Initial guess generation strategies for spaceplane trajectory optimisation

By Federico TOSO,[1)] and Christie MADDOCK[1)]

[1)] *Aerospace Centre of Excellence, University of Strathclyde, Glasgow, United Kingdom*

Trajectory optimisation for spaceplanes is a highly constrained problem due to long integration times, high energy environment and a broad spectrum of different flight conditions from sea level to above the Karman line. In this paper the generation of the initial guess for a gradient solver is presented on a sample problem: the ascent of a multi-stage reusable lifting body vehicle. Different multistart strategies to generate the first pool of results are presented and their performances analysed for computational time, objective function value and maximum violation of constraints. The effect on the convergence rate introducing weighting of the constraints, white noise and longer timestep is presented. An improvement in convergence rate is achieved with the introduction of noise at the cost of computational time.

Key Words: first guess, trajectory optimisation, spaceplane

Nomenclature

$\mathbf{c}$	:	optimisation constraint vector
C_D	:	drag coefficient
C_L	:	lift coefficient
D	:	drag
g	:	gravitational acceleration
h	:	altitude
I_{sp}	:	specific impulse
F_T	:	thrust
L	:	lift
M	:	Mach number
m	:	mass
n_p	:	number of phases
n_e	:	number of elements
n_c	:	number of control nodes
r	:	radius
t	:	time
$\mathbf{u}$	:	optimisation control vector
v	:	velocity
S_{ref}	:	aerodynamic surface reference area
q	:	dynamic pressure
α	:	angle of attack
γ	:	flight path angle
θ	:	longitude
λ	:	latitude
μ	:	bank angle
τ	:	throttle
χ	:	heading angle
ω	:	angular velocity

Subscripts

0	:	initial
gross	:	gross (vehicle mass)
r	:	radial
zfm	:	zero fuel mass
ϕ	:	tangential

1. Introduction

In the recent years, an increasing number of space access providers are focusing on recovery and re-usability of their launchers, using lifting surfaces to enable a glided return of the first stage. An extension of this design approach leads to spaceplanes, whether fully or partially flight-capable. The conceptual design and analysis of these vehicles present a challenging test case for multidisciplinary design optimisation, either of the vehicle for a fixed mission or of the mission options for a fixed vehicle design.

Multidisciplinary design and trajectory optimisation approaches have been used to create a set of tools tailored for the analysis of spaceplane design and operations. This paper will present and evaluate different optimisation approaches using a multi-stage to orbit, hybrid propulsion spaceplane as a test case. The spaceplane is designed for the small satellite market with payloads up to 1000 kg for insertion into LEO.

The specific vehicle under test is composed of a reusable first stage with a hybrid air-breathing and rocket propulsion system, and a rocket powered expendable second stage for orbit insertion after a coasting arc. The single objective of the optimisation is the maximisation of the payload mass deployed in the desired orbit assuming a fixed gross take-off mass. The trajectory optimisation uses a flight phase decomposition method accounting for the ascent after take-off with a propulsion mode change, second stage coast and powered ascent and orbital insertion of the payload, solving for an open loop control law. The problem is subjected to a number of boundary and path constraints on the peak acceleration loads and maximum dynamic pressure.

Previous studies on spaceplane trajectory optimisation have generally been performed with a sequence of deterministic gradient-based algorithms[1)] or following generalised guidance laws based on user-defined criteria, such as maintaining constant dynamic pressure[2, 3)] in the air-breathing phase of flight. Those approaches are computationally intensive or fail to capture some coupling effects that define

the spaceplane design leading to local minima or suboptimal solutions.[4] Additionally, the optimisation solver is highly tailored to the particular problem and requires a high level of expertise of the user on both the problem and the optimisation algorithm.

The optimisation approach uses a local gradient-based optimiser. The convergence of this class of optimisers is strongly dependant on the first guess. The goal of this paper is to evaluate different first guess strategies aimed at reducing the dependency on the user to tune the optimisation parameters to the specific system under analysis. The problems are restricted to the trajectory optimisation of space access vehicles, allowing for example, for different target orbit conditions and different vehicle configurations. The approach attempts to intelligently automate the process balancing computational run time with the required user level of expertise and the convergence rate.

In the following, results are shown for variations of a multi-start approach. The approaches compare different random generators for all the optimisation control variables and a more problem-specific variant with randomly-generated constant trajectory control values for each mission phase and state variables based on the user-defined bounds. The first guess approaches were then run using a reduced number of integration nodes, a strategy that iteratively adds noise to unsuccessful first guess vectors to improve the convergence rate and one that adds weight to the constraints.

2. System models

The vehicle is a partially reusable, two stage to orbit system with a target mission to deliver a payload to a 200 km altitude, circular, equatorial orbit. The system uses a hybrid air-breathing/rocket propulsion system for the first stage to reach the upper atmosphere, and a second stage rocket to inject the payload into the target orbit.

The first stage has a spaceplane configuration capable of flight using either the air breathing or rocket mode of the engine with lifting surfaces for improved manoeuvrability and performance during ascent, and a glided recovery. The second stage is a more conventional, expendable rocket stage with no lifting surfaces and thruster-based control as it is expected to primarily operate above the atmosphere.

The gross masses of both stages and the dry mass (or zero fuel mass) of Stage 1 are fixed. The trajectory is optimised to maximise the total mass delivered into orbit, i.e., the payload plus Stage 2. As the gross mass of the vehicle at separation is fixed, the optimiser will minimise the fuel consumed during the second stage and optimise the performance of the first stage in order for the system to reach the target orbit subject to the various operational constraints.

The gross vehicle mass at take-off is $m_{gross}(t = t_0) = 100$ Mg, and just after stage separation is $m_{gross}(t = t_{sep+}) = 1$ Mg. The zero fuel mass (ZFM) of the system just prior to stage separation can be as low as $m_{zfm}(t = t_{sep-}) \geq 30$ Mg. The first stage dry mass fraction of 30% is a conservative assumption compared to other single and two-stage to orbit vehicles such as Reaction Engines' Skylon at 18.5%,[5] X-33 at 26.3%,[6] or Hyperion at 15.4%.[7]

2.1. Flight dynamics and control

The vehicle stages are modelled as variable mass points in an Earth Centred Earth Fixed reference frame. The state vector $\mathbf{x} = [h, \lambda, \theta, v, \gamma, \chi, m]$ contains the vehicle mass m, position and velocity, defined by altitude h, velocity v relative to the reference frame, flight path angle γ measured from the local horizon, heading angle χ defined clockwise from the local North, and λ and θ as latitude and longitude coordinates on the Earth.[8]

$$\dot{v} = \frac{F_T \cos\alpha - D}{m} - g_r \sin\gamma + g_\phi \cos\gamma \cos\chi + \omega_E^2 r \cos\lambda \left(\sin\gamma \cos\lambda - \cos\gamma \sin\chi \sin\lambda\right) \quad (1)$$

$$\dot{\gamma} = \frac{F_T \sin\alpha \cos\mu + L}{mv} - \frac{g_r}{v}\cos\gamma - \frac{g_\phi}{v}\sin\gamma\cos\chi + \frac{v}{r}\cos\gamma + 2\omega_E \cos\chi\cos\lambda + \omega_E^2 \left(\frac{r}{v}\right)\cos\lambda\left(\sin\chi\sin\gamma\sin\lambda + \cos\gamma\cos\lambda\right) \quad (2)$$

$$\dot{\chi} = \frac{L\sin\mu}{mv\cos\gamma} - g_\phi \sin\chi - \left(\frac{v}{r}\right)\cos\gamma\cos\chi\tan\lambda + 2\omega_E\left(\sin\chi\cos\lambda\tan\gamma - \sin\lambda\right) - \omega_E^2\left(\frac{r}{v\cos\gamma}\right)\cos\lambda\sin\gamma\cos\chi \quad (3)$$

$$\dot{m} = -\dot{m}_p \quad (4)$$

where D and L are the aerodynamic lift and drag forces, and $\dot{m}_p$ is the total propellant mass flow rate.

The vehicle controls are the direction and magnitude of the thrust vector F_T through the modulation of the throttle, the angle of attack α and bank angle μ of the vehicle. The propulsion system is non-gimbled therefore the thrust vector is aligned with the longitudinal body axis of the vehicle. The coordinates are converted into the Earth Centred Inertial frame to calculate the Keplerian orbital parameters for the target orbit.

2.2. Environment

The Earth is modelled using the WGS84 spheroid model; the radius r_E from the centre to the surface is a function of the latitude λ. The planet rotates with a constant rate of $\omega_\text{E} = 7.292115 \times 10^{-5}$ rad/s. The gravitational force is divided in radial g_r and tangential components g_ϕ accounting for the spherical harmonics J_2, J_3 and J_4, the Earth radius $r_\text{E}(\lambda)$, the vehicle altitude above the Earth's surface h and the latitude.

The atmospheric model is based on the International Standard Atmosphere for altitudes below 84.852 km. The model is extended above this maximum altitude h_{max} by assuming a constant temperature $T(h > h_{max}) = 186.87$ K and using the exponential gas laws for pressure and density.

2.3. Aerodynamics

The aerodynamic model of the first stage is based on the published wind tunnel data of the X-34 unmanned suborbital reusable launch test vehicle.[12] The dataset contains values for C_L and C_D over a range of angles of attack $-4^\circ \leq \alpha \leq 24^\circ$ and Mach numbers $0.4 \leq M \leq 6$. This set of data points was used to create four surrogate models for $C_L(\alpha, M \leq 1)$, $C_L(\alpha, M > 1)$ and $C_D(\alpha, M \leq 1)$, $C_D(\alpha, M > 1)$ with the lowest order, bi-variate polynomial response surface guaranteeing a coefficient of determination ≥ 0.99.

To compute the equivalent aerodynamic surface for the scaled test vehicle here, the published wing loading value 655.59 kg/m^2 of the X-34 test vehicle[12] is kept constant. The surface area increases linearly with the vehicle mass m_{gross}, resulting in the value for the aerodynamic reference area $S_{ref} = 152.54$ m^2.

The second stage is a cylinder with no lifting or control surfaces. The aerodynamic model has a constant $C_D = 1$ and $C_L = 0$, and a reference surface $S_{ref} = 1$ m^2.

2.4. Propulsion

One of the advantages of an air-breathing propulsion system for launch vehicles is that it minimises the mass of onboard oxidiser required as the engine can use atmospheric oxygen. This leads to more aircraft-like vehicle configurations to also take advantage of aerodynamic lift in the lower atmosphere and at velocities up to hypersonic speeds.

The engine modelled for the first stage of the vehicle is based on predicted performance of SABRE (Synergistic Air-Breathing Rocket Engine), a liquid hydrogen fuelled combined cycle rocket engine under development by Reaction Engines Ltd.[5]

The maximum specific impulse in air breathing mode is $I_{sp,AB} = 9000$ s, with losses due to atmospheric pressure and Mach. The peak efficiency of the engine is at sea level and $M = 3.5$. While operating in rocket mode, the maximum specific impulse (when operating in a vacuum) is reduced to $I_{sp,R} = 450$ s, with expansion losses calculated considering a nozzle exit radius of 1 m. The maximum thrust at peak efficiency for both operating modes is 3 MN.

The second stage propulsion is modelled using the rocket equation with a nozzle exit radius of 0.5 m and maximum thrust in a vacuum of 10 kN. The vacuum I_{sp} is 450 s.

The throttle control $\tau \in [0, 1]$ proportionally scales the total propellant mass flow rate and thrust.

3. Transcription and optimisation

The mission is first divided in n_p discrete phases by the user. Each phase is characterised by its own set of boundary conditions, specific system or environmental models, integration algorithms, optimisation control bounds, node distribution schemes and linking conditions. Different constraints can be declared for each phase. As well as improving flexibility, this approach can be used to handle mathematical discontinuities such as aerodynamics across the transonic regime, engine operational mode switches and stage separations (for vehicle mass discontinuities).

The optimal control problem is formulated with a direct transcription, multiple shooting method. Each phase is divided into n_e shooting elements, and each element has n_c control nodes spaced with a user-defined distribution scheme in time. The value of the controls between nodes is obtained with interpolation. Each element is integrated forward in time based on a user-defined numerical integration scheme.

3.1. Application to problem

For the vehicle modelled in Section 2, the ascent trajectory was divided into $n_p = 3$ phases corresponding to:

- *Phase 1*: Stages 1+2 vehicle ascent in air-breathing engine mode starting from take-off ($t = t_0$)
- *Phase 2*: Stages 1+2 vehicle ascent in rocket engine mode until stage separation ($t = t_{sep}$)
- *Phase 3*: Stage 2 rocket ascent and injection into orbit

Each phase was divided into $n_e = 2$ elements, each with $n_c = 7$ control nodes distributed in time with Chebyshev polynomials. The values for the controls at the integration nodes are computed with a Piecewise Cubic Hermite Interpolating Polynomial. The integration is performed with a 4^{th} order Runge Kutta method with a fixed time step of 1 s.

The starting conditions for the test case are listed in Table 1. The final conditions require the payload, and second stage, to be injected into the target Keplerian orbit with a semi-major axis of ($r_E + 200$) km and eccentricity $e = 0$ (i.e., circular). The trajectory was designed to be planar, within the equatorial plane. The system is launched from the equator, targeting an orbit with an inclination of 0°. This also means the bank angle should be $\mu(t) = 0$.

Table 1. Initial state variables at $t = t_0$.

Altitude, h_0	10 km
Latitude, λ_0	0°N
Longitude, θ_0	0°E
Velocity, v_0	150 m/s
Flight path angle, γ_0	0°
Heading angle, χ_0	90° (due east)
Vehicle mass, $m_{gross}(t_0)$	100×10^3 kg

Nonlinear constraints are added to match the state and control variables between each of the multiple shooting elements, and between the phases subject to the defined linking criteria (e.g., for stage separation, there are no matching constraints on the vehicle mass). The lower and upper bounds for these are given in Table 2.

Path constraints are set to limit the maximum dynamic pressure q and accelerations on the longitudinal (x) and axial (z) body axes of the vehicle.

$$\max q(t) \leq 20 \text{ kPa} \tag{5}$$

$$\max a_x(t) \leq 6 g_0 \tag{6}$$

$$\max a_z(t) \leq 6 g_0 \tag{7}$$

The objective function is the maximisation of the vehicle mass injected into the target orbit. The value is normalised over 1000 kg, equivalent to the upper bound on the final mass of the second stage.

Table 2. Bounds on the state and control variables.

	Phase 1	Phase 2	Phase 3
States			
Altitude, h (km)	[1, 100]	[10, 200]	[10, 200]
Velocity, v (km/s)	[0.1, 2.5]	[1, 8]	[1, 8]
Flight path angle, γ (deg)	[−60, 90]	[−60, 90]	[−60, 90]
Heading angle, χ (deg)	[−180, 180]	[−180, 180]	[−180, 180]
Latitude, λ (deg N)	[−90, 90]	[−90, 90]	[−90, 90]
Longitude, θ (deg E)	[−180, 180]	[−180, 180]	[−180, 180]
Vehicle mass, m (Mg)	[30, 100]	[30, 100]	[0.1, 1]
Controls			
Angle of attack, α (deg)	[−4, 24]	[−4, 24]	[−30, 30]
Throttle, τ	[0.8, 1]	[0, 1]	[0, 1]

The constraints are scaled based on the difference between the upper and lower phase-specific bounds for matching state and control vectors, and the absolute maximum values for the path constraints.

The optimal control problem is solved with the sequential quadratic programming (SQP) solver of the single objective, nonlinear constraint gradient based `fmincon` algorithm in the optimisation toolbox of MATLAB. The constraint violation tolerance of the solver is set to 10^{-4} for all cases.

Given the initial conditions, the time duration of the element and the open loop control law, a specific trajectory is obtained through integration. Thus, each of the multiple shooting element is defined by: the initial states from which the trajectory element starts, the values of the trajectory controls at every node, and the duration of the integration. These 3 arrays are repeated for every element and comprise the optimisation vector $\mathbf{u}$. Cases such as the first element, where the starting condition is fixed by the user, are excluded.

Prior to the local optimisation, the first guess for the optimisation vector $\mathbf{u}$ is scaled relative to the lower and upper bounds, such that $\mathbf{u} \in [0, 1]$.

4. First guess generation

For many random generation-based first guess strategies, a heavily constrained problem can often start from an initial infeasible point that violates the constraints and reduces the convergence rate, or close to a region of local optimality that is sub-optimal compared to the global optimum. The sensitivity of the solution to the first guess is a well known issue for local, gradient-based optimisation algorithms.

An experienced choice of variable bounds, educated first guess and well tuned transcription settings can increase the probability of convergence, and optimality of the solution. One of the problems with this approach is that tight bounds can bias or force the algorithm towards a local optimum with limited exploration of the full search space. Furthermore, it can be beneficial to understand any unforeseen exploitation of the models to feedback into a better problem definition. A first guess strategy that does not require a high level of experience on the side of the user can be beneficial to study different configurations and can save time in the problem set-up and run.

This paper shows some results within a study into the comparison of semi-automated first guess generation routines, specific for trajectory optimisation of spaceplane launch systems. The goal is to reduce the dependency on the expertise level (and possibly bias) of the user, while increasing the exploration of the search space and convergence rate, and at the same time, minimising the complexity and computational run time.

The first guess strategies analysed here are all variations of the multi-start method. The first comparison is a baseline looking at randomly generated values for the entire optimisation vector, using Latin Hypercube Sampling (LHS) and the Mersenne Twister pseudo-random number generator. A variation is then compared that balances some problem-specific knowledge with automation: the matching state vectors are set to the mean value between the lower and upper bounds, while the trajectory controls are held constant within each phase at a randomly generated value. A second comparison looks at methods for improving the convergence rate for LHS, altering the weighting of the constraints relative to the objective function, changing the integration timestep and introducing white noise into the first guess vectors for runs that failed to converge.

4.1. Random generated guesses

The first approach uses randomly generated values (within the specified bounds) for all elements in $\mathbf{u}$. A population size is specified dictating the number of first guesses to generate. Each first guess $\mathbf{u}_{fg}$ is then locally optimised with the results stored in an archive. The archive is ranked according to value of the objective function and constraint satisfaction.

Two different methods were used to generate the randomised values. The first uses Mersenne Twister method in the pseudorandom generator in Matlab. All the runs have been performed sequentially with the same, default random seed. The second approach creates the population through Latin Hypercube Sampling. This method is chosen to more evenly spread the starting guesses within the search space and achieve a more uniform sampling.

A third strategy mixes some known physics of the general problem with random generation. The trajectory controls per phase are held constant for the entire phase at a randomly generated value between the upper and lower

user-defined control bounds. The values for the state vector between elements and time of flight are set as the mid-point between user-specified lower and upper bounds.

A population size of 100 was used for all random-based approaches, and was run 3 times, leading to a total comparison of 300 first guesses.

4.2. Latin Hypercube Sampling comparison

Using first guesses from the LHS, other techniques were compared to increase the convergence rate. Three of such techniques are compared here: 1- a weighting factor is added to bias the optimiser towards constraint satisfaction over finding a better objective function, 2- the integration timestep was increased to allow for a computationally faster initial optimisation, with a smaller subset of potentially feasible solutions re-optimised with the original timestep, and 3- random noise was introduced to the non-converged first guess vectors to produce a second population that was then optimised.

A smaller population with 32 individuals was selected to analyse the effect of the introduction of bias on the constraint vectors, whose values are multiplied by 10^4 while the cost function is not altered.

Another approach tested the introduction of white noise with varying intensities applied to the whole population after a first pass through the optimisation routine and normalised between 0 and 1. For each first guess, noise amplitudes of 10^{-7}, 10^{-5} and 10^{-3} are added, starting from the smallest one and proceeding in ascending order until an optimised solution is found.

An evaluation of the constraint violation is done through post-processing after the introduction of white noise to retain solutions that violate the optimiser tolerances by factors of 10 relative to the optimiser constraint tolerance of 10^{-4} (e.g., 10^{-3}, 10^{-2}).

Due to the long time needed to reprocess the whole population, effectively linearly scaling the running time by the number of iterations required to achieve convergence, another approach is tested. By using a longer timestep in the first pass, the initial computation before the introduction of the white noise takes less time.

5. Results and discussion

All the computations have been performed on the same desktop class machine, with 8GB of RAM and a Intel i7-3770 CPU at standard clock speed, using Matlab version 2016a running with all 4 physical cores in the parallel pool.

The focus of the presented strategies is the minimisation of the user input required for the guess generation, aiming to automate the process as much as possible without sacrificing computational efficiency.

All the cases analysed have been post-processed to evaluate the convergence rate and the best objective function value with relaxed constraints.

5.1. Population generation strategies

Table 3 summarises the results obtained from the populations with 100 individuals for the random, LHS and constant control law approaches. The case that satisfies the constraints within the tolerance of the problem and with the best objective function value of $m_f = 881.3009$ kg, has been found with the random constant control approach. The controls of the angle of attack and throttle, and the altitude and velocity time histories for this case are plotted in Fig. 1. This result shouldn't be used as an absolute indicator of the goodness of the approach due to the small sample size of the populations analysed in this study.

The results can be post-processed to examine the potential feasibility of solutions that did not meet the strict optimisation constraint tolerance. This analysis is performed by recomputing the constraint violations of all individuals of the populations.

All the cases analysed have a low convergence rate and increasing the constraint tolerance gives access to a bigger pool of solutions. By selecting the individuals with $\max(\mathbf{c}) \leq 10^{-3}$, the other 2 approaches have their best objective function within 1% of difference to the random constant control case.

The second set of results compares different approaches applied to the same initial population generation method, introducing refinement strategies. The number of individuals in each set is reduced to 32, to compensate for the extra computational time needed to run the cases with white noise.

The best result obtained is from the case with longer timestep and noise, with the best objective function value of $m_f = 881.3776$ kg, however this population had also the longest average runtime, despite the longer timestep used for the first optimisation run.

In the cases where white noise is applied, the difference in convergence rate between the tolerances of $\max(\mathbf{c}) \leq 10^{-4}$ and $\max(\mathbf{c}) \leq 10^{-3}$ is due to the halting the noise routine when the second constraint tolerance is satisfied. Looking at the solutions that satisfy the condition $\max(\mathbf{c}) \leq 10^{-3}$, in all cases the convergence rate increases from 6.25% without noise to 34.375% with noise, highlighting the benefits of this approach.

While the small population size and the single problem analysed prevent from selecting the best approach on absolute terms, they're useful indicators to continue the development of an automatic generation of the initial guess.

Both the post-processing of results by relaxing the constraints and the introduction of white noise show improvement on the convergence rate and can be used to generate good initial guesses even when analysing small populations.

References

1) Yokoyama, N: Trajectory optimization of space plane using genetic algorithm combined with gradient method, International Congress of Aeronautical Sciences, 2002.
2) Markl, Walter A: An initial guess generator for launch and reentry vehicle trajectory optimization, Institut für Flugmechanik und Flugregelung der Universität Stuttgart, Germany, 2001.
3) Young, D., Kokan, T., Clark, I., Tanner, C., and Wilhite, A.: Lazarus- A SSTO Hypersonic Vehicle Concept Utilizing RBCC and HEDM Propulsion Technologies, 14th AIAA/AHI Space Planes and Hypersonic Systems and Technologies Conference, 2006.

Table 3. Summary of the results obtained with population generated with random values, LHS, and random constant control laws.

	Random	LHS	Random constant control
Population size	100	100	100
Mean duration/run [s]	445.84	370.49	422.04
$\max(c) \leq 10^{-4}$			
Convergence rate	3% (3/100)	4% (4/100)	3% (3/100)
Best objective function value [kg]	862.5741	603.7174	881.3009
$\max(c) \leq 10^{-3}$			
Convergence rate	8% (8/100)	8% (8/100)	10% (10/100)
Best objective function value [kg]	878.9514	876.5926	881.3009
$\max(c) \leq 10^{-2}$			
Convergence rate	21% (21/100)	22% (22/100)	28% (28/100)
Best objective function value [kg]	878.9514	876.5926	881.3009

Table 4. Summary of the results on LHS, noise, constraint:objective function weighting (bias), and integration timestep increase.

	LHS	LHS, noise	LHS, bias	LHS, bias, noise	LHS, $3t_{step}$, noise
Population size	32	32	32	32	32
Mean duration/run [s]	438.94	1211.41	261.82	1112.77	1350.59
$\max(c) \leq 10^{-4}$					
Convergence rate	6.25% (2/32)	0% (0/32)	0% (0/32)	9.34% (3/32)	6.25% (2/32)
Best objective function value [kg]	670.7514	-	-	877.3658	881.3776
$\max(c) \leq 10^{-3}$					
Convergence rate	6.25% (2/32)	34.38% (11/32)	6.25% (2/32)	34.38% (11/32)	34.38% (11/32)
Best objective function value [kg]	670.7514	879.0903	867.9605	877.7117	881.3776
$\max(c) \leq 10^{-2}$					
Convergence rate	25% (8/32)	37.5% (12/32)	6.25% (2/32)	46.88% (15/32)	40.63% (13/32)
Best objective function value [kg]	877.862	879.0903	867.9605	877.7117	881.3776

4) Forbes-Spyratos, S. O., Kearney, M. P., Smart, M. K., and Jahn, I. H.: Trajectory Design of a Rocket-Scramjet-Rocket Multi-Stage Launch System, 21st AIAA International Space Planes and Hypersonics Technologies Conference, 2017.
5) Varvill, R., Bond, A.: A comparison of propulsion concepts for SSTO reusable launchers, *Journal of the British Interplanetary Society* **56**(3/4), 2003, pp. 108-117.
6) Letchworth, G.: X-33 reusable launch vehicle demonstrator, spaceport and range, AIAA SPACE 2011 Conference & Exposition, 2011
7) Olds, J.: Hyperion-An SSTO vision vehicle concept utilizing rocket-based combined cycle propulsion, 9th International Space Planes and Hypersonic Systems and Technologies Conference, 1999.
8) Tewari, A: *Atmospheric and space flight dynamics*, Springer, 2007.
9) Toso, F. and Maddock, C.: Deployed payload analysis for a single stage to orbit spaceplane, International Astronautical Conference, Guadalajara, Mexico, 2016.
10) Toso, F., Maddock, C. and Minisci, E: Optimisation of Ascent Trajectories for Lifting Body Space Access Vehicles, Space transportation solutions and innovations symposium, International Astronautical Congress, Jerusalem, Israel, 2015.
11) Balesdent, M., Bérend, N., Dépincé, P. and Chriette, A:, Multidisciplinary Design Optimization of Multi-Stage Launch Vehicle using Flight Phases Decomposition, *International Journal for Simulation and Multidisciplinary Design Optimization* **4**(3-4), 2010, pp. 117–125.
12) Brauckmann, G.: X-34 Vehicle Aerodynamic Characteristics, *Journal of Spacecraft and Rockets* **36**(2), 1999, pp. 229–239.

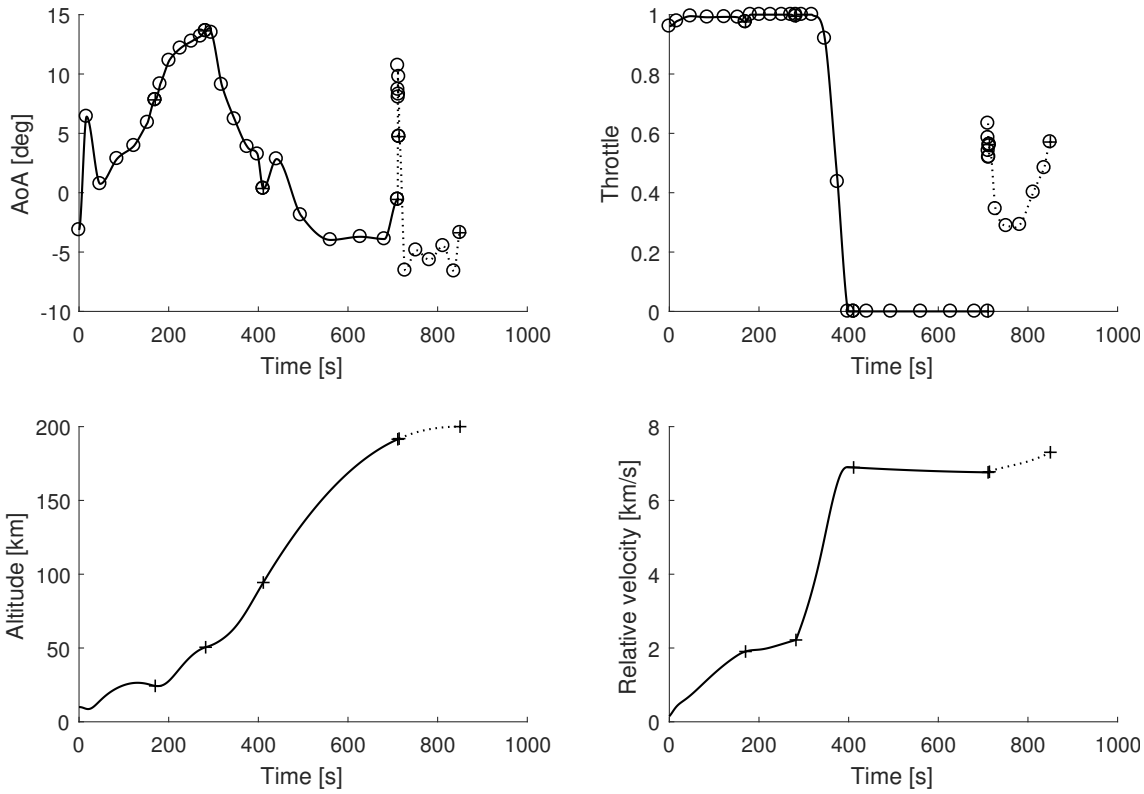

Fig. 1. *Top*: Angle of attack and throttle controls time histories of the solution with highest value of objective function. *Bottom*: Time history of the altitude and relative velocity for the same trajectory. First stage flight identified by a solid line, second stage be a dotted line. Round markers identify the control nodes, the junction points between multiple shooting elements are identified by plus signs.