Proceedings of the 11th
KACC, October 1996

Adaptive Control with multiple model (using genetic algorithm)

SeongChul Kwon*, Juhyun Park, Sangchul Won
*Dep. of Electronic and electric engineering ,POSTECH, Hyoja-dong San 31,Pohang,KOREA
Tel: (0562)279-2894; Fax: (0562)279-2903;E-mail; bestlife@jane.postech.ac.kr

Abstracts It is a well-known problem that the adaptive control has a poor transient response. In order to improve this problem, the scheme that model-reference adaptive control (MRAC) uses the genetic algorithm (GA) in the search for parameters is proposed. Use genetic algorithm (GA) in the searching for controller's parameter set and conventional gradient method for fine tuning. And show the reduction of the oscillations in transient response comparing with the conventional MRAC.

Keywords Model-reference adaptive control, genetic algorithm, transient response

1. INTRODUCTION

In direct and indirect adaptive control for a linear time-invariant plant, a lot of globally stable algorithms that result in zero steady-state tracking error have been developed [1]. These algorithms have revealed that the tracking error is quite often oscillatory, however , with large amplitudes during the transient phase. In order to improve the transient performance, several methods is suggested. One of them uses the multiple models of the plant to be controlled and switching between them [2].(but too many models.) In this paper, we suggest the adaptive control method using genetic algorithm to improve the transient performance.

Genetic algorithms, [3] has been useful in solving a wide variety of problems, since Holland introduced the concept of natural selection to optimization in 1975, Especially in control problem, the genetic algorithm has applied to the areas of optimization of controller parameters, system identifications, neural network and fuzzy controllers. For examples, parameter optimization of PID controller [4][5], system identifications [6], and updating the weight of neural-network controllers [7]. In [5] parameters of PD controller is optimized in on-line.

In this paper we applied the genetic algorithms in parameter optimization of direct adaptive controller and improved the transient response. Using the genetic algorithms we did a approximate search for adaptive controller parameters and then, using the conventional gradient method for the fine tuning.

In the subsequent sections we show the genetic algorithms we used, how to apply this to the adaptive control and the simulation results.

2. GENETIC ALGORITHMS

Genetic algorithms is defined in FAQ of newsgroups as following.

"A GA is an optimization method that starts with some encoded procedure. mutates it stochastically, and use a selection process to prefer the mutants with high fitness and perhaps a recombination process to combine properties of the successful mutants."

In [3] (Goldberg, 1989), the conventional and standard form of genetic algorithms is described. Some Deviation from that is showed as follows.

2.1 Representation and Selection

The standard GA has been used the binary representation, but we use the real-valued representation. The advantages for using real-valued representation is that no need for encoding and the quantization error resulted from that. (For more details see [8]) So crossover and mutation operators is different from that of binary representation.

2.2 Crossover and Mutation

We use the arithmetical crossover and non-uniform mutation operator. In next, s_v^t and s_w^t is the chromosome in vector form.

2.2.1 Arithmetical crossover

Arithmetical crossover is defined as a linear combination of two vectors: if s_v^t and s_w^t are to be crossed, the resulting offspring are

$$s_v^{t+1} = a \cdot s_w^t + (1-a)s_v^t$$

$$s_w^{t+1} = a \cdot s_v^t + (1-a)s_w^t$$

,where a can be a constant or varying with time, but we set this value a constant.

2.2.2 Non-uniform Mutation

If $s_v^t = \langle v_1, v_2, \ldots, v_n \rangle$ is a chromosome and v_k was the selected element for a mutation (bound domain of v_k is $[\, l_k, u_k \,]$, the lower and upper bound of v_k), the result is a vector $s_v^{t+1} = \langle v_1, v_2, \ldots, v_k', \ldots, v_n \rangle$, with $k \in \{1, \ldots, n\}$,and

$$v_k' = \begin{cases} v_k + D(t, u_k - v_k) & \text{if a random digit is 0} \\ v_k - D(t, v_k - l_k) & \text{if a random digit is 1} \end{cases}$$

,where the function $\Delta(t,y) = y \cdot (1 - r^{(1-t/T)^b})$ returns a value in the range [0,y] such that the probability being close to 0 increases as t increases. r is a random number from [0..1] , T is the maximal generation number, and b is a system parameter determining the degree of non-uniformity. (we select this value 2)

2.3 Others in GA.

- To prevent the parameter falling into the premature convergence, we keep the population maintaining some diversity .
- To make the next generation, we use first population and sub-population generated by crossover & mutation.
- We use the linear fitness scaling.(the best fitness is twice the average fitness)
- we use the basic roulette wheel section method.

3. MRAC WITH GA

3.1 Conventional MRAC

The control problem we wish to deal with in this paper is the basic MRAC problem treated in standard text books on the subject([1]). The plant to be controlled is linear-time-invariant (LTI) system with single input and single output related by

$$x_p = A \cdot x_p + B \cdot u \qquad y_p = C \cdot x_p$$

The reference model to followed is LTI system with input, r and output , ym

$$x_m = A \cdot x_m + B \cdot u \qquad y_m = C \cdot x_m$$

This MRAC has the poor transient response that show the serious oscillations. We wish to improve this problem with genetic algorithms and show in next section.

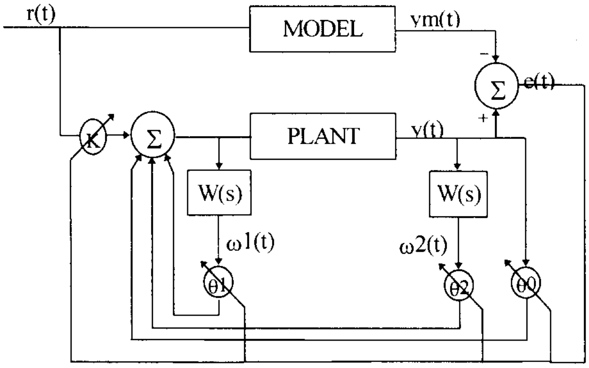

Figure 1. Conventional Direct MRAC

3.2 MRAC with GA

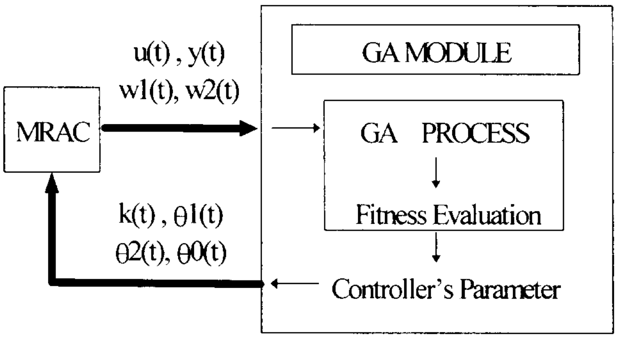

Figure 2 Architecture of MRAC with GA

In MRAC with GA, GA module generate the controller's parameter at every time, the MRAC block control the plant using those parameters. The Overall architecture is Figure 2 .

The GA module uses the model of real plant to predict the one step ahead plant output. Basically this model follows the reference model and we select the parameters of the controller that makes the real plant's output follow the predicted output of the plant model .The Procedure that determine the best parameter set of controller is as follows.

- Procedure

1. Initialize the population at random.
2. Collect the real plant input and output. (u, y_p and y_m)
3. Controller candidates using that consist of the old generation and newly making generation the parameter set is applied to plant model and predict the one-step ahead output.
4. For each controller candidates, the fitness is calculated and find the fittest control input candidate.
5. We use the control input to the real plant till next time step.

In above procedure, the fitness is evaluated by the weighted sum of three terms.

$$t1 = y\_hat\ (t+1) - y_p(t)\ ;$$
$$t2 = y\_hat\ (t+1) - y_m(t)\ ;$$
$$t3 = u(t)\ ;$$
$$O(i) = a \cdot t1 + b \cdot t2 + g\, t3 \qquad (1)$$

$$F(i) = \begin{cases} \dfrac{1}{O(i)} & \text{if } O(i) \text{ is not } 0 \\ 2*\max(F(j)) & \text{if } O(i) \text{ is } 0, \text{ and } j \neq i \end{cases}$$

,where y_hat(t+1) is the one step ahead output of the plant model, $y_p(t)$ the real plant output, $y_m(t)$ the reference model output, and $u(t)$ the control input made of each parameter set. The coefficients α , β , γ is determined by trial-and-error. β represent how plant model in GA module follows reference model output by the control parameter set , α how the plant model is similar to the real plant output, and γ represent the relative ratio of the control input to the $t1, t2$. Object value O(i) is evaluated as above and converted to the fitness value(non negative value) in order to use in GA. If O(i) is 0, we put twice the largest value of F(j) ($j \neq i$) to the F(i) in order to have lager value than other. And the penalty is given to the parameter set if $|y\_hat(t+1) - y_m(t)| > 1$.

If the controller's parameters generated by GA module is in some region for some generation, the gradient method is used for fine tuning of parameters. (In here, if the parameter set has the variance of .5 for 10 generation)

4. SIMULATION RESULTS

The control parameters used in the GA module is as follows.

Crossover probability	0.6
Mutation Probability	0.3
α, β, γ	150 , 500 ,2*10^{-6}
Population Size	200

And we use the second order plant and reference model as followings

$$\text{Plant}: \quad G_p(s) = \frac{1}{(s-1)(s-2)}$$
$$\text{Reference Model}: \qquad (2)$$
$$G_m(s) = \frac{1}{s+1}$$

The simulation results is in Figure 3., Figure 4 and Figure 5. In Figure 3., the output of plant and reference model in conventional MRAC is showed. There are the considerable oscillation appeared in the adaptive process and they degrade the transient performance. Figure 4 shows the output of the plant and reference model output in MRAC with GA. And Figure 5 shows the error and control input in MRAC with GA. As Figure 4 and Figure 5 show the facts, we can know that this scheme reduce the oscillation and the transient response is improved.

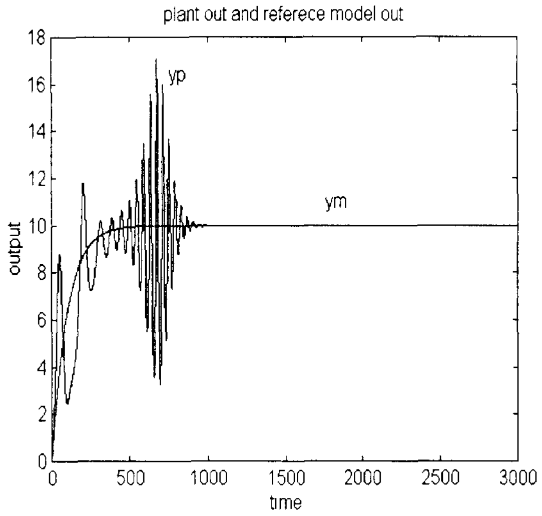

Figure 3. Plant output of conventional MRAC

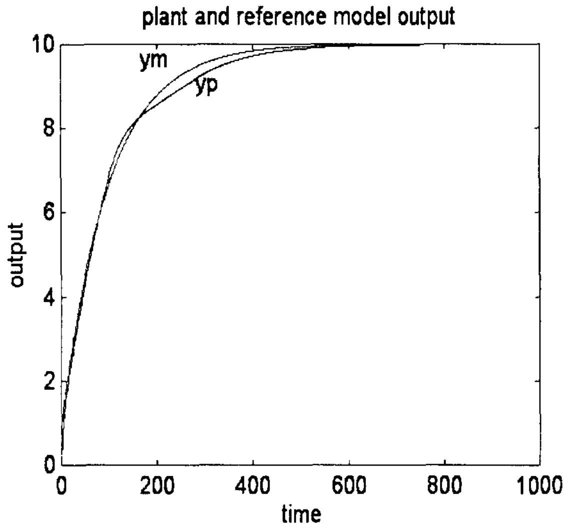

Figure 4. MRAC with GA

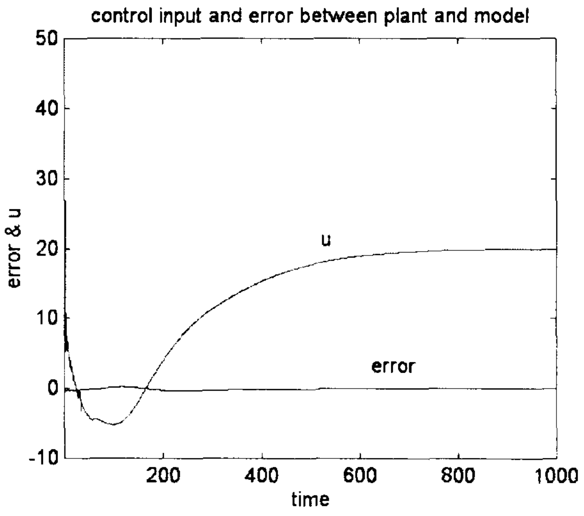

Figure 5. The error & control input MRAC with GA

5. CONCLUSIONS

That adaptive control has the poor transient response is the well-known problem. In this paper, we apply the genetic algorithm to MRAC in order to improve transient performance The simulation results show that MRAC with GA reduce the oscillations and improve the transient response considerably .In further study, this scheme needs the theoretical proof and also needs more effective controller structure for genetic algorithms that we can force the transient performance specifications to the plant.

REFERENCES

[1] K. S. Narendra and A. M. Annaswamy, *Stable Adaptive Systems.* Englewood Cliffs, NJ: Prentice-Hall, 1989

[2] K. S. Narendra and J. Balakrishnan, *Improving Transient Response of Adaptive Control Systems using Multiple Models and Switching.* IEEE Trans. Automat. Contr., vol.39, no. 9, pp1861-1866, 1994

[3] David E. Goldberg, *Genetic Algorithms in Search, Optimization, and Machine Learning.* New York: Addison-wesley

[4] B. Porter and A. H. Jones, *Genetic Tuning of Digital PID Controllers,* Electric letters 23rd April, vol. 28, no. 9,1992.

[5] L. L. Porter II and K. M. Passino, *Genetic Model Reference Adaptive Control,* IEEE International Symposium on Intelligent Control, pp. 219-224, 1994

[6] K. Kristtinsson and G. Dumont, *System Identification and Control using genetic algorithms,* IEEE Trans. On Systems, Man and Cybernetics, vol. 22, no. 5, pp 1033-1046, 1992.

[7] Y. Ichikawa and T. Sawa, *Neural Network Application for Direct Feedback Controllers,* IEEE Trans. on neural networks, vol. 3, no. 2, 1992.

[8] Z. Michalewicz, *Genetic Algorithms + Data Structures = Evolution Programs,* 2nd Edition, Springer-Verlag