CONTROL SYSTEM DESIGN FOR A GAS TURBINE ENGINE USING EVOLUTIONARY COMPUTING FOR MULTIDISCIPLINARY OPTIMIZATION

Valceres V. R. e Silva[*]
vvrsilva@ufsj.edu.br

Wael Khatib[†]

Peter J. Fleming[†]
P.Fleming@sheffield.ac.uk

[*]Universidade Federal de São João del Rei
Praça Frei Orlando 170 - 36307 352 - São João del Rei - MG

[†]The University of Sheffield
Mappin Street, S1 3JD - Sheffield - UK

ABSTRACT

Multidisciplinary optimization (MDO) is concerned with complex systems exhibiting challenges in terms of organization and scale. Thus, it is well suited to be applied to complex multivariable control design. Collaborative optimization is one approach for dealing with complex multidisciplinary optimization problems. Three MDO architectures, including collaborative optimization, are applied to control system design for a gas turbine engine, in order to improve the design search process by exploring possible solutions with parallel, but independent search strands. The optimization is carried out through a multiobjective genetic algorithm framework.

KEYWORDS: Genetic algorithms, gas turbines, optimization, PI controllers.

1 INTRODUCTION

There is a significant body of research devoted to the study of design and optimization of a number of interacting or coupled systems. Most of this research tends to be related to aerostructural design and is called multidisciplinary optimization. The design of an airplane requires the bringing together of resources representing structures, metallurgy, aerodynamics, performance, control and other disciplines in order to produce an optimal design. The main challenges faced in MDO design problems are computational cost and organizational complexity (Sobieszczanski-Sobieski and Haftka, 1996). The complexity of design optimization depends on the complexity of the pertinent disciplines, the size of the problem, and the nature of the objectives and constraints. Comparing with an aggregation of many disciplines, the problem grows very much in complexity, if there is more than one discipline controlling the same design variables for a particular objective. This is mainly due to the effect of coupling between the variables. Organizational complexity is due to the fact that the various disciplines traditionally reflect different analysis methods, schools of thought, software and hardware platforms, standards, etc. The organizational challenge in MDO is for an efficient exchange of data, systems integration and other aspects of communication.

Evolutionary computing refers to computer-based problem solving systems that use evolutionary algorithms (EAs). EAs generally use computational models that exploit mechanisms based on the neo-Darwinian theory of evolution. The main techniques used in EAs include: genetic algorithms

Artigo submetido em 21/11/2004
1a. Revisão em 30/03/2005
2a. Revisão em 11/01/2006
3a. Revisão em 12/07/2007
Aceito sob recomendação do Editor Associado
Prof. Takashi Yoneyama

(GAs), evolutionary programming (EP), evolution strategies (ES) and genetic programming (GP). EAs are amenable to parallelization and can help reduce the computational cost. These algorithms are stochastic in nature and can usually start an optimization process without much a priori knowledge. No derivative information is required as in the traditional gradient based methods and this helps EAs deal with difficult search spaces characterized by multimodal disjoint feasible areas.

Most design problems are multiple objectives in nature, including MDO problems. These objectives are often conflicting or competing. The concept of Pareto optimality is a powerful method for dealing with multiple objectives. Using this approach, the designer is no longer searching for a single optimum, rather a compromise satisfying the various objectives. and constraints. The collection of compromise solutions is referred to as the non-dominated set. Within this set, attempted improvement in one objective will result in degradation in one or more of the others. EAs are amenable to multiobjective optimization (MO). This is because an EA works on a population of solutions instead of the traditional single point search. The search with this population can help achieve a faster and more comprehensive mapping of the trade-off hyper surface. An overview of the application of EAs to MDO can be found in Khatib and Fleming (1997).

2 THE GAS TURBINE ENGINE

Gas turbine engines (GTE) are highly nonlinear plants that have multiple inputs and outputs. The operating conditions span extremes of temperature, pressure and load conditions. The engine performance requirements cover a wide flight envelope that includes a continuum of set points of altitude and speed in terms of the Mach number. These requirements add to the complexity of designing suitable controllers that can achieve high performance levels while maintaining stability and safe operation with minimum overall cost. H_∞ and PI (proportional and integral) controllers have been designed for the GTE using simplified models obtained through the response surface variable complexity modelling technique (Silva et *al*., 2007). It was also obtained inprovements on this engine's performance by reducing fuel consumption, increasing thrust in dash missions and minimizing turbine blade temperature (Silva et *al*., 2005). In this work, multidisciplinary collaborative optimization structures are used to split the PI controller design problem in three and thus, optimization is carried out.

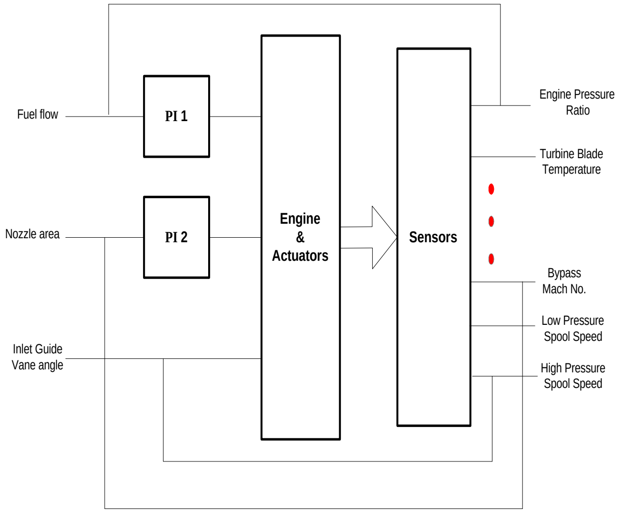

Figure 1: PI controller structure for the Spey GTE.

The engine model has three inputs: fuel flow (wFE), exhaust nozzle area (A8) and inlet guide vane angle (IGV). Sensors provided from engine outputs include: high pressure spool speed (NH), low pressure spool speed (NL), engine pressure ratio (EPR) and fan pressure ratio. These measurements can be used to provide various pairings of input-output for closed-loop control. Important engine variables such as thrust (XGN) and surge margin (LPSM) cannot be measured directly. Such variables are controlled implicitly through other related measurable values such as pressure ratios and bypass duct Mach number (DPUP). Silva and Fleming (2002) used a non-linear model for control configuration and PI (proportional and integral) controller design, and also for a H_∞ controller design. Using findings from these studies, a closed-loop control strategy is chosen for parametric optimization of a designed PI controller structure for a particular operating point to demonstrate the use of MDO architectures for control design (Fig. 1).

Two PI controllers are implemented for controlling XGN and LPSM. Because these two variables cannot be measured directly, they are controlled implicitly through EPR and the DPUP respectively. The third input, inlet guide vane angle (IGV), is gain scheduled against the measured output values of the NH. The two PI controllers supplied with the Rolls-Royce SIMULINK model of this engine, use the structure proposed by Åström and Häagglund (1995), and it is shown in Figure 2.

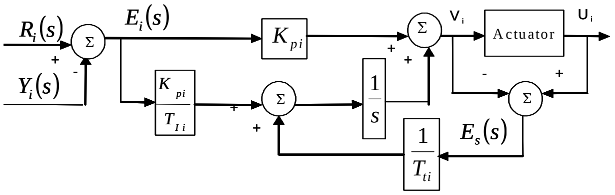

Figure 2: PI controller with anti-windup.

When there is no actuator saturation, the anti-windup feedback signal E$_i$(s) is zero The expression of the output U$_i$ of the i-*th* controller is given by equation 1.

$$U_i\left(s\right) = K_{pi} E_i\left(s\right) + \frac{K_{pi}}{T_{Ii} s} E_i\left(s\right) \tag{1}$$

where, E$_i$ = R$_i$ – Y$_i$, is the error signal for loop i, R$_i$ is the set point signal for loop i, and i = 1, 2. With reference to Figure 2, K$_{pi}$ and T$_{Ii}$ denote the parameters of the i-*th* PI controller. The parameter T$_{ti}$ is known as the tracking time constant, and controls the effect of the integral anti-windup mechanism. This structure has been applied to design optimized PI controllers with genetic algorithm for industrial plants (Ghaffari, 2007).

The design is expected to satisfy the following objectives and constraints:

a. XGN > 48.6 kN (performance requirement).

b. TBT < 1713^{o}C (physical limit).

c. LPSM > 10% (stability limit).

d. XGN rise and settling time for speed of response performance.

e. steady state error limits for NH and NL.

f. steady state error limit for exhaust nozzle area (A8).

g. limits on overshoot/undershoot for transient regions of NH and NL.

h. other engine limits for spool speeds and exhaust nozzle area.

A successful design of an engine controller can be achieved as part of, say, an overall MDO design of an aircraft. Such designs are still not attempted on a full scale in industry due to the limitations of available resources for MDO. However, MDO techniques can be applied to the design process for the controller *per se*. To demonstrate this, we present an example for designing a PI control strategy for a GTE that comprises two PI controllers. A nonlinear SIMULINK model for a Spey GTE is used to design and evaluate controller performance.

3 MULTIOBJECTIVE GENETIC ALGORITHM

Multiobjective optimization and decision making refers mainly to simultaneous optimization in order to achieve optimal trade-off solutions satisfying various objectives. These objectives tend to be conflicting or competing. There is not usually one unique solution but rather a family of compromise solutions that need to be analyzed by a decision maker.

The multiobjective genetic algorithm (MOGA) combines the characteristics of a powerful evolutionary optimization strategy, the genetic algorithm with the concept of Pareto optimality (Giannakoglou, 2002) to produce solutions illustrative of a problem's trade-off set. A MOGA evolves a population of solution estimates thereby conferring an immediate benefit over conventional multiobjective optimization methods.

Mathematically, the multiobjective optimization (MO) problem is to find a vector of design variables x, that is within the feasible region in the universe $\Re$, to minimize (or maximize) a vector of objective functions F(x). Some or all of the component functions can be non-linear. Most practical problems are also bounded by a vector of constraints g(x). Multiobjective optimization can be expressed as follows:

$$\begin{array}{l}\text{Minimize: } F(x) = \{f_1(x), f_2(x), \ldots, f_n(x)\} \\ \text{subject to: } g(x) \leq 0\end{array} \tag{2}$$

where $g\left(x\right)$ is the constraint vector and $f_i\left(x\right)$ is the i-th objective function.

The set of trade-off solutions that express the best performance in all of the objectives is known as the Pareto or the non-dominated set.

The concept of Pareto-optimality constitutes by itself the origin of research in multiobjective optimization. In a multiobjective minimization problem, a feasible vector $x^* \in X$ is Pareto-optimal if and only if there is no feasible vector $x \in X$ such that for all **i** $\in$ **{1, 2, ..., n}**

$$f_i\left(x^*\right) \leq f_i\left(x\right) \tag{3}$$

and for at least one **i** $\in$ **{1, 2, ..., n}**

$$f_i\left(x^*\right) < f_i\left(x\right) \tag{4}$$

The decision making process picks the best solution from the non-dominated set (Pareto-optimal) according to some preference information.

Most real-world optimization problems are multi-modal. There often exist several criteria to be considered by the designer. The compromise of better performance for all of them has to be achieved. Fonseca and Fleming (1995) use the ranking approach for assigning fitness to each individual in the population. They define the individual's rank simply as the number of members of a population in a generation that dominate it. Thus, non-dominated individuals are assigned rank zero, while the lowest possible rank in any generation of population is rank **n-1**, where **n** is the number of individuals in the population. The fitness is then assigned to each individual by interpolating from the best to the worst, according to some function, that can be linear, exponential or other type.

Any attempted improvement for a member of this set in one of the objectives will result in deterioration in performance in one or more of the other objectives.

The work described here employs a genetic algorithm with an implementation of multiobjective optimization as proposed by Fonseca and Fleming (1993).

4 MDO ARCHITECTURES

Balling and Sobieszczanski-Sobieski (1996) introduce a consistent method for classifying the various approaches for formulating MDO problems using compact and consistent notation. For practical purposes, the various approaches can be grouped into three main categories:

i. Single-level approaches

j. Collaborative optimization

k. Concurrent sub-space optimization

A detailed discussion of these methods can be found in Khatib and Fleming (1998). Single-level approaches address a design problem as one whole unit. Concurrent sub-space optimization relies on recursive iteration loops and can be less efficient. The collaborative optimization (CO) approach works through decomposition of complex large-scale problems into smaller sub-problems or elements. Each element proceeds with its own optimization using separate decision variables. The outputs of the various disciplines can be pooled into a shared resource area to be redistributed or observed by a system designer or optimizer to ensure a viable overall design. The decomposition boundaries depend on the physical organization of the problem, the available resources and/or the mathematical limits. This gives rise to soft and hard boundaries. The evolutionary collaborative architecture proposed by Khatib and Fleming (1998) allows the various disciplines to progress simultaneously in search of optimal designs. This approach draws on a seemingly good match between the two elements of CO and EAs. They share amenability to parallelization and this is also reflected in existing practices in industry.

5 MOGA-PI DESIGN ARCHITECTURE

For single level optimization of a MIMO (Multi Input Multi Output) system, in terms of the underlying GA engine in the MOGA (multiobjective genetic algorithm) framework, the two main operators that guide the optimization process are selection and crossover. The selection pressure which favors the fitter individuals is based on how well each of the individuals performs in terms of Pareto optimality. Grouping the proportional and integral terms for both controllers together for each individual suggests that an integral term, say, might be promoted in the population based on influence of a proportional term in the same individual. This form of nepotism can be inefficient in computing terms and might even lead to less optimal solutions.

Three different frameworks using MOGA are applied, employing one single-level and two collaborative design approaches. The single-level MOGA addresses the two controllers and all the objectives simultaneously. Each individual in the MOGA population is a chromosome made up of the four controller parameters in a Gray-coded binary representation. The two CO implementations are different in the way the design problem is decomposed.

CO I has two optimization sub-problems: one dealing with the proportional gains in both controllers, and the other dealing with the integral gains. The objectives for each of these sub-problems reflect potential scope of influence. The proportional gains are designed to optimize all the objectives and constraints of the problem simultaneously, while the integral gains are considered for optimization of XGN and TBT dynamics only. The optimization process for P1 and P2 can be described as follows:

$$
\begin{array}{ll}
\text{Maximise:} & \text{LPSM} \\
\text{Minimize:} & \text{NH and NL steady-state errors} \\
 & \text{A8 steady-state error} \\
 & \text{NH and NL overshoot/undershoot.} \\
\text{Satisfy:} & \text{XGN} \geq 48.6\ \text{kN} \\
 & \text{TBT} \leq 1713^\circ\text{C} \\
 & \text{XGN rise time} \leq 1.0\ \text{s} \\
 & \text{XGN settling time} \leq 1.4\ \text{s} \\
 & \text{NH} \leq 102\% \\
 & -8^\circ \leq \text{IGV} \leq 32^\circ \\
 & 0.25\ \text{m}^2 \leq \text{A8} \leq 0.34\ \text{m}^2
\end{array}
\qquad (5)
$$

The optimization process fo I1 and I2 has no soft objectives. It is a constraint satisfaction problem in which MOGA is used to satisfy the following constraints:

$$\begin{array}{l}
\text{TBT} \leq 1713^{\circ}\text{C} \\
\text{XGN rise time} \leq 1.0 \text{ s} \\
\text{XGN settling time} \leq 1.4 \text{ s} \\
\text{NL} \leq 102\% \\
\text{-}8^{\circ} \leq \text{IGV} \leq 32^{\circ} \\
0.25 \text{ m}^2 \leq \text{A8} \leq 0.34 \text{ m}^2
\end{array} \tag{6}$$

CO II designed for this problem depicts the boundaries by having each of two sub-problems dealing with one of the two PI controllers. The pertinent objectives also reflect the level of influence: PI1 design optimizes XGN, NH and NL dynamics (objectives d and g), steady state errors for NH and NL (objectives e) whilst minimizing TBT (objective b). PI2 design optimizes the A8 variations (objective f) while maintaining LPSM and TBT within safe limits (objectives b and c). It can be noticed that TBT is common to both controllers. This suggests coupling in the system and can help the two design sub-problems achieve a satisfactory overall results. The optimization process for PI1 can be described as follows:

$$\begin{array}{ll}
\text{Minimize:} & \text{NH and NL steady-state errors} \\
 & \text{NH and NL overshoot/undershoot.} \\
\text{Satisfy:} & \text{XGN} \geq 48.6 \text{ kN} \\
 & \text{TBT} \leq 1713^{\circ}\text{C} \\
 & \text{XGN rise time} \leq 1.0 \text{ s} \\
 & \text{XGN settling time} \leq 1.4 \text{ s} \\
 & \text{NL} \leq 102\% \\
 & \text{-}8^{\circ} \leq \text{IGV} \leq 32^{\circ} \\
 & 0.25 \text{ m}^2 \leq \text{A8} \leq 0.34 \text{ m}^2
\end{array} \tag{7}$$

The optimization process for PI2 can be described as follows:

$$\begin{array}{ll}
\text{Maximise:} & \text{LPSM} \\
\text{Minimize:} & \text{A8 steady-state error} \\
\text{Satisfy:} & \text{TBT} \leq 1713^{\circ}\text{C} \\
 & \text{NL} \leq 102\% \\
 & \text{-}8^{\circ} \leq \text{IGV} \leq 32^{\circ} \\
 & 0.25 \text{ m}^2 \leq \text{A8} \leq 0.34 \text{ m}^2
\end{array} \tag{8}$$

The work described here employs a GA with an implementation of MO as proposed by Fonseca and Fleming (1993). This multiobjective GA (MOGA) presents a simple but powerful tool for design optimization were preferences can be articulated progressively by the decision maker as more insight into the problem in hand is gained.

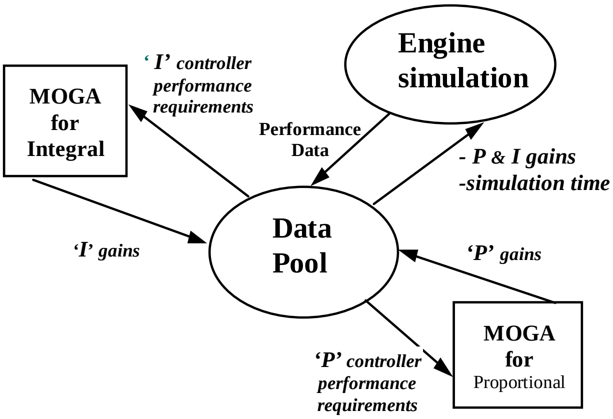

Figure 3: MOGA CO I architecture.

6 IMPLEMENTATION

The three different frameworks described in the previous section are used for parametric optimization of a PI controller structure. The MOGA generates two sets of PI controller gains for the wFE and A8 loops. The MOGA parameters for crossover and mutation are 90% and 5% respectively.

Because of the parallel nature of MOGA, the two sub-problems in CO I and CO II run concurrently. Each problem has its own population of solutions. Each population experiences selection and crossover driven by its own pertinent objectives. The two populations do not share or individuals through migration. To achieve a consistent overall design, each population presents its proposed solutions to a shared data pool which performs the engine simulations (Figure 3). Results of the simulations are fed back into the pool and each separate MOGA scans for its relevant outputs to drive its own GA operators.

The control response is measured after a step input in thrust demand, corresponding to changes from 85% to 90% of high pressure spool speed. Using two populations of solutions simply increases the requirements to compute the MOGA operators for ranking, selection, crossover and mutation. These costs are negligible when compared to the cost of engine simulations. The number of simulations required is the same as for the single-level optimization case. Each of the three MOGAs employs 40 individuals and is evolved for 50 generations.

7 RESULTS

The multidisciplinary optimization and multiobjective optimization produce a family of non-dominated solutions, for each of the three aproaches: 21, 12 and 12 controller sets for the single-level, CO I and CO II respectively. Figures

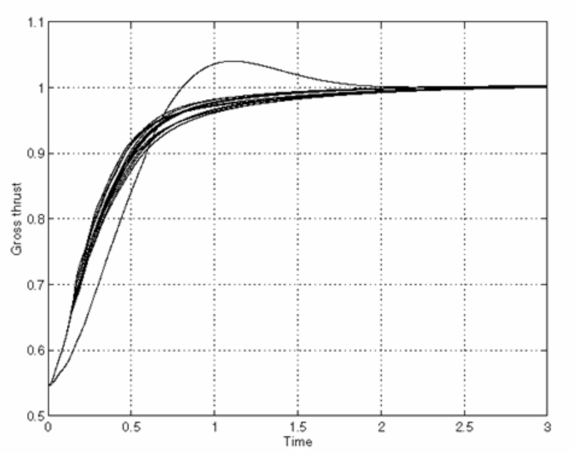

Figure 4: Thrust responses for single-level approach.

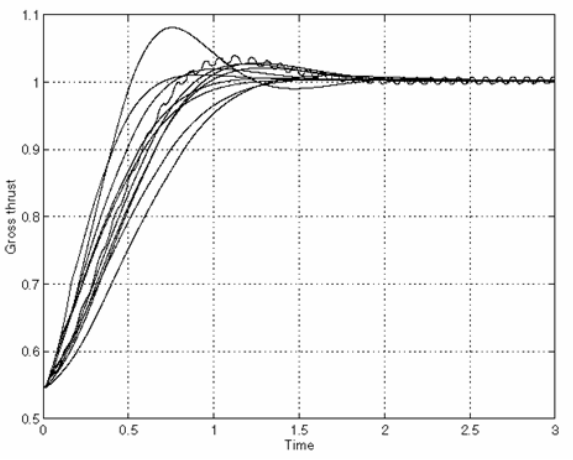

Figure 5: Thrust responses for CO I approach.

4 – 5 present XGN step responses for changes from 35% to 87% of thrust demand, for the obtained controllers of the three design architectures. The responses are normalized for the desired point. Most of these controllers exhibit similar performance characteristics for the three structures. For the CO II approach, the responses are slower than for the other two cases. Although for the single-level, the responses seem to be the fastest, it can also be seen that some responses for the CO I structure outperformed the single level, even considering this criterium.

To study the control performance in more detail, further filtering is applied to choose one controller set for analysis from each approach. The full thermodynamic model is evaluated for all the controllers in each set. The objectives representing the steady-state errors are all catered for adequately by all solutions and are ignored. The solutions are now further ranked but only using a subset of the objectives a – d only, to choose the best controller using multiobjective ranking. All the performance graphs (Fig. 7 - 10) refer to the final selected controller set for each case. They indicate that the CO I controller outperforms the other two. CO II perfomance indicates slower, though stable, responses when compared to the other two. The superiority of CO I was found to hold for all the other nondominated controllers in each case.

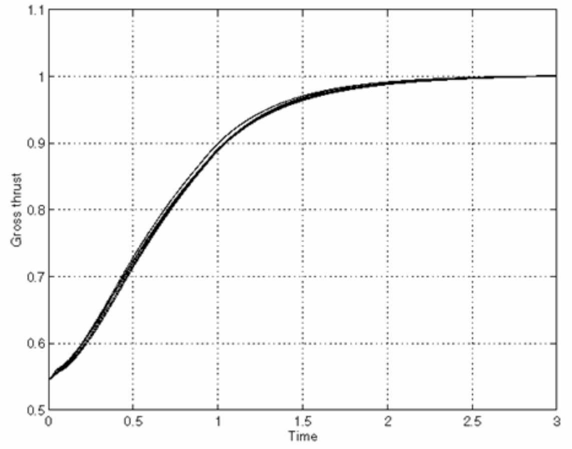

Figure 6: Thrust responses for CO II approach.

In all the response graphs, CO II appears to indicate the slowest performance indicating overdamping, but the responses are still within limits. The thrust response indicates that the single-level controller achieves slightly faster response. The transient behavior however, appears to indicate that CO I gives the most satisfactory results overall. LPSM and TBT overshoot are worst for the single-level case.

Thus, decomposing the problem into the two different types of controller (P and I) as in CO I gives better results than spliting the actual controllers along the structures corresponding to different input-output combinations as in CO II. This decomposition also outperforms the classical single-level all in one approach. The engine is one whole unit made up of various interacting sub-units. The single level approach can allow some perfomance dynamics, pertinent only to a subset of the controller parameters, to influence other controller parameters. If this is the case, it will lead to inefficiencies. CO I appears to offer a better decomposition scheme than CO II. This allows the relevant perfomance metrics to be allocated to each optimization strand in a more efficient and appropriate way.

In order to calculate the computational cost of MOGA-PI approach, it is useful to consider relative computing costs (in seconds of CPU time) of the design stages as follows:

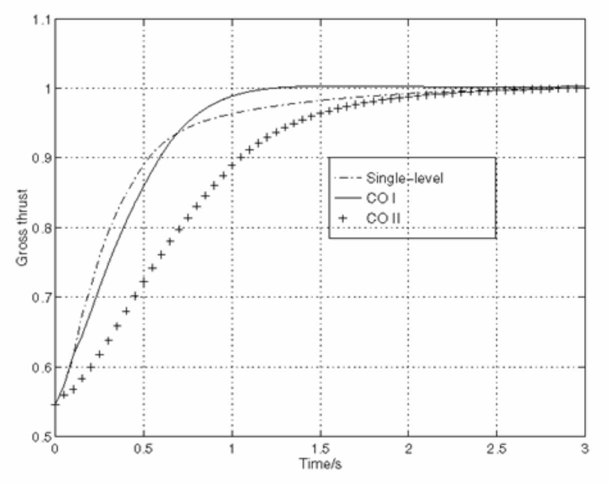

Figure 7: Thrust responses for the three design approaches.

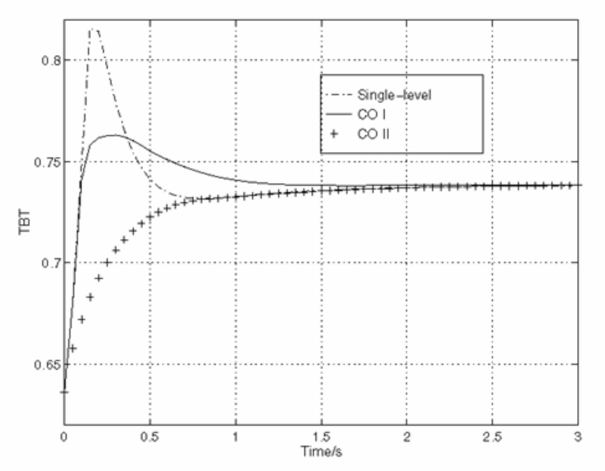

Figure 8: TBT responses for the three design approaches.

- Model evaluations

  PI stable model evaluation: a = 27.8383 s/evaluation

  PI unstable model evaluation: b = 11.4950 s/evaluation

- Optimization costs: there are five operators used in the MOGA: selection, crossover, mutation, multiobjective ranking and fitness assignment. The average cost of all of these operators is: c = 0.006167 s/individual

The optimization was carried out over 5**0** generations of **40** individuals each. Thus, the computational cost is: c = 50*40*(a + b)/2 + 50*40*c) = 39345.624 seconds

The evaluation of the models is the main cost of the overall optimization process.

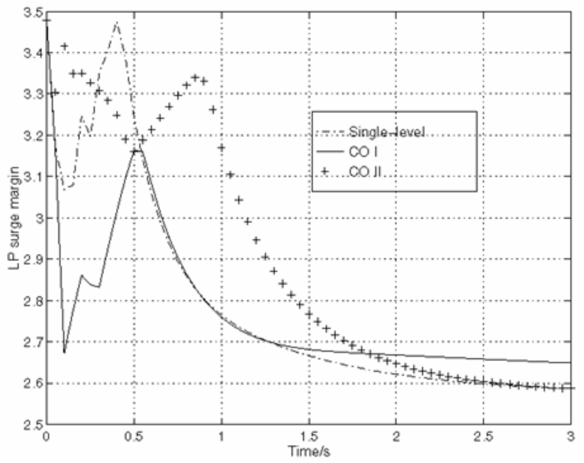

Figure 9: LPSM responses for the three design approaches.

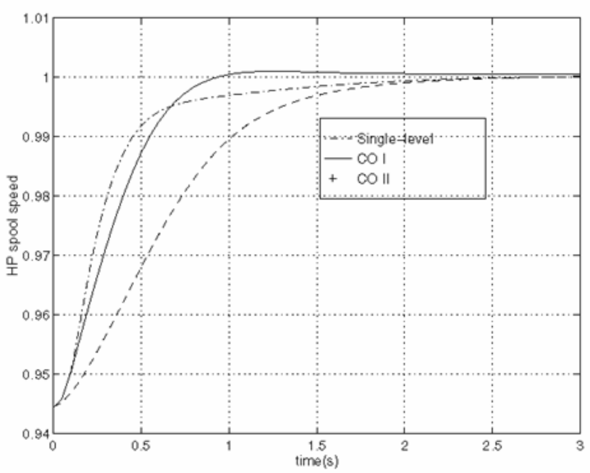

Figure 10: NH responses for the three design approaches.

8 CONCLUSIONS

Multidisciplinary optimization is used to tackle some difficult design problems involving interacting systems with strong coupling, and is traditionally based in the aircraft industry.

Complex systems tend to have strong coupling between the various elements, creating complex optimization scenarios in control design problems, and are usually multidisciplinary in nature. MDO challenges are envisaged to be present in complex control scenarios such as multivariable control.

This is the case for the optimization of the PI controllers for a GTE. The design methodology using collaborative optimization together with MOGA decomposes the problem along convenient and efficient boundaries. This allows for improvements in terms of computational cost, problem set-up and efficiency issues, and can also lead to improvements in the performance of the final designs.

Some improvements are possible using a collaborative optimization framework for PI controllers design. These improvements are reflected in the actual control performance in the engine response. The compromise solutions showed the trade-offs between the variables and can help the designer understands the interactions between the sub-systems. The final overall designs can also offer improvements over traditional single-level all-in-one approaches.

REFERENCES

Åström, K.J. and Häagglund, T. (1995), "PID controllers: Theory design and tuning", (2^{nd} Edition), Instrument Society of America.

Balling, R.J. and Sobieszczanski-Sobieski, J., (1996). "Optimization of Coupled Systems: A Critical Overview of Approaches". *AIAA Journal*, **34(1)**, pp. 6-17.

Chipperfield, A.J. and Fleming, P.J., (1996). "Systems integration using evolutionary algorithms". *International Conference on Control'96 - IEE Conference Publication n° 427*, **1**, pp. 705-710.

Fonseca, C.M. and Fleming, P.J., (1993). "Genetic Algorithms for Multiobjective Optimization: Formulation, Discussion and Generalisation". In *Genetic Algorithms: Proceedings of the Fifth International Conference* (S. Forrest, ed.), San Mateo, CA: Morgan Kaufmann.

Fonseca, C.M. and Fleming, P.J., (1995). "An overview of evolutionary algorithms in multiobjective optimization". *Evolutionary Computing*. Vol. 3, N°. 1, pp. 1-16.

Ghaffari, A., Mehrabian, A.R. and Mohammad-Zaheri, M., (2007). "Identification and control of power plant de-superheater using soft computing techniques". *Engineering Applications of Artificial Intelligence*, Vol. 20, Issue 2, pp. 273-287.

Giannakoglou, K.C., (2002), "Design of optimal aerodynamic shapes using stochastic optimization methods and computational intelligence", *Progress in Aerospace Sciences*, Vol. 38-1, pp. 43-76.

Khatib W. and Fleming, P.J., (1998). "Evolutionary Computing Applied to MDO Test Problems". *7th AIAA/USAF/NASA/ISSMO Symposium on Multidisciplinary Analysis and Optimzation,* St. Louis, MO.

Khatib W. and Fleming, P.J., (1997). "Evolutionary computing for multidisciplinary optimization". *Proc 2nd IEE/IEEE International Conference on Genetic Algorithms in Engineering Systems: Innovations and Applications GALESIA'97*, Glasgow, pp 7-12.

Silva, V.V.R., Fleming, P.J., Sugimoto, J, and Yokoyama, R., (2007). "Multiobjective optimization using variable complexity modelling for control system design". *Applied Soft Computing, In Press, Corrected Proof, Available online 18 March 2007*,

Silva, V.V.R., Khatib, W. and Fleming, P.J., "Performance optimization of gas turbine engine". *Engineering Applications of Artificial Intelligence*, Vol. 18, Issue 5, August 2005, pp. 575-583.

Silva, V.V.R. and Fleming, P.J., (2002). "Control configuration design using evolutionary computing". *Proceedings of XV Triennial IFAC World Congress.* Barcelona, Spain.

Sobieszczanski-Sobieski, J. and Haftka, R.T., (1996). "Multidisciplinary Aerospace Design Optimization: Survey of Recent Developments". *AIAA* Paper 96-0711, Reno, NV.