

# An Identification Method for a Block Oriented Model

X.P. Xu, F. Dai

School of Sciences  
Xi'an University of Technology  
Xi'an, China

F. Wang

School of Mathematics and Statistics  
Xi'an Jiaotong University  
Xi'an, China

F.C. Qian

School of Automation and Information Engineering  
Xi'an University of Technology  
Xi'an, China

**Abstract**—A block oriented nonlinear system consists of a series of blocks. These blocks represent both memoryless nonlinearity and linear dynamics that comprise the overall input-output dynamics of system. Under this category, Hammerstein model is one of the block oriented model. The parameter identification method of Hammerstein model is proposed in this paper. The basic idea is that the nonlinear transfer function of the Hammerstein model can be changed to an intermediate model initially. Then, the parameters of the intermediate model are obtained via an improved particle swarm optimization algorithm. Next, through the relations of the parameters of the intermediate model and those of the Hammerstein model, the estimates of the parameters of the Hammerstein model are obtained. Finally, in simulation experiments, compared with other methods, the feasibility of the presented method is demonstrated.

**Keywords**—nonlinear system; block oriented model; system identification; intelligent algorithm

## I. INTRODUCTION

Recently, a considerable amount of research has been carried out on modelling, identification, and control of the nonlinear systems. Most dynamical systems can be better expressed by the nonlinear models, which can describe the global behavior of the nonlinear system over the whole operating range, rather than by linear ones that are only able to approximate the nonlinear system around a given operating point. One of the most frequently studied classes of the nonlinear models are the so-called block oriented nonlinear models [1, 2]. Under this category, the nonlinear systems are represented as Wiener, Hammerstein, Wiener-Hammerstein, and Hammerstein-Wiener type of models, etc. These block oriented nonlinear model structures have been successfully used to represent nonlinear systems in a number of practical applications in the areas of chemical processes [3], control [4], compression techniques [5], nonlinear aeroelastic/aeroservoelastic modelling [6], etc.

The objective of this article is to investigate the parameter estimation problem of Hammerstein model. Firstly, the nonlinear transfer function of the Hammerstein model can be converted to an intermediate linear model based on the function expansion. Then, the parameters of the intermediate model are obtained by an improved particle swarm

optimization algorithm. Next, through the relations of the parameters of the intermediate model and those of the Hammerstein model, the parameters of the Hammerstein model are identified simultaneously. Finally, the results of simulation experiments are used to demonstrate the effectiveness of the presented identification approach.

## II. PARTICLE SWARM OPTIMIZATION ALGORITHM

Particle swarm optimization is a population-based stochastic optimization algorithm, firstly introduced by Kennedy and Eberhart in 1995 [7]. Although particle swarm optimization algorithm is often ranged in evolutionary computation, it is actually an incarnation of swarm intelligence. It was developed through simulation of a simplified social system, and has been found to be robust in solving continuous nonlinear optimization problems. The main advantages of the particle swarm optimization algorithm are summarized as: simple concept, easy implementation, robustness to control parameters, and computational efficiency when compared with mathematical algorithm and other heuristic optimization techniques. Thereby, recently, particle swarm optimization algorithm has been successfully applied to various fields [8-10].

### A. Basic Particle Swarm Optimization Algorithm

Suppose that the search space is  $D$ -dimensional, then the  $i$ -th particle of the swarm can be represented by a  $D$ -dimensional vector,  $X_i = (x_{i1}, x_{i2}, \dots, x_{iD})^T$ . The velocity of this particle, can be represented by another  $D$ -dimensional vector  $V_i = (v_{i1}, v_{i2}, \dots, v_{iD})^T$ . The best previously visited position of the  $i$ -th particle is denoted as  $P_i = (p_{i1}, p_{i2}, \dots, p_{iD})^T$ . Defining  $g$  as the index of the best particle in the swarm, and let the superscripts denote the iteration number, then the swarm is manipulated according to the following two equations.

$$v_{id}^{n+1} = v_{id}^n + cr_1^n (p_{id}^n - x_{id}^n) + cr_2^n (p_{gd}^n - x_{id}^n), \quad (1)$$

$$x_{id}^{n+1} = x_{id}^n + v_{id}^{n+1}, \quad (2)$$

where  $d = 1, 2, \dots, D$ ;  $i = 1, 2, \dots, N$ , and  $N$  is the size of the swarm;  $c$  is a positive constant, called acceleration constant;  $r_1$ ,

$r_2$  are random numbers, uniformly distributed in  $[0, 1]$ ; and  $n = 1, 2, \dots$ , determines the iteration number.

### B. Improved Particle Swarm Optimization Algorithm

In basic particle swarm optimization algorithm, the lack of a control mechanism for the velocity resulted in low efficiency for particle swarm optimization, compared to Evolutionary Computation techniques. Specifically, particle swarm optimization located the area of the optimum faster than Evolutionary Computation techniques, but once in the region of the optimum, it could not adjust its velocity step size to continue the search at a finer grain. The problem was addressed by incorporating a weight parameter for the previous velocity of the particle. Thus, in the latest versions of the particle swarm optimization, Eq. (1) is changed to the following ones [11].

$$v_{id}^{n+1} = \chi(wv_{id}^n + c_1r_1^n(p_{id}^n - x_{id}^n) + c_2r_2^n(p_{gd}^n - x_{id}^n)), \quad (3)$$

where  $w$  is called inertia weight;  $c_1, c_2$  are two positive constants, called cognitive and social parameter respectively; and  $\chi$  is a constriction factor, which is used, alternatively to  $w$  to limit velocity. The role of the inertia weight  $w$ , in Equation (3), is considered critical for the convergence behavior of the particle swarm optimization algorithm.  $w$  is employed to control the impact of the previous history of velocities on the current one. Accordingly, the parameter  $w$  regulates the trade-off between the global and local exploration abilities of the swarm. A large  $w$  facilitates global exploration; while a small one tends to facilitate local exploration. A suitable value for  $w$  usually provides balance between global and local exploration abilities and consequently results in a reduction of the number of iterations required to locate the optimum solution. Initially,  $w$  was constant. However, experimental results indicated that it is better to initially set the inertia to a large value, to promote global exploration of the search space, and gradually decrease it to get more refined solutions. Thus, in this paper,  $w$  is set according to the following equation.

$$w = w_{\max} - \frac{w_{\max} - w_{\min}}{iter_{\max}} \times iter, \quad (4)$$

where  $iter_{\max}$  is the maximum number of iterations, and  $iter$  is the current number of iterations,  $w_{\max}$  and  $w_{\min}$  are the initial and final inertia weight. The search procedures of the improved algorithm are as shown below.

Step 1. Let initialization iterative number be  $iter = 1$ , population size be  $N$ , the termination iterative number be  $iter_{\max}$ , the value of the initial inertia weight factor be  $w_{\max}$ , the value of the final weight factor be  $w_{\min}$ , and etc.

Step 2. Initialize a population of particles with random positions and velocities in the  $D$ -dimension space by uniform probability distribution function.

Step 3. Evaluate particle's fitness value.

Step 4. Compare each particle's fitness value with the particle's  $p_i$ . If the current fitness value is better than  $p_i$ , then

set the  $p_i$  value equals to the current fitness value and the  $p_i$  location equals to the current location in space.

Step 5. Compare the fitness value with the population's overall previous best. If the current fitness value is better than  $p_g$ , reset  $p_g$  to the current particle's array index and value.

Step 6. Change the velocity,  $v_i$ , and position of the particle,  $x_i$ , according to Eqs. (3) and (2), respectively.

Step 7. Change the inertia weight factor,  $w$ , according to Eq. (4).

Step 8. Return to Step 3 until a stop criterion (e.g., a sufficiently good fitness value, a maximal number of iterations) is met.

### III. IDENTIFICATION PROCESS

Consider a single-input single-output Hammerstein model shown in Figure 1. Let  $u(t)$ ,  $y(t)$  and  $v(t)$  be a measurement input, the system output and an noise, respectively,  $x(t)$  be an intermediate input signal.

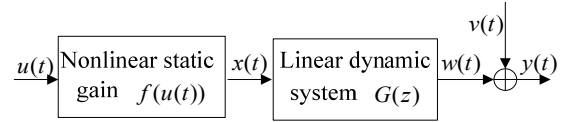


FIGURE 1. HAMMERSTEIN MODEL

The nonlinear static gain can be approximately expressed by Eq. (5).

$$x(t) = f(u(t)) = r_1u(t) + r_2u^2(t) + \dots + r_pu^p(t) = R^T U(t). \quad (5)$$

The following form of the transfer function can express the linear dynamic system.

$$G(z) = \frac{y(t)}{x(t)} = \frac{B(z^{-1})}{A(z^{-1})} = \frac{b_0 + b_1z^{-1} + \dots + b_mz^{-m}}{1 + a_1z^{-1} + \dots + a_nz^{-n}} \quad (6)$$

where  $m, n$  are the polynomial's order, and  $m \leq n$  generally.

The linear transfer function of Eq. (6) can be described by the following difference equation.

$$A(z^{-1})y(t) = B(z^{-1})x(t) + e(t), \quad (7)$$

where  $e(t) = A(z^{-1})v(t)$  can be interpreted as a random fitting error.

It follows from Eqs. (5) and (7) that

$$A(z^{-1})y(t) = \sum_{j=0}^m \sum_{i=1}^p \alpha_{ij} z^{-j} u^i(t) + e(t), \quad (8)$$

where  $\alpha_{ij} = r_i b_j$ ,  $i=1, 2, \dots, p, j=1, 2, \dots, m$ .

From Eq. (8), we have the following formula.

$$y(t) = -\sum_{i=1}^n a_i y(t-i) + \sum_{j=0}^m \sum_{i=1}^p \alpha_{ij} u^i(t-j) + e(t) \quad (9)$$

And Eq. (9) can be expressed by following vectors form.

$$y(t) = \theta \varphi(t) + e(t), \quad (10)$$

where

$$\theta = (-a_1, -a_2, \dots, -a_n, \alpha_{10}, \alpha_{11}, \dots, \alpha_{1m}, \alpha_{20}, \alpha_{21}, \dots, \alpha_{2m}, \dots, \alpha_{p0}, \alpha_{p1}, \dots, \alpha_{pm}),$$

$$\varphi(t) = (y(t-1), y(t-2), \dots, y(t-n), u(t), u(t-1), \dots, u(t-m), u^2(t), u^2(t-1), \dots, u^2(t-m), \dots, u^p(t), u^p(t-1), \dots, u^p(t-m))^T$$

Obviously,  $y(t)$  is a function of the power of  $u(t)$ , and is called an intermediate model throughout this paper. It is assumed that the assessment value of the parameter vector  $\theta$  in the intermediate model is

$$\hat{\theta} = (-\hat{a}_1, -\hat{a}_2, \dots, -\hat{a}_n, \hat{\alpha}_{10}, \hat{\alpha}_{11}, \dots, \hat{\alpha}_{1m}, \hat{\alpha}_{20}, \hat{\alpha}_{21}, \dots, \hat{\alpha}_{2m}, \dots, \hat{\alpha}_{p0}, \hat{\alpha}_{p1}, \dots, \hat{\alpha}_{pm}).$$

Thus the deviation of the assessment can be judged by Eq. (11).

$$J_h = \sum_{k=k_0}^{k_0+h} [y(k) - \hat{y}(k)]^2, \quad (11)$$

where  $h$  is the window width of identification,  $\hat{y}(k)$  are the input values of the obtained assessment model.

We can solve the minimum of Eq. (11) and obtain the corresponding parameter vector  $\theta$  of the intermediate model based on the proposed particle swarm optimization algorithm.

The parameters of the Hammerstein model are obtained by the intermediate parameters as follows. Without loss of generality, suppose that the final gain of the linear dynamic subsystem of the Hammerstein model is 1, namely,

$$G(\infty) = \frac{b_0 + b_1 + \dots + b_m}{1 + a_1 + a_2 + \dots + a_n} = 1 \quad (12)$$

Thereby, Eq. (12) can be expressed by Eq. (13).

$$C^T B = C^T A, \quad (13)$$

where  $C = (1, 1, \dots, 1)^T$ ,  $A = (1, a_1, \dots, a_n)^T$ ,  $B = (b_0, b_1, \dots, b_m)^T$ . Then

$$H = RB^T, \quad (14)$$

where  $R = (r_1, r_2, \dots, r_p)^T$ , and  $H = \begin{pmatrix} \alpha_{10} & \alpha_{11} & \dots & \alpha_{1m} \\ \alpha_{20} & \alpha_{21} & \dots & \alpha_{2m} \\ \vdots & \vdots & \dots & \vdots \\ \alpha_{p0} & \alpha_{p1} & \dots & \alpha_{pm} \end{pmatrix}$ .

Right-multiplying  $C$  in each side of Eq. (14) gives

$$HC = RB^T C. \quad (15)$$

Consequently, it follows from Eq. (15),  $B^T C = C^T B$  and Eq. (13) that

$$R = \frac{HC}{B^T C} = \frac{HC}{C^T B} = \frac{HC}{C^T A}. \quad (16)$$

Transposing each side of Eq. (14) yields

$$H^T = BR^T. \quad (17)$$

Right-multiplying  $C$  in each side of Eq. (17) gives

$$H^T C = BR^T C. \quad (18)$$

Thereby, by Eq. (18),  $R^T C = C^T R$  and Eq. (16), we have

$$B = \frac{C^T A H^T C}{C^T H C}. \quad (19)$$

To sum up, firstly, we solve Eq. (11) to get the parameter vector  $\theta$  of the intermediate model using the improved algorithm, by which the estimates of  $A$  and  $H$  can be realized by Eqs. (14) and (13). Then, the estimates of  $R$  and  $B$  are fulfilled from Eqs. (16) and (19). Consequently, estimates of the parameters of Hammerstein model are obtained.

#### IV. SIMULATION EXPERIMENTS

Consider the following Hammerstein model.

$$x(t) = 0.3u(t) + 0.9u^2(t), \quad G(z) = \frac{0.5 + 1.4z^{-1} + 0.3z^{-2} - 1.1z^{-3}}{1 - 0.2z^{-1} - 0.3z^{-2} + 0.4z^{-3}},$$

and  $v(t)$  is Gaussian noise with zero mean and standard deviation,  $\sigma = 0.04$ ,  $u(t)$  is Gaussian noise with zero mean and standard deviation,  $\sigma = 1$ .

In the identification process for  $\theta$ , the parameter values of the identification algorithm are set as follows. Let window width of estimation:  $h=50$ ,  $N=20$ ,  $w_{max}$  is 0.9,  $w_{min}$  is 0.4,  $iter_{max}$  is 1000,  $c_1$  and  $c_2$  are 1.48 and 2, respectively,  $V_{max}$  is 2, and system model parameters' initial values are all picked out randomly from  $[-0.4, 1.5]$ .

The parameter vector  $\theta$  of the intermediate model can be obtained by using the improved particle swarm optimization algorithm as follows.

$$\theta = (0.2008, 0.2997, -0.3984, 0.1489, 0.4176, 0.0887, -0.3286, 0.4495, 1.2610, 0.2677, -0.9920)$$

Then, the following parameter matrices  $H$ ,  $A$  are formed via Eqs. (13) and (14).

$$H = \begin{pmatrix} 0.1489 & 0.4176 & 0.0887 & -0.3286 \\ 0.4495 & 1.2610 & 0.2677 & -0.9920 \end{pmatrix}, \quad A = (1 \quad -0.2008 \quad -0.2997 \quad 0.3984)^T.$$

Next, the parameter vector  $R$  can be obtained based on Eq. (16).

$$R = (0.2987 \quad 0.9015)^T.$$

Finally, according to Eq. (19) the parameter vector B is  $B = (0.4986 \ 1.3988 \ 0.2969 \ -1.1004)^T$ .

From the above simulation results, it can be seen that the estimates of the parameters by the improved particle swarm optimization algorithm are almost closer to the true values. Moreover, the step response curves of the real model and the identified model are both shown in Figure 2, respectively. And the corresponding error curve is shown in Figure 3. From the figures, it can be seen that the two curves are almost overlapped, which illustrates that the identification results by the proposed algorithm is good.

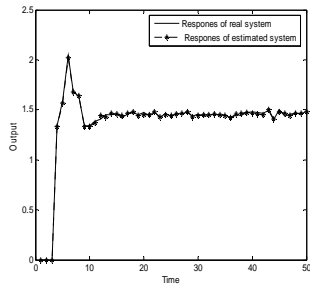


FIGURE II. THE STEP RESPONSE

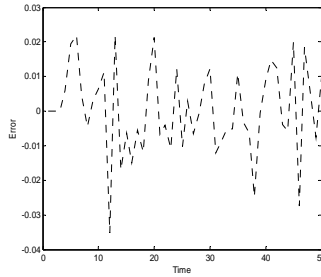


FIGURE III. ERROR CURVE

It can be seen that Eq. (10) is similar to a typical form of a linear model through the function expansion. So the following results of the identification are obtained by the extended least squares recursive method.

$$\theta = (0.1734, 0.2868, -0.3845, 0.1458, 0.3939, 0.0867, -0.3137, 0.4642, 1.2541, 0.2761, -0.9988),$$

$$H = \begin{pmatrix} 0.1458 & 0.3939 & 0.0867 & -0.3137 \\ 0.4642 & 1.2541 & 0.2761 & -0.9988 \end{pmatrix},$$

$$A = (1 \ -0.1734 \ -0.2868 \ 0.3845)^T,$$

$$R = (0.2798 \ 0.8909)^T, \quad B = (0.5210 \ 1.4077 \ 0.3099 \ -1.1211)^T.$$

Compared the above identification results, it can be seen that the identification accuracy of the presented method obviously improved. In the simulation, the presented method is easily implemented.

## V. CONCLUSIONS

This paper presented an identification method for the single-input single-output Hammerstein. Moreover, compared

with other method, the results show that the presented method has the better precision and robustness, and it meets the practical engineering problems. At the same time, the application of the particle swarm optimization algorithm is further developed.

## ACKNOWLEDGEMENT

This work is supported by National Natural Science Foundation of China under Grant No. 61273127 and 91120014, Specialized Research Fund for the Doctoral Program of Higher Education under Grant No. 20116118110008, the Scientific Research Program Funded by Shaanxi Provincial Education Department under Grant No. 14JK1538, the Project Supported by Shaanxi Provincial Natural Science Foundation research of China under Grant No. 2014JM8325, the Doctoral Scientific Research Start-up Funds of Teachers of Xi'an University of Technology of China under Grant No. 108-211006 and the Technology Project of Xi'an City of China under Grant No. CXY1345(3).

## REFERENCES

- [1] Gómez C. & Baeyens E., Identification of block-oriented systems using orthonormal bases. *Journal of Process Control*, 14(3), pp. 685-697, 2004.
- [2] Kibangou A.Y. & Favier G., Tensor analysis-based model structure determination and parameter estimation for block-oriented nonlinear systems. *IEEE Journal of topics in Signal Processing*, 4(3), pp. 514-525, 2010.
- [3] Bloemen H., Chou C., Boom T., et al., Wiener model identification and predictive control for dual composition control of a distillation column. *Journal of Process Control*, 11(3), pp. 601-620, 2001.
- [4] Fruzzetti K., Palazoglu A. & McDonald K., Nonlinear model predictive control using Hammerstein models. *Journal of Process Control*, 7(1), pp. 31-41, 1997.
- [5] Ng K. & Ravishanker V., Block-oriented compression techniques for large statistical databases. *IEEE Transactions on Knowledge and Data Engineering*, 9(2), pp. 314-328, 1997.
- [6] Baldelli H., Lind R. & Brenner M., Nonlinear aeroelastic/aeroservoelastic modeling by block-oriented identification. *Journal of Guidance, Control, and Dynamics*, 28(5), pp. 1057-1063, 2005.
- [7] Kennedy J. & Eberhart R.C., Particle swarm optimization. *Proc. of IEEE Int. Conf. on Neural Networks*, Perth, Australia, pp. 1942-1948, 1995.
- [8] Yoshida H., Kawata K., Fukuyama Y., et al., Particle swarm optimization for reactive power and voltage control considering security assessment. *IEEE Transactions on Power Systems*, 15(11), pp. 1232-1239, 2000.
- [9] Abido A., Optimal design of power-system stabilizers using particle swarm. *IEEE Transactions on Energy Conversion*, 17(3), pp. 406-413, 2002.
- [10] Gaing Z.L., A particle swarm optimization approach for optimum design of PID controller in AVR system. *IEEE Transactions on Energy Conversion*, 19(2), pp. 384-391, 2004.
- [11] Shi Y. & Eberhart R., A modified particle swarm optimizer. *Proc. of IEEE Int. Conf. Evolutionary Computation*, Anchorage, AK, pp. 69-73, 1998.