Monograph:

Rodriguez-Vazquez, K., Fonesca, C.M. and Fleming, P.J. (1999) Identifying the Structure of Non-Linear Dynamic Systems Using Multiobjective Genetic Programming. Research Report. ACSE Research Report 745 . Department of Automatic Control and Systems Engineering

Department of
AUTOMATIC
CONTROL
and SYSTEMS
ENGINEERING
UNIVERSITY of SHEFFIELD

IDENTIFYING THE STRUCTURE OF NON-LINEAR DYNAMIC SYSTEMS USING MULTIOBJECTIVE GENETIC PROGRAMMING

K. Rodríguez-Vázquez*, C.M. Fonseca and P.J. Fleming***

** Department of Automatic Control and Systems Engineering*
University of Sheffield, Mappin Street S1 3JD, Sheffield, U.K.
katya@acse.shef.ac.uk, P.Fleming@sheffield.ac.uk

***A.D.E.E.C.-U.C.E.H., Universidade do Algarve*
Campus de Gambelas, 8000 FARO, Portugal

Abstract

A method for the identifying the structure of non-linear polynomial dynamic models is presented. This approach uses an evolutionary algorithm, Genetic Programming, in a multiobjective fashion to generate global models which describe the dynamic behaviour of the non-linear system under investigation. The introduction of the validation stage of system identification into the multiobjective tool is also explored, in order to direct the identification process to a set of global models of the system.

Keywords: multiobjective optimisation, gas turbine engine, non-linear system identification, genetic programming.

Research Report #745

April 1999

IDENTIFYING THE STRUCTURE OF NON-LINEAR DYNAMIC SYSTEMS USING MULTIOBJECTIVE GENETIC PROGRAMMING

K. Rodríguez-Vázquez*, C.M. Fonseca and P.J. Fleming***

** Department of Automatic Control and Systems Engineering*
University of Sheffield, Mappin Street S1 3JD, Sheffield, U.K.
katya@acse.shef.ac.uk, P.Fleming@sheffield.ac.uk
***A.D.E.E.C.-U.C.E.H., Universidade do Algarve*
Campus de Gambelas, 8000 FARO, Portugal

Abstract. A method for the identifying the structure of non-linear polynomial dynamic models is presented. This approach uses an evolutionary algorithm, Genetic Programming, in a multiobjective fashion to generate global models which describe the dynamic behaviour of the non-linear system under investigation. The introduction of the validation stage of system identification into the multiobjective tool is also explored, in order to direct the identification process to a set of global models of the system.

Keywords: multiobjective optimisation, gas turbine engine, non-linear system identification, genetic programming.

1 INTRODUCTION

System identification is concerned with building a model from input-output observations of a system, creating a mathematical description of the system. Linear system identification methods have been widely covered (e.g. Söderström and Stoica, 1989; Ljung, 1987). However, most real systems are non-linear and linear models may not adequately describe their associated dynamic behaviour (Haber and Unbehauen, 1990). But, non-linear system identification remains a difficult task because, often, no initial information is generally available about the system structure Also, typically, the derived model should satisfy, in some optimal fashion, a number of objectives involving, for example, performance and model parsimony. Here we describe a multiobjective optimisation approach to non-linear system identification based on the genetic programming (GP) paradigm, which is used to identify NARMAX model structures.

Previous work (Rodríguez-Vázquez et al., 1997a) has investigated an alternative identification approach based on genetic programming (Koza, 1992) and the NARMAX model (Leontaritis and Billings, 1985). Although it used a single measure of performance, it proved effective. Recently, Rodríguez-Vázquez et al. (1997b) have translated this method into a multiobjective genetic programming (MOGP) approach.

This work is further developed here and applied to the identification of NARMAX model structures of a real system, a Rolls-Royce Spey gas turbine engine. The paper is organised as follows. Section 2 introduces Evolutionary Algorithms (EAs), focusing on Genetic Programming (GP). Section 3 presents a brief introduction of multiobjective optimisation and the definition of Pareto-optimality. In sections 4 and 5, the formulation of the identification problem as a multiobjective optimisation problem is described. In section 6, two identification examples are analysed: a simulated system and an actual systems. Finally, conclusions are drawn.

2 EVOLUTIONARY ALGORITHMS: GENETIC PROGRAMMING

Evolutionary Algorithms (EA) are computing paradigms which apply the theory of Natural Selection or *survival of the fittest* to a population of individuals (or candidate solutions) in order to produce fitter individuals (improved candidate solutions) in pursuit of a prespecified objective. Different classes of these algorithms (*Evolution Strategies*, ES, *Evolutionary Programming*, EP and *Genetic Algorithms*, GA) were simultaneously developed during the 1960's. The first work on developing Evolution Strategies was developed by Schwefel and Rechenberg in Germany (Schwefel, 1965; Rechenberg, 1973), while around the same time, Fogel, Owens and Walsh (1966) and Holland (1975) began work in the field of Evolutionary Programming and Genetic Algorithms, respectively, in the United States.

These three evolutionary approaches have been the starting point for the development of new forms of EAs. one of these being the concept of Genetic Programming (Koza. 1992). which is a subclass of genetic algorithms. Because this work is based on GPs. a general description of GAs and GPs follows.

2.1 Genetic Algorithms

Genetic algorithms, the evolutionary algorithm proposed by Holland and the most popular of the EC methods, consists of a reproductive plan for selecting successful genotypes (individuals) to be used to create the offspring of a new population, and a set of genetic operators, crossover and mutation. The basic GA works on a fixed-length bit string encoding, where the problem addressed is defined in an objective function that indicates the fitness of any potential solution. Here, individuals are considered at the phenotypic level, which is the value in the domain over which the objective function is defined. On the other hand, an individual's genotype is the representation of its phenotype at a lower level, which the computer stores and the GA manipulates.

The general procedure for a GA is shown in Figure 1. The algorithm starts by creating a random initial population of individuals or candidate solutions, (*Initialise P(t=0)*). Then, each individual, x_i, where $i=1,...,N$, is mapped onto the decision variable domain or phenotypic level, evaluated according to a specific function and assigned a *fitness* measure $\mu(x_i)$, (*Evaluate P(t=0)*).

```
t = 0
Initialise P(t)
P(t) = [x1 ... xN]
Evaluate P(t)
F(x1 ... xN) = [μ1 ... μN]
while (Termination Criterion not fulfilled) do {
Parent Selection P(t+1)
Crossover P(t+1)
Mutate P(t+1)
Evaluate P(t+1)
P(t) = P(t + 1)
t = t + 1
}
```

Figure 1. Basic structure of a genetic algorithm.

The parent selection process uses this fitness measure to determine the selection of potential individuals (*Parent_Selection P(t)*) to be recombined (*Crossover P(t)*) and mutated (*Mutate P(t)*). The crossover operator causes that pair of individuals (parents) to exchange genetic information by selecting a random position along the string. The mutation process creates a new individual (offspring) by altering the genetic information of the parent string according to some rules. Finally, the new individuals are evaluated in order to assign them a fitness measure creating a new population. This process continues until some termination criterion is satisfied.

2.2 Genetic Programming

Genetic Programming (Koza, 1992; 1994) is a subclass of GAs, in which the potential solutions of the population are expressed as programs instead of individuals represented as bit strings. The fact that many problems can be expressed as computer programs makes GP a more powerful tool than its predecessor, the GA. Here, these programs, which are composed of functions and terminals appropriate to the problem domain, are encoded as hierarchical tree structures, providing a dynamic and variable representation. Figure 2 shows an example of a hierarchical tree that expresses the following computer program (in reverse Polish notation),

(/ (- (a a) (* b b)) (* (2 (+ a c))))*

which is equivalent to

$$\frac{a^2 - b^2}{2(a+c)}$$

The internal nodes of the tree structure are elements from the function set (operations), and leaf nodes are the input data from the terminal set. In this example T = {a,b,c} and F = {'+', '-', '*', '/'}.

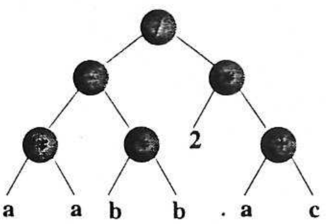


Figure 2. Hierarchical tree encoding.

Genetic Operators

Like the standard GA, the two main operators are crossover and mutation. The crossover operation produces a pair of computer programs that inherit characteristics from both parent programs by selecting a random node in each of the hierarchical trees structures of the parents (Figure 3a) and exchanging the associated subexpressions (Figure 3b). Because of the dynamic representation used in genetic programming, the parents are typically of different size, shape and content, and the offspring are also generally different. Note that parent tree 1

(Fig. 3a) and the tree shown in Figure 2, are slightly different but both represent the same computer program. Thus, in genetic programming, the process of mapping the genotype onto the phenotype does not correspond to a one-to-one relationship; a particular computer program can be expressed by more than one different tree structure allowing a diversification of the population.

Mutation is performed by randomly selecting a node that can be an internal or terminal node, and replacing the associated subexpression with a randomly generated subprogram.

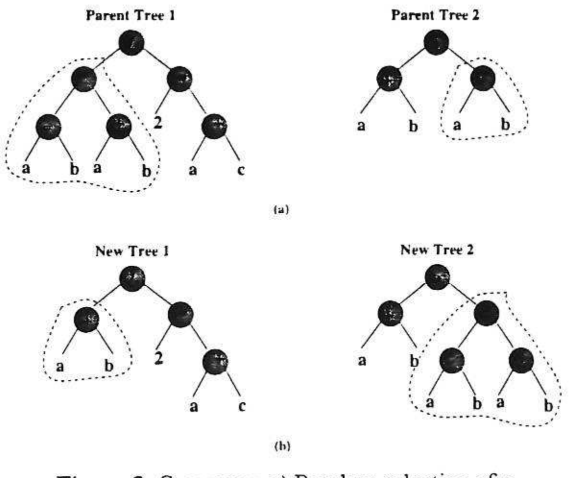


Figure 3. Crossover. a) Random selection of a subexpression. b) Offspring.

The fitness measure of each computer program is assigned in terms of how well it performs in the particular problem domain. This fitness value depends on the problem but is generally defined as the error produced by the computer program. In conventional genetic programming approaches, and, in general, any evolutionary algorithm, the assigned fitness measure is based on the evaluation of a scalar function. But, these population-based methods possess the characteristic of simultaneously searching for multiple solutions and, more, can evaluate several aspects of the problem. For this reason, this work presents an extension of the conventional or single fitness measure genetic programming mapping into a multiobjective genetic programming approach. The next section describes general concepts in multiobjective optimisation and introduces the MultiObjective Genetic Algorithm (MOGA) that is the basis of the approach reported here.

3 MULTIOBJECTIVE OPTIMISATION

In many practical applications, design problems may be represented by a set of competing requirements (or objectives) which can be formulated as a multiobjective optimisation problem. There is seldom a unique solution to such problems since objectives may be in conflict with one another. Rather, a set of solutions emerges where, in the absence of information concerning the importance of each objective relative to the others, each solution is deemed equivalent to the others in the set. Thus, the concept of multiobjective optimisation is defined as the problem of finding the vector of decision variables $\bar{x}$, that optimises the n components vector function

$$\bar{f}(\bar{x}) = \left(f_1(\bar{x}),\dots,f_n(\bar{x})\right) \quad (1)$$

producing a set of equally efficient alternative solutions known as the non-dominated or Pareto-Optimal set (Ben-Tal, 1980).

Pareto-Optimality

The concept of Pareto-optimality or nondominance may be described in the context of the following minimisation problem. Given two n component objective function vectors, $\bar{f}_u$ and $\bar{f}_v$, one can say that $\bar{f}_u$ dominates $\bar{f}_v$ (is Pareto-optimal) if

$$\forall i \in \{1,\dots,n\}, f_{u_i} \le f_{v_i} \wedge \exists i \in \{1,\dots n\}, f_{u_i} < f_{v_i} \quad (2)$$

Then, the set of all Pareto-optimal decision vectors is called the Pareto-optimal or admissible set of the problem.

A number of researchers have used evolutionary algorithms, taking advantage of their population-based approach, with the aim of simultaneously optimising multiple functions. Goldberg (1989) pointed out that the first attempt at multiobjective optimisation using evolutionary algorithms was made by Rosenberg (1967). He suggested a multiple properties function for the simulation of a population of single-celled organisms. However, he only considered a single property in his simulation, but it was the beginning of further multiobjective evolutionary approaches.

Since Rosenberg's work, a variety of approaches have appeared and these are summarised in Fonseca and Fleming (1995). One class - aggregating methods - combines multiple objectives into a single fitness measure to enable evolutionary algorithms to operate on scalar fitness information. Previous aggregating approaches using genetic programming have had the aim of generating parsimonious computer programs by incorporating the number of nodes included in their associated tree representation into the fitness measure (Koza, 1992). Iba et al. (1994) and Zhang and Mühlenbein (1996) have proposed a Minimum Description Length (MDL) based fitness function which is a trade-off of the model code length and the residual error. Siegel and Chaffe (1996) have introduced a different approach that incorporates a time constraint in order to penalise evolved programs that take excessive execution time.

Other multiobjective evolutionary approaches do not use a combined fitness function. Instead, they assign a fitness measure on the basis of a separate evaluation of each of the multiple objectives. An example is the MultiObjective Genetic Algorithm (MOGA) approach proposed by Fonseca and Fleming (1993). This is a Pareto-based technique which is based on the concept of nondominance or Pareto-optimality. Here, fitness of an individual is assigned on the basis of relative non-dominance, i.e. all non-dominated individuals in the population are assigned rank 1, those individuals dominated by one or more points are ranked 2 or higher. Fonseca and Fleming's approach also includes schemes to combat the formation of lethals (mating restriction) and genetic drift (fitness sharing) - special problems arising from the evolution of a set of Pareto-optimal solutions.

The MOGA approach is extended here to be used in conjunction with genetic programming to provide a multiobjective optimisation tool in which we can include an alternative way of controlling the growth of genetic programming parameters in the growth tree. The next section presents a general description of MOGA.

3.1 Multiobjective Genetic Algorithm

The multiobjective genetic algorithm approach proposed by Fonseca and Fleming (1993) uses a rank-based fitness assignment, where the rank of a certain individual x_i at generation t is related to the number of individuals $p_i(t)$ in the current population by which it is dominated. This is given by

$$rank(x_i, t) = 1 + p_i(t). \qquad (3)$$

All non-dominated individuals are assigned rank 1 and remaining individuals are penalised according to equation (3).

Fitness Assignment

Fitness is assigned by interpolating from the best individual (rank=1) to the worst, and then the fitness assigned to individuals with the same rank is averaged where the global population fitness is kept constant. However, such fitness assignment tends to produce premature convergence (Goldberg, 1989) due to the fact that all non-dominated (best rank) points are considered equally fit (Figure 4).

In order to overcome this deficiency, Fonseca and Fleming have used a niche induction method to promote the distribution of the population over the Pareto-optimal front in order to maintain diversity. This is achieved by a method of fitness sharing which encourages the reproduction of isolated individuals and favours diversification.

Preference Information

Preference information is introduced in the form of a goal vector, which provides a means of evolving only a specific region of the search space. This allows the decision maker to focus on a region of the Pareto front by providing external information to the selection algorithm.

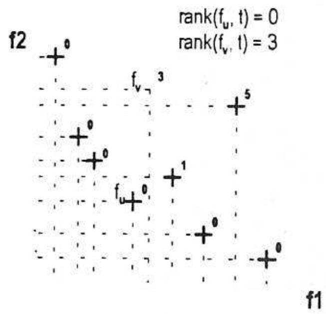
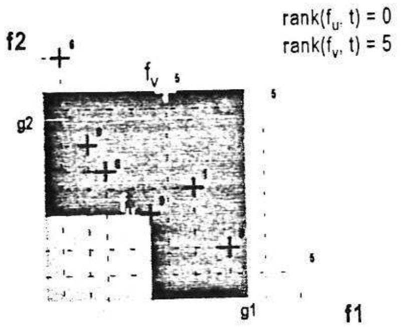


Figure 4. Pareto-Ranking without preference information.

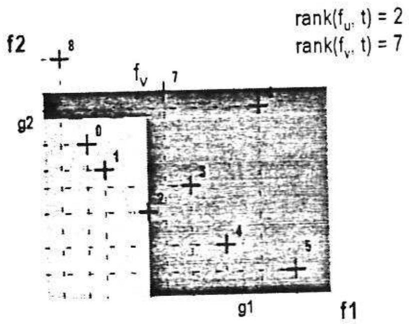
The ranking procedure described previously is modified to introduce the goal information by altering the way in which individuals are compared with one another. Degradation in vector components that meet their goals is acceptable here, provided that it results in the improvement of other components that do not satisfy their goals and provided that goal boundaries are not violated. This permits discrimination between individuals (solutions) even though they are non-dominated. This concept is formalised in terms of a transitive relational operator (preferability), instead of the simple *partially less than* operator, based on Pareto-dominance. The preferability operator additionally takes into account whether or not the objectives meet their goals. Figure 5 shows a bi-objective example of the Pareto-ranking preference information tool. In Figure 5a, both objectives are assigned the same priority, whereas Figure 5b illustrates Pareto-ranking when one objective has a higher priority than the others.

The combination of the notion of preferability coupled with the concept of nondominance introduces a preference articulation framework for multiobjective and constraint optimisation. Because rank-based fitness assignment and fitness sharing of MOGA take place in the objective function domain instead of the parameter domain, they can be directly applied to genetic programming. Thus, the structure of MOGA can be mapped onto genetic programming by introducing a hierarchical tree representation with its associated genetic operators.

The concepts of multiobjective fitness function and genetic programming have been previously applied by Langdon (1995). He evolved list data structures, with genetic programming defining a list which supports ten different operations. Here, each operation is a separate program tree within a composite individual and has its own fitness sub-core. He used the Pareto-ranking method proposed by Goldberg (1989), with tournament selection, in order to choose which individual to breed from or to remove from the population. However, he did not consider prioritisation of objectives.



(a)



(b)

Figure 5. Pareto-ranking with preference information:
a) f1 and f2 have the same priority;
b) f2 has a higher priority than f1.

4 REPRESENTATION OF NON-LINEAR SYSTEMS

While there are many model representations that may be used in system identification studies, this work is based upon an input-output model of a system. To describe this relationship, Leontaritis and Billings (1985) have introduced a non-linear difference equation model known as the NARMAX (Non-linear AutoRegressive Moving Average with eXtra inputs) model. This model represents a wide class of non-linear systems expressed by the following equation,

$$y(k) = F^{\ell}\big(y(k-1),\ldots,y(k-n_y),u(k-1),\ldots,$$
$$u(k-n_u),e(k-1),\ldots,e(k-n_e)\big)+e(k) \qquad (4)$$

where $y(k)$, $u(k)$ and $e(k)$ represent the output, input and noise signals, respectively, n_y, n_u, and n_e are their associated maximum lags, and F is some unknown non-linear function of degree ℓ. Since $e(k)$ is unknown, equation (4) can be expressed in a simplified form as

$$y(k) = F^{\ell}\big(y(k-1),\ldots,y(k-n_y),u(k-1),\ldots,u(k-n_u)\big) \qquad (5)$$

which is known as the NARX (Non-linear AutoRegressive with eXtra inputs) model.

Leontaritis and Billings also claim that the polynomial representation is one of the most common and has been demonstrated to work well in practical applications (Billings and Fadzil, 1985; Billings et al., 1989). Thus, the polynomial NARMAX model is represented as

$$y(k)=\theta_0+\sum_{i_1=1}^{n}\theta_{i_1}x_{i_1}(k)+\sum_{i_1=1}^{n}\sum_{i_2=i_1}^{n}\theta_{i_1}\theta_{i_2}x_{i_1}(k)x_{i_2}(k)$$
$$+\ldots+\sum_{i_1=1}^{n}\ldots\sum_{i_\ell=i_{\ell-1}}^{n}\theta_{i_1}\ldots\theta_{i_\ell}x_{i_1}(k)\ldots x_{i_\ell}(k)+e(k), \qquad (6)$$

where $n=n_y+n_u$ (the sum of the corresponding output and input maximum lags), θ_i are scalar coefficients and $x_i(k)$ represents lagged terms in y and u. Note that the polynomial model is non-linear in the output and input variables but linear in the parameters. Therefore, the set of coefficients can estimated by a Least Squares (LS) algorithm.

5 PROBLEM DEFINITION

Potential model structures of the form of equation (6) can be expressed in a hierarchical tree structure as shown in Figure 6. These models are written in reverse Polish notation as

(ADD (ADD X1 X4) (MULT (ADD X2 X3)(ADD X1 X2)))

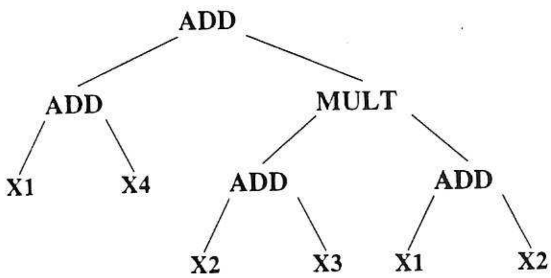


Figure 6. Polynomial NARX model represented as a hierarchical tree.

This is equivalent to the non-linear model defined by

$$y(k)=\theta_0+\theta_1 y(k-1)+\theta_2 y(k-2)+\theta_3 u(k-1)$$
$$+\ldots+\theta_4 y(k-1)^2+\theta_5 y(k-1)y(k-2) \qquad (7)$$

where X1, X2, X3 and X4 are 1.0 (the constant term), y(k-1), y(k-2) and u(k-1), respectively. The associated θ parameters are then estimated by using a least squares algorithm. Because this work restricts itself to the polynomial representation of the NARMAX model, the function set, **F,** is composed of two operations only.

F={ADD, MULT}

The terminal set, **T,** consists of all linear terms in the output, input and noise with maximum lags defined by n_y, n_u and n_e, respectively, and the real value 1.0 representing the constant term. This is defined as

$$T=\{X0, ..., Xn_y, Xn_y+1, ..., Xn_y+n_u, Xn_y+n_u+1, ..., Xn_y+n_u+n_e\}$$

The terminal set is expressed in a look-up table as shown in Table 1.

Table 1. Terminal Set

Element of T	Linear Term
X0	1.0
X1	y(k-1)
...	...
Xn_y	$y(k-n_y)$
Xn_y+1	u(k-1)
...	...
Xn_y+n_u	$u(k-n_u)$
Xn_y+n_u+1	e(k-1)
...	...
$Xn_y+n_u+n_e$	$e(k-n_e)$

Thus, polynomial NARMAX models are represented as programs which manipulate independent variables (terminal set) to generate an output, y(k).

6 MOGP IN PRACTICE

In order to generate programs that represent not only valid models of the system but also parsimonious models, a set of objectives is specified whichaddresses the two main themes of (i) model structure (complexity) and (ii) model performance.

6.1 Example 1: Simple Wiener Process.

Based on these two attributes, the MOGP method described above is demonstrated on the simple Wiener model and compared with two conventional identification techniques, stepwise regression and orthogonal regression (Haber and Unbenhauen, 1990).

The simple Wiener process embodies a linear dynamic part defined by the differential equation,

$$10\dot{v}(t)+v(t)=u(t) \quad \textbf{(8)}$$

and a static non-linear part expressed by

$$y(k)=2+v(k)+v^2(k) \quad \textbf{(9)}$$

The input-output data used here are defined in Haber and Unbehauen (1990), as described in Figure 7.

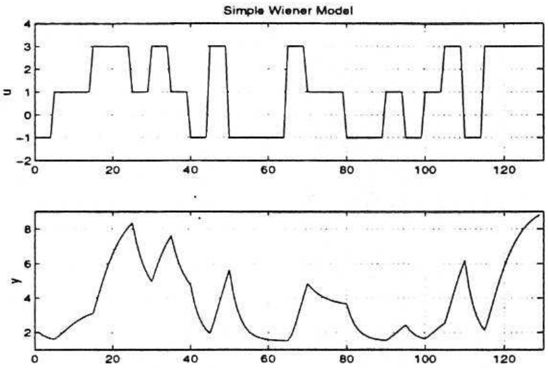


Figure 7. Simple Wiener Process Input-Output Data.

The multiobjective genetic programming (MOGP) approach was run considering five objectives representing the structure and the performance of the models. These were:- the number of terms, *NT*, degree of non-linearity, *DEG*, maximum lag, *LAG*, residual variance, *VAR*, and the long-term prediction error, *LTPE*. Crossover and mutation probabilities were 0.9 and 0.1, respectively. The MOGP method evolved for 100 generations using a population of 200 tree expressions. The method was run several times and produced similar families of solutions each time.

For the purpose of analysis, results of one run are presented in Table 2. In terms of performance (*VAR* and *LTPE*) all models emerging from the MOGP approach dominate those obtained by the stepwise and orthogonal regression methods. For models $MOGP_3$ and $MOGP_4$, an improvement in the VAR criterion is also achieved - that which the stepwise and orthogonal regression methods explicitly targetted.

In terms of model complexity, Table 3 shows the structures of the polynomial NARX models which are similar and have some terms in common.

This identification example is a simple one based upon simulated data, however it serves to illustrate the potential of the MOGP-identification. In the next section MOGP is applied to actual data obtained through a study of an industrial application.

Table 2. Comparative Performance of the Identification Methods.

Model	NT	DEG	LAG	$VAR_{x10^{-3}}$	$LTPE_{x10^{-3}}$
$MOGP_1$	6	2	1	2.3839	6.0221
$MOGP_2$	6	2	2	2.1978	6.7967
$MOGP_3$	7	2	2	1.6484	6.4279
$MOGP_4$	7	2	2	1.6474	7.8151
Stepwise	7	2	2	1.6808	7.8526
Orthogonal	7	2	2	5.2243	26.8080

Table 3. Simple Wiener Model Structures.

Term	$MOGP_1$	$MOGP_2$	$MOGP_3$	$MOGP_4$	Stepwise	Orthogonal
c	▲	▲	▲	▲	▲	▲
y(k-1)	▲	▲	▲	▲	▲	▲
y(k-2)				▲	▲	▲
u(k-1)	▲		▲			▲
u(k-2)		▲				▲
$y(k-1)^2$	▲	▲	▲	▲	▲	
y(k-1)y(k-2)				▲		
$y(k-2)^2$					▲	
y(k-1)u(k-1)	▲	▲		▲	▲	▲
$u(k-1)^2$	▲	▲	▲	▲	▲	▲
y(k-2)u(k-1)			▲			
u(k-1)u(k-2)			▲			

6.2 Example 2: Gas Turbine Engine Modelling

Here, MOGP is applied to sets of actual input-output observation data from a gas turbine engine system. This case study utilises real engine data records, from a Rolls-Royce Spey engine, collected by the Defence Evaluation and Research Agency, Pyestock, U.K.

Most previous research on system identification of aircraft engines has been based on *linear* frequency and time domain identification and has not been directly applicable to *non-linear* modelling. Some approaches which deal with non-linear modelling are based upon the identification of local linear models. The basis of these approaches is a decomposition of the system's full range of operation into a number of possible overlapping operating regions. In each operating region, a simple local model is applied. These local models are then combined in some way to yield a global model. However, an important question arises: how many local models are required to cover the operating range of the system? It is clear, therefore, that there is a compromise to be made between the number and size of local models and their complexity.

The identification approach presented in this paper deals with the identification of *non-linear* models that can cope with the entire range of working conditions. This global model is then used to model the relationship between the fuel flow consumption and the spool speed of a gas turbine engine. The identification is based upon a set of multisine excitation signals (Evans *et al.*, 1992) at different operating conditions, applied to the engine system which is described in the next section.

6.2.1. Description of the engine system

A schematic of the measurement system is shown in Figure 8. The reheat system is inoperative during the tests and the compressor bleed valve is closed. The angle of the inlet guide vanes and the reheat nozzle area are fixed at their low speed positions for the duration of the tests. The engine speed control is operated in open-loop and a perturbed fuel demand signal fed to the fuel feed system, which regulates the fuel flow to the engine by means of a stepper valve.

The fuel feed system exhibits both linear and non-linear dynamics, which affect the actual fuel flow applied to the gas turbine. It is important to eliminate these effects from the estimated engine models and this is achieved by measuring the actual fuel flow downstream of the fuel feed, using a turbine fuel meter. The speed of the low pressure (LP) shaft is measured by counting the rotations of the turbine blades and the speed of the high pressure (HP) shaft by measuring the rotation of a gear linked to the shaft itself.

The shaft speeds are the primary outputs of a gas turbine, from which the internal engine pressures and thrust can be calculated. The dynamic relationship between these shaft speeds and the measured fuel flow is the purpose of this model identification example.

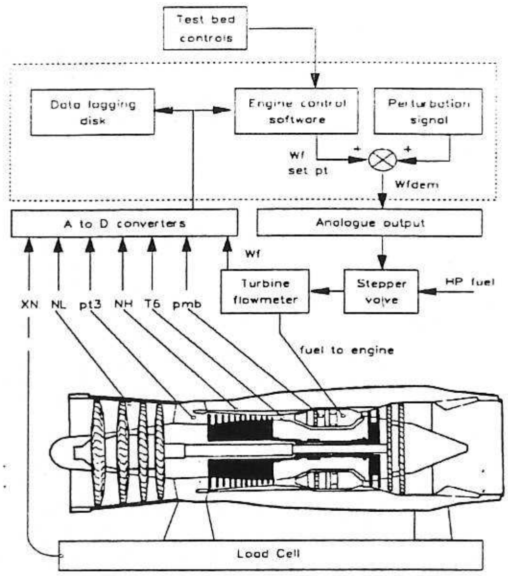


Figure 8. Schematic of the measure tests conducted on a Rolls-Royce Spey engine.

6.2.2. Result Analysis

Using the MOGP identification approach we are able to include a set of objectives which assess the performance of a candidate model at individual points over the operating range of the engine. A series of MOGP approaches is detailed below.

The initial approach did not take into account the validation criteria, dealing only with model complexity and performance (as in Example 1). These were considered as "soft" objectives (that is, not "hard" constraints) and had the same level of priority. With this approach only linear models emerged to model the relationship between fuel flow demand and the HP shaft speed. These results are graphically represented in Figure 9, where the eleven objectives are ranged along the *x-axis* and the performance achieved for each objective is indicated in the *y direction*. In this "parallel co-ordinates representation", each line in the graph represents a potential solution to the design problem, indicating the achieved objective values for that solution. All solutions are both non-dominant **and** satisfy the prescribed *goals* as represented by the "×" marks. Five models emerged which exhibited similar performance in terms of residual and long term prediction error measures. From Figure 9, the number of models (indicated by individual lines) that satisfy all the objectives belong to the ARX model (linear) expressed by

$$y(k) = \sum_{i=1}^{n_y} a_i y(k-i) + \sum_{i=1}^{n_u} b_i u(k-i) + c \qquad (10)$$

where $n_y = n_u = 2$ (maximum lags), and c is a constant value. The structure of each model is specified in Table 4 (Recall that while the MOGP process obtained only linear models in the non-dominant set, the formulation permitted non-linear model descriptions).

Table 4: Fuel flow-HP shaft speed linear model structures.

Model	a_1	a_2	b_1	b_2	c
1	▲		▲		▲
2		▲	▲	▲	▲
3	▲		▲	▲	▲
4	▲	▲	▲		▲
5	▲	▲	▲	▲	▲

Linear Validation

Since linear engine models were obtained from the previous experiment, the first step in model validation is to test the associated correlation functions. Therefore, in a second approach to the identification of this system, a validation stage was included, based on the evaluation of the autocorrelation of the residuals (*ACF*), the crosscorrelation between the residuals and input (*CCF*). This functions are given as

$$\Phi_{\varepsilon\varepsilon}(\tau) = E[\varepsilon(k-\tau)\varepsilon(k)] = \delta(\tau)$$

$$\Phi_{u\varepsilon}(\tau) = E[u(k-\tau)\varepsilon(k)] = 0 \quad \forall\tau \qquad (11)$$

The correlation objective functions were cast as constraints. The target value to be attained was given by the 95% confidence limit. Scalar measures of correlation were selected to be

$$CCF = \max\{|\Phi_{u\varepsilon}(\tau)|\},$$
$$ACF = \max\{|\Phi_{\varepsilon\varepsilon}(\tau)|\}, \qquad (12)$$

where $\Phi_{\varepsilon\varepsilon}(\tau)$ is set equal to 0, for $\tau=0$, otherwise ACF would always be δ.

The identification then sought valid models in the space of all possibilities. In this experiment, this method provided a set of non-linear, instead of linear, models which satisfy the validation requirements. Note that in Figure 10 there are two linear models but these do not meet the constraints related to the validation criteria, Objectives 12 and 13. The optimisation produced these linear models because objective 2 (see Table 5) was assigned the same level of priority as the correlation functions. Note that models with a non-linear degree of two and three possess a few more terms than the linear models generated in the previous approach (see Objective 1).

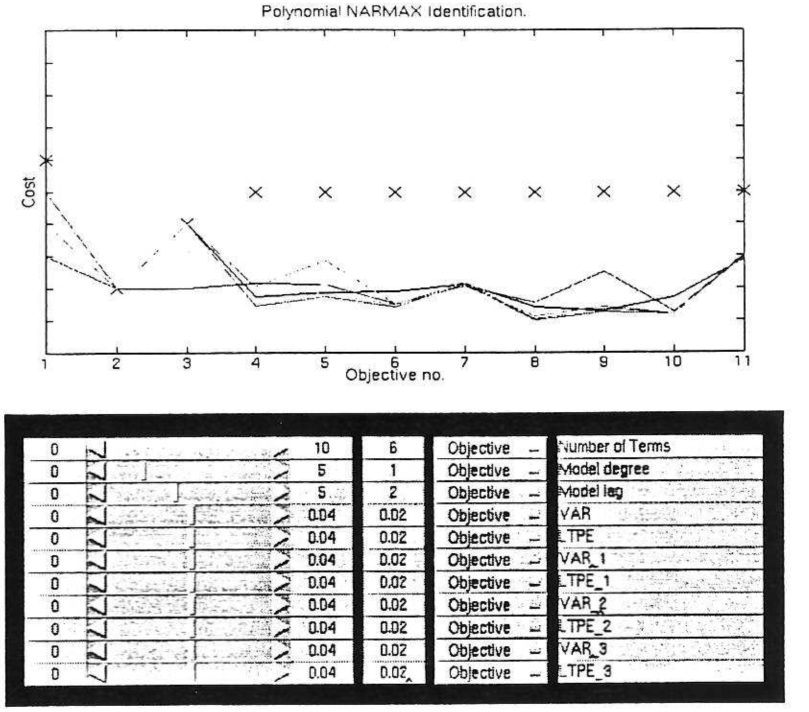


Figure 9: Multiobjective genetic programming framework (multisine input signals).

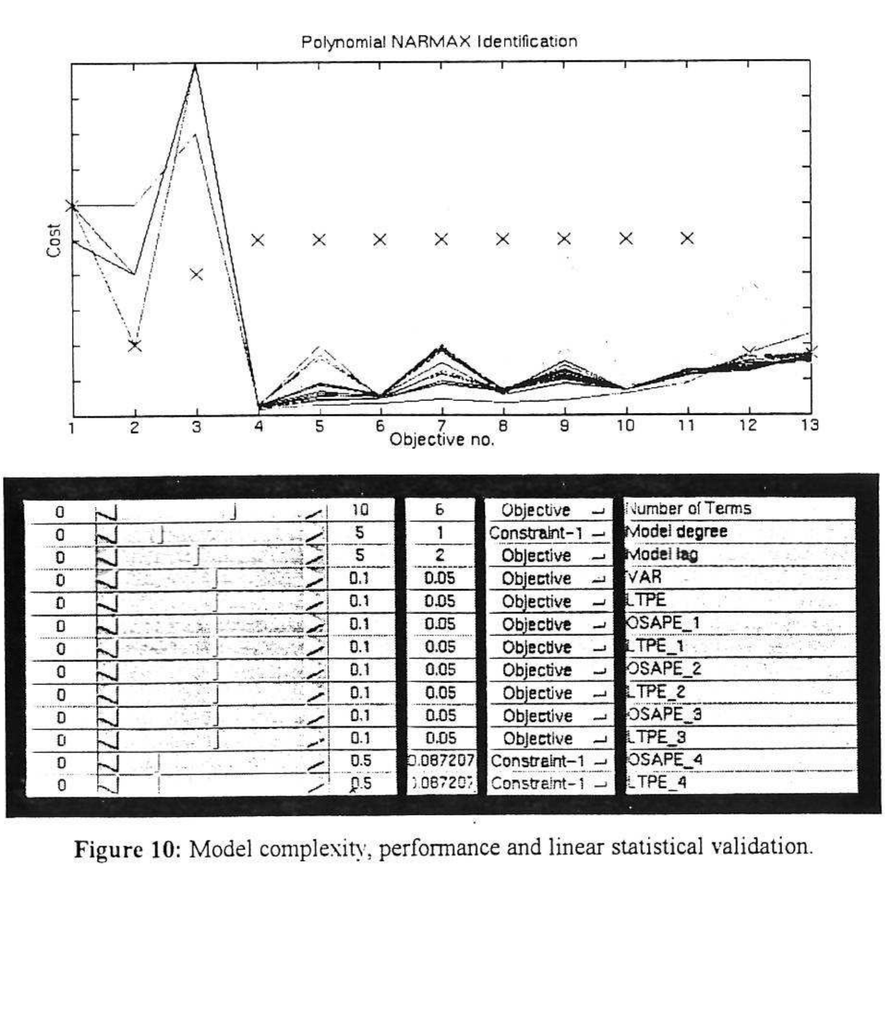


Figure 10: Model complexity, performance and linear statistical validation.

Validation of Non-linear Engine Models

Although the ACF and CCF constraints were satisfied, these two conditions are not sufficient in order to provide unbiased non-linear models. For the purpose of non-linear system validation, high-order correlation tests (Billings and Voon, 1986) must also be evaluated and these are described by

$$\Phi_{\varepsilon(\varepsilon u)}(\tau) = E\left[\varepsilon(k)\varepsilon(k-\tau)u(k-\tau)\right] = 0 \quad \tau \geq 0$$
$$\Phi_{u^2{}'\varepsilon}(\tau) = E\left[\left\{u^2(k) - E\left\{u^2(k)\right\}\right\}\varepsilon(k-\tau)\right] = 0 \quad \forall\tau$$
$$\Phi_{u^2{}'\varepsilon^2}(\tau) = E\left[\left\{u^2(k) - E\left\{u^2(k)\right\}\right\}\varepsilon^2(k-\tau)\right] = 0 \quad \forall\tau \tag{13}$$

The objectives considered for the next application of MOGP now relate to model structure, performance and *non-linear* validation aspects (as summarised in Table 5). This objective function vector is defined as

$$F = \left[NT, DEG, LAG, VAR_i, LTPE_i, ACF, CCF, HOC_j\right] \tag{14}$$

where NT is the number of model terms, DEG is the degree of non-linearity, LAG is the maximum model lag, VAR_i and $LTPE_i$ correspond to the residuals variance and long-term prediction error, respectively, and i identifies the test signal used. The validation stage was based on the evaluation of the autocorrelation of the residuals (ACF), the crosscorrelation between the residuals and input (CCF), and higher order correlation functions (HOC_j). The HOC_j functions, as defined in equation (13), are used to determine whether the correct non-linear terms are detected in the model

Table 5. Description of the objectives considered in the MOGP-identification procedure.

Attribute	Objective	Description
Model complexity	1. Model size	Number of process and noise terms
	2. Model degree	Maximum order term
	3. Model lag	Maximum lagged input, output and noise terms
Model Performance	4. Residual variance	Variance of the predictive error between the OSAPE and the measured outputs
	5. Long-term prediction error	Variance of the LTPE
Model validation	6. ACF 7. CCF 8-10 HOC, as defined in equation (13)	Autocorrelation, crosscorrelation and higher-order correlation based functions

By evaluating the higher-order correlation tests, the number of non-dominated (or preferred) models tended to be more selective because of the number of restrictions (constraints) that had to be satisfied. The final set of valid non-linear models generated by means of this new approach is described in Table 6. As in the previous approach, objective two (model degree) and the higher-order correlation functions were considered as "hard" objectives, i.e. constraints.

As can be observed from these results, even though the terminal set $\mathcal{T}$ in the GP method included past values of the residuals ($\{e(k-1), \ldots, e(k-n_e)\}$), no NARMAX structures arose. Instead, only NARX model structures emerged.

Validation Using Global Simulation

In this section, the previous set of quadratic models is validated on different data sets from the ones used for identification. Ramp testing signals were used to exercise the nonlinear models over a wide operating range. Figures 11 to 13 show the measured and the long-term predictive outputs of models 1 to 3, respectively, over a triangular wave of period 100s with amplitude such as to cause variations of the NH from 65% to 85%.

Table 6: Quadratic Polynomial Models.

Term\Model	1	2	3
c	▲	▲	▲
y(k-1)	▲	▲	▲
y(k-2)	▲	▲	▲
u(k-1)			▲
u(k-2)			▲
$y(k-1)^2$			▲
$y(k-2)^2$	▲		
y(k-1)u(k-1)	▲	▲	
y(k-2)u(k-2)	▲	▲	

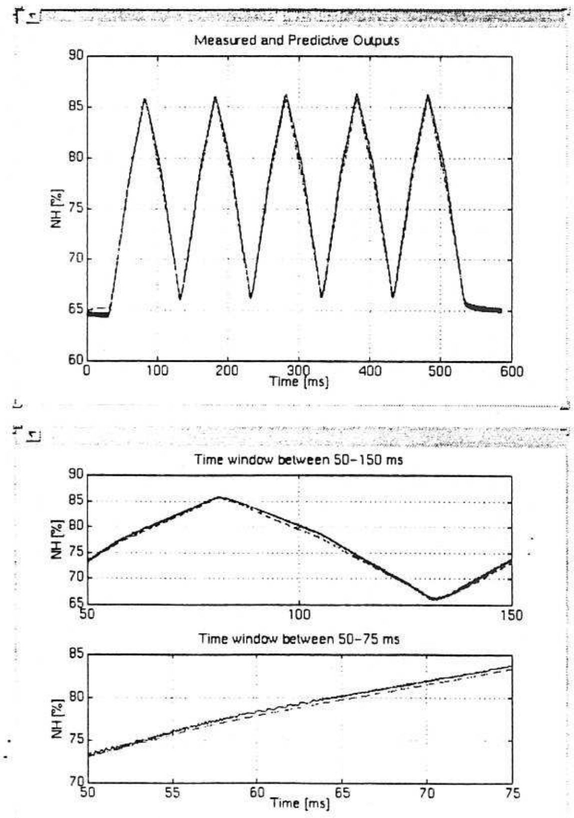


Figure 11: Predictive output of a triangular wave using model 1. Solid line (measured), dashed line (predictive).

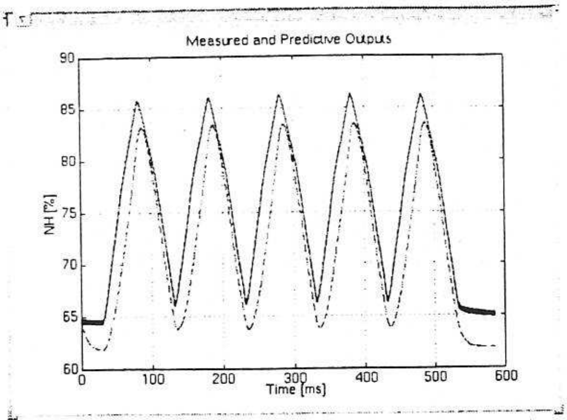


Figure 12: Predictive output of a triangular wave using model 2.Solid line (measured), dashed line (predictive).

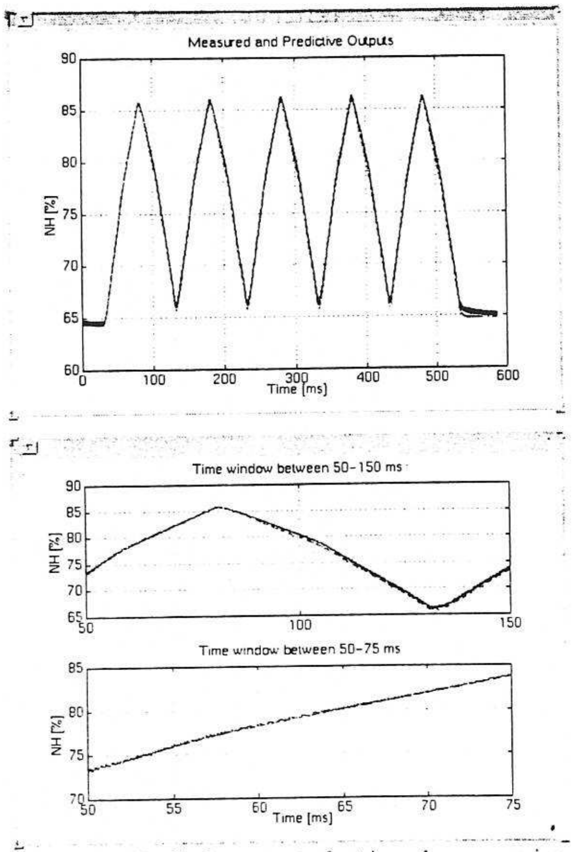


Figure 13: Predictive output of a triangular wave using model 3. Solid line (measured), dashed line (predictive).

As illustrated in Figs 11-13, models 1 and 3 exhibit a better response than model 2. For the three-level periodic sequence (Fig. 14), again model 2 gives an inferior predictive response. Model 3 predicts better than model 1 which shows a small offset of NH.

Based upon the non-linear structure of model 3, three-level (Figure14) and large transient (Figure 15) data were additionally used for modelling the engine relationship between the measured fuel flow and the high pressure shaft speed. As a result of this study, this model is selected as the most appropriate; the structure and parameters of this quadratic NARX engine model are

$$\begin{aligned} y(k) = & -1.16236x10^{0} + 7.96090x10^{-1}\, y(k-1) \\ & + 2.39363x10^{-1}\, y(k-2) + 4.33425x10^{-3}\, u(k-1) \\ & - 5.90401x10^{-4}\, u(k-2) - 4.37337x10^{-4}\, y(k-1)^{2} \end{aligned} \quad \textbf{(15)}$$

7. CONCLUSIONS

An evolutionary algorithm based upon the NARMAXrepresentation has been introduced as an alternative approach for non-linear system identification problems. Genetic programming has proved to be a powerful tool for formulating and solving complex system identification problems, in particular for determining system structure.

Additionally, the incorporation of a multiobjective approach has enabled the separate consideration of different objectives related to model complexity and model performance. The validation process can be also included in this multiobjective framework.

In comparison with *conventional* identification techniques (as used in Example 1), multiobjective evolutionary identification methods provide a family of candidate models which satisfy diverse objectives. This enables the modeller to select the appropriate model depending on specific circumstances.

In the search for a valid engine model (Example 2), the multiobjective GP identification method has been demonstrated to perform effectively. The set-up of statistical validation criteria as "hard" constraints instead of being considered as "soft" objectives, directed the search process to reveal valid models.

Note that this differs from an operating regime approach (Johansen and Murray-Smith, 1996) which approximates the system by *local linear* models; instead, it has the advantage of producing a simple and accurate *non-linear global* model for this test case involving an industrial application.

Fitness function evaluation is time consuming when using this approach, in view of the multiple objectives involved. Compared with "conventional" methods, more time is spent running the identification method. While this is a shortcoming of this approach. it is offset by the ability of this evolutionary-based identification method to produce valid non-linear models which confer a high level of credibility on the model. The ability to cope with different situations, as shown by the different test responses for the gas turbine engine example, demonstrates the flexibility of the approach.

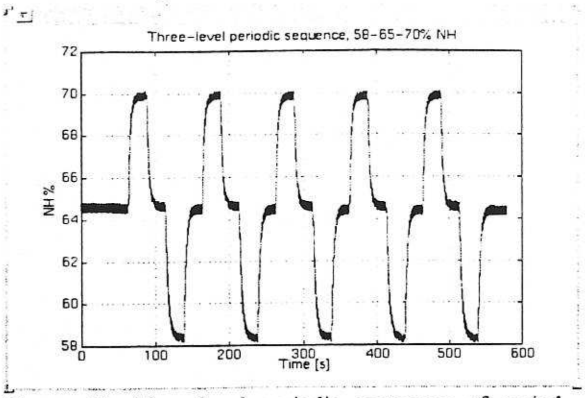


Figure 14: Three-level periodic sequences of period 100s, each period consisting of quarter-periods at levels 65%, 58%, 65% and 70% of the NH shaft speed.

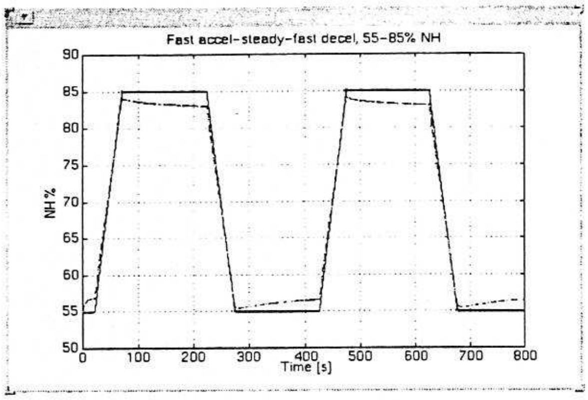


Figure 15: Fast acceleration/deceleration signal (55% to 85% NH).

ACKNOWLEDGEMENTS

Katya Rodríguez Vázquez gratefully acknowledges the financial support of Consejo Nacional de Ciencia y Tecnología (CONACyT). We also wish to acknowledge the support of the Engineering and Physical Sciences Research Council (UK) through the grant GR/K31343. Rolls-Royce plc (especially Dr Dave Hill) and DERA (Pyestock).

REFERENCES

BEN-TAL, A. (1980). Characterization of Pareto and Lexicographic Optimal Solutions. *Multiple Criteria Decision Making. Theory and Applications*, **Vol. 77** of Lecture Notes in Economics and Mathematical Systems. (Fandel and Gal, eds.). 1-11.

BILLINGS, S.A. AND FADZIL, M.B. (1985). The Practical Identification of Systems with Non-linearities. *7th IFAC Symposium on Identification and System Parameter Estimation*. 155-160.

BILLINGS, S.A., S. CHEN AND R.J. BACKHOUSE. (1989). Identification of Linear and Non-linear Models of a Turbochraged Automotive Diesel Engine. *Mechanical Systems and Signal Processing*, 3(2). 123-142.

EVANS, D.C., D. REES AND D.L. JONES (1992) Design of Tests Signals for Identification of Linear Systems with Non-Linear Distortions. *IEEE Trans. on Instrumentation and Measurement*. **41**(6). pp. 768-774.

FOGEL, L.J., A.J. OWENS AND M.J. WALSH. (1966). *Artificial Intelligence Through Simulated Evolution*. New York: John Wiley.

FONSECA, C.M. AND P.J. FLEMING. (1993). Genetic Algorithms for Multi-Objective Optimization: Formulation, Discussion and Generalization. *Proc. of the Fifth Int. Conf. on Genetic Algorithms*, (Forrest. Ed.), 416-423.

FONSECA, C.M. AND P.J. FLEMING. (1995). An Overview of Evolutionary Algorithms in Multiobjective Optimization, *Evolutionary Computation* **3(1)**, 1-16.

GOLDBERG, D.A. (1989). *Genetic Algorithms in Search, Optimisation and Machine Learning*, Addison-Wesley.

HABER, R. AND H. UNBENHAUEN (1990) Structure Identification of Non-Linear Dynamic Systems: A Survey of Input/Output Approaches. *Automatica*, 26 (4), pp. 651-677.

HOLLAND, J.H. (1975). *Adaptation in Natural and Artificial Systems*. Ann Arbor: The University of Michigan Press.

IBA, H., H. DE GARIS AND T. SATO. (1994). Genetic Programming Using a Minimum Desciption Length Principle, *Advances in Ge... tic Programming*, (Kinnear, Ed.), 265-284.

JOHANSEN, T.A. AND R. MURRAY-SMITH (1996) The Operating Regime Approach to Non-Linear Modelling and Control. In *Multiple Model Approaches to Modelling and Control* (R. Murray-Smith, ed.) Taylor and Francis, pp. 3-71.

KOZA, J.R. (1992). *Genetic Programming: On the Programming of Computers by Means of Natural Selection*, MIT Press.

KOZA, J.R. (1994). *Genetic Programming II: Automatic Discovery of Reusable Programs*. MIT Press.

LEONTARIS,I.J. AND S.A. BILLINGS. (1985). Input-Output Parametric Models for Non-Linear Systems. *Int. J. Control* **41(2)**, 311-341.

LANGDON, W.B. (1995). Evolving Data Structures with Genetic Programming. *Proc. of the 6th Int. Conference on Genetic Algorithms*, Forrest (Ed.). pp. 1-8.

LJUNG, L.S. (1987). *System Identification: Theory for the User*. Prentice Hall.

RECHENBERG, I. (1973). *Evolutionsstrategie: Optimierung Technischer Systeme nach Prinzipien der Biologischen Evolution*. Sttutgart: Frommann-Holzboog.

RODRÍGUEZ-VÁZQUEZ, K., C.M. FONSECA AND P.J. FLEMING. (1997a). An Evolutionary Approach to Non-Linear System Identification, *11th IFAC Symposium on System Identification*, **3**, 1395-1400.

RODRÍGUEZ-VÁZQUEZ, K., C.M. FONSECA AND P.J. FLEMING. (1997b). Multiobjective Genetic Programming: A Non-linear System Identification Application. *Late Breaking Papers at the Genetic Programming 97 Conference*. 207-212.

ROSENBERG, R.S. (1967). *Simulation of Genetic Populations with Biochemical Properties*. PhD Thesis, University of Michigan.

SCHWEFEL, H.P. (1965). *Kybernetische Evolution als Strategie der Experimentellen Forschung in der Strömungstechnik*. Diploma Thesis, Technical University of Berlin.

SHARMAN, K.C., A.I. ESPARCIA-ALCÁZAR AND Y.LI (1995). Evolving Signal Processing Algorithms by Genetic Programming, *Proc. of the IEE/IEEE GALESIA*, 473-480.

SIEGEL, E.V. AND A.D. CHAFFE. (1996). Genetically Optimizing the Speed of Programs Evolved to Play Tetris, *Advances in Genetic Programming 2*, (Angeline and Kinnear, Eds.), 279-298.

SÖDERSTRÖM, T. AND P. STOICA. (1989). *System Identification*. Prentice Hall.

ZHANG, B.T. AND H. MÜHLENBEIN. (1996). Adaptive Fitness Functions for Dynamic Growing/Pruning, *Advances in Genetic Programming 2*, (Angeline and Kinnear, Eds.), 241-256.