AFOSR-TR-~~96~~ 97

0032

REPORT DOCUMENTATION PAGE

1. AGENCY USE ONLY (Leave blank)	2. REPORT DATE	3. REPORT TYPE AND DATES COVERED
	01.03.1994	09.15.1992 - 09.14.1993

4. TITLE AND SUBTITLE
Study of Neuro-Controllers for Motion Control Systems with Distributed Mechanical Flexibility.

5. FUNDING NUMBERS
F49620
AFOSR-92-J-0490

6. AUTHOR(S)
Dr. Sabri Cetinkunt
Associate Professor

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)
Department of Mechanical Engineering (MC 251)
University of Illinois at Chicago
842 West Taylor Street
Chicago, IL 60607-7022

8. PERFORMING ORGANIZATION REPORT NUMBER

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)
Air Force Office of Scientific Research NA
Bolling AFB, D.C. 20332-6448

10. SPONSORING/MONITORING AGENCY REPORT NUMBER

11. SUPPLEMENTARY NOTES

12a. DISTRIBUTION/AVAILABILITY STATEMENT
unlimited

19970117 228

13. ABSTRACT *(Maximum 200 words)*

Control of motion systems involving distributed mechanical flexibility is studied using artificial neural networks. Infinite dimensional nature of the problem due to distributed flexibility, nonlinear dynamics of mechanical structural systems, and fault tolerant operation requirements are taken into consideration. Three different neuro-controller architectures are studied:
1) Hopfield nets for modal parameter estimation and real-time solution of LQ optimal control problem,
2) Feedforward nets using EKF learning algorithm as a fast learning, trainable nonlinear adaptive controller,
3) CMAC neural network controller for high precision motion control.

Thre results are summarized and details are presented in refereed publications.

14. SUBJECT TERMS	15. NUMBER OF PAGES
	19
	16. PRICE CODE

17. SECURITY CLASSIFICATION OF REPORT	18. SECURITY CLASSIFICATION OF THIS PAGE	19. SECURITY CLASSIFICATION OF ABSTRACT	20. LIMITATION OF ABSTRACT

Study of Neuro-Controllers for Motion Control Systems with Distributed Mechanical Flexibility

by

Dr. Sabri Cetinkunt
Associate Professor
Department of Mechanical Engineering
University of Illinois at Chicago
Chicago, IL 60607-7022

Phone: 312 996 9611
FAX : 312 413 0447
Email: cetinkun@fred.me.uic.edu

Award number	:	AFOSR-92-J-0490
Amount	:	$ 47,600
Period	:	09.15.1992 - 09.14.1993
PIs	:	Dr. Sabri Cetinkunt
Project Title	:	Study of Neuro-Controllers for Motion Control Systems with Distributed Mechanical Flexibility

I. Introduction

The purpose of this research was to study the applications of artificial neural networks (ANNs) as intelligent controllers in motion control. Specifically, the motion control systems which exhibit distributed mechanical flexibility was the focus.

The accurate and high bandwidth motion control of systems which have distributed mechanical flexibility is a difficult and long standing problem. The difficulty is the result of the following characteristics of such systems:

1. nonlinear dynamics due to large relative rotational motion of mechanisms and structural components,
2. large dynamic order due to distributed flexibility,
3. the need for fault tolerant operation in presence of uncertainties in the operating conditions.

Dynamics of structural systems, undergoing large angular rotations and operating at high speeds, are highly nonlinear. Examples of such systems include high performance aircraft, large space manipulators, mechanical linkages. Off-line modeling based on the principles of structural mechanics is very hard to keep-track, although symbolic processing computer tools can help reduce that complexity. Furthermore, the systems operate in large variety of conditions (i.e. a partially damaged combat aircraft), all of which can not be anticipated and modeled off-line. The distributed flexibility of inertia results in infinite dimensional dynamic model. The system state can be approximated by modal truncation which results in a finite dimensional dynamic model. The question of how many modes to include in the model remains an unresolved issue. The necessary number of modes depends on the operating conditions and required accuracy in the model. As for all control systems, the robustness is a fundamental requirement in control of systems involving distributed mechanical flexibility. Particularly in life critical applications (combat aircraft, space operations), the control system should be able to learn and adapt to the changing conditions.

Why should we study artificial neural networks in control of mechanically flexible systems? Artificial neural networks have the following properties which make them very attractive for the above stated control problem:

1. learning ability of ANNs by modifying the interconnection weights,
2. the distributed storage of information and hence fault tolerant operation,
3. parallel implementation and hence increase in real-time implementation speed by several orders of magnitude.

The implication of the first property is that a proper ANN architecture can learn the nonlinear dynamics of a mechanical system which has distributed flexibility. The traditional dynamic system identification methods are based on an assumed parameterization of the system dynamics. If the assumed parameterized model is not general enough, the recursive parameter estimation algorithms may never converge. Likewise, over-parametrization may result in long convergence time and non-uniqueness of solution. In this regard, ANNs can be considered as generic powerful parameterizers.

There is no need for a-priori information to parameterize the system. One only has to choose the proper ANN architecture and size. Once this is done, the only thing needed is the desired input, measurement signals, and the output be connected to the input-output (I/O) lines of the ANN. The artificial neural networks are called the neuro-controllers when they are used in control of dynamic systems.

The artificial neural networks are engineering models which mimic the structure of the human brain . The basic building block is always a simple neuron. The neuron operates on its input and generates a function. The input to the neuron may be the weighted sum of many input signals (Fig.1). The output of the neuron is a single signal, but it may be connected to many other neurons with various weights. The function of the neuron (called the activation function) may be a static function (linear, saturation, or nonlinear) or a dynamic filter like function with local memory. Human brain neuron is believed to have local memory, where the current output is not only the function of the current input, but also the past input values. Feedforward type neural nets use neuron models with static activation functions, i.e. back propagation nets. Recurrent nets generally use neuron models with filter characteristics, i.e. Hopfield nets.

The main classification of the ANNs are based on their architecture. Common properties among all architectures are that they all use simple neuron models as the main building block or processing element, and that they are highly interconnected. The interconnection type determines the architecture. If the neurons are organized in layers, and the output of one layer feeds into the input of the next layer, it is called the *feedforward* type ANN (Fig.2a). If the output of a neuron in a layer is connected to a neuron in the earlier layer or back to itself, this creates a feedback loop. Such ANNs are called the *recurrent* type ANNs (Fig.2b). Feedforward nets are essentially a general nonlinear function approximators. They can learn (approximate, interpolate, and extrapolate) any nonlinear static input-output function. In order to capture the memory behavior of dynamic system response, a window history of input-output samples must be provided to the feedforward ANN (FF-ANN). The window size should be long enough to cover the significant part of the impulse response of the dynamic system. Hence, the FF-ANNs can be used not only to learn nonlinear static functions, but also nonlinear dynamic functions by properly sizing its dimension and providing a window of time history of nonlinear dynamic system input-output. Cerebellar model articulation controller (CMAC) neural network also belongs to this class.

Recurrent neural nets are dynamic systems. When formulated properly, the state of the neural net evolves in time and converges to some values. The converged values of states are such that they represent the local minimum of the energy function of the network. A recurrent network is uniquely defined by its architecture (i.e. Hopfield net), size, initial conditions, and the network connection weights and biases. The connection weights and biases determine the shape of energy function of a network. Therefore, the way the connection weights and biases defined determines the task it performs. The key idea is to formulate the recurrent ANN interconnection weights and biases such that they create an energy surface whose minimum is the solution of the physical optimal control or estimation problem we seek to solve. In fact, recurrent nets can be used to solve any optimization problem.

II. Summary of Results

Here we will briefly summarize the main accomplishments of the work conducted for a period of one year between Sept. 15, 1992 - Sept. 14, 1993, and supported by Air Force Office of Scientific Research. The details of every item can be found in the papers published in refereed journals and conferences. The copies of these papers are also attached to this report.

The primary objective of this project was to study the potential uses of artificial neural networks in motion control of mechanical systems which involve distributed mechanical flexibility. Three different classes of artificial neural networks were studied and applied as learning controllers in various case studies:

1. Hopfield neural networks were formulated such that they could be used as modal parameter estimators for control of linear structural systems (AIAA paper'93). The same type of ANN was also used for the real-time solution of the discrete-time LQ-optimal control problem with applications to structural systems (WAM'93 paper, and Int. J. of Neural Networks'93). A Hopfield net (HN) is a fully connected recurrent ANN type. In order to uniquely define a HN, the net size (the number of states or neurons), the neuron activation function and its saturation values (i.e. C tanh(x) function), the neuron impedance (Ri, Ci), and the interconnection weights and biases must be specified. Changing the net interconnection values and biases changes the shape of the energy function. Therefore, the network states would converge to different values if the weights and biases are different. Any optimization problem can be addressed with HN by formulating the HN such that the energy wells created by the specific weight and biases are the solution of the optimization problem. We addressed two different problems in control of systems involving distributed mechanical flexibility: 1. modal parameter estimation, 2. real-time solution of discrete-time LQ optimal control problem. The approach in both cases is to relate the Hopfield net interconnection and biases to the measured states or given dynamic model parameters. Then, as the Hopfield net states evolve in time and converges, the converged values are the solution of the optimization problem. The values to which the Hopfield net states converge represent the states which result in minimum network energy wells (Fig.3). In the case of modal parameter identification the Hopfield net states converge to the true modal parameter values. In the case of real-time LQ optimal control problem solution, the Hopfield net states converge to the optimal

trajectory and control solution. In the latter case, the order of the Hopfield net is proportional to the number of sampling time intervals taken in the LQ optimal control problem. As this number gets large, the real-time implementation of Hopfield net may not be practical. Therefore, a given computational hardware resource will dictate the time window of optimization can be performed for the optimal trajectory planning and control. This is particularly important in high performance flight control systems. A typical representative result of this approach is shown in Fig.4. Figure on the left shows the convergence of the HN approach in identifying the modal parameters of a flexible beam. Here we show only the input coupling modal gain. The other parameters of the modal dynamics (modal frequencies, damping ratios, other modal gains at the input and output locations) were also successfully estimated. The figure on the right shows the convergence of the HN solution for the LQ optimal control of flexible beam. These are the results of HN after 10,000 iterations. It clearly shows that the HN solution converged to those of Riccati solution. The two different utilization of the Hopfield net, one for the modal parameter identification and the other for the optimal control, are combined to form an adaptive learning controller (Fig.5). This is an ongoing work. The primary applications are large space structures, flexible wing aircraft flight control, missile trajectory guidance and control, and vibration control.

2. Back propagation neural networks were used in a trainable controller architecture where the neural net learns the inverse dynamics of a mechanical system, including distributed flexibility, and uses it in real-time motion control of the system (IEEE paper). The neuro-controller runs in parallel with a teacher. The teacher may be another controller or even a human operator. The neuro-controller is trained to control the dynamic system just as the teacher does. A repetitive motion sequence is designed for the training cycle. The same architectural approach is also studied with a special class of feedforward neural nets. The goal was to improve the learning rate compared to the back propagation learning algorithm. The neuro-controller is FF-ANN type where the first layer has nonlinear sigmoidal neurons, where as the last layer has neurons with linear activation functions (Fig.6). Furthermore, the input distribution nodes are fully connected to both the hidden and the output layers. It is shown that the learning rate using an extended Kalman filter type learning algorithm was two orders of magnitude faster than the back propagation learning rate. The details of this work is presented in the paper submitted to the AIAA J. of Guidance, Control, and Dynamics (AIAA'93-2). It is also shown that the error between the desired and actual output of the neuro-controllers converge to zero exponentially (Fig. 7)

3. Cerebellar Model Articulation Controller (CMAC) is a special simple class of artificial neural networks. A new mapping algorithm was developed which is the most significant component of a CMAC. We developed C-code that implements CMAC architecture in real-time on an IBM-PC. This work was also partially supported by National Institute of Standards and Technology for high precision machine tool control applications, and is on-going. The theoretical development of the work is

completed and mature enough that we are working on the real-time implementation of this approach (ACC'93 paper, ASME-JDSMC'93 paper). Unlike, the back propagation nets where for every error all of the connection weights are modified, the CMAC net modifies only a small portion of the interconnection weights for a specified error. In other words, CMAC learns a nonlinear function locally. For a given input state, the CMAC maps it to a finite number of memory locations (Fig.8). The important features of the mapping algorithm are that it must map similar states to similar over lapping memory locations, and it must map different states to different memory locations. This provides the generalization and interpolation ability to the CMAC net. Furthermore, it reduces the amount of memory required by the input space. Let us assume that we map each distinct sample of input space to one unique memory location. Assume that we have a input space of dimension 10, with each space range is discretized to 1 in 1000 part. The total number of distinct input space is 10^30. If we use one to one mapping, the required memory (10^30) is more than all the computer memory exists in the world. However, all of the input space locations are not encountered in a given operation. For instance, a robot arm passes through only a small percentage of all of points in its workspace during a trajectory, not all of the points in the workspace. Recognizing this fact one can develop mapping algorithms that require much less memory of practical size. One approach is to use hash-coding techniques. Furthermore, we developed new mapping algorithms that work reasonably well and does not need hash-coding. Hash-coding reduces the efficiency of the CMAC. Our approach for the application of CMAC in the context of the control of structural system is shown in Figure 9. The CMAC runs in parallel with a standard controller, i.e. PID type. Initially, the CMAC may have zero knowledge of the system. The control action is the sum of the CMAC and the PID controller outputs. The CMAC modifies its memory content based on the PID output. As long as there is output from the PID controller, it means that there is a tracking error and the learning continues. The learning stops when all of the control action comes from the CMAC, and none from the PID controller. Zero control command from PID means that there is no error. The CMAC alone is able to perfectly control the system. Figure 8 shows an example of CMAC learning of a nonlinear function. The results of a tracking motion control accuracy under the presence of large friction is shown in Fig. 10. The system model is a simple mass-force system. However, there is large friction in the system which is unknown to the controller. The velocity and position tracking accuracies are compared under PID control alone and the CMAC+PID control together in the table given on the right. In high precision machine tools, we need velocity control at the level of 10 micrometers per minute with tracking accuracy of 1 %, and position control accuracy less than 100 nanometers..

The following issues are found to be significant for practical applications of neuro-controllers:

- Learning convergence rate of back propagation networks is too slow, and needs to be improved for real-time control applications. In contrast, the CMAC neural network and extended Kalman Filtering based algorithms learn orders of magnitude faster than back propagation nets and seems to be more practical for real world problems.
- The back propagation through time and Hopfield network approach may have very similar properties that are yet to be understood. However, the back propagation thru time will require long periods of off-line training. Whereas, Hopfield nets can be implemented in real-time provided the analog VLSI or optical computing technologies become available. The difference is that the training set needs to be presented over and over to the back propagation network until the learning is sufficiently accurate. However, the complexity of the mathematical operations prohibits their analog VLSI or optical computing implementation. Hopfield nets, on the other hand, are nonlinear dynamic systems whose evolution in time must be calculated and the convergence rate is a function of the network time constant. Since the operations involved in implementing Hopfield nets are much more suitable for implementation on analog VLSI or optical computing, they hold a better promise for real-time. implementation than back propagation nets do.
- CMAC learning algorithm is very practical, has a fast learning rate, and may be the first one that will show significant practical application success among other artificial neural network based controller architectures. Our study of this approach for ultra-precision motion control with applications in precision machine tools has been very encouraging. The potentials of this approach should be further explored for structural control. We have developed a real-time implementation of CMAC using ANSI-C and TMS 320 DSP chip based PC bus board.
- Robustness of CMAC controller approach we proposed should be studied. It seems that the controller have very good robustness properties due to the fact that CMAC works in parallel with a PID type controller.
- The CMAC controller may take the form of appropriate controller for different situation automatically through the learning process, i.e. it may converge to an optimal adaptive PID controller, or a sliding mode controller, or a robust H_inf controller. These characteristics, which the CMAC controller may have but unknown to its user, should be studied in fundamental level.

Publications describing the accomplishments of this work:

Lin, H.C., Cetinkunt, S., ``Applications of Hopfield Neural Networks in Identification and Control of Structural Systems'', Int. Journal of Neural Networks, submitted, Sept. 1993.

Chiu, H-T.,Cetinkunt, S., ``Trainable Neural Network for Mechanically Flexible Systems Based on Nonlinear Filtering'', AIAA J. of Guidance, Control and Dynamics, submitted, August 1993.

Larsen, G., Cetinkunt, S., Donmez, A., ``CMAC Based Learning Controller for Precision Servo Control of Machine Tool Axes'', ASME Journal of Dynamic Systems, Measurement, and Control, submitted for publication, July 1993.

Cetinkunt, S., Chiu, H-T., ``Estimation of Modal Parameters of Linear Structural Systems Using Hopfield Neural Networks'', AIAA Journal of Guidance, Control, and Dynamics, accepted for publication, May 1993.

Cetinkunt, S., Chiu, H-T., ``Trainable Neuro-Controllers for Motion Control Systems Involving Mechanical Flexibility'', IEEE Trans. on Aerospace and Electronic Systems, submitted, April 1992.

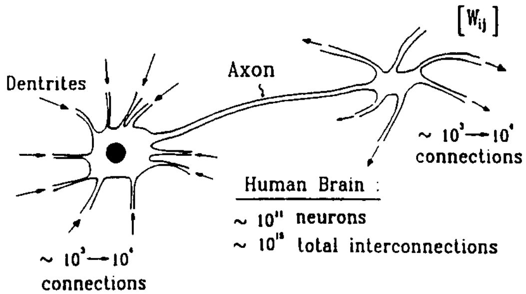
Cetinkunt, S., Lin, H-C., ``Real-time LQ Optimal Control of Structural Systems Using Hopfield Nets'', ASME Winter Annual Meeting, 1993, New Orleans, LA.

Cetinkunt, S., Donmez, A., ``CMAC Based Learning Controller for Servo Control of a Single Point Diamond Turning Machine'', American Control Conference, June 2-4, 1993, San Francisco, CA.

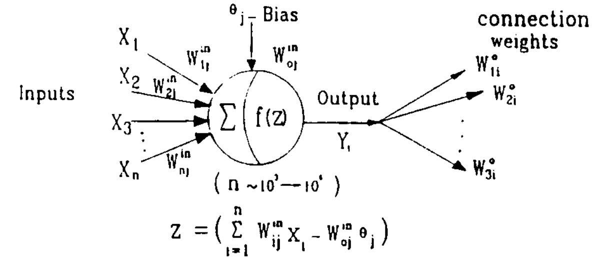
Cetinkunt, S., Chiu, H.-T., ``Estimation of Modal Parameters of Linear Structural Systems Using Hopfield Neural Networks'', ASME Winter Annual Meeting, Nov. 8-13, 1992, Anaheim, CA, DSC-Vol. 45.

Cetinkunt, S., Chiu, H.-T., ``A Study of Learning Controllers for Tip Position Control of a Flexible Arm Using Artificial Neural Networks,'' 8th International Conference on CAD/CAM, Robotics, and Factories of the Future, Metz, France, August 17-99, 1992.

Cetinkunt, S., Chiu, H.-T., ``A Study of Learning Controllers for Tip Position Control of a Flexible Arm Using Artificial Neural Networks'', ASME Winter Annual Meeting, Dec. 1-6, 1991, Atlanta, GA., DSC-Vol. 31, pp. 15-19.



A simplified biological model of a neuron



A simplified mathematical model of a neuron

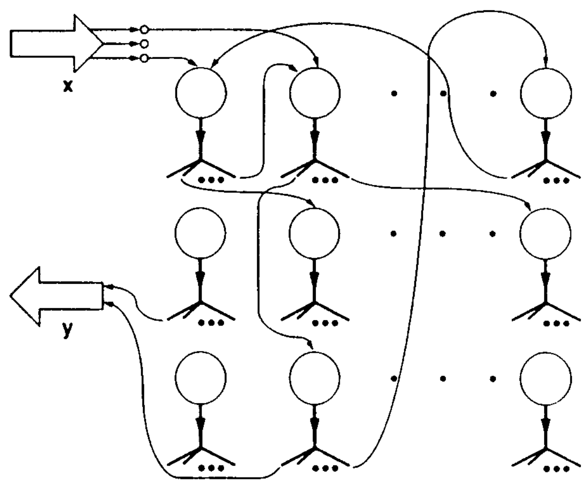
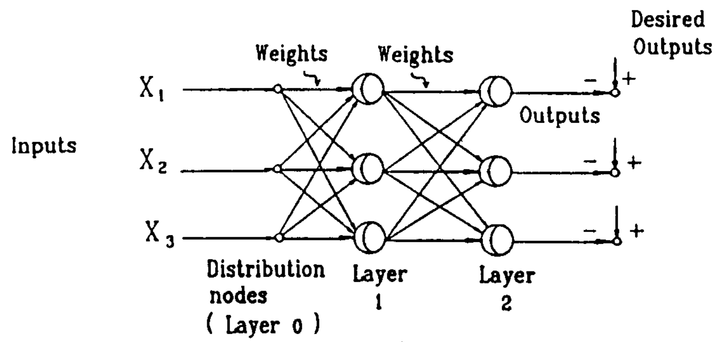


Fig.1 - The basic building block of artificial neural network is a neuron model. Here a neuron model is shown which mimics the neurons of human brain. Neural networks are the result of many massively interconnected neurons.



(a)

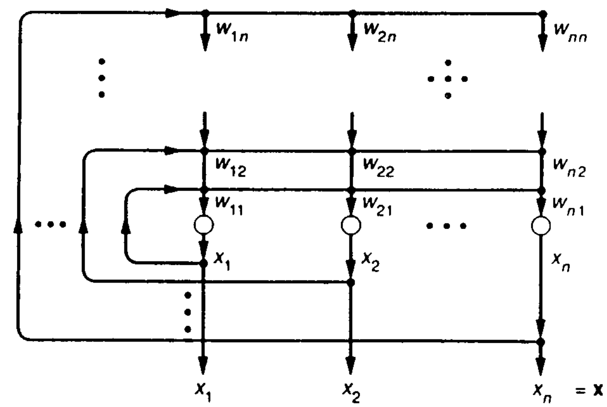


Fig.2 - The interconnection between neurons essentially determines the artificial neural network architecture. a) feedforward artificial neural network architecture, b) recurrent artificial neural network architecture.

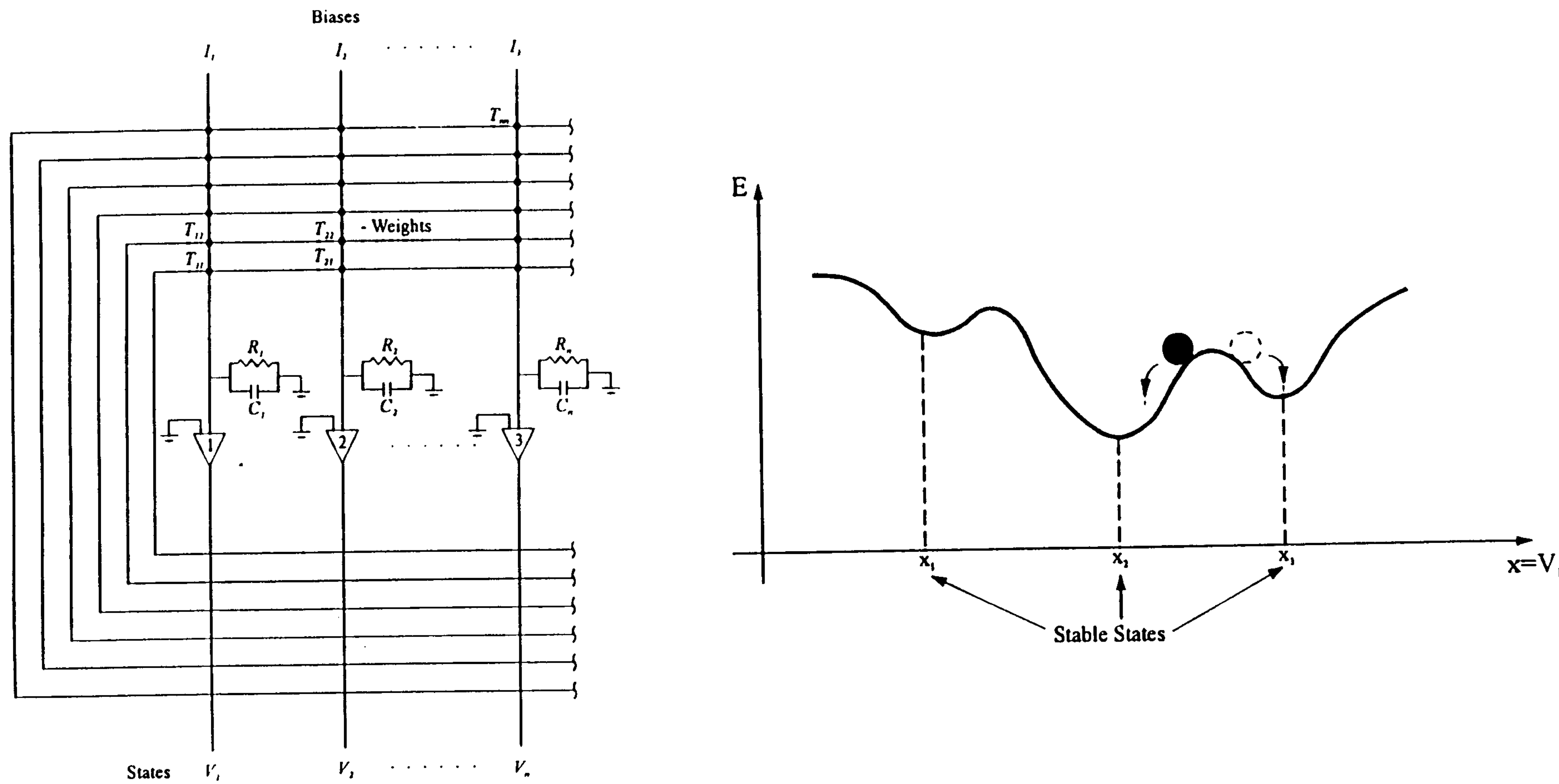


Fig.3 - Hopfield net architecture and its energy function showing stable states. Depending on the initial starting state of the network, the states converge to the closest local minimum of the energy function. Different interconnection weights and biases create different energy surfaces.

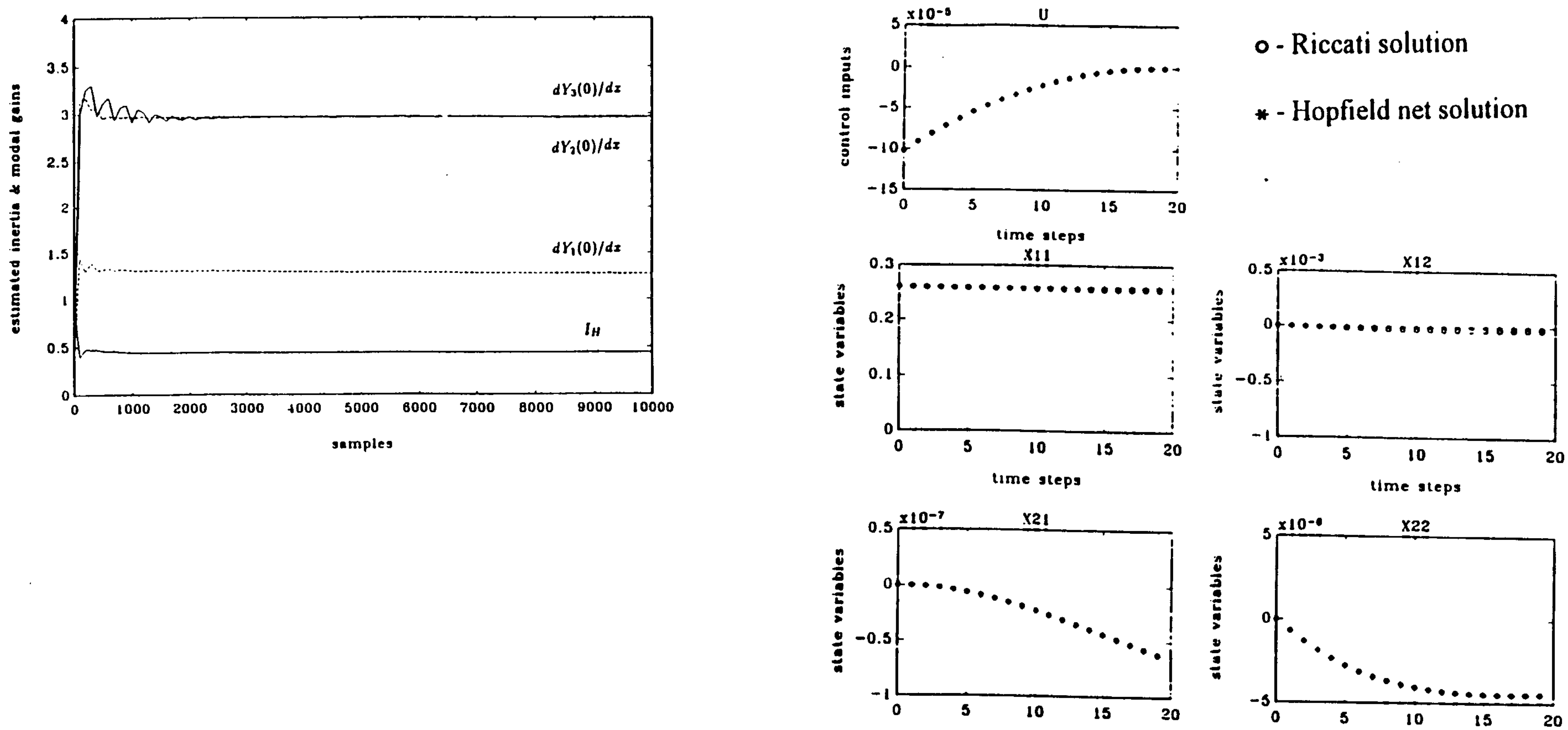
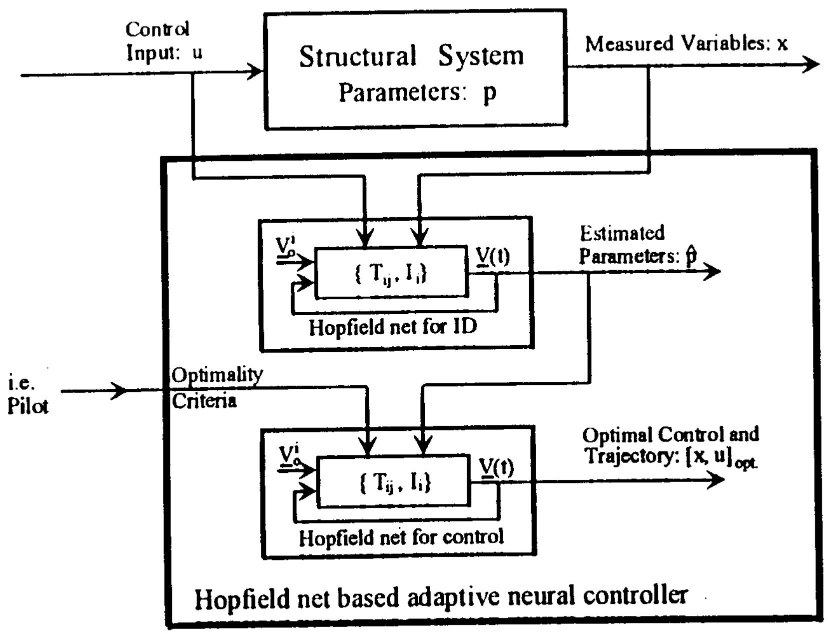


Fig.4 - Samples of results from the application of Hopfield nets to the modal parameter identification problem (figure on the left), and the discrete-time LQ optimal control problem solution (figures on the right).



Adaptive neural control of structural systems using Hopfield nets.

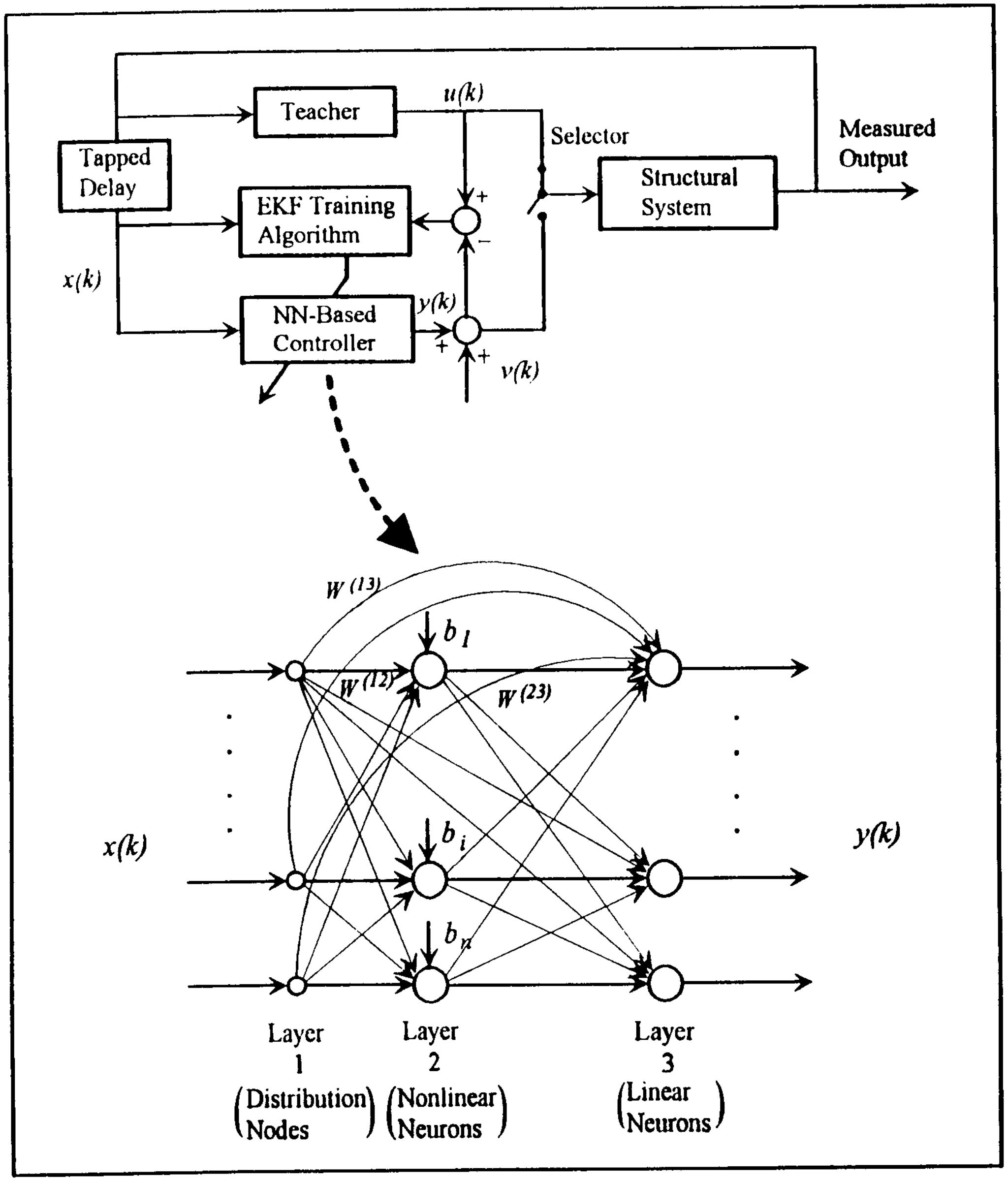
* Insensitive to variations in the structure dynamics details, robust emergent collective convergence propreties of Hopfield net.

* Guarantee to converge to the closest (local) minimum within the order of the time constant of the network.

* Real-time parameter identification and finite time linear quadratic optimal control problem can be solved (in real-time!).

* Potential implementation using analog VLSI and/or optical computers with convergence time in the order of 10 microseconds to 0.1 microsec.

Fig.5 - Real-time adaptive parameter estimation and optimal control solution for structural systems using Hopfield nets.



* **Special class of feedforward neural net :**
  Layer 2 nonlinear neurons, Layer 3 linear neurons

* **Learning argorthm based on nonlinear filtering (Extended Kalman Fiter) increases the convergence rate by 2 orders of magnitude compared to backpropagation algorithm.**

Fig.6 - A trainable neuro-controller architecture for structural systems using feedforward neural nets and EKF learning algorithm. The first layer (distribution nodes) are connected to both second and third layers. The third layer neurons are linear.

Convergence of Learning Algorithm

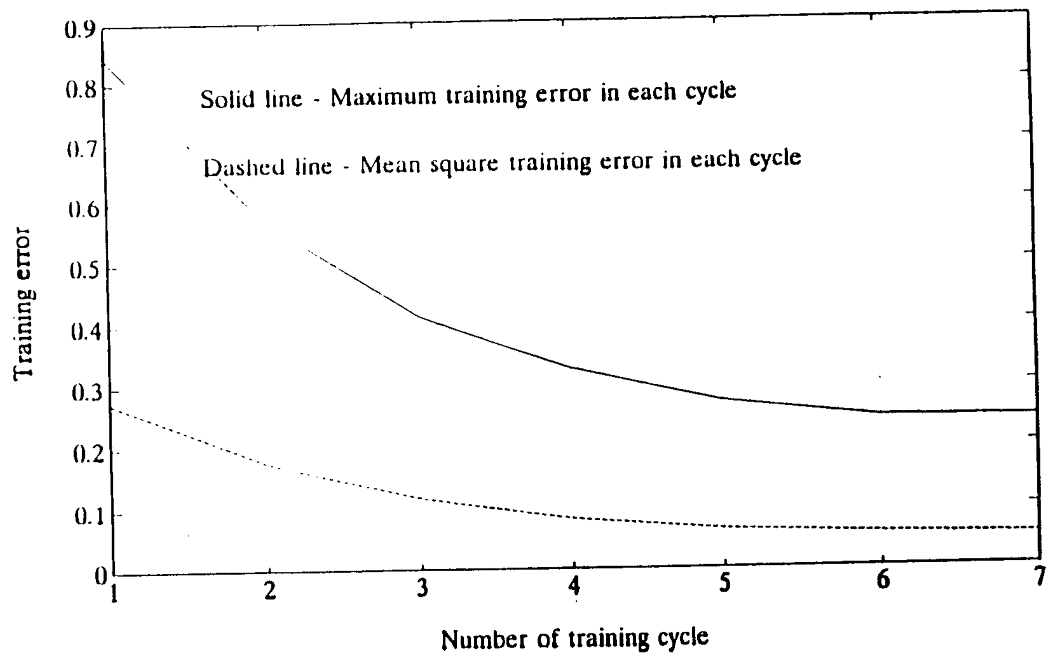
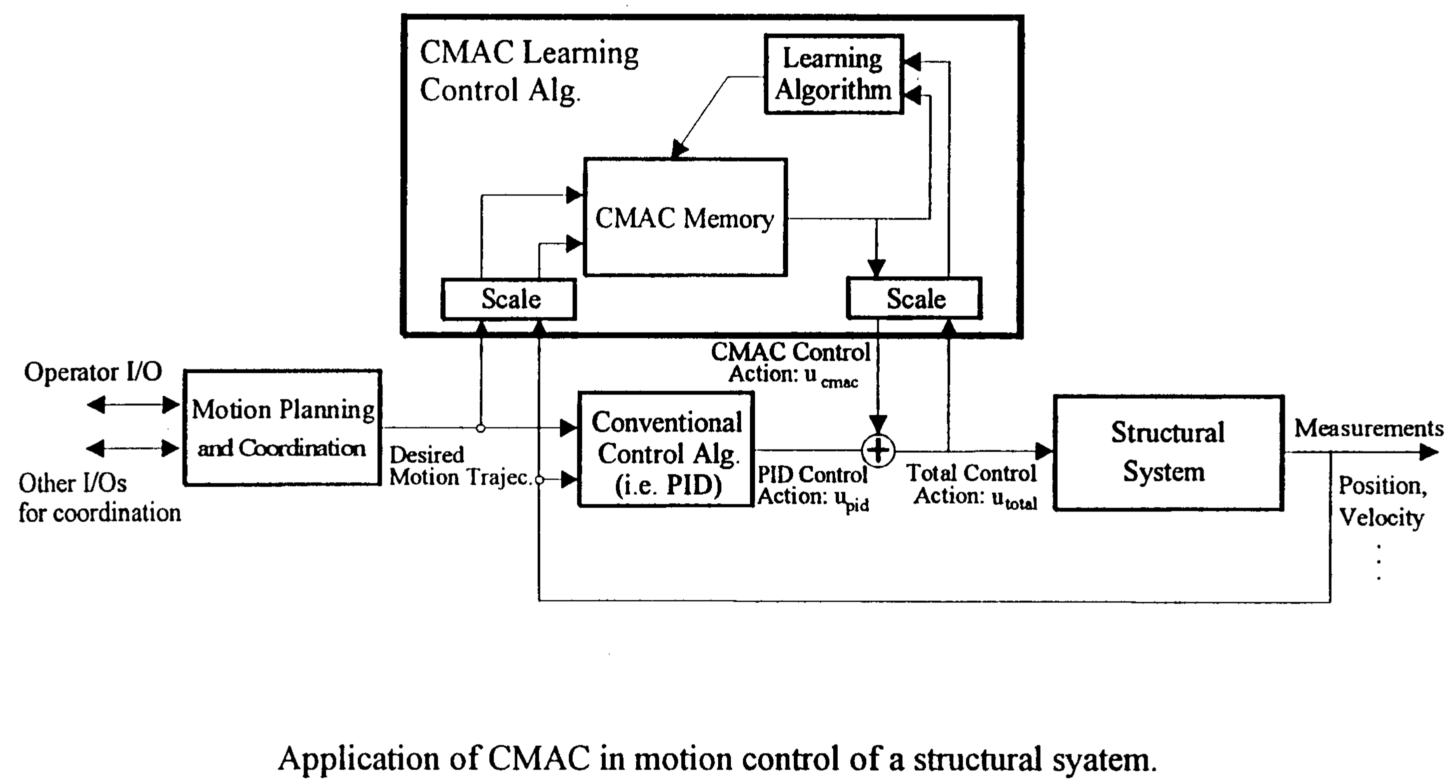
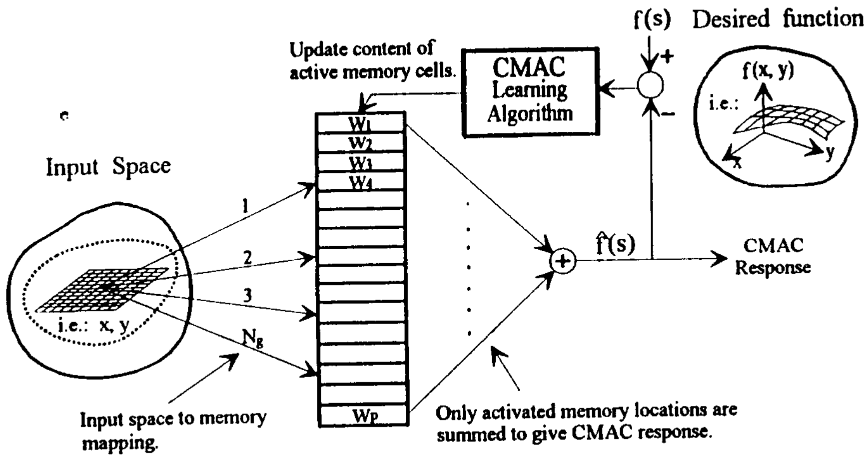


Fig.7 - Convergence of the learning of the neuro-controller shown in Fig. 6. The maximum and mean square of the training error decays exponentially as the number of training cycles increases.

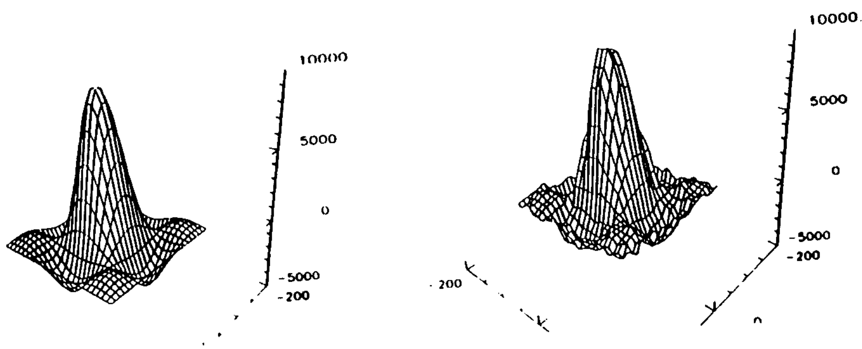


Application of CMAC in motion control of a structural syatem.

Fig.8 - Application of CMAC neural network in motion control of structural systems. The CMAC is a general function approximator with interpolation and learning capability. It is similiar to adaptive look-up table approach. It works in parallel with a PID type controller and eventually takes care of the control decisions as the learning converges.



Block diagram of basic CMAC algorithm: memory size (p), generalization size (Ng), I/O scaling and mapping, learning algorithm, number of I/Os are the CMAC design parameters.

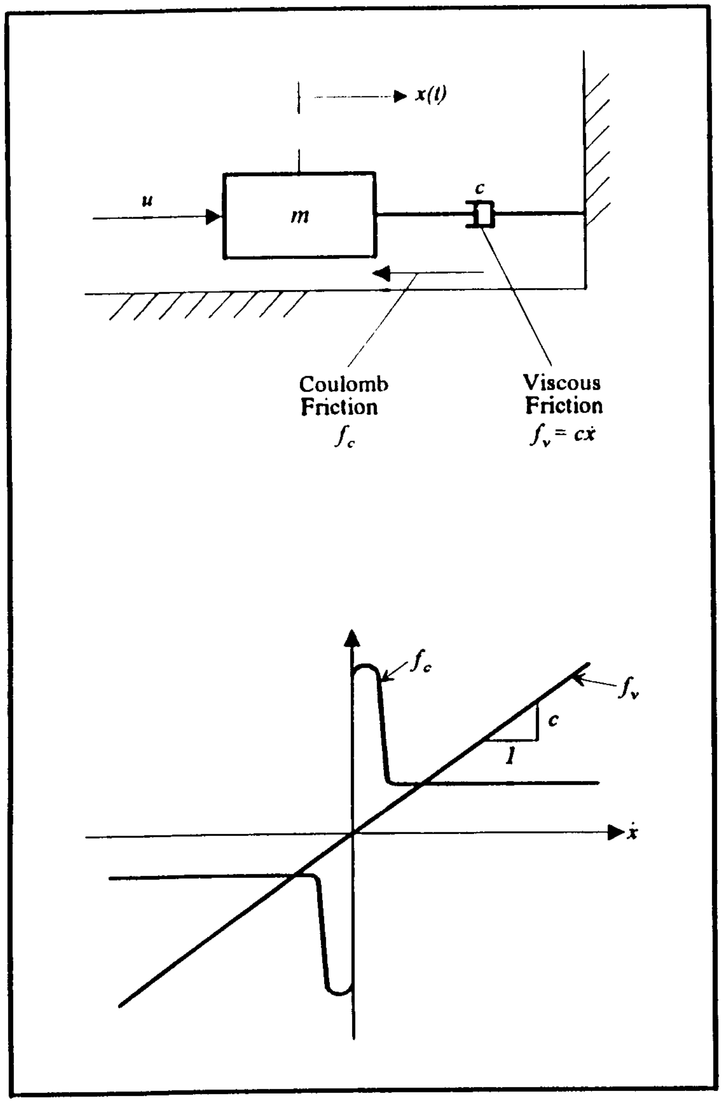


A nonlinear function learned by a CMAC module : nonlinear function is *sin(x)sin(y)/(xy)*. CMAC memory size is 4000, generalization size is 16, learning rate is 0.125.

Fig.9 - a) The CMAC algorithm, and b) its application to a nonlinear function learning problem.

* Distributed storage and processing of information, generalization ability.

* Much faster learning rate (two order of magnitude) compared to backpropagation neural networks, similar rate with EKF based training of FF-ANNs.



Velocity	Maximum position and velocity errors					
	PD Controller		CMAC conected			
			Initial		Trained	
$\mu m/min$	P_{err} (nm)	V_{err} ($\mu m/min$)	P_{err} (nm)	V_{err} ($\mu m/min$)	P_{err} (nm)	V_{err} ($\mu m/min$)
1000	6000.0	721.0	897.0	389.0	1.66	0.99
500	5212.0	414.3	702.3	207.5	0.77	0.89
100	642.9	67.8	103.2	72.3	0.374	0.176
50	*	*	NA	NA	0.283	0.521
20	*	*	NA	NA	0.048	0.231
10	*	*	NA	NA	0.039	0.230
1	*	*	NA	NA	0.033	0.022

*** PD control algorithm not available to start the motion due to large friction**

Fig.10 - Results of motion control accuracy: dynamic system is a mass-force system which has very large stiction and Coulomb friction, motion control at very low speeds is studied. The PID controller alone and CMAC+PID controller approach are compared.