ON-LINE GENETIC AUTO-TUNING OF MAGNETIC BEARING CONTROLLERS

P. Schroder,[1] B. Green,[2] N. Grum,[2] P. J. Fleming[1]

ABSTRACT

A prototype large electrical machine running on active magnetic bearings is described. This rig is controlled by a digital signal processor connected by a custom interface to MATLAB/Simulink hosted by a PC. The on-line tuning of a PID controller is set up as an optimisation problem from MATLAB and a multiobjective genetic algorithm is used to drive the optimisation. The results of an optimisation are presented and analysed.

INTRODUCTION

The high reliability and minimal maintenance of magnetic bearings makes them ideal for use in completely sealed or *canned* applications. Canned machines have a number of advantages, including the elimination of potentially unreliable seals, prevention of contamination and a reduction in through-life maintenance costs. Rolls-Royce and Associates Limited (RRA) have constructed a prototype large canned pump levitated on magnetic bearings in order to achieve a very long maintenance free operating life.

This paper demonstrates a novel, but practical approach to the design of active magnetic bearing (AMB) control systems. Modelling the non-linear characteristics of AMBs, especially in canned applications, requires detailed analysis of the magnetic circuit dynamics. This has made the off-line design of high performance controllers for the AMBs on RRA's canned pump difficult. Modern advanced control techniques have been shown to control AMB systems effectively [1-4]. However, these techniques require the development of accurate models and can also be hampered by an involved design process. As a result, a typical industrial control design will begin with the design of a PID controller on a crude model which will subsequently be tuned up manually on a prototype plant to achieve a desired standard of performance. This work demonstrates a convenient method for automating this PID tuning process to produce an optimal design. It is assumed that a stabilising controller already exists and that its performance must be improved. A stabilising

[1]Department of Automatic Control and Systems Engineering, The University of Sheffield, Mappin Street, Sheffield, S1 3JD, UK, tel.: +44 114 222 5250, fax: +44 114 273 1729, e-mail: P.Schroder@Sheffield.ac.uk.
[2]Rolls-Royce and Associates Limited, P.O. Box 31, Derby, DE24 8BJ, UK, tel.: +44 1332 661 461 x5959, fax: +44 1332 622 948.

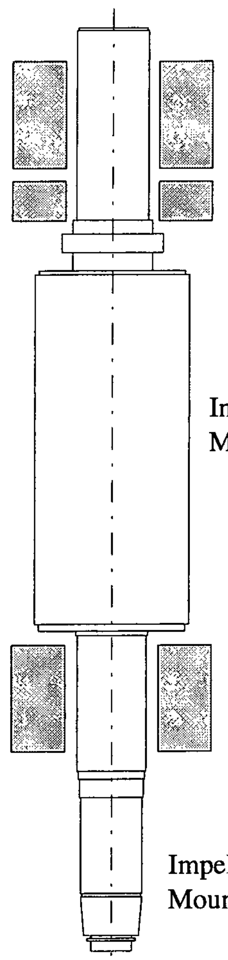

Figure 1: Schematic of the rotor

controller has been developed for the pump rig by manual tuning on-line. A systematic technique for designing optimal PID controllers on-line is presented here and demonstrated to improve the performance of the controller significantly.

The rig is controlled by a digital controller running on a digital signal processor (DSP) card mounted inside a PC. Controllers are specified in MATLAB/Simulink, a dynamic simulation environment. An auto-code generator and a custom interface to Simulink allow the controller parameters on the DSP to be adjusted from MATLAB/Simulink without interrupting control. Sensor data from the rig can also be logged by MATLAB in real-time. It is thus possible to formulate a hardware-in-the-loop optimisation problem in MATLAB with the controller parameters as design variables and measures of the rotor's response as optimisation objectives.

This type of problem would prove difficult for a conventional optimiser as the optimisation is highly non-linear and subject to random noise. Genetic algorithms, however, are comparatively robust to these problems as they use a population of potential solutions and are stochastic in nature. A multiobjective genetic algorithm (MOGA) is therefore chosen as the optimisation engine for designing PID controllers on-line. The advantage of using a multiobjective optimiser is that different measures of performance can be optimised simultaneously without a priori defining their relative importance [5]. Different measures of tracking performance are used as optimisation objectives, with their design specifications as targets for the optimiser. The designer is ultimately presented with many different controllers, all of which satisfy the specification. The designer then chooses the controller which offers the best performance for the application.

THE APPLICATION

The application consists of a large electric motor driven pump rotor levitated on active magnetic bearings. The rotor is mounted vertically and weighs approximately 200 kg. The AMB system fitted to the machine provides rotor control in two orthogonal directions radially at the drive end of the pump, two orthogonal directions radially at the non-drive end and one direction vertically (the thrust/axial bearing). Figure 1 shows a schematic of the rotor and indicates the bearing locations. The pump's impeller is mounted at the bottom of the rotor and the entire rotor is sealed in a stainless steel can.

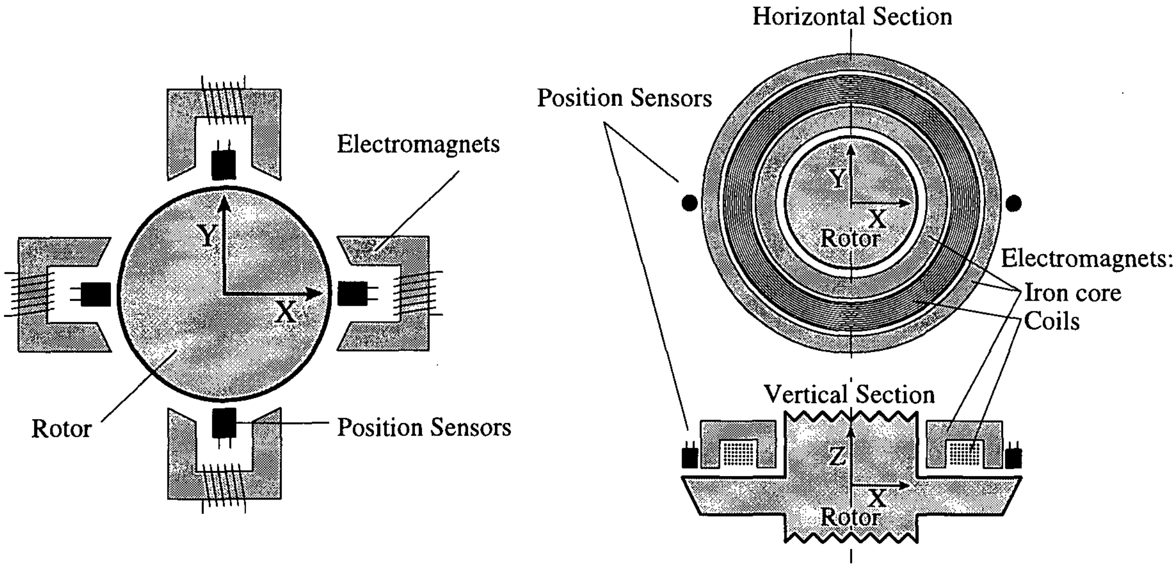

Figure 2: Configuration of the radial and axial bearings

Each of the four radial bearing systems consists of a pair of electromagnets mounted one either side of the rotor journal as shown on the left of Figure 2. A position sensor is also mounted on either side of the journal. The analogue voltages from each sensor pair are conditioned and the differential voltage routed to a digital control system. The control system generates an analogue output demand signal which is fed to a power amplifier and this in turn, drives current into the appropriate pair of electromagnets situated on each side of the rotor. Each electromagnet produces an attractive force acting on the rotor. The net radial force generated provides levitation at the bearing.

The axial bearing operates as a single electromagnet mounted vertically above the rotor. As with the radial bearings, the electromagnet produces an attractive force. This levitates the rotor and is countered by the rotor weight acting downwards against gravity. Associated with the thrust bearing is a pair of position sensors and a power amplifier. These are interconnected to the digital control system in the same manner as a radial bearing. The right of Figure 2 shows a vertical and horizontal cross section of the axial bearing.

INTERFACE DESCRIPTION

A complete PC-based digital control system has been assembled, utilising a TMS320C40 DSP industry standard card with an appropriate analogue input/output module. This module provides up to 16 analogue input channels, 8 analogue output channels and 4 digital input/output lines for connection to external equipment. The cards are sited in a 100MHZ Pentium PC and form a complete standalone dedicated control system for an AMB application. The connection of the control system to the application rig is illustrated in Figure 3. The communications link between the cards in the PC is carried out by the industry standard DSPlink protocol which allows fast internal data transfers to occur and simplifies the interfacing of external hardware to the control system. The computing power and architecture of the TMS320C40 DSP allows complex strategies such as multi-variable and coupled controllers to be implemented. The structure of the PC-based controller also lends itself to the implementation of high integrity control systems with redundancy management and self-checking functions.

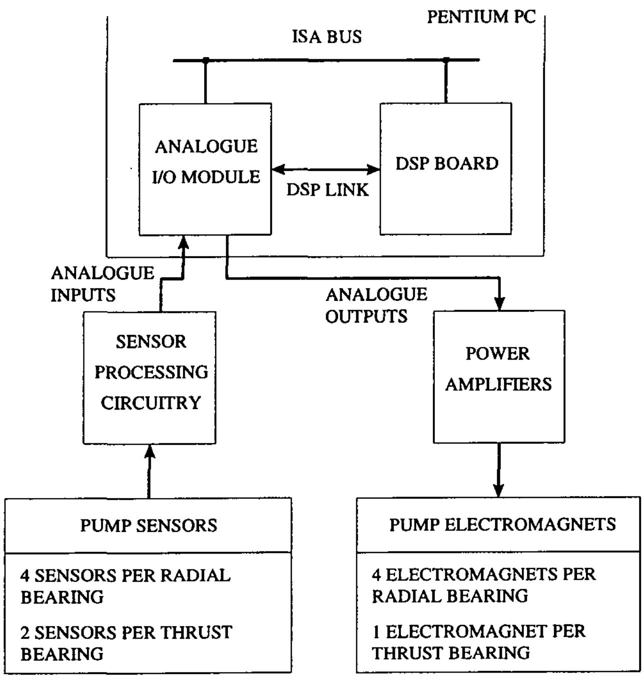

Figure 3: The digital control system

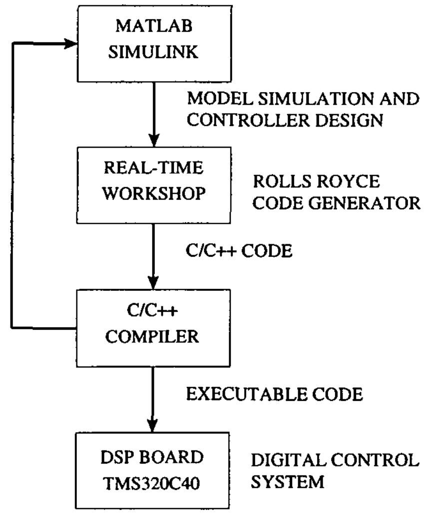

Figure 4: Software design process

Figure 4 details the complete design process which has been developed and installed on the PC-based digital control system. Industry standard MATLAB/Simulink software development tools are used extensively throughout the process. The high level block diagram Simulink package is used for AMB system modelling and simulation work and digital controller development. The designed controller can then be extracted from the overall system schematic diagram and real-time executable code produced for the controller by using the other software tools shown in Figure 4. The Rolls-Royce Simulink Rapid Real-Time Code Generator software provides an interface between Simulink and the hardware located inside the PC-based control system. This is accomplished by using the MathWorks Real-Time Workshop tool and custom software routines to generate C-code which is suitable for running on the hardware platform. A Tartan TMS320C40 C/C++ compiler generates executable code for the DSP board.

A specially written blockset software file is installed on the on the PC-based control system. This provides a library of Simulink blocks which allows the DSP board to interface with the analogue I/O module. This module is configured such that the hardware interrupts which synchronise the analogue to digital sampling functions on the board can also be used to synchronise the Simulink design. This provides a fully synchronous design solution which is independent of the DSP code execution time.

The target development system is comprised of the PC-based digital control system running Simulink and a TMS320C40 target processor on an industry standard TIM module located on the DSP board. The host computer runs the Simulink Real-Time Workshop, the Tartan C/C++ compiler and the software interface utility. The DSP board is connected to the PC using shared memory via an ISA expansion slot. The analogue I/O module is connected to the DSP board via the DSPLink interface. The real-time code is compiled on the PC by the Tartan compiler and the executable code is downloaded to the DSP board by using a software download utility. The download utility initialises the DSP hardware, loads the application executable code file into the DSP's memory area and initiates program execution. Another software utility allows on-line adjustment of Simulink schematic parameters while the executable code for the schematic is running on the hardware platform.

HARDWARE-IN-THE-LOOP DESIGN

The motivation for designing controllers directly onto real hardware is to avoid the need to develop the very accurate models needed to design them off-line. The software interface in use here allows controller parameters to be altered on the rig from within the MATLAB environment and also allows sensor data to be logged back into MATLAB. This enables an optimisation problem to be constructed with the controller parameters as decision variables and direct measures of the controller's performance as optimisation objectives. The numerical power of MATLAB is then used to drive the optimisation towards a good solution using a multiobjective genetic algorithm.

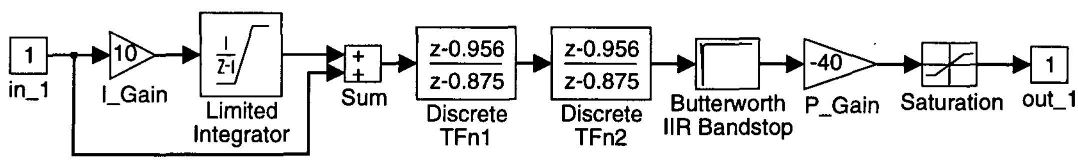

Figure 5: Simulink diagram of the controller structure

The structure of the controller must be specified in advance to avoid the need to recompile the code for each controller evaluated. The structure adopted here is that of a PI controller with two phase lead terms and a notch filter in series with it. This structure is used because a stabilising controller in this form already exists and a conventional PID structure had previously been found to perform less well due to noise on the derivative term. Figure 5 shows the controller structure in Simulink form.

The notch filter is tuned to the resonant frequency of the rotor which is significantly higher in frequency than any of the other dynamics in the system. Without this filter, the rotor simply vibrates at this frequency, clearly compromising its stability. As the filter is already accurately tuned to this frequency, it is not included in the optimisation's search domain.

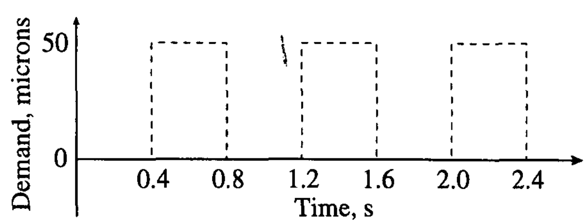

Figure 6: Demand signal applied for controller evaluation

The optimisation objective functions operate the rig in two modes. First, a known stabilising controller is used to levitate the rotor. Second, the parameters under evaluation are switched in and their performance monitored. If the controller is unstable, the stabilising controller is switched back in before the rotor moves too far from its desired location. A 1.25Hz square wave is constantly applied as a demand signal, (see Figure 6). This excites the system and allows various performance measures to be taken. The metrics used here are the peak overshoot (rising and falling) and the mean absolute error when the demand signal is high and when it is zero. After a controller has been evaluated the rig is reset with the stabilising controller installed, to ensure that every controller is evaluated against the same metrics.

MULTIOBJECTIVE OPTIMISATION

A multiobjective genetic algorithm (MOGA) is used here to find a set of optimal controllers. It is implemented using the GA Toolbox for MATLAB [6], with additional extensions to accommodate multiobjective ranking, sharing and mating restrictions [7].

Multiobjective or Pareto ranking is based upon the dominance of an individual, i.e. how many individuals out-perform it in the objective space. This kind of ranking is non-unique; for example, a number individuals may be ranked 0, i.e. non-dominated. Figure 8 demonstrates the way in which Pareto optimal ranking is achieved for a two objective minimisation problem. Note, for example, that the solution ranked 5 is dominated by 5 other solutions in a multiobjective sense.

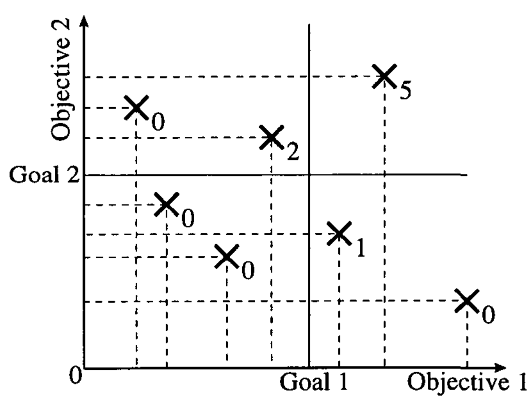

Figure 7: Pareto optimal ranking

Ranking may also be combined with goal and/or priority information to discriminate between non-dominated solutions. For example, a solution in which all the goals are satisfied may be considered superior, or *preferable*, to a non-dominated one in which some components go beyond the goal boundaries. The two points in Figure 7 that are inside the goal boundaries are therefore preferable and would consequently be ranked better than the other two non-dominated points. Here, design specifications such as the maximum tolerable percentage overshoot are used as goals. Details of the function and purpose of the various operators used in the MOGA can be found in references [6-9].

The controller parameters are represented here in a binary chromosome constructed of 4 sections of 12 bits representing the proportional and integral gains and the pole and zero locations of the lead terms (both lead elements are set to be the same). The search is constrained to within ±30% of the parameters of the original stabilising controller to ensure swift convergence.

RESULTS

An optimisation was performed for the non-drive end x-axis controller, with the other axes levitated using the stabilising controller. Figure 8 shows the step response of the manually tuned controller used as a starting point for this optimisation. It is clearly rather oscillatory, has a large overshoot and takes a long time to settle. The objective function measures associated with this controller are used as goals for optimisation, so that the MOGA tries to find solutions that dominate it (are better in every performance measure).

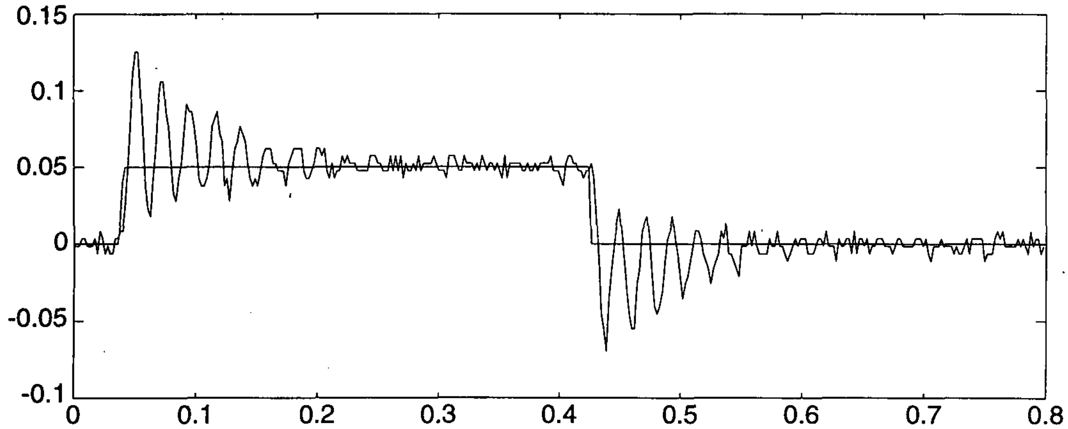

Figure 8: Response of the manually tuned controller

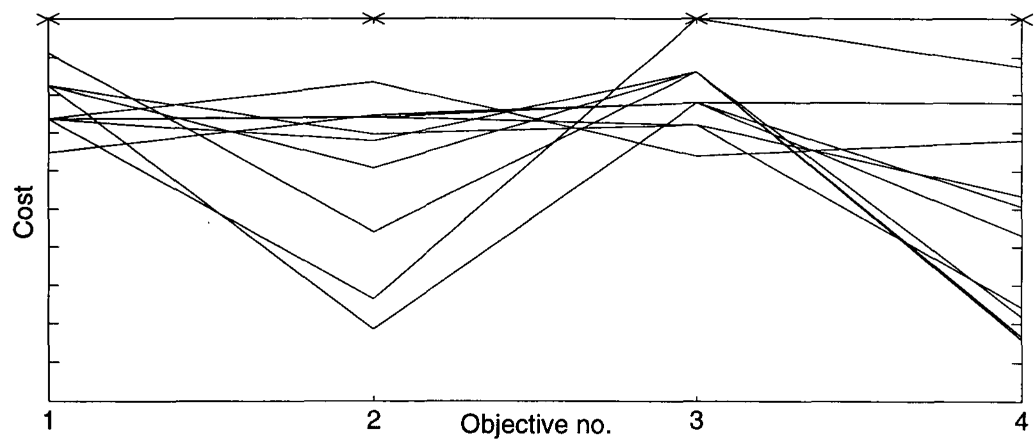

Figure 9: Sample trade-off graph

Figure 9 shows a random subset of solutions from a typical trade-off graph for the AMB control system. The x-axis shows the design objectives and the y-axis shows the objective domain performance of the controllers. Each line represents a single non-dominated solution's performance against each objective. Trade-offs between adjacent objectives result in the crossing of lines whereas concurrent lines represent non-competing objectives. Table 1 shows the objectives and their displayed ranges, the maximum of each objective's displayed range is the same as its goal (the performance of the original controller). The 'x' marks in the figure represent the optimisation targets or goals. Every controller shown in Figure 9 exhibits a significant performance improvement over the original controller, with different solutions achieving greater improvements in different objectives.

No.	Objective Name	Min	Max (goals)
1	Overshoot (rising)	50%	160.9%
2	Mean error (high)	10%	20.3%
3	Overshoot (falling)	50%	120.1%
4	Mean error (zero)	10%	20.0%

Table 1: The objectives and their displayed ranges

An interesting feature of the trade-off graph is the difference in overshoot between rising and falling step demands (note the graph's scales, Table 1), with better performance being achievable with zero demand. This anisotropic behaviour is due to the off-centre location of the high demand position causing the plant to operate in a more non-linear region. This trade-off surface encompasses the achievable performance with this controller configuration and specification. It is possible to improve on any of the objectives beyond what is apparent here, but doing this will violate the goal boundary for some or all the other objectives. Study of Figure 9 reveals that there is some trade-off between all the objectives, with no single objective appearing to trade-off heavily with any other. Comparison of the overshoot objectives (1 and 3) reveals that there is little trade-off between them, and similarly for the mean error objectives (2 and 4). This is expected as they are similar performance measures.

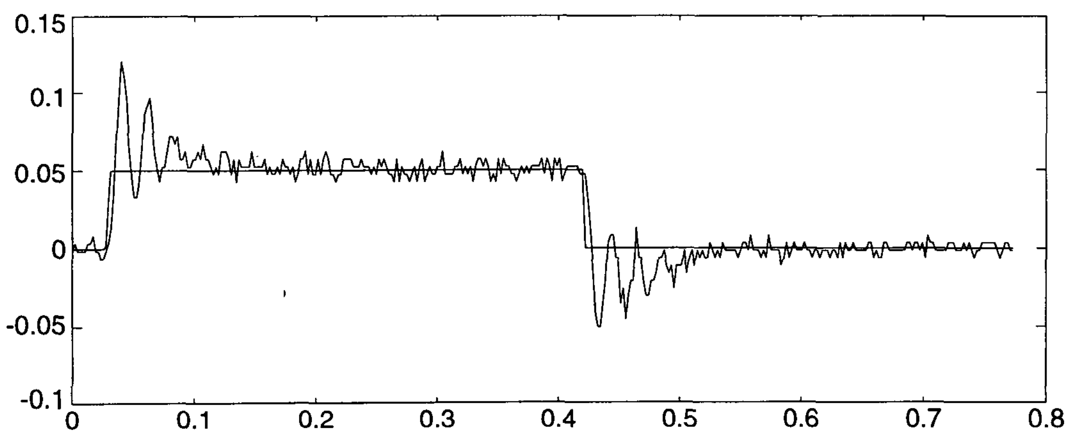

Figure 10: Rotor's response under controller 1

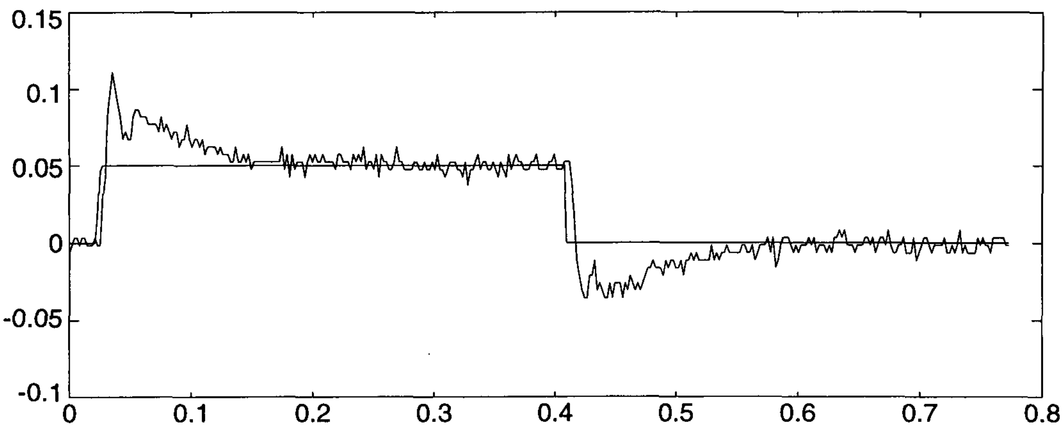

Figure 11: Rotor's response under controller 2

To further illustrate the inherent trade-offs present in the system, response graphs of two pareto optimal controllers are shown in Figures 10 and 11. Controller 1 has a large rising overshoot, moderate falling overshoot and it exhibits good mean absolute error characteristics. Controller 2 has a small overshoot for both rising and falling demands, but has a large mean absolute error.

Despite the optimisation being allowed to range up to ±30% from the original parameters, neither of the two displayed controllers have any parameters that differ by more than 16% from the original. Table 2 shows the parameters of the original and both example controllers. Quite different responses are apparent with only small changes in the parameters, this level of sensitivity makes high performance difficult to achieve by manual tuning.

Parameter	Original	Controller 1	Controller 2
Gain	40	44.8	43.9
Integral gain	10	11.5	9.3
Pole location (s - domain)	-250	-250.8	-211.1
Zero location (s - domain)	-700	-776.6	-756.5

Table 2: The controllers' parameters

The mean absolute error at zero demand is probably the most important objective as this is the condition the system will be in most of the time. In a conventional weighted sum optimisation, this objective would have been weighted as most important and the designer would not have been aware of the size of penalty that would be paid in terms of the transient performance. The unknown and non-linear nature of the trade-off surface would also hinder a single objective optimiser. The multiobjective optimisation is not susceptible to these problems as it treats every objective equally in a Pareto sense. Bias in favour of one objective or another is introduced after the optimisation by the designer when it is clear what effect this has on the other objectives. This demonstrates the power of multiobjective optimisation to explore the system's capabilities and present the designer with an unbiased set of optimal controllers such that the most suitable one may be selected.

CONCLUDING REMARKS

A multiobjective genetic algorithm is used as a design tool for generating optimal active magnetic bearing controllers for a Rolls-Royce large electric machine application. The MOGA is used to search a controller parameter selection problem for the non-linear AMB system. The optimisation is performed directly onto the AMB rig, and several measures of AMB performance are used as objectives for the optimisation. From these a great deal of information about the limiting characteristics of the many possible controllers can be inferred. A collection of satisfactory controllers is generated, from which a controller with a good performance for the application can be selected.

This powerful design technique not only gives insight into the behaviour of the system, but allows the designer to select the most appropriate compromise control solution for the particular system under development. In the end, it is the designer who makes the decision about what controller structure and parameters are to be used. The MOGA is simply used as an efficient way to explore the possibilities offered by each alternative.

ACKNOWLEDGEMENTS

The authors gratefully acknowledge the support of this research by Rolls - Royce & Associates Limited and UK EPSRC grants on "Evolutionary Algorithms in Systems Integration and Performance Optimisation" (GR/K 36591) and "Multiobjective Genetic Algorithms" (GR/J 70857).

REFERENCES

[1] P. Schroder, A. J. Chipperfield, P. J. Fleming, N. Grum, "Robust multivariable control of active magnetic bearings", *Proc. European Control Conference, ECC 97,* Brussels, 1997.

[2] R. L. Fittro, C. R. Knospe, L. S. Stephens, "Experimental results of μ-synthesis applied to point compliance minimisation", *Proc. 5th International Symposium on Magnetic Bearings,* Kanazawa, Japan, pp. 203-208, 1996.

[3] K. Nonami, W. He, H. Nishimura, "Robust control of magnetic levitation systems by means of H_∞ control / μ-synthesis", *JSME International Journal, Series C, Vol. 37,* No. 3, 1994.

[4] J. C. Ponsart, J. Lottin, P. Mouille, "Non-linear control of active magnetic bearings : digital implementation", *Proc. 5th International Symposium on Magnetic Bearings,* Kanazawa, Japan, pp. 77-82, 1996.

[5] P. Schroder, A. J. Chipperfield, P. J. Fleming, N. Grum, "Multiobjective optimisation of distributed active magnetic bearing controllers", *Proc. GALESIA 97,* Glasgow, 1997.

[6] A. J. Chipperfield, P. J. Fleming, H. P. Pohlheim, "A genetic algorithm toolbox for MATLAB", *Proc. Int. Conf. Systems Engineering,* Coventry, UK, pp. 200-207, 1994.

[7] C. M. Fonseca, P. J. Fleming, "An overview of evolutionary algorithms in multiobjective optimisation" *Evolutionary Computation,* Vol. 1, No. 1, pp. 25-49, 1993.

[8] D. E. Goldberg, J. Richardson, "Genetic algorithms with sharing for multimodal function optimisation", *Proc. 2nd Int. Conf. on Genetic Algorithms,* J. J. Grefenstette (Ed.), pp. 41-49, 1987.

[9] K. Deb, D. E. Goldberg, "An investigation of niche and species formation in genetic function optimisation", *Proc. 3rd Int. Conf. on Genetic Algorithms,* J. D. Schaffer (Ed.), pp. 42-50, 1989.