PARAMETER ESTIMATION OF WIENER–HAMMERSTEIN MODELS *VIA* GENETIC ALGORITHMS

H.E. Emara-Shabaik*, Y.L. Abdel-Magid, and K.H. Al-Ajmi

King Fahd University of Petroleum & Minerals
Dhahran, Saudi Arabia

الخلاصة :

تعاني الطرق التقليدية لتقدير معاملات النماذج الرياضية من صعوبات مختلفة خاصة في حالات النظم غير الخطية ومع وجود التأثيرات العشوائية. نقدم في هذا البحث خوارزم جيني معدل لتحديد معاملات نماذج وينر– هامرشتين غير الخطية. توضح نتائج المحاكات العددية فاعلية الطريقة مع استخدام دواخل مختلفة مع وجود ضوضاء بنسب مختلفة كذلك طُبق الخوارزم بنجاح لاستنباط أنموذج رياضي يمثل مُولد غير خطي للتيار المستمر.

ABSTRACT

Conventional methods of estimating model parameters have difficulties with both nonlinear systems and with systems operating in noisy environments. In this paper, a modified genetic algorithm is used as a procedure to solve the parameter identification problem of the nonlinear Wiener–Hammerstein models. Numerical simulations are presented to illustrate the effectiveness of the proposed algorithm based on different input signals, and different noise-to-signal ratios of the output. Also, the algorithm is applied to model a DC generator with some nonlinear characteristics.

Keywords: Nonlinear models; Wiener–Hammerstein models; parameter estimation; genetic algorithms; nonlinear DC generator.

*Address for correspondence:
KFUPM Box 1783, Systems Engineering Department
King Fahd University of Petroleum & Minerals
Dhahran 31261, Saudi Arabia
shabaik@ccse.kfupm.edu.sa
amagid@kfupm.edu.sa
alajmi@ccse.kfupm.edu.sa

PARAMETER ESTIMATION OF WIENER–HAMMERSTEIN MODELS *VIA* GENETIC ALGORITHMS

SYMBOLS

t	Time variable
$u(t)$	Input signal
$y(t)$	Output signal
$\hat{y}(t)$	Predicted output signal
$e(t)$	White noise (a sequence of independent random variables)
λ^2	Variance of white noise
q^{-1}	Backward time shift operator
θ	Parameter vector
$\hat{\theta}$	Estimate of parameter vector
θ_0	True value of parameter vector
n_θ	Dimension of parameter vector
N	Number of data points
V	Cost function
$\varepsilon(t,\theta)$	Prediction error corresponding to the parameter vector θ
k	Generation index in genetic algorithms
$P(k)$	Population in the kth generation
p	Population size
f	Fitness function
P_p	Perturbation probability
P_c	Crossover probability
P_m	Mutation probability
s	Reproduction rate.

1. INTRODUCTION

Most physical systems and processes are in reality nonlinear. Although they may admit an approximate linear behavior in a restricted operating range, they can only be adequately characterized by appropriate nonlinear models. Often, this characterization has to be accomplished *via* identification techniques based on available input/output data.

Among the many model structures, which have been widely used for nonlinear systems, is the Wiener–Hammerstein model [1]. The Wiener–Hammerstein model, or the general model, is defined as a dynamic linear system in cascade with a static nonlinear element followed by another dynamic linear system. It can be shown that Wiener–Hammerstein models are special classes of the Volterra series model representation. At the same time, they have a major advantage over the Volterra series models, since they are represented by a finite number of parameters. Moreover, their structure allows the extension of the vast estimation and control techniques available for linear systems.

Wiener–Hammerstein parameter estimation methods, *i.e.* prediction error or output error, are in essence local search techniques that explore for optimum by using gradient-based algorithms. There is the risk, with such methods, of not reaching the global optimum. This is due to the fact that the Wiener–Hammerstein models are nonlinear in their parameters. Therefore, there is a necessity for estimation techniques whose estimates are more likely to converge towards the global optimum. Such methods are characterized as the evolutionary computation (EC) techniques, among which are the Genetic Algorithms (GAs). Genetic Algorithms (GA), Evolutionary Programming (EP), Evolutionary Strategies (ES), and Evolutionary Algorithms (EA) are search techniques for solving problems based on ideas extracted

from the understanding of natural evolution. They have common roots, and ideas from one technique can be borrowed in the other techniques [2].

GAs have appeared in the literature as optimization problem solving techniques. They are guided random search procedures which were first introduced by Holland [3]. The GAs are distinguished from other optimization techniques in that they require no gradient information and are much less likely to become trapped in local minima of multi-modal surfaces. In addition, GAs have shown to be quite insensitive to the presence of noise. Genetic algorithms are independent of the complexity of the performance index considered. It suffices to specify the objective function and to place finite bounds on the estimated parameters in order to successfully run GAs. On the other hand, they have higher computational load than conventional gradient-based methods.

GAs employ different genetic operators to manipulate parameter estimates in a population of candidate solutions over several generations to improve their fitness gradually. Normally, the parameters to be optimized are represented by binary codes. However, the binary representation has the drawback of excessive computations when applied to multidimensional numerical optimization, like the case of Wiener–Hammerstein parameter estimation problem. Therefore, adapting the GAs to work on the real parameter values rather than their binary codes will result in computational simplicity advantages in algorithms. GAs have been applied to systems identification by several researchers. Kristinsson and Dumont [4] have applied GAs to the identification of linear systems. In their work, they have estimated the poles and zeros of both continuous and discrete transfer functions. Sheta and DeJong [5] have considered parameter estimation of a noise-free system containing a mix of bilinear and sinusoidal terms. They, as well, have used GAs in the parameter estimation of a general noisy nonlinear moving average system. Also, using GAs, they have estimated the parameters of a general static nonlinearity of a polynomial type from noisy measurements. In their simulation, they considered uniformly distributed random inputs as well as sum of sinusoids while the noise was considered to be uniformly distributed. Etter *et al.* [6] have demonstrated the applicability of GAs to identify the parameters of a two-pole IIR filter. They have considered the two cases where the assumed model structure set contains the true system, and when it does not. In the first case the error function is a well-behaved unimodal, while in the second case it is a bimodal. In both cases, they have demonstrated the ability of GAs to distinguish the global minimum.

In this work, we present and apply a genetic algorithm (GA) to estimate the parameters of nonlinear Wiener–Hammerstein models. The rest of this paper is organized as follows. The formulation and structure of nonlinear systems represented by Wiener–Hammerstein models are presented in Section 2. The proposed GA is detailed in Section 3. The effectiveness and accuracy of the proposed GA are examined in Section 4 by using numerical simulations. Four input signals and different noise-to-signal ratios (NSR) are incorporated in the simulation in order to test the GA under several realistic conditions. In Section 5, the GA is applied to identify a nonlinear DC generator in terms of a Wiener–Hammerstein model. The conclusions are given in Section 6.

2. NONLINEAR WIENER–HAMMERSTEIN MODELS

The nonlinear model whose parameters to be identified is depicted in Figure 1. It is composed of a zero-memory polynomial nonlinearity embedded between two dynamic linear systems and has a correlated noise corrupting its output $\hat{y}(t)$.

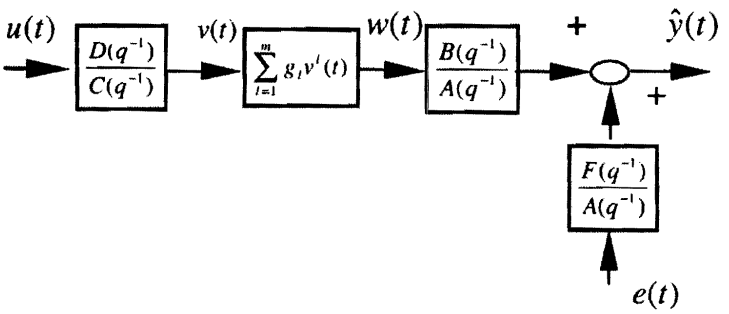

Figure 1. A general Wiener–Hammerstein model with correlated measurements noise.

Mathematically, this model is represented as follows:

$$A(q^{-1})\hat{y}(t) = B(q^{-1})w(t) + F(q^{-1})e(t) \tag{2.1}$$

with

$$w(t) = g_1 v(t) + g_2 v^2(t) + \ldots + g_m v^m(t) \tag{2.2}$$

and

$$C(q^{-1})v(t) = D(q^{-1})u(t) \tag{2.3}$$

where

$$\begin{aligned} A(q^{-1}) &= 1 + a_1 q^{-1} + \ldots + a_{n_a} q^{-n_a} \\ B(q^{-1}) &= 1 + b_1 q^{-1} + \ldots + b_{n_b} q^{-n_b} \\ C(q^{-1}) &= 1 + c_1 q^{-1} + \ldots + c_{n_c} q^{-n_c} \\ D(q^{-1}) &= 1 + d_1 q^{-1} + \ldots + d_{n_d} q^{-n_d} \\ F(q^{-1}) &= 1 + f_1 q^{-1} + \ldots + f_{n_f} q^{-n_f} \end{aligned} \tag{2.4}$$

and q^{-1} is the backward time-shift operator. In this representation, $e(t)$ is an independent identically distributed white noise, with zero mean and variance λ^2. The orders m, n_a, n_b, n_c, n_d, and n_f, or their upper values, are assumed to be known. Let θ be the vector of unknown model parameters defined by,

$$\theta = (a_1 \ldots a_{n_a} b_1 \ldots b_{n_b} c_1 \ldots c_{n_c} d_1 \ldots d_{n_d} f_1 \ldots f_{n_f} g_1 \ldots g_m)^{\mathrm{T}}.$$

The vector θ is to be estimated on the basis of minimizing some error criterion. A prediction error criterion, as described next, will be applied here.

2.1 Criterion for Parameter Estimation Based on Prediction Error (PE)

The prediction error (PE) algorithm is a modification of the least squares (LS) method in order to eliminate the bias problem caused by the correlated noise [7]. This bias is eliminated by incorporating a noise model where the system's, as well as the noise model parameters are estimated simultaneously. For Wiener–Hammerstein models, the variables $v(t)$ and $w(t)$ are not accessible. Therefore, they must be computed using the most recent estimates of $C(q^{-1})$, $D(q^{-1})$, and g_i's. The PE identification scheme is shown in Figure 2.

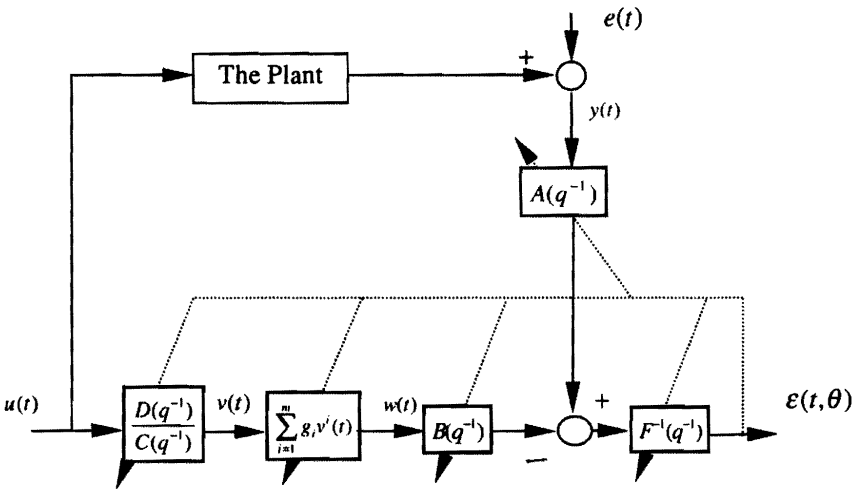

Figure 2. Prediction Error method for the Wiener–Hammerstein model.

The prediction error $\varepsilon(t,\theta)$ is defined as:

$$F(q^{-1})\varepsilon(t,\theta) = A(q^{-1})y(t) - B(q^{-1})w(t) \tag{2.5}$$

and the n_θ parameter vector θ is defined as

$$\theta = (a_1 ... a_{n_a} b_1 ... b_{n_b} c_1 ... c_{n_c} d_1 ... d_{n_d} f_1 ... f_{n_f} g_1 ... g_m)^T .$$

In order to obtain an estimate of the parameter vector θ *via* the PE method, the following criterion function is to be minimized

$$V_N(\theta) = \frac{1}{N}\sum_{t=1}^{N} \varepsilon^2(t,\theta) \tag{2.6}$$

where N is the number of input/output observations. Such a criterion is a nonlinear function of the model parameters. In general, it is multimodal and conventional gradient-based techniques have the drawback of possibly producing a local rather than a global optimal solution. Therefore, in this paper, we propose to apply a GA to solve the optimization problem posed by Equation (2.6) in terms of the system parameter vector θ. The structure and the steps of the GA are discussed in the following section.

3. WIENER-HAMMERSTEIN PARAMETER ESTIMATION USING GAS

In this paper we consider applying GA to estimate the parameters of a Wiener–Hammerstein model following the main procedure of Figure 3. The algorithm is initialized with a randomly generated population. Each member of the population represents a randomly selected candidate parameter estimates vector. Any *a priori* available information, about the system to be modeled, is used to define the parameters' space from which candidate solutions are randomly selected. In the Wiener–Hammerstein system, we are considering the linear subsystems to be stable with minimum phase, and the nonlinearity to be bounded.

Four genetic operators are adopted to explore the parameters' space for the optimal solution. These operators are reproduction, perturbation, crossover, and mutation which will be described later. The performance of the GA in searching for an optimal solution is highly dependent on the algorithm control parameters such as the population size, crossover rate, mutation rate, ... *etc.* The algorithm performance is also problem dependent and appears to be highly dependent on such control parameters in a nonlinear fashion [8]. In the following, the GA control parameters, as it pertains to the Wiener–Hammerstein parameter estimation problem, are discussed.

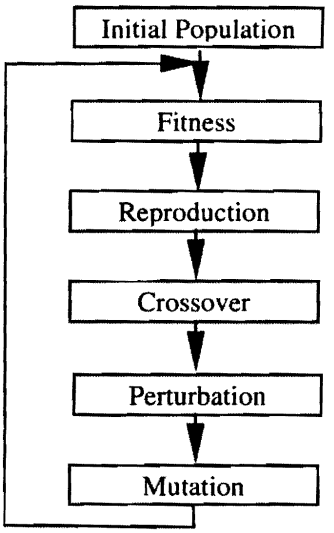

Figure 3. A general type genetic algorithm.

- *Generation.* In GAs, a generation k is a full cycle during which all the algorithm operations are implemented. The GAs will terminate according to either one of the following conditions.
  - Parameters converged to some acceptable estimates as judged by the resulting fitness values.
  - When the relative change in the parameter estimates from generation to generation is no longer significant (*i.e.* less than some threshold).
  - When an upper limit on the number of generations is reached.

  Parameters' space. Each parameter is searched for in a pre-specified region. *A priori* knowledge about the system to be modeled will be used to define such a parameter space. It is to be expected that the larger the parameter space is, the more the number of generations that is required for the GA to converge to the actual value. Requiring the linear subsystems to be stable with minimum phase, and the nonlinearity to be bounded constitutes the *a priori* information for parameter estimation problem considered in this paper.

- *Population and population size.* The population $P(k)$ is defined as the collection of all the candidate solutions, or parameter vectors $\hat{\theta}$'s, at generation k. The initial population $P(0)$ is chosen randomly such that all the parameters fall in their respective spaces. The population size p affects both the ultimate performance and the efficiency of the GA. It is to be expected that increasing the population size will lead to improved performance of the GA in terms of preventing convergence to suboptimal solutions. On the other hand, a large population requires more evaluations per generation, possibly resulting in slowing down the rate of convergence [8]. GAs with a variable population size are an alternative. This idea has been used in [9] to optimize static functions. They used the age of an individual to vary the size of the population at every stage of the algorithm. According to their results, the variable population size resulted in slight improvement of the objective function value at the cost of a significant increase in the computational load. Usually, a constant-size population is adopted [4], [8].

- *Fitness functions.* A suitable fitness function is used to rank the candidate solutions. In Wiener–Hammerstein model identification, the goodness of a parameter vector estimate $\hat{\theta}$ can be judged according to its prediction error cost function $V_N(\hat{\theta})$ as defined by Equation (2.6). The smaller the value of the cost function, the higher the fitness should be. A suitable fitness function for the parameter estimation problem of this paper is defined as follows,

$$f(\hat{\theta}) = \frac{1}{V_N(\hat{\theta}) + 1} \,. \tag{3.1}$$

- *Reproduction.* Reproduction is based on the principle of survival of the fittest. The fittest s percentage of the population at generation k will be copied and these copies will be inserted in the population of generation $k+1$. Using the biased Roulette Wheel Selection (RWS) approach, an estimated parameter vector $\hat{\theta}$ is copied a number of times in accordance with its fitness. That is to say that, each time a copy is needed, the biased wheel is spun and depending on the outcome the individual with the matching fitness will be copied [3]. Keeping the fittest individual will insure the convergence towards the optimum solution.

- *Perturbation.* Perturbation is an addition, with a small probability P_p, of a random value to the parameter vector $\hat{\theta}$. The perturbation is taken from a normal distribution with a zero mean and a variance equals to $1 - f(\hat{\theta})$. Therefore, it is a form of controlled mutation which is sometimes referred to in the literature as Gaussian mutation [10]. This operator is introduced in order to enrich the diversity of the population by searching for a possible better competitor within a neighborhood of a highly fit individual.

- *Crossover.* Reproduction directs the search towards the best existing parameter vectors, or strings, but does not create new strings. In nature, an offspring is not an exact clone of a parent. It usually has two parents and inherits genes from both. The crossover operator takes valuable information from both parents and combines it to generate offsprings. Therefore, this operator adds needed diversity in the search to prevent possible entrapment at a local minimum. To apply this operator, two different parameter vectors from the reproduced population are selected at random and are cut at a random parameter location. Two new $\hat{\theta}$'s are generated by interchanging the cut parts of the parents. This operator is applied with a crossover probability P_c which is usually of high value in order to allow more frequent information exchange between fit $\hat{\theta}$'s. It should be mentioned here that reproduction and crossover give GAs much of their power by directing the search towards the better areas using already existing knowledge [4].

- *Mutation.* Mutation is an operator which increases the variability of the population. According to the value of $f(\hat{\theta})$, mutation is used to alter $\hat{\theta}$ or a single parameter within $\hat{\theta}$. Using the real values of the parameters, mutation is applied here in two different ways, according to the value of $f(\hat{\theta})$. If $f(\hat{\theta})$ is high, then a small mutation is conducted by interchanging two digits in the parameter value. On the other hand, if $f(\hat{\theta})$ is low, mutation is applied by flipping the parameter's sign. The mutation operator is applied with a small mutation probability P_m [4].
- *Filling.* The perturbation, crossover, and mutation operators are conducted on the best parameter vectors and contribute an s percentage of p. The remaining $(100-s)$ percentage of p are new randomly generated parameter vectors inserted in the next generation in order to keep a constant population size. Adding new strings will enrich the diversity of the GA population and hence prevents search stagnation and premature convergence.
- Setting the parameters of a GA is generally done by experimentation to suit the problem being solved, [10]. The flow chart in Figure 4 summarizes the steps of the GA adopted for the Wiener–Hammerstein model parameter estimation.

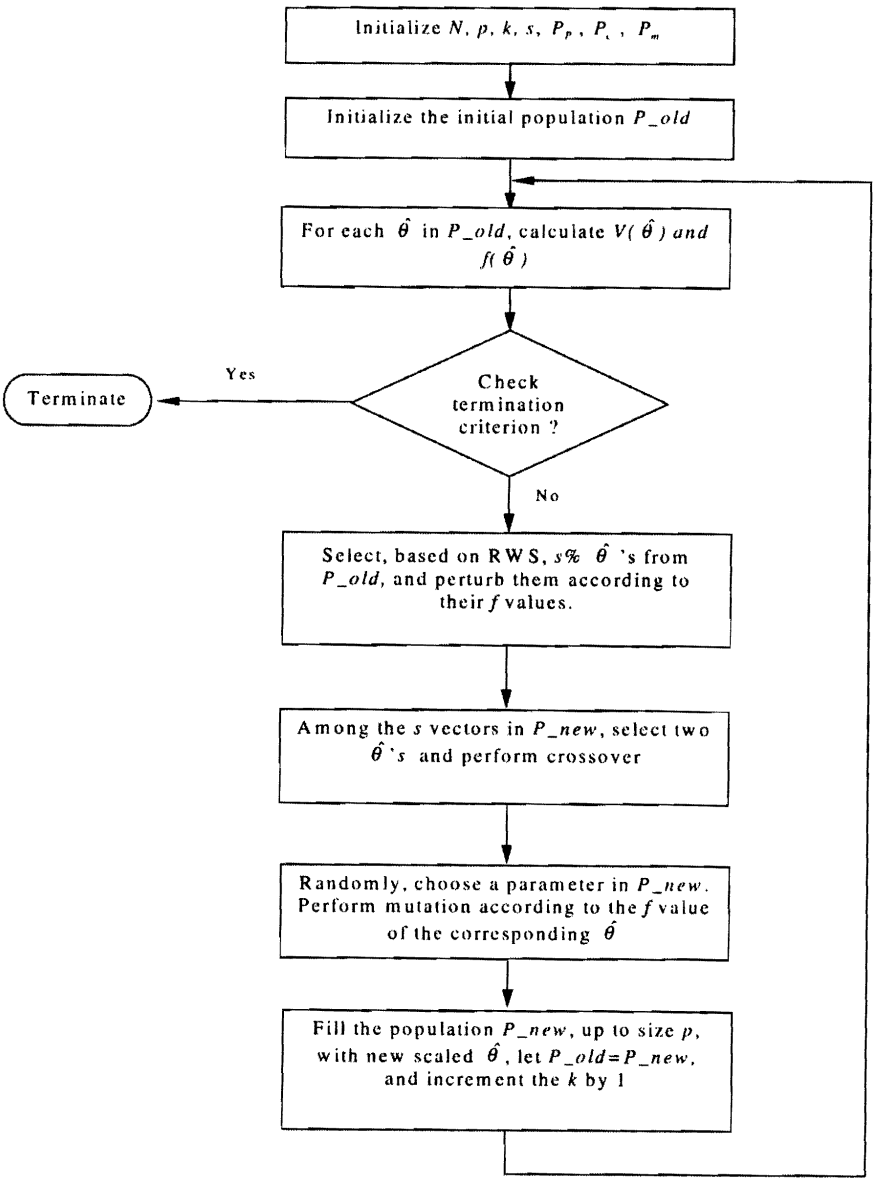

Figure 4. Flow chart of the adopted genetic algorithm.

4. SIMULATION RESULTS AND DISCUSSION

In this section, the performance characteristics of the proposed GA are evaluated *via* numerical simulations. In the following simulations, we use several input signals $u(t)$. Namely,

- A Gaussian distributed input with a zero-mean and variance of 0.5.
- A uniformly distributed input with $-1 \le u(t) \le 1$.
- A PRBS input signal shifting between -1 and 1.
- A four-level Pseudo-Random Signal (4-level PRS) switching randomly between 4 equally spaced levels in the range [–1,1] with equal probability. The switching times are uniformly distributed between 3 to 10 samples.

The output noise $e(t)$ is taken as a zero-mean i.i.d. Gaussian sequence. The variance of $e(t)$ is varied in order to generate different NSRs in the simulation. The simulated Wiener–Hammerstein is given by:

$$A(q^{-1}) = 1 + a_1 q^{-1} + a_2 q^{-2}$$
$$B(q^{-1}) = 1 + b_1 q^{-1}$$
$$C(q^{-1}) = 1 + c_1 q^{-1} + c_2 q^{-2}$$
$$D(q^{-1}) = 1 + d_1 q^{-1}$$
$$F(q^{-1}) = 1 + f_1 q^{-1}$$

The linear subsystems are considered to be stable with minimum phase. The zero-memory nonlinear power polynomial is given by:

$$w(t) = g_1 v(t) + g_2 v^2(t) + g_3 v^3(t).$$

And the actual system's parameters are as follows,

$$\theta_0 = [a_1, a_2, b_1, c_1, c_2, d_1, f_1, g_1, g_2, g_3]^{\mathrm{T}} = [-0.4, 0.25, 0.3, 0.5, -0.25, 0.7, 0.6, 0.5, -1.8, 2.5]^{\mathrm{T}}.$$

The Genetic Algorithms used for identification herein are based on the real values of the parameters rather than their binary codes. The GA control parameters are chosen, by simulation, as:

- a population P of a fixed size, $p = 300$, is initialized randomly;
- only 25% of P is to be selected for reproduction;
- the perturbation probability, P_p, is 0.05;
- the crossover probability, P_c, is 0.9; and
- the mutation probability, P_m, is 0.3.

The parameters' spaces of the linear subsystems are restricted such that their poles and zeros are inside the unit circle in order to fulfill the stability and minimum phase requirements. The parameters of the nonlinearity are restricted to the real line segment of –5 to 5. Thirty numerical experiments, with different realizations of u and e, are conducted. Table 1 gives the mean normalized error (MNE), as defined below, corresponding to output noise levels of 10% and 50% NSR in each case:

$$\mathrm{MNE} = \frac{\left\| \hat{\theta}_{mean} - \theta_0 \right\|^2}{\left\| \theta_0 \right\|^2},$$

where $\hat{\theta}_{mean}$ is the mean value of the estimates over the 30 experiments conducted. While the MNE is a composite measure of bias in the parameters, the average standard deviation (ASD) is a measure of their variance. The ASD is defined as below:

$$\mathrm{ASD} = \frac{\sum_{i=1}^{n_\theta} \sigma_i}{n_\theta},$$

where σ_i is the standard deviation of the ith parameter over the 30 runs.

Table 2 shows the average values, over the 30 runs, of the estimated parameters and their standard deviation as well as the true parameter values.

Considering the Euclidean error norm of the parameter estimates, Figure 5 gives a comparison of the parameter estimates based on the four different inputs with 10% NSR. The error norm is defined as follows: Error norm $= \left\| \hat{\theta}(t) - \theta_0 \right\|^2$.

Figures 6-(a) and (b) give the best of generation parameter estimates.

The results in Tables 1 and 2 indicate that the GA performs almost equally well under both NSR levels. Moreover, it indicates the effectiveness of the GA in estimating the parameters of the Wiener–Hammerstein model, under various excitations, since the estimation errors in all cases are acceptably small.

5. APPLICATION TO THE ESTIMATION OF A NONLINEAR DC GENERATOR

In this section, a nonlinear DC generator system, as described in [11], is considered. The field and armature circuits of such a generator are schematically shown in Figure 7.

The linear circuits are represented by the following transfer functions:

$$G_1(s) = \frac{I_f(s)}{E_f(s)} = \frac{1}{L_f s + R_f} \tag{5.1}$$

Table 1. Perfomance Summary of the Estimates Obtained by GA.

Input Signal	NSR	MNE	ASD
Gaussian		0.0542	0.0138
Uniform		0.0560	0.0155
PRBS	10%	0.0602	0.0376
4-level PRS		0.0897	0.0684
Gaussian		0.0561	0.0467
Uniform		0.0625	0.0347
PRBS	50%	0.0736	0.0705
4-level PRS		0.0975	0.0846

Table 2. Estimation Results Based on Gaussian Input and 10% NSR.

	True Values	Parameter Estimates
a_1	−0.4	−0.3887 ± 0.0033
a_2	0.25	0.2483 ± 0.0027
b_1	0.3	0.3177 ± 0.0068
c_1	0.5	0.3892 ± 0.0210
c_2	−0.25	−0.1889 ± 0.0048
d_1	0.7	0.5779 ± 0.0154
f_1	0.6	0.5727 ± 0.0089
g_1	0.5	0.5180 ± 0.0367
g_2	−1.8	−1.7914 ± 0.0102
g_3	2.5	2.4802 ± 0.0282

$$G_2(s) = \frac{E_a(s)}{E_g(s)} = \frac{Z_l(s)}{L_a s + R_a + Z_l(s)} \quad . \tag{5.2}$$

The coupling between the field current i_f and the armature induced voltage e_g is characterized by a nonlinearity consisting of dead-zone combined with saturation as given below:

$$e_g = \begin{cases} 0 & |i_f| \le \gamma_3 \\ \text{sign}(i_f).\gamma_1.\left\{1 - e^{(-\text{sign}(i_f).i_f + \gamma_3)/\gamma_2}\right\} & |i_f| > \gamma_3 \end{cases} \quad . \tag{5.3}$$

The various constants in the Equations (5.1), (5.2) and (5.3) are as taken by Chen and Fassois [11] to represent the true system. The GA is used to estimate the parameters of a Wiener–Hammerstein model with polynomial type static nonlinearity to represent the DC generator. The estimated parameters are given in Table 3, as shown below.

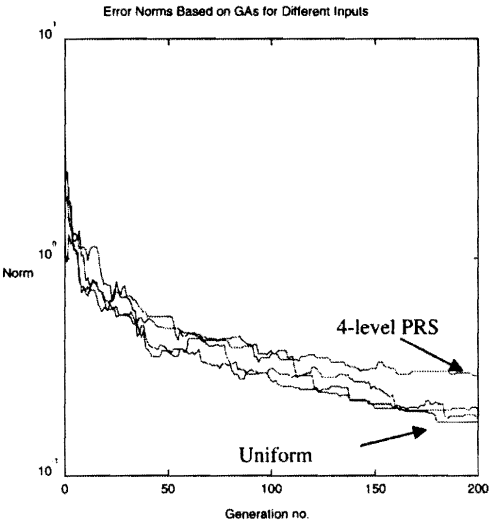

Figure 5. Estimation error norms due to different inputs with 10% NSR.

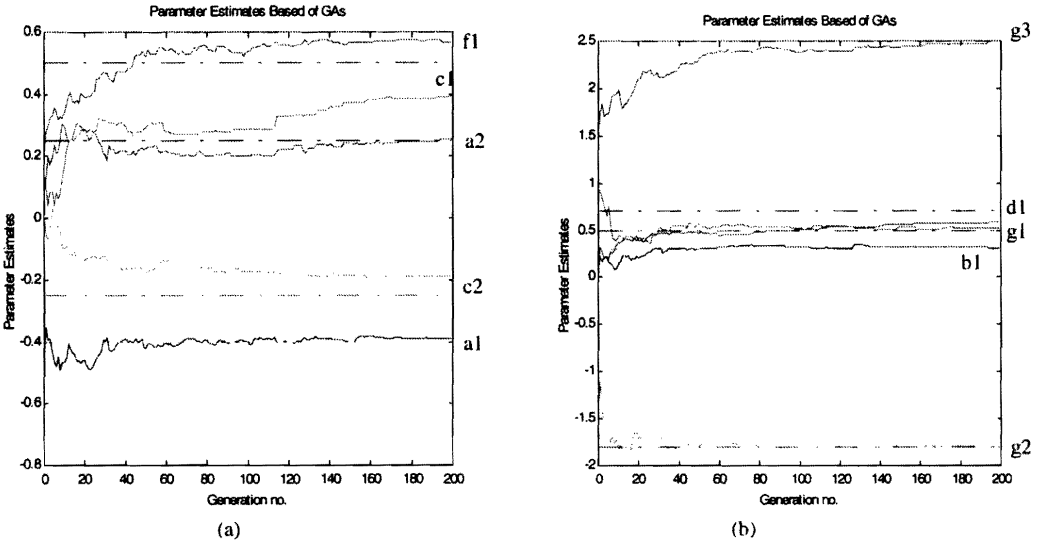

Figure 6. Parameter estimates based on GAs with 10% NSR and Gaussian input.

Using a 4-level pseudo-random input signal as shown in Figure 8, the actual system output and the estimated Wiener–Hammerstein model output are generated, and are shown as in Figure 9. It is evident from that figure that the Wiener–Hammerstein model output matches very well the actual system output.

Table 3. The Nonlinear DC Generator Model Parameters.

No. samples	100
No. generations	200
a_1	-0.8520 ± 0.0118
a_2	-0.0613 ± 0.0244
b_1	0.0467 ± 0.1511
c_1	-0.1629 ± 0.2444
c_2	0.3340 ± 0.0760
d_1	-0.0937 ± 0.1934
f_1	0.5435 ± 0.0326
g_1	0.0199 ± 0.0270
g_2	-0.1698 ± 0.2085
g_3	1.2756 ± 0.1234

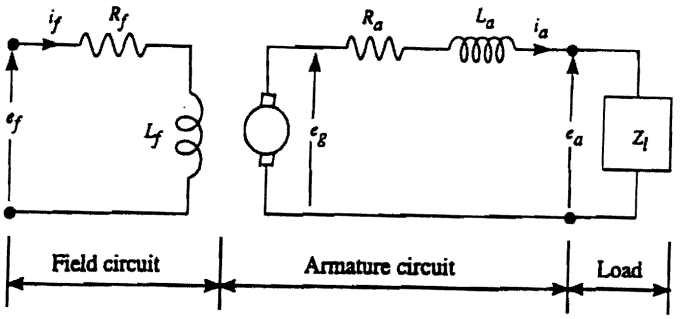

Figure 7. Schematic diagram of the nonlinear DC generator.

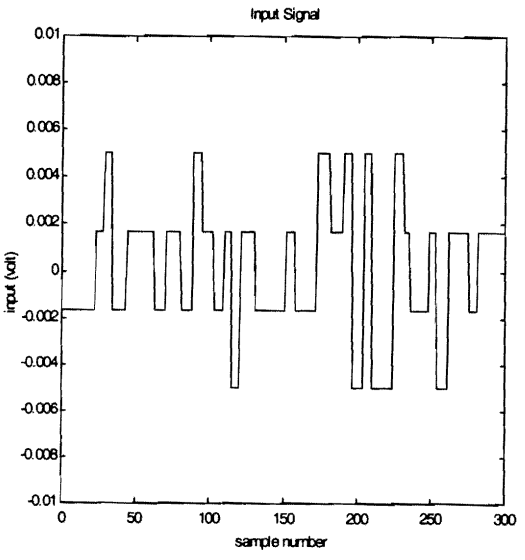

Figure 8. A 4-level pseudo-random input.

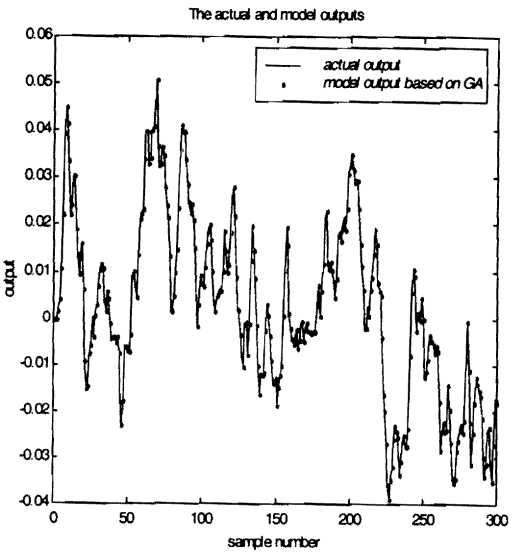

Figure 9. Actual and estimated GA outputs of the DC generator.

Moreover, Figure 10 illustrates the normalized sample autocorrelation function of the prediction errors and a $\pm 1.96/\sqrt{N}$ band. The normalized sample crosscorrelation function between the input and the residuals and a 95% confidence interval are shown in Figure 11. These figures indicate that the Wiener–Hammerstein model is a valid representation for the nonlinear DC generator considered here. Moreover, they indicate the effectiveness of the developed GA in estimating the model parameters.

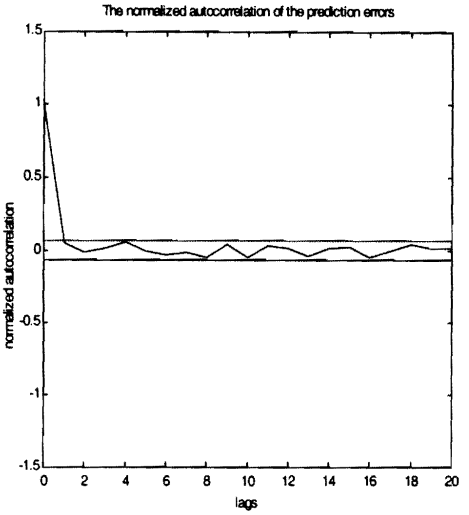

Figure 10. The normalized autocorrelation of the prediction errors.

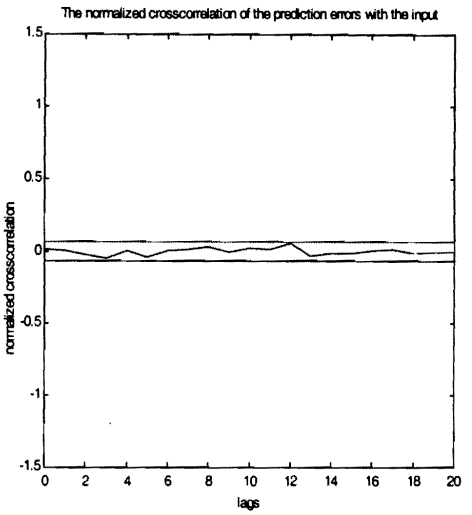

Figure 11. The normalized crosscorrelation between the prediction errors and the input.

6. CONCLUSIONS

Genetic Algorithms are powerful search procedures that have been used successfully to solve optimization problems. In this paper, we have modified the general GA in order to work as a parameter estimator of the Wiener–Hammerstein model. The developed GA has been tested under different input signals as well as different output NSRs. In all cases, it successfully produced parameter estimates that converge towards their actual values. Furthermore, the algorithm is successfully applied to estimate the parameters of Wiener–Hammerstein model to represent a DC generator with a dead zone combined with a saturation type nonlinearity.

ACKNOWLEDGEMENT

The authors wish to acknowledge the support provided by King Fahd University of Petroleum and Minerals.

REFERENCES

[1] R. Haber and H. Unbehaun, "Structure Identification of Nonlinear Dynamic Systems — a Survey on Input/Output Approaches", *Automatica*, **26(4)** (1990), pp. 651–677.

[2] R. Hinterding, "Representation, Mutation and Crossover Issues in Evolutionary Computation", *CEC 2000*, San Diego, CA, pp. 916–923.

[3] J.H. Holland, 1975. *Adaptation in Natural and Artificial Systems*. Ann Arbor: University of Michigan Press.

[4] K. Kristinsson and G. Dumont, "System Identification and Control Using Genetic Algorithms", *IEEE Transactions on Systems, Man, and Cybernetics*, **22(5)** (1992), pp. 1033–1046.

[5] A. Sheta and K. DeJong, 1996. "Parameter Estimation of Nonlinear Systems in Noisy Environments Using Genetic Algorithms", *IEEE Int. Symp. Int. Control, Dearborn, MI*, pp. 360–365.

[6] D. Etter, M. Hicks, and K. Cho, "Recursive Adaptive Filter Design Using an Adaptive Genetic Algorithm", *IEEE International conference on Acoustics, Speech and Signal Processing*, **2** (1982), pp. 635–638.

[7] Hosam E. Emara-Shabaik, Mohammed S. Ahmed, and Khaled H. Al-Ajmi, "Parameter Estimation of Wiener–Hammerstein Models", *JSME International Journal, Series C*, **44**(1) (2001), pp. 118–124.

[8] J. Grefenstette, "Optimization of Control Parameters for Genetic Algorithms", *IEEE Transactions of Systems, Man, and Cybernetics*, **16**(1) (1986), pp. 122–128.

[9] J. Arabas, Z. Michalewicz, and J. Mulawka, "GAVaPS — a Genetic Algorithm with Varying Population Size", *IEEE First Conf. of Evolutionary Computation*, **1** (1994), pp. 73–78.

[10] A.E. Eiben, R. Hinterding, and Z. Michalewicz, "Parameter Control in Evolutionary Algorithms", *IEEE Transactions on Evolutionary Computation*, **3**(2) (1999), pp. 124–141.

[11] C.-H. Chen and S. Fassois, "On the Estimation of Stochastic Wiener–Hammerstein-Type Systems with Non-Smooth Non-Linearity", *American Control Conference*, 1997, pp. 1494–1498.

Paper Received 17 May 2000; Revised 21 February 2001; Accepted 22 April 2001.