Worst Case Analysis of Control Law for Re-entry Vehicles Using Hybrid Differential Evolution

P.P. Menon[1], D.G. Bates[1], I. Postlethwaite[1], A. Marcos[2], V. Fernandez[2], and S. Bennani[3]

[1] Control and Instrumentation Research Group, Department of Engineering, University of Leicester, Leicester, LE1 7RH, UK
[2] Advanced Projects Division, DEIMOS-SPACE S.L., Madrid, Spain
[3] Guidance Navigation and Control Group, ESA/ESTEC, Noordwijk, The Netherlands
ppm6@le.ac.uk

Summary. The development and application of the differential evolution (DE) optimisation algorithm to the problem of worst-case analysis of nonlinear control laws for hypersonic re-entry vehicles is described. The algorithm is applied to the problem of evaluating a proposed nonlinear handling qualities clearance criterion for a detailed simulation model of a hypersonic re-entry vehicle (also known as a reusable launch vehicle (RLV)) having a full-authority nonlinear dynamic inversion (NDI) flight control law. A hybrid version of the differential evolution algorithm, incorporating local gradient-based optimisation, is also developed and evaluated. Comparisons of computational complexity and global convergence properties reveal the significant benefits which may be obtained through hybridisation of the standard differential evolution algorithm. The proposed optimisation-based approach to worst-case analysis is shown to have significant potential for improving both the reliability and efficiency of the flight clearance process for next generation RLV's.

1 Introduction

Atmospheric re-entry is an important and safety-critical part of the reusable launch vehicle mission. During the re-entry flight phase, the space vehicle follows a predefined trajectory towards the designated landing point, travelling from space to the dense atmosphere of earth. As a result, the vehicle is subjected to high levels of uncertainty and variations in key flight parameters during the course of its mission. A primary requirement for re-entry guidance and flight control laws is that they exhibit sufficient levels of robustness to allow close tracking of the pre-defined trajectory in spite of high levels of uncertainty and disturbances. In order to demonstrate that this requirement is satisfied, maximum deviations from the prescribed trajectory due to uncertainty in flight parameters such as mass, centre-of-gravity locations, inertias and aerothermodynamic parameters, as well as actuator and sensor uncertainties need to be precisely evaluated in simulation , prior to any test flight. This process of "flight clearance must be carried out in all normal and various failure conditions, and in the presence of all possible parameter variations.

U.K. Chakraborty (Ed.): Advances in Differential Evolution, SCI 143, pp. 319–333, 2008.
springerlink.com

The task of analysing and quantifying the robustness properties of the RLV flight control algorithms is a very lengthy and expensive one, where different combinations of large numbers of uncertain parameters must be investigated such that an estimate about the worst case stability and performance of the control laws can be made. For nonlinear flight clearance problems, the current industrial standard is to use a gridding approach, where either the clearance criteria are evaluated for all combinations of the extreme points of the vehicle's uncertain parameters or Monte-Carlo simulation is employed to randomly sample the parameter space, [1]. Unfortunately, the computational effort involved in the resulting clearance assessment increases exponentially with the number of uncertain parameters that are to be considered (combinations of extreme points) or with the desired confidence levels for the clearance results (Monte-Carlo simulation). Another difficulty with these approaches is the fact that there is no guarantee that the worst case uncertainty combination has in fact been found, since it is possible that the worst-case combination of uncertain parameters does not lie on the extreme points, or in the sampled set used by Monte-Carlo approaches. A promising approach to address the above difficulties is to use advanced optimisation algorithms to search the parameter space for worst-cases that violate the particular clearance criterion under investigation. Clearly, given that the parameter space for this type of problem will in general be highly nonlinear and non-convex, [6], 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 optimisation methods to the flight clearance problem for high-performance aircraft, and has shown that, when hybridised with appropriate gradient-based algorithms, they have the potential to improve significantly both the reliability and efficiency of the flight clearance process, [24, 25].

In this chapter, the flight clearance problem for a highly detailed simulation model of a generic RLV over a lower atmospheric phase of its re-entry trajectory is considered. The flight control law included in the model has been designed using nonlinear dynamic inversion (NDI) methods to provide robust trajectory tracking over the specified flight phase. The clearance problem is solved using differential evolution and a hybrid version of differential evolution. Differential evolution is a relatively new global optimisation method, introduced by Storn and Price in [11]. This method belongs to the same class of evolutionary global optimisation techniques as GA, but unlike GA it does not require either a selection operator or a particular encoding scheme. To reduce the computational complexity of the approach, the DE algorithm is hybridised with a local gradient-based optimisation method 'fmincon'. The contributions of this chapter are as follows. We demonstrate conclusively, for a realistic, industry-standard re-entry vehicle simulation model with an NDI control law, the ability of the DE global optimisation algorithm to avoid getting trapped in local solutions to the flight clearance problem. We also show, however, that incorporation of local optimisation methods into global algorithms can drastically reduce computation times *and* improve convergence to the global solution. To the authors knowledge, this is the first time that advanced optimisation methods of this type have been applied to the problem of worst-case analysis for space applications.

2 RLV Model, Control Law and Clearance Criterion

The generic RLV high-fidelity simulation model is based on the HL-20 aerodynamic database and X38-type geometric and aerodynamic surface configuration, and has a dry mass of 19,100-lb. This simulation model has been developed by DEIMOS Space S.L. for the European Space Agency (ESA) to act as a research platform for the investigation of re-entry and autoland guidance, navigation and control systems, [23].

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 (US standard 1969 and Earth gravity and geoid models). Figure 1 shows a block diagram schematic of the RLV simulation model, which 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 desired dynamics. 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 [23]. The parameters in the model, and associated uncertainty values, are accessible through a database consisting of a collection of XML files accessible by the user.

Table 1. RLV Model Uncertain Parameters

Parameter	Bound	Description
Δ_{mass}	[-2313.3, 2313.3]	variation in dry mass from nominal (11566.55 kg)
Δ_{Ixx}	[-1627, +1627]	variation in M.I about X (8135.0 $4kgm^2$)
Δ_{Iyy}	[-15185, +15185]	variation in M.I about Y (75926.0 kgm^2)
Δ_{Izz}	[-15863, +15863]	variation in M.I about Z (79315.0 kgm^2)
Δ_{Ixz}	[-628.8, +628.8]	variation in Product of inertia XZ (3144.0 kgm^2
$\Delta_{x_{cog}}$	[-0.4912, +0.4912]	variation in X c.g from nominal (4.9213 m)
$\Delta_{y_{cog}}$	[-0.01, +0.01]	variation in Y c.g from nominal (0.0 m)
$\Delta_{z_{cog}}$	[-0.1009, +0.1009]	variation in Z c.g from nominal (1.0094976 m)

The complete re-entry trajectory for the vehicle takes 1680 seconds of simulation time and is divided in flight phases based on dynamic pressure and atmospheric layer. The present analysis focuses on a lower atmosphere phase starting at 1588 seconds and ending at 1675 seconds that covers the 32 to 20 km

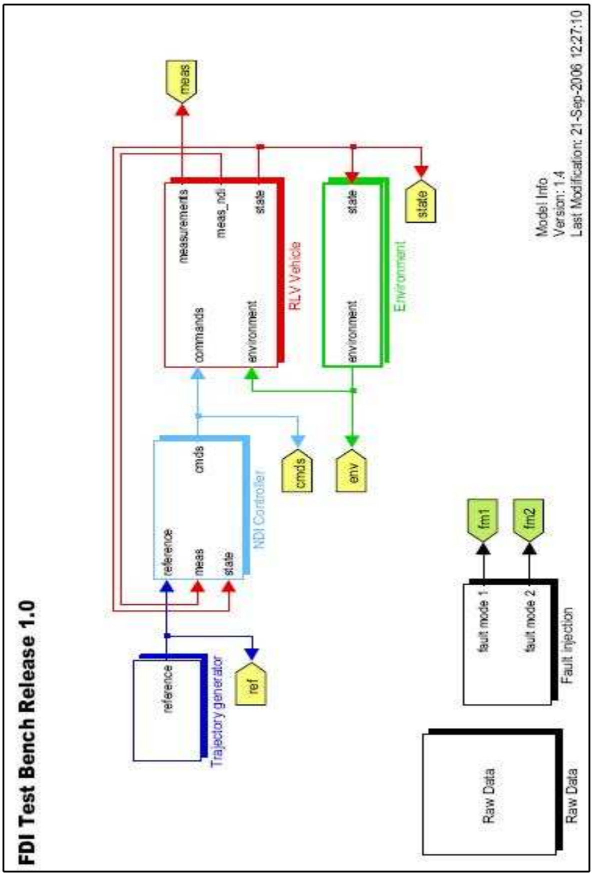

Fig. 1. Block schematic of RLV

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 description and allowed ranges of the uncertain parameters considered for the present analysis are given in Table 1. As can be seen from the table, in the present analysis we focus mainly on uncertainty in the parameters representing the vehicle's mass, inertias and centre-of-gravity.

2.1 Clearance Criterion

To analyse the robustness of the NDI control law in tracking AoA trajectories over the considered flight phase, a cost J is defined by Equation (1),

$$J = \|\alpha_{ref} - \alpha_{\Delta}\|_{\infty} \tag{1}$$

where α_{ref} represents the reference AoA trajectory and α_Δ represents the actual AoA trajectory followed by the vehicle in simulation in the presence of any uncertainty Δ. This particular clearance criterion was chosen for this study as criteria of this type are widely used throughout the European aerospace industry for the clearance of flight control laws for high performance aircraft [1,6]. The uncertain parameter vector Δ consists of the parameters defined in Table 1, and its dimension is hence fixed at 8. The worst-case analysis problem is posed as identifying the Δ^* vector such that the following maximisation problem is solved.

$$max J = \|\alpha_{ref} - \alpha_\Delta\|_\infty \tag{2}$$

$$sub.to \;\; \underline{\Delta} \leq \Delta \leq \overline{\Delta} \tag{3}$$

where $\underline{\Delta}$ and $\overline{\Delta}$ define the lower and upper bounds on the uncertain parameters. The maximum cost value J^* corresponds to the uncertain parameters Δ^* that give the maximum deviation from the reference trajectory α_{ref}. The resulting optimisation problem is obviously nonlinear and nonconvex in general. Note that in this chapter we focus on a clearance criterion involving AoA only. However, the optimisation framework proposed is generic, and thus many other types of clearance could be assessed in a similar way.

3 Optimisation Based Worst Case Analysis

In this chapter the robustness analysis of an NDI flight control law for a RLV is formulated as an optimisation problem and solved using a global optimisation algorithm, DE, and its hybrid version. The optimisation problem itself is to find the combination of real parametric uncertainties that gives the worst value of the criterion defined in Eq. 1. Since this and many other clearance criteria must be checked over a huge number of conditions and re-entry vehicle configurations, it is imperative to find the most computationally efficient approach to the problem. Previous efforts to apply optimisation methods to similar problem, [1] Chapter 7, have revealed that the nonlinear optimisation problems arising in flight clearance, while having relatively low order, often have multiple local optima and expensive function evaluations. Therefore, the issue of whether to use local or global optimisation, and the associated impact on computation times is a key consideration for this problem.

In [1] Chapter 21, local optimisation methods such as SQP (Sequential Quadratic Programming), and L-BFGS-B (Limited memory Broyden-Fletcher-Goldfarb-Shanno method with Bounded constraints) were used to evaluate a range of linear clearance criteria for the HIRM+ (High Incidence Research Model) aircraft model. In [1] Chapter 22, global optimisation schemes such as Genetic Algorithms (GA), Adaptive Simulated Annealing (ASA) and Multi Coordinate Search (MCS) were also applied to evaluate nonlinear clearance criteria for the same aircraft model. In [5, 6] global optimisation methods such as GA and ASA were applied to the ADMIRE model with a different flight clearance criterion. In [25], a number of optimisation schemes were employed and compared, evaluating a nonlinear clearance criterion for ADMIRE aircraft.

4 Differential Evolution

The global optimisation method considered in this study is differential evolution , a relatively new global optimisation method, introduced by Storn and Price in [11]. This method belongs to the same class of evolutionary global optimisation techniques as Genetic Algorithm (GA) [15], but unlike GA it does not require either a selection operator or a particular encoding scheme. Essentially a sub-type of GA, despite its apparent simplicity, the quality of the solutions computed using this approach has been claimed to be often better than that achieved using other evolutionary algorithms, both in terms of accuracy and computational overhead [11].

The DE method has recently been applied to several problems in different fields of engineering design, with promising results. In [17], for example, it was applied to find the optimal solution for a mechanical design example formulated as a mixed integer discrete continuous optimisation problem. In [18], DE was successfully applied in system design application, in particular handling the non-linear design specification constraints. In [10], the DE method was applied and compared with other local and global optimisation schemes in an aerodynamic shape optimisation problem for an aerofoil. The application of differential evolution and its hybridised versions with neural networks and local search methods for aerodynamic shape optimisation has been reported in [27]. In [25], a nonlinear flight clearance criterion for a modern high performance aircraft was posed and solved using both standard and hybrid GA and DE optimisation methods. In that study, it was demonstrated that a hybrid version of the DE algorithm significantly outperforms the corresponding GA method.

The DE method consists of the following four main steps 1) Random initialisation, 2) Mutation 3) Crossover 4) Evaluation and Selection. There are different schemes of DE available based on the operators. The one used in the present studies is referred as "$DE/rand/1/bin$". The steps of this scheme will be described in detail in the sequel.

4.1 Random Initialisation

Like other evolutionary algorithms, DE works with a fixed number, N_p, of potential solution vectors, initially generated at random according to

$$\mathbf{x}_i \;=\; \mathbf{x}^L \;+\; \rho_i(\mathbf{x}^U - \mathbf{x}^L),\;\; i = 1, 2, ..., N_p \tag{4}$$

where $\mathbf{x}^U$ and $\mathbf{x}^L$ are the upper and lower bounds of the parameters of the solution vector and ρ_i is a vector of random numbers in the range [0 1]. N_p is fixed at 30 in the current study. Each $\mathbf{x}_i$ consists of elements $(x_{1i}, x_{2i}, ..., x_{di})$, which are the uncertain parameters defined in Table 1. The dimension d of the optimisation problem considered is, therefore, 8. The fitness of each of these N_p solution vectors is evaluated using the cost function given in Eq. 1.

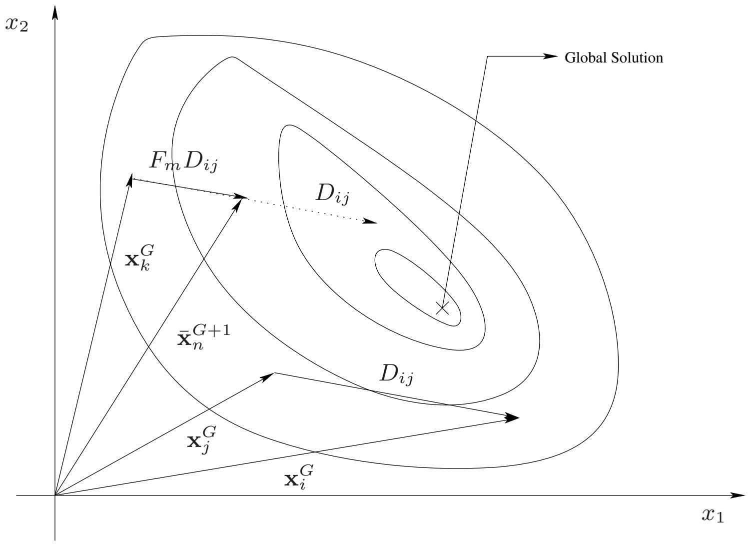

Fig. 2. DE mutation strategy

4.2 Mutation

The scaled difference vector $F_m D_{ij}$ between two random solution vectors $\mathbf{x}_i$ and $\mathbf{x}_j$ is added to another randomly selected solution vector $\mathbf{x}_k$ to generate the new mutated solution vector $\bar{\mathbf{x}}_n^{G+1}$, i.e.,

$$\bar{\mathbf{x}}_n^{G+1} = \mathbf{x}_k^G + F_m D_{ij}, \quad D_{ij} = \mathbf{x}_i^G - \mathbf{x}_j^G \tag{5}$$

where F_m is the mutation scale factor, a real valued number in the range [0, 1], (fixed at 0.8 in this study), and G represents the iteration number. Fig. 2 shows a simple two dimensional example of the mutation operation used in the DE scheme. The difference vector D_{ij} determines the search direction and F_m determines the step size in that direction from the point $\mathbf{x}_k^G$.

4.3 Crossover

During crossover, each element of the n^{th} solution vector of the new iteration, $\mathbf{x}^{G+1}$, is reproduced from the mutant vector $\bar{\mathbf{x}}_n^{G+1}$ and a chosen parent individual $\mathbf{x}_n^G$ as given in Eq. 6,

$$x_{ji}^{G+1} = \begin{cases} x_{ji}^G, & if a generated random number > \rho_c \\ \bar{x}_{ji}^{G+1}, & otherwise; \end{cases} \tag{6}$$

where $j = 1, 2, \ldots, d$ and $i = 1, 2, \ldots, N_p$. Note that $\bar{\mathbf{x}}_n^{G+1}$ has elements $(\bar{x}_{1n}^{G+1}, \bar{x}_{2n}^{G+1}, ..., \bar{x}_{dn}^{G+1})$ and $\mathbf{x}_n^G$ has elements $(x_{1n}^G, x_{2n}^G, ..., x_{dn}^G)$. $\rho_c \in [0, 1]$ is the crossover factor, which is fixed at 0.8 in the present study.

4.4 Evaluation and Selection

After crossover, the fitness of the new candidate $\mathbf{x}_n^{G+1}$ is evaluated and if the new candidate $\mathbf{x}_n^{G+1}$ has a better fitness than the parent candidate $\mathbf{x}_n^{G}$, then $\mathbf{x}_n^{G+1}$ is selected to become part of the next iteration. Otherwise $\mathbf{x}_n^{G}$ is selected and subsequently identified as $\mathbf{x}_n^{G+1}$.

4.5 Termination Criterion

Many different termination criteria can be employed. In the present study, an adaptive termination criterion is used that is dependent on improvement in the solution accuracy over a finite number of successive generations along with an upper limit on the computational budget. The algorithm terminates the search if there is no improvement on the best solution achieved (above a defined accuracy level, here chosen as 10^{-6}) for a defined successive number of generations. This number of generations is fixed at 20. Also, if the optimisation exceeds the defined computational budget, fixed at 2250, the algorithm is terminated. Defining the computational budget as a termination criterion is standard practice in aerospace industry applications.

5 Hybrid Optimisation

Global optimisation methods based on evolutionary principles are generally accepted as having a high probability of converging to the global or near global solution, if allowed to run for a long enough time with sufficient initial candidates and reasonably appropriate probabilities for the evolutionary optimisation parameters. As shown by the preceding results, however, the rate of convergence can be very slow, and moreover, there is still no guarantee of convergence to the true global solution. Local optimisation methods, on the other hand, can very rapidly find optimal solutions, but the quality of those solutions entirely depends on the starting point chosen for the optimisation routine. In order to try to extract the best from both schemes, several researchers have proposed combining the two approaches [16], [19], [20]. In such hybrid schemes there is the possibility of incorporating domain knowledge, which gives them an advantage over a pure blind search based on evolutionary principles. In [25], a Hybrid GA (HGA) scheme was developed using a switching strategy originally proposed in [20], and applied to a nonlinear flight clearance problem. The performance of the HGA scheme was compared to that of a novel Hybrid DE (HDE) scheme. For a recent comprehensive overview of other approaches to hybrid optimisation (also known as memetic algorithms), the reader is referred to [21].

5.1 Hybrid DE

In [22], the conventional DE methodology was augmented by combining it with a downhill simplex local optimisation scheme. This hybrid scheme was applied to an aerofoil shape optimisation problem and was found to significantly improve the convergence properties of the method. At each iteration, local optimisation was applied to the best individual in a current random set. The hybrid DE scheme employed in this study applies gradient-based local optimisation, again using *"fmincon"*, to a solution vector *randomly* selected from the current set - for our problem, this was seen to give better results than using the *best* solution vector, as proposed in [22]. Use of the local optimisation method based on gradient estimation, specifically the function *"fmincon"* provided in [7], is considered in present study to hybridise the DE algorithm. Local optimisation methods can, of course, get locked into a local optimum in the case of nonconvex and/or multimodal surfaces, however, they are also much more computationally efficient than global optimisation approaches. Whether a local method converges to a local or global optimum completely depends on the initial starting point in the search space, and the convexity of the search space. In the present context, the aim is to obtain local improvements in the search space and thereby accelerate the search to global solution. Crucially however, in typical flight clearance problems very little information is available as to where to start the optimisation - the number of uncertain parameters and strong nonlinearity of the system mean that even advanced knowledge of flight mechanics provides little insight into how to choose initial values for the uncertain parameters. In such case, a hybrid version of DE will be very beneficial in finding the true global solution, through ensuring an adequate coverage of search space. The function *"fmincon"* finds the constrained minimum of a scalar function of several variables starting

Table 2. Hybrid Differential Evolution Algorithm - Pseudo-Code:

1. Initialize random candidate solutions in search space
2. Evaluate fitness of each solution and choose best fitness
3. Apply DE for a fixed number of initial iterations(say 10); Update best fitness value in each iteration
4. While termination criteria not satisfied
   a) calculate the improvement in best fitness
      i. If *Improvement* in best fitness
      ii. Continue DE
      iii. else
      iv. Choose a random solution from current set, say X_0
      v. Apply local optimisation with X_0 as initial point (termination occurs when exceeding the defined maximum number of function evaluations)
      vi. If *Improvement* in best fitness
      vii. Replace X_0 with the new solution
      viii. else
      ix. Keep X_0 in the set
      x. end
      xi. end
5. end of While

at an initial estimate. In the present analysis, constraints are due only to the upper and lower bounds of the uncertainty in the variables. A medium scale optimisation scheme is chosen where the gradients are estimated by the function itself using the finite difference method. The function uses the sequential quadratic programming (SQP) method - for further details of the *"fmincon"* optimisation strategy, the reader is referred to [7]. When the local scheme is chosen, the optimisation starts from the given initial condition and continues until it either converges or reaches a defined maximum number of cost function evaluations. The algorithm is simple, and tries to search for the global optimum in a "greedy" way, demanding improvement in the achieved optimum value in every iteration. A pseudo-code for the hybrid DE algorithm is given in Table 2.

6 Worst-Case Analysis Results

The optimisation-based worst-case analysis procedure is implemented in the Matlab 2006A and Simulink 6.1 environments. The various uncertain parameters listed in Table 1 are considered as the optimisation parameters and these variables are normalised by multiplication with an appropriate scaling factor. Prior to simulation, the respective entries of the uncertain variables in the XML database are accessed and updated with the new set of values provided by the optimisation algorithm. The cost function as given in Eqn.(1) is evaluated at the end of every simulation. The optimisation algorithm iterates, identifies the

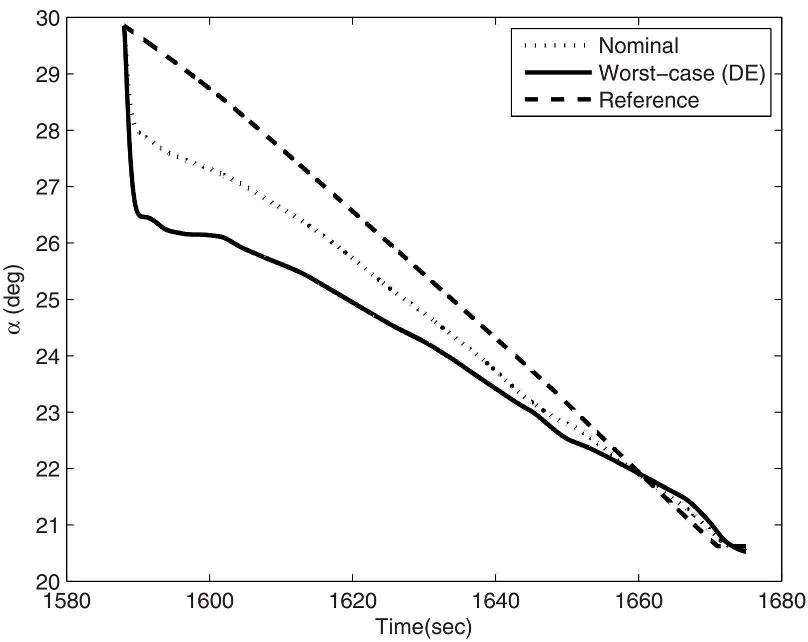

Fig. 3. Re-entry Angle of Attack trajectory

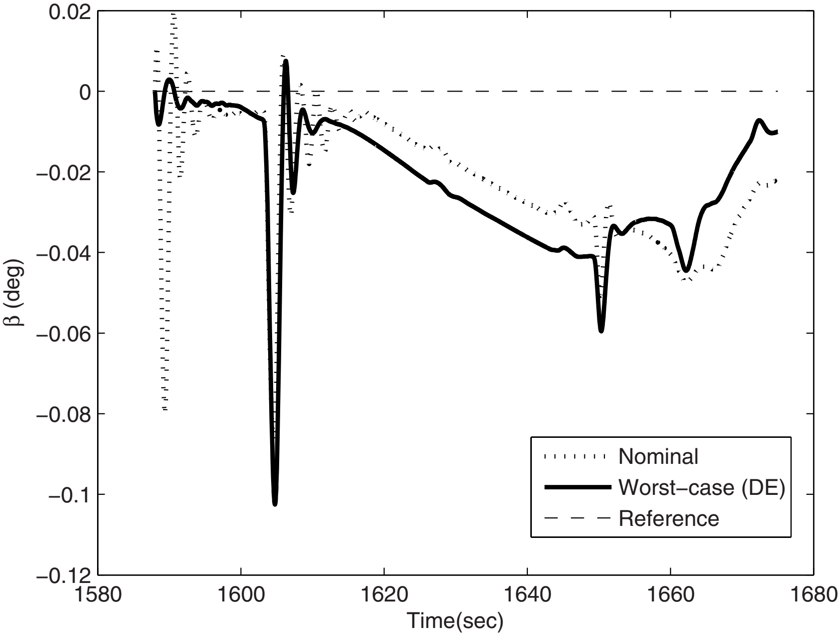

Fig. 4. Re-entry Angle of Sideslip trajectory

potential solutions and eventually converges to the global solution. However, to control the computational complexity we defined an adaptive termination criterion for the worst case analysis problem. In addition, an upper bound on the computational budget is also provided, fixed at 2250.

For the problem considered here, the DE algorithm took a total number of 2250 simulations, which is the computational budget termination criterion. The normalised worst-case obtained is $[\Delta_{mass}, \Delta_{Ixx}, \Delta_{Iyy}, \Delta_{Izz}, \Delta_{Ixz}, \Delta_{x_{cog}}, \Delta_{y_{cog}}, \Delta_{z_{cog}}] = [-0.9995, 0.9897, 0.9937, 0.3428, -0.9914, -0.9986, -0.1967, 0.9949]$.

Figure 3 shows the corresponding reference trajectory, worst-case and nominal angle-of-attack responses for the RLV model. Figure 4 shows the corresponding nominal and worst-case deviations from the desired zero value of $\beta(t)$. Interestingly, although the present cost function depends only on the value of $\alpha(t)$, the significant amount of cross-coupling between longitudinal and lateral dynamics at high AoA results in the worst-case β trajectory also being significantly different from the nominal response. To explore this issue further, a multi objective clearance criteria can be considered in this same framework, to identify the set of worst-case uncertain parameters for all of the controlled variables that define the reference trajectory.

Figure 5 shows the convergence of the DE algorithm. The x-axis indicates the function evaluations and the y-axis represents the maximum best cost value achieved over iterations. It can be seen from this figure that the DE algorithm shows good performance in the initial runs but subsequently the convergence rate

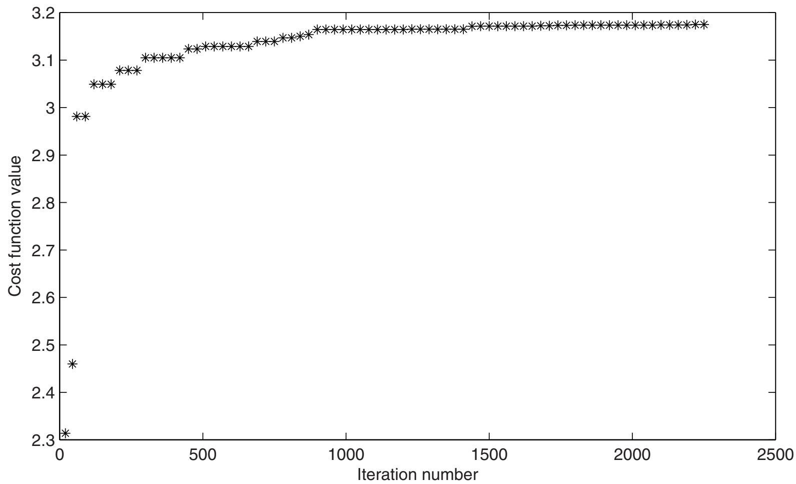

Fig. 5. Convergence of DE optimisation

becomes significantly slower. A second experiment was conducted with double the allowable computational budget criterion, with the adaptive termination criterion part left untouched. In this case, the maximum allowable computational budget was 5000. The repeated optimisation took a total number of 3250 simulations and the normalised worst-case obtained was [Δ_{mass}, Δ_{Ixx}, Δ_{Iyy}, Δ_{Izz}, Δ_{Ixz}, $\Delta_{x_{cog}}$, $\Delta_{y_{cog}}$, $\Delta_{z_{cog}}$]=[$-1,\ 1,\ 1,\ 1,\ -1,\ -1,\ 1,\ 1$], producing a worst-case cost value 3.178. The difference in the worst-case solution obtained from the two experiments can be explained with the help of a sensitivity analysis about the solution obtained from the optimisation. Figure 6 shows the results of a sensitivity analysis conducted about the global solution, by varying one parameter at a time and fixing all the other parameter values to their worst-case values. It can be noticed that the parameter $\Delta_{x_{cog}}$ has the greatest influence on the dynamics of the model, while the dynamics are relatively insensitive to the parameters Δ_{Ixx} and Δ_{Izz}. The presence of such insensitive parameters can make the optimisation convergence very slow. A possible way to avoid such a situation is by providing a termination condition of variation for insensitivity in parameter space.

The fact that the worst-case value of the uncertainties describing mass, centre-of-gravity and inertia variations are all on their maximum or minimum bounds is not surprising, and agrees well both with flight mechanics intuition and with the results of previous studies. The situation will becomes much more complex however when stability derivatives, sensor errors, etc are included, since in this case the corresponding worst-cases will not necessarily lie on the uncertain parameter bounds.

When compared with the standard DE algorithm, the HDE algorithm took a total number of 1775 simulations to converge. The normalised worst-case

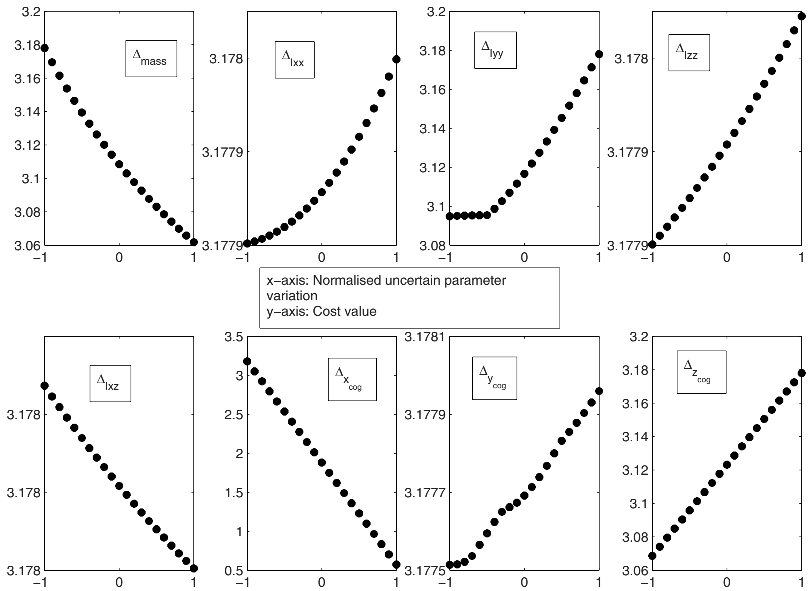

Fig. 6. Sensitivity analysis about the global solution

obtained is $[\Delta_{mass}, \Delta_{Ixx}, \Delta_{Iyy}, \Delta_{Izz}, \Delta_{Ixz}, \Delta_{x_{cog}}, \Delta_{y_{cog}}, \Delta_{z_{cog}}]$=$[-1,\ 1,\ 1,\ 1,\ -1,\ -1,\ 1,\ 1]$. Thus, to obtain a solution of the same quality, HDE has taken 45% less simulations than those required for the standard DE algorithm. This clearly demonstrates the significant computational savings which may be made by hybridising global optimisation algorithms with local gradient-based methods. Such savings are particularly crucial in the context of flight clerance, where computational cost is one of the key drivers for industrial applications.

7 Conclusions

In this chapter, differential evolution and hybrid differential evolution algorithms were applied to perform a worst-case analysis of a nonlinear-dynamic inversion (NDI) flight control law for a realistic simulation model of a re-entry vehicle over a particular phase of the trajectory for re-entry flight. A clearance criterion was defined based on the maximisation of the infinity norm of the error vector between the reference trajectory in Angle-of-Attack and the actual trajectory obtained by simulation of the model. The results of the study suggest that the proposed optimisation-based approach has the potential to improve significantly both the reliability and efficiency of the flight clearance process for future Reusable Launch Vehicles.

References

1. Fielding, C., Varga, A., Bennani, S., Selier, M. (eds.): Advanced techniques for clearance of flight control laws. Springer, Heidelberg (2002)
2. Menon, P.P., Kim, J., Bates, D.G., Postlethwaite, I.: Improved Clearance of Flight Control Laws Using Hybrid Optimisation. In: Proc. of the IEEE Conference on Cybernetics and Intelligent Systems, Singapore (December 2004)
3. Forssell, L.S., Hovmark, G., Hyden, Å., Johansson, F.: The aero-data model in a research environment (ADMIRE) for flight control robustness evaluation, GARTUER/TP-119-7 (August 1, 2001), `http://www.foi.se/admire/main.html`
4. Rundqwist, L., Stahl-Gunnarsson, K., Enhagen, J.: Rate limiters with phase compensation in JAS39 GRIPEN. In: Proc. of the European Control Conference, July 1997, pp. 2451–2457 (1997)
5. Forssell, L.S., Hyden, Å.: Flight control system validation using global nonlinear optimisation algorithms. In: Proc. of the European Control Conference, Cambridge, U.K. (September 2003)
6. Forssell, L.S.: Flight clearance analysis using global nonlinear optimisation based search algorithms. In: Proc. of the AIAA Guidance, Navigation, and Control Conference, Austin, Texas (August 2003)
7. Optimization toolbox users guide, Version 2, The MathWorks (September 2000)
8. Back, T., Hammel, U., Schwefel, H.P.: Evolutionary computation: comments on the history and current state. IEEE Transactions on Evolutionary Computation 1(1), 3–17 (1997)
9. Fleming, P.J., Purshouse, R.C.: Evolutionary algorithms in control systems engineering: a survey. Control Engineering Practice 10, 1223–1241 (2002)
10. Rogalsky, T., Derksen, R.W., Kocabiyik, S.: Differential evolution in aerodynamic optimization. Canadian Aeronautics and Space Institute Journal 46, 183–190 (2000)
11. Storn, R., Price, K.: Differential evolution: a simple and efficient heuristic for global optimization over continuous space. Journal of Global Optimization 11, 341–369 (1997)
12. Clerc, M., Kennedy, J.: The particle swarm - explosion, stability, and convergence in a multidimensional complex space. IEEE Transactions on Evolutionary Computation 6(1), 58–73 (2002)
13. Dorigo, M., Gambardella, L.M.: Ant colony system: a cooperative learning approach to the traveling salesman problem. IEEE Transactions on Evolutionary Computation 1(1), 53–66 (1997)
14. Zelinka, I., Lampinen, J.: SOMA - self-organizing migrating algorithm. In: 6th International Conference on Soft Computing, Brno, Czech Republic (2000) ISBN 80-214-1609-2
15. Goldberg, D.E.: Genetic algorithms in search, optimization and machine learning. Addison-Wesley, Reading (1989)
16. Davis, L. (ed.): Handbook of genetic algorithms. Van Nostrand Reinhold, New York (1991)
17. Lampinen, J., Zelinka, I.: Mechanical engineering design by differential evolution. In: Corne, D., Dorigo, M., Glover, F. (eds.) New Ideas in Optimisation, pp. 127–146. McGraw-Hill, London (1999)
18. Storn, R.: System design by constraint adaptation and differential evolution. IEEE Transactions on Evolutionary Computation 3(1), 22–34 (1999)

19. Yen, J., Liao, J.C., Randolph, D., Lee, B.: A hybrid approach to modeling metabolic systems using genetic algorithm and simplex method. In: Proc. of the 11th IEEE Conference on Artificial Intelligence for Applications, Los Angeles, CA, Feburary 1995, pp. 277–283 (1995)
20. Lobo, F.G., Goldberg, D.E.: Decision making in a hybrid genetic algorithm, IlliGAL Report No. 96009 (September 1996)
21. Krasnogor, N., Smith, J.: A tutorial for competent memetic algorithms: model, taxonomy, and design issues. IEEE Transactions on Evolutionary Computation 9(5), 474–488 (2005)
22. Rogalsky, T., Derksen, R.W.: Hybridization of differential evolution for aerodynamic design. In: Proc. of the 8th Annual Conference of the Computational Fluid Dynamics Society of Canada, pp. 729–736 (2000)
23. FDI Test Bench Software User Manual, FDITB-DME-SUM, Version 1.1 (September 18, 2006)
24. Menon, P.P., Bates, D.G., Postlethwaite, I.: Robustness Analysis of Nonlinear Flight Control Laws over Continuous Regions of the Flight Envelope. In: Proceedings of the IFAC Symposium on Robust Control Design (July 2006)
25. Menon, P.P., Kim, J., Bates, D.G., Postelthwaite, I.: Clearance of nonlinear flight control laws using hybrid evolutionary optimisation. IEEE Transactions on Evolutionary Computation 10(6), 689–699 (2006)
26. Menon, P.P., Bates, D.G., Postlethwaite, I.: A Deterministic Hybrid Optimisation Algorithm for Nonlinear Flight Control Systems Analysis. In: Proceedings of the American Control Conference (June 2006)
27. Madavan, N.: Aerodynamic shape optimisation using hybrid differential evolution, AIAA-2003-3792, 21st AIAA Applied aerodynamic conference, Orlando, Florida, USA (2003)