'95 KACC (1995. 10. 23 ~ 25)

An Optimization Method Using Simulated Annealing for Universal Learning Network

Junichi MURATA*, °Akihito TAJIRI*, Kotaro HIRASAWA*, and Masanao OHBAYASHI*
*Department of Electrical Engineering, Kyushu University
6-10-1 Hakozaki, Higashi-ku, Fukuoka 812, Japan
Tel: +81-92-641-1101 (Ext 5315); Fax: +81-92-631-2790;
E-mail: murata@ee.kyushu-u.ac.jp

Abstract A method is presented for optimization of Universal Learning Networks(ULN), where, together with gradient method, Simulated Annealing (SA) is employed to elude local minima. The effectiveness of the method is shown by its application to control of a crane system.

Keywords Universal Learning Networks, Neural Networks, Simulated Annealing, Local Minima, Gradient Method

1. INTRODUCTION

In this paper, a new optimization method for Universal Learning Network (ULN) [1][2]is proposed aiming at elusion of local minima and automatic system design. ULN is based on the idea that most of the large scale complicated systems can be modeled by the network which consists of nonlinearly operated nodes and branches that may have arbitrary time delays. ULN usually requires the optimization by a gradient method, however it has a problem that parameters in ULN are often thrown into local minima. The first aim is to overcome the local minima. The concrete idea to solve the problem is to combine Simulate Annealing [3] with the gradient method.

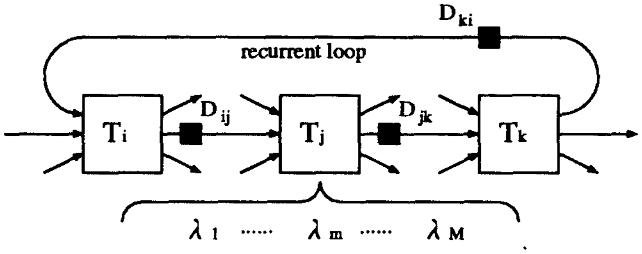
The second aim is automatic system design. It means that, in system design, human tasks based on a priori information should be replaced by the optimization of an appropriate objective function. And thus the system should be designed automatically and appropriately even in the lack of human knowledge.

In the simulation of application to controller design for a nonlinear crane system, the effectiveness of the method in achieving the two aims appears clearly from the comparison between the gradient method and the combined method.

2. UNIVERSAL LEARNING NETWORK

ULN has already been reported before, which can be used as a fundamental tool in modeling and control of large-scale complicated systems such as economic, social and living systems as well as industrial plants.

To be brief, ULN is an extension to a Neural Networks; it consists of various types of nonlinear functional nodes including sigmoid functions, and arbitrary time delay elements including unit time delay between the nodes. ULN has the general structure which covers the recurrent and feedforward Neural Networks, and can represent wider class of systems.



T_i: Nonlinear function node.
D_{ij} : Time delay from T_i node to T_j node.
λ_m : Parameters in ULN.

Fig.1 Structure of Universal Learning Network

3. THE PROPOSED METHOD: COMBINATION SA AND GRADIENT

The searching for the optimal values of parameters in ULN is based on a gradient method. But it has a problem that the obtained solution tends to be thrown into local minima because of the local narrow searching. Therefore this paper makes a proposal for the searching method which overcomes the problem.

The basic idea is to combine Simulated Annealing (SA) with the gradient method. As is generally known, SA is a stochastic searching method and has the advantage of searching a wider area than a gradient method. However, as it deals with discrete problems in general, in order to treat the continuous parameters involved in ULN, it is necessary to add new ideas to SA. They are to set some discrete points for each parameter in ULN. And the parameter takes only the value on the established points. Here we propose the searching procedures as follows:

1. The neighborhood of each solution, which is to be searched for the next solution, consists of some discrete points around the solution.
2. The neighborhood moves as the solution moves so that SA algorithm can search wider region.
3. After obtaining a first approximation of the optimal solution, the same procedure is repeated again with the finer discretization in a restricted region around the approximate solution.
4. And finally the gradient method is employed with the solution given by SA as the initial solution.

The proposed algorithm of the searching can be also applied to Neural Networks.

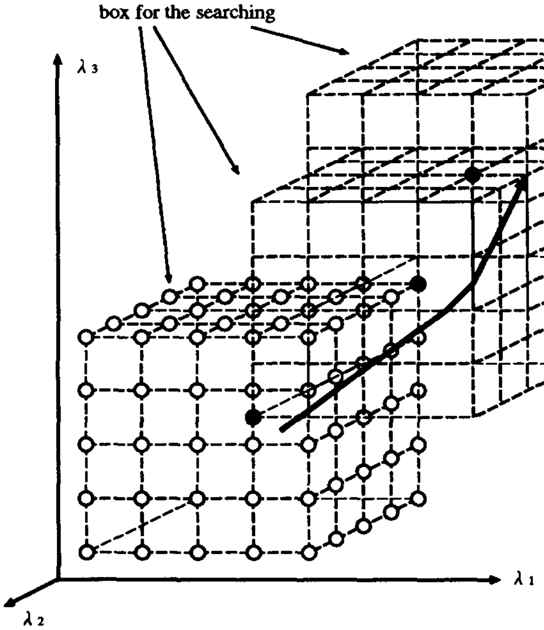


Fig.2 Idea for SA Searching

As an example, Fig.2 shows the case that ULN includes three parameters λ_1, λ_2, λ_3 to be decided. A black circle represents the solution at present. Initially it is given at random before searching. And white discrete circles around the black one are the next candidates for solutions. In advance, the length of a side and the density of the meshes on the box depicted by the dotted lines are fixed according to the problem. And the box itself moves around, which has always the present solution at the center, and is established every time the solution is updated obeying SA rules. Such ideas can make it possible that the limited searching space comes near the infinite space. After SA searching, the gradient method is employed in greater detail for the optimal solution between white circles.

4. NONLINEAR CRANE CONTROL SYSTEM

Simulation studies are carried out to certify the validity of the proposed searching method as well as to evaluate ULN ability in controller design. A crane system is adopted as the plant to be controlled. The crane system and its controller are represented by ULN, and the parameters in the ULN controller are optimized using the proposed method.

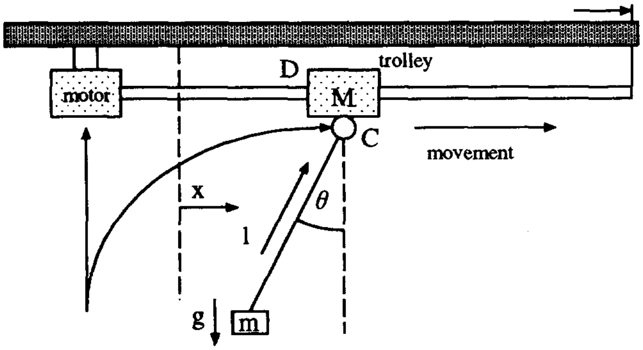


Fig.3 Nonlinear Crane System

Fig.3 shows the nonlinear crane system used in this simulation so as to verify the effectiveness of the proposed method. The notation in it is defined as follows:

g	:	gravitative acceleration
C,D	:	coefficients of the friction
M	:	mass of the trolley
m	:	mass of the load
x	:	location of the trolley
θ	:	angle of the load
l	:	height of the load from initial position

The aim is to bring the trolley to the target position and to winch the load to the target height at the same time without swinging the load hard. The kinetic equations of the crane system are represented in the follows:

$$\frac{d^2x}{dt^2} = -\frac{mg}{M}\theta - \frac{D+G}{M}\frac{dx}{dt} + \frac{G}{M}u_d \tag{1}$$

$$\frac{d^2\theta}{dt^2} = -\frac{M+m}{lM}g\theta - \frac{D+G}{lM}\frac{dx}{dt} + \frac{G}{lM}u_d \tag{2}$$

$$\frac{d^2l}{dt^2} = -\frac{C+G_m}{m} + \frac{G_m}{m}u_m \tag{3}$$

where u_d, u_m are control inputs from the controller. G is the coefficient for converting voltage to driving power.

Then to make a network of the system, the variables x,θ, l are replaced as follows:

$$\begin{aligned} h(T_1,t) &= x \\ h(T_2,t) &= \tfrac{dx}{dt} \\ h(T_3,t) &= \theta \\ h(T_4,t) &= \tfrac{d\theta}{dt} \\ h(T_5,t) &= l \\ h(T_6,t) &= \tfrac{dl}{dt} \end{aligned} \tag{4}$$

where $h(T_i,t)$ means output from T_i node at time t. And introducing ΔT as the sampling time, we get the following difference equations which govern the ULN describing the crane system

$$h(T_1,t) = a_{11}h(T_1,t-1) + a_{21}h(T_2,t-1) \tag{5}$$

$$h(T_2,t) = a_{22}h(T_2,t-1)+a_{32}h(T_3,t-1)+b_1u_d(t) \tag{6}$$

$$h(T_3,t) = a_{33}h(T_3,t-1) + a_{43}h(T_4,t-1) \tag{7}$$

$$\begin{aligned} h(T_4,t) &= \tfrac{a_{24}}{h(T_5,t-1)}h(T_2,t-1) \\ &+\tfrac{a_{34}}{h(T_5,t-1)}h(T_3,t-1) \\ &+a_{44}h(T_4,t-1) \\ &+\tfrac{b_1}{h(T_5,t-1)}u_d(t) \end{aligned} \tag{8}$$

$$h(T_5,t) = a_{55}h(T_5,t-1) + a_{65}h(T_6,t-1) \tag{9}$$

$$h(T_6,t) = a_{66}h(T_6,t-1) + b_2u_m(t) \tag{10}$$

where the coefficients $a_{11},..., b_1, b_2$ are easily calculated from coefficients in equations (1)-(3).

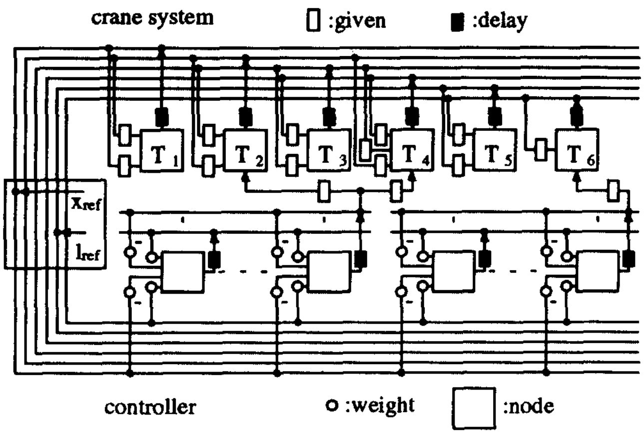


Fig.4 Network of Crane Control System

Fig.4 represents the network of the crane control system schematically. The upper half of the figure is crane system, including 6 nodes, the lower part is controller nodes. The small white circles are weight parameters which are tuned by the proposed method so as to minimize a performance index of control in this simulation. The index, which is called evaluation function in this paper, is based on square errors as follows:

$$\begin{aligned} E &= E_d + E_m \\ E_d &= \tfrac{1}{2}\sum_s\{Q_{11}(x(s) - x_{ref})^2 \\ &+Q_{13}(\theta(s))^2 + Q_{14}(\tfrac{d\theta(s)}{ds})^2 \\ &+R_1(u_d(s))^2\} + \tfrac{1}{2}Q_{12}(\tfrac{dx(t_f)}{ds})^2 \\ E_m &= \tfrac{1}{2}\sum_s\{Q_{21}(l(s) - l_{ref})^2 \\ &+R_2(u_m(s))^2\} + \tfrac{1}{2}Q_{22}(\tfrac{dl(t_f)}{ds})^2 \end{aligned} \tag{11}$$

where Q,R are weight coefficients. If it is necessary to emphasis some of the terms in E, the weight coefficients are given bigger value than the others. Simbols x_{ref},l_{ref} are reference values, and s is time, especially t_f means the final time when the control ends.

5. SIMULATION

5.1 Computer Simulation 1

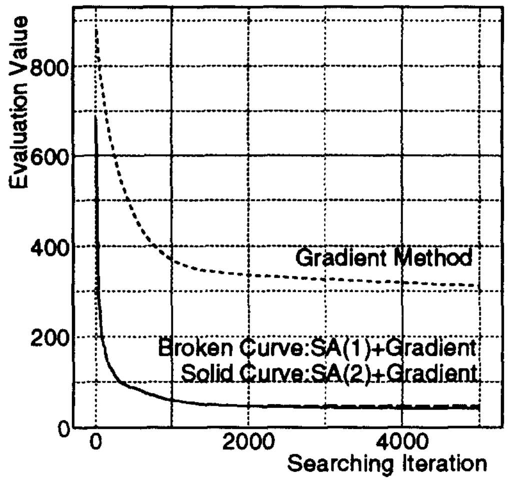


Fig.5 Results from Different Optimization Methods

In this simulation, the two methods are compared: just only gradient method; the SA-gradient-combined method. Fig.5 shows the changes in evaluations as the searching proceeds. Three curves in the figure show the results of different methods, all of which were carried out 5,000 iterations. The solid curve shows the best performance. The method comprises of 2,000 iterations of SA, 2,000 iterations of finer SA and 1,000 iterations of the gradient method. The broken curve has the second best performance. The method comprises of 2,000 iterations of SA, 3,000 iterations of the gradient method.

On the other hand, the dotted curve represents the result obtained by just only using the gradient method, however the curve shows poorer result than the combined method. Even though the number of iterations was extended to 250,000, the gradient method could not give the evaluation value less than 100. The reason might be that the obtained solution fell into a local minimum or that the updating the solution almost stopped because of too small gradient. After all, the proposed method succeeds in achieving the first aim, in other words overcoming local minima.

5.2 Computer Simulation 2

In designing the control system, there are some available information usually. Considering them, it is possible to make the system easily.

It is obvious that, for the crane control system, the controller must output zero signal when the error between the controlled outputs and their reference values are zero. Therefore the system obtains good tuned parameters, provided that the threshold terms are removed from the controller nodes and that the error signals are fed to those nodes. An excellent optimization method can be expected to design a comparably good control system without using such beforehand available information. In order to compare these two cases in this simulation, the design of the crane control system with the thresholds in nodes but without the errors, is practiced on purpose.

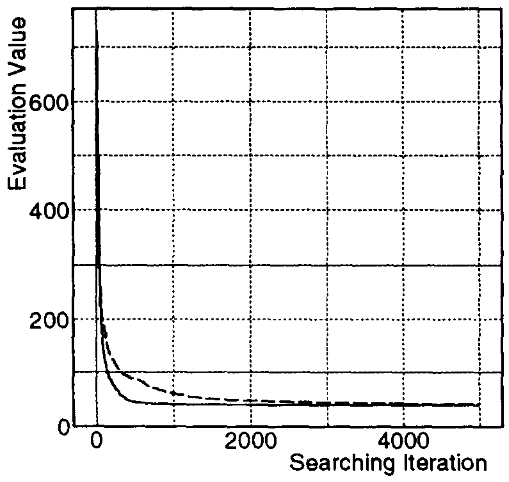


Fig.6 Results with/without Beforehand Available Information

Fig.6 is the compared simulation graph: the solid curve is the result of the case with the beforehand available information; the broken curve is the result without them. According to Fig.6, it is said that after searching the broken curve could catch up with the solid curve till the nearly same level. In other words, it is said that the second aim is achieved.

6. CONCLUSION

The purpose is to minimize the performance index of control, and the optimization for ULN is carried out by tuning the parameters. As a result of the simulation, searching wider area comes true by the SA-gradient-combined method. Therefore, the aim of overcoming local minima is achieved.

The second aim is automatic designing the control system, namely the optimization by exploiting the searching ability fully with very few artificial intervention. Even without human knowledge the system is optimized till the nearly same level of the system based on the knowledge. After all, it can be said that the optimization method proposed here is surely in the way to the ideal system designing that is perfectly automatic designing.

REFERENCES

[1] K.Hirasawa et al, "Universal Learning Network and Computation of its Higher Order Derivatives," Memoirs of the Faculty of Engineering, Kyushu University, Vol. 55, No.2, June 1995.

[2] K.Hirasawa et al, "Universal Learning Network and Computation of its Higher Order Derivatives," IEEE, International Conference on Neural Networks(ICNN'95), 1995.

[3] Kirkpatrick,S., Gelatt,C,D. and Vecchi,M.P., "Optimization by simulated annealing," Science, 220, 4598, pp.671-680, 1983.