

## Article

# Multi-Objective Optimisation under Uncertainty with Unscented Temporal Finite Elements

Lorenzo A. Ricciardi , Christie Alisa Maddock \*  and Massimiliano Vasile 

Aerospace Centre of Excellence, University of Strathclyde, Glasgow G1 1XJ, UK;  
lorenzo.a.ricciardi@gmail.com (L.A.R.); massimiliano.vasile@strath.ac.uk (M.V.)

\* Correspondence: christie.maddock@strath.ac.uk

**Abstract:** This paper presents a novel method for multi-objective optimisation under uncertainty developed to study a range of mission trade-offs, and the impact of uncertainties on the evaluation of launch system mission designs. A memetic multi-objective optimisation algorithm, named MOD-HOC, which combines the Direct Finite Elements in Time transcription method with Multi Agent Collaborative Search, is extended to account for model uncertainties. An Unscented Transformation is used to capture the first two statistical moments of the quantities of interest. A quantification model of the uncertainty was developed for the atmospheric model parameters. An optimisation under uncertainty was run for the design of descent trajectories for a spaceplane-based two-stage launch system.



**Citation:** Ricciardi, L.A.; Maddock, C.A.; Vasile, M. Multi-Objective Optimisation under Uncertainty with Unscented Temporal Finite Elements. *Mathematics* **2021**, *9*, 3010. <https://doi.org/10.3390/math9233010>

Academic Editors: David Greiner, António Gaspar-Cunha, Daniel Hernández-Sosa, Edmondo Minisci and Aleš Zamuda

Received: 9 September 2021

Accepted: 16 November 2021

Published: 24 November 2021

**Publisher's Note:** MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.



**Copyright:** © 2021 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

**Keywords:** optimal control; multi-objective optimisation; robust design; trajectory optimisation; uncertainty quantification; unscented transformation; spaceplanes; space systems; launchers

## 1. Introduction

This paper presents a novel method for multi-objective optimisation under uncertainty, developed to study a range of mission trade-offs and the impact of uncertainties on system models for space launch systems. This is applied to the analysis and design of descent trajectories for a two-stage, partially re-usable launch system based on the Orbital-500R, a commercial system developed by Orbital Access Ltd. (Prestwick, UK) [1]. The set of Pareto-optimal solutions show the trade-off between minimising the induced acceleration limits and maximising the robustness of the solutions by minimising the sensitivity to uncertainties.

Uncertainty quantification (UQ), the science of quantifying the uncertainty in the desired performance of a system, can be a key step in analysing the robustness of a control solution and of the whole guidance, navigation and control chain. Common approaches to UQ use extensive Monte Carlo simulations to account for errors, unmodelled components and disturbances. At a system level, UQ analysis can translate into the assessment of the reliability of the system as a whole, or only of one or more components. An uncertainty quantification analysis is, therefore, a fundamental step towards de-risking any technological solution as it provides a quantification of the variation in performance and probability of recoverable or unrecoverable system failures, given existing information.

The goal here is, therefore, to design a robust guidance trajectory considering the uncertainty related to the atmospheric model which in turn affects the aero-, aerothermal and flight dynamics.

The trajectories are designed using a MODHOC (Multi-Objective Direct Hybrid Optimal Control) solver [2,3]. MODHOC, first developed in collaboration with ESA, is based on a transcription using temporal finite elements (DFET, Direct Finite Elements in Time) of the optimal control problem and a solution of the transcribed problem with a multi-agent multi-objective optimisation algorithm (MACS, Multi-Agent Collaborative Search).

This paper presents an extension of the multi-objective optimal control to account for uncertainties. The extension is based on an unscented transformation to capture the first two statistical moments of the quantities of interest. The result is an unscented multi-objective optimal control approach that can efficiently handle the level of uncertainty in model parameters.

Ross et al. [4] introduced unscented optimal control as the combination of the unscented transform by Julier and Uhlmann [5] combined with deterministic optimal control theory to directly manage uncertainties within an open-loop control framework. This has been applied to single objective optimisation problems mainly in the field of guidance and attitude control [6,7], using pseudospectral optimal control methods [8] with the common Legendre and Chebyshev polynomials as the bases. Ross et al. [9] later extended the work to account for path constraints on the states and controls. This paper builds on this work by adapting and applying it to a multi-objective optimal control problem, subject to nonlinear boundary and path constraints. The transcription method here uses finite elements on a temporal basis, using Bernstein polynomials, which have been mathematically shown to ensure both the states and controls representations remains feasible over the entire time domain, not just at collocation nodes [10].

The methods presented will be applied to trade-off studies on the first stage of a multi-stage horizontal take-off and landing launch system. The analysis focuses on the unpowered descent trajectory of a spaceplane, starting from the stage separation point at 100 km altitude. Uncertainties are introduced on the atmospheric parameters, which in turn strongly affect the aerodynamics and flight performance of the vehicle. The optimisation will trade-off robustness, by minimising the effect of these uncertainties, against the structural loads induced by the flight dynamics.

The paper is structured first introducing the mathematical method and implementation, followed by the description of the applied test case and results. Specifically, Section 2 describes the integration of an unscented transform into a multi-objective optimal control problem, including the direct transcription method using temporal finite elements with Bernstein bases. Section 3 details the solution of the multi-objective nonlinear programming problem through an adaption of a global evolutionary Multi-Agent Collaborative Search (MACS) algorithm tailored to optimal control problems. Section 4 presents the quantification of the uncertainty on the atmospheric parameters predicted by the International and US-76 Standard Atmospheric models. Section 5 describes the vehicle and environment models for the launch vehicle test case, with Section 6 presenting and discussing the results, including a validation of the methods and results against a Monte Carlo analysis.

## 2. Unscented Multi-Objective Optimal Control

In order to perform robust optimisation of the trajectory, an Unscented Transformation [5] was included in the formulation of the optimal control problem. An unscented transformation is defined as “the application of a given nonlinear transformation to a discrete distribution of points, computed so as to capture a set of known statistics of an unknown distribution, is referred to as an unscented transformation” [11]. These points are referred to as sigma points.

Unscented transformations capture the first statistical moments, mean and covariance, of the distributions of the states of a system subject to uncertainty and undergoing arbitrary nonlinear transformations by propagating a small number of sigma points. If the system depends on  $N_{uq}$  uncertain variables, whose mean and covariances are known, the unscented transformation requires the propagation of  $(2N_{uq} + 1)$  samples. The first sigma point takes the mean value for all the uncertain variables, while the others assume the mean plus (or minus) the square root of the matrix of the covariances of the uncertain variables. All the sigma points are propagated simultaneously with the mean and covariance of the final states computed as a weighted combination of the final states of each sigma point.

Let the dynamics of the system be given by

$$\dot{\mathbf{x}} = f(\mathbf{x}(\mathbf{u}, t), \mathbf{u}(t), \mathbf{b}, t) \quad (1)$$

where  $\mathbf{x}$  is the system state vector,  $\mathbf{u}$  are the controls,  $t$  is time, and  $\mathbf{b}$  are additional static (time independent) parameters. Similar to (1), the dynamics of each sigma point  $\chi_i$  are given by

$$\dot{\chi}_i = f(\chi_i(\mathbf{u}, t), \mathbf{u}(t), \mathbf{b}_i, t) \quad (2)$$

where  $i = [1, \dots, (2N_{uq} + 1)]$ . Each sigma point has a different value for the static variables, its dynamics evolve independently of the other sigma points, but all sigma points are controlled by the same control law  $\mathbf{u}$ . The goal is to find a single control law that, when applied to all sigma points, allows the system to reach a desired final condition and to be optimal, in some sense. The particular values for each static variable  $\mathbf{b}_i$  is decided by the application of the Unscented Transformation.

A known problem of the unscented transformation is that it can generate covariance matrices that are not semidefinite positive. To avoid this problem, the Square Root Unscented Transformation [12] was implemented. Algorithmically, it is very similar to the standard Unscented Transformation but differs in the way the samples are generated and has the advantage that the resulting covariance matrix is guaranteed to be semidefinite positive (up to machine precision). The sigma points are computed from the Cholesky factorisation of the covariance matrix, which decomposes the matrix into lower triangular matrix with real and positive diagonal entries, and its conjugate transpose [12].

The problem can be described as follows. Let  $\mathbf{X}$  be a state vector of length  $N_\sigma N_x$  defined as

$$\mathbf{X} = [\chi^0, \chi^1, \dots, \chi^{N_\sigma}]^T \quad (3)$$

where  $N_\sigma$  is the number of sigma points and  $N_x$  is the number of states of the system. The state dynamics are then defined as

$$\dot{\mathbf{X}} = \begin{bmatrix} \mathbf{F}(\chi_1, \mathbf{u}, \mathbf{b}_0, t) \\ \mathbf{F}(\chi_2, \mathbf{u}, \mathbf{b}_1, t) \\ \vdots \\ \mathbf{F}(\chi_{N_\sigma}, \mathbf{u}, \mathbf{b}_{N_\sigma}, t) \end{bmatrix} = \mathbf{F}(\mathbf{X}, \mathbf{u}, \mathbf{B}, t) \quad (4)$$

The multi-objective unscented optimal control problem is then formulated as

$$\min_{\mathbf{u} \in U} \mathbf{J}(\mathbf{X}, \mathbf{u}, \mathbf{B}, t) \quad (5)$$

$$\dot{\mathbf{X}} = \mathbf{F}(\mathbf{X}, \mathbf{u}, \mathbf{B}, t) \quad (6)$$

$$\mathbf{g}(\mathbf{X}, \mathbf{u}, \mathbf{B}, t) \geq 0 \quad (7)$$

$$\boldsymbol{\psi}(\mathbf{X}(t_0), \mathbf{X}(t_f), \mathbf{u}(t_0), \mathbf{u}(t_f), \mathbf{b}, t_0, t_f) \geq 0 \quad (8)$$

$$t \in [t_0, t_f]$$

where  $\mathbf{J} = [J_1, \dots, J_m]^T$  is, in general, a vector function of the state variables  $\chi_i : [t_0, t_f] \rightarrow \mathbb{R}^{N_x}$ , control variables  $\mathbf{u} \in L^\infty(U \subset \mathbb{R}^{N_u})$  and time  $t$ . Functions  $\mathbf{X}$  belong to the Sobolev space  $W^{1,\infty}$ , objective functions are  $J_i : \mathbb{R}^{2N_x+N_x} \times \mathbb{R}^{N_u} \times [t_0, t_f] \rightarrow \mathbb{R}$ ,  $\mathbf{F} : \mathbb{R}^{N_x N_\sigma} \times \mathbb{R}^{N_u} \times [t_0, t_f] \rightarrow \mathbb{R}^{N_x N_\sigma}$ , algebraic constraint function  $\mathbf{g} : \mathbb{R}^{N_x N_\sigma} \times \mathbb{R}^{N_u} \times [t_0, t_f] \rightarrow \mathbb{R}^{N_g}$ , and boundary condition functions  $\boldsymbol{\psi} : \mathbb{R}^{2N_x N_\sigma + 2} \rightarrow \mathbb{R}^{N_\psi}$ .

#### Direct Transcription with Temporal Finite Elements

The optimal control problem in (5)–(8) is transcribed into a many-objective, nonlinear programming problem via Direct Transcription with Finite Elements in Time (DFET) [13]. DFET was first proposed by Vasile [14] in 2000, and uses finite elements in time on spectral bases to transcribe the differential and algebraic constraints, and objective function into a set of algebraic equations. The formation allows different bases to be selected for both the states and controls, and for different segments. As a scheme, DFET has been proven to be robust, accurate and flexible [13].

For the continuous optimal control problem in (5), the time domain  $\mathcal{T} = [t_0, t_f]$  is decomposed into  $N$  finite elements  $\mathcal{T}_j(\tau_{j-1}, \tau_j)$ , with each element normalised to the interval  $[-1, 1]$  through the transformation

$$\tau = 2 \frac{t - \frac{1}{2}(t_j - t_{j-1})}{t_j - t_{j-1}} \quad t_{j-1} \leq t \leq t_j \quad \text{for } j = 1, \dots, N. \quad (9)$$

This ensures the domain of the basis functions are consistent irrespective of the element size. The differential constraints in (6) are first recast in weak variation form as

$$\int_{\mathcal{T}_j} \dot{\mathbf{w}}^T \mathbf{X} + \mathbf{w}^T \mathbf{F}(\mathbf{X}, \mathbf{u}, \mathbf{b}, t) dt - \mathbf{w}^T(t_j) \mathbf{X}_j^b + \mathbf{w}^T(t_{j-1}) \mathbf{X}_{j-1}^b = 0 \quad (10)$$

where  $\mathbf{w}$  are generalised weight functions, and  $\mathbf{X}_j^b$  and  $\mathbf{X}_{j-1}^b$  are the values of the states at the boundaries of each element. For each element  $\mathcal{T}_j$ , the states, controls, and weight functions can be parameterised according to the basis functions  $\mathbf{f}_{s,j}$  such that:

$$\mathbf{X}_j = \sum_{s=0}^{l_x} f_{s,j}^{X_j}(\tau) \mathbf{X}_{s,j} \quad (11)$$

$$\mathbf{u}_j = \sum_{s=0}^{l_u} f_{s,j}^{u_j}(\tau) \mathbf{u}_{s,j} \quad (12)$$

$$\mathbf{w}_j = \sum_{s=0}^{l_x+1} f_{s,j}^{w_j}(\tau) \mathbf{w}_{s,j} \quad (13)$$

For this paper, Bernstein polynomials are used as the bases functions for all the elements, of order  $l_x$  for the states,  $l_u$  for the controls and  $(l_x + 1)$  for the weights. Bernstein basis polynomials are defined generally as

$$b_{\nu,n}(t) = \binom{n}{\nu} t^\nu (1-t)^{n-\nu} \quad \text{for } \nu = 0, \dots, n \text{ and } 0 \leq t \leq 1 \quad (14)$$

where  $n$  is the order. Bernstein bases have the advantage of smooth control profiles with no oscillations near discontinuities or step changes, meaning the polynomial representation of both states and controls remains within the feasible set [10,15].

Recasting (10) into Gauss quadrature using the polynomials in (11)–(13) gives

$$\sum_{k=0}^{l_u} \beta_k \left[ \dot{\mathbf{w}}_j(\tau_k)^T \mathbf{X}_j(\tau_k) + \mathbf{w}_j(\tau_k)^T \mathbf{F}_j(\tau_k) \frac{\Delta t_j}{2} \right] - \mathbf{w}^T(1) \mathbf{X}_j^b + \mathbf{w}^T(-1) \mathbf{X}_{j-1}^b = 0 \quad (15)$$

where  $\tau_k$  and  $\beta_k$  are Gauss nodes and weights, respectively,  $\Delta t_j = (t_j - t_{j-1})$  and  $\mathbf{F}_j(\tau_k)$  is the shorthand notation for  $\mathbf{F}(\mathbf{X}_j(\tau_k), \mathbf{u}_j(\tau_k), \mathbf{b}, t(\tau_k))$ . Since (15) must be valid for every arbitrary  $\mathbf{w}_{s,j}$ , this can be written as a system of equations for each element:

$$\sum_{k=0}^{l_u} \beta_k \left[ \dot{f}_{0,j}(\tau_k) \mathbf{X}_j(\tau_k) + f_{1,j}(\tau_k) \mathbf{F}_j(\tau_k) \frac{\Delta t_j}{2} \right] + \mathbf{X}_{j-1}^b = 0 \quad \text{for } k = 0 \quad (16)$$

$$\sum_{k=0}^{l_u} \beta_k \left[ \dot{f}_{s,j}(\tau_k) \mathbf{X}_j(\tau_k) + f_{s,j}(\tau_k) \mathbf{F}_j(\tau_k) \frac{\Delta t_j}{2} \right] = 0 \quad \text{for } k = 1, \dots, l_x \quad (17)$$

$$\sum_{k=0}^{l_u} \beta_k \left[ \dot{f}_{l_x+1,j}(\tau_k) \mathbf{X}_j(\tau_k) + f_{l_x+1,j}(\tau_k) \mathbf{F}_j(\tau_k) \frac{\Delta t_j}{2} \right] - \mathbf{X}_j^b = 0 \quad \text{for } k = (l_x + 1) \quad (18)$$

The path constraints from (7) are directly collocated at the Gauss nodes, generating a set of constraint equations for each element given by

$$\mathbf{g}(\mathbf{X}_j(\tau_k), \mathbf{u}_j(\tau_k), \mathbf{b}, t(\tau_k)) \geq 0 \quad (19)$$

Constraints are also imposed on the boundary states of all adjacent elements to ensure continuity.

The transcribed objective functions from (5) are therefore

$$\tilde{J}_i = \begin{cases} \sum_{k=0}^{l_u} \phi_k(\mathbf{X}_k, \mathbf{u}_k, \tau_k, \mathbf{b}) \\ \phi_i(\mathbf{X}_0^b, \mathbf{X}_f^b, t_0, t_f, \mathbf{b}) \end{cases} \quad (20)$$

which equate to the two terms in a Bolza optimisation problem [16].

The time domain  $\mathcal{T}$  corresponds to a single time period  $[t_0, t_f]$ . For launch systems, however, trajectories often have multiple phases either in series, or in parallel. For example, a multi-stage vehicle can have one phase per vehicle stage with all phases connected in series for the ascent, and/or branching parallel phases for the upper stage ascent, and first stage descent and landing. For a problem with  $N_p$  distinct phases, the dynamic constraints (15), path constraints (19), boundary constraints (8) and objective functions (20) are defined per phase. An additional set of  $N_p$  boundary constraints are introduced to manage the connections between phases defined by

$$\psi_{s_p}(\mathbf{X}_{0, \mathbf{I}_{s_p}}^b, \mathbf{X}_{f, \mathbf{I}_{s_p}}^b, t_{0, \mathbf{I}_{s_p}}, t_{f, \mathbf{I}_{s_p}}) \geq 0 \quad s_p = 1, \dots, N_p \quad (21)$$

where the index vector  $\mathbf{I}_{s_p}$  collects all the indexes of the phases that are connected by the constraint  $\psi_{s_p}$ . Note that, while the number of phases  $N_p$  is fixed, their temporal order is defined by the phase boundary constraints (21).

The resulting multi-objective nonlinear programming (MONLP) problem coming from the transcription of (5)–(8), with the inclusion of phase constraints (21), is given by

$$\begin{aligned} \min_{\mathbf{y} \in Y, \mathbf{p} \in \Pi} \quad & \tilde{\mathbf{J}}(\mathbf{y}, \mathbf{p}) \\ \text{C}(\mathbf{y}, \mathbf{p}) \geq & 0 \end{aligned} \quad (22)$$

where  $\mathbf{y} = [\mathbf{X}_{0,1}, \dots, \mathbf{X}_{s,j}, \dots, \mathbf{X}_{l_x,N}]^T$ ,  $Y$  is a box in  $\mathbb{R}^{n_Y}$  with  $n_Y = n(l_x + 1)N$ ,  $\mathbf{p} = [\mathbf{u}_{0,1}, \dots, \mathbf{u}_{s,j}, \dots, \mathbf{u}_{l_x,N}, \mathbf{b}^*]^T$  is a solution, or decision, vector that collects all the static and discretised control variables with  $\mathbf{b}^* = [\mathbf{b}, \mathbf{x}_0^b, \mathbf{x}_f^b, t_0, t_f]^T$ ,  $\Pi \subseteq \mathbb{R}^{n_s} \times \mathbb{R}^{n_b^*}$  with  $n_s = n_u(l_u + 1)N$  (assuming that each element has the same number of control parameters) and  $n_b^* = n_b + 2n + 2$ , and  $\mathbf{C}$  collects all path and boundary constraints.

Similar to (5), the solution of (22) is a subset  $\Omega_\Pi \subset \Pi$  that satisfies the constraints  $\mathbf{C}$  and contains solution vectors that are Pareto efficient. Given the subset  $\Omega_\Pi$  of feasible solution vectors, a solution vector  $\mathbf{p}^* \in \Omega_\Pi$  is said to be Pareto efficient if  $\mathbf{p}^* \not\succ \mathbf{p}, \forall \mathbf{p} \in \Omega_\Pi$ . The symbol of dominance  $\succ$  indicates that, if  $\mathbf{p}_1 \succ \mathbf{p}_2$ , then  $\tilde{J}_i(\mathbf{p}_2) \leq \tilde{J}_i(\mathbf{p}_1)$  for  $i = 1, \dots, m$  and  $\exists j$  such that  $\tilde{J}_j(\mathbf{p}_2) < \tilde{J}_j(\mathbf{p}_1)$ . In other words, a solution is non-dominated if the values of any of the objective functions, using that solution, cannot be improved without sacrificing at least one of the other objectives [17]. For continuous functions, the subset  $\Omega_\Pi$  is a manifold in  $\mathbb{R}^{n_s + n_b^*}$  with dimension  $(n_s + n_b^*) \leq (m - 1)$  [18]. In the following, the goal is to identify a pre-defined countable number of Pareto-efficient solutions contained in  $\Omega_\Pi$ .

### 3. Solution of the Transcribed Problem

The MONLP problem in (22) is solved with an adaption of the Multi-Agent Collaborative Search (MACS) tailored to optimal control problems [10]. MACSoc combines

the stochastic agent-based global search in MACS [19,20] with a local refinement of the solutions [21,22] (see Algorithm 1).

---

**Algorithm 1** MACS optimal control (MACSoc)

---

```

1: Initialise population  $\mathcal{P}_0$  and global archive  $\mathcal{A}_0, k = 0, \rho_B = 1$ 
2: Initialise weight vectors  $\omega$ 
3: while  $n\_fun\_eval < max\_fun\_eval$  do
4:   Run individualistic heuristics on  $\mathcal{P}_k$  using bi-level formulation
5:    $\mathcal{P}_k \rightarrow \mathcal{P}_k^+$ 
6:   Update archive  $\mathcal{A}_k$  with potential field filter
7:   Run social heuristics combining  $\mathcal{P}_k^+$  and  $\mathcal{A}_k$  using bilevel formulation
8:   Update archive  $\mathcal{A}_k$  with potential field filter
9:    $\mathcal{P}_k^+ \rightarrow \mathcal{P}_k^*$ 
10:  if local search triggered then
11:    Run gradient based refinement using single level formulation
12:     $\mathcal{P}_k^* \rightarrow \mathcal{P}_k^*$ 
13:    Update archive  $\mathcal{A}_k$  with potential field filter
14:     $\mathcal{P}_k^* \rightarrow \mathcal{P}_{k+1}$ 
15:  else
16:     $\mathcal{P}_k^+ \rightarrow \mathcal{P}_{k+1}$ 
17:  end if
18:   $k = k + 1$ 
19:  Update  $\rho_B$ 
20: end while

```

---

At the start of MACSoc, an initial population  $\mathcal{P}_0$  is generated with  $N_a$  agents representing feasible candidate solutions. Next, a set of  $N_w$  uniformly spread weight vectors  $\omega$  are generated. Each agent is associated with a different weight vector, allowing the agent to converge to a different part of the Pareto-optimal set (set of non-dominated solutions for the multi-objective optimisation problem).

The global search generates candidate solutions for the decision vector using a combination of social and individualistic actions (lines 4 and 7 in Algorithm 1). Each action generates a candidate decision vector, starting from the current solution allocated to a given agent  $j$ , and submits it to a bi-level optimisation problem, where the inner level makes the candidate decision vector feasible with respect to constraints, and the outer level assesses whether the solution of the inner level represents an improvement with respect to the current solution allocated to agent  $j$ . All feasible and non-dominated solutions are added to the current population  $\mathcal{P}_k$ , and saved in an archive  $\mathcal{A}_k$  (lines 5, 6, 8, 9 and 13 in Algorithm 1). A local refinement is triggered periodically after a user-defined number of iterations, and at the end of the algorithm, which update the current population and archive (lines 10–17 in Algorithm 1). The local refinement solves a single level scalarised version of (22).

The entire process alternates between social and individualistic actions, with periodic local refinement, until a maximum number of calls to the objective vector  $max\_fun\_eval$  is reached.

### 3.1. Bi-Level Global Optimisation Problem

The NLP problem for the global optimisation is defined by

$$\begin{aligned}
 & \min_{\mathbf{p}^*} \tilde{\mathbf{J}}(\mathbf{y}^*, \mathbf{p}^*) \\
 & (\mathbf{y}^*, \mathbf{p}^*) = \underset{\mathbf{y}, \mathbf{p}}{\operatorname{argmin}} \{ \delta_p(\mathbf{y}, \mathbf{p}) \mid \mathbf{C}(\mathbf{y}, \mathbf{p}) \geq 0 \}
 \end{aligned} \tag{23}$$

and represents two optimisation sub-problems at two different levels. The outer level minimises the objective function vector  $\tilde{\mathbf{J}}$  and generates a first set of candidate solutions



**p.** The inner level looks for state  $\mathbf{y}^*$  and control  $\mathbf{p}^*$  vectors that satisfy the constraint functions  $\mathbf{C}$ , and minimise a cost function based on the candidate solutions of the outer loop  $\delta_p = \|\mathbf{p}^* - \mathbf{p}\|$  that look the closest feasible solution to the candidate solution provided by the outer loop. The feasible solution is then passed back to the outer loop to evaluate the objective functions  $\mathbf{J}$  with  $(\mathbf{y}^*, \mathbf{p}^*)$ . The inner level problem is solved with a local, gradient-based optimiser such as SQP or interior point.

In order to reduce the number of iterations required by the inner level to converge, the outer level stores the feasible states  $\mathbf{y}^*$  from one iteration to be used as a first guess for the inner level at the next iteration. As shown in Figure 1, the feasible states  $\mathbf{y}_k^*$  are preserved from iteration  $k$  to iteration  $(k + 1)$ ; therefore, the outer level only generates candidate solutions for  $\mathbf{p}_{k+1}$ . Thus, for iteration  $(k + 1)$ , the inner level is given  $(\mathbf{y}_k^*, \mathbf{p}_{k+1})$  as initial guesses for states and controls. Despite  $\mathbf{y}_k^*$  being associated with  $\mathbf{p}_k^*$ , it has been shown to work well as an initial guess also when associated with  $\mathbf{p}_{k+1}$ .

When individualistic actions are applied, each agent generates one or more candidate solution vectors through three mechanisms that are triggered sequentially in the order: Inertia  $\rightarrow$  Pattern Search  $\rightarrow$  Differential Evolution. If any of these mechanisms produces an improved solution, the process is stopped and proceeds to update the population and archive (line 5 and 6 of Algorithm 1).

**Inertia** is triggered by agent  $j$  only if, in the previous iteration, agent  $j$  generated an improved solution. In this case, a step with random length is taken in the direction defined by  $(\mathbf{p}_k^* - \mathbf{p}_{k-1}^*)$ .

**Pattern Search** will change one optimisation parameter at a time, by a random amount in each direction, within a given neighbourhood  $\mathcal{B}_j$  of agent  $j$ . The order by which the parameters are changed is a random permutation of the number of decision parameters. The process is repeated until either an improvement is registered or the maximum number of trials has been reached. As in Ricciardi and Vasile [20], the maximum number of trials is dynamically adjusted during the optimisation process: when the archive is empty, the maximum number of parameters scanned is equal to the total number of optimisation parameters. This maximum value is decreased linearly as the archive fills up, until only one optimisation parameter is changed when the archive is full. The neighbourhood  $\mathcal{B}_j$  is a box centred in the position of the agent in parameter space and with the edges equal to the edges of the search space  $\Pi$  multiplied by the scaling parameter  $\rho_{\mathcal{B}_j}$ .

**Differential Evolution** generates a sample with the simple heuristic:

$$\mathbf{p}_{trial,j} = \mathbf{p}_j + \xi_1 \mathbf{e}((\mathbf{p}_j - \mathbf{p}_{j_1}) + c_F(\mathbf{p}_{j_2} - \mathbf{p}_{j_3})) \quad (24)$$

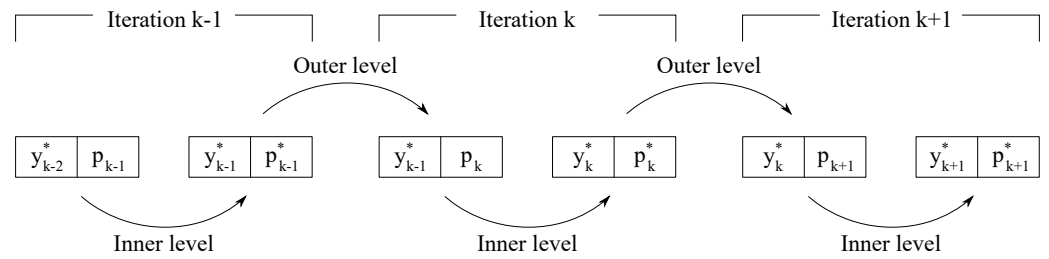
where  $\mathbf{p}_j$  is the current candidate solution,  $\mathbf{p}_{j_1}, \mathbf{p}_{j_2}, \mathbf{p}_{j_3}$  are three randomly chosen solutions from the current population  $\mathcal{P}_k$ ,  $\xi_1$  is a uniformly distributed random number in the unit interval,  $c_F$  is a user-defined constant and  $\mathbf{e}$  is a mask vector defined as

$$e_j = \begin{cases} 1, & \text{if } \xi_2 < CR \\ 0, & \text{otherwise} \end{cases} \quad (25)$$

where  $\xi_2$  is another uniformly distributed random number in the unit interval, and  $CR$  is the crossover rate. For the following test cases,  $c_F = 0.9$  and  $CR = 1$ .

If no improvement is made after trying all the three heuristics,  $\rho_B \rightarrow 0.5\rho_B$ ; if instead an improvement is made,  $\rho_B \rightarrow 2\rho_B$  until the initial value  $\rho_B = 1$  is reached again.

When social actions are applied, the outer level uses the population to generate a candidate solution using the same heuristics of Differential Evolution (24), with the parent solutions  $\mathbf{p}_{j_1}, \mathbf{p}_{j_2}, \mathbf{p}_{j_3}$  chosen from the union of the current population  $\mathcal{P}_k$  and the current archive  $\mathcal{A}_k$ .



**Figure 1.** Schematic representation of the bilevel approach acting on a single solution.

A candidate solution  $(\mathbf{y}^*, \mathbf{p}^*)$  generated by the inner level is evaluated in the outer level by computing the weighted Chebychev norm

$$\Phi_i = \max_i \omega_i (\tilde{f}_i(\mathbf{y}^*, \mathbf{p}^*) - z_i) \quad (26)$$

$$z_i = \min_{\mathcal{P}_k \cup \mathcal{A}_k} \tilde{f}_i$$

where  $\omega$  is the weight vector in objective space and  $\mathbf{z}$  is the current utopia point.

Given the value of  $\Phi_i$  at step  $k$ , or  $\Phi_i^k$ , an improvement corresponds to  $\Phi_i^{k+1} < \Phi_i^k$ . This improvement criterion has two very important properties: first, it allows the search to reach even non-convex parts of the Pareto front, and, second, if the weights are chosen appropriately, it enables an efficient convergence to the global minimum for each objective function.

Note that, if the inner level does not converge to the required tolerance, the objective functions of the outer level are recalculated to be the infinity norm of the constraint violation plus the maximum values of each objective functions in the archive and population. This creates an adaptive rejection mechanism: if none of the agents are feasible, the ones that best satisfy the feasibility are entered in the archive, with the next iterations trying to improve their feasibility. Once an agent finds a feasible solution, it will explore the search space through the global bi-level approach, generating several feasible and non-dominated solutions. These solutions will enter in the archive as they will dominate many of the existing infeasible solutions, and due to the social actions, some agents will be directly moved onto those solutions, allowing the entire population to converge to feasible solutions in a handful of iterations.

Finally, if any candidate solutions for  $\mathbf{y}$  and  $\mathbf{p}$  fall outside the boundaries of the search space  $\mathcal{Y} \times \Pi$ , the solution vector is reduced until it is back within the search space.

### 3.2. Single Level Local Search

The local refinement solves the following scalarised problem for each agent  $j$ :

$$\begin{aligned} \min_{\epsilon \geq 0} \quad & \epsilon \\ & \omega_{i,j} \vartheta_{i,j}(\mathbf{y}, \mathbf{p}) \leq \epsilon \quad \text{for } i = 1, \dots, m \\ & \mathbf{C}(\mathbf{y}, \mathbf{p}) \geq 0 \end{aligned} \quad (27)$$

where  $\omega_{i,j}$  is the  $i^{\text{th}}$  component of the weights for the  $j^{\text{th}}$  agent,  $\vartheta_{i,j}$  is the  $i^{\text{th}}$  component of the rescaled objective vector of the  $j^{\text{th}}$  agent, and  $\epsilon$  is a slack variable.

This reformulation of the problem, which uses Pascoletti–Serafini scalarisation [23], constrains the movement of the agent to within a descent cone defined by the point  $(\epsilon \mathbf{d}_j + \mathbf{1}_j)$  and along the direction  $\mathbf{d}_j = (1/\omega_{1,j}, \dots, 1/\omega_{i,j}, \dots, 1/\omega_{m,j})$ . The rescaled vector of the objective functions is therefore

$$\vartheta_j(\mathbf{y}, \mathbf{p}) = \frac{\tilde{f}_{i,j}(\mathbf{y}, \mathbf{p}) - \tilde{z}_i}{z_{i,j}^* - \tilde{z}_i} \quad \text{for } i = 1, \dots, m \quad (28)$$



where  $\mathbf{z}_j^*$  is equal to  $\tilde{\mathbf{J}}_j(\bar{\mathbf{y}}, \bar{\mathbf{p}})$ ,  $(\bar{\mathbf{y}}, \bar{\mathbf{p}})$  is the initial guess for the solution of (27), and  $\tilde{\mathbf{z}} = (\mathbf{z} - \mathbf{z}_A)$  where  $\mathbf{z}_A$  is the nadir of the archive. The components of the vector  $\zeta_j$  are derived from the normalisation

$$\zeta_{i,j} = \frac{z_i}{z_{i,j}^* - \tilde{z}_i} \quad \text{for } i = 1, \dots, m \quad (29)$$

This allows the components of  $\vartheta_j(\mathbf{y}, \mathbf{p})$  to have values of 1 at the beginning of the local search, and 0 if the agent converges to the target point  $\tilde{\mathbf{z}}$ . Thus, the single level approach avoids biases when the objectives have significantly different scales.

The weighted Chebychev norm in (26) and (27) are equivalent and lead to the same optimal solution if the target point for the Pascoletti–Serafini scalarisation coincides with the utopia point, and the weight vectors are the same [24]. By combining (26) in the global search phase with (27) in the refinement phase, the algorithm ensures a smooth transition from global exploration of the Pareto set, to local convergence.

### 3.3. Archiving Strategy

MACSoc, through MACS, employs an archiving strategy described in Ricciardi and Vasile [20]. When the elements in the archive  $\mathcal{A}$  are less than the maximum allowed cardinality of  $\mathcal{A}$ , every new feasible and non-dominated solution is recorded in the archive. Once the defined maximum size for the archive is reached, new elements are added to  $\mathcal{A}$  only if they minimise the potential function,

$$E(\tilde{\mathbf{J}}_1, \dots, \tilde{\mathbf{J}}_{N_A}) = \sum_{i=1}^{N_A} \sum_{j=i+1}^{N_A} \frac{1}{(\tilde{\mathbf{J}}_i - \tilde{\mathbf{J}}_j)^T (\tilde{\mathbf{J}}_i - \tilde{\mathbf{J}}_j)} \quad (30)$$

where  $N_A$  is the number of elements in the archive  $\mathcal{A}$ .

To avoid biasing in the rejection–retention process when the objectives have different scales, the objective values of the set of non-dominated solutions are all normalised between 0 and 1. This leads to a combinatorial problem that can be solved approximately but efficiently and returns a uniformly spread set of points [20].

### 3.4. Generation of the Initial Feasible Population

Before the optimisation starts, MACSoc generates an initial population of agents  $P_0$  representing feasible candidate solutions.

A first guess for the candidate solutions is generated using Latin Hypercube sampling within the given boundaries, which gives a near-random sample of parameter values from a multidimensional distribution [18]. State variables for each phase are initialised with a linear interpolation between initial and final conditions. For each phase, each equation in (15) is optimised using the inner level subproblem (23) to ensure feasibility through a local gradient-based optimiser. Additional constraints within the phase are then added in, and the problem is re-optimised. Lastly, the linking constraints between phases are then included, and the full resulting problem is optimised for a final time (always using a local optimiser with the inner level objective function in (23)). If at the end of the initialisation phase, an agent is associated with a solution that is not feasible within the prescribed tolerance, that solution is still included in the initial population  $P_0$  and submitted to the subsequent optimisation cycle.

For the test case, the tolerance on feasibility is set to  $10^{-6}$ , both for the initial population generation, and for determining feasibility within loop. By default, the maximum number of calls to the constraint function is set equal to  $10(n_b^* + n_s + n_Y)$ . An interior point NLP solver was used as it delivered a more robust and consistent convergence to feasible solutions.

### 3.5. Definition of the Descent Directions and Target Points

The weight vectors for the bi-level global search are generated as follows: first, a simplex in objective space is generated through simplex lattice design [25]. Then, the points of this simplex lattice are projected on the unit sphere by dividing their position vectors by their distance to the origin. This gives a fairly uniform distribution of weight vectors (and thus descent directions) in any  $N_w$  dimensional space.

In order to generate a more uniform distribution, however, these weight vectors are refined using a local optimisation of the same potential function  $E$  given in (30),

$$\begin{aligned} \min E(\omega_1, \dots, \omega_{N_w}) \\ \omega_i^T \omega_i = 1 \end{aligned} \quad (31)$$

While this approach is valid for general  $m$ -objective problems, for two objectives, it is simpler and faster to generate uniform angularly spaced weight vectors. In the following,  $N_a = N_w$  and each agent is associated with the closest descent direction in criteria space, at the initialisation stage, with the constraint that no two agents can have the same descent direction.

For the single level approach, the weight vectors  $\omega_j = [\sqrt{2}, \dots, \sqrt{2}]^T$  are allocated to all agents except to those  $m$  agents that minimise each individual objective function. For these  $m$  agents, the weight vectors are  $\omega_j = [0, \dots, j, \dots, 0]^T$  with  $j = 1, \dots, m$ . These weights are orthogonal because they correspond to the  $m$  orthogonal directions in criteria space. If agent  $j$  associated with weight  $\omega_j$  does not generate any improvement after two iterations, a new random orthogonal weight is associated with  $j$  and (27) is solved with the added constraints,

$$\tilde{J}_i \leq z_i \quad \forall i \neq j \quad (32)$$

The reason for the different choice of weight vectors between the bi-level and the single level formulation can be explained as follows: the bi-level formulation explores globally the search space with a population of agents, thus there is the need to maximise the spreading of the solutions; on the contrary, the single-level is used to improve the local convergence of each agent in a normalised criteria space. Thus, the goal of the single level is to return dominating solutions without altering too much their spreading in criteria space.

## 4. Uncertainty Model for Atmospheric Parameters

Preliminary design and trade-off studies, in particular those employing computationally intensive multidisciplinary design and multi-objective optimisations, typically use global static atmospheric models such as US-76 Standard Atmospheric model, or the International Standard Atmospheric (ISA) model. These models predict the atmospheric pressure  $p$ , temperature  $T$  and density  $\rho$  as function solely of altitude, and employ simplified algebraic expressions for the different atmospheric layers. The models are based on average values for year-round, mid-latitude conditions with moderate solar activity. The ISA is valid up to an altitude above mean sea level of 86 km, while US-76 has an extension up to 1000 km. Due to this, US-76 is often preferred in the field of space launchers over ISA. These two models rely on similar assumptions and methodologies, differing only in the prediction of the temperature in the upper atmosphere (above 32 km).

In order to assess the robustness of the mission design against uncertainties in the atmospheric model, a model of the uncertainty was developed using higher fidelity atmospheric models to create a data set. A number of empirical, global reference atmospheric models exist, for example, to analyse the effect of atmospheric drag on satellites. The Committee on Space Research (COSPAR) in their International Reference Atmosphere reports details three of these models [26]: NRLMSISE-00 [27], Jacchia Bowman reference atmospheric model JB2008 [28], and DTM2013 [29].

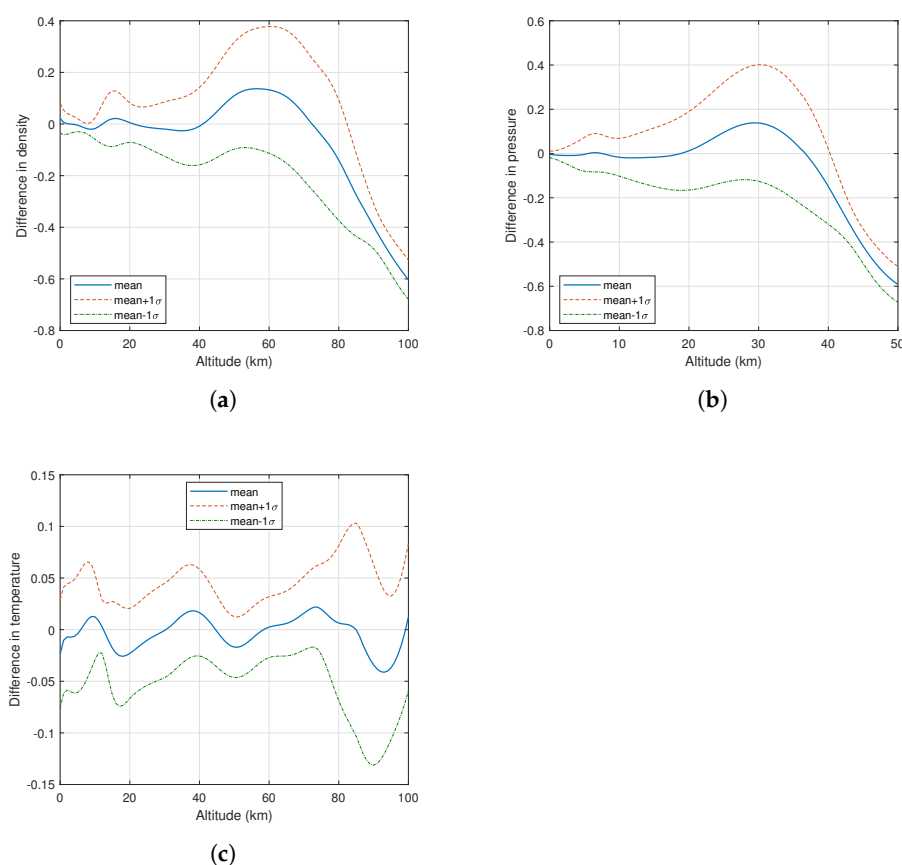
The NRLMSISE-00 is a model developed by the US Naval Research Laboratory and accounts for geographic, temporal, solar and magnetic effects through inputs for: date, time of day, geodetic altitude from 0 to 1000 km, geodetic latitude and longitude, local

apparent solar time, 81-day average of F10.7 solar flux, daily F10.7 solar flux for previous day, and daily magnetic index.

A statistical analysis of the difference in the models was performed treating those all the input parameters save altitude as uncertain. A set of  $10^5$  quasi-random samples were generated for the altitude using a low discrepancy Halton sequence [30], with the corresponding values for  $T$ ,  $P$  and  $\rho$  computed using the NRLMSISE-00 model for all altitudes in the range between 0 and 100 km. The altitude limit of 100 km was driven by the test case of the descent phase of a first stage, reusable spaceplane which is designed to operate within this range.

Every sample of the NRLMSISE-00 model was treated as an equivalent static global atmospheric model, and used to compare against the US-76 model. These differences were treated as random fluctuations in order to quantify the uncertainty.

As shown in Figure 2, the means and standard deviations  $\sigma$  were determined for each of the three atmospheric parameters. It can be observed that, for the temperature, the mean differences, or relative errors, are very low, with a  $1\sigma$  relative error around 5% for altitudes below 80 km. The mean relative errors for pressure and density are also low for relatively low altitudes, but increase at higher altitudes, with the largest  $1\sigma$  bands around 60 km. Above 40 km, the pressure and density have a very low absolute value, so a large relative error still means a low absolute error.



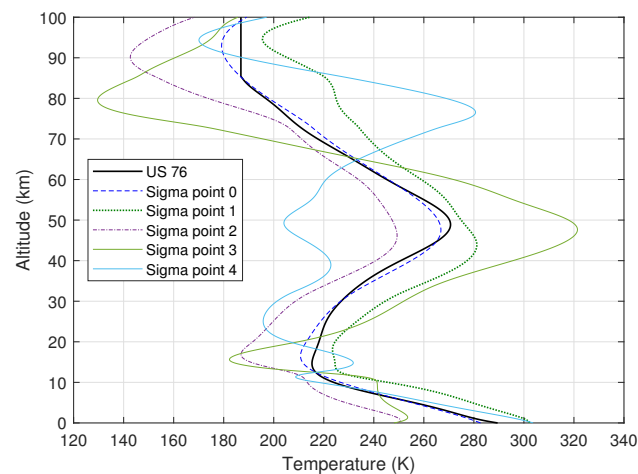
**Figure 2.** Mean and standard deviation ( $\pm 1\sigma$ ) of the differences between every sample (representing an equivalent global static atmospheric model) and the US-76 model for each of the atmospheric parameters as a function of altitude. (a) Atmospheric density ( $\text{kg}/\text{m}^3$ ); (b) Atmospheric pressure (kPa); (c) Atmospheric temperature (K).

Since the approach employed is that of the Square Root Unscented Transformation, different models were generated: one employing the mean relative error, and the others adding to the mean the Cholesky factorisation of the covariance matrix of the uncertain quantities at each altitude. Figure 3 shows the five generated temperature profiles, and

their comparison with respect to the US-76 model. A similar approach was followed for the density.

As it can be seen, the Sigma Point 0 model is quite close to the US-76 model. Sigma points 1 and 2 add the standard deviation to this mean, while Sigma points 3 and 4 also include the correlation between variations in temperature and variations in density. The correlation between temperature and density changes sign repeatedly as the altitude changes, thus the profiles assume values lower than one standard deviation only to cross the mean and assume values higher than one standard deviation elsewhere.

Among the  $10^5$  samples generated by the Halton sequence on the NRLMSISE-00 model, several profiles do indeed have this kind of shape, which is significantly different to the US-76 model. The temperature affects the computation of the Mach number, on which the aerodynamic coefficients depend. The dependence of the aerodynamic coefficients on the Mach number is stronger around Mach 1 and weaker for high Mach numbers, thus it is not easy to foresee the effect of these variations.



**Figure 3.** Temperature profiles for the models of each sigma point, and comparison with the US76 model.

## 5. Vehicle System Models

The Orbital-500R launch system is composed of a first stage reusable spaceplane, capable of rocket-powered ascent and an unpowered, glided descent, and an expendable, rocket-based upper stage. The system is air-launched from a carrier aircraft. The system was designed to launch from the UK and carry small payloads (500 kg) to Low Earth Orbits (e.g., up to 1500 km). The dry mass of the spaceplane is 20 tonnes.

The flight dynamics are modelled as a variable-mass point with three degrees of freedom in the Earth-centered Earth-fixed reference frame, subject to gravitational, aerodynamic lift and drag forces. The state vector contains the translational position and velocity components,  $\mathbf{x} = [h, \lambda, \theta, v, \gamma, \chi]$ , where  $h = r_E - r$  is the altitude given radial distance  $r$  and the Earth's radius  $r_E(\lambda, \theta)$ ,  $(\lambda, \theta)$  are the geodetic latitude and longitude,  $v$  is the magnitude of the relative velocity vector directed by the flight path angle  $\gamma$  and the flight heading angle  $\chi$ . The vehicle is controlled through the angle of attack  $\alpha$ , and the bank angle  $\mu$  of the vehicle. The dynamic model is therefore [31]

$$\dot{r} = \dot{h} = v \sin \gamma \quad (33)$$

$$\dot{\lambda} = \frac{v \cos \gamma \cos \chi}{r} \quad (34)$$

$$\dot{\theta} = \frac{v \cos \gamma \sin \chi}{r \cos \lambda} \quad (35)$$

$$\dot{v} = \frac{-D}{m} - g_r \sin \gamma + g_\phi \cos \gamma \cos \chi + \omega_E^2 r \cos \lambda (\sin \gamma \cos \lambda - \cos \gamma \cos \chi \sin \lambda) \quad (36)$$

$$\begin{aligned} \dot{\gamma} = & \frac{1}{v} \left( \frac{L \cos \mu}{m} - g_r \cos \gamma - g_\phi \sin \gamma \cos \chi \right) + \frac{v}{r} \cos \gamma \\ & + \frac{\omega_E^2 r}{v} \cos \lambda (\sin \gamma \cos \chi \sin \lambda + \cos \gamma \cos \lambda) + 2\omega_E \sin \chi \cos \lambda \end{aligned} \quad (37)$$

$$\begin{aligned} \dot{\chi} = & \frac{1}{v \cos \gamma} \left( \frac{L \sin \mu}{m} - g_\phi \sin \chi \right) + \frac{v}{r} \cos \gamma \sin \chi \tan \lambda \\ & + \omega_E^2 r \frac{\sin \chi \sin \lambda \cos \lambda}{v \cos \gamma} + 2\omega_E (\sin \lambda - \tan \gamma \cos \chi \cos \lambda) \end{aligned} \quad (38)$$

The Earth is modelled using WSG-84, with gravitational acceleration expressed in radial  $g_r$  and tangential  $g_\phi$  components.

The accelerations induced on the vehicle are determined, using the relative vehicle body reference frame  $\mathcal{B}$ .

$$\begin{aligned} a_x^{\mathcal{B}} = & \frac{-D \cos \alpha + L \sin \alpha}{m} - g_r (\sin \gamma \cos \alpha + \cos \gamma \cos \mu \sin \alpha) \\ & + g_\phi (\cos \gamma \cos \chi \cos \alpha - \cos \chi \sin \gamma \cos \mu \sin \alpha - \sin \chi \sin \mu \sin \alpha) \\ & - \frac{v}{r} \cos \gamma \sin \alpha (\cos \mu - \sin \chi \tan \lambda \sin \mu) \end{aligned} \quad (39)$$

$$\begin{aligned} a_y^{\mathcal{B}} = & g_r \cos \gamma \sin \mu + g_\phi (\cos \chi \sin \gamma \sin \mu - \cos \mu \sin \chi) \\ & + \frac{v}{r} \cos \gamma (\sin \mu + \cos \mu \sin \chi \tan \lambda) \end{aligned} \quad (40)$$

$$\begin{aligned} a_z^{\mathcal{B}} = & \frac{-D \sin \alpha - L \cos \alpha}{m} + g_r (\cos \gamma \cos \mu \cos \alpha - \sin \gamma \sin \alpha) \\ & + g_\phi (\cos \chi \sin \gamma \cos \mu \cos \alpha + \sin \chi \sin \mu \cos \alpha + \cos \gamma \cos \chi \sin \alpha) \\ & + \frac{v}{r} \cos \alpha \cos \gamma (\cos \mu - \sin \chi \tan \lambda \sin \mu) \end{aligned} \quad (41)$$

Surrogate models were used for the lift  $c_L$  and drag  $c_D$  coefficients as a function of Mach number  $M$ , angle of attack  $\alpha$  and altitude  $h$ . An artificial neural network was employed, using an aerodynamic database for the training data coming from a mix of panel methods and CFD simulations (see Stindt et al. [32] for details on the generation of the aerodynamic database). The lift and drag forces were computed assuming no wind, with the relative velocity  $\mathbf{v}_{rel} = \mathbf{v} - \boldsymbol{\omega}_E \mathbf{r}$ .

$$L = \frac{1}{2} C_L(M, \alpha, h) \rho S_{ref} v_{rel}^2 \quad (42)$$

$$D = \frac{1}{2} C_D(M, \alpha, h) \rho S_{ref} v_{rel}^2 \quad (43)$$

where  $S_{ref}$  is the reference area of the vehicle, and  $\rho$  is the atmospheric density.

Maddock et al. [1], Ricciardi et al. [10] performed a number trade-off studies through a multi-disciplinary design optimisation on design of the Orbital-500R launch system and on various missions, analysing the full trajectories for both stages. Extrapolating from these results, a nominal starting point for the descent was set. Table 1 lists the initial and final

conditions of the state variables. A final condition was imposed on the expected value of the altitude, with a value of 10 km. In addition, the trajectories of all sigma points were required to have a flight path angle greater or equal to  $-20^\circ$ .

**Table 1.** Values and constraints on the boundary values of the state variables.

	Initial Condition	Final Condition (Requirement)
Altitude $h$	90 km	$h = 10$ km
Latitude $\lambda$	$60^\circ$ N	
Longitude $\theta$	$-12^\circ$ E	
Velocity $v$	3 km/s	
Flight path angle $\gamma$	$-5^\circ$	
Heading angle $\chi$	$20^\circ$	$\chi \geq -20^\circ$

## 6. Application and Results

### 6.1. Problem Formulation and Set-Up

For the applied test case here, the initial conditions are fixed. The final conditions are affected by the uncertainty, and are expressed as difference between the mean and target values. A boundary constraint is added as follows:

$$\psi(\mathbf{X}(t_f)) = \mu_\chi - \bar{x}(t_f) = 0 \quad (44)$$

where  $\mu_\chi$  is the mean of the final states of the sigma points, and  $\bar{x}(t_f)$  is the target value.

The first objective uses a metric to minimise the average induced acceleration on the vehicle. The general formulation is

$$J_1 = \int_{t_0}^{t_f} E(\|\mathbf{a}\|^2) dt \quad (45)$$

where the transcribed version using DFET is

$$J_1 = \sum_j^N \frac{\delta t_j}{2} \sum_k^{l_u} \beta_k \sum_i^{N_\sigma} a^2(\chi_i, \tau_k) W_{\sigma,i} \quad (46)$$

where  $a = \|\mathbf{a}\| = \sqrt{a_x^2 + a_y^2 + a_z^2}$ , given in (39)–(41). As part of the generation of the sigma points,  $N_\sigma$  weights  $\mathbf{W}_\sigma$  are generated, associated with each sigma point. The computation of the mean of a quantity, within the transcribed NLP, is the weighted sum of the quantities associated with each sigma point [12].

The second objective aims to reduce the uncertainty of the final state by minimising the sum of the square of all the entries of the covariance matrix:

$$J_2 = \sum_{i,j} \left( \text{Cov}_{i,j}(\mathbf{X}(t_f)) \right)^2 \quad (47)$$

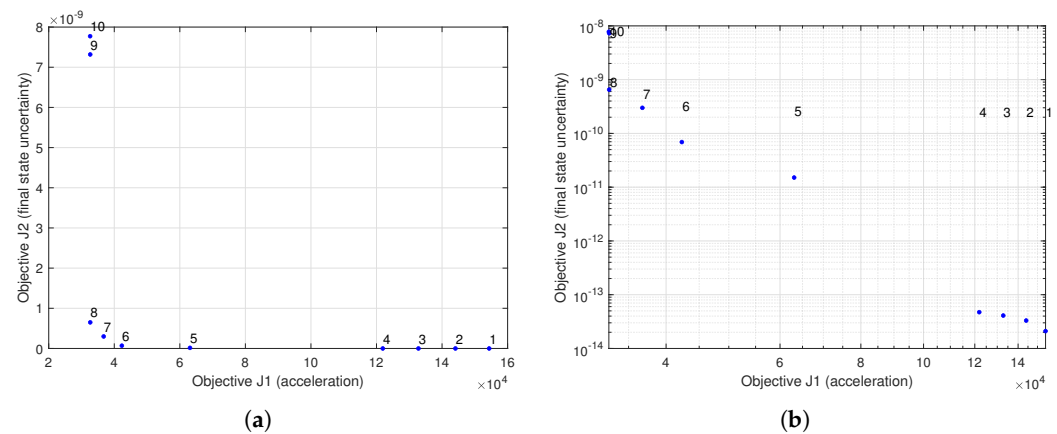
where  $\text{Cov}_{i,j}$  is computed using the standard algebraic manipulations employed for the Square Root Unscented transformation, with the additional consideration that no update of the Cholesky factorisation is needed since no measurement is here performed and thus no error is present. This formulation has the advantage that the quantity to compute is smooth and differentiable, it involves all components of the covariance matrix, and does not require iterative procedures like decomposition in eigenvalues to compute the principal axes of the ellipsoid of the uncertainty. In order to give each element of the covariance matrix the same weight even if the quantities of interest have different scales, the state variables were scaled by the same factors internally employed by MODHOC to ensure that all variables assume values between 0 and 1.



The equations were transcribed through DFET, using 6 elements of order 7 for all the states and the controls. Since all five sigma points are propagated simultaneously, there are 30 elements for the states and 12 elements for the controls. MODHOC was run for a total of 30,000 function evaluations, keeping 10 solutions in the archive.

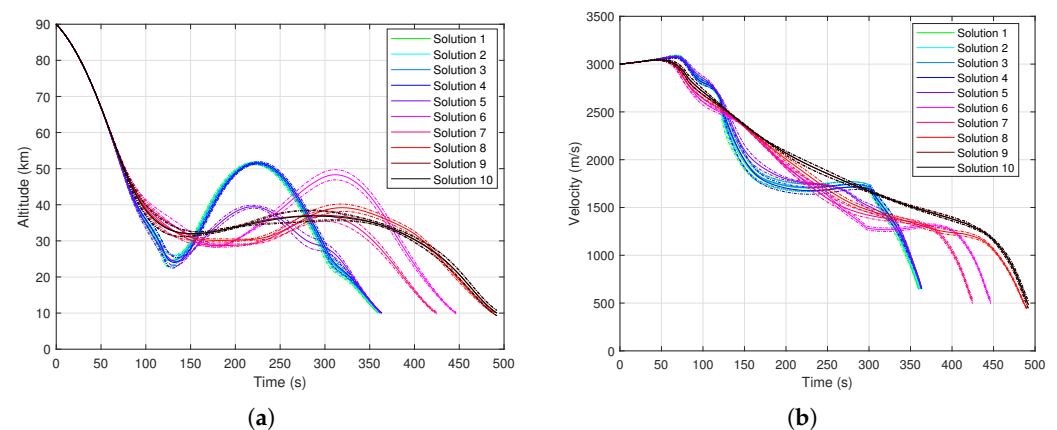
## 6.2. Trajectory Results

The computed Pareto front is shown in Figure 4, which confirms a trade-off between the two objectives corresponding to metrics to minimising the induced acceleration load on the vehicle ( $J_1$ ), and maximising the robustness of the trajectory by minimising the covariance of the final values of the state variables ( $J_2$ ).

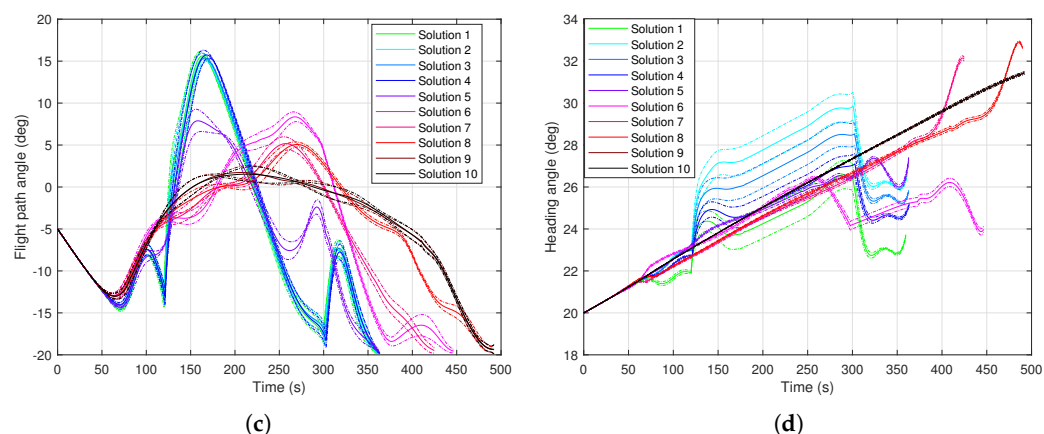


**Figure 4.** Pareto Front for the two-objective problem. (a) Linear scale; (b) Log-log scale.

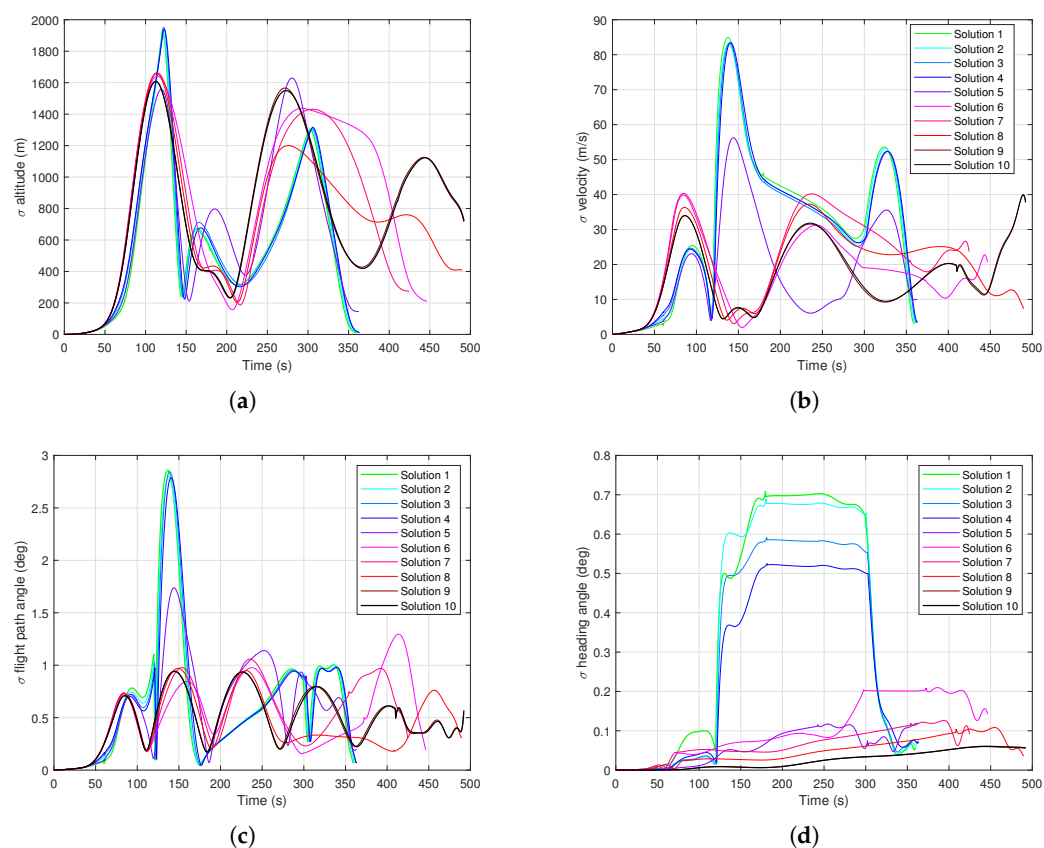
Figure 5 shows the time histories of the state variables for altitude, velocity vector components given by the magnitude, flight path and heading angles. The solutions are shown for a set of Pareto-optimal solutions from Figure 4, along with the  $\pm 1\sigma$  uncertainty indicated by the dashed and dotted lines. It can be seen that Solutions 1 to 4 (in greens and blues) have lower uncertainty for the final state, as expected as Solution 1 represents the extrema for  $\min(J_2)$ . This can also be seen from Figure 6, which show the standard deviation of the same four state variables in Figure 5. As previously stated, even if the uncertainty on atmospheric density is relatively high at high altitudes, its effect is quite limited for the first part of the trajectory as the absolute value is very small (e.g.,  $10^{-6}$  at 85 km,  $10^{-3}$  Pa at 40 km). The uncertainty in the density starts to have a noticeable effect as the altitudes get to approximately 40 km and below.



**Figure 5.** Cont.

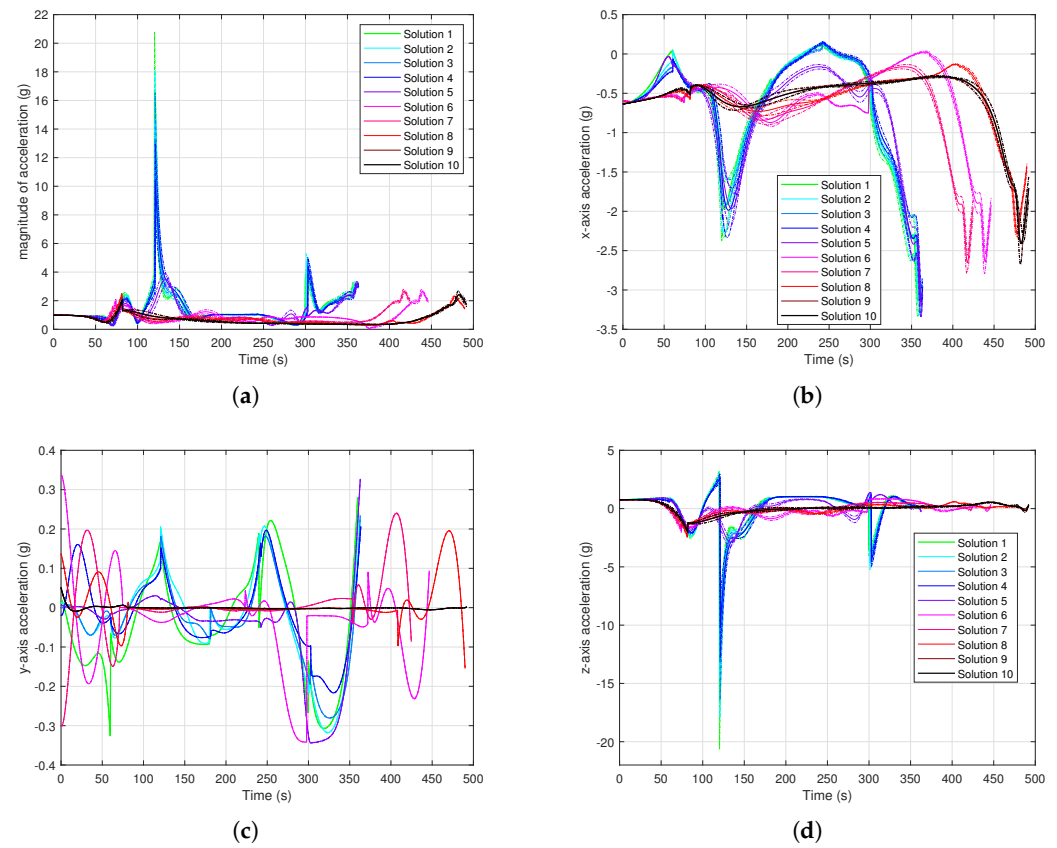


**Figure 5.** State variables for set of 10 Pareto-optimal solutions, with dashed and dotted lines showing  $\pm 1\sigma$  uncertainty. (a) Altitude; (b) Magnitude of the velocity; (c) Flight path angle; (d) Heading angle.



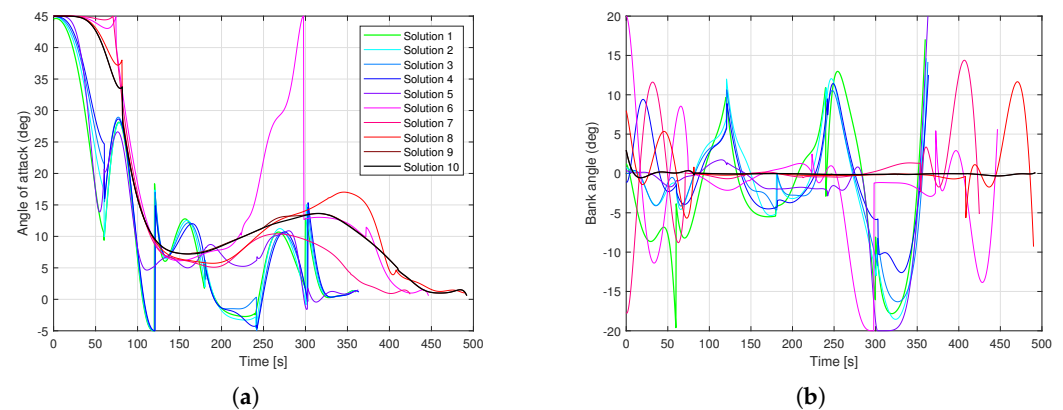
**Figure 6.** Standard deviations of state variables. (a) Altitude; (b) Magnitude of velocity; (c) Flight path angle; (d) Heading angle.

As the Pareto front (Figure 4) indicated, the solutions with a lower uncertainty of final states have higher acceleration loads, as shown in Figure 7. This further shows the breakdown between the accelerations in the three vehicle (or body) axes, as well as the magnitude of the acceleration vector.



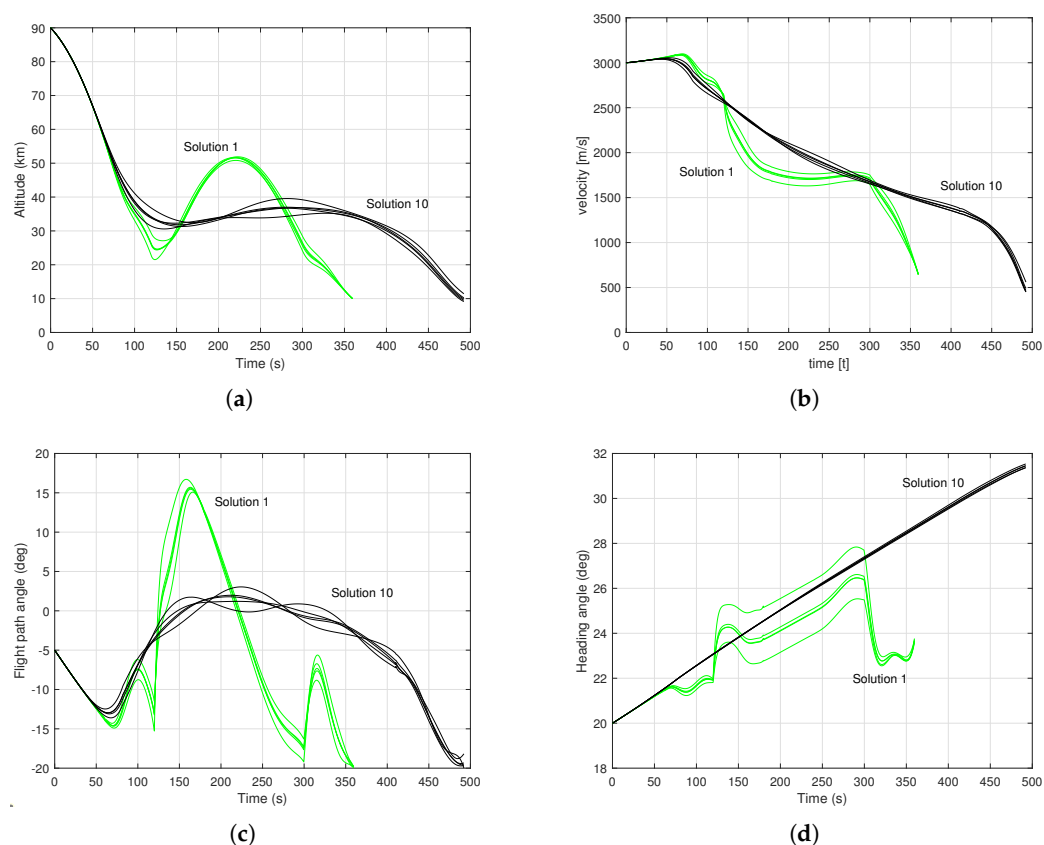
**Figure 7.** Magnitude and components (in vehicle body reference frame) of the induced accelerations, normalised against sea level gravity  $g = g_0$ . (a) Magnitude of acceleration; (b) Component in x-axis; (c) Component in y-axis; (d) Component in z-axis.

Figure 8 shows the time history for the two control variables: angle of attack  $\alpha$  and bank angle  $\mu$ . In all cases, the angle of attack starts with the maximum possible value of 45 deg, and then progressively decreases to a more moderate value around 5–10 deg. Solutions with lower accelerations stay in this regime for a while, and finally conclude with a value around 0 deg. Solutions with lower final uncertainty instead have a progressive decrease in  $\alpha$  until small negative values, then a sharp increase to values around 10–15 deg, and finally stabilise around 0 deg.



**Figure 8.** Control variables, which are consistent across all solutions in the Pareto-optimal set. (a) Angle of attack; (b) Bank attack.

Finally, Figures 9 show the time history of the states for all sigma points for extrema Solutions 1 and 10. As it is evident, the green lines have a much lower scattering at the final time than the black lines, indicating that Solution 1 (green) is subject to less uncertainty than solution 10 (black). These figures also give an idea of the complexity of the problem tackled by this approach, where the same control law is applied to multiple independent sigma points (lines with the same colour) and is able to steer the system to a given expected final state while also reducing the uncertainty associated with the final state, or reducing the expected acceleration load.



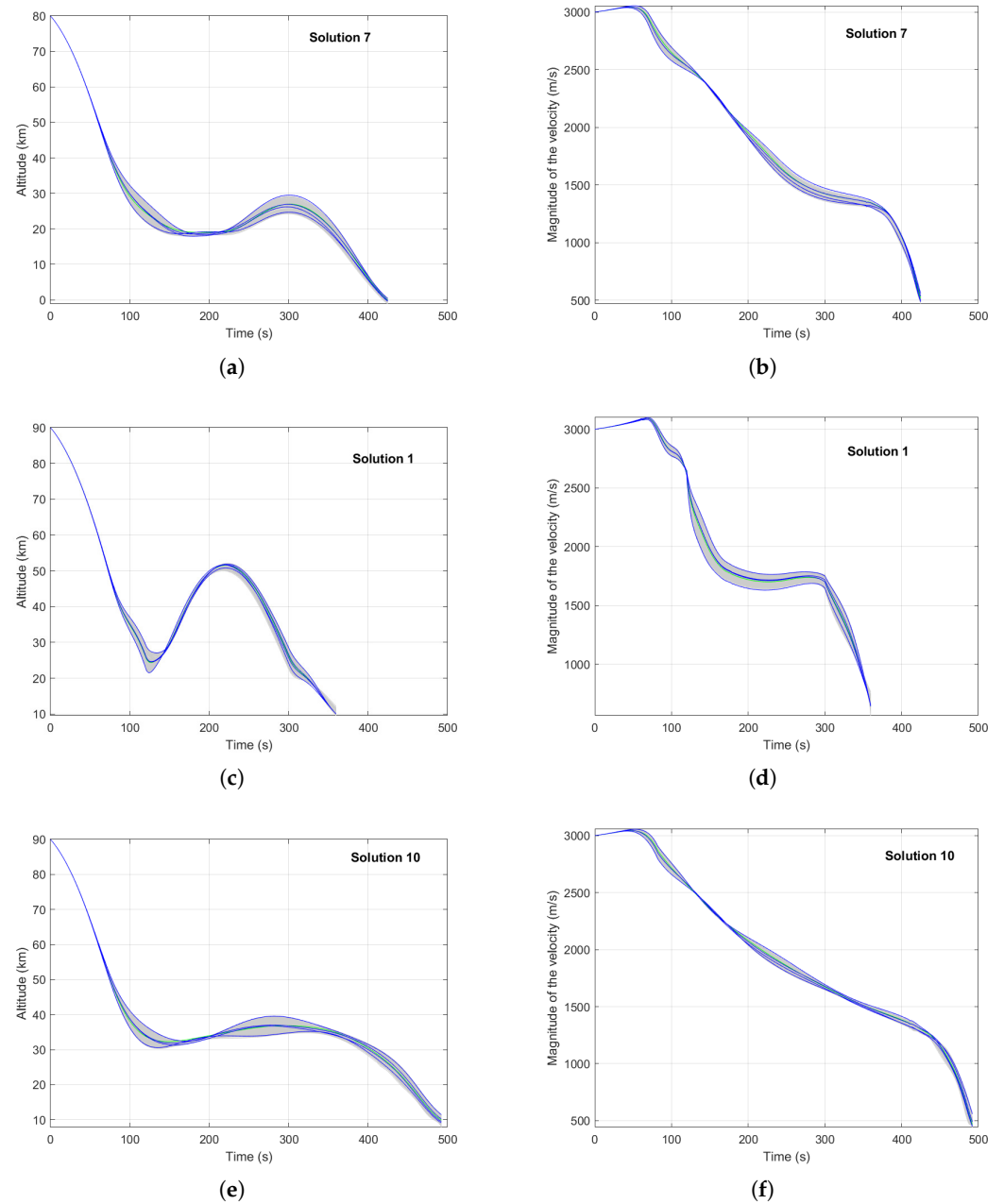
**Figure 9.** State variables of sigma points for the two extrema of the Pareto-optimal set (Solutions 1 and 10). (a) Altitude; (b) Velocity; (c) Flight path angle; (d) Heading angle.

### 6.3. Validation

A Monte Carlo analysis was run to validate the method and results. Starting from the Pareto-optimal set of 10 solutions, the solution was re-integrated using a different sample for the atmospheric model. A 100 different samples (for the atmospheric model) were taken for each Pareto-optimal solution, using the same sampling bounds as used to generate the sigma points (i.e., 1 standard deviation for each of the two atmospheric parameters, density and temperature).

Figure 10 shows the results of the Monte Carlo analysis for a middle set solutions (Solution 7, Figure 10a,b), and the two extrema (Solutions 1 and 10, Figure 10c,d). In each plot, the five sigma points are shown (Point 0 is green, and Points 2–4 in blue), with each of the 100 Monte Carlo runs shown as a grey line.

The values of the objective functions are compared in Table 2, looking at the optimised values against the mean values for  $J_1$  and  $J_2$  across the Monte Carlo runs.



**Figure 10.** Monte Carlo analysis for Solution 7 (a,b), Solution 1 (c,d) and Solution 10 (e,f). Green is the central Sigma point 0, blue are Sigma points 1–4, and grey lines are the 100 Monte Carlo samples. (a) Altitude, Solution 7; (b) Velocity, Solution 7; (c) Altitude, Solution 1; (d) Velocity, Solution 1; (e) Altitude, Solution 10; (f) Velocity, Solution 10.

**Table 2.** Comparison of the mean value of the objective functions across the 100 runs of the Monte Carlo analysis with the optimised values.

		Optimisation	Monte Carlo
Solution 1	Objective function $J_1$	$1.5375 \times 10^5$	$1.4696 \times 10^5$
	Objective function $J_2$	$2.1029 \times 10^{-14}$	$1.0542 \times 10^{-8}$
	Mean final altitude	10.00 km (target)	10.354 km
Solution 7	Objective function $J_1$	$3.6761 \times 10^4$	$3.5939 \times 10^4$
	Objective function $J_2$	$2.9916 \times 10^{-10}$	$5.7082 \times 10^{-10}$
	Mean final altitude	10.00 km (target)	9.9277 km
Solution 10	Objective function $J_1$	$3.2629 \times 10^4$	$3.2396 \times 10^4$
	Objective function $J_2$	$7.7731 \times 10^{-9}$	$1.8189 \times 10^{-8}$
	Mean final altitude	10.00 km (target)	9.9014 km

The results from Monte Carlo analysis are consistent with the results found using the unscented transform integrated into the multi-objective optimal control solver MODHOC. Small deviations can be seen towards the middle-end of the trajectory at altitudes below 50 km. The five sigma points chosen are all orthogonal, however the sample points for the Monte Carlo run are not so constrained (though the boundaries are the same). The absolute values for atmospheric parameters, in particular density  $\rho$  and pressure  $p$ , have higher values in lower atmosphere, so it is consistent with the expectation that this area would experience an increased impact, coupled with the larger standard deviation between 40–60 km.

## 7. Conclusions

This paper presented an extension of a memetic multi-objective optimisation algorithm MODHOC to perform optimisation under uncertainty. This was applied to a test case to determine a set of Pareto-optimal solutions for the descent trajectory of a first-stage spaceplane, based on the conceptual commercial launch vehicle Orbital 500R. The robust multi-objective optimisation traded off a performance objective, the induced accelerations on the vehicle corresponding to the dynamics loads, with a metric on the robustness of the trajectory to uncertainties on the atmospheric temperature, pressure and density.

Through a square root unscented transformation, different atmospheric models were generated for five sigma points. All the sigma points share the same control law, thus making the trajectory robust against model uncertainty. While only the first two statistical moments of the uncertain values were considered for this work, future work will account for higher order moments, making the resulting trajectory even more robust. A larger number of sigma points will also be required, however, and the resulting optimal control problem becomes progressively larger, requiring the use of large scale optimisation code.

**Author Contributions:** Conceptualization, L.A.R., C.A.M. and M.V.; Formal analysis, L.A.R., C.A.M. and M.V.; Funding acquisition, C.A.M. and M.V.; Investigation, L.A.R. and C.A.M.; Methodology, M.V.; Project administration, C.A.M., M.V.; Software, L.A.R.; Supervision, C.A.M. and M.V.; Writing—original draft, L.A.R. and C.A.M.; Writing—review & editing, C.A.M. and M.V. All authors have read and agreed to the published version of the manuscript.

**Funding:** This work has been partially funded by the UK Space Agency and European Space Agency General Support Technology Programme (GSTP) 4000126322/18/NL/LvH/zk on De-risking of the GNC of the Orbital 500-R, and ESA Network Partnership Initiative (NPI) 4000112763/14/NL/GLC/ats on Multi-Objective Hybrid Optimal Control Problems.

**Data Availability Statement:** The software tools DFET and MACS are available open source under the Strathclyde Mechanical and Aerospace Research Toolbox (SMART) for Optimisation and Optimal Control (o2c) hosted at <https://github.com/strath-ace/smart-o2c>, accessed on 15 November 2021.



**Acknowledgments:** The work here uses as a test case a variant of the Orbital-500R launch system that was created and developed by Orbital Access Ltd., and analysed through the Future Space Payload Launcher UK programme which included University of Strathclyde, BAE Systems, Fluid Gravity Engineering Ltd, University of Glasgow and others and was led by Orbital Access Ltd.

**Conflicts of Interest:** The authors declare no conflict of interest. The funders had no role in the design of the study; in the collection, analyses, or interpretation of data; in the writing of the manuscript, or in the decision to publish the results.

## References

- Maddock, C.A.; Ricciardi, L.; West, M.; West, J.; Kontis, K.; Rengarajan, S.; Evans, D.; Milne, A.; McIntyre, S. Conceptual design analysis for a two-stage-to-orbit semi-reusable launch system for small satellites. *Acta Astronaut.* **2018**, *152*, 782–792. [\[CrossRef\]](#)
- Ricciardi, L.A.; Vasile, M. MODHOC: Multi Objective Direct Hybrid Optimal Control. In Proceedings of the International Conference on Astrodynamics Tools and Techniques, Oberpfaffenhofen, Germany, 6–9 November 2018.
- Vasile, M. Multi-objective optimal control: A direct approach. In *Satellite Dynamics and Space Missions*; Springer: Berlin/Heidelberg, Germany, 2019; pp. 257–289. [\[CrossRef\]](#)
- Ross, I.M.; Proulx, R.J.; Karpenko, M. Unscented optimal control for space flight. In Proceedings of the International Symposium on Space Flight Dynamics, Laurel, MD, USA, 5–9 May 2014.
- Julier, S.J.; Uhlmann, J.K. *A General Method for Approximating Nonlinear Transformations of Probability Distributions*; Technical Report; Robotics Research Group, Department of Engineering Science, University of Oxford: Oxford, UK, 1996.
- Ross, I.M.; Proulx, R.J.; Karpenko, M.; Gong, Q. Riemann–Stieltjes optimal control problems for uncertain dynamic systems. *J. Guid. Control Dyn.* **2015**, *38*, 1251–1263. [\[CrossRef\]](#)
- Ozaki, N.; Campagnola, S.; Funase, R.; Yam, C.H. Stochastic differential dynamic programming with unscented transform for low-thrust trajectory design. *J. Guid. Control. Dyn.* **2018**, *41*, 377–387. [\[CrossRef\]](#)
- Longuski, J.M.; Guzmán, J.J.; Prussing, J.E. *Optimal Control with Aerospace Applications*; Springer: Berlin/Heidelberg, Germany, 2014.
- Ross, I.M.; Karpenko, M.; Proulx, R.J. Path constraints in stochastic and unscented optimal control: Theory, application and experimental results. In Proceedings of the American Control Conference, Boston, MA, USA, 6–8 July 2016. [\[CrossRef\]](#)
- Ricciardi, L.A.; Maddock, C.A.; Vasile, M. Direct solution of multi-objective optimal control problems applied to spaceplane mission design. *J. Guid. Control Dyn.* **2019**, *42*, 30–46. [\[CrossRef\]](#)
- Uhlmann, J.K. Dynamic Map Building and Localisation: New Theoretical Foundations. Ph.D. Thesis, University of Oxford, Oxford, UK, 1995.
- Van der Merwe, R.; Wan, E.A. The square-root unscented Kalman filter for state and parameter-estimation. In Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing, Salt Lake City, UT, USA, 7–11 May 2001. [\[CrossRef\]](#)
- Vasile, M. Finite Elements in Time: A Direct Transcription Method for Optimal Control Problems. In Proceedings of the AIAA/AAS Astrodynamics Specialist Conference, Toronto, ON, Canada, 2–5 August 2010. [\[CrossRef\]](#)
- Vasile, M.; Finzi, A. Direct Lunar Descent Optimisation by Finite Elements in Time Approach. *Int. J. Mech. Control* **2000**, *1*.
- Darehmiraki, M.; Farahi, M.H.; Effati, S. A Novel Method to Solve a Class of Distributed Optimal Control Problems Using Bezier Curves. *J. Comput. Nonlinear Dyn.* **2016**, *11*, 061008. [\[CrossRef\]](#)
- Betts, J.T. *Practical Methods for Optimal Control and Estimation Using Nonlinear Programming*; Advances in Design and Control; SIAM: Philadelphia, PA, USA, 2010.
- Chankong, V.; Haimes, Y.Y. *Multiobjective Decision Making: Theory and Methodology*; Courier Dover Publications: Mineola, NY, USA, 2008.
- Hillermeier, C. *Nonlinear Multiobjective Optimization*; International Series of Numerical Mathematics; Birkhäuser Basel: Basel, Switzerland, 2001. [\[CrossRef\]](#)
- Zuiani, F.; Kawakatsu, Y.; Vasile, M. Multi-objective optimisation of many-revolution, low-thrust orbit raising for Destiny mission. In Proceedings of the AAS/AIAA Space Flight Mechanics Conference, Kauai, HI, USA, 10–14 February 2013.
- Ricciardi, L.A.; Vasile, M. Improved archiving and search strategies for Multi-agent Collaborative Search. In Proceedings of the International Conference on Evolutionary and Deterministic Methods for Design, Optimization and Control with Applications to Industrial and Societal Problems, Glasgow, UK, 14–16 September 2015.
- Ricciardi, L.A.; Vasile, M.; Maddock, C. Global solution of multi-objective optimal control problems with Multi Agent Collaborative Search and Direct Finite Elements Transcription. In Proceedings of the IEEE Congress on Evolutionary Computation, Vancouver, BC, Canada, 24–29 July 2016. [\[CrossRef\]](#)
- Ricciardi, L.A.; Vasile, M.; Toso, F.; Maddock, C.A. Multi-objective optimal control of the ascent trajectories of launch vehicles. In Proceedings of the AIAA/AAS Astrodynamics Specialist Conference, Napa, CA, USA, 15–18 February 2016.
- Pascoletti, A.; Serafini, P. Scalarizing vector optimization problems. *J. Optim. Theory Appl.* **1984**, *42*, 499–524. [\[CrossRef\]](#)
- Eichfelder, G. *Adaptive Scalarization Methods in Multiobjective Optimization*; Springer: Berlin/Heidelberg, Germany, 2008. [\[CrossRef\]](#)
- Chasalow, S.D.; Brand, R.J. Algorithm AS 299: Generation of Simplex Lattice Points. *J. R. Stat. Soc. Ser. C (Appl. Stat.)* **1995**, *44*, 534–545. [\[CrossRef\]](#)

- 
26. Bruinsma, S.; Arnold, D.; Jäggi, A.; Sánchez-Ortiz, N. Semi-empirical thermosphere model evaluation at low altitude with GOCE densities. *J. Space Weather Space Clim.* **2017**, *7*, A4. [\[CrossRef\]](#)
  27. Picone, J.M.; Hedin, A.E.; Drob, D.P.; Aikin, A.C. NRLMSISE-00 empirical model of the atmosphere: Statistical comparisons and scientific issues. *J. Geophys. Res. Space Phys.* **2002**, *107*, SIA 15-1–SIA 15-16. [\[CrossRef\]](#)
  28. Bowman, B.; Tobiska, W.K.; Marcos, F.; Huang, C.; Lin, C.; Burke, W. A new empirical thermospheric density model JB2008 using new solar and geomagnetic indices. In Proceedings of the AIAA/AAS Astrodynamics Specialist Conference, Cambridge, MD, USA, 29 June–2 July 2008.
  29. Bruinsma, S. The DTM-2013 thermosphere model. *J. Space Weather Space Clim.* **2015**, *5*, A1. [\[CrossRef\]](#)
  30. Niederreiter, H. *Random Number Generation and Quasi-MONTE Carlo Methods*; SIAM: Philadelphia, PA, USA, 1992.
  31. Zipfel, P. *Modeling and Simulation of Aerospace Vehicle Dynamics, Second Edition*; AIAA Education Series; AIAA: Reston, VA, USA, 2007. [\[CrossRef\]](#)
  32. Stindt, T.; Merrifield, J.; Fossati, M.; Ricciardi, L.A.; Maddock, C.A.; West, M.; Kontis, K.; Farkin, B.; McIntyre, S. Aerodynamic database development for a future reusable space launch vehicle, the Orbital 500R. In Proceedings of the International Conference on Flight Vehicles, Aerothermodynamics and Re-entry Missions and Engineering, Monopoli, Italy, 30 September–3 October 2019.