International Journal of Electrical, Electronic and Communication Sciences
ISSN: 2517-9438
Vol:2, No:6, 2008

On-line Identification of Continuous-time Hammerstein Systems via RBF Networks and Immune Algorithm

Tomohiro Hachino, Kengo Nagatomo, and Hitoshi Takata

***Abstract*—**This paper deals with an on-line identification method of continuous-time Hammerstein systems by using the radial basis function (RBF) networks and immune algorithm (IA). An unknown nonlinear static part to be estimated is approximately represented by the RBF network. The IA is efficiently combined with the recursive least-squares (RLS) method. The objective function for the identification is regarded as the antigen. The candidates of the RBF parameters such as the centers and widths are coded into binary bit strings as the antibodies and searched by the IA. On the other hand, the candidates of both the weighting parameters of the RBF network and the system parameters of the linear dynamic part are updated by the RLS method. Simulation results are shown to illustrate the proposed method.

***Keywords*—**Continuous-time System, Hammerstein System, On-line Identification, Immune Algorithm, RBF network.

I. Introduction

MOST practical systems have inherently nonlinear characteristics such as saturation or dead-zone and often have time-varying dynamics. The development of accurate identification method for such systems is of great importance for precise analysis, control design or prediction. One of typical approaches for nonlinear system identification is use of the block oriented models such as Hammerstein model [1]–[7]. The Hammerstein model is expressed by a memoryless nonlinear static part followed by a linear dynamic part and has many advantages for control design or stability analysis due to the model structure [1]. Several identification methods have been proposed for the Hammerstein models by using correlation theory [2], neural networks [3], orthogonal functions [4], polynomials [5], piecewise linear model [6], automatic choosing function model [7], and so on. However these identification methods are based on the discrete-time model. Generally, parameters in the discrete-time model do not directly correspond to the physical values, therefore the identification algorithm based on the continuous-time model is often desirable to carry out analysis or control design for real systems successfully. Moreover many conventional identification schemes for Hammerstein models are essentially off-line and not feasible for time-varying systems.

Manuscript received June 28, 2008.

Tomohiro Hachino is with the Department of Electrical and Electronics Engineering, Kagoshima University, Kagoshima, 890-0065 Japan (phone and fax: +81-99-285-8392; email: hachino@eee.kagoshima-u.ac.jp).

Kengo Nagatomo is with the Department of Electrical and Electronics Engineering, Kagoshima University, Kagoshima, 890-0065 Japan.

Hitoshi Takata is with the Department of Electrical and Electronics Engineering, Kagoshima University, Kagoshima, 890-0065 Japan (email: takata@eee.kagoshima-u.ac.jp).

In this paper an on-line identification method of continuous-time Hammerstein systems is proposed by using radial basis function (RBF) networks and immune algorithm (IA). The IA is an optimization technique which is inspired by the body's immune systems [8]–[10]. Comparing with the genetic algorithm, the IA can maintain the diversity of gene because it has a mechanism to promote and suppress antibodies production by introducing of the affinity between antibodies. The IA has also a memory mechanism where an effective solution (antibody) is stored in the memory cell and operates quickly when a similar antigen reappears. In the proposed method, the IA is efficiently combined with the recursive least-squares (RLS) method in order to track the time-varying system parameters and nonlinear function. The objective function for the identification is regarded as the antigen. The candidates of the centers and widths of the RBF are coded into binary bit strings as the antibodies and searched by the IA. On the other hand, the candidates of both the weighting parameters of the RBF network and the system parameters of the linear dynamic part are updated by the RLS method.

This paper is organized as follows. In section II the problem is formulated. In section III on-line identification method using the RBF network model is presented where the adjusting parameters, i.e. the centers and widths of the RBF are fixed. In section IV, the identification algorithm combining the RLS method with the IA is proposed. In section V numerical simulation is carried out for the case that the objective system changes stepwisely in time. Finally some conclusions are given in section VI.

II. Statement of Problem

Consider a single-input, single-output continuous-time nonlinear system described by the Hammerstein model shown in Fig.1:

$$\begin{cases} \displaystyle\sum_{i=0}^{n} a_i p^{n-i} y(t) = \sum_{j=0}^{r} b_j p^{r-j} x(t) \qquad (a_0 = 1, \quad n \geq r) \\ x(t) = f(u(t)) \end{cases} \tag{1}$$

where $u(t)$ and $y(t)$ are input and output signals, respectively. $x(t)$ is intermediate signal that is not accessible for measurement. $f(\cdot)$ is unknown nonlinear function. p denotes a differential operator. $A(p) = \sum_{i=0}^{n} a_i p^{n-i}$ and $B(p) = \sum_{j=0}^{r} b_j p^{r-j}$ are the denominator and numerator polynomials of the linear dynamic part, respectively. n and r are assumed to be known. The problem is to identify the system parameters $\{a_i\}$ and

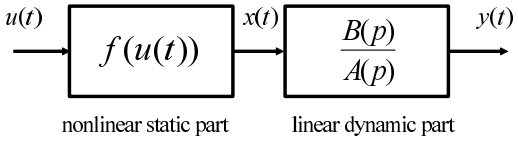


Fig. 1. Continuous-time Hammerstein model

$\{b_j\}$ of the linear dynamic part, and nonlinear static function $f(\cdot)$ on line, from input and output data.

III. Identification Model

The following state variable filter $F(p)$ is introduced in order to evaluate higher order derivatives of signals:

$$F(p)=\frac{1}{p^q+\gamma_1 p^{q-1}+\cdots\cdots+\gamma_q} \qquad (q>n). \tag{2}$$

Multiplying both sides of (1) by $F(p)$ yields

$$\begin{cases} \displaystyle\sum_{i=0}^{n} a_i p^{n-i} y^f(t) = \sum_{j=0}^{r} b_j p^{r-j} x^f(t) \\ x^f(t) = F(p) f(u(t)) \end{cases} \tag{3}$$

where

$$\begin{cases} y^f(t) = F(p) y(t) \\ x^f(t) = F(p) x(t). \end{cases} \tag{4}$$

When $F(p)$ has a transport lag characteristic, the filter $F(p)$ and the nonlinear function $f(\cdot)$ are exchangeable [11], [12] and it follows that $F(p)f(u(t)) = f(F(p)u(t)) = f(u^f(t))$. Thus (3) becomes

$$\begin{cases} \displaystyle\sum_{i=0}^{n} a_i p^{n-i} y^f(t) = \sum_{j=0}^{r} b_j p^{r-j} x^f(t) \\ x^f(t) = f(u^f(t)). \end{cases} \tag{5}$$

In general the Butterworth filter has approximately a transport lag characteristic for frequencies $\omega \le \omega_c$, where ω_c is the cutoff frequency. Therefore, the Butterworth filter modified by the all-pass filter [12] is utilized as the delayed state variable filter $F(p)$ in this paper.

The unknown nonlinear function in (5) is represented by the RBF network depicted in Fig.2 as

$$f(u^f(t))=w_0+\sum_{i=1}^{M} w_i \phi_i(u^f(t))+\varepsilon(t), \tag{6}$$

where

$$\phi_i(u^f(t))=\exp\{-(\|u^f(t)-c_i\|^2/d_i^2)\} \tag{7}$$

is the Gaussian function, M is the number of the RBF. c_i and d_i are the ith center and width of the RBF, respectively. w_i is the weighting parameter associated with the ith RBF, and $\|\cdot\|$ denotes the Euclidean norm. $\varepsilon(t)$ is an approximation error.

Substituting (6) into (5) yields the following identification model:

$$p^n y^f(t) = \boldsymbol{z}^{\mathrm{T}}(t)\boldsymbol{\theta} + v(t) \tag{8}$$

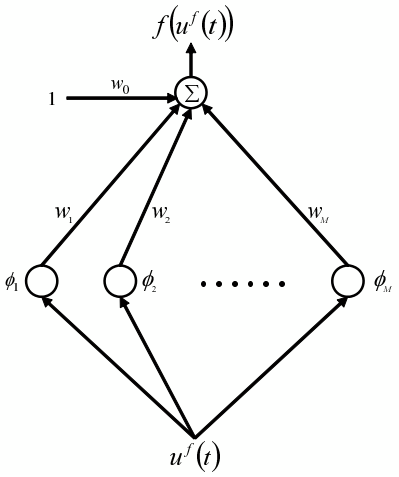


Fig. 2. RBF network

where $v(t) = \sum_{j=0}^{r} b_j p^{r-j}\varepsilon(t)$ is an equation error, and

$$\begin{cases} \boldsymbol{\theta} = [\boldsymbol{\theta}_a^{\mathrm{T}}, \boldsymbol{\theta}_{b_0}^{\mathrm{T}}, \boldsymbol{\theta}_{b_1}^{\mathrm{T}}, \cdots, \boldsymbol{\theta}_{b_r}^{\mathrm{T}}, \theta_{w_0}]^{\mathrm{T}} \\ \quad \boldsymbol{\theta}_a = [a_1, a_2, \cdots, a_n]^{\mathrm{T}} \\ \quad \boldsymbol{\theta}_{b_i} = [\theta_{b_{i,1}}, \theta_{b_{i,2}}, \cdots, \theta_{b_{i,M}}]^{\mathrm{T}} \\ \qquad = [b_i w_1, b_i w_2, \cdots, b_i w_M]^{\mathrm{T}} \\ \quad \theta_{w_0} = [b_r w_0] \\ \boldsymbol{z}(t) = [\boldsymbol{z}_a^{\mathrm{T}}(t), \boldsymbol{z}_{b_0}^{\mathrm{T}}(t), \boldsymbol{z}_{b_1}^{\mathrm{T}}(t), \cdots, \boldsymbol{z}_{b_r}^{\mathrm{T}}(t), 1]^{\mathrm{T}} \\ \quad \boldsymbol{z}_a(t) = [-p^{n-1}y^f(t), -p^{n-2}y^f(t), \cdots, \\ \qquad\qquad -y^f(t)]^{\mathrm{T}} \\ \quad \boldsymbol{z}_{b_i}(t) = [p^{r-i}\phi_1(u^f(t)), p^{r-i}\phi_2(u^f(t)), \cdots, \\ \qquad\qquad p^{r-i}\phi_M(u^f(t))]^{\mathrm{T}}. \end{cases} \tag{9}$$

Applying the RLS method to (8), the unknown parameter vector $\boldsymbol{\theta}$ can be estimated recursively as follows:

$$\begin{aligned} \widehat{\boldsymbol{\theta}}(k) &= \widehat{\boldsymbol{\theta}}(k-1) + \boldsymbol{L}(k)\epsilon(k) \\ \epsilon(k) &= p^n y^f(k) - \boldsymbol{z}^T(k)\widehat{\boldsymbol{\theta}}(k-1) \\ \boldsymbol{L}(k) &= \frac{\boldsymbol{P}(k-1)\boldsymbol{z}(k)}{1+\boldsymbol{z}^T(k)\boldsymbol{P}(k-1)\boldsymbol{z}(k)} \\ \boldsymbol{P}(k) &= \boldsymbol{P}(k-1) - \\ &\quad \frac{\boldsymbol{P}(k-1)\boldsymbol{z}(k)\boldsymbol{z}^T(k)\boldsymbol{P}(k-1)}{1+\boldsymbol{z}^T(k)\boldsymbol{P}(k-1)\boldsymbol{z}(k)} \end{aligned} \tag{10}$$

where kT (T: sampling period) is written as k for simplicity. Thus the weighting parameters of the RBF are estimated by

$$\begin{cases} [\hat{w}_1(k), \hat{w}_2(k), \cdots, \hat{w}_M(k)]^{\mathrm{T}} = \hat{\boldsymbol{\theta}}_{b_r}(k) \\ \hat{w}_0(k) = \hat{\theta}_{w_0}(k) \end{cases} \tag{11}$$

putting $\hat{b}_r(k) = 1$ without loss of generality. Therefore the nonlinear static function is composed by (11) as

$$\hat{f}(u^f(k))=\hat{w}_0(k)+\sum_{i=1}^{M}\hat{w}_i(k)\phi_i(u^f(k)). \tag{12}$$

The denominator parameters of the linear part have been already obtained as $\hat{\boldsymbol{\theta}}_a(k)$, and the numerator parameters of the linear part are estimated by using the linear least-squares technique again as

$$\hat{b}_i(k) = \sum_{j=1}^{M} \hat{\theta}_{b_{r,j}}(k)\hat{\theta}_{b_{i,j}}(k) / \sum_{j=1}^{M} \hat{\theta}^2_{b_{r,j}}(k) \qquad (i = 0, 1, \cdots, r-1). \tag{13}$$

IV. On-line Identification by RLS Method and IA

The accuracy of the above on-line identification algorithm greatly depends on the adjusting parameters of the RBF, i.e. the centers and widths of the RBF. In this section $\boldsymbol{\Omega} = [\{c_i\}, \{d_i\}]$ is determined on line by using the IA. The objective function for the identification, mean square of the equation error, is regarded as the antigen. The candidates of $\boldsymbol{\Omega}$ are coded into binary bit strings as the antibodies and searched by the IA. The candidates of both the weighting parameters of the RBF network and the system parameters of the linear part are updated by the RLS method mentioned in section III.

The proposed one-line identification algorithm is as follows:

step 1: Initialization

Generate an initial population of antibodies which consists of Q binary bit strings $\boldsymbol{\Omega}_l^a[0]$ $(l = 1, 2, \cdots, Q)$ and an initial population of memory cells which consists of R binary bit strings $\boldsymbol{\Omega}_l^m[0]$ $(l = 1, 2, \cdots, R)$ randomly.

Set the initial value $\hat{\boldsymbol{\theta}}(0)$ and $\boldsymbol{P}(0)$ for the RLS method.

Let the generation index of the IA $g = 0$ and the time step index $k = 1$.

step 2: Decoding

Decode Q strings into real values $\tilde{\boldsymbol{\Omega}}_l^a[g]$ for antibodies and R strings into real values $\tilde{\boldsymbol{\Omega}}_l^m[g]$ for memory cells.

step 3: Filtering

Calculate $u^f(k)$, $y^f(k)$, and their higher order derivatives by using the delayed state variable filter.

step 4: Construction of the signal vectors

Calculate the signal vectors $\boldsymbol{z}_l^a(k)$ and $\boldsymbol{z}_l^m(k)$ by using $\tilde{\boldsymbol{\Omega}}_l^a[g]$ and $\tilde{\boldsymbol{\Omega}}_l^m[g]$, respectively.

step 5: Estimation of $\hat{\boldsymbol{\theta}}$

Update $\hat{\boldsymbol{\theta}}_l^a(k)$ and $\hat{\boldsymbol{\theta}}_l^m(k)$ using the RLS method in (10).

step 6: Fitness value calculation

Calculate the values of the objective function (antigen) for both antibodies and memory cells by using $\tilde{\boldsymbol{\Omega}}_l^a[g]$ and $\tilde{\boldsymbol{\Omega}}_l^m[g]$, and corresponding $\hat{\boldsymbol{\theta}}_l^a(k)$ and $\hat{\boldsymbol{\theta}}_l^m(k)$:

$$J_l^a(\tilde{\boldsymbol{\Omega}}_l^a[g], \hat{\boldsymbol{\theta}}_l^a(k)) = \frac{1}{W}\sum_{j=0}^{W-1}\Big\{p^n y^f(k-j) - \boldsymbol{z}_l^{a\mathrm{T}}(k-j)\hat{\boldsymbol{\theta}}_l^a(k)\Big\}^2 \qquad (l = 1, 2, \cdots, Q) \tag{14}$$

$$J_l^m(\tilde{\boldsymbol{\Omega}}_l^m[g], \hat{\boldsymbol{\theta}}_l^m(k)) = \frac{1}{W}\sum_{j=0}^{W-1}\Big\{p^n y^f(k-j) - \boldsymbol{z}_l^{m\mathrm{T}}(k-j)\hat{\boldsymbol{\theta}}_l^m(k)\Big\}^2 \qquad (l = 1, 2, \cdots, R), \tag{15}$$

where W is the time window length.

Compute the affinity between the antigen and each of antibodies, i.e. the fitness value of the antibody $\boldsymbol{\Omega}_l^a[g]$ and $\hat{\boldsymbol{\theta}}_l^a(k)$:

$$F_l^a = 1/J_l^a \qquad (l = 1, 2, \cdots, Q) \tag{16}$$

and the affinity between the antigen and each of memory cells, i.e. the fitness value of the memory cell $\boldsymbol{\Omega}_l^m[g]$ and $\hat{\boldsymbol{\theta}}_l^m(k)$:

$$F_l^m = 1/J_l^m \qquad (l = 1, 2, \cdots, R). \tag{17}$$

step 7: Determination of the estimated model

Determine the estimated model at the current time step k with $\hat{\boldsymbol{\Omega}}[g] = [\{\hat{c}_i[g]\}, \{\hat{d}_i[g]\}]$ which has the best fitness value among the antibodies or memory cells and corresponding $\hat{\boldsymbol{\theta}}(k)$ and $\hat{f}(u(k))$.

step 8: Repetition for the RLS method

If $k \neq \lambda G$ $(\lambda = 1, 2, \cdots)$ where G is prespecified integer, set the time step index $k = k + 1$ and go to step 3.

step 9: Density calculation

Calculate the affinity $ay_{l,w}$ between the antibody $\boldsymbol{\Omega}_l^a[g]$ and $\boldsymbol{\Omega}_w^a[g]$

$$ay_{l,w} = \frac{1}{1 + H_{l,w}} \tag{18}$$

where $H_{l,w}$ is the normalized Hamming distance between two antibodies.

Compute the density C_l^a for each antibody $\boldsymbol{\Omega}_l^a[g]$ among the population of antibodies:

$$C_l^a = \frac{1}{Q}\sum_{w=1}^{Q} ac_{l,w}, \qquad ac_{l,w} = \begin{cases} 1 & ay_{l,w} \geq T_{ac1} \\ 0 & \text{otherwise} \end{cases} \tag{19}$$

where T_{ac1} is the threshold for the density calculation.

step 10: Differentiation into the memory cells

Differentiate the best $\hat{\boldsymbol{\Omega}}[g]$ into the memory cell and remove a randomly selected memory cell from the population of the memory cells, if the maximum affinity between $\hat{\boldsymbol{\Omega}}[g]$ and each of the memory cells is below the threshold T_{ac2}.

step 11: Promotion and suppression for antibody production

Calculate the expected value for each of antibody to survive to the next generation:

$$E_l^a = F_l^a / C_l^a \qquad (l = 1, 2, \cdots, Q). \tag{20}$$

Reproduce each of antibodies in proportion to its expected value. Each of antibodies is reproduced with the probability of $E_l^a / \sum_{j=1}^{Q} E_j^a$, where the worst 30% antibodies are exchanged with randomly generated antibodies.

step 12: Crossover

Pick up two antibodies randomly and decide whether to cross them over or not according to the crossover probability P_c. If a crossover required, exchange genes at a crossing position selected randomly.

step 13: Mutation

Alter a gene of antibody ("0" or "1") according to the mutation probability P_m.

step 14: Repetition for the IA

Increase the generation index as $g = g + 1$ and the time step index as $k = k + 1$, and go to step 2.

V. Illustrative Example

Consider a system described by the following Hammerstein system:

$$\begin{cases} \ddot{y}(t) + a_1\dot{y}(t) + a_2 y(t) = b_0 x(t) \\ x(t) = f(u(t)) \end{cases} \quad (21)$$

where the system parameters of the linear dynamic part and the nonlinear static part change stepwisely as shown in Table I.

The output signal is generated by a random signal of band-pass 1.0[rad/s]. The sampling period is taken to be $T = 0.25$[s] and the identification is carried out until $t = 400$[s]. The third-order Butterworth filter modified by the all-pass filter is utilized as the delayed state variable filter, whose cutoff frequency is $\omega_c = 5.0$[rad/s]. The number of the RBF is $M = 4$. The time window length in (14) and (15) is taken to be $W = 50$. G in step 8 of section IV is 50. The initial values for the RLS method are chosen as follows:

$\hat{\boldsymbol{\theta}}(0) = \mathbf{1}$
$\boldsymbol{P}(0) = 10^3\boldsymbol{I}$ ($\boldsymbol{I}$: unit matrix).

The design parameters of the IA are given as follows:

the number of antibodies $Q = 60$
the number of memory cells $R = 30$
crossover probability $P_c = 0.8$
mutation probability $P_m = 0.05$
search range of the RBF centers
$[c_{min}, c_{max}] = [-2.81, 3.19]$
search range of the RBF widths
$[d_{min}, d_{max}] = [0.01, 20]$
threshold $T_{ac1} = 0.70$, $T_{ac2} = 0.95$.

Fig.3 shows the true system parameters a_1 and a_2, and estimated system parameters $\hat{a}_1$ and $\hat{a}_2$. Since the estimated model is normalized by $\hat{b}_0$, estimation result of b_0 is omitted in Fig.3. It can be confirmed that the proposed method can track the time-varying system parameters well.

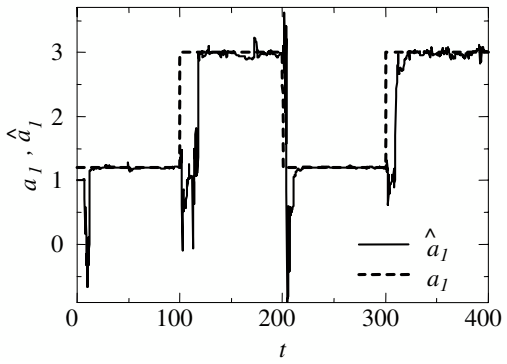
The estimated nonlinear static functions at the time $t = 100, 200, 300$, and 400[s] are depicted in Fig.4. Clearly the estimated nonlinear functions at every step are very close to the true nonlinear functions on the data region.

TABLE I
System Parameters and Nonlinear Function

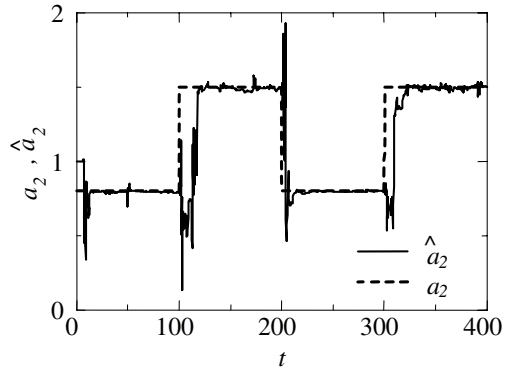
t [s]	a_1	a_2	b_0	$f(u(t))$
[0, 100] (200, 300]	1.20	0.80	1.0	$0.5u(t) + u^2(t)$
(100, 200] (300, 400]	3.0	1.50	1.0	$u(t) + 0.5u^3(t)$

VI. Conclusions

In this paper we have presented an on-line identification method of continuous-time Hammerstein systems by the RBF networks and IA. The IA is efficiently combined with the RLS method in order to track the time-varying system parameters of the linear dynamic part and nonlinear static part. Simulation results show that the propose method can be easily applied to the continuous-time Hammerstein systems in case that the system parameters of the linear dynamic part and nonlinear static part change stepwisely in time. An examination for the identification of noisy systems will be one of the future works.



(a) a_1

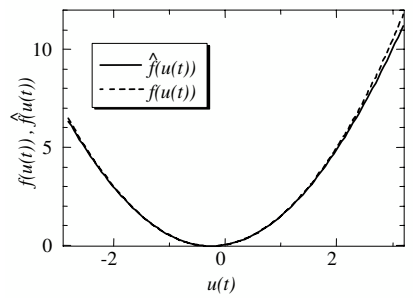


(b) a_2

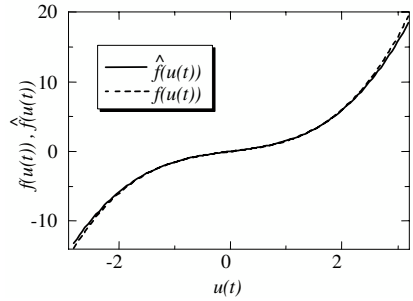
Fig. 3. True system parameters and estimated system parameters

References

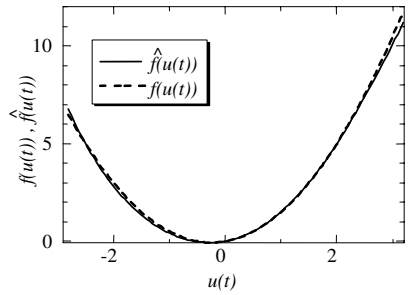
[1] O. Nelles, *Nonlinear System Identification*, Springer, 2000.
[2] S. A. Billings and S. Y. Fakhouri, "Identification of systems containing linear dynamic and static nonlinear elements", *Automatica*, Vol.18, No.1, pp.15–26, 1982.
[3] H. Al-Duwaish and M. N. Karim, "A new method for the identification of Hammerstein model", *Automatica*, Vol.33, No.10, pp.1871–1875, 1997.
[4] F. C. Kung and D. H. Shih, "Analysis and identification of Hammerstein model non-linear delay systems using block-pulse function expansions", *Int. J. Control*, Vol.43, No.1, pp.139–147, 1986.
[5] S. Adachi and H. Murakami, "Generalized predictive control system design based on non-linear identification by using Hammerstein model (in Japanese)", *Trans. of the Institute of Systems, Control and Information Engineers*, Vol.8, No.3, pp.115–121, 1995.
[6] T. Hatanaka, K. Uosaki, and M. Koga, "Evolutionary computation approach to Hammerstein model identification", *Proc. of the 4th Asian Control Conference*, pp.1730–1735, 2002.
[7] T. Hachino and H. Takata, "Structure selection and identification of Hammerstein type nonlinear systems using automatic choosing function model and genetic algorithm", *IEICE Trans. on Fundamentals of Electronics, Communications and Computer Sciences*, Vol.E88-A, No.10, pp.2541–2547, 2005.



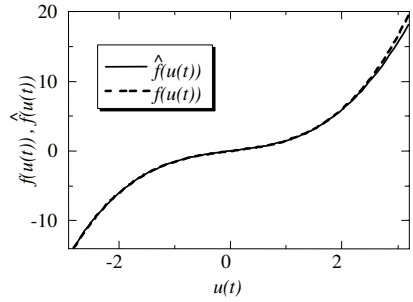
(a) $t = 100$[s]



(b) $t = 200$[s]



(c) $t = 300$[s]



(d) $t = 400$[s]

Fig. 4. Estimated nonlinear functions

[8] J. D. Farmer, N. H. Pakard, and A. S. Perelson, "The immune system, adaptation, and machine learning", *Physica 22D*, pp.187-204, 1986.
[9] Y. Ishida, H. Hirayama, H. Fujita, A. Ishiguro, and K. Mori, *Immunity-Based Systems and Its Applications - Intelligent Systems by Artificial Immune Systems (in Japanese)*, CORONA publishing co., 1998.
[10] K. Mori, M. Tsukiyama, and T. Fukuda, "Immune algorithm with searching diversity and its application to resource allocation problem (in Japanese)", *IEEJ Trans. on Electronics, Information and Systems*, Vol.113-C, No.10, pp.872–878, 1993.
[11] R. H. Kohr, "A method for the determination of a differential equation model for simple nonlinear systems", *IEEE Trans. on Electronic Computers*, Vol.12, pp.394–400, 1963.
[12] K. M. Tsang and S. A. Billings, "Identification of continuous time nonlinear systems using delayed state variable filters", *Int. J. Control*, Vol.60, No.2, pp.159–180, 1994.

Tomohiro Hachino received the B.S., M.S. and Dr. Eng. degrees in Electrical Engineering from Kyushu Institute of Technology, in 1991, 1993 and 1996, respectively. He is currently an associate professor in the Department of Electrical and Electronics Engineering, Kagoshima University. His research interests include control and identification for nonlinear systems. Dr. Hachino is a member of IEEJ, SICE, and ISCIE.

Kengo Nagatomo received the B.S. and M.S. degrees in Electrical and Electronics Engineering from Kagoshima University, in 2006 and 2008, respectively. In 2008, he joined NEC Corporation. His research interests include control and identification for nonlinear systems.

Hitoshi Takata received the B.S. degree in Electrical Engineering from Kyushu Institute of Technology in 1968, and the M.S. and Dr. Eng. degrees in Electrical Engineering from Kyushu University, in 1970 and 1974, respectively. He is currently a professor in the Department of Electrical and Electronics Engineering, Kagoshima University. His research interests include control, linearization, and identification for nonlinear systems. Dr. Takata is a member of IEEE, IEEJ, SICE, and ISCIE.