Proceedings of the 17th World Congress
The International Federation of Automatic Control
Seoul, Korea, July 6-11, 2008

Multiobjective Worst-Case Analysis of a Re-Entry Vehicle Control Law ⋆

P. P. Menon * **I. Postlethwaite** * **S. Bennani** ** **A. Marcos** *** **D. G. Bates** *

* *Control and Instrumentation Research Group, Department of Engineering, University of Leicester, Leicester, UK. (Email: ppm6,ixp,dgb3@le.ac.uk).*
** *ESA/ESTEC (TEC-ECN), Keplerlaan 1, 2201 AZ Noordwijk, The Netherlands. (Email: samir.bennani@esa.int).*
*** *Deimos Space, Ronda de Poniente, 19, 28760 Tres Cantos (Madrid), Spain (Tel: +34 91 806 3462; e-mail:andres.marcos@deimos-space.com).*

Abstract: This paper reports results of a joint study between ESA and the University of Leicester on worst-case analysis of NDI control laws for an industrial standard Reusable Launch Vehicle. Multiple performance objectives over a particular phase of the atmospheric re-entry are considered simultaneously in the analysis, yielding valuable information about the trade-offs involved in satisfying different clearance criteria. Two different multiobjective optimisation algorithms are employed to identify the pareto front of the multiple performance objectives. In the initial analysis, a fast, elitist, evolutionary multiobjective optimisation algorithm known as nondominated sorting genetic algorithm(NSGA-II) is employed. A hybrid multi objective optimisation algorithm which adaptively switches between three different strategies such as NSGA-II, differential evolution and the metropolis algorithm, is also developed and applied to the clearance problem. The results of our analysis show that the proposed optimisation-based approach has the potential to significantly improve both the reliability and efficiency of the flight clearance process for future re-entry vehicles.

1. INTRODUCTION

Atmospheric re-entry is a critical phase of the Reusable Launch Vehicle (RLV) mission. During re-entry flight, the space vehicle follows a predefined trajectory toward the designated landing point, traveling from the emptiness of space to the dense atmosphere of earth. As the Mach number decreases, the vehicle is exposed to rapid changes in aerodynamic flight parameters, whose values are known to control law designers only within large uncertainty ranges. Thus, the re-entry guidance and control laws need to be highly robust in tracking the pre-defined trajectory. In order to guarantee the safety of the mission, the worst-case deviations from the prescribed trajectory have to be calculated for the expected levels of variation and uncertainty in flight parameters such as aerothermodynamic parameters, mass, inertia, actuator and sensor uncertainties. As the safety of the RLV is dependent on the control laws, it must be shown that robust trajectory tracking is achieved for all normal and various failure conditions, and in the presence of all possible parameter variations.

The RLV clearance task is thus to quantify the worst perturbation from the nominal vehicle trajectory which could be generated by simultaneous variations in multiple uncertain parameters, many of which are not necessarily linear or time invariant. This is, in fact, an example of the type of robustness analysis problems which have been intensively studied in aircraft control in recent years, Fielding et al. (2002); Bates and Hagstrom (2007); Menon et al. (2006, 2007), and for which many powerful tools are now available. The classical approach to worst-case analysis usually employs analytical measures of (linear) robustness such as gain/phase margins or nonlinear continuation/bifurcation analysis against single parameter variations. More modern approaches based on robust control theory convert the given closed loop system to a linear fractional transformation (LFT)-based model and subsequently employ techniques such as μ-analysis and ν-gap metric analysis to assess the robustness of multi-loop flight control laws to multiple sources of uncertainty (chapters 17 and 18, Fielding et al. (2002)). Nonlinear extensions of these approaches using Integral Quadratic Constraints (IQC's) Biannic et al. (2006) and Sum-Of-Squares (SOS) programming, Krishnaswamy et al. (2005) have also recently been developed. A common feature of all of the above methods is that they make certain demands on the closed-loop simulation model under investigation, e.g. the generation of LFT-based models for μ-analysis. In contrast, the statistical techniques that are widely used in industry, such as Monte Carlo simulation, can be applied to any type of simulation model with a minimum of effort on the part of the designer. Monte Carlo simulation can, however, only provide statistical estimates of true worst-case behaviour, and this approach can become extremely computationally expensive if high confidence levels are required for the analysis results.

In this paper, we describe an alternative (or complementary) approach, based on the use of hybrid multiobjective optimisation algorithms, which delivers extremely reliable analysis results, while minimising both computational overheads and additional modelling requirements. Importantly, the proposed optimisation-based analysis approach is applicable to linear or nonlinear simulation models with linear, nonlinear or parameter/time varying uncertain parameters. Clearly, given that the parameter space for this type of problem will in general be highly nonlinear and non-convex, Fielding et al. (2002), global optimisation methods will be required to avoid getting trapped in locally optimal solutions. Previous work by the authors has explored the applicability of various evolutionary, Menon et al. (2006) and deterministic, Menon et al. (2006(b), global optimisation algorithms to the flight clearance problem, Bates and Hagstrom (2007), and has also demonstrated the usefulness of hybridising global optimisation algorithms with local gradient-based methods such as SQP for enhancing the rate of convergence to the global solution. In the specific context of the re-entry vehicle clearance problem, novel optimisation-based approaches to the flight clearance problem were described in

⋆ This work was supported by the European Space Agency under Research Contract No. 19784/06/NL/JD/na and by an EPSRC Platform Grant No. EP/D029937/1 The authors gratefully acknowledge the contribution of DEIMOS Space S.L. in providing the RLV simulation model and control law used in this study.

Table 1. Uncertain Parameters in RLV Model

Parameter	Bound	Description
Δ_{mass}	[-2313.3, 2313.3]	variation in RLV dry mass from nominal one (11566.55 kg)
Δ_{Ixx}	[-1627, +1627]	variation in Moment of inertia about Y (8135.0 $4kgm^2$)
Δ_{Iyy}	[-15185, +15185]	variation in Moment of inertia about Y (75926.0 kgm^2)
Δ_{Izz}	[-15863, +15863]	variation in Moment of inertia about Z (79315.0 kgm^2)
Δ_{Ixz}	[-628.8, +628.8]	variation in Product of inertia XZ (3144.0 kgm^2
Δ_{xcog}	[-0.4912, +0.4912]	variation in X center of gravity from nominal one (4.9213 m)
Δ_{ycog}	[-0.01, +0.01]	variation in Y center of gravity from nominal one (0.0 m)
Δ_{zcog}	[-0.1009, +0.1009]	variation in X centre of gravity from nominal one (1.0094976 m)

Menon et al. (2007). In all of the above studies, however, only worst-case analysis problems involving a single performance objective were considered.

In actual RLV mission scenarios, it is often necessary to have an estimate of the worst-case performance of the control laws with respect to multiple performance objectives or clearance criteria, in order that the designers can understand the performance trade-offs that exist between different clearance criteria. In such situations, the flight clearance problem can clearly be considered from the perspective of multiobjective optimisation. The re-entry vehicle flight envelope is described along a designated nominal flight path, which is defined in terms of desired trajectories for several different states of the vehicle. The safety of the control law is checked by considering separate phases of the re-entry, using some pre-defined stability and performance criteria.

In this paper, we consider simultaneously multiple conflicting performance criteria and estimate the worst-case performance of the NDI re-entry control laws designed for the RLV mission. The focus is thus to identify the set of uncertain parameters corresponding to the Pareto-optimal front, which results in worst-case performance of the control law. Multiobjective optimisation algorithms are employed to identify the nondominated solution set of uncertain parameters that generates the worst-case performance. In the initial analysis, a fast, elitist, evolutionary multiobjective optimisation algorithm known as nondominated sorting genetic algorithm(NSGA-II), Deb et al. (2002), is employed. Motivated by the previous improved results from the hybridisation of optimisation techniques reported in Menon et al. (2006); Bates and Hagstrom (2007), a hybrid algorithm which adaptively switches between three different strategies such as NSGA-II, differential evolution and the metropolis algorithm is also developed and applied to the clearance problem. While the use of evolutionary multiobjective optimisation techniques in the context of worst-case analysis of control laws is, to our knowledge, completely novel, we note that these techniques have previously been successfully applied to different design problems in RLV mission design, such as computing the optimal re-entry trajectory, Gang et al. (2005); Arora (2002).

2. RLV MODEL AND FLIGHT CONTROL LAW

The high-fidelity simulation model of a generic RLV considered in this study is based on the HL-20 aerodynamic database and X38-type geometric and aerodynamic surface configurations, and has a dry mass of 19,100-lb. This simulation model has been developed by DEIMOS Space for the European Space Agency (ESA) to act as a research platform for the investigation of re-entry and auto land guidance, navigation and control systems, Fernandez et al. (2006). The model consists of a reference trajectory generator, a nonlinear dynamic inversion (NDI)-based flight control system, nonlinear actuator models, the RLV dynamics, sensors such as gyros and accelerometers, and detailed environment models. The RLV simulation model is implemented in the Matlab Simulink environment.

The reference trajectory is defined in terms of Angle of Attack (AoA or α), Angle of Side Slip (AoSS or β), and bank angle ϕ. The NDI controller provides the elevator, aileron, rudder and brake control inputs according to the specified dynamics for the vehicle. The controller also includes actuator allocation functions depending on the commanded moments, altitude and velocity of the RLV. More details of the model and its associated flight control system are available in Fernandez et al. (2006). The parameters in the model, and associated uncertainty values, are accessible through a database consisting of a collection of XML files.

The complete re-entry trajectory for the vehicle takes 1680 seconds of simulation time and is divided into different flight phases based on dynamic pressure and atmospheric layer. The present analysis focuses on a lower atmosphere flight phase starting at 1588 seconds and ending at 1675 seconds that covers the 32 to 20 km altitude range. The reference trajectory in this segment includes a reduction of AoA from 30 degrees to nearly 20 degrees, while keeping a zero AoSS and with a defined bank angle variation.

The uncertainties considered in the present analysis are the parameters representing the vehicle's mass, inertias, centre-of-gravity and aerodynamic coefficients (longitudinal and lateral). The individual description and allowed ranges of the configuration and inertia uncertain parameters considered for the present analysis are given in Table 1. The longitudinal aerodynamic uncertainties considered are coefficient of Lift (Δ_{C_L}), coefficient of Drag (Δ_{C_D}), coefficient of pitching moment (Δ_{C_m}) and coefficient of Lift to Drag ratio ($\Delta_{L/D}$). These are all considered as non-dimensional. For example, in the model the total coefficient of Drag is modelled as $C_{D_{nom}} * (1 + \Delta_{C_D})$. The lateral aerodynamic uncertainties considered are uncertainties in rolling moment coefficient (Δ_{C_l}), yawing moment coefficient (Δ_{C_n}) and sideforce coefficent (Δ_{C_y}). Based on current industrial practice, all longitudinal and lateral aerodynamic uncertainties are normalised to unity in this study and vary with respect to Mach - these parameters are thus time varying parameters Fernandez et al. (2006).

2.1 Multiobjective worst-case analysis

In the industrial flight clearance process, it is of significant interest to control law designers to identify the set of uncertain parameters that will provide worst-case performance as measured by different, often contradictory, clearance objectives. An example of such a clearance problem is addressed in this paper within the proposed multiobjective optimisation framework. The aim is to identify the pareto-optimal front for the maximum deviations of angle of sideslip (β) and angle of attack (α) over the **SIM PH5ALY2** phase of the RLV mission trajectory. The multi-objective clearance criterion is thus defined as follows:

$$\max\ J1, J2 \quad \text{where} \tag{1}$$

$$J1 = \|\alpha_{ref} - \alpha_{\Delta}\|_{\infty}, \tag{2}$$

$$J2 = \|\beta_{ref} - \beta_{\Delta}\|_{\infty} \tag{3}$$

$$\text{subject to: } \underline{\Delta} \leq \Delta \leq \overline{\Delta} \tag{4}$$

where α_{ref} and β_{ref} are the reference AoA and AoSS trajectory. α_Δ and β_Δ represent the AoA and AoSS trajectories of the RLV subject to the multiple parameter uncertainty perturbation vector Δ. The upper and lower bounds of the uncertain parameter vector are given by $\overline{\Delta}$ and $\underline{\Delta}$. Equation (4) provides the constraints on the values of the uncertain parameters, as given in Table (1).

3. MULTIOBJECTIVE OPTIMISATION

A general formulation of the multiobjective optimisation problem can be described as follows. Consider

$$F(x) = \{F_1(x), F_2(x), ..., F_k(x)\},\ x \in S \tag{5}$$

where $F_i(x)(i = 1,2,...,k)$ is a scalar objective function which maps decision variable x into the objective space, $F_i :\ \mathbb{R}^n \mapsto \mathbb{R}$. The n-dimensional variable x is constrained to lie in the feasible region S. The feasibility region S is constrained by m inequality and p equality constraints, i.e.

$$S = \{x :\ g_j(x) \geq 0,\ h_l(x) = 0,\ j = 1,2,...,m,\ l = 1,2,...,p\} \tag{6}$$

In multiobjective optimisation problems, the desired objectives can often be conflicting, i.e. it is not necessarily possible to satisfy all the objectives simultaneously. In such cases, *Pareto optimal* solutions, which are a set of non-inferior solutions, are computed. Consider two candidate solution vectors $x = (x_1, x_2, ..., x_n)$ and $y = (y_1, y_2, ..., y_n)$. The pareto concept may be explained as follows: The vector x dominates vector y if and only if, $x_i \leq y_i\ \forall i = 1,2,...,n$ and $x_i \leq y_i$ for at least one i. Hence, when comparing the two candidates x and y there are three possibilities: x dominates y, x is dominated by y or x and y are nondominated. This is the *Pareto Dominance* property which is used to define the *Pareto Optimal* points. The solution vector x is Pareto optimal if and only if there does not exist another solution y such that $f_i(y)$ dominates $f_i(x)$ for every $i = 1,2,...,k$. The resulting solution is one in which one objective can not be further improved without the degradation of at least one another objective. The resulting set of solutions thus represents the globally optimal solutions to the tradeoff problem defined by the given conflicting objectives. The set of all Pareto optimal solutions is called the *Pareto optimal set* and is denoted as PO^*. The set $PF^* = \{[f_1(x), f_2(x), ..., f_k(x)]^T | x \in PO^*\}$, is called the *Pareto front*.

The objectives of the multiobjective optimisation are: (i) to reduce the distance between the nondominated front and the Pareto optimal front, (ii) to achieve preferably a good (in general uniform) distribution of the solutions and (iii) to have a maximum spread of the obtained nondominated front, implying the coverage of a wide range of values for each objective by the nondominated solutions. There are thus two key issues in solving multiobjective problems. Firstly, accomplishing the fitness assignment and selection, respectively, such that the search is guided toward the Pareto-optimal set. This can be achieved by, for example, selection by switching objectives (Vector Evaluated Genetic Algorithm reported in Schaffer (1985)), aggregation selection along with parameter variation (weighting-based genetic algorithm reported in Hajela and Lin (1992)), pareto-based selection (multiobjective genetic algorithm reported in Fonseca and Fleming (1998a,b) or by using the nondominated sorting genetic algorithm (NSGA-II) reported in Deb et al. (2002)). The subsequent issue is to maintain a diverse population to prevent the premature convergence and achieve a well distributed nondominated set. Maintaining diversity among the candidate solutions is addressed by different methods such as fitness sharing, restricted mutation, isolation by distance, reinitialization and crowding. Following a review of the applicability of the many different algorithms which may be found in the literature, we chose the NSGA-II multiobjective optimisation method as the starting point for our study and subsequently developed a more advanced hybrid algorithm for application to our multiobjective flight clearance problem.

Table 2. Nondominated sorting genetic algorithm

(1) Generate a random initial population P_0 of size N, evaluate the objectives
(2) Rank and sort the P_0 based on non-domination level
(3) For initial iteration, create the offspring population Q_0 of size N, using binary tournament selection, crossover and mutation operators.
(4) **While** the Termination criteria is not satisfied (e.g. Maximum iteration fixed at say 25)
  (a) Generate the combined population R_{gen} including the population P_{gen} and offspring Q_{gen}; $[R_{gen} = P_{gen} \cup Q_{gen}]$
  (b) Sort the combined population R_{gen} into a set of nondominated fronts $\mathscr{F} = (\mathscr{F}_1, \mathscr{F}_2, ...)$
  (c) $P_{gen+1} = \{\mathbf{0}\}$ and $j = 1$
  (d) **Do**
    (i) Calculate the crowding distance in $\mathscr{F}_j$
    (ii) Associate the j^{th} nondominated front $\mathscr{F}_j$ to the new generation population P_{gen+1}; $[P_{gen+1} = P_{gen+1} \cup \mathscr{F}_j]$
    (iii) Increment the counter $j = j + 1$
    (iv) Check subsequent front $\mathscr{F}_j$ for including in the population
    (v) Sort the $\mathscr{F}_j$ in ascending order and choose the first $N - |P_{gen+1}|$ elements from the front. $[P_{gen+1} = P_{gen+1} \cup \mathscr{F}_j(1 : (N - |P_{gen+1}|))]$
  (e) **While** the new generation population is filled with N candidates; $[|P_{gen+1}| + |\mathscr{F}_i| \leq N]$
  (f) Generate offspring Q_{gen+1} of size N, using binary tournament selection, crossover and mutation operators.
  (g) Increment the generation counter $gen = gen + 1$
(5) end of **While**

4. NONDOMINATED SORTING GENETIC ALGORITHM

The nondominated sorting genetic algorithm (NSGA-II) proposed by Deb et al. (2002) differs from simple GA's mainly in the way the selection operator is used. Fast nondominated sorting, crowded distance calculation and the crowded comparison operator are the basic building blocks of the NSGA-II. The algorithm starts with a randomly generated initial population P_0. The initial population is the set of candidate solution vectors consisting of various randomly generated uncertain parameters within the defined bounds. The multiple objectives of the candidate solutions in the population P_0 are evaluated. The initial population is sorted according to non-domination into different fronts $\mathscr{F}_1, \mathscr{F}_2, ...$ The first front $\mathscr{F}_1$ includes all the candidate solutions that are completely nondominant in the current population. The second front $\mathscr{F}_2$ is dominated by individuals in $\mathscr{F}_1$ and so on and so forth. Every candidate solution in the front is assigned a rank, according to which front they belong to. The $\mathscr{F}_1$ members have rank 1 and $\mathscr{F}_2$ with rank 2 and so on. An important point to note here is that the assumption followed in NSGA is that the problem involves the minimisation of multiple objectives, and hence the maximisation problems obtained in worst-case analysis must be suitably transformed into minimisation problems before the optimisation algorithm is applied. Initially, the offspring population Q_0 is created using the binary tournament selection, crossover and mutation. Also, note that elitism is ensured by mixing the offspring population Q_0 with the current generation population P_0. Subsequently further nondomination level sorting and crowding distance computation is done; the selection of the subsequent population is performed and the algorithm is iterated until the termination criteria is satisfied. At present a termination criteria of a fixed computational budget is used. More details on the algorithm are available in Deb et al. (2002) and the references therein. The pseudo code for the NSGA-II algorithm is given in Table (2). The nondomiianted sorting, crowding distance assignment and selection are discussed in more detail in the following subsections.

4.1 Nondominated sorting

A candidate solution vector p can be associated with a domination count n_p and a domination set S_p. n_p is the number of solutions in the population dominating the candidate solution p. S_p consists of the set of solutions that p dominates. The sorting procedure begins by assigning n_p zero to all the candidates in the population. Initially S_p is empty. For each individual q in the population, if p dominates q then the candidate q is included in S_p ($S_p = S_p \cup q$). Otherwise, n_p is incremented by one. While $n_p = 0$, no individual dominates candidate p and the candidate p belongs to the first front $\mathscr{F}_1$, having rank 1 ($\mathscr{F}_1 = \mathscr{F}_1 \cup p$). This procedure is carried out for all the candidate solutions in the population. For any member q having $n_p = 0$, a separate list Q is generated. This belongs to the second nondominated front. The procedure is continued with each member in Q and the subsequent front is identified and so on.

4.2 Crowding distance assignment

The population is sorted according to each objective function value in ascending order of magnitude. Then, for each objective function the boundary solutions are assigned an infinite distance value. All other intermediate solutions are assigned the distance value equal to the normalised absolute difference in the objective function values of the two adjacent solutions in the sorted set. The overall crowding distance is calculated as the sum of individual distance values corresponding to each objective function value, which are normalised prior to calculation. The basic idea behind the crowding distance calculation is to calculate the euclidian distance between each individual in the sorted front $\mathscr{F}$ based on their n objectives in the m dimensional space. The individuals on the boundary are always selected since they have been assigned an infinite distance value.

4.3 Crowded comparison operator for selection

Once the candidate solutions are sorted based on non-domination and the crowding distance assigned, selection is carried out by employing the crowded-comparison operator ($\prec_n$). This operator ensures a uniformly spread-out pareto-optimal front. Every candidate solution in the population has: (i) nondomination rank (p_{rank}) calculated in the nondominated sorting step, i.e the candidate solution in front $\mathscr{F}_j$ will have the rank as $p_{rank} = j$. (ii)the crowding distance ($p_{distance}$). For two candidate solutions with differing nondomination ranks, the candidate having lower rank is selected. When the ranks of two candidates are equal, i.e of the same front, the solution located in a lesser crowded region (larger crowding distance value) is selected.

5. HYBRID MULTIOBJECTIVE OPTIMISATION

An adaptive hybrid search algorithm for multiobjective optimisation problems was recently proposed in Vrugt and Robinson (2007). While this algorithm achieved both improved performance *and* reduced computational overheads on a number of challenging academic test problems, it has so far not been applied to any serious realistic applications. In this paper, we apply the adaptive hybrid approach of Vrugt and Robinson (2007) using three algorithms: NSGA-II, differential evolution(DE) and metropolis search(AM). Like NSGA-II, this algorithm also begins with N randomly generated individuals in an initial population P_0. The multiple objectives of individuals in the population P_0 are evaluated. Once the objectives are evaluated each candidate in the population is assigned a rank using the nondominated sorting algorithm.

Initially, the offspring population Q_0 is created by k different algorithms$\{Q_0^1, Q_0^2, ..., Q_0^k\}$, i.e. NSGA-II (Deb et al. (2002)), DE (Stron and Price (1997)), and MH (Haario et al. (2001)) in the present case, each contributing equally $\frac{N}{k}$ members. Initially, k different algorithms generate equal numbers of offsprings such that the total population size is N. Subsequently as the algorithm is iterated, the k different algorithms create a prespecified number of offspring points, $N = \{N_{gen}^1, ..., N_{gen}^k\}$, that are determined by the adaptive procedures and gen representing the particular iteration or generation.

Table 3. Hybrid multiobjective optimisation

(1) Generate a random initial population P_0 of size N, evaluate the objectives
(2) Rank and sort the P_0 based on non-domination level
(3) Select k different optimisation algorithms from the pool of optimisers
(4) Create the offspring population Q_0, k different algorithms contribute equally to generate the offspring population size N
(5) **While** the Termination criteria is not satisfied (e.g. Maximum iteration fixed at say 25)
  (a) Generate the combined population R_{gen} including the population P_{gen} and offspring Q_{gen}; $[R_{gen} = P_{gen} \cup Q_{gen}]$
  (b) Sort the combined population R_{gen} into a set of nondominated fronts $\mathscr{F} = (\mathscr{F}_1, \mathscr{F}_2, ...)$
  (c) $P_{gen+1} = \{\mathbf{0}\}$ and $j = 1$
  (d) **Do**
    (i) Calculate the crowding distance in $\mathscr{F}_j$
    (ii) Associate the j^{th} nondominated front $\mathscr{F}_j$ to the new generation population P_{gen+1}; $[P_{gen+1} = P_{gen+1} \cup \mathscr{F}_j]$
    (iii) Increment the counter $j = j + 1$
    (iv) Check subsequent front $\mathscr{F}_j$ for including in the population
    (v) sort the $\mathscr{F}_j$ in ascending order and choose the first $N - |P_{gen+1}|$ elements from the front. $[P_{gen+1} = P_{gen+1} \cup \mathscr{F}_j(1 : (N - |P_{gen+1}|)]$
  (e) **While** the new generation population is filled with N candidates; $|P_{gen+1}| + |\mathscr{F}_i| \leq N$
  (f) Estimate the contribution to P_{gen+1} by each of the k optimisation algorithms
  (g) Calculate the number of offspring to be generated by k different algorithms N_{gen+1}^k by adaptive weighting scheme that depend on the contribution of the individual algorithms to the present generation population
  (h) Rescale the weights in case the number of offspring corresponding to any algorithm is below the atleast threshold value, say fixed at 2 [This ensures the existence of the algorithms]
  (i) Generate offspring Q_{gen+1} of size N by producing N_{gen+1}^k offspring points with each k optimisation algorithms
  (j) Increment the generation counter $gen = gen + 1$
(6) end of **While**

The philosophy behind the adaptation strategy is to weight more those individual algorithms that contribute more to the selected new population of the next generation. The update of $N = \{N_{gen}^1, ..., N_{gen}^k\}$ according to

$$N_{gen}^i = N * (P_{gen}^i / N_{gen-1}^i) / \sum_{i=1}^{k} (P_{gen}^i / N_{gen-1}^i)$$

P_{gen}^i is the number of offspring points the i^{th} individual algorithm contributed to the new population. N_{gen-1}^i is the number of offsprings the i^{th} individual algorithm contributed in the population of the previous generation. This provides an adaptive weighting between the individual algorithms. A minimum threshold between the algorithms is provided and is fixed at 2. This avoid the possibility of an algorithm becoming totally inactive during the adaptation and thereby becoming eliminated from contributing to future generations.

A combined population of size $2N$ is generated by mixing the offspring population with the initial population $R_0 = P_0 \cap Q_0$. The mixing of the population is aimed to ensure the elitism. The R_0 is ranked using the nondominated sorting method. Hence, the nondominated solutions in the previous population will also be included in the new nondomianted front along with those contributed by the new offspring population. A crowding

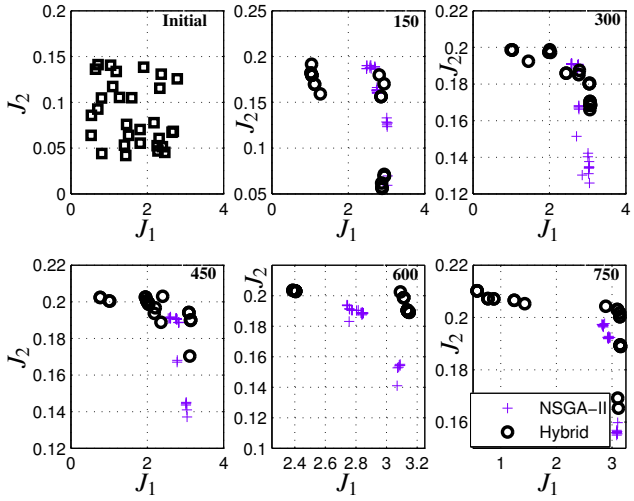

Fig. 1. Multiobjective result of RLV

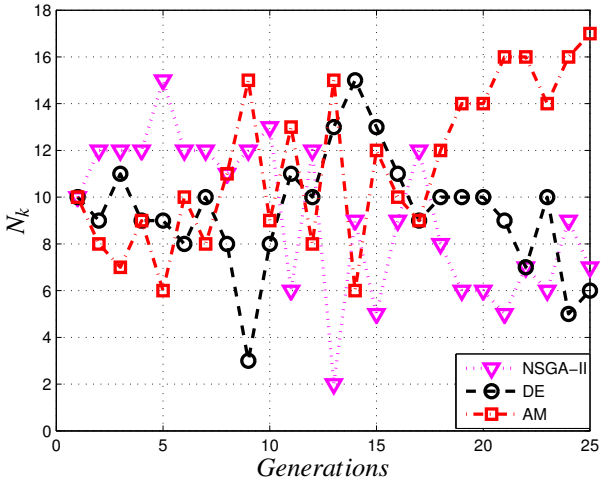

Fig. 2. Adaptation of algorithms

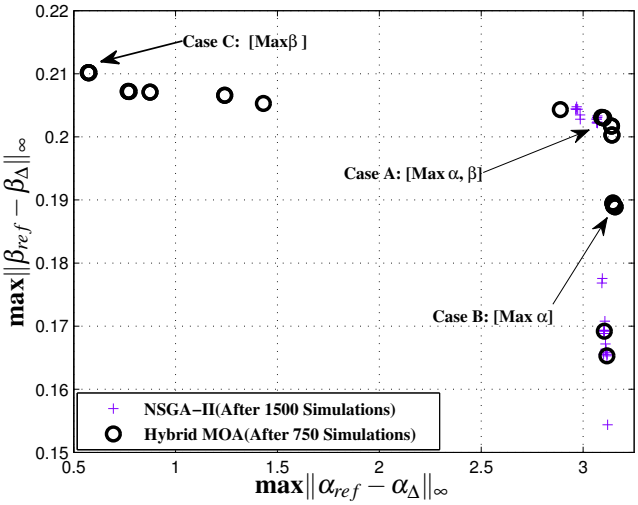

Fig. 3. Multiobjective result of RLV

distance is also assigned with the nondominated members. The new population is selected by considering the ranking and crowding distance as in NSGA-II. The new population P_1 is subsequently used to create the offsprings. The pseudo code of the hybrid multiobjective optimisation algorithm is given in Table 3.

6. WORST-CASE ANALYSIS RESULTS

First, the NSGA-II algorithm as given in Table 2 is applied to evaluate the pareto front for the multiobjective worst-case performance problem defined in equations (1 -4) in Section II. The NSGA-II algorithm begins with an initial population size of 30. The initial population of 30 candidate solution vectors, is randomly generated over the bounded search space. The multiple objectives J_1 and J_2 corresponding to each candidate solution uncertain parameter vector are evaluated by simulating the RLV simulation model discussed in Section II. The NSGA-II algorithm, which consists of non-dominating sort, crowding distance calculation and genetic operators such as binary tournament selection, crossover and mutation operators, is iterated until the termination criteria is satisfied. A predefined computational budget, fixed at 25 generations, is employed as the termination criteria. The points marked $'+'$ in figure 1 represents the pareto front obtained by the NSGA-II algorithm. The first subfigure of figure (1) shows the objective function values for the initial population. The subplots in figure 1 show the evolution of the pareto front after every 150 simulations.

To investigate the potential for improvement in the performance of the NSGA-II algorithm, we also applied the hybrid multiobjective optimisation algorithm as given in Table 3 to identify the pareto front for the worst-case performance problem. To ensure a fair comparison, the hybrid multiobjective optimisation is supplied with the same initial population as in the case of the NSGA-II trial. The evaluation of the objective function and the termination criterion used for the hybrid multiobjective optimisation are identical to that of NSGA-II. However, for this algorithm, the intermediate candidate solution vectors (offspring population) are generated using different algorithms adaptively to enhance the convergence to the pareto front. The adaptation of the different algorithms over the generations is as shown in Figure 2. For example, at the 10^{th} generation, the number of intermediate candidate solutions (offspring population) generated by NSGA-II, DE, and AM algorithms are 13, 8 and 9 respectively. Note that, at the beginning, the populations generated by the different algorithms are equal, fixed at 10.

During the initial part of the optimisation, the NSGA-II algorithm exhibits the highest reproductive success, owing to the proficiency of classical genetic operators in quickly identifying the global front as shown in figure 2. In the middle stages of the optimisation the other two algorithms have more success, the DE in particular. After the 17^{th} generation the utility of the NSGA-II and DE algorithms are reduced. As is clear from Figures 1 and 3, the adaptive combination of different algorithms is highly effective in increasing the diversity of solutions along the pareto front once the NSGA-II algorithm has successfully identified the front over the initial stages of the optimisation.

The evolution of the pareto optimal front is as shown by the points marked in 'o' in Figure 1 at intervals of 150 simulations. It can be seen from Figure 1 that at the end of 150 simulations, the performance of both the algorithms is nearly the same, however the spread of the solution is much greater in the case of the hybrid algorithm. After 600 and 750 simulations it can be clearly seen that the hybrid multiobjective optimisation is outperforming the NSGA-II algorithm alone, even when the NSGA-II algorithm is allowed to run to double the computational budget. In Figure 3, the final results obtained by the NSGA-II are compared with those obtained using the hybrid algorithm, and the improved efficiency and performance of the hybrid algorithm are clearly revealed. The results marked as

Case A comprise the solution providing maximum deviation of $\alpha(t)$ and $\beta(t)$ simultaneously, obtained from the hybrid multiobjective optimisation. Cases B and C in Figure 3 show the solutions providing individual maxima of $\alpha(t)$ and $\beta(t)$, also obtained from hybrid multiobjective optimisation. These results clearly demonstrates the potential of the proposed multiobjective optimisation analysis approach to reveal the inherent trade-offs the apply in complex flight clearance problems. Time-domain simulation results demonstrating the simultaneous maximum deviation of $\alpha(t)$ and $\beta(t)$ from their specified trajectories are shown in Figure 4. Similarly, Figure 5 shows the simulation results corresponding to the cases B and C. The maximum deviations from the reference trajectory in each case are marked in the figures. Note that in case C, which is the maximum deviation of $\beta(t)$, the $\alpha(t)$ deviation from the reference trajectory is minimal. However in case B, which corresponds to the maximum deviation in $\alpha(t)$, the associated $\beta(t)$ deviations are also large, implying significant cross-coupling in that direction.

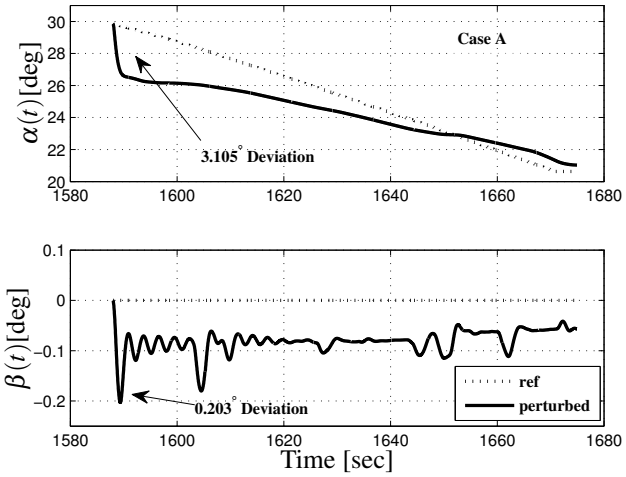

Fig. 4. Simulation results - Case A

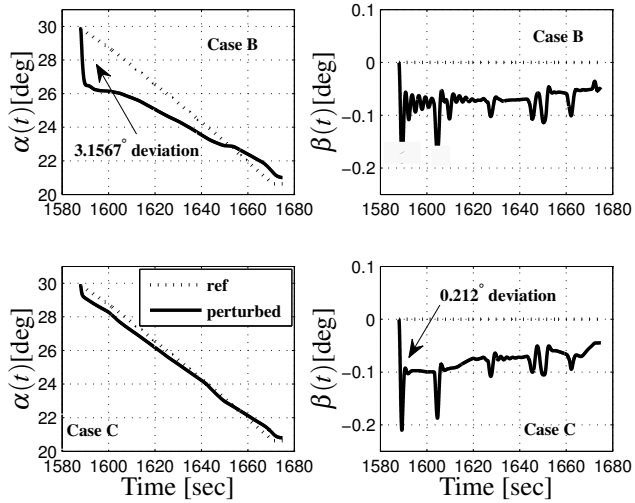

Fig. 5. Simulation results - Case B and C

7. CONCLUSIONS

Results of a joint study between ESA and the University of Leicester on worst-case analysis of NDI control laws for an industrial standard Reusable Launch Vehicle were presented. Multiple conflicting performance objectives over a particular phase of the atmospheric re-entry were considered simultaneously in the analysis, yielding valuable information about the trade-offs involved in satisfying trajectory tracking requirements on different states of the vehicle. Two different multiobjective optimisation algorithms were employed to identify the pareto front corresponding to the worst-case. In the initial analysis, a fast, elitist, evolutionary multiobjective optimisation algorithm known as nondominated sorting genetic algorithm(NSGA-II) was employed. A hybrid adaptive algorithm consisting of three different strategies such as NSGA-II, differential evolution and the metropolis algorithm was also developed and applied to the clearance problem. The results of our study show that the hybrid multiobjective optimisation-based approach in particular has the potential to significantly improve both the reliability and efficiency of the flight clearance process for future re-entry vehicles.

REFERENCES

R. Arora. Reentry trajectory optimization: evolutionary approach. *9th AIAA/ISSMO Symposium on Multidisciplinary Analysis and Optimization*, AIAA-2002-5466 , Atlanta, Georgia, September, 2002.

D.G. Bates and M. Hagstrom (Eds.). *Nonlinear Analysis and Synthesis Techniques in Aircraft Control*. Springer, 2007.

J.M. Biannic, C. Roose A. Knauf Design and robustness analysis of fighter aircraft flight control laws. *European Journal of Control*, 12(1), pp.71–85, 2006

K. Deb, A. Pratap, S. Agarwal, and T. Meyarivan. A fast and elitist multiobjective genetic algorithm: NSGA-II. *IEEE Transactions on Evolutionary Computation*, VOL. 6, NO. 2, APRIL 2002, pp. 182–197.

V. Fernandez, L.F. Penin, and A Caramagno. FDI test bench software user manual, *Report FDITB-DME-SUM, Version 1.1*, 18 September 2006.

C. Fielding, A. Varga, S. Bennani, and M. Selier (Eds.). *Advanced Techniques for Clearance of Flight Control Laws*. Springer, 2002.

C.M. Fonseca and P.J. Fleming. Multiobjective optimization and multiple constraint handling with evolutionary algorithms part i: A unified formulation. *IEEE Transactions on Systems, Man, and Cybernetics* Vol. 28(1), pp. 26-37, 1998.

C.M. Fonseca and P.J. Fleming. Multiobjective optimization and multiple constraint handling with evolutionary algorithms part ii: Application example. *IEEE Transactions on Systems, Man, and Cybernetics* Vol. 28(1), pp. 38-47, 1998.

L.S. Forssell. Flight clearance analysis using global nonlinear optimisation-based search algorithms. *AIAA Guidance, Navigation and Control Conference*, Austin, 2003.

C. Gang, X. Min, W. Zi-ming and C. Si-lu. RLV reentry trajectory multiobjective optimization design based on NSGA-II algorithm. *AIAA Atmospheric Flight Mechanics Conference and Exhibit*, San Francisco, August, 2005.

H. Haario, E. Saksman and J Tamminen. An adaptive Metropolis algorithm, *Bernoulli*, Vol. 7, No. 2, 2001.

P. Hajela and C.Y. Lin. Genetic search strategies in multicriterion optimal design. *Structural Optimization* 4, pp. 99-107, 1992.

K. Krishnaswamy, G. Papageorgiou, S. Glavaski, and A. Papachristodoulou. Analysis of aircraft pitch axis stability augmentation system using sum of squares optimization. *Proceedings of American Control Conference*, pp. 2701–2702, 2005

P.P. Menon, J. Kim, D.G. Bates and I. Postelthwaite. Clearance of nonlinear flight control laws using hybrid evolutionary optimisation. *IEEE Transactions on Evolutionary Computation*, Vol. 10, No. 6, 2006, pp. 689–699.

P.P. Menon, D.G. Bates, I. Postelthwaite. Nonlinear robustness analysis of flight control laws for highly augmented aircraft. *IFAC Journal of Control Engineering Practice*, Vol. 15, 2007, pp. 655-662.

P. P. Menon, D. G. Bates and I. Postlethwaite. A deterministic hybrid optimisation algorithm for nonlinear flight control systems analysis *Proceedings of American Control Conference* 2006.

J.D. Schaffer. Multiple objective optimization with vector evaluated genetic algorithms. *Proceedings of an International Conference on Genetic Algorithms and Their Applications*, Pittsburgh, PA, pp. 93-100, 1985.

R. Storn and K Price. Differential evolution - a simple and efficient heuristic for global optimization over continuous spaces, *Journal of Global optimisation*, Vol. 11, pp.341–359, 1997.

J.A. Vrugt and B.A. Robinson. Improved evolutionary optimisation from genetically adaptive multimethod search. *PNAS*, Vol. 104, No. 3, pp. 708–711, January, 2007.