

Genetic Algorithm-based Modeling and Optimization of Control Parameters of an Air Motor

Rapelang R. Marumo and M.O. Tokhi¹

*Department of Mechanical Engineering, University of Botswana, Gaborone.
Botswana; ¹Department of Automatic Control & Systems Engineering,
University of Sheffield Mappin Street. Sheffield, UK*

ABSTRACT

Challenging optimization problems, which elude acceptable solutions via conventional methods, arise regularly in control systems engineering. Genetic algorithms are applied here to optimize the control gains for a controller of a pneumatic drive. We show that, by using minimum information specific to the system, near optimal values of the control gains can be obtained within 10 generations. Two main motivating factors are behind this kind of study; namely, the response of pneumatic drives is very slow, which leads to its inability to attain set points due to high hysteresis. Moreover, the dynamic model of the system is highly nonlinear, which greatly complicates controller design and development. To address these problem areas, two streams of research efforts have evolved—using conventional methods and adopting a strategy that does not require a mathematical model of the system.

KEYWORDS

Air motor, genetic algorithm, PID, recursive least squares

1. INTRODUCTION

This paper reports on investigations into the control of air motor speed using intelligent based methods. In recent years, much attention has been given to the

Reprint requests to: Rapelang R. Marumo, Department of Mechanical Engineering, The University of Botswana, Gaborone. Botswana; e-mail: marumorr@mopipi.ub.bw

optimization of control parameters for fluid power systems, in order to improve their performance. When a PID or a state feedback controller is used to control pneumatic servo systems, it is very difficult in practice to determine the control gains theoretically because of the high nonlinearities of the system and the uncertainty in system modeling. Thus, in general control gains are tuned manually. Manual gain tuning of fluid systems is time consuming even for an experienced control technician, and even then it is often uncertain whether the values obtained are optimal or near optimal. Furthermore, the control gains for a pneumatic servo drive usually depend on both the magnitudes of the load mass and reference position. Gain tuning schemes based on fuzzy logic have in the past been developed to automate the tuning procedure for gain based fluid power system controllers. Practitioners of control systems engineering are increasingly addressing problems for which existing control theory and practice is unprepared. These problems typically require consideration of multiple performance criteria and non-commensurable control variables. The system involved may be difficult to formalize mathematically and may be poorly understood.

Many researchers have considered less conventional techniques in order to find a satisfactory solution. Evolutionary algorithms (EAs) have proven to be one such popular alternative. However, much experience regarding a particular system's behavior is needed to construct a knowledge base for such schemes. The evolutionary algorithm is a robust search and optimization methodology that is able to cope with ill-behaved problem domains, exhibiting attributes such as multimodality, discontinuity, time-variance, randomness, and noise. It permits a remarkable level of flexibility with regard to performance assessment and design specification. GAs are general-purpose search procedures, optimization methods, or learning mechanisms based on Darwinian principles of biological evolution, that is, reproduction and 'the survival of the fittest', along with genetic recombination (Goldberg, 1989). When provided with suitable objective function the performance of the control system can rapidly be optimized by using GA (Varsek, Urbacic and Filipic, 1993). The paper is organized as follows: Section 2 presents the air motor system and identification, Section 3 provides an overview of GAs, Section 4 describes intelligent optimization of air motor control parameters, and Section 5 presents comparative assessment of conventional and intelligent based controllers. The paper is finally concluded in Section 6.

THE AIR MOTOR SYSTEM - IDENTIFICATION PROCEDURE

The aim of this study is to optimize control gains using intelligent based methods, and GAs have proven to overcome the problem of shallow extrema in the solution space and are likely to achieve global solution. Hence, GA's performance over fuzzy and neural networks justifies its choice for the optimization technique used in this study. Before controlling the system, the system must be identified. System identification is one of the most fundamental requirements for many engineering and scientific applications. The objective of system identification is to find exact or approximate models of dynamic systems based on observed input and output data. Well developed techniques exist for parameter estimation of linear models and linear-in-the-parameters nonlinear models. Techniques for selection of structure and for non-linear-in-the-parameters estimation are still the subject of ongoing research. Once a model of the physical system is obtained, it can be used for solving various problems such as, to control the physical system or to predict its behavior under different operating conditions. A number of techniques have been devised by many researchers to determine models that best describe input/output behavior of a system. In many cases when it is difficult to obtain a model structure for a system with traditional system identification techniques, intelligent techniques are desired that can describe the system in the best possible way (Elanayar and Yung, 1994). For ease of reference, the dynamic characteristics of the air motor system have been conveniently divided into three main regions called low speed (below 390 rev/min), medium speed (390 to 540 rev/min) and high speed (540 to 680 rev/min). The system is highly non-linear in the low speed region and hence a controller with intelligence such as a GA-based controller is proposed. There exist many good public domain genetic algorithm packages and none of these provide an environment that is immediately compatible with existing tools in control domains.

2.1 Model validation using correlation tests

Once a model of the system is obtained, it is required to verify whether the model is good enough to represent the system. A number of tests are available in the literature. These include correlation tests, one-step-ahead (OSA) prediction, mean square error, and estimation and test data. In this paper, a system identification strategy is developed for the low speed range of an air motor using GA and OSA

prediction. Correlation tests are very convincing methods of model validation. If the model of a system is adequate, then the residuals or prediction errors $\varepsilon(t)$ should be unpredictable from all linear and nonlinear combinations of past inputs and outputs. The derivation of simple tests can detect these conditions, but it can be shown that the five conditions should hold (Billings and Voon, 1986). Therefore, the system is validated using the following correlation model validity tests:

$$\phi_{\varepsilon\varepsilon}(\tau) = E[\varepsilon(t-\tau)\varepsilon(t)] = \delta(\tau) \quad (1)$$

$$\phi_{u\varepsilon}(\tau) = E[u(t-\tau)\varepsilon(t)] = 0 \quad \forall \tau \quad (2)$$

$$\phi_{u^2\varepsilon}(\tau) = E[(u^2(t-\tau) - u^2(t))\varepsilon(t)] = 0 \quad \forall \tau \quad (3)$$

$$\phi_{u^2\varepsilon^2}(\tau) = E[(u^2(t-\tau) - u^2(t))\varepsilon^2(t)] = 0 \quad \forall \tau \quad (4)$$

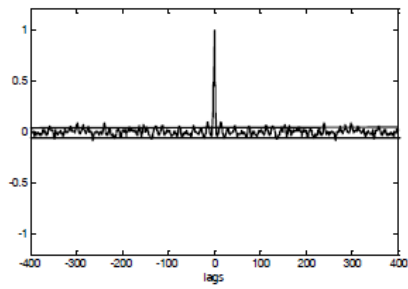
$$\phi_{\varepsilon(u\varepsilon)}(\tau) = E[(\varepsilon(t)\varepsilon(t-1-\tau) - u(t-1-\tau))] = 0 \quad \tau \geq 0 \quad (5)$$

Where $\phi_{u\varepsilon}(\tau)$ indicates the cross-correlation function between $u(t)$ and $\varepsilon(t)$, $\varepsilon u(t) = \varepsilon(t+1)u(t+1)$, and $\delta(\tau)$ is an impulse function.

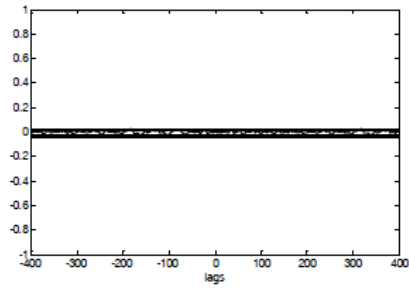
All five tests defined by equations (1) to (5) should be satisfied if the $u(\bullet)$'s and $y(\bullet)$'s used as network input nodes. In practice, normalized correlations are computed. The sampled correlation functions between two sequences $\psi_1(t)$ and $\psi_2(t)$ are given by:

$$\hat{\phi}_{\psi_1\psi_2}(\tau) = \frac{\sum_{t=1}^{N-\tau} \psi_1(t)\psi_2(t+\tau)}{\sqrt{\sum_{t=1}^N \psi_1^2(t) \sum_{t=1}^N \psi_2^2(t)}} \quad (6)$$

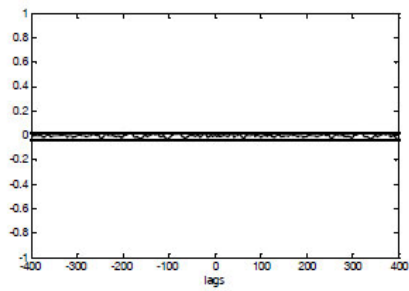
Normalization ensures that all the correlation functions lie in the range $-1 \leq \phi_{\psi_1\psi_2}(\tau) \leq 1$, irrespective of the signal strengths. In general, the correlation will never be exactly zero. Therefore, if the correlation functions in equations (1) to (5) are within the 95% confidence intervals, $\pm 1.96/\sqrt{N}$, where, N is the total number of data points, and then the model is regarded as satisfactory. The representations of five model validity correlation tests are shown in Figure 1. Model validation tests,



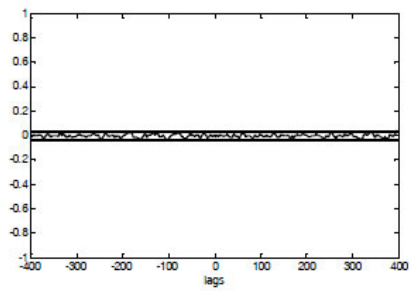
a) Auto-correlation of residuals



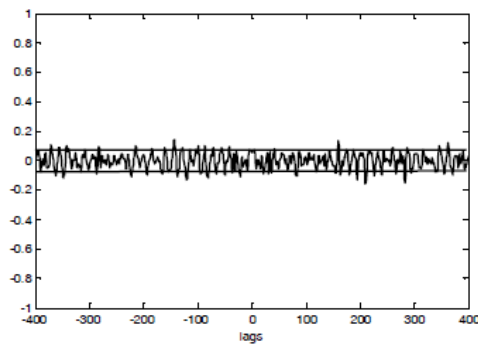
b) Cross-correlation of residuals and input



c) Cross-correlation of residuals and



d) Cross-correlation of residuals input
square square and input square



e) Cross-correlation of residuals and (input*residuals)

Fig. 1: Model validity correlation tests

including training and test validation and correlation tests were carried out to validate the models. It is noted in Figure 1 that all correlation function tests are within the 95% confidence interval, indicating a high level of accurate and unbiased performance of the model in approximating the dynamics of the air motor system

3. OVERVIEW OF GAS

The underlying principles of GAs were first published by (Holland, 1962) and constitute one of a few intelligent techniques commonly used for system identification and modeling of dynamical systems. The major advantage of GA for system identification is that it simultaneously evaluates many points in the parameter space and converges toward the global solution (Kargupta & Smith, 1991). EAs are global, parallel search and optimization methods founded on the principles of natural selection (Darwin, 1859) and population genetics (Fisher, 1930). The correctness of the EA framework as an abstraction of natural evolution has been challenged, for example by Channon and Damper (2000), but this should not be of undue concern to the engineer, who is using the algorithm for its robust search and optimization properties. In general, any iterative, population based approach that uses selection and random variation to generate new solutions can be regarded as an EA. The evolutionary computing (EC) field has its origins in four landmark evolutionary approaches: evolutionary programming (EP) (Fogel, Owen & Walsh 1966), evolutionary strategies (ES) (Schwefel, 1965; Rechenberg, 1973) genetic algorithm (GA) (Holland, 1975), and genetic programming (GP) (Koza, 1992). The GA was popularized by (Goldberg, 1989) and probably as a result, the majority of control applications in the literature adopt this approach. It is important to recognize that boundaries between all four approaches are no longer distinct, with contemporary thinking progressing towards a more holistic EA framework (Michalewicz & Fogel, 2000). The key benefit of EC is flexibility and thus the designer should not feel constrained by requirement to conform to strict classification. EAs work with a population of potential solutions to a problem. Each individual within the population represents a particular solution to the problem, generally expressed in some form of genetic code. The population is evolved, over generations, to produce better solutions to the problem. The technique is efficient in modeling nonlinear systems or if the system has nonlinearities to a degree. A GA is

a population based global search procedure inspired by the laws of natural selection and genetics. The genetic algorithm evolves a collection of elements called a population of individuals. Each individual of the population represent a trial solution to the problem of maximizing a given fitness function. The procedure for identifying a dynamic system consists of four basic steps, as schematically shown in Figure 2.

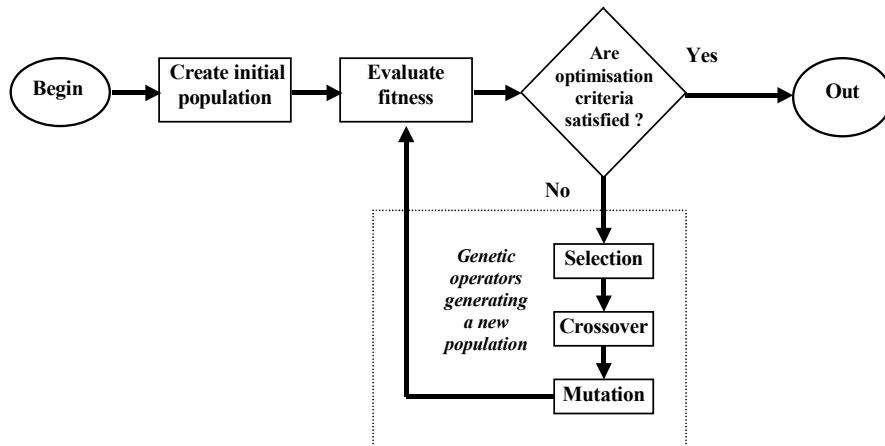


Fig. 2: Schematic of GA's system identification procedure

The standard simple genetic algorithm (SGA) works as follows:

- Step 1. Randomly generate an initial population
- Step 2. Compute the fitness of each individual of the current population
- Step 3. Generate the next population by applying three generic operators: reproduction, crossover, and mutation.
- Step 4. If an end condition is not satisfied, go to step 2. The three genetic operators for the SGA are as follows:
 - a) Reproduction (selection). This operator produces an intermediate population, extracting individuals from the old population by means of roulette wheel selection, where the extraction probability of each individual is proportional to its fitness.
 - b) Recombination (crossover), this is applied to two chromosomes with the probability of recombination, which creates two new chromosomes

by selecting a random position along the chromosomes and splicing the section that appears before the selection position in the first string with the section that appears after the selection position in the second string and vice versa.

- c) Mutation, this is a simple bit wise changing of genes with a small probability, called the probability of mutation.

3.1 Alternative features of EA

Many variations have been developed within the overall EA template, with the aim of improved performance and adaptation to particular problem domain. Thus, it may be more helpful or appropriate to regard EC as a general problem-solving methodology, rather than a specific parameter-less tool. However, many applications have their basis in the standard GA (SGA) presented by (Goldberg, 1989). The specific details of each EA will vary from application to application. The choice of representation, evaluation, and genetic operators must be made with the particular application in mind. A short summary of the key issues and developments made thus far is herein presented. Comprehensive discussion can be found in (Michalewicz & Fogel, 2000).

3.1.2 Representation. The EA works on an encoding of decision variables. The classic GA is to encode each variable as a binary string and concatenate each variable to form a single solution. However, any representation is permitted for which suitable variation operators can be devised. Direct floating-point representations are becoming increasingly popular (Michalewicz, 1966).

3.1.3 Evaluation. The EA requires a scalar value representing the performance of each solution. This may be an absolute quantity or may be relative to other solutions. Techniques for the conversion of raw performance to EA fitness value have received much attention in the approach, especially for multi-objective problems. Individual fitness values may then be modified to encourage niching behavior through the use of fitness sharing (Goldberg & Richardson, 1987).

3.1.4 Selection. Individual solutions must be selected for application of genetic operators and for subsequent reinsertion into the population if a generational gap is used. The standard roulette-wheel selection method is known to produce biased results, leading to a phenomenon known as genetic drift. Two central aims in the development of alternatives are to eliminate statistical bias and to achieve potential parallelism. Other selection methods have been proposed, such as tournaments

between two individuals and stochastic universal sampling (Baker, 1987).

3.1.5 Variation. Genetic operators have been subject to intensive discussion, over both the composition and purpose of the various operators. Some researchers have abandoned recombination, whilst others regard the effect of mutation as minimal. Essentially, recombination tends to direct the search to superior areas of the search space, whilst mutation acts to explore new areas of search space and to ensure that genetic material cannot irretrievably be lost. Choices of values of the parameters of the operators, such as mutation rate and mutation size, are critical to the success of the algorithm. Note that the parameters may vary adaptively over the course of a run in order to improve the search results and these parameters may themselves be subject to evolution (Rechenberg, 1973).

3.1.6 Population. EAs evolve a population of solutions, usually comprising a static number between 20 and 100 individuals (Fleming & Purhouse, 2002), over a number of generations. The population is initialized with random values, possibly seeded with some known good solutions. At each subsequent generation, the population will comprise the results of genetic manipulation, some remnants of past population and possibly few randomly generated individuals. An EA is defined as elitist if it deliberately reintroduces past high-performance individuals. Researchers have also experimented with various population topologies. For example, sub-populations can be implemented as islands, which evolve in parallel, with limited migration of individuals between islands. The island model provides one mechanism for the incorporation of explicit parallelism within an evolutionary algorithm. Single process running on a single processor may represent each subpopulation. The migrants could represent data flow between the processes. Chipperfield and Fleming (1995), provide a broader overview and comparison of parallel EAs. Population based nature of GA renders them well suited to multi-objective optimization problems and enables the evolution of a Pareto-optimal set of solutions (Schroder et al., 2001). MOGA, described by (Fonseca & Fleming, 1998a,b) was used in this work to evolve a set of Pareto-optimal solutions.

3.1.7 Representation utility for control. Typical control problems contain various attributes that an EA must account for in some way or another. These include representation of design parameters, inclusion of constraints, and assessment of performance and methods for coping with likely properties of the fitness landscape.

3.1.8 Numeric representation. Each solution to a problem consists of a number

of design parameters (decision variables). These are encoded as chromosome, which can then be manipulated by genetic operators. Thus, a mapping is required between the actual decision variables (phenotypes) and their genetic equivalent (genotypes). Choice of representation is very flexible, so long as suitable genetic operators can be developed to support the representation. The most popular approach is to encode the set of parameters in a concatenated binary string, as shown in Figure 3.

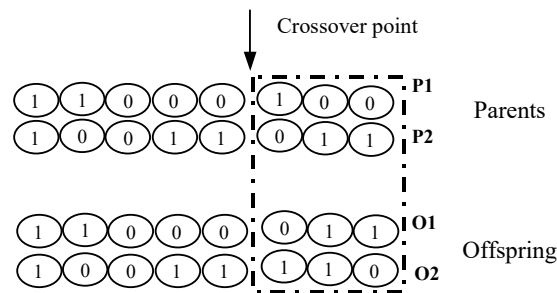


Fig. 3: Mapping between chromosome and decision variables

Both discrete and continuous variables can be catered for. A binary, or n-array coding can be used to represent variables to an arbitrary solution. It should be noted that, however, the size of the search space increases exponentially with chromosome length.

3.1.9 Symbolic representation. The representation of decision variables can be symbolic rather than numeric. The EA known as GP facilitates the representation and manipulation of variable length structures (Koza, 1992). The structures are composed of functions and terminals (potential inputs) that are defined in a library.

3.1.10 Constraints handling. Most engineering problems involve constraints, including the need to have a stable system and to observe actuator performance limits. Five main methods have been proposed to deal with constraints within EA solutions:

1. Coding: The most efficient approach is to embed the constraints in the genetic code, making it impossible to generate infeasible solutions.
2. Penalty functions: Unfortunately it may be impossible, or far from obvious, how to adopt the coding method. In this case, penalty functions can be

incorporated. These assign a very high cost (or correspondingly very low fitness) to infeasible solutions. The approach can be rather ungainly but has been sufficiently effective in many applications.

3. Repair algorithm: As an alternative to simply penalising infeasible solutions, a repair algorithm can be used to attempt to convert them to neighbouring feasible solutions. The fitness of these new solution is typically reduced by an amount equivalent to the cost of the associated repair
4. Indirect method of representation: Instead of repairing the feasible chromosomes, it is also possible to reinterpret the meaning of the existing chromosome. This idea forms the basis of the indirect methods of representation, in which genotypic information passes through an interpreter that builds solutions in the phenotypic domain. In this case, it can prove very difficult to analyse the effects of genetic operators due to the complicated mapping between genotype and phenotype. However, feasible solutions are guaranteed providing that the interpreter is carefully constructed.
5. Multi-objective optimization (MO): The final technique, which may become more popular, is commonly known as MOEA. This expands the research of single objective and its other branch called multi-objective genetic algorithm (MOGA) is proving to yield convincingly promising results. These techniques recast the constraints as objectives to be met and consequently, solved as multi-objective optimization problems.

3.1.11 Cost functions. The method by which potential solutions are assessed is a critical component of EA. If the cost function is inappropriate then the EA search is unlikely to progress in an acceptable direction. Initially the raw performance measures (objectives) must be defined. This is very much application dependent process. The next consideration is the means by which the performance data will be obtained. This is often model based. The raw performance data must then be translated into a non-negative scalar or fitness value. This fitness value represents the expected number of times that an individual will be selected for reproduction. Many engineering problems consists of multiple objectives and for the purpose of EA these must be combined to form a single value. The SGA routine adopted is based on De Jong's first test function. The objective function is defined as described by equation (7).

$$J = \max \sum_{k=0}^{N-1} \sqrt{u_k}, \quad (7)$$

$$[\text{min.range} \leq u \leq \text{max.range}]$$

Where N is the number of control steps over which the problem is to be solved, and u_k is the control input. Note that this is a maximization problem and the GA toolbox routines are implemented to minimize this objective function, i.e. J is multiplied by -1 to produce a minimization problem.

Or

$$f_1(x) = \sum_{i=1}^N x_i^2, \quad (8)$$

$$[\text{min.range} \leq x_i \leq \text{max.range}]$$

This optimization technique can be used as minimization problem or maximization problem as in equation (8), depending on the user's choice. In this study, the approach used was to maximize the function:

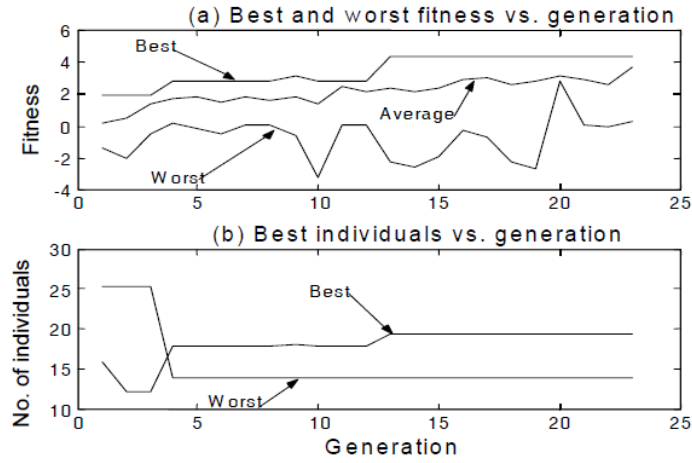
$$z = f(x,y) \quad (9)$$

Where z is the parameter to be maximized, x and y are the considered ranges; f is the fitness function and must always be positive. The system identification procedure has shown that the low frequency model lies within $[-0.5$ and $0.5]$ ranges. Therefore, the fitness function was chosen as positive 5, to ensure that the maximized parameter is always positive.

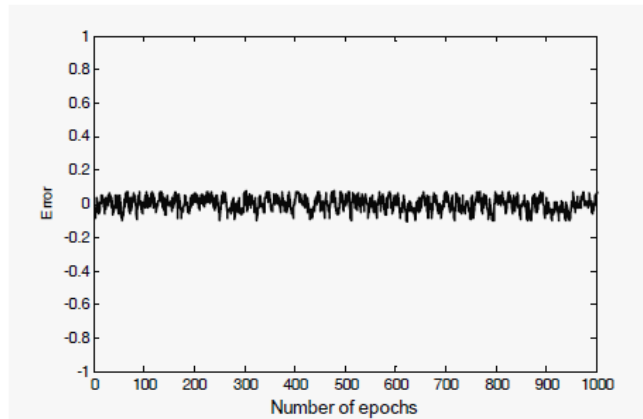
4. OPTIMIZATION OF AIR MOTOR PARAMETERS

Researchers have explored optimization techniques to solve various control engineering problems. Recognizing the difficulty in implementing unconstrained H-infinity controllers, in which the order of the controller is much higher than that of the plant. Chen and Cheng (1998) proposed a structure specified H-infinity controller. EA was used to search for good solutions within the admissible domain of the controller parameters (obtained via Routh-Hurwitz stability criterion). Fonseca and Fleming (1998) applied a MOGA to the optimization of the low-pressure spool

speed governor of a Rolls-Royce Pegasus gas turbine engine. The results highlighted the importance of progressive preference articulation and interaction with the designer, since only a small proportion on the non-dominated set was found to be of practical relevance.



i. Actual and predicted output



(ii) Error between actual and predicted output

Fig. 4: Prediction using GA

In the air motor study, investigations were carried out using GA based OSA

prediction with different initial values and operator rates. GA parameters used to find results in this study are population size = 20, maximum generation = 1000, generation gap = 0.8 crossover = 0.9 and mutation = 0.09. The termination of GA cannot be predefined, as it is a stochastic search process. It is, to a considerable extent, problematic and application dependant as well. Usually, GA search process is terminated after a pre-specified number of generations and quality of the best members is tested with respect to a desired result. If the result is not acceptable at this level, a fresh search may be initiated by changing the parameters and operators involved in GA. Besides the factors described above, some other important issues such as parallelism, computation time and multiple objectives have to be considered while solving a problem with GA-based search process. Though, GA is highly parallel in nature, yet it is sometimes painfully time consuming, which requires a great deal of attention in applications such as control and signal processing. A number of methods including GA-based parallel methods have been investigated to enhance the computational speed of GA. Another feature of the GA that has made it very attractive among scientists and engineers is its capability to solve multi-objective problems. Although similar in concept, many variations exist in GA (Chipperfield, Fleming, Pohlheim & Fonseca, 1994).

Figure 4 shows the convergence of the GA algorithm. It is shown that by using minimum information specific to the system, near optimal values of control gains can be obtained within ten generations. GA converges within the first ten generations and repeat again for the next ten. The GA achieved the best mean-square error level of 0.00285. Fig.4 (i) shows the worst, average and best fitness and Fig.4 (ii) shows elitism where it is ensured that, the most fit individual in each generation will be carried to the next generation without being modified (survival of the fittest).

4.1 Modeling with recursive least squares

For comparison purposes, the air motor speed characterization with same input/output data was also modeled using a recursive least squares (RLS) algorithm. When a data containing input, u , and output, y from an unknown system is available, the original system can be estimated using recursive least squares. The objective is to look for the minimum value of the integral absolute errors, (iae or θ). The loss function is then defined as:

$$V(\hat{\theta}, k) = \frac{1}{2} \sum_{i=1}^k \lambda^{k-i} (y(i) - \phi^T(i)\hat{\theta}(k))^2 \quad (10)$$

where λ is called the forgetting factor and $0 < \lambda \leq 1$.

The update gain $L(k)$ can be expressed as:

$$L(k) = P(k-1)\phi(k)(\lambda + \phi^T(k)P(k-1)\phi(k))^{-1} \quad (11)$$

and the covariance matrix $P(k)$ can be expressed as:

$$P(k) = (I - L(k)\phi^T(k))P(k-1)\frac{1}{\lambda} \quad (12)$$

where I is the least square identity matrix.

θ can then be calculated recursively using as:

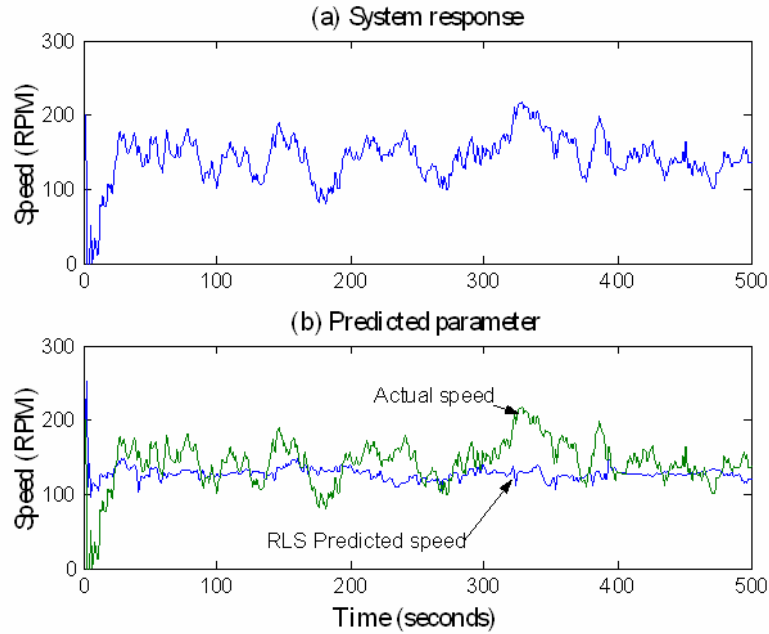
$$\hat{\theta}(k) = \hat{\theta}(k-1) + L(k)(y(k) - \phi^T(k)\hat{\theta}(k-1)) \quad (13)$$

The RLS type used in this study is the forgetting factor type. The concept of forgetting is such that older information is gradually discarded in favour of the most recent information or giving less weight to older data and more weight to recent data (Chipperfield et al. 1994). Fig. 5 shows the RLS convergence. An integral average error (IAE) of 2.83 was achieved and comparison shows that a better air motor speed characterisation is achieved using GAs.

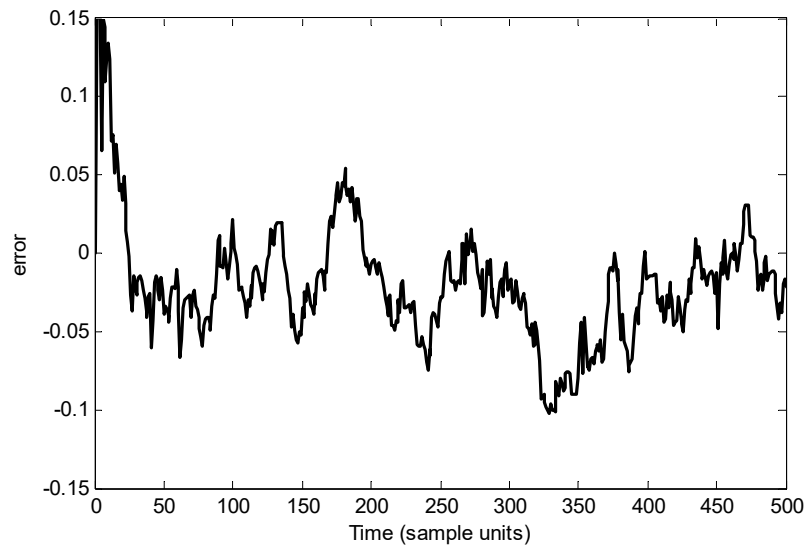
4.2 Modeling and control with PID

An early use of EAs was as an alternative means of tuning proportional-integral-derivative (PID) controllers. Oliver et al. (1991) used a standard GA to determine initial estimates for the values of PID parameters. They applied their methodology to a variety of classes of linear time-invariant (LTI) system encompassing minimum phase, non-minimum phase and unstable systems. They improved the efficiency of their algorithm by identifying ancestral (already-assessed) chromosomes and avoiding re-evaluation of these. Wang and Kwok (1992) tailored an EA using inversion and pre-

selection micro-operators to PID controller tuning. They stressed the benefit of flexibility with regard to cost function (there be no requirement for mathematical tractability) and in particular, alluded to the concept of Pareto-optimality. In one such example, Dakev, Whidborne, Chipperfield and Fleming (1997) applied MOGA to indirect H-infinity design of an electromagnetic suspension (EMS) control system for a



(i) Actual and predicted output



(ii) Error between actual and predicted output

Fig. 5: Prediction using RLS

Maglev vehicle. This is a critical, inherently unstable system that requires active control. The approach permitted good visualization of design trade-offs, such as that exists between the air gap and the passenger cabin acceleration, underpinned by a robust H-infinity controller for all alternative designs. In an independent enquiry, Porter and Jones (1992) proposed an EA-based technique as a simple generic method of tuning digital PID controllers. Vlachos et al. (1999) applied an EA to the tuning of decentralized proportional-integral (PI) controllers for multivariable processes. Controller performance was defined in terms of time domain bounds on the closed loop responses for both reference following loop interactions. This approach afforded good visualization of the performance of potential solutions. PID controller tuning is an art first proposed by (Vahidi, Stefano & Peng, 1942). It is possible to express controller as a PID in order to have better understanding of the meaning of coefficients.

Characterization of an air motor speed using PID control techniques is shown in Fig. 6. Just like in GA and RLS optimization, the idea is to find the control sequence that minimizes the cost function. The equation of PID can be expressed as:

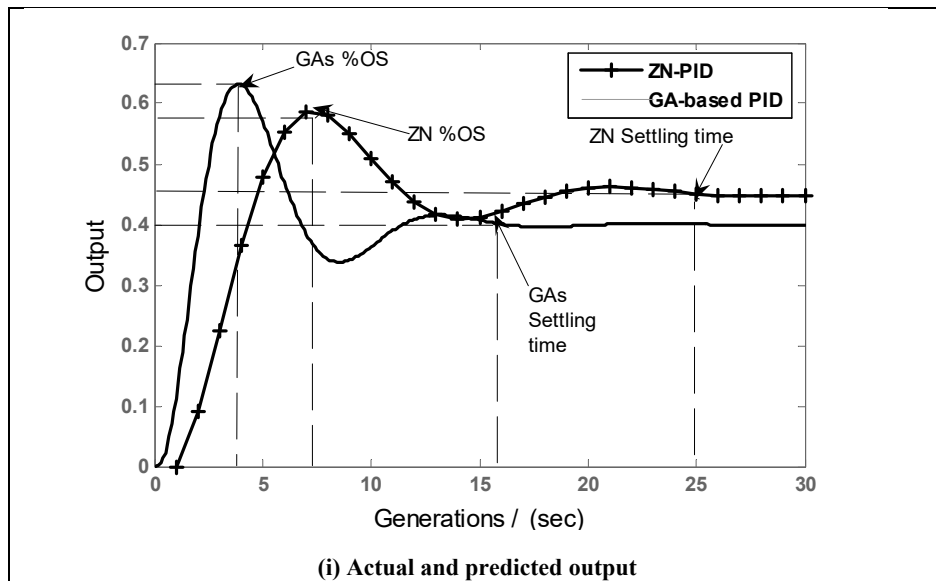
$$u(t) = [K_p + \frac{K_i}{1-z^{-1}} + K_d(1-z^{-1})](r(t-1)-y(t-1)) \tag{14}$$

where $u(t)$, $y(t)$ and $r(t)$ represent the system’s input, output and the reference signal respectively.

It can easily be noticed that, the motor has peculiar behavior at low speeds. PID’s do not fare well compared to GAs and RLS. It is however known that PID technique is a time consuming method, which needs experience, and even the most experience technician would take a considerable length of time to tune the plant. The PID parameters (K_p , K_i and K_T) used in this study are 0.001, 10, and 0.01 respectively.

6.0 COMPARATIVE ASSESSMENT BETWEEN GA-BASED, RLS AND ZN PID CONTROLLERS

A comparative assessment of the above techniques reveals that, PID is unable to handle the non-linear dynamics of the air motor system at low speeds. Likewise, recursive least squares without forgetting factor did not fair well when presented with



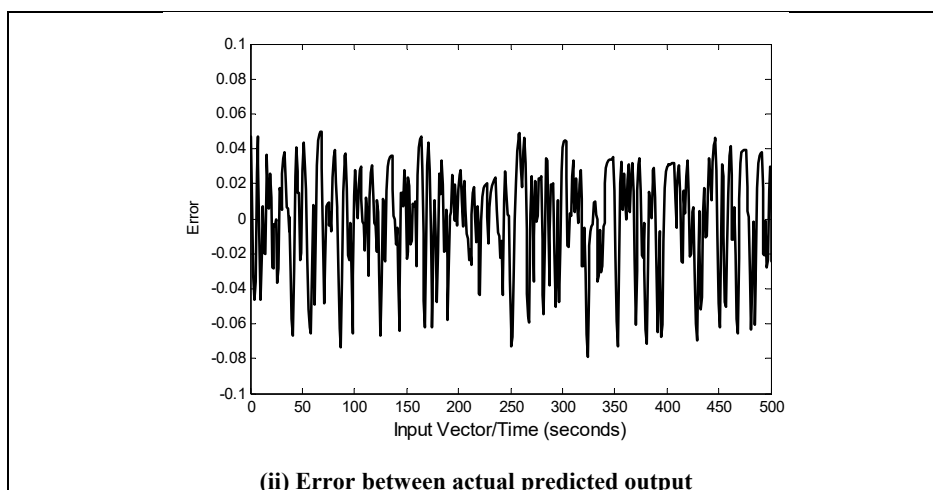


Fig. 6: Prediction using PID

low speed data. Recursive least squares with forgetting factor between 0.95 to 1 produced better results of an air motor model when presented with low speed data. A fixed forgetting of 0.95 was used in this study. GA-based models produced better results and outweighed its RLS counterpart. GA-based controller shows a close match (minimum mismatch) between the actual and the predicted output, indicating good tracking capability.

7.0 CONCLUSION

An optimization process based on GA has been implemented in a typical industrial environment characterizing an air motor system. Practical constraints influenced the way in which the tuning process was implemented. Nevertheless, significant controller performance improvements were realized using EA's automated process, which could be implemented in a much shorter time. The system has been modeled with GA modeling technique. It has been shown that by using the minimum information specific to the system, which may be easily estimated using the system's input/output together with certain trial and error experiments, optimal gains for good control performance gains can be achieved within 10 generations. Comparison of the results shows that GA based modeling technique has performed quite well in approximating the response of the system against its RLS counterpart. The approach adopted is the first step forward towards controller optimization of the air motor parameters using GA based techniques. In summary, significant future EA applications

are likely to focus on two main areas: off-line, multi-criterion, design-aid tools and robust control, on-line search and improvement algorithms. Research in both these areas is shaping up and the rate at which EA is applied to real world problems is increasing at fast pace. Further investigation into this technique, should be undertaken, where by a MOEA and MOGA neuro-fuzzy based controllers are proposed.

REFERENCES

- Baker, J.E. 1987. Reducing bias and inefficiency in the selection algorithm. . *Proceedings of the second international conference on genetic algorithms*. Cambridge, USA, p. 14-21.
- Billings, S.A. and Voon, W. 1986. Correlation-based model validity tests for nonlinear models. *International Journal of Control* 44, pp. 235-244.
- Channon, A.D. and Damper, R.I. 2000. Towards the evolutionary emergence of increasingly complex advantageous behaviours. *International Journal of Systems Science*, **31(7)**, 843-60.
- Chen, B.S. and Cheng, Y.M. 1998. A structure-specified H-infinity optimal control design for practical applications: A genetic approach. *IEEE Transaction on Control Systems Technology*, **6(4)**, 707-718.
- Chipperfield, A.J. Fleming, P.J. Pohlheim H. and Fonseca, C. 1994. A genetic algorithm toolbox for MATLAB. *Proceedings of the International Conference in Systems Engineering, Coventry, UK*.
- Dakev, N.V., Whidborne, J.F., Chipperfield, A.J and Fleming, P.J 1997. Evolutionary H-infinity design of an electromagnetic suspension control system for a maglev vehicle. *Proceedings of the Institution of Mechanical Engineers Part I*, **211**, 345-55.
- Darwin, C. 1859. *The origin of species*. London: John Murray.
- Elanayar, S.V.T. and Yung C.S. 1994. Radial basis function neural networks for approximation and estimation of nonlinear stochastic dynamic systems. *IEEE Transaction on Neural networks*, **5(4)**, 594-603.
- Fisher, R.A. 1930. *The genetical theory of natural selection*: Oxford: Clarendon Press.
- Fleming, P.J. and Perhouse, R.C. 2002. Evolutionary algorithms in control systems engineering: a survey. *Control Engineering Practice* **10**, 1223-41.
- Fogel, L.J., Owens, A.J., and Walsh, M.J. 1996. *Artificial intelligence through simulated evolution*. New York: Wiley.
- Fonseca, C.M., and Fleming, P.J. 1998a. Multi-objective optimization and multi constraint handling with evolutionary algorithms-Part 1: A unified formulation. *IEEE Trans System, Man and Cybernetics*, **28(1)**, 26-37.
- Fonseca, C.M., and Fleming, P.J. 1998b. Multi-objective optimization and multi constraint handling with evolutionary algorithms-Part II: Application example.

- IEEE Trans System, Man and Cybernetics*, **28(1)**, 38-47.
- Goldberg, D.E. 1989. Genetic algorithms, in search, optimization and machine learning, Addison Wesley Publishing Company.
- Goldberg, D.E. and Richardson, J. 1987. Genetic algorithms with sharing for multimodal function optimization. *Proceedings of the Second International Conference on Genetic Algorithms*. Cambridge, MA, USA, p. 41-49.
- Holland, J.H. 1975. *Adaptation in natural and artificial systems*. Ann Arbor: The University of Michigan Press.
- Kargupta, H. and Smith, R.E., 1991. System identification with evolving polynomial networks. *Proceedings of the 4th International Conference on Genetic Algorithms*, University of California, San Diego, USA, p. 370-6.
- Koza, J.R. 1992. *Genetic programming-on the programming of computers by means of natural selection*. Cambridge, MA: the MIT Press.
- Kristinsson K. and Dumont, G.A. 1992. System identification and control using genetic algorithms, *IEEE trans. Syst. Man Cybern.* **22(5)**, p. 1032-46.
- Leontaritis, T.J., and Billing, S.A 1985. Input-output parametric models for non-linear systems, Part 1: Deterministic non-linear systems and Part 2: Stochastic non-linear systems. *International Journal of Control*, **41(2)**, m303-328 and 329-344.
- Marumo, R. and Tokhi, M.O. 2003. Modeling and control of pneumatic motor using neural networks, *AFCON*, Cape Town, South Africa, p. 100-112.
- Marumo, R. and Tokhi, M.O. 2004. Intelligent modeling and control of a pneumatic motor, *Proc. 17th Canadian Conf. on Electrical and Computer Engineering*, Niagara Falls, Canada, p.1163-66.
- Michalewicz, Z. 1966. *Genetic algorithms+data structure=evolution programs*. Berlin: Springer.
- Michalewicz, Z. and Fogel, D.B. 2000. *How to solve it: modern heuristics*. Berlin, Springer.
- Oliviera, P., Sequeira, J. and Senticiro, J. 1991. Selection of controller parameters using genetic algorithms. In *Engineering Systems with Intelligence. Concepts, Tools and Applications*, edited by S.G. Tzafestas, Dordrecht, Kluwer, p. 431-8.
- Porter, B. and Jones, A.H 1992. Genetic tuning of digital PID controllers. *Electronics Letters*, **28(9)**, 843-4.
- Rechenberg, I. 1973. *Evolutionsstrategie: Optimierung technischer systeme nach prinzipien der biologischem evolution*. Stuttgart: Frommann-Holzboog.
- Schroder, P., Green, B., Grum, N. and Fleming, P.J. 2001. On-line evolution of robust control systems: An industrial active magnetic bearing application. *Control Engineering Practice* **9(1)**, 37-49.
- Schwefel, H.P. 1965. *Kybernetische Evolution als Strategie der Experimentellen Forschung in der Strömungstechnik*. Diploma Thesis, Technical University of Berlin.
- Tzes, A., Peng, P.Y. and Guthy, J. 1998. Genetic-based fuzzy clustering for DC-motor friction identification and compensation. *IEEE Transaction on Control Systems Technology*, **6(4)**, 462-472.

- Vahidi, A. Stefanopoulo A. and Peng, H. 1942. Recursive least squares with forgetting for online estimation of vehicle mass and road grade: theory and experiments. Available at: http://ghost.engin.umich.edu/JournalPaperMassGrade_Final.pdf
- Varsek, A., Urbacic T. and Filipic, B. 1993. Genetic algorithms in computer design and tuning, *IEEE Trans. Sys. Man. And Cyber.*, **23(5)**, 1330-9.
- Vlachos, C., Williams, D. and Gomm., J.B. 1999. Genetic approach to decentralised PI controller tuning for multivariable process. *IEE Proceedings-Control Theory and applications*, **146(1)**, 58-64.
- Wang, P. and Kwok, D.P. 1992. Auto tuning of classical PID controllers using an advanced genetic algorithm. *International conference on Industrial Electronics, Control, Instrumentation and Automation (IECON 92)*, 3, 1224-9.