'94 KACC (1994. 10. 17 ~ 20)

Genetic Algorithms for Neural Network Control Systems

° Il-Kwon Jeong, Ju-Jang Lee

Department of Electrical Engineering
Korea Advanced Institute of Science and Technology
373-1 Kusong-dong Yusong-gu Taejon 305-701 Korea
Fax : 82-42-869-3410 E-mail : jik@odyssey.kaist.ac.kr

ABSTRACT

We show an application of a genetic algorithm to control systems including neural networks. Genetic algorithms are getting more popular nowadays because of their simplicity and robustness. Genetic algorithms are global search techniques for optimization and many other problems. A feed-forward neural network which is widely used in control applications usually learns by error back propagation algorithm(EBP). But, when there exist certain constraints, EBP can not be applied. We apply a modified genetic algorithm to such a case. We show simulation examples of two cart-pole nonlinear systems: single pole and double pole.

1. INTRODUCTION

Control of a certain system can be represented as finding an appropriate input for the desired output response. For that purpose, we generally need a model describing the characteristics of the system, but it is difficult to get such a model when the system is unknown or time-varying. Adaptive control and variable structure control have been studied to overcome the uncertainties of the model. However, it is hard to apply them to an unknwon complicated system, since both of them have some constraints.

Genetic algorithms(GA's) are used in various control system problems nowadays. A GA is a search method based on the natural selection and genetics while neural networks and fuzzy theory originate from human information processing and inference procedures [1][2]. The current main search methods assume the smooth search space and the existence of its derivative, and most of them are using the gradient following technique.

GA is different from conventional optimization methods in several ways. GA is a parallel and global search that searches multiple points so it is more likely to get the global solution. It makes no assumption on the search space, so it is simple and can be applied to various problems. In control area, it has been used in identification, adaptation of fuzzy membership functions and neural network controller[3][4][5]. However, GA is inherently slow and not good at fine tuning of the solutions.

On the other hand, neural networks have advantages of learning and various input-output mapping capability. There are many paradigms for neural networks, and feed-forward networks are frequently used for complex systems modeling and control[6][7]. Back propagation(BP) learning algorithm for feed forward networks has the problem of local minima and parameter sensitivity[8]. GA is a suitable learning algorithm for neural networks, since it does not have those problems. However, as menthiond earlier, GA is slow and poor at fine tuning. Therefore, GA-BP, a hybrid GA merged with BP was proposed[9], but was not desirable due to the constraint imposed by BP.

In this paper, we propose a modified GA which is designed to improve the speed of convergence and the hill-climbing capability. We apply the proposed algorithm to optimization of a neural network. In section 2, general feature of GA is described. In section 3, a modified GA is proposed. In section 4, cart-pole problems are described, and simulation results are included in section 5.

2. GENETIC ALGORITHMS

GA is a search method based on the natural selection and genetics. The central theme of the research on GA's has been the robustness, the balance between the efficiency and the efficacy necessary for

survival in many different environments. GA is computationally simple yet powerful and it is not limited by assumptions about the search space.

If we want more humanlike optimization tools, the most important goal of optimization should be improvement. Although GA can not guarantee that the solution will converge to the optimum, it tries to find the optimum, that is, it works for the improvement. GA's are diffrent from normal search procedures in four ways.

1. GA's work with a coding of the parameter set.
2. GA's search from a population of points.
3. GA's use objective function information, not derivatives or other auxiliary knowledge.
4. GA's use probabilistic transition rules.

A simple GA is really easy to use, yet powerful. GA searches the solution by transforming the individuals in a population using genetic operators and determining the population for the next generation. In a simple GA, following three basic genetic operators are used.

Reproduction : Reproduction probability is proportional to the fitness value(objective function value) of a string(individual).

Crossover : Crossover needs mating of two individuals. The information of two randomly selected individuals is partly interchanged according to the crossover site. Crossover is applied to take valuable information from the parents, and it is applied with the crossover probability.

Mutation : This operator insures against a bit loss and can be a source of new bits. Since mutation is a random walk through the string space, it must be used sparingly.

There are three differences of GA from random search. First, the existence of the direction of search due to the selection probability. Second, the fact that the better strings make more offsprings and finally, being likely to be improved in average fitness after generations.

3. A MODIFIED GENETIC ALGORITHM

GA is a very useful algorithm because of its versatility. However, it has three major limitations. First, the performance is degraded as the problem size grows. Secondly, premature convergence occurs when GA can not find the optimal solution due to loss of some important characters(genes) in strings. The reason is that GA heavily depends on crossover and the mutation probability is generally too small to move the search to other space. Lastly, GA lacks hill-climbing capability. The reason is also that the mutation probability is much smaller than the crossover probability.

To prevent the premature convergence and to improve hill-climbing capability, we suggest a modified mutation operator as following.

The modified mutation probability, p_m is given as

$$p_m(i+1) = \begin{cases} p_{m0}, & \text{if } \textit{the fittest is the same for } N_{reset} \textit{ generations} \\ p_m(i), & \text{if } p_m(i) \le p_{m\_low} \\ p_m(i)*k, & \textit{other case} \end{cases} \quad (1)$$

where i is the generation number, p_{m_low} and k is a positive constant less than 1.

Thus, the adaptive mutation probability can be enlarged whenever needed. The enlarged mutation probability increases the diversity of the population so it prevents the population from premature convergence. When the mutation probability is its lowest value p_{m_low} it operates as a normal mutation operator.

In this paper, we apply GA to optimize a neural network. As mentioned earlier, GA is suitable for a learning method of neural networks when conventional method can not work. To apply GA to neural networks, we should be able to code the information of a neural network. Since we only deal with the optimization of weights here, all the information of a neural network can be represented as an array of weight values. Coding might be done by an array of the weights' real values. Because binary coding needs too many bits for a string, neural network representation in this paper uses the string which consists of integers with constant bound. Since the true weight values are real, we scale the integer to get a real value. Mutation is defined as a change in the value of an integer.

4. CART-POLE PROBLEMS

Pole balancing problem is difficult since the system is nonlinear. The problem becomes harder when we do not have any *a priori* information about the cart-pole system. Two cases of this class of problems are considered in this paper. The first is the balancing of a pole on a cart(referred to as the single pole problem) and the second is the balancing of two poles on a cart(referred to as the double pole problem). The double pole problem is more difficult than the single pole problem. Fig. 1 and Fig. 2 show the single pole and the double pole repectively.

The equations of motions for the single pole problem can be found in many papers. Those are

$$\ddot{\theta} = \frac{(M+m)g\sin\theta - \cos\theta[u + ml\dot{\theta}^2\sin\theta]}{(4/3)(M+m)l - ml(\cos\theta)^2} \quad (2)$$

$$\ddot{x} = \frac{u + ml[\dot{\theta}^2 \sin\theta - \ddot{\theta}\cos\theta]}{M+m} \quad (3)$$

where M is the mass of the cart and m is the mass of the pole. l is the half of the pole length.

The equations of motions for the double pole system can be derived from dynamic equilibrium equations. The equations are given as

$$\ddot{x} = (F + \frac{3}{4}[m_1 \sin\theta_1 (l_1 \dot{\theta}_1^2 - g\cos\theta_1) - m_2 \sin\theta_2 (l_2 \dot{\theta}_2^2 - g\cos\theta_2)]) / [M + m_1(1 - \frac{3}{4}\cos^2\theta_1) + m_2(1 - \frac{3}{4}\cos^2\theta_2)] \quad (4)$$

$$\ddot{\theta}_1 = [\frac{3}{4}(g\sin\theta_1 - \ddot{x}\cos\theta_1)] / l_1 \quad (5)$$

$$\ddot{\theta}_2 = [\frac{3}{4}(g\sin\theta_2 + \ddot{x}\cos\theta_2)] / l_2 \quad (6)$$

where the following notation is used.

l_1	Half length of long pole
l_2	Half length of short pole
m_1	Mass of long pole
m_2	Mass of short pole
x	Cart position
$\dot{x}$	Cart velocity
θ_1	Angle of long pole
θ_2	Angle of short pole
$\dot{\theta}_1$	Angular velocity of long pole
$\dot{\theta}_2$	Angular velocity of short pole

The control objective is to balance the system such that the poles do not fall beyond a predefined vertical angle(15^o) and cart remains within the bounds of the horizontal track(± 2.4 meters from the center). It is well known for the double pole system that the poles must be of different length in order to be balanced. In this problem, the size of the region of controllability is determined by the ratio of the natural frequencies of the poles, i.e., the ratio of the lengths of the poles.

5. SIMULATION RESULTS

The equations of motions for both the single pole and the double pole problems are simulated using a Euler integration method with a step size of 0.01s. A population of neural networks is randomly initialized with each set of weights and biases. The learning process starts by providing each of the neural networks in the initial population with the initial state of the cart-pole system to be controlled and the network's output response is applied as a force to the simulated cart-pole system. A feed-forward neural network is fed with the four states or six states of the single pole system or the double pole system respectively. The simulation is done with $M = 1.0\ kg$, $m = m_1 = 0.1\ kg$, $m_2 = 0.01\ kg$, $l = l_1 = 0.5\ m$ and $l_2 = 0.05\ m$. Fig. 3 shows the overall system block diagram.

Feed forward neural networks with four and six nodes in the input layer(corresponding to the states), ten nodes in the hidden layer and a single node in the output layer is used. The bias nodes in the input and hidden layer are set to 1.0. The activation of the nodes is given by

$$f(x) = -1 + 2/(1 + e^{-x}) \quad (7)$$

The output of the neural network is amplified by a factor of 10 before it is applied to the simulated cart-pole system. The output of the neural network continuously varies between $-10N$ and $+10N$, as opposed to the bang-bang approach used in other studies.

The coding for the weights and biases of the neural network is a vector of integers between -100 and 100. The real values of the weights and biases are between -10 and 10 so the resolution is 0.1. Because of the integer representation, the problem becomes a kind of combinatorial optimization. So, a conventional method like EBP can not be applied. A population of 100 networks was used. GA parameters are as following. $p_c = 0.8$, $p_{m0} = 0.5$, $p_{m-low} = 0.03$, $N_{reset} = 5$ and $k = 0.9$.

The fitness measure for a network is the simulated time until failure occurs. No other information was used for GA. GA was able to discover a good control strategy which was successful for 100,000 time steps in about 200 and about 650 generations for the single-pole and double-pole respectively.

Fig. 4 shows the result of the single pole system. Initial condition for θ is 0.1 *rad* and all other states are zeros. Fig. 5 shows the result of the double pole system. θ_1 is initially 2.5^o and other states are zeros. In order to recover from this starting position, a force must be applied such that the longer pole tilts further right until the faster swinging shorther pole is tilted to the right more than the longer one, which can be happen because the angular acceleration is inversely proportional to the length of the pole. Once the shorter pole has tilted sufficiently more than the longer pole, an opposite force is applied and both the poles are swung to the vertical position together.

6. CONCLUSION

We have proposed a modified GA which was designed to prevent the premature convergence and to speed up the convergence to the solution. The results on the two systems studied here indicate that GA can be used to discover neural networks for controlling nonlinear unstable plants using only failure information and no *a priori* knowledge. The control problem considered in this paper can not be solved by EBP because there is a constraint in weight values and no error signal is available. Further work includes simultaneous search for the weights and the architecture of a neural network.

REFERENCES

[1] D. E. Goldberg, *Genetic Algorithms in Search, Optimization, and Machine Learning.* Reading, MA: Addison-Wesley, 1989.

[2] L. Davis, *Handbook of Genetic Algorithms.* Reading, MA: Van Nostrand Reinhold, 1991.

[3] K. Kristinsson and G. A. Dumont, "System identification and control using genetic algorithms," *IEEE Trans. Syst., Man, Cybern.*, vol. 22, no. 5, pp. 1033-1046, Sep., 1992.

[4] C. L. Karr and E. J. Gentry, "Fuzzy control of pH using genetic algorithms," *IEEE Trans. Fuzzy Syst.*, vol. 1, no. 1, pp. 46-53, Feb., 1993.

[5] Y. Ichikawa and T. Sawa, "Neural network application for direct feedback controllers," *IEEE Trans. Neural Networks*, vol. 3, no. 2, pp. 224-231, Mar., 1992.

[6] A. G. Barto, R. S. Sutton and C. W. Anderson, "Neuronlike adaptive elements that can solve difficult learning control problems," *IEEE Trans. Syst., Man, Cybern.*, vol. SMC-13, no. 5, pp. 834-846, Sep., 1983.

[7] C. W. Anderson, "Learning to control an inverted pendulum using neural networks," *IEEE Control Syst. MAg.*, pp. 31-37, Apr., 1989.

[8] J. Hertz, A. Krogh and R. G. Palmer, *Introduction to the Theory of Neural Computation.* Reading, MA: Addison-Wesley, 1991.

[9] M. McInerney and A. P. Dhawan, "Use of genetic algorithms with backpropagation in training of feed-forward neural nwtworks," *IEEE International Conference on Neural Networks*, pp. 203-208, 1993.

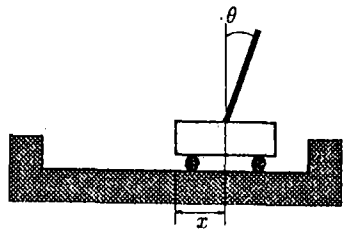


Fig. 1 Single pole system

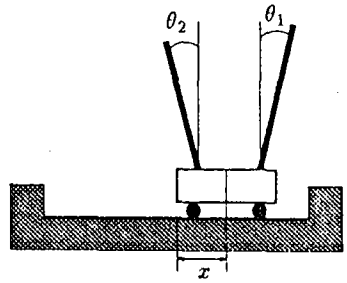


Fig. 2 Double pole system

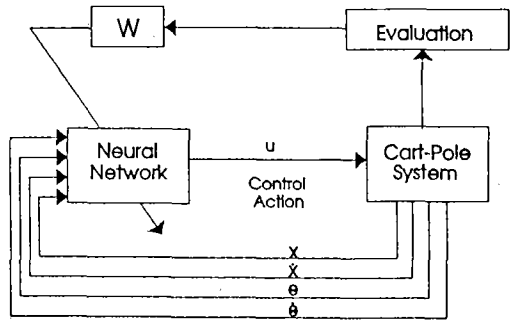
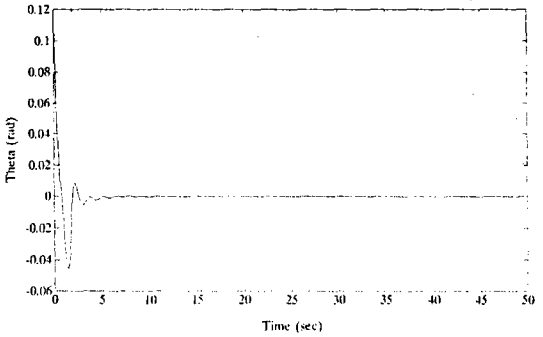
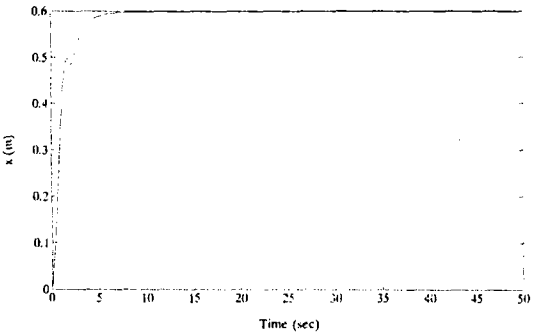


Fig. 3 Overall system block diagram

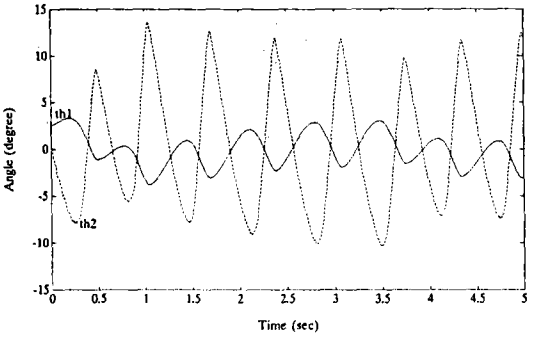


(a) θ

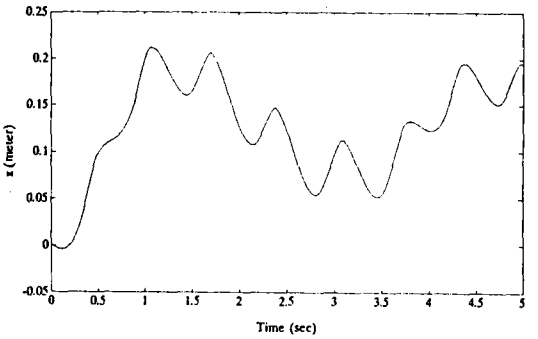


(b) x

Fig. 4 Result of the single pole system



(a) θ_1, θ_2



(b) x

Fig. 5 Result of the double pole system