

Multiobjective Gas Turbine Engine Controller Design Using Genetic Algorithms

Andrew Chipperfield and Peter Fleming

Abstract—This paper describes the use of multiobjective genetic algorithms (MOGA's) in the design of a multivariable control system for a gas turbine engine. The mechanisms employed to facilitate multiobjective search with the genetic algorithm are described with the aid of an example. It is shown that the MOGA confers a number of advantages over conventional multiobjective optimization methods by evolving a family of Pareto-optimal solutions rather than a single solution estimate. This allows the engineer to examine the trade-offs between the different design objectives and configurations during the course of an optimization. In addition, the paper demonstrates how the genetic algorithm can be used to search in both controller structure and parameter space thereby offering a potentially more general approach to optimization in controller design than traditional numerical methods. While the example in the paper deals with control system design, the approach described can be expected to be applicable to more general problems in the fields of computer aided design (CAD) and computer aided engineering (CAE).

I. INTRODUCTION

MODERN gas turbine engines are highly complex systems that require equally complex controllers in order to remain stable while satisfying the demands placed on them by the pilot and operating conditions. Because of these complexities, extensive use is made of computer aided control system design (CACSD) methods to design controllers to meet the desired performance specifications. In particular, optimization-based methods in CACSD have been shown to be a valuable tool in assisting the control engineer in selecting suitable controller parameters [1].

Many engineering problems are, however, characterized by a number of noncommensurate design objectives that must all be obtained for the solution to be satisfactory. Due to the nature of trade-offs involved, particularly in cases where multidisciplinary design criteria are specified, e.g., size, cost, and performance, such problems seldom have a unique solution. Instead, a set of equally admissible, or nondominated solutions, are sought from an appropriately formulated optimization problem.

However, parametric optimization methods are numerically intensive and require repeated application to identify the trade-offs between different design objectives. In this paper, we consider the application of genetic algorithms (GA's) [2] to the design of gas turbine engine control systems. After a brief introduction to multiobjective optimization, the concept

of a multiobjective genetic algorithm (MOGA) is introduced through a design example of a control system for a short take-off vertical landing aeroengine. It is shown that the MOGA confers an immediate advantage over conventional multiobjective optimization methods by evolving a family of Pareto-optimal solutions. Thus, the relative trade-offs between design objectives may be easily identified and a more informed choice made for the final controller structure.

II. MULTIOBJECTIVE OPTIMIZATION

The use of multiobjective optimization (MO) recognizes that most practical problems require a number of design criteria to be satisfied simultaneously, viz:

$$\min_{x \in \Omega} F(x)$$

where $x = [x_1, x_2, \dots, x_n]$ and Ω define the set of free variables, x , subject to any constraints and $F(x) = [f_1(x), f_2(x), \dots, f_n(x)]$ are the design objectives to be minimized.

Clearly, for this set of functions, $F(x)$, it can be seen that there is no one ideal "optimal" solution, rather a set of Pareto-optimal solutions for which an improvement in one of the design objectives will lead to a degradation in one or more of the remaining objectives. Such solutions are also known as noninferior or nondominated solutions to the multiobjective optimization problem.

Conventionally, members of the Pareto-optimal solution set are sought through solution of an appropriately formulated nonlinear programming problem. A number of approaches are currently employed including the ϵ -constraint, weighted-sum, and goal-attainment methods [3]. However, such approaches require precise expression of a usually not-well-understood set of weights and goals. If the trade-off surface between the design objectives is to be better understood, repeated application of such methods will be necessary. In addition, nonlinear programming methods cannot handle multimodality and discontinuities in function space well and can thus only be expected to produce local solutions.

Genetic algorithms, on the other hand, do not require derivative information or a formal initial estimate of the solution region. Because of the stochastic nature of the search mechanism, GA's are capable of searching the entire solution space with more likelihood of finding the global optimum than conventional optimization methods. Indeed, conventional methods usually require the objective function to be well behaved, whereas the generational nature of GA's can tolerate noisy, discontinuous, and time-varying function evaluations

Manuscript received September 14, 1995; revised November 22, 1995. This work was supported by the U.K. EPSRC under Grant GR/K 36591.

The authors are with the Department of Automatic Control and Systems Engineering, The University of Sheffield, Sheffield, S1 3JD, U.K.

Publisher Item Identifier S 0278-0046(96)03302-3.

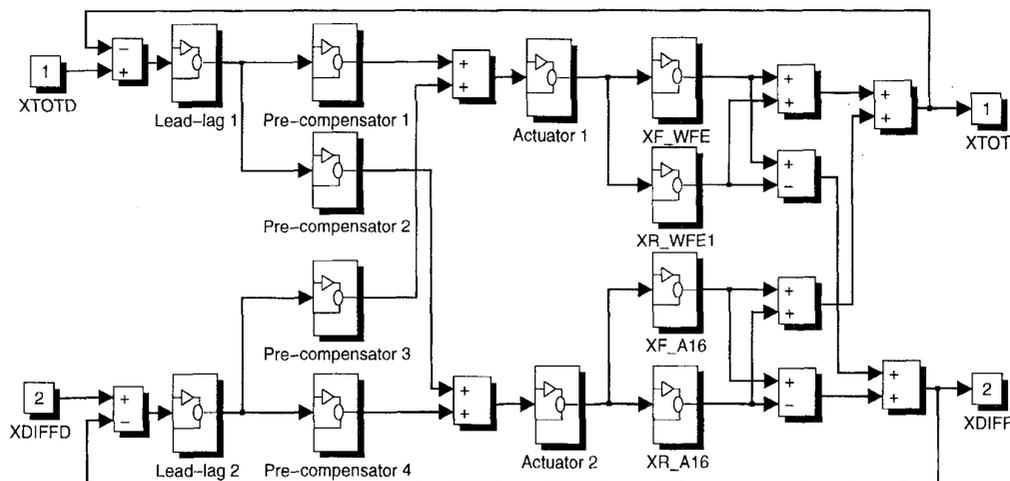


Fig. 1. SIMULINK model of the ASTOVL engine.

[4]. Single objective GA's do still require some combination of the design objectives although the relative importance may be changed during the course of the search process.

Multiojective GA's [5], employing the principles of Pareto dominance as the basis for assigning a scalar fitness value to a vector valued objective function, allow individual solution estimates to be compared with one another even in the total absence of information regarding the relative importance of the objectives. Using rank-based selection and niching techniques, it is feasible to generate populations of non-dominated solution estimates without having to combine the objectives in some way. This is advantageous because the combination of noncommensurate objectives requires precise understanding of the interplay between those objectives if the optimization is to be meaningful. The use of Pareto rank-based fitness assignment permits different nondominated individuals to be sampled at the same rate thereby according the same preference to all Pareto-optimal solutions.

Because MOGA's are susceptible to unstable converged populations, due to the potential for very different genotypes to result in nondominated individuals, a particular problem is the production of lethals when fit members of the population are mated. The search then becomes inefficient and the GA is likely to converge to some suboptimal solution. However, the use of mating restrictions, to reduce the production of lethals, enhances the stability of the population whilst allowing a wide diversity in genetic material.

III. DESIGN EXAMPLE

This example application demonstrates how MOGA's may be used to select the controller structure and suitable parameter sets for a multivariable flight control system. The system considered is a propulsion unit for an advanced short take-off, vertical landing (ASTOVL) aeroengine [6], shown in Fig. 1. There are two inputs to the system, XTOTD and XDIFFD, and it is required that the pilot have control of the fore-aft differential thrust (XDIFF) and the total engine thrust (XTOT). The design problem is to find a set of precompensators that

satisfy a number of time-response design specifications while minimizing the interaction between the loops of the system.

The time-domain performance requirements, in response to a step in demand at one of the inputs, are

- 1) 70% rise-time ≤ 0.35 seconds,
- 2) 10% settling-time ≤ 0.5 s, and
- 3) maximum overshoot $\leq 10\%$,

at the associated output. The amount of interaction, or cross-coupling, between modes is measured as

$$\int_0^{\infty} (XTOT)^2 dt$$

when excited by a step input to XDIFFD, and vice-versa, and should be less than 0.05 for this example.

IV. IMPLEMENTATION

The ASTOVL propulsion unit was modeled directly using the SIMULINK package [7] as shown in Fig. 1. The objective functions were written as *m*-files using commands from the Control Systems Toolbox. The precompensators for this problem were allowed to be either first or second order or simple gains, Fig. 2(a).

Using a structured chromosome representation [4], Fig. 2(b), it is possible to allow the free parameters for each possible precompensator configuration to reside in all individuals. Here, high-level genes, labeled P_1 to P_4 in Fig. 2(b) and encoded as integers, are used to determine which precompensator structures are active in a particular chromosome. Associated with each precompensator, P_i , are three sets of real-valued parameters, A_i , B_i , and C_i , corresponding to the gains and time constants of the permissible precompensator structures. Thus, the values of the precompensator structure flags select which set of parameters are valid with each precompensator and therefore the order of the precompensators. In this way, a chromosome may contain a number of possibly good representations at any one time, although only the set defined by the values of the high-level genes will be active.

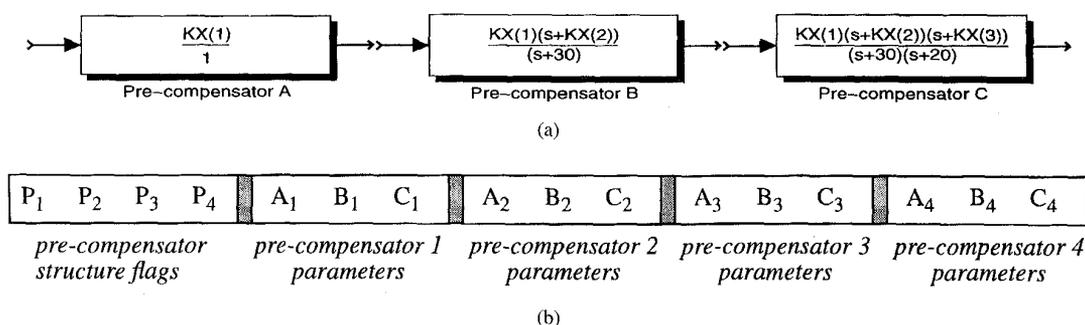


Fig. 2. Representing the precompensator structures and parameter sets. (a) Precompensator structures. (b) Chromosome structure.

In addition, as the overall structure of the controller varies with the set of active pre-compensators, an additional objective was included that measured the complexity of the controller. This was calculated by summing the values of the precompensator flags, thus

$$\sum_{i=1}^4 P_i \leq 9.$$

Thus, a total chromosome length of 28 elements was used and nine design objectives should be satisfied.

A Genetic Algorithm Toolbox for use with the MATLAB CACSD package [8], developed in house, was used to implement the GA with additional extensions to accommodate multiobjective ranking, sharing, and mating restrictions in the objective domain [5].

Multiobjective ranking is based upon the dominance of an individual, i.e., how many individuals outperform it in objective space, Fig. 3. From Fig. 3, it may also be seen that multiobjective ranking is nonunique, a number of individuals are ranked 0, i.e., nondominated. Ranking may also be combined with goal and/or priority information to discriminate between nondominated solutions. For example, a solution in which all the goals are satisfied can be considered superior, or preferable, to a nondominated one in which some components go beyond the goal boundaries. Similarly, an improvement in an objective component such that the objective value lies inside the goal boundary should be accepted even when this causes a degradation in another objective component provided that the corresponding goal is satisfied. In cases where objectives are assigned different priorities, higher priority objectives are optimized in a Pareto fashion until their goals are met at which point the remaining objectives are optimized (see [9]). In this example, the goals were set to the values given in the previous section and all objectives were assigned the same priority.

All preferred individuals thus achieve the same fitness; however, the number of actual offspring may differ due to the stochastic nature of the selection mechanism. Thus, an accumulation of the imbalances in reproduction can lead the search into an arbitrary area of the trade-off surface. This phenomenon is known as genetic drift [10] and can drastically reduce the quality and efficiency of search. Proposed as a solution to the problem genetic drift, fitness sharing [11] penalizes the fitness of individuals in popular neighborhoods in favor of more remote individuals of similar fitness. A niche

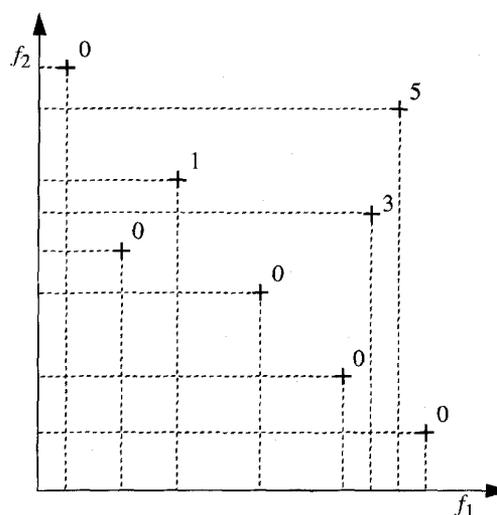


Fig. 3. Pareto ranking.

count is used to derive the density of individuals in genotypic, phenotypic, or objective space. This value is derived from a sharing function that determines the contribution made by individuals to niches. Raw fitness values may then be adjusted in response to niche counts and the total fitness in the population redistributed over the individuals. However, the use of fitness sharing has been restricted by difficulties in determining appropriate niche sizes.

The approach adopted in this example overcomes this problem by using techniques developed from kernel density estimation methods in statistical analysis, in particular the Epanechnikov kernel [12]. A constant factor is used in computing the sharing function and has been shown in [12] to be appropriate for performing sharing in Euclidean decision variable space.

Recombining arbitrary pairs of nondominated individuals from the trade-off surface can result in the production of an unacceptably large number of unfit offspring, or lethals. A further refinement to the MOGA is therefore to bias the manner in which individuals are paired for recombination, often termed mating restriction [13]. This is generally achieved by restricting reproduction to pairs of individuals that are within some given distance of each other. A common practice is to use the niche radius from the sharing function and the approach here employed the smoothing parameter from the

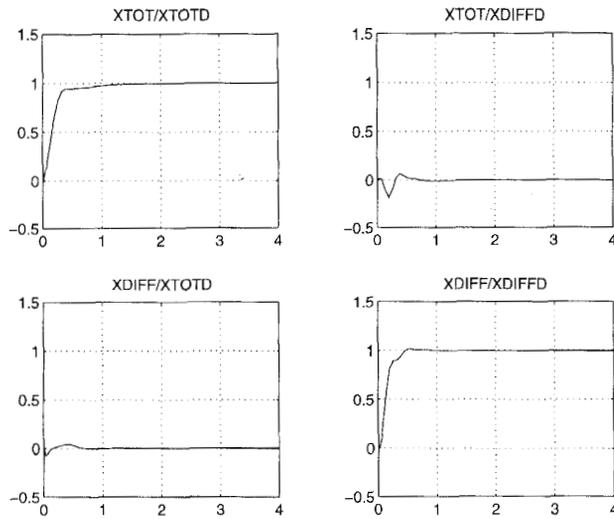


Fig. 4. Typical optimized ASTOVL response.

kernel density function as the mating restriction limit. The interested reader is referred to [14] where a full account of the development of MOGA's may be found.

The crossover operator employed was intermediate recombination [15] applied with probability 0.7. As the chromosome contains many inactive elements, the probability of applying breeder GA mutation [15] was set to 0.1. The use of adaptive mutation rates may have been more appropriate for this example and representation, although the (seemingly) high mutation rate is consistent with the use of real-valued operators and the average number of active parameters. No fine-tuning of operator rates was attempted.

Finally, in order to reduce the computational burden of evaluating the objectives, re-evaluation of individuals was only performed if they had been affected by the genetic operators [16]. This reduced the number function evaluations required by 20–30%.

V. RESULTS

Using a population size of 40, the GA was run for 100 generations in the first instance. A list of the best 50 individuals was continually maintained during the execution of the GA allowing the final selection of controller to be made from the best structures found by the GA over all generations.

Fig. 4 shows a typical response for a controller found by the MOGA. It can be clearly seen that all of the design objectives have been satisfied. However, from such responses it is difficult to determine the relative merits of one controller against another over the entire population. This is particularly true if on-line preference articulation is to be used to guide the search during an optimization.

Fig. 5 illustrates a typical trade-off graph for the ASTOVL controller and a user interface for interactively setting design goals and examining nondominated solutions in a population or database. In the plot, each line represents a nondominated individual found by the MOGA. Objectives 1, 2, and 3 are the rise-time, settling-time, and overshoot, respectively, for the XTOT channel, and objective 4 is the cross-coupling between

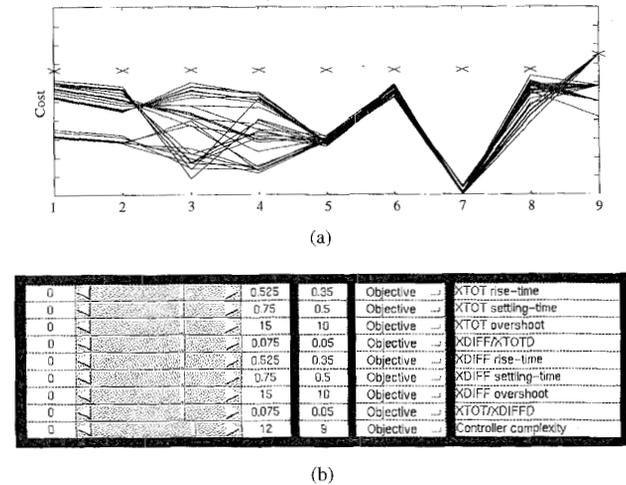


Fig. 5. Sample ASTOVL. (a) Trade-off graph. (b) Design objectives.

XTOTD and XDIFF. Objectives 5 to 8 correspond to the same design specifications on the XDIFF channel and objective 9 is the overall controller complexity. The y -axis shows the performance of individuals in each objective domain with cross-marks showing the design goals.

In Fig. 5, trade-offs between adjacent design objectives result in the crossing of lines between them whereas noncrossing lines indicate that objectives do not compete with one another. For example, the XTOT channel settling-time, overshoot, and cross-coupling (objectives 2, 3, and 4) appear to compete quite heavily, while the same trade-offs are not exhibited by the XDIFF channel. Only the preferred individuals, those that satisfy the design goals, are shown. When no individuals are preferred, the nondominated individuals are displayed. An additional feature of the user interface is the ability to move the position of the objectives on the x -axis. This affords the control engineer a convenient mechanism for examining the trade-offs between nonadjacent design objectives.

Having satisfied the original design goals, the control engineer is now free to enhance the performance of the controllers. The relative degree of under- or over-attainment of the design goals is clearly visible in Fig. 5 and the designer may take advantage of this information when setting new design goals.

Fig. 6 shows the new trade-off graph produced when the goals are reset and the MOGA is allowed to continue for a further 25 generations. The cross-marks on the plot correspond to the new goal set of [0.25, 0.4, 5.0, 0.03, 0.25, 0.4, 5.0, 0.03, 12]. In this goal set, all of the performance goals have been tightened while the controller complexity has been relaxed. This allows more complex controllers to be considered in order to meet the stricter performance requirements. However, as many satisfactory structures already exist in this region, the most complex controllers have not had their parameters tuned sufficiently to meet these new design requirements.

By changing the values of the goals, the search is forced to examine a smaller area of the trade-off surface. Individuals that do not now satisfy the design goals are no longer preferred and the population is forced to evolve toward a new region of the design space. Thus, a more accurate picture of the trade-off surface in that region is constructed.

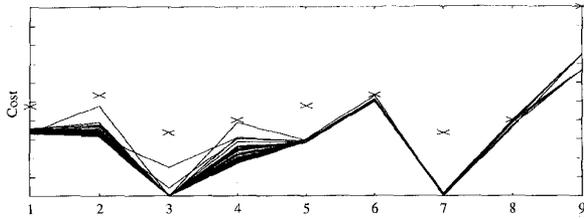


Fig. 6. Trade-off graph for revised design objectives.

VI. CONCLUDING REMARKS

This paper has shown how MOGA's may be applied to the design of gas turbine engine control systems. Using a single unified formulation, a number of competing design objectives may be simultaneously optimized through search in both controller structure and parameter space. The MOGA approach has a clear advantage over conventional multiobjective optimization methods in that it allows a number of nondominated controller structures to be examined in a single design cycle. Furthermore, the MOGA permits the optimization goals to be changed during the design cycle without the need to reformulate objective functions or restart the search.

A simple user interface has been demonstrated that allows the control engineer to examine the trade-offs and interplay between design objectives. The control engineer may interact with the MOGA through the successive articulation of preferences, guiding the optimization on the basis of design requirements rather than the properties of the objective functions. Such a process allows a closer interaction between the control engineer and the primary design tools, hopefully leading to a more informed design procedure.

While interactive use is desirable, the numerically intensive nature of evaluating objective functions may render such an approach infeasible. In such cases, parallel processing techniques could be employed to alleviate the computational burden. Similarly, the niching mechanisms which arise in some distributed population structures may prove beneficial to the MOGA. Finally, while this paper has considered the application of MOGA's to gas turbine engine design, the procedures and techniques discussed should prove useful in the wider field of CACSD and computer aided engineering (CAE) in general.

ACKNOWLEDGMENT

The authors would like thank Dr. C. Fonseca for the multiobjective extensions to the GA Toolbox and the user interface.

REFERENCES

- [1] H. Mukai, "Algorithms for multicriterion optimization," *IEEE Trans. Automat. Contr.*, vol. AC-25, pp. 177-186, 1980.
- [2] D. E. Goldberg, *Genetic Algorithms in Search, Optimization and Machine Learning*. Reading, MA: Addison-Wesley, 1989.
- [3] C.-L. Hwang and A. S. M. Masud, *Multiple Objective Decision Making—Methods and Applications: A State of the Art Survey*. Berlin, Germany: Springer-Verlag, 1979.
- [4] D. Dasgupta and D. R. McGregor, "Nonstationary function optimization using the structured genetic algorithm," *Parallel Problem Solving from Nature 2*, R. Manner and B. Manderick, Eds., pp. 145-154, 1992.
- [5] C. M. Fonseca and P. J. Fleming, "Genetic algorithms for multiobjective optimization: Formulation, discussion and generalization," in *Proc. 5th Int. Conf. on Genetic Algorithms*, S. Forrest, Ed., pp. 416-423, 1993.
- [6] S. D. Hancock, "Gas turbine engine controller design using multi-objective optimization techniques," Ph.D. dissertation, University of Wales, Bangor, U.K., 1992.
- [7] A. C. W. Grace, "SIMULINK, an integrated environment for simulation and control," in *MATLAB Toolboxes and Applications for Control*, A. J. Chipperfield and P. J. Fleming, Eds. Stevenage, U.K.: Peregrinus, 1993.
- [8] A. J. Chipperfield, P. J. Fleming and H. P. Pohlheim, "A genetic algorithm toolbox for MATLAB," in *Proc. Int. Conf. Systems Engineering*, Coventry, U.K., Sept. 6-8, 1994, pp. 200-207.
- [9] C. M. Fonseca and P. J. Fleming, "Multiobjective optimal controller design with genetic algorithms," in *Proc. Inst. Elect. Eng. Contr. '94*, 1994, pp. 745-749.
- [10] D. E. Goldberg and P. Sergest, "Finite Markov chain analysis of genetic algorithms," in *Proc. 2nd Int. Conf. Genetic Algorithms*, J. J. Grefenstette, Ed., 1987, pp. 1-8.
- [11] D. E. Goldberg and J. Richardson, "Genetic algorithms with sharing for multimodal function optimization," in *Proc. 2nd Int. Conf. Genetic Algorithms*, J. J. Grefenstette, Ed., 1987, pp. 41-49.
- [12] B. W. Silverman, *Density Estimation for Statistics and Data Analysis*. London, U.K.: Chapman and Hall, 1986.
- [13] K. Deb and D. E. Goldberg, "An investigation of niche and species formation in genetic function optimization," in *Proc. 3rd Int. Conf. Genetic Algorithms*, J. D. Schaffer, Ed., 1989, pp. 42-50.
- [14] C. M. Fonseca and P. J. Fleming, "Multiobjective genetic algorithms made easy: Selection, sharing and mating restriction," in *Proc. IEE/IEEE GALESIA '95*, IEE Pub. 414, 1995, pp. 45-52.
- [15] H. Muhlenbein and D. Schlierkamp-Voosen, "Predictive models for the breeder genetic algorithm: I. Continuous parameter optimization," *Evolutionary Computation*, vol. 1, no. 1, pp. 25-49, 1993.
- [16] P. Oliveira, J. Sequeira, and J. Sentiero, "Selection of controller parameters using genetic algorithms," *Engineering Systems with Intelligence*. Norwell, MA: Kluwer, 1991, pp. 431-438.



Andrew Chipperfield received the B.Sc degree in computer systems engineering from the University of Wales, Bangor, U.K., in 1989, and the Ph.D degree in parallel processing in computer aided control system design in 1995 from the University of Sheffield, Sheffield, U.K.

He is currently a Research Associate in the Department of Automatic Control and Systems Engineering at the University of Sheffield with research interests in genetic algorithms for control systems engineering, optimization in control system design, adaptive control, and systems integration for aerospace applications. He also has a number of years experience of computer control in industrial organizations.



Peter Fleming is Professor of Industrial Systems and Control in the Department of Automatic Control and Systems Engineering at the University of Sheffield. He is currently Head of Department and Director of the Rolls-Royce University Technology Centre for Control and Systems Engineering. He has previously held the positions of Professor of Computer Systems Engineering at the University of Wales (Bangor), Associate Professor at Syracuse University, NY, and Research Scientist at NASA, Langley, VA. His research interests include industrial applications of control, computer aided control system design, and control applications of optimization and parallel processing. He has published extensively in these fields.

Prof. Fleming is Chair of the International Federation for Automatic Control Technical Committee on Algorithms and Architectures for Real-Time Control.