Identification of Continuous Time-Delay Systems Using the Genetic Algorithm

Tomohiro Hachino, Zi-Jiang Yang and Teruo Tsuji
Department of Electrical Engineering
Kyushu Institute of Technology
1-1 Sensui-cho, Tobata-ku, Kitakyushu, 804, JAPAN

Abstract- This report proposes a novel method of identification of continuous time-delay systems from sampled input-output data. By the aid of a digital pre-filter, an approximated discrete-time estimation model is first derived, in which the system parameters remain in their original form and the time delay need not be an integral multiple of the sampling period. Then an identification method combining the common linear least squares (LS) method or the instrumental variable (IV) method with the genetic algorithm (GA) is proposed. That is, the time-delay is selected by the GA, and the system parameters are estimated by the LS or IV method. Furthermore, the proposed method is extended to the case of multi-input multi-output systems where the time-delays in the individual input channels may differ each other. Simulation results show that our method yields consistent estimates even in the presence of high measurement noises.

1 Introduction

Since many practical systems have inherent delays, the problem of identifying such a system is of great importance for analysis, synthesis and prediction. However the identification of time-delay systems is greatly complicated, because the estimation model becomes nonlinear in the parameters with respect to the time-delays.

Numerous identification methods of time-delay systems based on the discrete-time model have been proposed[1)~3)]. In the case of the discrete-time model, the sampling period is required to be very small such that the time-delays are an integral multiple of the sampling period, whereas if the sampling period is too small, the identification becomes very difficult. Moreover, the parameters in the discrete-time model usually do not correspond to the physical parameters. Therefore, the importance of the identification based on the continuous-time model has been recognized in recent years[4)], and we also consider the identification of time-delay systems based on the continuous-time model.

There are two typical approaches based on the continuous-time model. One approach is based on the approximation of the time-delays in the frequency domain by a rational transfer function or the padé approximation[5)~7)]. This approach requires estimation of more parameters bacause the order of the approximated system model is increased, and it is not easy to separate the parameters concerned with the time-delays from those concerned with the dynamics of the system. Moreover the unacceptable approximation error occurs when the system has large time-delays.

The other approach is based on the nonlinear estimation method like nonlinear least squares method, etc.[8)~12)] In this approach, unfortunately the estimates of the time-delays and the system parameters often converge to local optima if we do not choose suitable initial values of the parameters and forgetting factor. Moreover, this method is infeasible when the measurement is corrupted by a high level noise.

In the above mentioned works, the identification of time-delay systems which have multi time-delays are seldom considered. Since the identification of such a system becomes more difficult, to the best of author's knowledge, few works which deal with the identification of multi-input multi-output time-delay systems have been reported.

The GA is a probabilistic search procedure based on the mechanics of natural selection and natural genetics[13)]. Recently the GA has received considerable attention in various fields[14),15)], because it has high ability for global optimization.

In this report, we propose a method of identification of continuous time-delay systems from sampled input-output data combining the LS or IV method with the GA. At first, by the aid of a digital pre-filter, we get an approximated discrete-time estimation model, in which the system parameters remain in their original form, and the time-delay need not be an integral multiple of the sampling period. Then, candidates of the unknown time-delay in this model are coded into binary bit strings, and the system parameters are estimated based on the LS or IV method using each of candidates of the decoded time-delay which is selected by the GA. Furthermore, reproduction, crossover and mutation are repeated such that the fitness value of the population increases. In this case, the fitness values are represented by the function of the least squares errors calculated using the estimates of the parameters. Thus the estimates of the time-delay and the system parameters can be obtained. It is shown by simulations that the method combining the IV method with the GA gives consistent estimates even in the presence of high measurement noises. Furthermore we extend the proposed method to the identification of multi-input multi-output continuous time-delay systems, and show that our method is very valid for such a system.

2 Identification of single-input single-output continuous time-delay systems

2.1 Statement of the problem

Consider the following single-input single-output continuous time-delay system:

$$\sum_{i=0}^{n} a_i p^{n-i} x(t) = \sum_{i=1}^{m} b_i p^{m-i} u(t-\tau) \qquad (a_0 = 1) \quad (1)$$

where p is a differential operator, $u(t)$, $x(t)$ are real input and output, and τ is the time-delay. n and m are assumed to be known $(n \geq m)$.

It is assumed that a zero-order hold is utilized such that

$$u(t) = \bar{u}(k) \qquad (k-1)T \leq t < kT \quad (2)$$

where T is the sampling period.

Practically the measurement of the output variable is corrupted by a stochastic measurement noise.

$$y(k) = x(k) + v(k) \quad (3)$$

Our goal is to identify the time-delay and the system parameters from sampled data of the input and the noisy output.

2.2 Approximated discrete-time estimation model

An approximated discrete-time estimation model is obtained by the aid of a digital pre-filter in order to avoid the direct signal derivatives. In this report we introduce a low-pass pre-filter $Q(p)$ as

$$Q(p) = \frac{1}{(\alpha p + 1)^n} \quad (4)$$

where α is the time constant which determines the pass-band of $Q(p)$.

Multipling both sides of Eq.(1) by $Q(p)$ and using the bilinear transformation based on the block-pulse functions[4),16),17)], we can obtain the following approximated discrete-time estimation model of the original system[4),16),17)]:

$$\xi_{0\bar{y}}(k) + \sum_{i=1}^{n} a_i \xi_{i\bar{y}}(k) = \sum_{i=1}^{m} b_i \xi_{(n-m+i)\bar{u}}(k-\tilde{\tau}) + r(k) \quad (5)$$

where

$$\xi_{i\bar{y}}(k) = Q_0(z^{-1}) \left(\frac{T}{2}\right)^i (1+z^{-1})^i (1-z^{-1})^{n-i} \bar{y}(k)$$

$$\xi_{(n-m+i)\bar{u}}(k-\tilde{\tau}) =$$

$$Q_0(z^{-1}) \left(\frac{T}{2}\right)^{n-m+i} (1+z^{-1})^{n-m+i} (1-z^{-1})^{m-i} \bar{u}(k-l)$$

$$r(k) = \sum_{i=0}^{n} a_i Q_0(z^{-1}) \left(\frac{T}{2}\right)^i (1+z^{-1})^i (1-z^{-1})^{n-i} v(k)$$

$$Q_0(z^{-1}) = \frac{1}{\left[\alpha(1-z^{-1}) + \frac{T}{2}(1+z^{-1})\right]^n} \quad (6)$$

Here $\bar{y}(k) = (1+z^{-1})y(k)/2$ is the block-pulse approximation of $y(t)$[4),16),17)] and $\tilde{\tau}$ is given by

$$\tilde{\tau} = \tau/T = l + \Delta/T \qquad (0 \leq \Delta < T, \quad l: \text{nonnegative integer}) \quad (7)$$

It should be noticed that generally it is desirable that the time-delay τ is an integral multiple of the sampling period in the identification based on the discrete-time model, while our approximated discrete-time estimation model does not require this restriction. Because in the case of $\Delta \neq 0$, we can get $\xi_{(n-m+i)\bar{u}}(k-\tilde{\tau})$ by the linear interpolation of $\xi_{(n-m+i)\bar{u}}(k-l)$ and $\xi_{(n-m+i)\bar{u}}(k-l-1)$.

Eq.(5) can be written in vector form:

$$\xi_{0\bar{y}}(k) = \mathbf{z}^T(k,\tilde{\tau})\boldsymbol{\theta} + r(k)$$
$$\mathbf{z}^T(k,\tilde{\tau}) =$$
$$\left[-\xi_{1\bar{y}}(k), \cdots, -\xi_{n\bar{y}}(k), \xi_{(n-m+1)\bar{u}}(k-\tilde{\tau}), \cdots, \xi_{n\bar{u}}(k-\tilde{\tau})\right]$$
$$\boldsymbol{\theta}^T = [a_1, \cdots, a_n, b_1, \cdots, b_m] \quad (8)$$

2.3 Identification of the system parameters

Before presenting the identification method of the parameters including the time-delay, in this section we describe the identification of only the system parameters in the case of the known time-delay.

From Eq.(8) the system parameters can be estimated by the following LS method:

$$\hat{\boldsymbol{\theta}}_{LS} = \left[\sum_{k=k_s+1}^{k_s+N} \mathbf{z}(k,\tilde{\tau})\mathbf{z}^T(k,\tilde{\tau})\right]^{-1} \cdot \left[\sum_{k=k_s+1}^{k_s+N} \mathbf{z}(k,\tilde{\tau})\xi_{0\bar{y}}(k)\right] \quad (9)$$

where Eq.(9) is obtained in the sense of minimizing the following mean squares of the equation error:

$$V_{LS}(\boldsymbol{\theta}_{LS}, \tilde{\tau}) = \frac{1}{N} \sum_{k=k_s+1}^{k_s+N} (\xi_{0\bar{y}}(k) - \mathbf{z}^T(k,\tilde{\tau})\boldsymbol{\theta}_{LS})^2 \quad (10)$$

When the noise effects can not be neglected, it is well known that the LS estimates are biased in general[18)]. In this case we can apply the IV method in a bootstrap manner with the filtered input-output data of the estimated system model as instrumental variables in order to get consistent estimates. The estimated real output $\hat{\bar{x}}(k)$ can be obtained by the bilinear transformation:

$$\hat{\bar{x}}(k) = \frac{\sum_{i=1}^{m} \hat{b}_i \left(\frac{T}{2}\right)^{n-m+i} (1+z^{-1})^{n-m+i} (1-z^{-1})^{m-i}}{\sum_{i=0}^{n} \hat{a}_i \left(\frac{T}{2}\right)^i (1+z^{-1})^i (1-z^{-1})^{n-i}} \bar{u}(k-l) \quad (11)$$

where $\hat{a}_i$, $\hat{b}_i$ are the estimates of the system parameters. l is given by Eq.(7) and if $\Delta \neq 0$, we have $\hat{\bar{x}}(k)$ by the above mentioned interpolation.

Define the IV vector as follows:

$$\mathbf{m}^T(k,\tilde{\tau}) =$$
$$\left[-\xi_{1\hat{\bar{x}}}(k), \cdots, -\xi_{n\hat{\bar{x}}}(k), \xi_{(n-m+1)\bar{u}}(k-\tilde{\tau}), \cdots, \xi_{n\bar{u}}(k-\tilde{\tau})\right] \quad (12)$$

Using this IV vector, we can estimate the system parameters by the IV method:

$$\hat{\boldsymbol{\theta}}_{IV} = \left[\sum_{k=k_s+1}^{k_s+N} \mathbf{m}(k,\hat{\tau})\mathbf{z}^T(k,\hat{\tau})\right]^{-1} \cdot \left[\sum_{k=k_s+1}^{k_s+N} \mathbf{m}(k,\hat{\tau})\xi_{0\bar{y}}(k)\right] \tag{13}$$

We should notice that the IV estimates do not minimize the mean squares of the equation error of Eq.(10)[19]. However if the IV estimates are consistent, they minimize the following mean squares of the output error:

$$\begin{aligned} V_{IV}(\boldsymbol{\theta}_{IV},\hat{\tau}) &= \frac{1}{N}\sum_{k=k_s+1}^{k_s+N}(\bar{y}(k)-\hat{\bar{x}}(k))^2 \\ &= \mathrm{E}\left[\left(\bar{y}(k)-\hat{\bar{x}}(k)\right)^2\right] \\ &= \mathrm{E}\left[\left(\bar{x}(k)-\hat{\bar{x}}(k)\right)^2\right] + \mathrm{E}[\bar{v}^2(k)] \\ &= \mathrm{E}[\bar{v}^2(k)] \end{aligned} \tag{14}$$

where E[·] denotes the expectation. Therefore in the next section we calculate the fitness value from the mean squares of the equation error of Eq.(10) for the LS method. On the other hand, for the IV method we use the mean squares of the output error of Eq.(14).

2.4 Identification algorithm using the GA

As mentioned above, in the conventional methods the estimates may converge to local optima, or are often affected by measurement noises. In this section, to overcome these problems we propose an identification method using the GA, by which the estimates of the time-delay and the system parameters can arrive at the global optimal point without being locked at local optima. Both the time-delay and the system parameters are identified here.

At first we generate an initial population which is consisted of binary bit strings as candidates of the time-delay. Then the system parameters are estimated by the LS or IV method for each of candidates of the time-delay which are decoded from the strings. And then the fitness values mapped from the least squares errors are calculated. Furthermore the genetic operations, i.e., reproduction based on fitness values, crossover, mutation are repeated such that the fitness value of the candidate population of the time-delay increases, in order to obtain the optimal estimates of the time-delay and the system parameters. The concrete algorithm is described as follows:

(1) Coding of time-delay τ:

Code the time-delay τ into binary strings of L bits.

(2) Initial population:

Generate an initial population of M strings randomly.

(3) System parameter identification:

(LS method)

Identify the system parameters from Eq.(9) by the LS method for each of candidates of the time-delay which are decoded from the strings.

(IV method)

Identify the system parameters from Eq.(13) by the IV method for each of candidates of the time-delay which are decoded from the strings. The IV vector of Eq.(12) is given by using the estimated time-delay $\hat{\tau}^*$ and the estimated system parameters $\hat{a}_i^*$, $\hat{b}_i^*$, which have the best fitness value over all the past generation.

In the early generation, since some output errors of individual strings often diverge because some estimated models are unstable, it is impossible to calculate Eq.(14). In this case we treat these individual strings as dead genes. However this treatment may cause early convergence due to loss of the diversity in a population. Therefore we use the LS method until prespecified G_{LS}-th generation, then the IV method is started at $(G_{LS}+1)$-th generation.

(4) Fitness value calculation:

The least squares error is mapped into the fitness value for each of the strings. That is, maxmizing the fitness value corresponds to minimizing the least squares error.

(LS method)

Using the mean squares of the equation error, the fitness value function is given by

$$J_{LS}(\hat{\boldsymbol{\theta}}_{LS},\hat{\tau}) = \frac{1}{\beta + \frac{1}{N}\sum_{k=k_s+1}^{k_s+N}(\xi_{0\bar{y}}(k) - \mathbf{z}^T(k,\hat{\tau})\hat{\boldsymbol{\theta}}_{LS})^2} \tag{15}$$

(IV method)

Using the mean squares of the output error, the fitness value function is given by

$$J_{IV}(\hat{\boldsymbol{\theta}}_{IV},\hat{\tau}) = \frac{1}{\beta + \frac{1}{N}\sum_{k=k_s+1}^{k_s+N}(\bar{y}(k) - \hat{\bar{x}}(k))^2} \tag{16}$$

where $\hat{\bar{x}}(k)$ is the estimated real output which is calculated by the bilinear transformation and β in Eqs.(15),(16) is the constant which adjusts the shape of the curve of the fitness value function(β is chosen to be 0.01 in the examples). If the estimated model is judged to be unstable, let J_{IV} zero.

(5) Reproduction

In this report reproduction is implimented as linear seach through a roulette wheel slots weighted in proportion to the fitness value of the individual string in old generation. Namely each of old population is reproduced with the probability of $J_{LS}(\hat{\boldsymbol{\theta}}_{LS},\hat{\tau})/\bar{J}_{LS}$ or $J_{IV}(\hat{\boldsymbol{\theta}}_{IV},\hat{\tau})/\bar{J}_{IV}$. Here $\bar{J}_{LS}$ and $\bar{J}_{IV}$ are sums of the fitness value.

Moreover we use a policy called elitist reproduction which guarantees that the current string is likely to survive from generation to generation.

(6) Crossover

Pick up two strings randomly and decide whether or not to cross them over according to the crossover probability P_c. If crossover is required, exchange strings at a crossing position. The crossing position is chosen randomly.

(7) Mutation

Alter a bit of string (0 or 1) according to the mutation probability P_m.

(8) Repeating

(3)~(7) step are repeated from generation to generation such that the fitness value of the population increases. The best time-delay is identified as the individual string with the best fitness value over all the past generation, and estimates of the system parameters are given by the LS or IV method with the best time-delay.

3 Extension to multi-input multi-output continuous time-delay systems

In this section we extend our method to the identification of multi-input multi-output systems where the time-delays in the individual input channels may differ each other. In general such an identification problem is considered to be difficult, and to the best of our knowledge, it seems to be seldom reported in spite of the practical demand.

Furthermore the enumerative method which searches multi time-delays at every point one by one in multi-dimentional solution space is computationally infeasible in most cases, because the number of candidate of solution becomes enormous. On the other hand, our method using the GA is suitable for computation on a personal computer, because we can search global optimal point by tens or hundreds of individual strings.

Consider the following multi-input multi-output continuous time-delay system:

$$\begin{bmatrix} x_1(t) \\ \vdots \\ \vdots \\ x_q(t) \end{bmatrix} = \begin{bmatrix} \frac{N_{11}(p)}{D_1(p)} & \frac{N_{12}(p)}{D_1(p)} & \cdots & \frac{N_{1r}(p)}{D_1(p)} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{N_{q1}(p)}{D_q(p)} & \frac{N_{q2}(p)}{D_q(p)} & \cdots & \frac{N_{qr}(p)}{D_q(p)} \end{bmatrix} \begin{bmatrix} u_1(t-\tau_1) \\ \vdots \\ \vdots \\ u_r(t-\tau_r) \end{bmatrix}$$

$$D_i(p) = \sum_{j=0}^{n} a_{ij} p^{n-j} \qquad (a_{i0} = 1 \ ; \ i = 1, 2, \cdots, q)$$

$$N_{ih}(p) = \sum_{j=1}^{m} b_{ihj} p^{m-j} \qquad (i = 1, 2, \cdots, q \ ; \ h = 1, 2, \cdots, r) \tag{17}$$

Eq.(17) can be written as a set of multi-input single-output systems:

$$\sum_{j=0}^{n} a_{ij} p^{n-j} x_i(t) = \sum_{h=1}^{r} \sum_{j=1}^{m} b_{ihj} p^{m-j} u_h(t-\tau_h) \qquad (a_{i0} = 1)$$
$$(i = 1, 2, \cdots, q) \tag{18}$$

where $u_h(t)$, $x_i(t)$ are the hth input and the ith real output, and τ_h is the time-delay in the hth input. n and m are assumed to be known ($n \geq m$).

The problem to be treated here is to identify the time-delays and the system parameters from sampled data of the inputs and noisy outputs.

The approximated discrete-time estimation model can be obtained in the same way as described in section 2.2:

$$\xi_{0\bar{y}_i}(k) = \mathbf{z}_i^T(k, \tilde{\tau}_1, \cdots, \tilde{\tau}_r)\boldsymbol{\theta}_i + r_i(k) \qquad (i = 1, 2, \cdots, q) \tag{19}$$

where

$$\begin{aligned} \mathbf{z}_i^T(k, \tilde{\tau}_1, \cdots, \tilde{\tau}_r) &= \left[-\mathbf{z}_{i\bar{y}}^T(k), \mathbf{z}_{i\bar{u}_1}^T(k-\tilde{\tau}_1), \cdots, \mathbf{z}_{i\bar{u}_r}^T(k-\tilde{\tau}_r)\right] \\ \mathbf{z}_{i\bar{y}}^T(k) &= [\xi_{1\bar{y}_i}(k), \cdots, \xi_{n\bar{y}_i}(k)] \\ \mathbf{z}_{i\bar{u}_h}^T(k-\tilde{\tau}_h) &= \left[\xi_{(n-m+1)\bar{u}_h}(k-\tilde{\tau}_h), \cdots, \xi_{n\bar{u}_h}(k-\tilde{\tau}_h)\right] \\ & \quad (h = 1, 2, \cdots, r) \\ \boldsymbol{\theta}_i^T &= \left[\boldsymbol{a}_i^T, \boldsymbol{b}_{i1}^T, \cdots, \boldsymbol{b}_{ir}^T\right] \\ \boldsymbol{a}_i^T &= [a_{i1}, a_{i2}, \cdots, a_{in}] \\ \boldsymbol{b}_{ih}^T &= [b_{ih1}, b_{ih2}, \cdots, b_{ihm}] \qquad (h = 1, 2, \cdots, r) \end{aligned} \tag{20}$$

An initial population which is consisted of binary bit strings as candidates of the time-delays τ_h $(h = 1, 2, \cdots, r)$ is generated randomly, and the identification is carried out by repeating the genetic operations of the GA. The time-delays coding and fitness value calculating are as follows and the remaining genetic operations are similar to those in section 2.4.

(1) Coding of time-delay τ_h $(h = 1, 2, \cdots, r)$:

Code the time-delays τ_h into binary bit strings of rL bits as the following.

$$\overbrace{\underbrace{1101\cdots1010}_{\tau_1}\ \underbrace{0110\cdots0101}_{\tau_2}\ \cdots\ \underbrace{1001\cdots1100}_{\tau_r}}^{rL\,bits}$$

(2) Fitness value calculation:

(LS method)

Using the mean squares of the equation error, the fitness function is given by

$$J_{LS}(\hat{\boldsymbol{\theta}}_{LS1}, \cdots, \hat{\boldsymbol{\theta}}_{LSq}, \hat{\tilde{\tau}}_1, \cdots, \hat{\tilde{\tau}}_r) = \frac{1}{\beta + \frac{1}{N}\sum_{i=1}^{q} w_i \sum_{k=k_s+1}^{k_s+N} (\xi_{0\bar{y}_i}(k) - \mathbf{z}_i^T(k, \hat{\tilde{\tau}}_1, \cdots, \hat{\tilde{\tau}}_r)\hat{\boldsymbol{\theta}}_{LSi})^2} \tag{21}$$

(IV method)

Using the mean squares of the output error, the fitness function is given by

$$J_{IV}(\hat{\boldsymbol{\theta}}_{IV1}, \cdots, \hat{\boldsymbol{\theta}}_{IVq}, \hat{\tilde{\tau}}_1, \cdots, \hat{\tilde{\tau}}_r) = \frac{1}{\beta + \frac{1}{N}\sum_{i=1}^{q} w_i \sum_{k=k_s+1}^{k_s+N} (\bar{y}_i(k) - \hat{\tilde{x}}_i(k))^2} \tag{22}$$

where $\hat{\tilde{x}}_i(k)$ $(i = 1, 2, \cdots, q)$ are the estimated real outputs. $w_i (i = 1, 2, \cdots, q)$ in Eqs.(21),(22) are weighting factors, which are usually designed to be 1. In the case when the powers of the outputs are quite different each other, it is recommendable to adjust w_i such that each output contributes to the fitness function fairly.

4 Illustrative examples

4.1 Example 1

Consider the following single-input single-output continuous time-delay system:

$$\begin{aligned} &\ddot{x}(t) + a_1\dot{x}(t) + a_2x(t) = b_1u(t-\tau) \\ &a_1 = 3.0,\quad a_2 = 4.0,\quad b_1 = 4.0,\quad \tau = 9.130 \end{aligned} \tag{23}$$

The input signal is output of a zero-order hold driven by a white signal filtered by a second-order Butterworth filter. The sampling period is taken to be $T = 0.05$, and α in the low-pass pre-filter $Q(p)$ is 0.4. Experiments are carried out through 20 realizations of the measurement noise, when NSR (noise to signal ratio) are 5%, 20% and 50% respectively. The simulation results are shown in Table 1~Table 3. Each of tables incudes the mean and standard deviation of the estimates. The design parameters of the GA are given as follows:

(1) population size $M = 40$ (2) string length $L = 14$
(3) generation number $G = 30$
($G_{LS} = 5$ in the IV method)
(4) crossover probability $P_c = 0.8$
(5) mutation probability $P_m = 0.03$

The search range of the time-delay is $[0, (2^{14} - 1)/1000] = [0, 16.383]$, and the resolution is 0.001.

From Table 1~Table 3 it is clear that though the LS method gives accurate estimates in the presence of low measurement noise, it does not yield acceptable results in the case of high measurement noise. However the IV method is very efficient in this case. As far as we know, the identification in the presence of such a high measurement noise is seldom reported. The estimates of the system parameters are very sensitive to the time-delay, which suggests an importance of identifying the time-delay exactly. Furthermore it is shown that the method based on our approximated discrete-time estimation model gives accurate results even if the time-delay is not an integral multiple of the sampling period like this example.

4.2 Example 2

We identify the following two-input two-output time-delay system described by

$$\begin{bmatrix} x_1(t) \\ \\ x_2(t) \end{bmatrix} = \begin{bmatrix} \dfrac{b_{111}p + b_{112}}{p^2 + a_{11}p + a_{12}} & \dfrac{b_{121}p + b_{122}}{p^2 + a_{11}p + a_{12}} \\ \dfrac{b_{211}p + b_{212}}{p^2 + a_{21}p + a_{22}} & \dfrac{b_{221}p + b_{222}}{p^2 + a_{21}p + a_{22}} \end{bmatrix} \begin{bmatrix} u_1(t-\tau_1) \\ \\ u_2(t-\tau_2) \end{bmatrix}$$

$$\begin{aligned} &[a_{11}\ a_{12}\ b_{111}\ b_{112}\ b_{121}\ b_{122}] = [3.0\ \ 4.0\ \ 1.0\ \ 2.0\ \ 2.0\ \ 3.0] \\ &[a_{21}\ a_{22}\ b_{211}\ b_{212}\ b_{221}\ b_{222}] = [2.0\ \ 2.0\ \ 2.0\ \ 1.0\ \ 1.0\ \ 4.0] \\ &\tau_1 = 2.570 \quad \tau_2 = 9.130 \end{aligned} \tag{24}$$

The inputs $u_h(t)$ $(h = 1, 2)$ are filtered white signals. The sampling period is taken to be $T = 0.05$, and α in $Q(p)$ is chosen to be 0.3. Simulation is carried out when the IV method is utilized and NSR=20%. The results are shown in Table 4. The design parameters of the GA are given as follows:

(1) population size $M = 80$ (2) string length $rL = 28$
(3) generation number $G = 40$ ($G_{LS} = 5$)
(4) crossover probability $P_c = 0.8$
(5) mutation probability $P_m = 0.03$

The search ranges of τ_1, τ_2 are $[0, (2^{14} - 1)/1000] = [0, 16.383]$ respectively, and each resolution is 0.001.

Table. 4 show that the method combining the IV method with the GA yields accurate estimates of the time-delays and the system parameters.

If we attempt to identify this system by searching the time-delays one by one with accuracy of the resolution 0.001, since $2^{28} \approx 2.7 \times 10^8$ times implementations are needed, it is not computationally feasible. On the other hand, our method is feasible adequately even on a personal computer, because it requires only $M \times G = 3200$ times implementations. Moreover in the case when we apply the nonlinear estimation method to the system where the time-delays greatly differ each other like this example, it is very difficult to choose suitable initial values of the parameters. This fact shows that compared with the other nonlinear estimation method, our method is much more excellent.

5 Conclusions

In this report we have proposed the identification algorithm combining the LS or IV method with the GA for single-input single-output continuous time-delay systems, and extend our method to the identification of multi-input multi-output continuous time-delay systems which has been considered to be difficult in general. In our approximated discrete-time estimation model, the time-delay need not be an integral multiple of the sampling period. Due to the advantage of the GA, the estimates of the time-delay and the system parameters can reach global optimal point without being locked at local optima. Simulation results show that the method combining the IV method with the GA gives consistent estimates even in the presence of high measurement noises.

References

[1] W. X. Zheng and C. B. Feng: Identification of Stochastic Time Lag Systems in the Presence of Colored Noise; Automatica, Vol.26, No.4, pp.769~779 (1990)

[2] W. X. Zheng and C. B. Feng: Optimizing Search-Based Identification of Stochastic Time-Delay Systems; International Journal of Systems Science,Vol.22, No.5, pp.783~792 (1991)

[3] G. Ferretti, C. Maffezzoni and R. Scattolini: Recursive Estimation of Time Delay in Sampled Systems; Automatica,Vol.27, No.4, pp.653~661 (1991)

[4] S. Sagara, Z. J. Yang, K. Wada: Identification of Continuous-Time Systems (in Japanese); Systems, Control and Information, Vol.37, No.5, pp.284~290 (1993)

[5] P. J. Gawthrop, and M. T. Nihtilä: Identification of Time-Delays using a polynomial identification method; System and Control Letters, Vol.5, No.4, pp.267~271 (1985)

[6] E. W. Bai and D. H. Chyung: Improving Delay Estimates Derived from Least-Square Algorithms and Padé Approximations; International Journal of Systems Science, Vol.24, No.4, pp.745~756 (1993)

[7] M. Agarwal and C. Canudas: On-line Estimation of Time Delay and Continuous-Time Process Parameters; International Journal of Control, Vol.46, No.1, pp.295~311 (1987)

[8] Z.Y. Zhao and S. Sagara: Consistent Estimation of the Time Delay in Continuous-Time Systems; Transactions of the society of instrument and control engineers, Vol.27, No.1, pp.64~69 (1991)

[9] D. C. Saha and G. P. Rao: Identification of Lumped Linear Systems in the Presence of Small Unknown Delays – the Poisson Moment Functional Approach; International Journal of Control, Vol.33, No.5, pp.945~951 (1981)

[10] G. P. Rao and L. Sivakumar: Identification of Time-Lag Systems via Walsh Functions; IEEE Trans.on Automatic Control, Vol.AC-24, No.5, pp.806~808 (1979)

[11] P. J. Gawthrop, M. T. Nihtilä and A. Besharati-Rad: Recursive Parameter Estimation of Continuous-Time Systems with Unknown Time Delay; Control-Theory and Advanced Technology, Vol.5, No.3, pp.227~248 (1989)

[12] A. E. Pearson and C. Y. Wuu: Decoupled Delay Estimation in the Identification of Differential Delay Systems; Automatica, Vol.20, No.6, pp.761~772 (1984)

[13] D. E. Goldberg: Genetic Algorithms in Search, Optimization, and Machine Learning; Addison-Wesley Publishing Company, Inc. (1989)

[14] J. E. Lansberry, L.Wozniak and D. E. Goldberg: Optimal Hydrogenerator Governor Tuning with a Genetic Algorithm; IEEE Trans.on Energy Conversion, Vol.7, No.4, pp.623~630 (1992)

[15] M. Tanaka, T. Hattori and T. Tanino: Jumped Detection and Identification of Linear Systems by the Genetic algorithm (in Japanese); Transactions of the society of instrument and control engineers, Vol.28, No.11, pp.1383~1385 (1992)

[16] S. Sagara, Z. J. Yang, K. Wada and T. Tsuji: Parameter Identification and Adaptive Control of Continuous Systems with Zero-Order Hold; IFAC World Congress, Sydney (1993)

[17] S. Sagara, K. Wada and K. Nakano: Fundamentals of Digital Control (in Japanese); CORONA PUBLISHING CO.,LTD., pp.127~128 (1992)

[18] T. Nakamizo: Signal Analysis and System Identification (in Japanese); CORONA PUBLISHING CO.,LTD. (1992)

[19] Söderström,T. and Stoica,P.: Comparison of Some Instrumental Variable Methods Consistency and Accuracy Aspects; Automatica, Vol.17, No.1, pp.101~115 (1981)

Table 1: Estimates of the time-delay and system parameters (NSR=5%)

	L S method		I V method	
N	1000	2000	1000	2000
$\hat{\tau}$ (9.130)	9.130 ±0.003	9.130 ±0.002	9.131 ±0.003	9.130 ±0.002
$\hat{a}_1$ (3.0)	3.0066 ±0.0264	3.0032 ±0.0180	3.0214 ±0.0270	3.0139 ±0.0187
$\hat{a}_2$ (4.0)	4.0069 ±0.0370	4.0025 ±0.0243	4.0219 ±0.0373	4.0131 ±0.0272
$\hat{b}_1$ (4.0)	4.0121 ±0.0367	4.0058 ±0.0248	4.0292 ±0.0374	4.0182 ±0.0267

Table 2: Estimates of the time-delay and system parameters (NSR=20%)

	L S method		I V method	
N	2000	4000	2000	4000
$\hat{\tau}$ (9.130)	9.120 ±0.006	9.119 ± 0.004	9.131 ±0.005	9.131 ±0.004
$\hat{a}_1$ (3.0)	2.8462 ±0.0580	2.8358 ±0.0381	3.0356 ±0.0597	3.0239 ±0.0368
$\hat{a}_2$ (4.0)	3.8321 ±0.0819	3.8311 ±0.0547	4.0267 ±0.0808	4.0253 ±0.0523
$\hat{b}_1$ (4.0)	3.8188 ±0.0845	3.8111 ±0.0574	4.0415 ±0.0834	4.0373 ±0.0533

Table 3: Estimates of the time-delay and system parameters (NSR=50%)

	L S method		I V method	
N	2000	4000	2000	4000
$\hat{\tau}$ (9.130)	9.081 ±0.015	9.081 ±0.012	9.133 ±0.014	9.132 ±0.008
$\hat{a}_1$ (3.0)	2.2795 ±0.0997	2.2669 ±0.0690	3.0774 ±0.1527	3.0463 ±0.0867
$\hat{a}_2$ (4.0)	3.2894 ±0.1469	3.2983 ±0.1036	4.0521 ±0.2076	4.0468 ±0.1223
$\hat{b}_1$ (4.0)	3.1858 ±0.1548	3.1702 ±0.1086	4.0861 ±0.2122	4.0736 ±0.1252

Table 4: Estimates of the time-delays and system parameters (N=2000)

$\hat{\tau}_1$ (2.570)	$\hat{\tau}_2$ (9.130)				
2.571 ±0.005	9.130 ±0.005				
$\hat{a}_{11}$ (3.0)	$\hat{a}_{12}$ (4.0)	$\hat{b}_{111}$ (1.0)	$\hat{b}_{112}$ (2.0)	$\hat{b}_{121}$ (2.0)	$\hat{b}_{122}$ (3.0)
2.9804 ±0.0466	3.9003 ±0.2008	1.0113 ±0.0279	1.9381 ±0.1076	2.0077 ±0.0417	2.9176 ±0.1675
$\hat{a}_{21}$ (2.0)	$\hat{a}_{22}$ (2.0)	$\hat{b}_{211}$ (2.0)	$\hat{b}_{212}$ (1.0)	$\hat{b}_{221}$ (1.0)	$\hat{b}_{222}$ (4.0)
2.0197 ±0.0393	2.0123 ±0.0359	2.0117 ±0.0302	1.0092 ±0.0456	0.9871 ±0.0381	4.0347 ±0.0863