Genetic Algorithm in Parameter Estimation of Nonlinear Dynamic Systems

E. Paterakis
manos@egnatia.ee.auth.gr

V. Petridis
petridis@vergina.eng.auth.gr

Ath. Kehagias
kehagias@egnatia.ee.auth.gr

http://skiron.control.ee.auth.gr/kehagias/index.htm

Automation and Robotics Lab
http://skiron.control.ee.auth.gr

Department of Electrical and Computer Engineering
Aristotle University -Thessaloniki – 54006
GREECE

Genetic Algorithm in Parameter Estimation of Nonlinear Dynamic Systems

E. Paterakis V. Petridis A. Kehagias

Department of Electrical and Computer Engineering
Aristotle University– Thessaloniki - Greece

1.Introduction

- We introduce a parameter estimation method for nonlinear dynamic systems.

- A simple genetic algorithm (GA) is used with a recursive probability selection mechanism.

- The method is applied to nonlinear systems with known structure and unknown parameters.

- The nonlinear dynamic system has to be identifiable.

- The selection probabilities have to satisfy an entropy criterion so as to enable the genetic algorithm to avoid poor solutions. This is a new feature that enhances the performance of the GA on the parameter estimation problem.

- Numerical simulations are given concerning the parameter estimation of a planar robotic manipulator with four (4) parameters.

2. Parameter Estimation Problem

The estimation problem consists in determining the member of the set that best describes the data according to a given criterion. The purpose of the estimation has to be outlined. If the final goal is a control strategy for a particular system the accuracy of estimation should be judged on the basis of the time response. However, if the final goal is to analyze the properties of a system the accuracy of estimation should be judged on the basis of deviations in the model parameters. Such problems can be found in engineering problems and also in biology, economy, medicine etc.

The data are produced by the actual system I when it operates under a given experimental condition. The output of the system at time t will be denoted by $y(t)$ (where $y(t) \in \Re^q$), and a sequence of outputs $y(0)$, … , $y(t)$ will be denoted by y_t. The vectors $u(t)$ (where $u(t) \in \Re^m$), u_t denote the input at time t and the sequence of applied inputs, respectively. Similarly $\varphi(t)$ (where $\varphi(t) \in \Re^n$), φ_t denote the system states at time t and the sequence of system states, respectively. The vector ϑ (where $\vartheta \in \Re^d$) denotes the parameter vector. Consider now the following family of discrete time dynamic systems $\mathsf{I}\left(f, g, t, y, \varphi, u, \vartheta_0\right)$:

$$\varphi(t) = f\left(\varphi(t-1), u(t); \vartheta_0\right),\ y(t) = g\left(\varphi(t), u(t); \vartheta_0\right) \qquad (1)$$

In the system of type (1) the functions $f(\cdot)$ and $g(\cdot)$ are known. The input u_t and the r initial conditions are also known. The output y_t is measurable and only the parameter vector ϑ_0 is unknown. All the possible ϑ that can be applied to I construct a model set which will be denoted by M. The model set is indexed by the finite dimensional parameter vector ϑ, so that a particular model in the model set will be denoted by $M(\vartheta)$. The vector ϑ ranges over a set denoted by D_M. (where $D_M \subseteq \Re^d$), which generally will be assumed to be compact. It is also assumed that $\vartheta_0 \in D_M$.. Taking into account noisy measurements, as in real world problems, the I changes to:

$$\varphi(t) = f\left(\varphi(t-1), \ldots, \varphi(t-r), u(t); \vartheta_0\right),\ \tilde{y}(t) = y(t) + w(t) \qquad (2)$$

Now $\hat{y}_t$ is the measurable output and $w(t)$ is white, zero mean output noise. Sampling the noisy system output $\tilde{y}(t)$ for T time steps, a data record C can be constructed, where $\dim C = q \cdot T$ and q is the number of outputs. Note that the dynamic system requires that a sufficient number of output are measured otherwise the system may be unidentifiable. An exhaustive search of M has to be precluded in realistic situations where M has a vast number of members. On the other hand GA is a global search scheme taking into account only a small part of M. The need for a global optimum search scheme arises from the fact that the fitness function may have local minima, either inherent or due to noisy measurements (Ljung, '87), (Söderström and Stoica '89). Also, GA has not the drawbacks of the standard estimation techniques in case of discontinuous objective functions. Hence, the use of this evolutionary scheme is very attractive for parameter estimation of nonlinear dynamical systems. In the GA, an evolvable population P of potential solutions has to be defined taking values over the D_M. Hence as P (where $P \subseteq D_M$) is denoted a set of distinct values of $\vartheta \in D_M$. The set P is finite dimensional, namely $P = \{\vartheta_1, \vartheta_2, \ldots, \vartheta_K\}$, where K is a fixed

number. The set P^j includes the potential solutions at the generation j. The output $y_i^j(t)$ corresponds to the parameter vector $\vartheta_i^j \in P^j$ where $1 \le i \le K$ is the index of the model and j is the generation index. The model validation is achieved through the discrepancy between the noisy measured output $\tilde{y}(t)$ and the model output $y_i^j(t)$ at time step t and it is given by $E_i^j(t)$, where $\|\cdot\|$ denotes Euclidean norm:

$$E_i^j(t) = \left\| y_i^j(t) - \tilde{y}(t) \right\|^2 \tag{3}$$

The sum of $E_i^j(t)$ for every time step t, where $0 \le t \le T$, defines the mean square error (MSE) function $w_i^j(T)$ for the particular model with parameter vector ϑ_i^j at the j generation:

$$w_i^j(T) = \sum_{t=0}^{T} E_i^j(t) \tag{4}$$

The function $w_i^j(T)$ is positive so it can stand as a fitness function. The estimate of ϑ_0 at the j-th generation is ϑ_i^{j*}:

$$\vartheta_i^{j*} = \arg\min_{s=1,2,\ldots K} w_s^j(T) \tag{5}$$

Therefore the parameter estimation problem can be stated as a static optimization problem of finding the value ϑ^* that minimizes the fitness function $w(T)$.

3. Production of Selection Probabilities

Based on the fitness function of (4), a probability is assigned to each member of P^j. This probability expresses the similarity of the model output to system output and it is relative to the performance of each other member of P^j. Several production schemes of selection probabilities have been proposed in literature. One of them is the following:

$$p_i^j(T) = \frac{1/w_i^j(T)}{\sum_{m=1}^{K} 1/w_m^j(T)} \tag{6}$$

The trial solutions with parameter vectors $\vartheta_i^j \in P^j$ where $1 \le i \le K$ have to be evaluated over the whole measurement data record *C* in order to calculate these selection probabilities. Another selection probability scheme is the following:

$$p_i^j(T) = \frac{e^{-w_i^j(T)}}{\sum_{s=1}^{K} e^{-w_s^j(T)}} \tag{7}$$

Substituting $w_i^j(T)$ from (4) we have the Boltzmann selection probability mechanism where σ is the temperature parameter, which is usually constant.

$$p_i^j(T) = \frac{e^{-\sum_{t=1}^{T} \frac{E_i^j(t)}{\sigma}}}{\sum_{s=1}^{K} e^{-\sum_{t=1}^{T} \frac{E_s^j(t)}{\sigma}}} \tag{8}$$

Here, we propose a new method of production of the selection probabilities. The method is based on the selection scheme (8), although it has two differences. First, the probabilities are calculated recursively and second, there is no need to take into account all the measurement record *C* in order to evaluate the trial solutions. The probability update rule for the time step t is defined as:

$$p_i^j(t) = \frac{p_i^j(t-1) \cdot e^{-\frac{E_i^j(t-1)}{\sigma(j)}}}{\sum_{s=1}^{K} p_s^j(t-1) \cdot e^{-\frac{E_s^j(t-1)}{\sigma(j)}}} \tag{9}$$

Where $p_s^j(0) = 1/K \quad \forall \quad j$ and $s = 1 \ldots K$ is the prior probability distribution.

The main point that makes this method different from the others is that the evaluation of each model selection probability is performed for a total of $T^j \le T$ time steps. The number of T^j time steps are defined by the entropy criterion, which is explained in Section 1. The role of the entropy criterion is twofold. First, it prevents premature convergence of the algorithm retaining thus the required population diversity. Second, it makes the algorithm able to use fewer output measurements than it would otherwise be necessary $(T^j \le T)$. Repeating for times $t = 0, t = 1, \ldots, t = T^j$ finally one can obtain:

$$p_i^j(T^j) = \frac{p_i^j(0) \cdot e^{-\sum_{t=0}^{T_i} \frac{E_i^j(t)}{\sigma(j)}}}{\sum_{s=1}^{K} p_s^j(0) \cdot e^{-\sum_{t=0}^{T_i} \frac{E_i^j(t)}{\sigma(j)}}} \tag{10}$$

4. Entropy Criterion

Genetic algorithm requires each generation to contain a sufficient variety of models. If one of the selection probabilities is allowed to tend to one and the others near zero, the next generation of models will only contain the best model of the previous generation. Thus, the genetic algorithm may get stuck with a poor model, which provides only a local minimum of estimation error. On the other hand, it is clear that if the selection probabilities do not concentrate sufficiently on the most promising models, then the search of the parameter space will be essentially random search. To avoid either of the above situations we introduce a novel criterion based on the entropy of the probabilities $p_1^j(t), p_2^j(t), .., p_K^j(t)$. The entropy $H^j(t)$ of $p_1^j(t), p_2^j(t), .., p_K^j(t)$ at time step t is defined by

$$H^j(t) = -\sum_{s=1}^{K} p_s^j(t) \cdot \log\left(p_s^j(t)\right) \qquad (11)$$

The maximum value of $H^j(t)$ is $\log(K)$, and is achieved when $p_1^j(t) = p_2^j(t) = .. = p_K^j(t) = 1/K$. The minimum value of $H^j(t)$ is zero, and is achieved for $p_i^j(t) = 1$, $p_s^j(t) = 0$, $s \neq i$, i.e. when all probability is concentrated on one model. Now consider the following dynamic threshold $\overline{H}^j(t)$ such as

$$\overline{H}(t) = \log(K)\frac{t}{T} \qquad (12)$$

where $1 \le t \le T$ in time steps. This is simply an increasing function with respect to t. It is clear that the inequality

$$H^j(t) < \overline{H}(t) \qquad (13)$$

will be satisfied for some $t \le T$. The above inequality express the *entropy criterion*; when the entropy of $p_1^j(t), p_2^j(t), .., p_K^j(t)$ falls bellow $\overline{H}(t)$, then probability update stops and the next generation of models is produced. Termination will take place at some value of entropy between 0 and $\log(K)$, ensuring that the selection probabilities are neither too concentrated, nor too diffuse. This means that probability update is performed for a variable number of time steps determined by the entropy criterion.

5. Description of the Simple Genetic Algorithm

Having the population P^j and the selection probabilities $p_1^j(t), p_2^j(t), .., p_K^j(t)$ we are ready to make the final step in order to pass to the next generation (j+1). First of all, the trial solutions in P^j have to be encoded by strings of bits. Note that in the initial generation the models $\vartheta_i^0 \in P^0$ have arbitrary parameter values but always in D_M. Each parameter vector $\vartheta_i^j \in P^j$ is translated as a chromosome and each parameter into it as a gene. Specifically, a mapping is taking place from D_M to a discrete set of bit strings. Assume that a gene is represented by n bits the chromosome occupies $n*d$ bits. Thus, there are 2^{n*d} parameter combinations resulting in 2^{n*d} models. What is important here is the change of the model set M to a discrete model set. The discretization has to be done taking into account the trade off between slowness of the algorithm and desirable accuracy of solution.

All the chromosomes are potential parents with a selection probability for mating. A steepest ascent hill-climbing operator is applied to the chromosome with the maximum selection probability and a elitism operator is applied. A roulette wheel is used to select the pair of parents. Genetic operators such as crossover and mutation are applied to selected parents. The whole genetic process is stopped when K individuals have been produced. Then a re-mapping is made from the discrete set of bit strings to D_M and a new set of solutions P^{j+1} is ready to be evaluated.

Schematically the flow of the parameter estimation algorithm is the following.

- Create a population with K members from a model set with arbitrary parameter vectors.
- Assign an initial probability to each model
  - START the parameter estimation loop.
    - START the production of selection probabilities loop.
      - Update recursively the selection probabilities.
      - IF the entropy criterion is not satisfied then CONTINUE in the loop
      - ELSE EXIT from loop.
    - START the production of the next generation
      - The steepest ascent hill-climbing operator is applied to the parameter vector with the maximum selection probability and the elitism operator is applied.
      - Selection of parents and application to them genetic operators such as crossover and mutation
    - REPEAT the production loop until K individuals have been produced.
  - IF the algorithm converges to a solution or the maximum number of generations is exceeded STOP ELSE CONTINUE in the parameter estimation loop.

6. Results

Our algorithm is now applied to the problem of estimating the parameters of the MIT Serial Link Direct Drive Arm. This is configured as a two-link horizontal manipulator by locking the azimuth angle at 180^0 . The input to the manipulator is the angle sequence $\overline{\omega}_t = \left(\overline{\omega}_{1t}, \overline{\omega}_{2t}\right)$. A PD controller controls the manipulator so joint angles ω_{1t}, ω_{2t} track the reference input $\overline{\omega}_t$. Therefore the problem is parameter estimation under closed loop conditions. Ignoring gravity forces, the equations describing the manipulator and controller are the following

$$\begin{bmatrix} K_{p1} & 0 \\ 0 & K_{p2} \end{bmatrix} \cdot \begin{bmatrix} \omega_1(t) - \overline{\omega}_1(t) \\ \omega_2(t) - \overline{\omega}_2(t) \end{bmatrix} + \begin{bmatrix} K_{u1} & 0 \\ 0 & K_{u2} \end{bmatrix} \cdot \begin{bmatrix} \dot{\omega}_1(t) - \dot{\overline{\omega}}_1(t) \\ \dot{\omega}_2(t) - \dot{\overline{\omega}}_2(t) \end{bmatrix} =$$

$$\begin{bmatrix} I_1 + I_2 + m_2 l_1 l_2 \cos(\omega_2) + \frac{1}{4}\left(m_1 l_1^2 + m_2 l_2^2\right) + m_2 l_1^2 & I_2 + \frac{1}{2} m_2 l_1 l_2 \cos(\omega_2) + \frac{1}{4} m_2 l_2^2 \\ I_2 + \frac{1}{2} m_2 l_1 l_2 \cos(\omega_2) + \frac{1}{4} m_2 l_2^2 & I_2 + \frac{1}{4} m_2 l_2^2 \end{bmatrix} \cdot \begin{bmatrix} \ddot{\omega}_1(t) \\ \ddot{\omega}_2(t) \end{bmatrix} +$$

$$\begin{bmatrix} F & -m_2 l_1 l_2 \sin(\omega_2) \\ m_2 l_1 l_2 \sin(\omega_2) & F \end{bmatrix} \cdot \begin{bmatrix} \dot{\omega}_1(t) \\ \dot{\omega}_2(t) \end{bmatrix} + \begin{bmatrix} -2 m_2 l_1 l_2 \sin(\omega_2) \\ 0 \end{bmatrix} \cdot \dot{\omega}_1(t)\dot{\omega}_2(t)$$

The inertial parameters are $I_1 = 8.095 \mathrm{Nt \cdot m \cdot s^2/rad}$, $I_2 = 0.253 \mathrm{Nt \cdot m \cdot s^2/rad}$. The coefficient of friction is $F = 0.0005 \mathrm{Nt \cdot m \cdot s/rad}$. The gain parameters of the PD controller are $K_{p1} = 2500$, $K_{u1} = 300$ for the first joint, and $K_{p2} = 400$, $K_{u2} = 30$ for the second joint. The mass and the length parameters, namely $m_1 = 120.1 \mathrm{Kgr}$, $m_2 = 2.1 \mathrm{Kgr}$, $l_1 = 0.462 \mathrm{m}$, $l_2 = 0.445 \mathrm{m}$, are considered unknown for the purposes of this example. The system equations are discretized in time (with a discretization step $dt = 0.01 \mathrm{sec}$) to obtain equations of the form $\phi(t) = f\left(\phi(t-1), u(t); \vartheta_0\right)$, $y(t) = g\left(\varphi(t); \vartheta_0\right)$:

$$\varphi(t) = \begin{bmatrix} \omega_1(t) \\ \omega_2(t) \\ \omega_1(t-1) \\ \omega_2(t-1) \end{bmatrix}, y(t) = \begin{bmatrix} \omega_1(t-1) \\ \omega_2(t-1) \\ l_1 \cos\left(\omega_1(t-1)\right) + l_2 \cos\left(\omega_1(t-1) - \omega_2(t-1)\right) \\ l_1 \sin\left(\omega_1(t-1)\right) + l_2 \sin\left(\omega_1(t-1) - \omega_2(t-1)\right) \end{bmatrix}, u(t) = \begin{bmatrix} \overline{\omega}_1(t) \\ \overline{\omega}_2(t) \end{bmatrix}, \vartheta = \begin{bmatrix} m_1 \\ m_2 \\ l_1 \\ l_2 \end{bmatrix}$$

where $\varphi(t), y(t), u(t)$ and ϑ are the state, output, input and parameter vector, respectively; the last two terms in $y(t)$ are the coordinates of the manipulator end point. This completes the description of the system.

The parameter estimation algorithm is applied using genetic algorithm parameters as follows: number of bits per parameter n=13, crossover probability q=0.8, population size K=50, number of crossover and mutation points p=4, number of generations $I_{max} = 5000$. The vector ϑ ranges over the set D_M. The lower bound of D_M at the each direction chosen to be 30% of the true value of the corresponding parameter, and the upper bound chosen to be 200% of the true value of the corresponding parameter.

Several experiments are run with the above setup, varying the level of the noise present in the measurements (noise free, $\pm 0.25^0$, $\pm 0.50^0$, $\pm 1^0$, $\pm 2^0$, $\pm 3^0$ and $\pm 5^0$); noise is white, zero-mean and uniformly distributed. For every choice of the model type and noise level one

hundred experiments are run; the accuracy of the parameter estimates for every experiment is expressed by the following quantity:

$$S = \frac{\frac{|\delta m_1|}{m_1} + \frac{|\delta m_2|}{m_2} + \frac{|\delta l_1|}{l_1} + \frac{|\delta l_2|}{l_2}}{4}$$

where δm_1 is the error in the estimate of m_1 and similarly for the remaining parameters. In other words, S is the average relative error in the parameter estimates.

The results using recursive selection probabilities are presented in Fig1. Each curve is the cumulative histogram of S for one hundred experiments at a given noise level. We also applied a genetic algorithm using the mean square error (MSE) to determine the selection probabilities (see equ.6). The cumulative results of MSE selection probabilities are presented in Fig2. As can be seen, these results are significantly worse than the results of recursive selection probability mechanism.

Finally, it must be mentioned that the average duration of one run of the recursive parameter estimation method is 7 minutes on HP Apollo 735 workstation. Runs involving models with recursive selection probability require around 500 generations, each generation requiring around 150 input/output measurements, while runs involving MSE selection probability require around 1200 generations, each generation requiring exactly 300 input/output measurements.

7. Conclusions

We have applied a GA on parameter estimation problem and tested its performance on the MIT Serial Link Direct Drive Arm. The test problem is hard because of the strong nonlinearities of the system, the insensitivity to mass values, the presence of noise in the data and the existence of the PD controller. The performance of the algorithm is quite good, even for high noise levels. At low noise levels, the number of models that are close to the true system with accuracy better than 2% is quite high.

The success of our method is attributed to the following features.

- **Recursive Probability Selection Mechanism**. The use of an exponential error function, and the resulting use of a competitive and multiplicative scheme, accentuate the competition between rival models and facilitates the selection of the best model in a search subset.

- **Entropy Criterion**. It ensures an appropriate balance between extreme and insufficient diversity, such that the algorithm is not trapped at local minima. The recursive nature of the probability selection mechanism and the use of the entropy criterion reduce the computation required for the algorithm to obtain parameter estimates within the specified accuracy.

-

An additional attractive feature of our algorithm is its generality; no particular assumptions have been made about the form of the system equations (1), nor about the probability distribution function of the noise on the measurements. In particular, $f(\cdot)$ and $g(\cdot)$ need not be continuous.