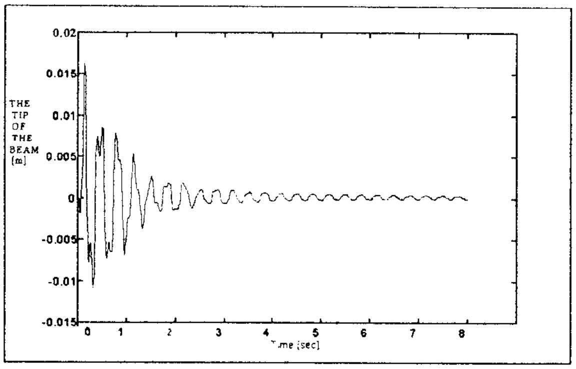

FIG 6: The close loop response of the FLC system (no fine tuning)

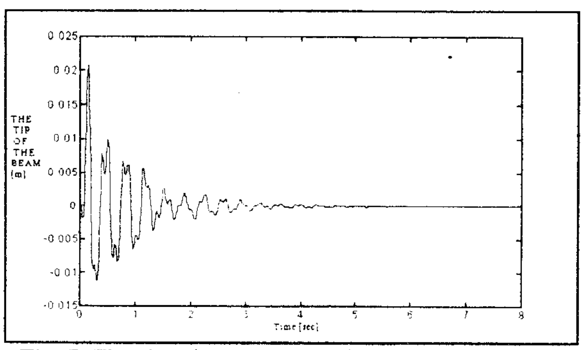

Fig. 7: The close loop response of the FLC system- (with fine tuning)

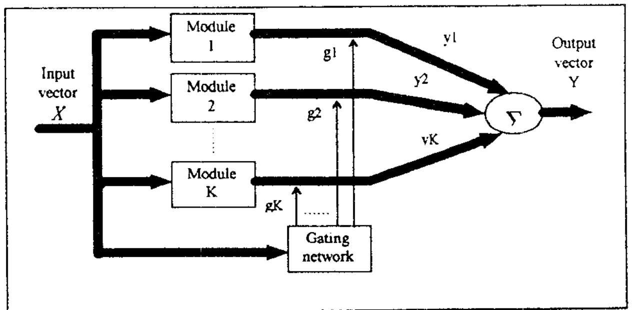

Fig 3: The Modular Networks

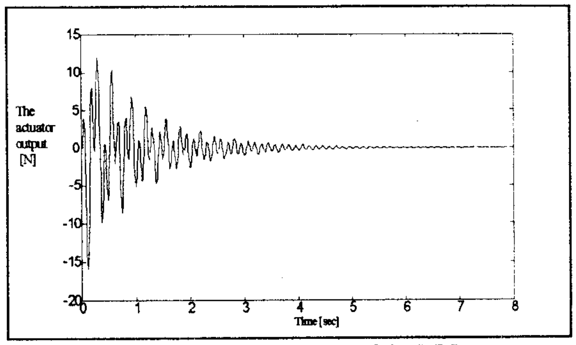

Fig8: The close loop response of the MN system.

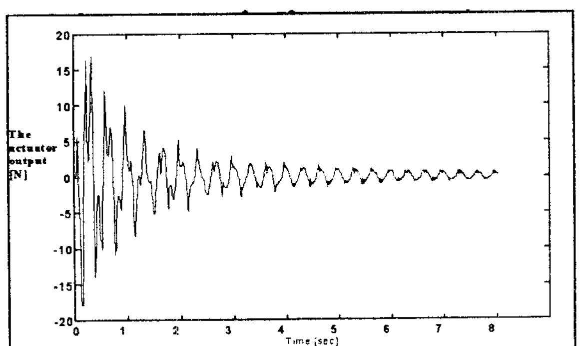

Fig 9: The close loop response of the FLC system-(no fine tuning) .

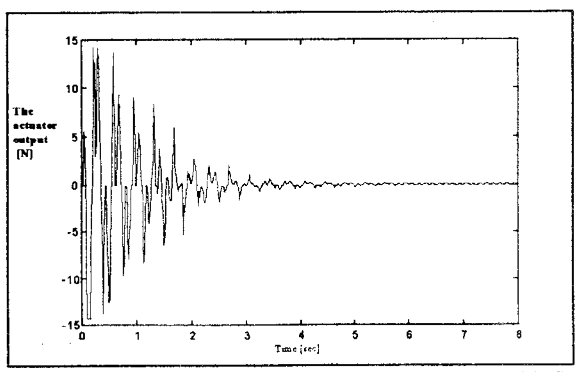

Fig 10: The close loop responseof the FLC system(with fine tuning).