01 Jan 1994

A Comparison of FAM and CMAC for Nonlinear Control

Arit Thammano

Cihan H. Dagli
Missouri University of Science and Technology, dagli@mst.edu

A Comparison of FAM and CMAC for Nonlinear Control

Arit Thammano and Cihan H. Dagli

Department of Engineering Management
University of Missouri-Rolla
Rolla, Mo. 65401

Abstract

In the past, various neural network-based controllers are proposed to master the nonlinear control problems with different level of success. The recent trend is to incorporate fuzzy logic to this process. This article compares a neural network-based controller, both local and global networks, with Fuzzy associative memories (FAM) on a nonlinear problem. CMAC and FAM are chosen as representatives of local generalization networks. CMAC controller is trained off-line, therefore, it can response to the incoming input immediately. CMAC can intrapolate its memory and give a reasonable control signal even the input has not been trained on. Backpropagation is picked as a representative of global generalization networks. All three systems are studied on a simple simulated control problem. This preliminary research will be adapted later to control the laser cutting machine. A performance measure that depends on the transient response and the steady state response of the controlled system is used. The results indicate that CMAC and FAM are comparable.

Introduction

During the last year neural network-based controllers are proposed in literature for modelling nonlinearities inherent on control problems. Various architectures are proposed. Learning algorithms adapted in these architectures depend local and global information captured from the control data. Recent trend is to incorporate fuzzy logic to the process. The quest for robust controller design is still continuing.

Most of the applications in the manufacturing process are nonlinear. However, many previous research show that the classical control concept cannot be used effectively to control nonlinear processes. Nonlinear mapping capabilities of neural networks are used extensively to solve control problems.

Previous research on control examined the use of fuzzy set theory and local networks. However, there are some niches to be penetrated. Some researchers [L. Gordon Kraft and David P. Campagna, 1990], [Lichtenwalner, F. Peter (1993)] had shown the comparison between the neural network controller and the traditional adaptive control system. The former indicates that the neural network performs best when the plant is nonlinear, even it takes quite a long time to learn the process. The latter shows that for the fiber placement composite manufacturing process, the neurocontroller behave like a PI controller when the network receive an input which it has not experienced before. However, after learning from experience, the performance greatly improves and exceeds that of conventional methods. Miller, W. T. et al (1990) reviews the comparison between CMAC and Backpropagation and shows that CMAC can learn a large variety of nonlinear function in a fewer iteration with a little or no learning interference due to recent learning in remote parts of the inputs space. These advantages of CMAC are due to the local generalization at the expense of large memory. Another research [Lin, C. S. and Hyongsuk Kim (1991)] confirms that learned information is distributively stored in adaptive critic learning control and no memory capability is wasted on useless states. The adaptive critic method is a humanlike self-learning scheme that learns performance evaluation as well as control actions based on experience. In adaptive critic method, the user specifies a utility function to be controlled and an acceptable range of system response. An additional neural network, called a critic network, has been adapted to evaluate the progress that the system is making. The output signal of the critic network indicates whether the system status is getting better or worse. Therefore, the action network which outputs the actions to the process is adapted to maximize the utility function of the critic network. Christopher G. Atkeson et al (1990) shows the benefit of using table-based controller to control robots. Because more work is needed in the fuzzy-based controller area, this paper presents the direct comparison between CMAC neural network [Albus, J. S. (1979)] and FAM [Kosko, B. (1992)]. These two methods are of interest because of their powerful architectures. Both of them are local generalizer and behave as associative memory. Therefore, they can learn and response to

0-7803-1896-X/94 $4.00 ©1994 IEEE

the process pretty fast. Some differences between these two are (1) FAM stores rules in its memory and processes the incoming inputs in parallel on real time basis, but CMAC precalculates its look-up table off-line, so it can response to the incoming input immediately. (2) The system input-output characteristics of CMAC are continuous but those of FAM are discrete. This research tries to point out the effects of these differences. The objective of the controller is to minimize the error of the desired variable between the target and the actual and to reduce the rise time to the minimum value. In addition, The Backpropagation neural network, the well-known representative of the global network, is also included to compare the ability of the local and global network on the process which the input patterns are not in the same direction.

In the next section, the two different controllers are explained briefly. More details of these methods can be found in the references. Each of the methods are simulated on the same control process under the same conditions. The result of each are discussed in the sections that follow.

Fuzzy Associative Memories

Proper fuzzy sets are the ones that violate the law of noncontradiction and excluded middle. Fuzzy set theory holds that all things are matters of degree. It reduces black and white logic to the mathematics of gray relationships. The fuzzy power set $F(2^X)$, which contains all fuzzy subsets of X, corresponds to the unit square when $X = \{x_1, x_2\}$. Figure 1 displays the fuzzy power set $F(2^X)$ in 2 dimensional unit hypercube. From Figure 1, the fuzzy subset A corresponds to the fit vector (1/4,3/4), therefore, A has membership degrees $m_A(x_1) = 1/3$ and $m_A(x_2) = 3/4$. The midpoint M of the unit cube has the maximum fuzziness.

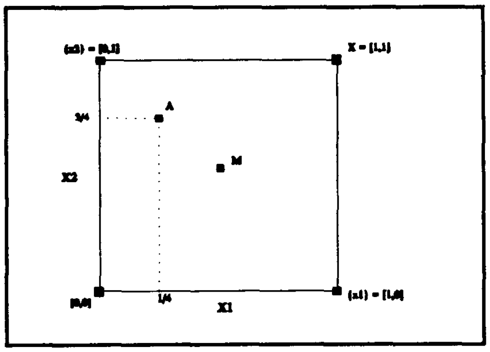

Figure 1

In order to find the FAM bank as shown in Figure 2, since the dynamic equations of the process are known and the process is not complicated, FAM rules need to be identified. FAM bank can be randomly calculated from the process equations of motion. On the other hand, if the process equations of motion are not known, the differential competitive learning (DCL) has to be applied to estimate the FAM bank. The differential competitive learning law [Kosko, Bart (1992)] is shown below :

$$\dot{m}_{ij} = \dot{S}_j(y_j)[S_i(x_i)-m_{ij}]+n_{ij} = y_j[S_j-m_{ij}]-S_j[S_i-m_{ij}]+n_{ij}$$

where y_j is the output signal of the j^{th} neuron. When the j^{th} neuron wins, $y_j = 1$ and it equals to 0 when the j^{th} neuron loses. S_j is the competitive signal, which is between zero and one. m_{ij} is the synaptic weights of the connection matrix M. The ij^{th} synapse is excitatory if $m_{ij} > 0$, inhibitory if $m_{ij} < 0$.

If the j^{th} neuron continues to win, S_j rapidly approaches unity, and learning ceases. The rapid burst of learning as S_j approaches unity helps prevent the j^{th} neuron wining too frequently. If this happens, it prematurely encodes a new synaptic pattern in m_j at the expense of the current m_j pattern. In differential competitive learning, the win signal S_j rapidly stops changing once the j^{th} neuron has secured its competitive victory. Differential competitive learning punishes losing with a sign change (when $y_j(s) = 0$). Then S_j rapidly falls to zero, and learning again ceases. Before S_j reaches zero, the competitive learning law reduces to the anticompetitive law

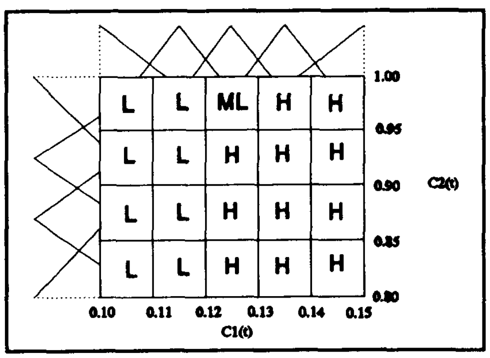

Figure 2

$$\dot{m}_{ij} = -S_j[S_i-m_{ij}]+n_{ij}$$

Note that the input and the output of the FAM system are the fuzzy sets and the output of the FAM, R, equals a weighted sum of the individual vectors R_k':

$$R = \sum_{k=1}^{m} w_k R_k'$$

Because the output of the FAM is also a fuzzy set, therefore, the fuzzy centroid defuzzification scheme is introduced to produce a single numerical output.

$$\textit{Fuzzy centroid } \bar{R} = \sum_{j=1}^{p} y_j m_R(y_j) \;/\; \sum_{j=1}^{p} m_R(y_j)$$

Cerebellar Model Articulation Controller

The basic idea behind the CMAC approach is to create the look-up table from the input-output of the system. Then using the data in the table as feedforward information to calculate the appropriate control signal. In this case, the value of the system parameters were known, hence, all the value in the table were precalculated and stored in the memory. As the input are fed into the controller, CMAC would be able to look up in its memory and provide the appropriate controller output.

The CMAC algorithm maps any input it receives into a set of points in a large conceptual memory in such a way that two inputs that are close in input space will have their points overlap in the memory as shown in Figure 3, with more overlap for closer inputs. If two inputs are far apart in the input space, there will be no overlap in their sets in the memory, and also no generalization. With a built-in local generalization, input vectors that are close in the input space will give outputs that are close, even the input has not been trained on, as long as there has been training in that region of the state-space.

The method which was used in this article to improve the value in memory is through first-order learning law :

$$m_{ij}(k+1) = m_{ij}(k) + \mu[u(k) - m_{ij}(k)]$$

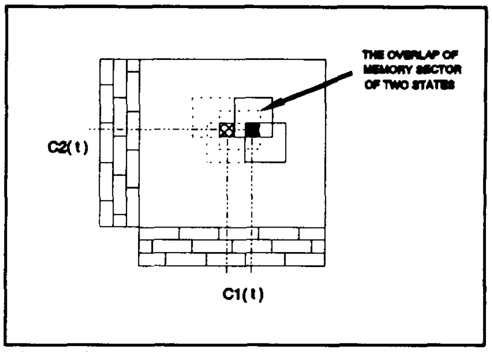

Figure 3

where m(k) is the present value of the memory location, m(k+1) is the updated value, u(k) is the desired output of the controller at time k ,and μ is the learning rate which is between zero and one. If the memory contents m(k) is larger than u(k), then m(k) is corrected by subtracting a number proportional to the error.

The control signal generated by the network is found by summing the values in the system associated with the current inputs. This signal is then fed to the process to maintain the actual output at the target.

Properties of CMAC and FAM

This section illustrates the similarities and differences between CMAC and FAM.

• Both CMAC and FAM are local generalizer. The input vectors that are close in the input space will provide the close outputs.

• Both of them use a look-up table method. Hence, they can be used appropriately to control the process because of their fast response.

• CMAC and FAM have the property that large network can be used and trained in reasonable time. This is because there is a small number of calculations per output.

• The input-output characteristics of CMAC are continuous but those of FAM are discrete. Therefore, CMAC uses more memory than FAM.

• CMAC has to be trained off-line before being used to control the process but FAM calculates the outputs on-line, hence, it uses more time to response. Since there is only a small number of calculations per output, the difference between CMAC and FAM 's processing time is very small.

Sample Process Description

Both control system algorithms were applied to the same process which is the bioreactor containing water, nutrients, and biological cells as shown in Figure 4. This problem has been suggested in *Neural Network for Control* by Anderson, Charles W. et al (1991). The state of this process is characterized by the number of cells and the amount of nutrients. The volume in the tank is maintained at a constant level by removing tank content at a rate equal to the incoming rate. This flow rate is the variable by which the bioreactor is controlled. The objective is to achieve and maintain a desired cell amount, $c_1^*(t)$,by altering the flow rate throughout a learning trial. In this article, $c_1^*(t)$ was setted at 0.1205. The initial conditions $c_1(0)$ is the random variable on the interval [0.10,0.14] and $c_2(0)$ is the random variable on the interval [0.8,1.0]. The system constraints are $0 \le c_1, c_2 \le 1$ and $0 \le r \le 2$. And the process equations of motion are :

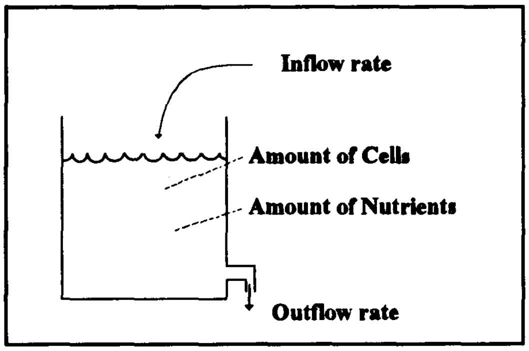

Figure 4 The bioreactor

$$c_1[t+1] = c_1[t] + 0.005(-c_1[t]r[t] + (1-c_2[t])e^{c_2[t]/0.48})$$

where $c_1(t)$: Amount of cells
$c_2(t)$: Amount of Nutrients
r : Flow rate

$$c_2[t+1] = c_2[t] + 0.005(-c_2[t]r[t] + (1-c_2[t])e^{c_2[t]/0.48}\frac{1.02}{1.02-c_2[t]})$$

Results and Conclusions

The result of the CMAC neural network system is shown in Figure 5. This CMAC controller was trained 15 iterations before being used to control the process. The system rise time was quite large but the system offset was very small.

The performance of the FAM method is plotted in Figure 6. The steady state performance was a little bit worse than CMAC but the rise time was very small. With these characteristics of the response, it indicates that FAM works as well as CMAC control system. The steady state response of FAM fluctuates because FAM's system input-output characteristics are discrete but those of the CMAC are continuous. There are the boundaries between each rules of FAM, hence, changing the rule from one to the other is not as smooth as CMAC. However, in the large process, which has many input, FAM is more favorable because FAM needs much smaller memory spaces than CMAC.

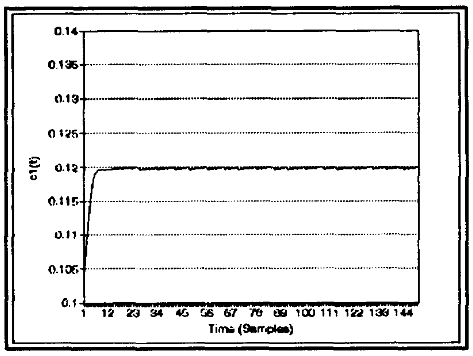

Figure 5

Finally, Backpropagation response, the global generalization neural network, is shown in Figure 7. The figure confirms that the global generalization neural network cannot be used to control the process, which the patterns of the inputs do not go in the same direction.

The CMAC and FAM based control system has been developed and implemented for this bioreator problem. These two methods have been chosen because of their fast learning characteristic and their local generalization background. Both of them give the favorable responses on this nonlinear process. They did track the target very well. CMAC gave slightly better steady state response than FAM. However, DCL will be applied to improve the steady state response of the FAM in the future research. Finally, the research shows that this type of process is not a good application for global generalization network.

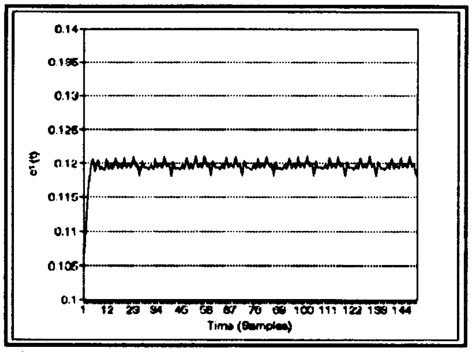

Figure 6

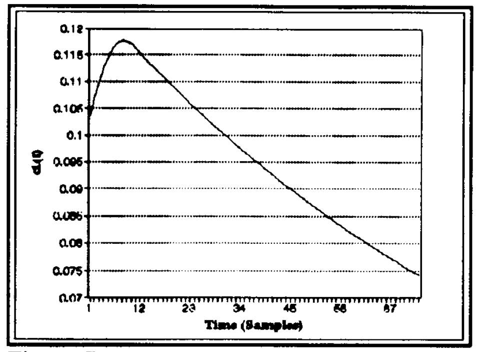

Figure 7

References

Albus, James S. (1979), "Mechanisms of planning and problem solving in the brain," *Mathematical Biosciences*, pp. 247-293.

Anderson, Charles W. and W. Thomas Miller, III (1991), "Challenging control problems," *Neural Networks for Control*, MIT press.

Atkeson, G. Christopher and David J. Reinkensmeyer (1990), "Using associative content-addressable memories to control robot," *Neural Networks for Control*, MIT Press.

Kosko, Bart (1992), *Neural Networks and Fuzzy Systems*, Prentice-Hall, New Jersey.

Kraft, L. Gordon and David P. Campagna (1990), "A comparison between CMAC neural network control and two traditional adaptive control systems," *IEEE Control Systems Magazine*, pp. 26-43, April 1990.

Lichtenwalner, F. Peter (1993), "An on-line learning neurocontroller for the fiber placement composite manufacturing process" in, *Intelligent Engineering Systems through Artificial Neural Networks*, Dagli, C. H., et. al. eds, pp. 529-534, ASME Press, New York.

Lin, Chun-Shin and Hyongsuk Kim (1991), "CMAC-Based adaptive critic self-learning control," *IEEE Transactions on Neural Networks*, pp. 530-532, September 1991.

Miller, W. Thomas, et. al.(1990), "CMAC : An associative neural network alternative to backpropagation," *Proceedings of the IEEE*, pp. 1561-1567, October 1990.

Sun, Kuo-Tung, et. al. (1993), "Sliding-mode-aided fuzzy control system design" in, *Intelligent Engineering Systems through Artificial Neural Networks*, Dagli, C. H., et. al. eds, pp. 517-522, ASME Press, New York.