소뇌모델 선형조합 회로망의 학습능률과 회로망 설계

Learning Performance and Design of Cerebellum Model Linear Associator Network

황 헌*, 백 풍 기**

H.Hwang, P.K.Baek

적 요

시스템의 적응 제어함수를 산출하는 네트워크인 소뇌모델 선형조합 회로망을 이용한 학습제어 기법은 시스템에 영향을 주는 제어인자들의 불확실성 및 모델링의 결여에도 불구하고 오히려 안정한 실시간 제어의 구현을 가능하게 함으로써 대단한 관심을 불러 일으켜 왔다. 그러나, 센서로부터의 정보처리와 인식 그리고 복잡한 비선형 시스템의 제어에 적용하기에는 회로망 자체의 내재적 문제점들이 여전히 남아있다. 소뇌모델 선형조합 회로망을 기지 또는 미지의 시스템 모델에 효과적으로 적용하기 위해서는 네트워크에 영향을 주는 제어인자가 시스템에 미치는 영향을 분석하는 것이 필수적이다.

분할 블럭의 크기, 학습이득, 입력편이 그리고 입력변수들의 영역과 같은 네트 제어인자들은 시스템의 학습 능률 및 소요 기억용량의 크기에 중대한 영향을 미침에도 불구하고 충분히 조사되지 못한 실태이다. 물론 이들 제어인자들의 결정에는 학습 대상이 되는 시스템 함수의 형태와 적용 학습 알고리즘이 반드시 고려되어야 한다.

본 논문에서는 학습 능률성에 미치는 이들 제어인자들의 상호영향도를 저자가 제안하였던 기본 학습 알고리즘에 의거하여 조사하였다. 분석적인 방법만으로 이러한 상호영향성을 조사하기는 매우 힘들거나 거의 불가능하다고 보아지기 때문에 학습 대상함수를 먼저 규정하여 다양한 컴퓨터 모의시험을 수행하였고 그 결과를 분석하였다. 컴퓨터 모의시험의 결과에 의하여 회로망의 시스템 적용시 고려할 설계 지침을 제시하였다.

1. INTRODUCTION

The sensor integrated adaptive controller for complicated systems such as a robot is confined mostly to the experimental stage because it requires heavy computing of real time parameter identification via processing of the sensory information based on some performance criteria and management of sensitivity on sensory readings. It usually involves a complex algorithm. For this reason, the robust adaptive controller based on the biological structure and function have drawn a great attention recently[5]. How to achieve a great degree of the robustness, adaptation, real time control and easy learning is the major focus on this area.

Research and application of the artificial neural net to the robot system control and visual perception have become widely spreaded for a few years around the world with the aim of realizing the structure and function of biological organisms espe-

* 成均館大學校 農業機械工學科

** 建國大學校 農業機械工學科

cially of human brain[4]. Through the massively parallel connection of processing elements with learning capability and fault-tolerance, the neural net approach is known to overcome the limitation and the weakness posed by the conventional sequential information procssing[6].

Some basic principles of how the cerebellum accomplishes motor behavior have been organized into a mathematical model, Cerebellar Model Articulated or Arithmetic Controller(CMAC) by Albus[1-3]. Since then, research on CMAC based general learning controller has been attempted to control various systems including a robot because of the simple structured nature of the net. From the viewpoint of the net characteristics, CMAC is rather named as Cerebellum Model Linear Associator Network(CMLAN).[7].

The learning convergence of the CMLAN was proved by author identifying the network as a kind of one layer linear associator having a linear activation function. Two types of basic learning algorithms of CMLAN, sequential error correction(SEC) and random error correction(REC) under delta rule have been proposed and analyzed with different learning gains[7, 8].

To apply CMLAN to various unmodeled or modeled systems more efficiently, it is necessary to analyze the effects of CMLAN control parameters on the trained net. Parameters such as size of the quantizing block K, learning gain G, input offset, and ranges of input variables play a key role in the learned performance, system memory requirement, and learning speed. However, these have not been fully investigated yet. Values of control parameters are chosen in most cases on an ad hoc basis. Values of parameters should be determined, of course, by also considering the shape of the desired function to be trained and learning algorithms applied.

In this paper, with the predetermined input variable offset and ranges (refer to Hwang and Baek[7] for the offset and uniform quantized scheme handling various input ranges), the interrelation of quantizing size K and learning gain G is investigated with learning performance under two types of learning schemes, SEC and REC. The system memory required for the CMLAN application depends on the quantizing size K and ranges of the input variables.

Since an analytic approach only seems to be very difficult and even impossible for this purpose, various simulations have been performed with quite differenct shaped model functions and their results were analyzed and some of them are presented. A general step following design guide was set up through the characteristics of the CMLAN network analyzed theoretically and experimentally.

2. SIMULATION

2-1. Basic Learning Algorithm

Equivalent learning period

Desired function to be learned:

$P = \sin(x)$

Input range : $x = [0, 360]$(deg)

Interval of sampled node inputs : 5 deg

Selected size of quantizing block :

5, 10, 20, 30, 40, 60

CMLAN offset : $1 = 1$ deg

(1) Batch Sequential Error Correction(SEC)

Learning gains which avoids divergence at the initial training are selected from

0.1~1.0 by 0.1

0.09~0.01 by 0.01

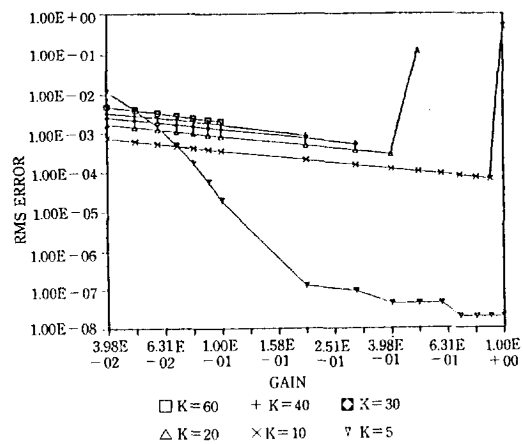
0.009~0.001 by 0.001

Number of training epoch : 0~100

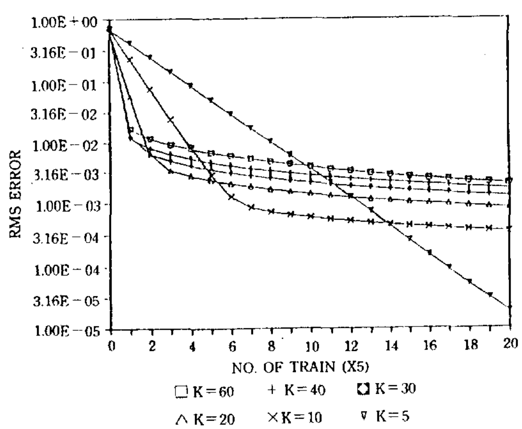
Since the sampled node interval was specified as 5 deg, the case of $K=5$ does not have any interference effect at all generating the orthogonal linear independednt CMLAN mapped binary pattern vectors. As a result, the function is trained completely by one shot when gain is one. It is not shown clearly in Fig. 1a because it was plotted at every training epoch of 100.

As K increases from five, values of the initial gains which avoid the divergence decrease. Gains located closely to the initial divergent gain converge at the early stage but show a diverging trend when the learning epoch increases.

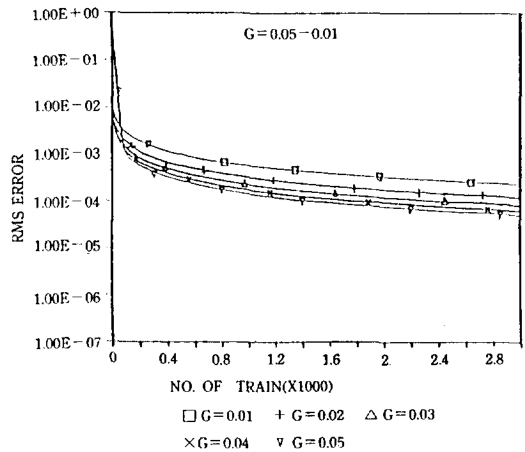
The distributing and interference effect can be seen in Fig. 1b. As values of K increase, the corrected delta values are distributed over the input space broadly resulting the fast RMS error convergence. However, the converging rate weakens earlier than smaller values of K do as learning epoch increases because of the interference.



(a) RMS error vs G with various K (learning epoch=100)



(b) RMS error vs learning epoch with various K (G=0.1)



(c) RMS error vs learning epoch with various G (K=40)

Fig. 1. Batch type SEC learning for P=sin(x)

When K is equal to 5, the converging slope is logarithmically straight because of the orthogonality of the pattern vectors. The slope is getting steep as learning gain increases and becomes infinite when the value of gain is one.

Fig. 1c shows the extended learning epoch up to 3000 with K=40. The parallel trend of the slopes for the various gains is maintained until it reaches its global minimum. It is not plotted but converged RMS errors learned from the unlimited learning epoch with G=0.05 and G=0.01 were 1.39531E-6 at the learning epoch of 19000 and 7.898105E-6 at the learning epoch of 55500 respectively. In case of the batch SEC, it can be seen that with a fixed K the high gain allows the system to converge faster and to reach lower global minimum than the low gain once it is within the converging range.

(2) On-Line Sequential Error Correction (SEC)

Selected learning gain :

0.2~1.0 by 0.2

Number of training epoch : 0~100

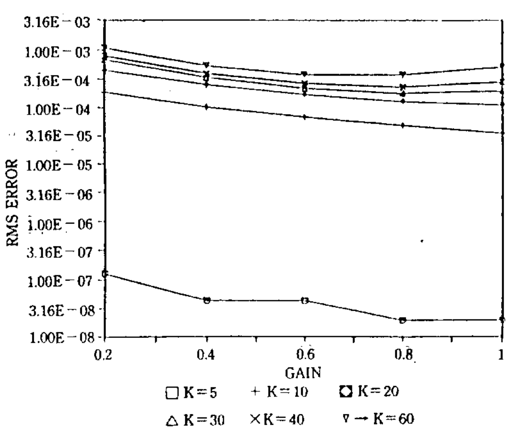
As batch type SEC does, on-line SEC has the same learning effect after one epoch training with

K=5 as shown in Fig. 2a and Fig. 2b. Fig. 2a shows the distributing and interference effect of K at the learning epoch of 100. When the value of K is large, the interference of learning is great with large values of G. However, this fact is rather slowly occurred with small values of K.

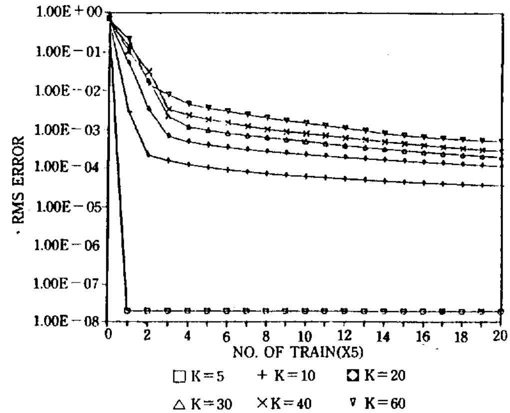
With smaller gains, the learned performance is rather low because of the less distributing effects. From this it can be seen that as K increases the shape of the learned RMS error trend becomes rather a bowl type. This bowl shape is flattened as learning epoch increases more and more.

At the early stage of learning, although it is not shown clearly in Fig. 2b because of the plotting interval of the epochs of learning, the large size of K shows a better learning. It is noted, however, the size of K should be proper to the shape of the function to be trained.

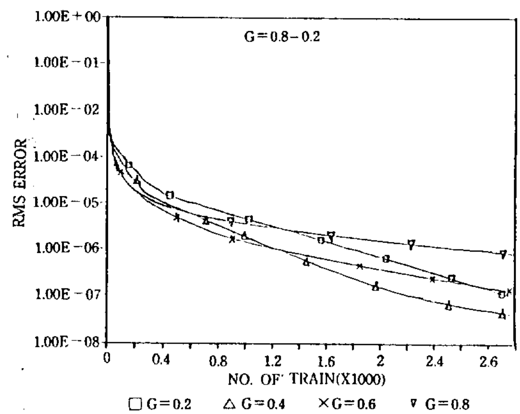
Fig. 2c shows trends of the various gains with a fixed K of 40 when learning epoch is extended up to 3000. It should be noted the smaller gain catches up the larger gain as the learning epoch increases. This is contrary to the result of the batch type SEC. Note, however, practically the learning period is also critical to the system performance as well. For this reason, it is not always recommended to reduce the gain value too small based on the result of Fig.2c when applying on-line SEC learning.



(a) RMS error vs G with various K
(lerning epoch=100)



(b) RMS error vs learning epoch with various K (G=1.0)



(c) RMS error vs learning epoch with various G (K=40)

Fig. 2. On-line type SEC learning for P=sin(x)

(3) Random Error Correction (REC)

Selected learning gain :

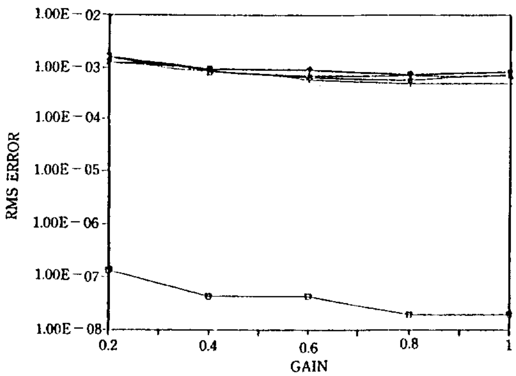
0.2~1.0 by 0.2

Number of training : 0~6000

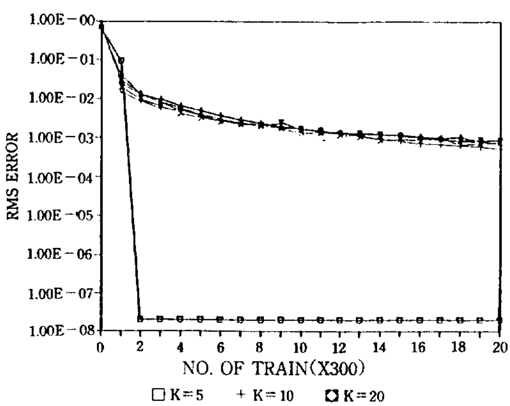
At the first glance, Fig. 3a seems to show a chaotic behavior of trends with respect to G and K. The REC learning has the similar behavior as SEC learning except the nonaccumulating property caused by its random selection of input patterns during each training epoch. However, in general, it has the weakness of a rather large oscillation of the trained performance because under this algorithm it does not consider the history of learnings.

With the relatively small size of K, the interference is not serious as gain increases at the learning number of 6000. The distributing effect is rather good at high gains. However, this effect is interfered by large values of K when the shape of function has varying curvature over K. Trends of the performance from K=10 to K=60 can be explained by the amount of the interaction between the distributing and the interference with various gains at this specific learned point.

Although it is not shown the number of training exactly in Fig. 3b, all sampled node inputs are selected at least once at training number of 403 with K=5. It is due to the characteristics of REC learning which randomly selects inputs among the sampled nodes.



(a) RMS error vs G with various K (learning number=6000)



(b) RMS error vs train number with various K (G=1.0)

Fig. 3. REC learning for P=sin(x)

2-2. Functional Shape

Equivalent learning period

Learning algorithms :

On-line sequential error correction

Random error correction

Selected learning gain :

0.2~0.8 by 0.2

Interval of sampled node inputs : 15 deg

Selected size of quantizing block :

20, 30, 40, 60, 80, 120

CMLAN offset : 1=1deg

(1) Function 1

Desired function to be learned (Fig. 4a) :

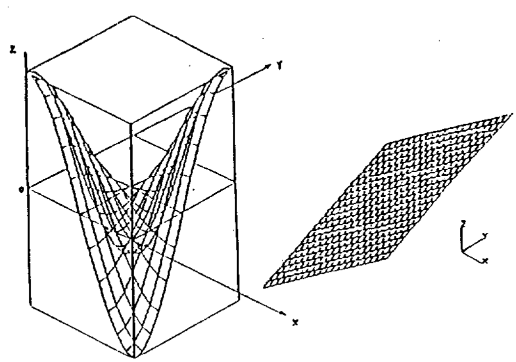
P=cos(x) cos(y)

Input range : x, y=[0, 180] (deg)

◇ On-line SEC :

Trends of learned RMS errors were investigated up to the learning epoch of 200. Fig. 5a shows the RMS error versus learning gain G for various K values at the learning epoch of 200. Fig 5b shows the RMS error versus learning epoch for various K values at G=0.2. Since the value of the sampled node interval is 15, the learned performance is excellent with K=20.

The RMS error does not vary significantly as G varies or learning epoch increases with relatively

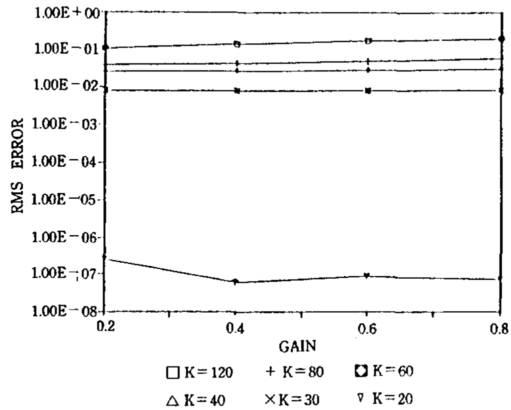


(a) p=cos(x) cos(y)

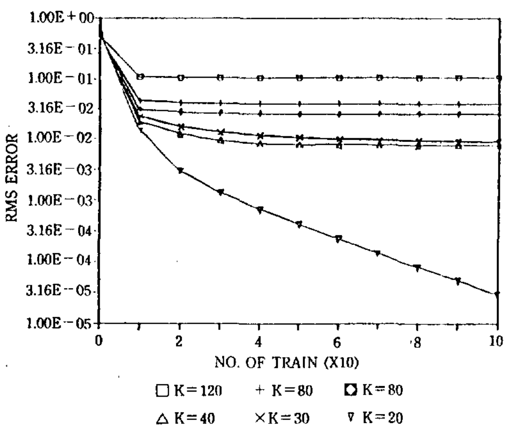
(b) p=10x−2y

Fig. 4. Desired functions to be trained

large K. This is due to the shape of the funtion to be learned. In other words, the function to be trained has a rather steep variation of values over the distributed region defined by K. As K increases, the value of G does not improve the system performance as it is desired. The difference between K=20 and K=30 will be reduced if as the sampled interval is defined less than 15 such as the sampled interval of 5.



(a) RMS error vs G with various K (learning epoch=200)

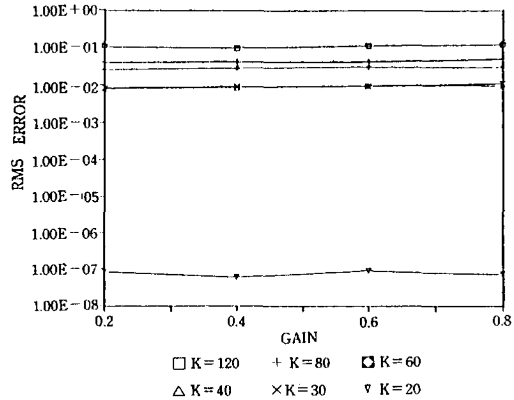


(b) RMS error vs learning epoch with various K (G=0.2)

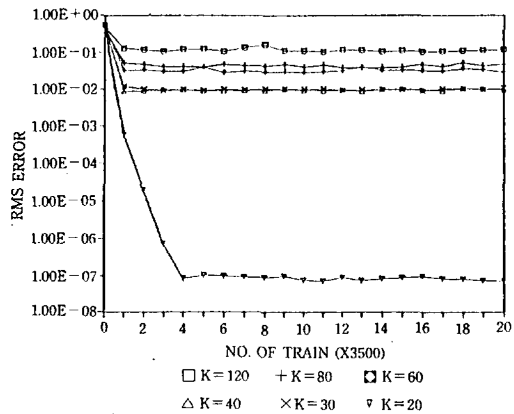
Fig. 5. On-line SEC learning for P=cos(x) cos(y)

◇ REC :

Trends of learned RMS errors were investigated up to the learning number of 70000 which results into the equivalent learning period of on-line SEC. As shown in Fig. 6, except the oscillating behavior of the learned RMS error with an increase of the learning number, the REC learning shows the similar learned performance and pattern of RMS error as on-line SEC does.



(a) RMS error vs G with various K (learning number=70000)



(b) RMS error vs train number with various K (G=0.8)

Fig. 6. REC learning for P=cos(x) cos(y)

(2) Function 2

Desired function to be learned (Fig. 4b) :

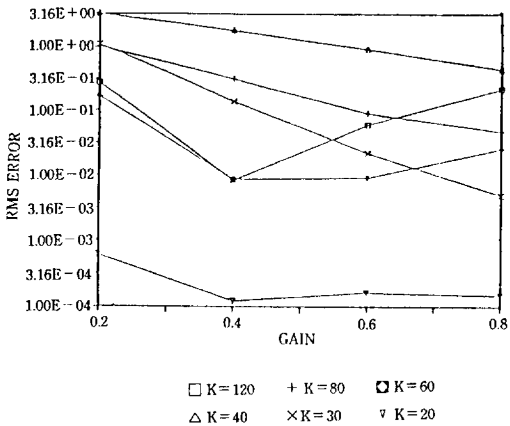
$P=10x-2y$

Input range : x, y=[-150, 150]

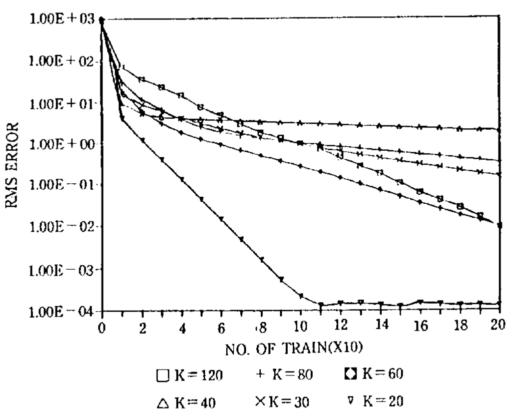
◇ On-line SEC :

Although the overall shape of the desired function is flat, with large value of K the effect of the interference caused by the sequentially accumulated learning error increases when learning gain is high.

With small values of K such as 30 and 40, larger gain shows better performance because of the small distributing effect with little interference as shown in Fig. 7a at the learning epoch of 200. As K increases, the combined effect of the distribution and interference makes the learned system be oscillatory with respect to gain.



(a) RMS error vs G with various K
(lerning epoch=200)

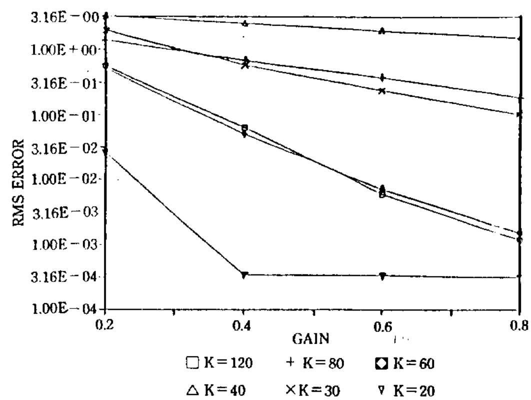


(b) RMS error vs learning epoch with various K
(G=0.4)

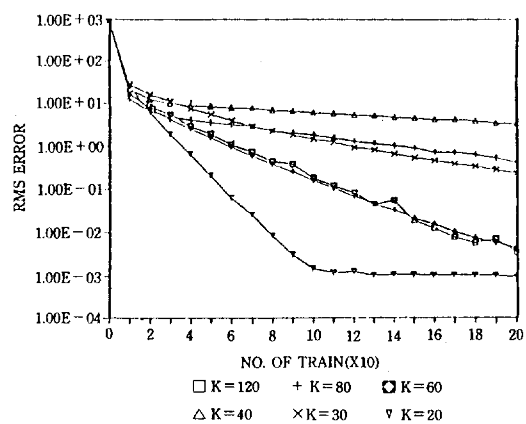
Fig. 7. On-line SEC learning for 10x−2y

It is shown that logarithmically straight variation occurs at K=80 with respect to G. However, the learned performance is quite poor compared to K =60. It is expected the oscillating behavior will occur as learning epoch increases.

Similarly to the case of the function type 1, at K=20 since the interval of the sampled node inputs is 15, the performance is rather excellent compared to other K values. Fig. 7b shows trends of the RMS error versus the learning epoch at G=0.4 with various K values.



(a) RMS error vs G with various K
(learning number=70000)



(b) RMS error vs learning epoch with various K
(G=0.8)

Fig. 8. REC learning for 10x−2y

◇ REC :

Since input nodes are selected randomly and the desired function is also a rather flat shape, in case of the large value of G the correction amount distributed by K is not accumulated much compared to the on-line SEC. Fig. 8a shows logarithmically linear trend of the performance improval as gain increases. Although it is not shown here, it is expected the effect of the gain gets smaller as the number of learning increases. Other trends can be analyzed similarly as on-line SEC. Fig. 8b shows trends of the RMS error versus the learning epoch at G=0.8 with various K values.

3. DESIGN GUIDE

The input offset of CMLAN is related to the continuity property. With a fixed input space as the offset of the quantized block becomes more precise, the mapped non-dimensional CMLAN input space gets bigger. As the CMLAN input space gets bigger, the size of the quantizing block should be properly increased to enlarge the distributing effect of the error correction and to reduce the required system memory. Given sampled node inputs, as the offset between the quantized blocks gets preciser, CMLAN can generate more distinct linear interpolated results at the untrained intermediated nodes.

The size of the quantizing block, K plays a key role in the storage and retrieval of the learned data in a distribute manner. In fact, CMLAN learns the unmodeled system behavior by slicing the desired fuction, which is usually nonlinear, into many precise linear segments. The size of K should be determined considering the slope of the function to be trained. Generally when the slope of the function is steep over the mapped input space, the value of K should decrease and vice versa. With the unproper size of K, the learning performance of CMLAN can not be improved by increasing the training number or by varying the learning gain. With the Nnumber of different inputs, ideally the system can do its best with the N number of memory. The reduction of the memory size, however, is one of the important merits in applying neuro nets while maintaining certain fault-tolerance. Learning gain has a function of the moderate adjustment toward the minimum of the LMS(Least Mean Square) error.

The way of learning is also quite critial to its performance. The designer should decide which learning should fit to their application best among REC, SEC, and hybrid. The REC learning is good for handling a quite large input space and good for off-line generation of the desired system behavior. While the SEC learning is good for the type of applictions such that on-line learning is required while the system is in action with on-line error measurement.

CMLAN can be implemented as a reference function generator or an adaptive control function generator. Considering human's motor behavior, many sub-CMLAN controllers can be connected hierarchically according to the level of the object to be controlled. The selective on-line learning can be implemented depending on whether the situation of the task environment is normal or abnormal.

When applying CMLAN to learn the unmodeled system behavior, the following simple design steps are suggested from the analyzed characteristics of the network.

1. Set CMLAN offset a little preciser than the anticipated.
2. Speify sampled node inputs.
3. Set up regular CMLAN input variable space using the proposed uniform quantizing scheme.
4. Size of K is selected 1/3~2/5 of CMLAN input variable range.
5. Learning gain is selected as 0.4 to 0.8 for the regular REC and SEC training.
6. Number of learning epoch for SEC is deter-

mined via on-line checking of the system improvement at every epoch. In case of REC, checking of the systems improvement is suggested at every 10 times number of the sampled node inputs.

[7] If the learned performance is not good, reduce K by half and increse G by about 0.05~0.1 and go to step [4] and repeat.

[8] Stop

4. CONCLUSION

The interrelation of control parameters specially for the quantizing value K and the learning gain G was investigated by analyzing trained results of the various model functions with basic learning algorithms. A general step following design guide was set up from the characteristic of the CMLAN analyzed theoretically and experimentally.

The CMLAN sytem controller can be extendeded to contol the integrated system behavior employing several sub-CMLANs of different function generator and controller connected each other hierarchically in a closed loop. Research and application of this concept for the task of the sensor integrated robot or system control is widely open.

The development of the nonlinear CMLAN meta connected network for a function generator and a decision controller is suggested for further research to overcome limitations posed by the linearity of the CMLAN network.

5. REFERENCES

1. Albus, J.S. Sept, 1975. A New Approach to Manipulator Control : The Cerebellar Model Articulation Controller(CMAC). Journal of Dynamic Systems, Measurement, and Control, Tran. of the ASME, Vol. 97, No. 3 : 220-227.
2. Albus, J.S. Sept, 1975. Data Storage in the Cerebella Model Articulation Controller(CMAC). Journal of Dynamic Systems, Measurement, and Control, Tran. of the ASME, Vol. 97, No. 3 : 228-233.
3. Albus, J.S. 1979. Mechanisms of Planning and Problem Solving in The Brain, Mathematical Biosciences : 247-291.
4. Anderson, J.A. and E. Rosenfeld. 1988. Neurocomputing : Foundations of research. MIT Press, Cambridge, MA.
5. DARPA Neural Network Study, 1987. AFCEA Int. Press, pp. 123-133, 445-450.
6. Rumelhart, D.E. and J.L. McClelland. 1987. Parallel Distributed Processing : Explorations in the Microstructure of Cognition. Vol. 1 and 2, Cambridge, MA : MIT Press/
7. 황 헌, 백풍기, 1990. 소뇌모델 선형조합 신경망의 구조 및 학습기능 연구(I)－분석 및 학습 알고리즘 개발－, 한국농업기계학회지 Vol, 15 No. 3 : 186-198.
8. 황 헌, 백풍기, 1990. 소뇌모델 선형조합 신경망의 구조 및 학습기능 연구(II)－학습 시뮬레이션 및 응용－, 한국농업기계학회지 Vol, 15 No. 3 : 199-206.