

Estimation of Parameters for Model Matching using Genetic Algorithms

Sheeba PS

Lokmanya Tilak College of Engineering
Koperkhairane
Navi Mumbai

ABSTRACT

This paper develops an optimization procedure to estimate the parameter values of a given dynamic system. The actual nonlinear model of the system is assumed available with the approximate range of parameter values. The model used in this work is the PS4 actuator with nozzle dynamics which is used for the liquid upper stage control system of PSLV. The objective is to estimate the parameter values so that the error between actual system output and the simulated output is minimized. This is achieved through Genetic Algorithms(GA) which is a global optimization technique. GAs are stochastic algorithms based on Darwin's theory of survival of the fittest. They are inspired by biological phenomena of natural genetics and natural selection. The basic elements of natural genetics- *reproduction, crossover and mutation*- are used in the genetic search procedures. GA is proved robust and efficient in finding optimal solutions in complex problem spaces.

Keywords:

Genetic Algorithm, Optimization, Parameter Estimation

1. INTRODUCTION

In order to deal in a systematic and efficient way with problems involving time dependent behavior we must have a description of the objects or process involved called a model. In many cases the processes are so complex that it is not possible to obtain reasonable models using physical insight. In such cases one is forced to use identification techniques. An identification experiment is performed by exciting the system and observing its input and output over a time interval. These signals are normally recorded in a computer mass storage for subsequent information processing. We then try to fit a parametric model of the process to the recorded input and output sequences. Optimization methods play an important role in modeling, because a model is not usually developed as an end in itself. Rather, the model is formulated in order to determine the values of free parameters that produce an optimum measure of goodness. Parameter estimation is defined as the experimental determination of values of parameters that govern the dynamic and/or nonlinear behavior, assuming that the structure of the process model is known. Optimization is the art of obtaining the best result under given circumstances. There is no single method available for solving all optimization problems efficiently. The optimum seeking methods are also known as mathematical programming tech-

niques. Genetic Algorithms, Simulated Annealing and Neural Network methods represent a new class of mathematical programming techniques.

The classical methods of optimization are useful in finding the optimal solution of continuous and differentiable functions. These methods are analytical and make use of the techniques of differential calculus in locating the optimum points. Since some of the practical problems involve objective functions that are not continuous and/or differentiable, the classical optimization techniques have limited scope in practical application. Since standard estimation algorithms employ gradient based searches and are subjected to entrapment in local minima, there is a need for a global optimization technique such as evolutionary programming.[2]

Optimization techniques, having reached a degree of maturity over the past several years, are being used in wide spectrum of industries, including aerospace, automotive, chemical, electrical and manufacturing industries. With rapidly advancing computer technology, computers are becoming more powerful, and correspondingly, the size and the complexity of the problems being solved using optimization techniques are also increasing. Optimization methods, coupled with modern tools of computer-aided design, are also being used to enhance the creative power of conceptual and detailed design of engineering systems.

Traditional optimization and search methods are classified as: calculus based, enumerative and random search methods. Calculus based methods subdivide into direct and indirect methods. Indirect methods seek local extrema by solving the usually nonlinear set of equations resulting from setting the gradient of the objective function equal to zero. On the other hand, direct methods seek optima by hopping on the function and move in a direction related to the local gradient. This is simply the notion of hill climbing to find the local best, climb the function in the steepest permissible direction. Both the methods are local in scope: the optimum they seek are best in a neighborhood of the current point and in many cases miss the higher peak. Also calculus based methods depend upon the existence of derivatives. Many practical spaces have little respect for the notion of a derivative and the smoothness this implies. The real world of search is fraught with discontinuities and vast multimodal noisy search spaces. For this reason and because of their inherently scope of search, we must reject calculus based methods. In enumerative schemes, within a finite search space or discretized infinite search space, the search algorithms starts looking at objective function values at every point in the space one at a time. Many practical spaces are simply too large to searching one at

a time and still have a chance of using the information to some practical end. Random search algorithms have achieved increasing popularity as researchers have recognized the shortcomings of calculus based and enumerative schemes. Yet, random walks and random schemes that search and save the best must also be discounted because of the efficiency requirement. In our haste to discount strictly random search methods, one must be careful to separate them from randomized techniques [1]. The genetic algorithm is an example of a search procedure that uses random choice as a tool to guide a highly exploitative search through a coding of a parameter space. In recent years, some optimization methods that are conceptually different from the traditional mathematical programming techniques have been developed [3]-[7]. These methods are based on certain biological, molecular and neurological phenomena. Genetic algorithms are based on the principle of natural genetics and natural selection. Simulated annealing is based on the simulation of thermal annealing of critically heated solids. In neural network based methods, the problem is modeled as a network consisting of several neurons and the network is trained suitably to solve the optimization problem efficiently [2], [3]. A system identification problem using genetic algorithms is addressed in [9]. A similar problem of identification of parameters of an induction motor using Genetic Algorithms is developed in [10]. In a recent paper [11], genetic algorithm is used to estimate the parameters of a warranty cost model for a Malaysian automotive industry. A numerical technique based genetic algorithm to determine the electrical parameters of photovoltaic solar cells is developed in [12]. Many practical optimum design problems are characterized by mixed continuous discrete variables and discontinuous and non convex design spaces. If standard nonlinear programming techniques are used for this type of problem they will be inefficient, computationally expensive, and in most cases find a relative optimum that is closest to the starting point. Genetic Algorithms are well suited for solving such problems and in most cases they can find the global optimum solution with a high probability.

2. BIOLOGICAL BACKGROUND OF GA

The fundamental unit of information in living systems is the gene. In general, a gene is defined as a portion of a chromosome that determines or affects a single character or phenotype. It comprises a segment of deoxyribonucleic acid (DNA), commonly packaged into structures called chromosomes. This genetic information is capable of producing a functional biological product which is most often a protein.

The genes which determine the characters of an individual are carried in the chromosomes and an individual usually has many genes for the determination of various different characters. The genes on the same chromosome do not assort out independently during their inheritance. The tendency of two genes to remain together in the same chromosome during hereditary transmission is called linkage. Genes located in the same chromosome show linkage. These linked genes may either remain together during the process of inheritance and, thus showing complete linkage or they may be separated, thus, displaying the incomplete linkage. The incomplete linkage takes place due to the occurrence of new combinations or recombinations of linked genes. The recombination, infact, occurs through a process known as crossing over. The crossing over is a process that produces new combinations or recombinations of genes by interchanging corresponding segments between chromosomes.

A little amount of DNA synthesis takes place during the crossing over process and that little amount of DNA is thought to repair the broken chromosomes. Crossing over provides a direct evidence

of the linear arrangement of genes in the chromosomes. Since, new combinations of genes are obtained by crossing over which produce variations in the organisms, therefore it is a necessary feature of evolution.

DNA is a relatively stable polymer and nucleotides generally display a very low tolerance for alterations in genetic information. An inheritable change in the phenotype or any change in a DNA sequence is called a mutation. In general, this operation is rare and random. The process of mutation is blind to its consequences; it throws up every possible combination of mutants, and natural selection then favours those which are better adapted to their environment. Favourable mutations that confer some advantages to the cell in which they occur are rare, being sufficient to provide the variations necessary for natural selection and thus evolution.

3. GENETIC ALGORITHMS

Genetic Algorithms are search algorithms based on the mechanics of natural selection and natural genetics. They combine survival of the fittest among string structures with a structured yet randomized information exchange to form a search algorithm with some of the innovative flair of human search. The Darwinian theory of evolution states that good characteristics, which help a member of the population survive, will gradually dominate the population as the members with bad characteristics die off. In every generation, a new set of artificial creatures (strings) is created using bits and pieces of the fittest of the old and occasional new part is tried for good measure. While randomized, Genetic Algorithms are no simple random walk. They efficiently exploit historical information to speculate on new search points with expected improved performance. A GA maintains a population of candidate solutions where each solution is usually coded as a binary string called a chromosome. A chromosome- also referred to as a genotype- encodes a parameter set for a set of variables being optimized. Each encoded parameter in a chromosome is called a gene. A decoded parameter set is called a phenotype. A set of chromosomes forms a population which is evaluated and ranked by a fitness evaluation function. The initial population is usually generated at random.[1]

3.1 How Genetic Algorithms are different from traditional methods?

GAs differ from traditional methods of search and optimization in four ways:

- (1) GAs work with a coding of the parameter set, not the parameters themselves
- (2) GAs search from a population of points, not a single point
- (3) GAs use payoff (objective function) information, not derivatives or other auxiliary knowledge
- (4) GAs use probabilistic transition rules, not deterministic rules.

These fundamental differences allow GAs to be more robust than conventional deterministic search and optimization routines.

3.2 GA Procedure

The first step in using GAs is to code the problem space. This is done by mapping the design variables into chromosomes through the use of an arbitrary alphabet. The alphabet of choice for GAs is binary, or 1's and 0's. Use of binary allows the GA to associate high fitness with similarities in the string. Other coding, while viable, require a more expensive decoding process, and may not allow

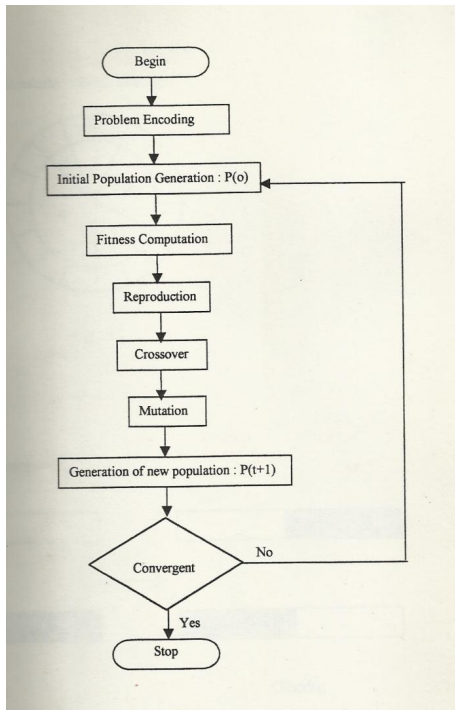


Fig. 1. Flowchart for GA

the GA to recognize similarities in "good" members of the population. An initial population of genetic strings with randomly chosen genes is created first. The size of the population remains constant throughout the genetic optimization. Various genetic operators are applied at given probabilities to generate new members. In order to form successive generations, parents are chosen from the current population based on their fitness. One generation after another is created until some stopping criterion is met. The implementation flow chart of Genetic Algorithm is as shown in Fig.1.

3.3 Genetic Operators

A GA in its simplest form uses three operators: selection, crossover and mutation.[1]

3.3.1 Selection. Selection or reproduction, is the process of creating a new generation. This is accomplished by copying strings or individuals from the last generation into the population of the new generation, based upon the evaluation of the individuals assigned fitness value.

The reproduction operator may be implemented in algorithmic form in a number of ways. Perhaps the easiest is to create a biased Roulette wheel where each current string in the population has a Roulette wheel slot sized in proportion to its fitness. For proportionate selection scheme, the selection procedure is as follows:

- Sum the fitness of all the population members; named as total fitness F_{sum}
- Generate a random number n between 0 and total fitness F_{sum}
- Return the first population member whose fitness added to the fitness of the preceding population members is greater than or equal to n .

For example, the circumference of the Roulette wheel is F_{sum} for all five chromosomes. Chromosome 4 is the fittest chromosome and

occupies the largest interval. Whereas chromosome 1 is the least fit which corresponds to a smaller interval within the Roulette wheel. To select a chromosome, a random number is generated in the interval $[0, F_{sum}]$ and the individual whose segment spans the random number is selected.

Another method of selection for this methodology is called Rank Based Selection. This procedure sorts the population based on fitness value i.e., all members must be ranked from best to worst according to the value of objective function. A roulette wheel is implemented where the i^{th} ranked member in the population is given an interval $[\phi_{i-1}, \phi_i]$ whose size depends on the population size, N , and its rank i , in the population:

$$\phi_i = \phi_{i-1} + \frac{2(N - i + 1)}{N(N + 1)}$$

where $\phi_0 = 0$ and $i = 1, 2, \dots, N$.

A uniform random number is then generated between 0 and 1; and the member i is selected as a parent if the number lies in the interval $[\phi_{i-1}, \phi_i]$. [8]

3.3.2 Crossover. Crossover is operator responsible for introducing most new solutions into the population. This is done by selecting, at random, two parent strings. These strings are then crossed to produce two offsprings. The operation is applied by first drawing a random number to determine the crossover point. The gene string is then split at same point in both parents. The genetic information contained in the strings is then swapped between the randomly determined crossing points. This process of sharing information between different strings gives GAs much of their power.

3.3.3 Mutation. Mutation is the process of randomly changing one bit of information in the string. The change is accomplished by changing a single gene from 0 to 1 or 1 to 0. Mutation prevents the GA from losing potentially important information or stagnating during the solution process. Mutation has the effect of providing a push from a particular solution which may be a local optimum rather than the global one sought. mutation is considered to be a secondary operator to crossover and reproduction that provides an insurance policy against premature convergence and loss of information.

3.3.4 Schema Theory. Consider a simple three dimensional space, and assume that the searching space of the solution of a problem can be encoded with three bits, this can be represented as a simple cube with the string 000 at the origin. The corners in this cube are numbered by bit strings and all adjacent corners are labeled by bit strings that differ by exactly 1 bit. If '*' represents a don't care or 'wild card' match symbol, then the front plane of the cube can be represented by the special string 0**.

Strings that contain "*" are referred to as schematic and each schema corresponds to a hyperplane in the search space. A schema represents all strings (a hyperplane or subset of the search space), which match it on all positions other than '*'. It is clear that every schema matches exactly 2^r strings, where 'r' is the number of don't care symbols, '*', in the schema template. Every binary encoding is a 'chromosome' which corresponds to a corner in the hypercube and is a member of the $2^L - 1$ different hyperplane, where L is the length of the binary encoding. Moreover, 3^L hyperplanes can be defined over the entire search space. Because GA operate on population of strings, one can track the proportional representation of a single schema representing a particular hyperplane in a population. One can also indicate whether that hyperplane will increase or decrease its representation in the population overtime, when fitness

based selection is combined with crossover to produce offspring from existing strings in the population.

3.3.5 Effect of Selection. Since a schema represents a set of strings, a fitness value $f(s,t)$ with schema "s", and the average fitness of the schema can be associated. $f(s,t)$ is then determined by all the matched strings in the population. If propotional selection is used in the reproduction phase the number of matched strings of a schema "s" in the next generation can be estimated.

Let $\zeta(s, t)$ be the number of strings matched by schema "s" at the current generation. The probability of its selection (in a single string selection) is equal to $\frac{f(s,t)}{F(t)}$, where $F(t)$ is the average fitness of the current population. The expected number of occurrences of "s" in the next generation is

$$\zeta(s, t + 1) = \zeta(s, t) \frac{f(s, t)}{F(t)} \quad (1)$$

Let

$$\epsilon = \frac{f(s, t) - F(t)}{F(t)} \quad (2)$$

If $\epsilon > 0$, it means that the schema has an above average fitness and viceversa.

$$\zeta(s, t) = \zeta(s, 0)(1 + \epsilon)^t \quad (3)$$

3.3.6 Effect of Crossover. During the evolution of GA, the genetic operations are disruptive to current schemata, therefore, their effects should be considered. Assuming that the length of chromosomes is L and one-point crossover is applied, in general, a crossover point is selected uniformly among $L-1$ possible positions. This implies that the probability of destruction of a schema "s" is

$$P_d(s) = \frac{\delta(s)}{L-1} \quad (4)$$

or the probability of a schema "s" survival is

$$P_s(s) = 1 - \frac{\delta(s)}{L-1} \quad (5)$$

where δ is the defining length of the schema "s" defined as the distance between the outermost fixed positions.

Assuming the operation rate of crossover is P_c , the probability of a schema "s" survival is:

$$P_s(s) = 1 - P_c \frac{\delta(s)}{L-1} \quad (6)$$

A schema "s" may still survive even if a crossover site is selected between fixed positions, the above equation can be modified as

$$P_s(s) \geq 1 - P_c \frac{\delta(s)}{L-1} \quad (7)$$

3.3.7 Effect of Mutation. If the bit mutation probability is P_m , then the probability of a single bit survival is $1 - P_m$. Denoting the order of schema "s" by $o(s)$, the probability of a schema "s" surviving a mutation is,

$$P_s(s) = (1 - P_m)O(s) \quad (8)$$

Since $P_m \ll 1$, this probability can be approximated by

$$P_s(s) \approx 1 - o(s)P_m \quad (9)$$

Combining the effect of selection, crossover and mutation, a new form of the reproductive schema growth equation is derived

$$\delta(s, t + 1) \geq \delta(s, t) \frac{f(s, t)}{F(t)} [1 - P_c \frac{\delta(s)}{L-1} - o(s)P_m] \quad (10)$$

From this equation, it can be concluded that a high average fitness value alone is not sufficient for a high growth rate. Indeed, short, low order, above average schemata receive exponentially increasing trials in subsequent generations of a GA. This conclusion is so important that, it is given a special name: Schema Theorem or Fundamental Theorem of Genetic Algorithms. [1]

4. CASE STUDY

In this case study, GA is applied to estimate the parameter values of the PS4 actuator with nozzle dynamics which is used for the liquid upper stage control system of PSLV.

4.1 Test Signal

The test signal used here is pseudo random binary sequences (PRBS) Fig 2. PRBS are probably the most convenient inputs for the purposes of identification by correlation integral techniques. These sequences are of periodic nature, their periods being relatively short, and yet their auto correlation integral provides a satisfactory approximation of a delta function. PRBS are two stage signals which may be generated by using a shift register of order n . The register state variables are fed with 0 or 1. Every initial state vector is allowed except the all zero state. When the clock pulse is applied, the value of the k^{th} state is transferred to the $(k + 1)^{th}$ state and a new value is introduced into the first state through the feedback path. This feedback is applied through the modulo-two gate (or gates) from the outputs of two (or more) stages of the shift register so that a single (or multiple) closed loop is formed. For a shift register with n states, the maximum period of sequence generated is 2^n . Infact, 2^n is an upper bound which cannot be attained as the occurrence of all zero state must be prevented. The maximum possible period is $N = 2^n - 1$. A PRBS input with period 63 is used as the test signal in this experiment. The auto correlation function of PRBS input signal, denoted by $x(t)$ is given by,

$$\phi_{xx}(\tau) = \frac{1}{T} \int_0^T x(t)x(t + \tau)dt$$

The cross correlation function is given by

$$\phi_{xy}(\tau) = \frac{1}{T} \int_0^T x(t)y(t + \tau)dt$$

From the above equations, we can find the impulse response $g(t)$ which is given by

$$g(t) = \frac{1}{T} \phi_{xx}^{-1} \phi_{xy}$$

4.2 Genetic Approach

The model used in the experiment is the PS4 actuator with nozzle dynamics. The aim is to estimate the value of motor inertia (J_m), viscous damping coefficient reflected to the motor side (B_m), nozzle inertia (J_{noz}) and natural frequency of nozzle (ω_{noz}) so that the error between the actual system output and the estimated output is minimized.

The actual values of the unknown parameters which was experimentally determined in VSSC were available. In this work, the range of parameter values were selected as one octave above and one octave below of the actual value. The approximate range of parameter values are:

$$J_m \in [5.96e - 06, 2.384e - 05]kgms^2$$

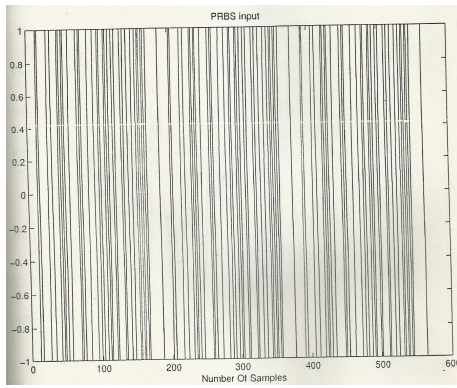


Fig. 2. PRBS input

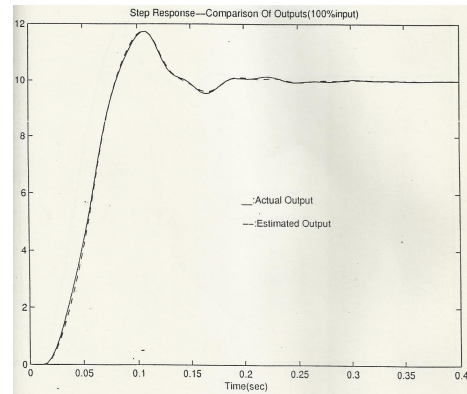


Fig. 3. Comparison of outputs

$$B_m \in [1.06e - 05, 4.32e - 05]kgms$$

$$J_{noz} \in [0.6375, 2.55]kgms^2$$

$$\omega_{noz} \in [47.1239, 188.4956]rad/s$$

From the given range of parameter values, randomly generated an initial population. The parameter values were appropriately scaled and coded. Each parameter value was represented as a 6 bit binary string. The strings were placed end to end so as to form chromosomes. The selection of parent chromosome is based on the notion of "fitness" which governs the extent to which an individual can influence future generation. Rank based parent selection was utilized. For this, a Roulette wheel was implemented. Random numbers between the interval 0 and 1 were generated by means of RAND command in Matlab. A particular member was selected if the random number generated falls between the specified interval according to the ranking formula. Higher ranked individuals were given a higher interval and the lower ranked individual were given a smaller interval so that there exists a probability of higher ranked member to be selected more. After parent selection, crossover operation was applied. One-point crossover was utilized. This was done by randomly choosing a crossover point between two parent strings using the Matlab command UNIFRND. After crossover operation, mutation was applied with a probability of 0.016. This was done by randomly selecting a member from the population and mutating a randomly chosen bit position. After crossover and mutation, the parameter were converted from binary to decimal and rescaled. The fitness function for the new population was computed. For parameter estimation problem, the fitness of a particular chromosome is roughly proportional to the inverse of the error. Actual system output was stored in the computer memory. For each parameter combination, the simulated output was compared with the actual system output. The sum of squares of error values between the actual system output and the simulated output for each parameter combinations were computed. After crossover and mutation, the parameter values were converted from binary to decimal and rescaled. The fitness function for the new population was computed. The new population was combined with the old population and ranked according to the fitness value. Parents for second generation were obtained by using the rank based selection scheme. The process was repeated for a specific number of generations or until no improvement in the fitness function value is reached.

| J_m | B_m | J_{noz} | ω_{noz} |
|------------|-------------|-----------|----------------|
| 1.192 e-05 | 2.237 e-05 | 1.7304 | 97.6138 |
| 5.96 e-05 | 1.08 e-05 | 1.275 | 87.5158 |
| 7.24 e-05 | 1.08 e-05 | 1.0473 | 67.3199 |
| 1.192 e-05 | 2.006 e-05 | 1.3205 | 94.2478 |
| 1.745 e-05 | 2.16 e-05 | 1.4571 | 107.7118 |
| 1.107 e-05 | 2.2469 e-05 | 1.275 | 117.8097 |
| 1.362 e-05 | 3.394 e-05 | 1.1839 | 94.2478 |
| 1.192 e-05 | 2.006 e-05 | 1.867 | 87.5158 |
| 1.618 e-05 | 3.394 e-05 | 1.275 | 97.6138 |
| 1.235 e-05 | 2.160 e-05 | 1.4571 | 107.8097 |

Fig. 4. Initial Population

4.3 Simulation Results

The optimum values of the parameters are estimated by using the proposed algorithm. The initial population was shuffled with the actual parameter values. The estimated parameters after 5 generations were found to be very much close to the actual values. From the simulation results it was found that if the initial population contains the optimal values anywhere in the population, then GA will converge to the Global optimal solution. Tables for initial and final populations are shown in Fig. 4 and 5 respectively. Comparison of outputs are shown in the Figure 3.

5. CONCLUSION AND SCOPE OF FURTHER WORK

In this paper, GA was used to estimate the parameter values of the PS4 actuator with nozzle dynamics. The investigation presented in this paper confirms the possibility of employing GA in parameter estimation for model matching. Although the conventional GA was tested, results are acceptable. It was found that if the initial population does not contain the optimal parameter values, the crossover operation was not sufficient to get the optimum solution, and it was the mutation operator which helped in arriving at the optimum value. When the actual values of the parameters are shuffled into

| J_m | B_m | J_{opt} | σ_{opt} |
|-------------|-------------|-----------|----------------|
| 1.2109 e-05 | 2.1257 e-05 | 1.2548 | 95.7438 |
| 1.2109 e-05 | 2.1257 e-05 | 1.2548 | 95.7438 |
| 1.2109 e-05 | 2.1257 e-05 | 1.2548 | 95.7438 |
| 1.2109 e-05 | 2.1257 e-05 | 1.2548 | 95.7438 |
| 1.2109 e-05 | 2.1257 e-05 | 1.2548 | 95.7438 |
| 1.2109 e-05 | 2.1257 e-05 | 1.2548 | 95.7438 |
| 1.2109 e-05 | 2.1257 e-05 | 1.2548 | 95.7438 |
| 1.2109 e-05 | 2.1257 e-05 | 1.2548 | 95.7438 |
| 1.2109 e-05 | 2.1257 e-05 | 1.2548 | 95.7438 |
| 1.2109 e-05 | 2.1257 e-05 | 1.2548 | 95.7438 |

Fig. 5. Final Population

the population, the GA was found to be converging to the global optimum with the crossover operator.

GAs applied to parameter estimation problem is essentially powerful since it can be applied in situations where gradient methods fail and is not susceptible to problem with local minima. In problems where multiple objective optimization is required, as in the case of system identification in which a number of input signals are to be simultaneously tested, the GA approach can be modified to provide a technique to solve such problem.

ACKNOWLEDGMENT

The author would like to acknowledge the support received from VSSC.

6. REFERENCES

- [1] Goldberg D.E., *Genetic Algorithms in search, optimization and machine learning*, Addison-Wesely, 1989.
- [2] S.S. Rao, *Engineering Optimization, Theory and Practice*, New Age International (P) Ltd, 1998.
- [3] ohn. J. Grefenstette, "Optimization of control parameters for genetic algorithms", *IEEE Transactions on systems, man and cybernetics*, Vol.16, No. 1, pp. 122-128, Jan 1986.
- [4] Raif Saloman, "Evolutionary algorithms and gradient search: Similarities and differences", *IEEE Transactions on Evolutionary computation*, Vol.2, No.2, pp. 45-55, July 1998.
- [5] K.F Man, K.S. Tang, S.K.Wong and W.A. Halang, "*Genetic Algorithms for control and signal processing*", Springer-Verlag, London Ltd, 1997.
- [6] L.Yao and W.A. Sethares, Nonlinear parameter estimation via Genetic Algorithm, *IEEE Transactions on signal processing*, Vol.42, pp. 927-935, April 1994.
- [7] Carlos M. Fnseca, Multiobjective optimization and multiple constraint handling with evolutionary algorithms Part I: A unified formulation, *IEEE Transactions on System, man and Cybernetics- Part A, systems , humans* , Vol. 28, No.1, pp. 26-37, Jan 1998.
- [8] Soderstrom and Stoica, *System Identification*, Prentice Hall International Ltd, 1989.

- [9] Kristinsson, K., System Identification and Control using Genetic Algorithms, *Systems, Man and Cybernetics*, Vol. 22, Issue 5, pp. 1033-1046, August 2002.
- [10] Alonge, F., D'Ippolito, F., Ferrante, G., Raimondi, F.M., Parameter identification of induction motor model using genetic algorithms, *IEE Proceedings of Control Theory and Applications*, Vol. 145, Issue 6, Nov 1998.
- [11] Nur Hayati Kasim, Hairudin Abdul Majid, Azurah A. Samah, Parameter estimation of warranty cost model using genetic algorithm, *International Journal of Soft Computing and Engineering* , Vol.2, Issue 5,pp. 2231-2307, November 2012.
- [12] M. Zagroubaa,A. Sellamia,M. Bouacha, Identification of PV solar cells and modules parameters using the genetic algorithms: Application to maximum power extraction, *Solar Energy*, Vol. 84, Issue 5, pp. 860866, May 2010.