Control and Cybernetics

vol. **30** (2001) No. 1

Optimal design of model following control with genetic algorithm

by

Xiaojun Zhang* **and Yuzo Yamane**

Department of Mechanical Enginneering, Ashikaga Institute of Technology
268-1 Omae, Ashikaga-city, Tochigi-prefecture, 326-0845 Japan
E-mail: d96105@ashitech.ac.jp

Abstract: This paper presents a genetic algorithm for the optimal design of model following control in which there are nonlinear disturbance and unceratin parameters, where the output is regulated to follow the output of reference model. The effectiveness of the proposed algorithm is illustrated by numerical examples.

Keywords: genetic algorithm, optimal design, model following control, feedback, feedforward.

1. Introduction

In this paper, we deal with the optimal design problem of model following control in which there are nonlinear disturbance and uncertain parameters, where the output is regulated to follow the output of a reference model. In order to determine the optimal feedback and feedforward parameters of our system, a genetic algorithm is carefully designed.

Model following control is also called approximate model matching control whose performance and dynamic property are determined by the reference model, that is, the control will correct the plant so as to coincide with the properties of the reference model. A lot of research works on output feedback have been restricted to the area of stability (Zeheb, 1986, Steiberg and Corless, 1985). The research works concerning model following control have also been done in the control theory of state feedback (Oya et al., 1989). When the plant involves uncertain parameters (such as unknown disturbance, nonlinear term), H_∞ optimal control, robust control as well as other linear control systems are discussed in a wide range (Frencis, 1987, Vidyasagar, 1985, Iwai et al., 1994, Chen et al., 1992). However, the discussed techniques of output feedback are focused on the stability of control system but not on the reference model following control.

The purpose of this paper is to provide a genetic algorithm for optimal design of gain parameters in a reference model following control system in which

*To whom correspondence should be addressed.

the output of the plant model is regulated to follow the output of a reference model based on the plant state feedback and the reference model state feedforward. Furthermore, some numerical examples are presented to illustrate the effectiveness of the proposed genetic algorithm.

2. Problem description

Consider the following SISO systems described by the n order controllable canonical form

$$\Sigma_p : \begin{cases} \dot{x}_p(t) = A_p x_p(t) + b u_p(t) \\ y_p(t) = c_p^T x_p(t) \end{cases} \tag{1}$$

$$\Sigma_m : \begin{cases} \dot{x}_m(t) = A_m x_m(t) + b u_m \\ y_m(t) = c_m^T x_m(t) \end{cases} \tag{2}$$

where Σ_p is the plant, Σ_m is the stable reference model, and

$$A_p = N + b\{a_p + \tilde{a}_p(x_p, t) + \Delta a_p\}^T, \; A_m = N + b a_m^T,$$
$$b = (0, 0, \cdots, 0, 1)^T \in R^n.$$

In the above equations, N is the nilpotent matrix, $\tilde{a}_p(x_p, t)$ is a term of nonlinear function, and Δa_p is a term of uncertainty with known upper and lower boundaries; a_m^T is the parameter of the reference model, u_m is a step signal. Parameters a_p and c_p are known, and $x_p(t)$ are observable.

The stable integral type interacter polynomial of the plant is expressed by:

$$\sigma(s) = s^\nu + \mu_\nu s^{\nu-1} + \ldots + \mu_1 + \frac{\mu_0}{s},$$

where $\mu = (\mu_0, \cdots, \mu_\nu) \in R^{\nu+1}$ and ν is the relative degree of the system.

In order to obtain robust model following performance when there exist time-varying parameters, an integral compensator is necessary. Therefore, the control input in (1) is assumed as (3), and the following error is given in (4).

$$u_p(t) = -k_p^T x_p(t) + k_m^T x_m(t) + m u_m(t)$$
$$- f_p^T \int x_p(t)\, dt + f_m^T \int x_m(t)\, dt \tag{3}$$

$$\xi = y_m(t) - y_p(t), \tag{4}$$

where k_p, f_p, k_m and f_m are the gains of feedback and feedforward signals, respectively.

The purpose of this paper is to present an integral type linear control scheme with fixed gains so that $\xi \to 0$ for any input u_m.

When $\Delta a_p = 0$, the following results are given in Yamane and Zhang (1996)

$$k_p^T = \frac{1}{w_{p.}} c_p^T \bar{\sigma}(A_p) \tag{5}$$

$$f_p^T = \frac{\mu_0}{w_p} c_1^T \tag{6}$$

$$\frac{k_m^T}{m} = \frac{1}{w_m} c_m^T \bar{\sigma}(A_m) \tag{7}$$

$$\frac{f_m^T}{m} = \frac{\mu_0}{w_m} c_m^T \tag{8}$$

where $w_p = c_p^T A_p^{\nu-1} b$, $w_m = c_m^T A_m^{\nu-1} b$, $m = \frac{w_m}{w_p}$, and $\bar{\sigma}(s) = s^\nu + \mu_\nu s_1^{\nu-1} + \cdots \cdots + \mu_1$.

Thus, controller gain can be calculated according to a $\mu = (\mu_0, \cdots, \mu_\nu) \in R^{\nu+1}$ from the equation involving the $\sigma(s)$ coefficient.

The μ can be altered arbitrarily in the polynomial stable range. When $\Delta a_p \neq 0$, the controller gain in (6) and (7) will not be determined. So, we can calculate controller gain using the $\Delta a_p = 0$ condition. If we introduce (3) into (1) and combine the reference model with its extended system, we can obtain

$$\begin{cases} \dot{\eta} = \Lambda \eta + \gamma u_m \\ \xi = \pi^T \eta. \end{cases} \tag{9}$$

Here, $\eta = (x_p^T, x_m^T, z)^T$, $z = \int \xi \, dt$, $\xi = y_m - y_p$

$$\Lambda = \begin{bmatrix} A_p - bk_p^T & bk_m^T & b\frac{\mu_0}{w_p} \\ 0 & A_m & 0 \\ -c_p & -c_m & 0 \end{bmatrix} \quad \pi = [-c_p^T, c_m^T, 0]^T, \ \gamma = [mb^T, b^T, 0]^T.$$

Let us suppose that Λ is a stable value. Then, the detailed question is as follows. Suppose Δa_p is an unknown number under u_m given arbitrarily. We can introduce an evaluating function J to obtain the following error ξ tending toward zero:

$$J = \int_0^\tau \{\xi^T Q \xi + u_p(t)^T R u_p(t)\} \, dt. \tag{10}$$

Here, J depends on $\Delta a_p, \mu, u_m, x_p(0), x_m(0), \tau$ ($\tau > 0$), and τ is the time of a test control. In our paper, to simplify, we assume some parameters, such as $x_p(0), x_m(0), u_m, \tau$, to be constant. Because Δa_p is an unknown number, the optimal design problem can be described as a min-max problem

$$\min_{\mu}\{\max_{\Delta a_p}\{J(\mu, \Delta a_p)\}\}. \tag{11}$$

In general, we can solve the min-max problems using GA methods. In solving for the J function, we assume that Δa_p^* was taken as Δa_p when J attained the max value; and μ^* was taken as μ when J attained the min value.

According to the evaluation function, in the following section an optimal design of model following scheme for $\Delta a_p \neq 0$ is presented based on a genetic algorithm.

3. Design of a controller with GA

In this section a genetic algorithm for optimal design of the robust model following controller is developed. The following problems will be included: initialization process, evaluation function, selection operation, crossover and mutation operations.

3.1. Initialization process

We suppose the initial generation number (gen) is 0, and the end number of generation is GEN, and assume the following vector

$$V = (\mu_0 \quad \mu_1 \quad \cdots \quad \mu_\nu)$$

is the chromosome representing the optimal robust solution of the model following system, and define an integer *pop_size* as the number of chromosomes. The number *pop_size* of chromosomes will be randomly initialized by the following steps:

Step 1. Determine an interior point, denoted by V_0, in the constraint set.

Step 2. Select randomly a direction d in $R^{2\times n}$ and define a chromosome V as $V_0 + M \cdot d$ if it is feasible, otherwise, set M as a random number in $[0, M]$ until $V_0 + M \cdot d$ is feasible, where M is a large positive number which ensures that all the genetic operators are probabilistically complete for the feasible solutions.

Step 3. Repeat Step 2 *pop_size* times and produce *pop_size* initial feasible solutions.

3.2. Stability and test control

Step 1. Check the interaction polynomial according to Hurwitz criterion, if it is stable then go to Step 2, otherwise, go to Step 3.

Step 2. Determine control gains according to (5)–(8), execute control test, and go to Section 3.3.

Step 3. Control test is not executed because stability of the system is unknown, go to Section 3.4.

3.3. Evaluation function

In our calculations we refer to an evaluating function, so that for some uncertain parameters, we first choose one of them that will be changed with a given step length. When the value of the given parameter changes from the lower to the upper boundary, the other uncertain parameters can be given random values in the known area while the J function attains the maximum value. Then, we can obtain μ corresponding to V_i using the GA method.

3.4. Selection operation

The selection process is based on spinning the roulette wheel *pop_size* times. Each time we select a single chromosome for a new population in the following way:

Step 1. Calculate the selection probability p_i and the cumulative probability α_k for each chromosome V_i $(i = 1, 2, \cdots, pop\_size)$ as follows:

$$F = \sum_{i=1}^{pop\_size} J(V_i) \tag{12}$$

$$p_i = \frac{J(V_i)}{F} \tag{13}$$

$$\alpha_k = \sum_{j=1}^{k} p_j \tag{14}$$

Step 2. Generate a random real number r_k in $[0, 1]$.

Step 3. If $r_k \le \alpha_1$, then select the first chromosome V_1, V_1 means V_k'; otherwise select the i-th chromosome V_i $(2 \le i \le pop\_size)$ such that $\alpha_{i-1} < r_k \le \alpha_i$.

Step 4. Repeat Steps 2 and 3 *pop_size* times and obtain *pop_size* copies of chromosomes.

In this process, the best chromosomes yield more copies, the average ones stay even, and the worst ones die off.

3.5. Crossover operation

In accordance with the crossover operation of the unimodal normal distribution crossover (UNDX) from Ono and Kobayashi (1997):

Step 1. Generate a random real number r_j; here, $0 \le r_j \le 1$, $j = 1, 2, \cdots \cdots, pop\_size$.

Step 2. Define a parameter P_c of a genetic process as the probability of crossover. This probability gives us the expected number P_c_size of chromosomes which undergo the crossover operation.

Firstly we generate a random real number r_j in $[0, 1]$; secondly, we select the given chromosome for crossover if $r_j < P_c$. This operation is repeated *pop_size* times and produces P_c parents, on the average. For each pair of parents (vectors V_i' and V_j'), based on UNDX, the crossover operator on V_i' and V_j' will produce two children, V_i'' and V_j'', conform to (15) and (16) below:

$$V_i'' = G + \mathcal{Z}_1 F_1 + \sum_{k=2}^{\nu} \mathcal{Z}_k F_k \tag{15}$$

$$V_j'' = G - \mathcal{Z}_1 F_1 - \sum_{k=2}^{\nu} \mathcal{Z}_k F_k \tag{16}$$

where $G = \frac{V_i'+V_j'}{2}$, $\mathcal{Z}_1 \sim N(0,\omega_1^2)$, $\mathcal{Z}_k \sim N(0,\omega_2^2)$ $(k = 2,\ldots,\nu)$, $\omega_1 = \rho d_1$, $\omega_2 = \frac{\beta d_2}{\sqrt{\nu}}$, $F_1 = \frac{V_i'-V_j'}{|V_i'-V_j'|}$, $F_i \perp F_j$ $(i,j = 1,\ldots,\nu)$ $(i \neq j)$. Here, based on UNDX, the d_1 is distance between V_i' and V_j', d_2 is vertical distance from the point of the chromosome 3 to line between V_i' and V_j'; ρ, β are the weights of the random real numbers in unimodal normal distribution function.

3.6. Mutation operation

We define a parameter P_m of a genetic process as the probability of mutation. This probability gives us the expected number $P_m \cdot pop\_size$ of chromosomes which undergo the mutation operation.

When generating a random real number r in $[0,1]$, we select the given chromosome for mutation if $r < P_m$. Let a parent for mutation, denoted by a vector

$$V = (\ \mu_0 \quad \mu_1 \quad \cdots \quad \mu_\nu\ )$$

be selected. Select randomly a direction d in $R^{2\times n}$ and define a chromosome V as $V_0 + M \cdot d$ if it is feasible, otherwise, we set M as a random number in $[0, M]$ until $V_0 + M \cdot d$ is feasible, where M is a large positive number defined in the initialization process.

gen $\leftarrow$ gen + 1

If gen $\leq$ GEN then go to Section 3.2, otherwise, simulation will end and the gain corresponding to the chromosome with maximum adaptation degree is used as control gain.

Repeat the above process for all chromosomes.

Following selection, crossover and mutation, the new population is ready for its next evaluation. The algorithm will terminate after a given number of cyclic repetitions of the above steps.

4. Numerical example

Here we will illustrate the effectiveness of proposed genetic algorithm for the optimal design of model following control by some numerical examples. Computer simulations were executed on NEC EWS4800/210II workstation with the following parameters: population size = 30, probability of crossover $P_c = 0.2$, probability of mutation $P_m = 0.4$.

The second order plant is described as follows

$$\dot{x}_p(t) = \begin{bmatrix} 0 & 1 \\ -0.5 + \tilde{a}_{p1} + \Delta a_{p1} & -1 + \tilde{a}_{p2} + \Delta a_{p2} \end{bmatrix} x_p(t) + \begin{bmatrix} 0 \\ 1 \end{bmatrix} u_p(t)$$

$$y_p(t) = [1 \quad 0] x_p(t)$$

where $x_p(t) = [x_{p1}(t), x_{p2}(t)]^T$ is the state vector of plant, $u_p(t)$ is the control signal, $\tilde{a}_{p1}$ and $\tilde{a}_{p2}$ are nonlinear disturbances; Δa_{p1} and Δa_{p2} are uniformly distributed variables on the intervals $[-0.2, 0.2]$ and $[-0.4, 0.4]$.

The reference model is described as follows

$$\dot{x}_m(t) = \begin{bmatrix} 0 & 1 \\ -2.5 & -3 \end{bmatrix} x_m(t) + \begin{bmatrix} 0 \\ 1 \end{bmatrix} u_m, \; y_m(t) = [1 \quad 0] x_m(t)$$

where $x_m(t) = [x_{m1}(t), x_{m2}(t)]^T$ is the state vector of the reference model. The initial state is

$$x_p(0) = (0.6, 0)^T, \; x_m(0) = (0, 0)^T.$$

The input of the reference model is $u_m = 4$.

The purpose of the control is that the output of the plant with parameter uncertainties and nonlinear factors follow the output of the reference model robustly and fast, and the control $u_p(t)$ is not performed with a too big signal. Based on the method stated in Section 3, the parameter μ_i of the interaction polynomial is determined. In this simulation study the test execution time T was 20 seconds, the weights of the evaluation function CP were Q = 3 and R = 0.5, the end generation number was GEN = 100.

The programs for the proposed genetic algorithm are written in C language. We use it to solve the above numerical example. In this example, $\nu = 2$, we restrict the parameters in the following set

$$\{(\mu_2, \mu_1, \mu_0) \mid 0 \le \mu_i \le 30, \; i = 0, 1, 2\}$$

which is clearly convex.

The following parameters

$$(\mu_2, \mu_1, \mu_0)^* = (19.7, 5.6, 0.11)$$

are used and the corresponding control gains are $k_p^T = (19.2, 4.6)$, $k_m^T = (17.2, 2.6)$, $f_p^T = (0.11, 0)$, $f_m^T = (0.11, 0)$. So, the following control

$$u_p(t) = -(19.2, 4.6) x_p(t) + (17.2, 2.6) x_m(t) + 0.11 \int \{x_{m1}(t) - x_{p1}(t)\}\, dt$$

is obtained. The total CPU time of the NEC EWS4800/210II workstation is 786.5 seconds. The plant and reference model output signals obtained by the genetic algorithm are shown in Figs. 1 and 2. In Fig. 1 the output of the

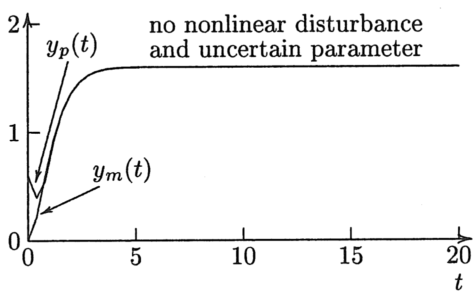

Figure 1. Plant and reference model output signals

plant is given when there is no parameter uncertainty and nonlinear factors, that is, $\tilde{a}_{p1}$, $\tilde{a}_{p2}$, Δa_{p1}, and Δa_{p2} are zero. Fig. 2 gives the output $y_p(t)$ when $\tilde{a}_{p1} = e^{-t}\sin t$, $\Delta a^*_{p1} = -0.2$, $\tilde{a}_{p2} = e^{-(x_{p1}(t)+x_{p2}(t)\sin t)}$, $\Delta a^*_{p2} = 0.39$.

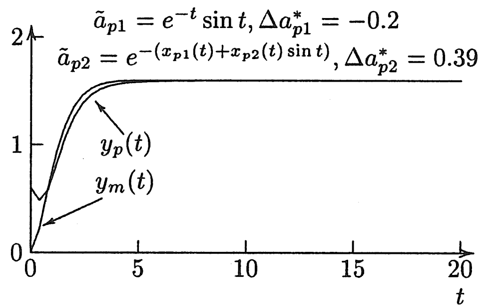

Figure 2. Plant and reference model output signals

From these simulation results it can be concluded that the output of the plant can fast and robustly follow the output of the reference in spite of the existence of plant parameter uncertainties and nonlinear factors by using the designed control $u_p(t)$.

5. Conclusion

This paper presented a genetic algorithm for optimal design of gain parameters in a reference model following control system in which the output of plant was regulated to follow the output of reference model, based on plant state feedback and reference model state feedforward. The effectiveness of the proposed genetic algorithm was illustrated by a numerical example.

In the example, the crossover probability 0.4 was chosen, because the proposed algorithm does not work often for values bigger than around 0.4, while the mutation probability was set at 0.2 here because the evaluation function does not decrease for values smaller than around 0.2.

References

CHEN, S., YONEZAWA, Y. and NISHIMURA, Y. (1992) A Design Method of MRACS for the Plants with Slowly Varying Undeterministic Disturbance. *Transactions of the Society of Instrument and Control Engineers*, **28**, 4, 478/483.

FRANCIS, B.A. (1987) A Course in H_∞ Control Theory. *Lecture Note on Control and Infomation Science* **88**, Springer-Verlag.

IWAI, Z., MIZUMOTO, I. and ADACHI, K. (1994) Model Output Following Control by Static Output Feedback and Its Robustness in the Presence of Parasitics. *Transactions of the Society of Instrument and Control Engineers*, **30**, 1, 31/38.

ONO, I. and KOBAYASHI, S. (1997) A Real-coded Genetic Algorithm for Function Optimization Using Unimodal Normal Distribution Crossover. *Proc. of 7th International Conference of Genetic Algorithm*, 246/253.

OYA, M., NISHIMURA, Y. and YONEZAWA, Y. (1989) Model Following Control for a Alass of Nonlinear Systems. *Transactions of the Society of Instrument and Control Engineers*, **25**, 2, 145/151.

STEINBERG, A. and CORLESS, M. (1985) Output Feedback Stabilization of Uncertain Dynamical Systems. *IEEE Trans.*, **AC-30-10**, 1025/1027.

VIDYASAGAR, M. (1985) *Control System Synthesis.* MIT Press.

YAMANE, Y. and ZHANG, X. (1996) Design of Robust Reference Model Control System. *Proc. Associated Symposium On System and Information*, 273/276.

ZEHEB, E. (1986) A Sufficient Condition for Output Feedback Stabilization of Uncertain Systems. *IEEE Transactions on Automatic Control*, **AC-31-11**, 1055/1057.