Mission-Driven Optimization of Vehicle Geometry and Trajectory for Hypersonic Glide Vehicles

Bradley Pascoe[a,*], Andrew Lock[a], Jack Williamson[a], Robert Watt[a], Ingo Jahn[a]

[a]*University of Southern Queensland, Springfield, 4300, Queensland, Australia*

Abstract

Hypersonic vehicles operate in a complex regime where thermal, aerodynamic, and structural considerations constrain possible vehicle designs and trajectories, motivating multidisciplinary analysis and optimization (MDAO) of these coupled systems. Such vehicles are frequently optimized at single or multiple design point conditions, which can give poor performance at off-design conditions experienced during flight. Control co-design instead optimizes a vehicle and its trajectory together for a specified mission. We present a gradient-based control co-design framework for hypersonic glide vehicles, applied to two missions: a maximum-range sprint and a maximum-range flight through spatial obstacles. The co-designed vehicles outperform a point-designed vehicle for their respective missions, flying 13% farther on the sprint mission and 5% farther on the obstacle mission than the point-design vehicle. Each converges to a distinct geometry parameterized by 31 design parameters that set its aerodynamic behavior and mass, with the optimizer trading one against the other as each mission requires. Fidelity escalation also proves beneficial: vehicles co-designed with surface-inclination and computational fluid dynamics (CFD) aerodynamics are more physically similar than vehicles co-designed for different missions, and warm-starting a CFD co-design optimization from a low-fidelity design reduces the compute time to 20 %. The framework extends readily to thermal, structural, and propulsive modeling, and the gradient-based approach keeps such high-dimensional problems tractable.

Keywords: Hypersonic glide vehicle, Vehicle optimization, Control co-design, Computational fluid dynamics, Trajectory optimization

1. Introduction

The high speeds of hypersonic flight provide the main benefit of the regime but also cause the complications that make it so challenging to operate in. Hypersonic speeds could allow for faster international travel for civilian passengers or enable cheaper access to space by using air-breathing engines as opposed to rockets. For defensive purposes, hypersonic capabilities are a growing field of interest. Maneuverable hypersonic vehicles not only travel faster but also provide reduced warning times to detection systems and can be tougher to intercept. Designing and operating a hypersonic vehicle is a complicated process as the vehicle may operate over a wide range of Mach numbers and altitudes, especially if it is an accelerator or hypersonic glide vehicle (HGV). Such a diverse flight envelope makes it difficult to optimize the vehicle's aerodynamic performance while also maintaining stability and control. The hypersonic regime is also associated with large dynamic pressures and thermal loads, which need to be managed. Hypersonic vehicles need to be highly integrated to manage the extreme structural and thermal loads, as well as generate net thrust for air-breathing vehicles.

The coupling of aerodynamics, structures, thermodynamics, and propulsion requires using multidisciplinary analysis and optimization (MDAO) if the optimum vehicle is to be obtained. The optimization of the aerodynamic performance is often performed standalone for waverider configurations. The hypersonic waverider was first envisaged as a re-entry glider, designed to efficiently generate lift to reduce deceleration loads and heating rates [1]. The analysis and optimization of waverider designs have developed in several ways. For the analysis and parameterization, several new methods have been developed. While the original waverider analysis assumed a 2D planar flow field creating a "Λ" shape design [1], swapping to a 2D axisymmetric flow field from a cone [2] or a 3D osculating cone flow [3] allowed for greater volumetric efficiency in the vehicle designs. Likewise, parameterization approaches have improved compared to the original straight line design [1]. Class-shape transformation has been applied both to the conceptual design of gliding configurations [4] and, in an improved form, to the aerodynamic shape optimization of hypersonic vehicles [5], while B-spline parameterizations have been used to optimize HGVs subject to stringent loading constraints [6]. More detailed documentation of such developments can be found in review papers [7]. The primary limitation in the majority of these approaches is the single-point evaluation, causing degradation in performance when operated away from the design point condition (typically specified by Mach number, angle of attack, and altitude). A hypersonic cruise vehicle with propulsion can maintain a steady Mach number, but HGVs and

*Corresponding author.
Email address: bradley.pascoe@unisq.edu.au (Bradley Pascoe)

accelerators traverse a wide flight envelope, so a single-point objective is at best a proxy metric for the mission performance. There are efforts to create wide-speed waveriders [8] that consider the performance of the waverider over a set of discrete Mach numbers or range. While this is an improvement over point design, it fails to consider the operation of the vehicle.

Vehicles are inherently valued for their dynamics, that is, their trajectory or movement through space. Unlike the literature on optimization of HGVs, the research for launch vehicle design systems is more likely to incorporate trajectory analysis than detailed aerodynamics, often relying on empirical or semi-empirical methods [9]. There are two approaches to incorporating trajectories into the evaluation of the vehicle undergoing optimization. The first is the evaluation of the vehicle's trajectory for fixed control inputs or a specified control law. This involves using a forward integrator to determine the performance of the specified vehicle configuration, often specifying the angle of attack as a constant [10] or a simple function of the flight conditions [11, 12]. A better approach is to solve the optimal control problem for the trajectory. Letting an optimizer decide what the optimum control inputs (open-loop) or control gains (closed-loop) are improves the vehicle's performance and allows its potential to be fully realized. This process, which optimizes both vehicle design parameters and the trajectory controls, is labeled 'control co-design' [13, 14, 15]. By considering both aspects together, the optimal combination can be identified, which is unlikely to be found using a sequential optimization (vehicle then trajectory optimization). For the application to HGVs, there are several recent approaches. Coulter et al. [16] used a Bayesian optimizer to optimize an X-43 model by using two deformation modes to improve trajectory distance. Cangelosi et al. [17] optimized a waverider model for two design parameters for maximum range in a differentiable configuration, and compared the performance to the starting nominal design. Ma et al. [18] optimized an HTV-2-like vehicle defined by seven parameters, using a gradient-based approach to improve the vehicle for maximum range from the nominal design. Further analysis included a propulsion system for the comparison of the two co-designed vehicles, one with and one without propulsion. Co-designing with propulsion enabled a 3.87% improvement in range compared to adding propulsion to the vehicle co-designed without it. Comparing the benefits of a fully coupled and an iterative optimization approach for vehicle and closed-loop trajectory control analysis, Pan et al. [19] performed a multi-objective optimization using an evolutionary algorithm. Co-design analysis has also been applied to boost-glide systems to show the benefit of including the boost stage for the design of a boost-glide vehicle [20, 21, 22].

Despite this existing research, there are several limitations. First, the benefit of co-design over conventional practice has not been quantified for the HGV. Most investigations either compare two co-designed vehicles or compare to a nominal vehicle and not a vehicle produced by the point-design process that co-design is intended to replace. Second, aerodynamic fidelity and degrees of freedom of the design are a trade-off against one another. Studies that employed computational fluid dynamics (CFD) simulations for aerodynamics carry far fewer design parameters than those which relied on cheaper methods. The objective of this paper is to clearly demonstrate the utility of the control co-design methodology, whereby optimizing a vehicle for a specific mission produces superior results compared to a vehicle optimized at a single design point. This analysis is then extended to show that fidelity escalation is a viable approach for hypersonic vehicle co-design, mitigating the computational cost of high-fidelity methods by warm-starting the optimization from the converged solution of a lower-fidelity co-design. To this end, three vehicles are optimized using low-fidelity, local surface inclination aerodynamics: one for a fixed point-design condition and two for distinct sprint and obstacle-avoidance missions. The mission-optimized vehicles are shown to outperform the point-designed vehicle when each is evaluated on its respective mission. The sprint co-design is then repeated using CFD-based aerodynamics, both cold-started from a nominal initial design and warm-started from the converged low-fidelity solution, to assess the design convergence and to quantify the computational benefit of fidelity escalation. Section 2 describes the co-design formulation, and Section 3 details the optimization solver and the vehicle, aerodynamic, and trajectory models used. Section 4 specifies the three design problems in terms of those models. Section 5 presents low-fidelity co-design results for the point-design, sprint, and obstacle-avoidance missions. Section 6 extends this to CFD-based co-design of the sprint mission, comparing the cold- and warm-started optimization. Section 7 summarizes the key findings and contributions of this work.

2. Co-Design Formulation

Control co-design is a specific form of multi-disciplinary analysis and optimization which seeks to optimize the performance of a system for a time-dependent mission. This requires the inclusion of modeling the time-dependent dynamics of the system with respect to some control, whether open-loop or closed loop. The controlled system in co-design is commonly referred to as the 'plant'; however, given the context of analyzing hypersonic glide vehicles, the system will be generally referred to as a vehicle. There are two frameworks which can be employed for control co-design: simultaneous and nested approaches. The simultaneous approach uses a single optimizer which optimizes both the design variables for the vehicle ($\boldsymbol{x}_\mathrm{v}$) as well as the controls ($\boldsymbol{u}$) and/or states ($\boldsymbol{x}_\mathrm{tr}$) for the trajectory. This is expressed as a single optimization problem seeking to find the trajectory variables, trajectory controls, mission time, and vehicle design variables which minimize the objective cost $\mathcal{J}$ for a specific mission,

$$\min_{\boldsymbol{x}_\mathrm{tr},\boldsymbol{u},t_f,\boldsymbol{x}_\mathrm{v}} \quad \mathcal{J}\left(\boldsymbol{x}_\mathrm{tr},\boldsymbol{u},t_f,\boldsymbol{x}_\mathrm{v}\right) \tag{1a}$$

$$\text{s.t.} \quad \boldsymbol{h}_\mathrm{v}\left(\boldsymbol{x}_\mathrm{v}\right) \geq 0 \tag{1b}$$

$$\dot{\boldsymbol{x}}_\mathrm{tr} = \boldsymbol{f}\left(t,\boldsymbol{x}_\mathrm{tr},\boldsymbol{u},\boldsymbol{x}_\mathrm{v}\right) \tag{1c}$$

$$\boldsymbol{h}_\mathrm{tr}\left(t,\boldsymbol{x}_\mathrm{tr},\boldsymbol{u},\boldsymbol{x}_\mathrm{v}\right) \geq 0 \tag{1d}$$

$$\boldsymbol{h}_{\mathrm{tr}_0}\left(t_0,\boldsymbol{x}_\mathrm{tr}(t_0),\boldsymbol{u}(t_0),\boldsymbol{x}_\mathrm{v}\right) \geq 0 \tag{1e}$$

$$\boldsymbol{h}_{\mathrm{tr}_f}\left(t_f,\boldsymbol{x}_\mathrm{tr}(t_f),\boldsymbol{u}(t_f),\boldsymbol{x}_\mathrm{v}\right) \geq 0. \tag{1f}$$

The optimization is constrained by five different constraints. Equation (1b) represents the static constraints, which are independent of the dynamics and are functions of just the vehicle design variables. These constraints can include physical limitations such as volume, shape or mass, or constraints evaluated at specific conditions such as lift-to-drag or stability at a flight condition. The four remaining constraints are applied to the dynamic system, affecting the feasibility of the allowed trajectories. Equation (1c) is a consistency condition on the dynamics, specifying the time evolution of the trajectory states and ensuring that the trajectory correctly evolves in time as determined by the governing physics. The remaining three constraints (Eqs. (1d)–(1f)) are applied to the controls and states of the dynamics, with the last two being specifically applied to the trajectory variables at the initial time t_0 and the final time t_f, respectively. These trajectory constraints can be used to ensure the vehicle operates within a safe flight envelope and airspace, such as by specifying operating height and speed limits. Specifying the initial trajectory variable constraints is equivalent to specifying the initial conditions of the trajectory optimization problem, such as the position and velocity of the vehicle. Final trajectory variable constraints instead specify the conditions the vehicle must reach at the end of the mission. If a vehicle cannot satisfy the trajectory constraints, whether at the initial or final time or along the path, then no feasible trajectory exists, which is problematic for nested co-design implementations.

Optimizing over all of these variables for trajectory and vehicle can create a high-dimensional space to search. It is possible to separate the optimization problem in Eq. (1) into a bi-level configuration to help reduce the outer dimensionality. This is the nested co-design approach: an outer optimizer handles the vehicle design variables, and a nested trajectory optimizer is used to identify the optimal trajectory for the specific vehicle. The outer-loop problem seeks to obtain the minimum objective cost and corresponding design variables,

$$\min_{\boldsymbol{x}_\mathrm{v}} \quad \mathcal{J}\left(\boldsymbol{x}_\mathrm{v}\right) \tag{2a}$$

$$\text{s.t.} \quad \boldsymbol{h}_\mathrm{v}\left(\boldsymbol{x}_\mathrm{v}\right) \geq 0. \tag{2b}$$

This formulation is identical to a standard point-design or trajectory-unaware formulation. The calculation of the cost is the driving difference, as a point-design problem will use the static properties of the vehicle to calculate the cost, estimating the mission performance with some chosen design metrics. The nested co-design problem instead optimizes the trajectory for the vehicle with the dynamic constraints. This is an optimal control problem, where the optimizer has control over the dynamic states and controls in order to obtain the optimum performance for the vehicle specified by the outer-loop optimizer,

$$\mathcal{J}(\boldsymbol{x}_\mathrm{v}) \coloneqq \min_{\boldsymbol{x}_\mathrm{tr},\boldsymbol{u},t_f} \quad \mathcal{J}_\mathrm{tr}\left(\boldsymbol{x}_\mathrm{tr},\boldsymbol{u},t_f;\,\boldsymbol{x}_\mathrm{v}\right) \tag{3a}$$

$$\text{s.t.} \quad \dot{\boldsymbol{x}}_\mathrm{tr} = \boldsymbol{f}\left(t,\boldsymbol{x}_\mathrm{tr},\boldsymbol{u};\,\boldsymbol{x}_\mathrm{v}\right) \tag{3b}$$

$$\boldsymbol{h}_\mathrm{tr}\left(t,\boldsymbol{x}_\mathrm{tr},\boldsymbol{u};\,\boldsymbol{x}_\mathrm{v}\right) \geq 0 \tag{3c}$$

$$\boldsymbol{h}_{\mathrm{tr}_0}\left(t_0,\boldsymbol{x}_\mathrm{tr}(t_0),\boldsymbol{u}(t_0);\,\boldsymbol{x}_\mathrm{v}\right) \geq 0 \tag{3d}$$

$$\boldsymbol{h}_{\mathrm{tr}_f}\left(t_f,\boldsymbol{x}_\mathrm{tr}(t_f),\boldsymbol{u}(t_f);\,\boldsymbol{x}_\mathrm{v}\right) \geq 0. \tag{3e}$$

where $\mathcal{J}_\mathrm{tr}$ is the trajectory cost for the fixed design $\boldsymbol{x}_\mathrm{v}$. Both the nested and simultaneous formulations pose the same optimization problem, differing in how the search over the vehicle and trajectory variables is partitioned rather than in the solution being sought. For a convex problem the two approaches converge to the same solution [15]; for non-convex problems, such as those considered here, either approach may converge to a local optimum, and the two need not find the same one. The choice of which control co-design implementation to use is problem-dependent. The cost per outer iteration is increased for the nested co-design due to inclusion of the trajectory optimization; however, the reduced dimensionality of the outer optimization problem should allow the system to converge in fewer outer iterations. The nested approach is therefore beneficial for problems where the nested trajectory optimization is relatively fast compared to the vehicle evaluation [15]. A further advantage of the nested approach is that every outer iterate is a realizable design: each vehicle is paired with its optimal, dynamically feasible trajectory, so the objective history directly tracks the improvement in mission performance throughout the optimization. In the simultaneous approach, the dynamics are generally satisfied only at convergence, so intermediate iterates do not correspond to achievable missions.

One of the difficulties with the nested co-design approach is calculating the performance when there is no feasible trajectory due to constraint violations. For the simultaneous approach, the optimizer is aware of the trajectory constraint violations as the sole optimizer handles all constraints. This is not the case for the nested approach which only communicates the mission performance to the outer optimizer. One approach to account for this is to incorporate additional static constraints to limit the design variable feasibility space. Ideally, these static feasibility constraints reduce the design variable space to a subspace in which a feasible trajectory can always be obtained; however, creating such constraints is not always possible [15]. An alternative approach is to communicate the constraint violation to the outer optimizer by implementing 'soft' constraints. In this approach, constraints may be violated but the violation is scaled and added to the objective cost. The outer optimizer, seeking to reduce the cost, will move toward the design space where feasible trajectories can be obtained and violations are avoided [23]. The present work takes the former approach, with a static constraint on the vehicle mass ensuring that the initial trajectory conditions lie within the allowed flight envelope (Section 3.6).

The trajectory dynamics rely on disciplinary models for the characteristics of the vehicle at a given state and control [24]. For a hypersonic glide vehicle these are principally the aerodynamic coefficients, which can be functions of Mach number,

angle of attack, sideslip angle, and/or altitude. These models can be coupled to the dynamics in two ways. In the direct approach, the aerodynamics are evaluated at each state and control requested by the trajectory optimizer, requiring an aerodynamic simulation at every collocation point of every iteration. In the surrogate-based approach, the aerodynamics are instead sampled beforehand, and a surrogate model fitted to these samples is evaluated in place of the simulation. Surrogates can take two forms. A global surrogate, $\tilde{\boldsymbol{C}}(\boldsymbol{x}_{\text{tr}}, \boldsymbol{u}, \boldsymbol{x}_{\text{v}})$, spans both the design and trajectory spaces, and can be generated offline and reused throughout the optimization; however, the number of samples required grows rapidly with the number of design variables included. A vehicle-specific surrogate, $\tilde{\boldsymbol{C}}(\boldsymbol{x}_{\text{tr}}, \boldsymbol{u}; \boldsymbol{x}_{\text{v}})$, spans only the trajectory variables for a fixed design, and must be regenerated within the loop whenever the design changes. These forms are written in full generality; in practice, a surrogate need only depend on the variables that influence the modeled quantity, which are typically a subset of, or derived from, the design, state, and control variables. For example, the aerodynamic coefficients in this work are functions of the Mach number and angle of attack, with the Mach number derived from the velocity and altitude states. In either form, the surrogate must be differentiable with respect to its trajectory inputs, as the gradient-based trajectory optimizer requires the derivatives of the dynamics with respect to the states and controls. For simultaneous co-design, the choice between direct and surrogate-based coupling depends on whether the number of samples needed for an accurate surrogate is smaller than the number of evaluations requested over the course of the optimization [24].

Nested co-design is naturally suited to vehicle-specific surrogates, as the design is held fixed throughout each trajectory optimization. Each trajectory optimization may require 10 to 1000 iterations, each evaluating the aerodynamics at every collocation point, and this is repeated at every outer iteration; a single sweep of the flight envelope per outer iteration replaces all of these evaluations. The advantage grows with aerodynamic fidelity, as the cost of each evaluation increases. With a surrogate, the trajectory optimization time becomes negligible compared to the time taken to generate the aerodynamic data, which is the regime in which the nested approach is most beneficial. A surrogate-based nested co-design framework is shown in the eXtended Design Structure Matrix (XDSM) format [25] in Fig. 1.

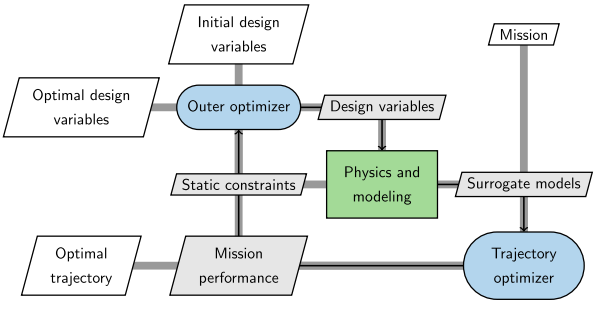


Figure 1: XDSM of the surrogate-based nested co-design framework

For the optimization of hypersonic glide vehicles, the XDSM can be refined to include more detail on the vehicle evaluation methodology. Figure 2 shows the XDSM used for the current investigation. The 'physics and modeling' process has been expanded into several distinct processes. The roles of the processes are discussed in Section 3. Not mentioned in the XDSM is the propagation of gradients. Gradient-based optimizers typically require far fewer function evaluations than gradient-free methods to converge, with the advantage growing with the number of design variables [26, 27]. Gradient-free methods, however, can treat each process as a black box, avoiding the need to ensure that every process in the framework is differentiable. As each evaluation here requires a complete pass through the framework, including an aerodynamic sweep and a trajectory optimization, the reduction in evaluations outweighs this cost, making the inclusion of gradients essential for solving the co-design problem efficiently. All the variables communicated between processes are also communicated with the gradients of those variables with respect to the design variables. This is achieved using forward-mode automatic differentiation and the chain rule, producing surrogate models that are differentiable with respect to the design variables (Section 3). The gradient of the optimal mission cost with respect to the design variables, $\boldsymbol{\nabla}_{\boldsymbol{x}_{\text{v}}} \mathcal{J}$, is then obtained from the surrogate gradients by post-optimality analysis of the converged trajectory optimization [28].

The outer problem of Eq. (2) is solved with a gradient-based optimizer. At every outer iteration the framework returns the mission cost $\mathcal{J}(\boldsymbol{x}_{\text{v}})$ and the vehicle constraints $\boldsymbol{h}_{\text{v}}(\boldsymbol{x}_{\text{v}})$, together with their gradients $\boldsymbol{\nabla}_{\boldsymbol{x}_{\text{v}}} \mathcal{J}$ and $\boldsymbol{\nabla}_{\boldsymbol{x}_{\text{v}}} \boldsymbol{h}_{\text{v}}$, from which the optimizer determines the next design point. A single outer iteration therefore comprises one complete pass through the framework of Fig. 2: mesh generation, evaluation of the vehicle properties and constraints, an aerodynamic sweep over the flight envelope, construction of the aerodynamic surrogates, and a nested trajectory optimization. The optimizer itself, its convergence criterion, and the initialization of the design vector are described in Section 3.1.

3. Vehicle and Trajectory Models

This section describes the models and solvers behind each process of the nested co-design framework shown in Fig. 2, taken in the order they appear in that figure. Alongside the specified outputs, every model provides the sensitivity of the outputs to the design parameters, as required by the gradient-based outer optimizer.

3.1. Outer Optimization

The outer problem of Eq. (2) is solved using sequential least-squares quadratic programming (SLSQP) [29, 30], a gradient-based method that solves a quadratic programming subproblem at each iteration, formed from a quasi-Newton approximation of the Hessian of the Lagrangian and a linearization of the constraints, using the pyslsqp implementation [31]. The optimization terminates once the constraint violation, and either the

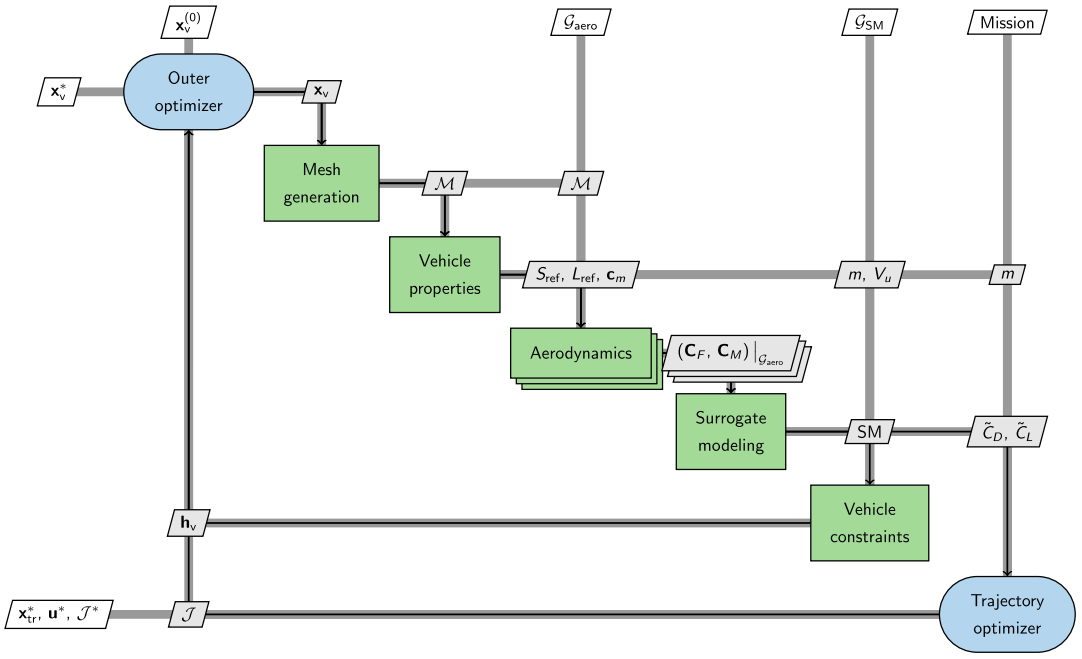


Figure 2: XDSM of the co-design framework for the optimization of a hypersonic glide vehicle

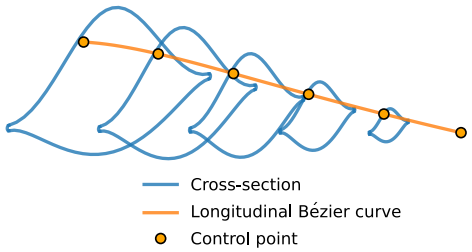
change in objective or the step length, fall below a tolerance of 10^{-4}. No iteration limit is imposed beyond the solver default of 100, which none of the optimizations presented here reach.

Two strategies are used to initialize the design vector, $\boldsymbol{x}_\mathrm{v}^{(0)}$, where the superscript (k) denotes the k-th outer iteration. A *cold start* places the initial design at the midpoint of the parameter bounds, $\boldsymbol{x}_\mathrm{v}^{(0)} = (\boldsymbol{x}_{\mathrm{v}_\mathrm{min}} + \boldsymbol{x}_{\mathrm{v}_\mathrm{max}})/2$, and requires no prior knowledge of the solution. A *warm start* instead initializes $\boldsymbol{x}_\mathrm{v}^{(0)}$ from an existing vehicle design of the same parameterization, providing a more informed initial guess. In this work, the warm start uses the converged design of a co-design optimization of the same mission performed at lower aerodynamic fidelity. Only the design vector is transferred: the physics evaluations are recomputed at the new fidelity, and the optimizer's Hessian approximation is rebuilt from scratch.

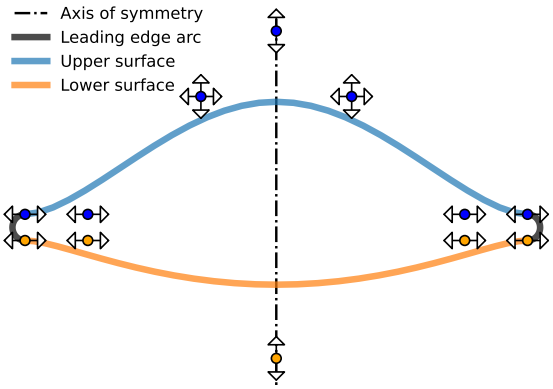
3.2. Vehicle Geometry

The vehicle is parameterized by a set of scalars that admit a wide range of aerodynamic shapes while maintaining a closed manifold, smooth mesh, and practical internal volume; suitability for launch and separation is not considered beyond a bound on the total length. Geometry is described primarily by two sets of Bézier curves, as shown in Fig. 3a. Bézier curves are a convenient choice as they allow a high level of parameterization while maintaining feasible geometries with a few sensible constraints (which are rarely active and not described here). Six curves traverse longitudinally with 6 control points each, including a fixed point at the nose. These longitudinal Bézier curves describe the control points for two nearly perpendicular Bézier curves which in turn describe the top and underside cross-section profile at a given axial station on the vehicle. In total, there are 30 Bézier control point parameters. One additional parameter prescribes total vehicle length, $x_\mathrm{v,Length}$, which in the current work is limited to 4 m, giving $N_d = 31$ design variables. The nose-tip leading edge is described with a half-sphere, and wing leading edges are described by a 180° arc, both with a constant diameter of 10 mm. The vehicle aft surface is created by transfinite interpolation between the four edges. Geometry generation is performed using the in-house repository hyperVehicle [32]. The surface mesh is generated using geometry definition functions adapted from the Eilmer geometry package distributed as part of GDTk maintained by the University of Queensland [33], and is formed of triangular elements. The vehicle is generated with the nose tip at the origin using forward-right-down body axes, such that the positive x-axis is directed through the nose.

The resulting mesh $\mathcal{M}$ consists of N_nodes surface nodes at positions $\boldsymbol{x}_k$, connected into N_panels triangular panels, each with area A_i, centroid $\boldsymbol{c}_i$, and outward unit normal $\hat{\boldsymbol{n}}_i$. The gradient-based outer optimizer requires accurate geometric sensitivities of the mesh, $\boldsymbol{\nabla}_{\boldsymbol{x}_\mathrm{v}}\mathcal{M}$. This requires obtaining the sensitivity of the mesh nodes to the design parameters. This is achieved using forward-mode automatic differentiation, implemented in a custom Python class with operator overloading. For a mesh with thousands of outputs and a smaller number of inputs, the forward-mode approach is more efficient than reverse-mode [34]. The implementation uses dual numbers, which carry the partial derivatives with respect to every design parameter alongside each value, avoiding the repeated geometry generation and step-size error of the finite differencing used previously [35].



(a) Example control points and corresponding longitudinal Bézier curve.



(b) Cross-section Bézier curves and control points. Arrows indicate control point degrees of freedom.

Figure 3: Schematics of vehicle profile defined by intersecting Bézier curves. Not to scale.

3.3. Vehicle Properties

The triangulated, watertight mesh surface allows for the quantification of the key vehicle properties. The vehicle mass is modeled as the sum of a fixed payload mass, $m_{\text{load}} = 300\,\text{kg}$, representing payload and avionics, and a structural shell with areal density $\rho_s = 30\,\text{kg}\,\text{m}^{-2}$ representing the outer structure and frame,

$$m \coloneqq m_{\text{load}} + \rho_s A_{\text{surface}} \tag{4}$$

where A_{surface} is the total surface area of the mesh. The payload is assumed to be located at the midpoint of the vehicle length, $\boldsymbol{c}_{\text{load}} = [-x_{\text{v,Length}}/2,\ 0,\ 0]^\top$, and the center of mass is

$$\boldsymbol{c}_m \coloneqq \frac{m_{\text{load}}\,\boldsymbol{c}_{\text{load}} + \rho_s \sum_{i=1}^{N_{\text{panels}}} A_i\,\boldsymbol{c}_i}{m} \tag{5}$$

where A_i and $\boldsymbol{c}_i$ are the area and centroid of the i-th of the N_{panels} panels in the surface mesh, respectively. The reference area and length are defined from the total enclosed volume V and vehicle length parameter $x_{\text{v,Length}}$,

$$S_{\text{ref}} \coloneqq V^{2/3}, \qquad \ell_{\text{ref}} \coloneqq x_{\text{v,Length}}. \tag{6}$$

The usable internal volume excludes a structural shell of thickness $t_s = 0.025\,\text{m}$ to account for structure, framing, and the limitations of curved internal volume, particularly in wing sections,

$$V_{\text{u}} \coloneqq V - t_s\,A_{\text{surface}} \tag{7}$$

Automatic differentiation provides the gradient of each of these properties with respect to the design parameters.

3.4. Aerodynamics Solver

For the present study, it is assumed that there is no Reynolds number dependence and the flow is treated as inviscid. The vehicle's aerodynamic performance is therefore a function of the Mach number, Ma, and the vehicle's orientation to the flow, as determined by the angle of attack α and sideslip angle β. The dynamics in Section 3.7 assume zero sideslip angle, creating a two-dimensional operating space for the vehicle aerodynamics. To generate the aerodynamic database for a surrogate model, simulations are conducted for a structured grid of discrete operating points $\mathcal{G}_{\text{aero}}$ spanning the flight envelope $\alpha \in [0,\ 15°]$, $\text{Ma} \in [2,\ 10]$, with the specific grid depending on the aerodynamic solver (Table 1).

Table 1: Angle-of-attack and Mach number sampling grids for aerodynamic database generation ($\mathcal{G}_{\text{aero}}$, with a reduced grid for CFD) and static margin constraint evaluation ($\mathcal{G}_{\text{SM}}$).

Grid	α, deg	Ma	Points
$\mathcal{G}_{\text{aero}}$	$0, 1.5, \ldots, 15$	$2, 3, \ldots, 10$	99
$\mathcal{G}_{\text{aero}}^{\text{CFD}}$	$0, 3, \ldots, 15$	$2, 4, \ldots, 10$	30
$\mathcal{G}_{\text{SM}}$	$2.5, 7.5, 12.5$	$3, 6, 9$	9

Each simulation provides the pressure for each panel of the triangulated vehicle geometry. The resulting forces and moments relative to the center of mass of the vehicle are

$$\boldsymbol{F} = \sum_{i=1}^{N_{\text{panels}}} \boldsymbol{F}_i = -\sum_{i=1}^{N_{\text{panels}}} p_i A_i \hat{\boldsymbol{n}}_i, \tag{8}$$

$$\boldsymbol{M} = \sum_{i=1}^{N_{\text{panels}}} (\boldsymbol{c}_i - \boldsymbol{c}_m) \times \boldsymbol{F}_i, \tag{9}$$

where p_i is the pressure on panel i and $\hat{\boldsymbol{n}}_i$ is its outward unit normal. The gradient-based outer optimizer also requires the Jacobians of the forces and moments with respect to the design variables, $\mathbf{J}_{\boldsymbol{x}_{\text{v}}}\boldsymbol{F}$ and $\mathbf{J}_{\boldsymbol{x}_{\text{v}}}\boldsymbol{M}$, where $\mathbf{J}_{\boldsymbol{x}_{\text{v}}}$ denotes the Jacobian with respect to $\boldsymbol{x}_{\text{v}}$. These follow from applying the product rule to each term of the sums, and require the sensitivities of the panel pressures, $\boldsymbol{\nabla}_{\boldsymbol{x}_{\text{v}}} p_i$, areas, $\boldsymbol{\nabla}_{\boldsymbol{x}_{\text{v}}} A_i$, normals, $\mathbf{J}_{\boldsymbol{x}_{\text{v}}}\hat{\boldsymbol{n}}_i$, and centroids, $\mathbf{J}_{\boldsymbol{x}_{\text{v}}}\boldsymbol{c}_i$, together with that of the center of mass, $\mathbf{J}_{\boldsymbol{x}_{\text{v}}}\boldsymbol{c}_m$. The mesh generation provides the geometric sensitivities for the area, normal vector, and centroid positions, while the method for calculating the pressure gradient depends upon the aerodynamic solver.

The forces and moments are non-dimensionalized by the freestream dynamic pressure q_∞ and the reference quantities S_{ref} and ℓ_{ref} defined in Section 3.3,

$$\boldsymbol{C}_F \coloneqq \frac{\boldsymbol{F}}{q_\infty S_{\text{ref}}}, \qquad \boldsymbol{C}_M \coloneqq \frac{\boldsymbol{M}}{q_\infty S_{\text{ref}} \ell_{\text{ref}}}. \tag{10}$$

The freestream dynamic pressure is

$$q_\infty \coloneqq \frac{1}{2}\rho_\infty v_\infty^2 = \frac{1}{2}\gamma p_\infty \mathrm{Ma}_\infty^2, \tag{11}$$

where ρ_∞, v_∞, p_∞, and Ma_∞ are the freestream density, velocity, pressure, and Mach number, and $\gamma = 1.4$ is the specific heat ratio of air. The lift, drag, and pitching moment coefficients C_L, C_D, and C_m are obtained by resolving $\boldsymbol{C}_F$ into the wind frame and extracting the pitching moment component of $\boldsymbol{C}_M$.

3.4.1. Local Surface Inclination Methods

At early conceptual design stages, simplified models which allow for rapid evaluation and development are preferred. For hypersonic flows, local surface inclination methods (panel methods) provide computationally inexpensive predictions of surface pressure. The surface inclination angle δ is calculated from the outward unit normal of each panel and the freestream velocity direction,

$$\delta_i \coloneqq \cos^{-1}\left(\hat{v}_\infty \cdot \hat{\boldsymbol{n}}_i\right) - \frac{\pi}{2} \tag{12}$$

with $\delta \in [-\pi/2, \pi/2]$, where positive values correspond to windward-facing panels, negative values to rearward-facing panels, and zero to panels tangential to the flow. Four schemes are employed depending on δ and the flow conditions: the tangent cone and normal shock schemes for windward panels, and the Prandtl–Meyer expansion and rear-face schemes for rearward panels. Full details and an illustrative C_p–δ comparison across Mach numbers are provided in Appendix A. Each scheme provides the coefficient of pressure $C_p(\delta, \mathrm{Ma}, \gamma)$, from which the panel pressure is

$$p \coloneqq q_\infty C_p(\delta, \mathrm{Ma}, \gamma) + p_\infty. \tag{13}$$

The pressure sensitivity with respect to the design parameters follows from the chain rule through the inclination angle δ_i,

$$\boldsymbol{\nabla}_{\boldsymbol{x}_\mathrm{v}} p_i = -\frac{q_\infty}{\cos\delta_i} C_{p,\delta}\left(\delta_i, \mathrm{Ma}, \gamma\right) \hat{\boldsymbol{v}}_\infty \cdot \mathbf{J}_{\boldsymbol{x}_\mathrm{v}} \hat{\boldsymbol{n}}_i, \tag{14}$$

where $C_{p,\delta} \coloneqq \partial C_p / \partial\delta$ is evaluated analytically or, where this is unavailable, from a cubic spline fit of C_p against δ. The aerodynamics for the panel methods are evaluated on the structured grid $\mathcal{G}_\mathrm{aero}$ listed in Table 1.

3.4.2. Inviscid Computational Fluid Dynamics

For higher fidelity, inviscid simulations are performed using ANSYS Fluent [36], at a much greater cost per outer iteration than the local surface inclination methods.

The vehicle is exported to a STEP file and meshed using the ANSYS Fluent watertight meshing workflow. The unstructured mesh is generated around the vehicle, using a supersonic outflow condition aligned with the rear of the surface. The wake region is not included in the simulation domain due to issues that occur from trying to capture the wake region of inviscid, hypersonic flows [37, 38]. The pressures on the rear surface are estimated using the Prandtl–Meyer expansion fan equations. As the simulations are inviscid, only 15 inflation layers are used to ensure that the flow is captured near the vehicle surface, which is treated as an inviscid wall boundary condition. The inflow is specified as a pressure far-field boundary condition on a domain boundary shaped to follow the vehicle, positioned so that a detached shock at Mach 2 does not reflect off the boundary and affect the flow over the vehicle. The inflow is defined by the Mach number, a freestream pressure $p_\infty = 500\,\mathrm{Pa}$, and a freestream temperature $T_\infty = 240\,\mathrm{K}$; the absolute values of p_∞ and T_∞ do not affect the aerodynamic coefficients since forces are non-dimensionalized by the dynamic pressure q_∞.

The simulations use Fluent's density-based solver and the high-speed numerics correction to simulate an ideal gas flow. The Green-Gauss cell-based gradient scheme and differentiable limiter [39] are used with the AUSM+ flux calculator [40] to enable the simulations to reliably converge in reasonable timeframes. The simulations are conducted until the lift, drag, and pitching moment are converged to within 0.1 % over the past 100 iterations. No hard residual convergence criterion is set; residuals typically reach the order of 1×10^{-4} by the time the forces and moments have converged. The surface properties of the vehicle are exported from the Fluent simulation and interpolated onto the original triangulated mesh for the vehicle.

The Fluent density-based solver does not possess adjoint capabilities for calculating the sensitivity of the flow to the vehicle surface geometry. Instead, an isentropic flow assumption is employed to calculate the pressure gradient on the surface as a function of the local surface properties [41, 42],

$$\boldsymbol{\nabla}_{\boldsymbol{x}_\mathrm{v}} p_i = -\gamma p_i \frac{\mathrm{Ma}_i^2}{\sqrt{\mathrm{Ma}_i^2 - 1}} \hat{\boldsymbol{v}}_i \cdot \mathbf{J}_{\boldsymbol{x}_\mathrm{v}} \hat{\boldsymbol{n}}_i, \tag{15}$$

where Ma_i is the local Mach number and $\hat{\boldsymbol{v}}_i$ is the unit vector in the direction of the local flow velocity at panel i. The expression is defined only for supersonic local flow; on panels where $\mathrm{Ma}_i \leq 1$, such as in the subsonic region behind a detached shock at the nose and leading edges, the pressure sensitivity is set to zero. Compared with the inviscid CFD simulation, the time required for the calculation of the gradients is negligible.

To reduce the total computational cost, CFD simulations are conducted on the coarser grid $\mathcal{G}_\mathrm{aero}^\mathrm{CFD}$ listed in Table 1.

3.5. Aerodynamic Surrogate Models

The structured grid sampling used to generate the aerodynamic databases enables the use of interpolation schemes for the surrogate model. Polynomial models, which have appeared elsewhere in the literature [35, 43], are an alternative that captures the gross aerodynamic behavior in the region of interest. Another alternative is to use Gaussian process models, which allow for gradient information to be used both to enhance the model's accuracy and predict gradients [44]. Further, they admit multi-fidelity techniques that allow fusion of small amounts of high-fidelity data with more abundant, cheaply available low-fidelity data to greatly improve accuracy on a limited compute budget [45].

The aerodynamic database contains the force and moment coefficients sampled over $\mathcal{G}_\mathrm{aero}$. For each of the lift, drag, and

pitching-moment coefficients, a cubic B-spline is fitted to the sampled values, allowing the coefficient to be evaluated continuously across the angle of attack and Mach number range,

$$\tilde{C}_j(\alpha, \mathrm{Ma}; \boldsymbol{x}_\mathrm{v}^{(k)}) \coloneqq \mathcal{S}[C_j](\alpha, \mathrm{Ma}), \qquad j \in \{L,\, D,\, m\}, \tag{16}$$

where $\mathcal{S}[\,\cdot\,]$ denotes the spline interpolant of a quantity sampled over $\mathcal{G}_\mathrm{aero}$. This is a vehicle-specific surrogate valid only for the current design $\boldsymbol{x}_\mathrm{v}^{(k)}$.

To propagate the design sensitivities through the trajectory optimization, the surrogate is extended to a first-order Taylor expansion in the design variables about the current design,

$$\begin{aligned}\tilde{C}_j(\alpha, \mathrm{Ma}, \boldsymbol{x}_\mathrm{v};\, \boldsymbol{x}_\mathrm{v}^{(k)}) \coloneqq\ & \mathcal{S}[C_j](\alpha, \mathrm{Ma}) \\ & + \sum_{i=1}^{N_d} (x_{\mathrm{v}i} - x_{\mathrm{v}i}^{(k)})\, \mathcal{S}\left[\frac{\partial C_j}{\partial x_{\mathrm{v}i}}\right](\alpha, \mathrm{Ma}),\end{aligned} \tag{17}$$

where the sensitivities at each sample point are provided by the aerodynamic solver (Section 3.4) and interpolated in the same manner as the coefficients. This inclusion of the gradients allows for aerodynamic predictions for vehicles with different design variables; however, the surrogate is only ever evaluated with $\boldsymbol{x}_\mathrm{v} = \boldsymbol{x}_\mathrm{v}^{(k)}$, where the expansion term vanishes. Differentiating the surrogate with respect to a design variable $x_{\mathrm{v}i}$ will return the corresponding interpolator function, $\mathcal{S}\left[\partial C_j/\partial x_{\mathrm{v}i}\right](\alpha, \mathrm{Ma})$, which can be evaluated to obtain the interpolated sensitivity. This allows the surrogate functions to reproduce both the values and the gradients of the aerodynamic database. For brevity, $\tilde{C}_j(\alpha, \mathrm{Ma})$ is used hereafter. Derivatives with respect to the trajectory variables required by the optimal control solver and the constraint evaluations, such as $\partial \tilde{C}_j/\partial\alpha$, are obtained analytically from the cubic splines.

3.6. Vehicle Constraints

For the parametric vehicle design, constraints are placed on the design parameters according to some lower and upper bounds, creating a hypercube search space for the outer optimization problem. In addition to these bounds, three vehicle constraints are used; these and the length bound are listed in Table 2. The first is the usable volume constraint, which is the enclosed volume of the vehicle minus the volume of the structural shell. The minimum volume is applied to ensure there is room for internal components such as a power system, avionics, or actuators for the vehicle. This approach is a proxy for modeling the detailed packaging of such componentry within the vehicle, and does not prevent the creation of thin and flat vehicles. Other approaches consider applying a sphericity (or volumetric efficiency) constraint to the vehicle. This approach was not taken, as the sphericity constraint penalizes the presence of any wings for the blended fuselage and wing design used for the vehicle. The second constraint added is the static margin, which measures the vehicle's static pitching stability [46, 47]. The static margin is calculated using the differentiable aerodynamic surrogate models,

$$\mathrm{SM}(\alpha, \mathrm{Ma}) \coloneqq \frac{\tilde{C}_{m,\alpha}\,(\alpha, \mathrm{Ma})}{\tilde{C}_{L,\alpha}\,(\alpha, \mathrm{Ma})}, \tag{18}$$

where $\tilde{C}_{j,\alpha} \coloneqq \partial \tilde{C}_j/\partial\alpha$ denotes the slope of the surrogate with respect to angle of attack, and $\mathrm{SM} < 0$ denotes pitching stability. As the dynamic equations do not capture the rotational dynamics of the vehicle, the static margin constraint is included to ensure that there are viable flight trajectories that the vehicle can operate in while statically stable. The static margin varies with angle of attack and Mach number, and is evaluated at the discrete points $\mathcal{G}_\mathrm{SM}$ listed in Table 1. This set does not extend to the bounds of the allowed flight envelope, which can result in regions of the flight envelope being unstable; the trajectory constraint of Section 3.7 prevents the vehicle from operating in these regions. There are no constraints applied on the moment coefficient; instead, it is assumed that the vehicle can be trimmed and is able to maintain the desired angle of attack.

The final static constraint applied is for the vehicle mass. The mass bounds are configured from the trajectory initial conditions used for the trajectory optimization, discussed in Section 3.7. The inclusion of this constraint ensures the vehicle's trajectory begins within the allowed Mach number range, helping to prevent vehicle designs with no feasible trajectory solutions.

Table 2: Static design constraints

	Bound	
Constraint	Lower	Upper
Length, m	2	4
Usable volume, m^3	0.45	–
Static margin[a]	–	0
Mass, kg	373	3121

[a]Evaluated at points in $\mathcal{G}_\mathrm{SM}$.

3.7. Flight Dynamics and Trajectory Optimization

The vehicle dynamics are governed by a simplified round-Earth, trimmed, three-degree-of-freedom system [46, 48],

$$\dot{\boldsymbol{x}} = \begin{bmatrix} \dot{s}_x \\ \dot{s}_y \\ \dot{H} \\ \dot{v} \\ \dot{\gamma} \\ \dot{\chi} \end{bmatrix} = \begin{bmatrix} v\cos\gamma\cos\chi\dfrac{r_e}{r_e+H} \\ v\cos\gamma\sin\chi\dfrac{r_e}{r_e+H} \\ v\sin\gamma \\ -D/m - g\sin\gamma \\ \dfrac{1}{mv}\left(L\cos\mu - mg\cos\gamma + \dfrac{mv^2\cos\gamma}{r_e+H}\right) \\ \dfrac{L\sin\mu}{mv\cos\gamma} \end{bmatrix} \tag{19}$$

where s_x, s_y are downrange and crossrange distance at ground level, respectively, H is altitude, v is airspeed, γ is the flight path angle, and χ is the heading angle. The control inputs are angle of attack, $\alpha(t)$, and bank angle, $\mu(t)$. The simplified round-Earth assumes a constant Earth radius of $r_e = 6\,356\,766\,\mathrm{m}$, and an inverse-square gravitational law for the gravitational acceleration $g(H)$, with $g_0 = 9.806\,65\,\mathrm{m\,s^{-2}}$ at sea level [49]. Lift and drag forces are calculated as

$$L = q\,S_\mathrm{ref}\tilde{C}_L(\alpha, \mathrm{Ma}), \qquad D = q\,S_\mathrm{ref}\tilde{C}_D(\alpha, \mathrm{Ma}), \tag{20}$$

where $q = \frac{1}{2}\rho(H)v^2$ is the dynamic pressure and $\tilde{C}_L$ and $\tilde{C}_D$ are the surrogate aerodynamic models of Section 3.5. The atmospheric density $\rho(H)$, along with the pressure and temperature used to calculate flow properties, is obtained using the 1976 U.S. Standard Atmosphere model [49] as a function of altitude.

The optimal control problem is solved by transcribing the dynamics using trapezoidal collocation with uniformly spaced segments. The optimal trajectory is then computed by the interior-point nonlinear program (NLP) solver IPOPT [50] and the CasADi interface [51]. The NLP sensitivity capability of CasADi [28] is used to provide the objective cost sensitivity with respect to the design parameters, $\nabla_{x_v}\mathcal{J}$. To ensure the optimal control solver reliably converges to a local optimum, and is thus suitable for inclusion in the automated outer-loop optimization, the NLP is solved on successively refined temporal discretizations. The trajectory is first converged using 25 segments. This solution is interpolated to initialize the states and controls for a trajectory with twice as many segments, which is then optimized using CasADi and IPOPT. This refinement process is repeated until convergence is achieved with 200 segments. Initial estimates of the state variables for the NLP solver are constructed by forward-integrating a trajectory with constant 7.5° angle of attack for a duration of 200 s. This was found to be sufficient to obtain NLP solutions which reliably converge to feasible solutions.

3.7.1. Trajectory Constraints

Several constraints are included to ensure a realistic and physically feasible optimized trajectory is obtained. No constraints are provided specifically on the final state of the vehicle, while the initial state is constrained, dependent upon the vehicle's properties. The initial position of the vehicle is specified by a downrange, crossrange, and altitude of 0 km, 0 km, and 50 km, respectively, and the initial flight path angle and heading angle are set to zero. To provide a fair comparison between different vehicles launched by the same first-stage booster, the initial velocity $v(0)$ is calculated from a constant initial total energy (sum of potential and kinetic) of 2210 MJ. This initial energy value represents a 500 kg vehicle with initial velocity of Mach 8.5 at an altitude of 50 km. While the true scaling of initial velocity is more complex than constant total energy, the total energy assumption suffices for demonstrating the co-design method. This also prevents the optimizer from directly favoring heavy vehicles, which have higher initial kinetic energy and momentum. The initial state is thus a function of vehicle parameters, and the corresponding sensitivities are incorporated in the automatic differentiation graph.

The constraints applied throughout the trajectory, $\boldsymbol{h}_{\text{tr}}$, are listed in Table 3 and are enforced by the optimal control solver. A lower bound of 10 km is applied to the altitude state. It is common in trajectory optimization to apply limits to the maximum dynamic pressure of 50 kPa or higher [52, 53, 54]. In the absence of control surfaces and rotational modeling, limits are also included for the minimum dynamic pressure to represent the minimum aerodynamic force required for control surface efficacy and maneuverability. In addition to the static constraint evaluated on $\mathcal{G}_{\text{SM}}$ (Section 3.6), the static margin is constrained to be non-positive at every point along the trajectory, $\text{SM}(\alpha(t), \text{Ma}(t)) \leq 0$, keeping the vehicle within statically stable regions of the flight envelope. The Mach number and angle-of-attack bounds are specified to ensure that the vehicle only operates within the established aerodynamic database. Limits are applied to the magnitude and the rate of change of the two control inputs: angle of attack and bank angle. As the real vehicle has rotational inertia, these angular control inputs should not change discontinuously, with limits taken from Carvalho and Mooij [55]. To account for structural loads on the vehicle, a constraint is applied to the aerodynamic load factor defined by,

$$n \coloneqq \frac{\sqrt{L^2 + D^2}}{mg}, \tag{21}$$

with $n = 2.5$ commonly used [55, 52, 53, 54].

Table 3: Trajectory constraints

	Bound	
Constraint	Lower	Upper
Altitude, km	10	–
Dynamic pressure, kPa	2	50
Mach number	2	10
Angle of attack, deg	0	15
Bank angle, deg	−180	180
Angle-of-attack rate, deg/s	−5	5
Bank angle rate, deg/s	−20	20
Static margin	–	0
Load factor	–	2.5

4. Design Problems

Three design problems are posed to demonstrate control co-design of hypersonic glide vehicles: a point-designed baseline and two mission-based co-design problems, the latter taken from Lock et al. [56]. The optimized vehicles are denoted $\mathcal{V}_{\text{pd}}$, $\mathcal{V}_{\text{sp}}$, and $\mathcal{V}_{\text{ob}}$, respectively, and the problems are summarized in Table 5. All three apply the vehicle constraints of Section 3.6, and the two missions additionally solve a trajectory subject to the constraints and common initial conditions of Section 3.7, differing from each other only in a single mission-specific constraint.

4.1. Point-Designed Baseline

The point-designed vehicle is optimized from static vehicle properties alone, without solving a trajectory. Its objective combines the lift-to-drag ratio at Mach 6 and $\alpha = 7.5°$, the midpoints of the allowed Mach number and angle-of-attack ranges, with a mass penalty,

$$\mathcal{J}_{\text{pd}}(\boldsymbol{x}_{\text{v}}) \coloneqq -\left.\frac{\tilde{C}_L}{\tilde{C}_D}\right|_{\alpha=7.5°,\ \text{Ma}=6} + \frac{m}{m_{\text{ref}}}, \tag{22}$$

where the lift-to-drag ratio and vehicle mass m depend on the design parameters $\boldsymbol{x}_{\text{v}}$ alone (Sections 3.3 and 3.5). Both terms

are proxies for mission performance: a more efficient vehicle glides further [57], and a lighter vehicle leaves the same boost stage with a higher velocity. The reference mass $m_{\text{ref}} = 500\,\text{kg}$ is chosen so that the lift-to-drag term dominates near the expected mass range. This value is a designer choice for balancing the contributions of the two terms, a limitation of point design that co-design avoids. In Fig. 2, the 'Trajectory optimizer' stage is replaced by evaluation of $\mathcal{J}_{\text{pd}}$ and all other stages are unchanged.

4.2. Sprint Mission

The sprint mission maximizes downrange distance,

$$\mathcal{J}_{\text{tr}}(\boldsymbol{x}_{\text{tr}}, t_f; \boldsymbol{x}_{\text{v}}) \coloneqq -s_x(t_f), \tag{23}$$

subject to the terminal time constraint $t_f \leq 420\,\text{s}$, where s_x is the downrange component of the state $\boldsymbol{x}_{\text{tr}}$ (Section 3.7). This limit is well below the unconstrained flight time of these vehicles, so flying at maximum lift-to-drag ratio is not necessarily optimal.

4.3. Obstacle Mission

The obstacle mission uses the same objective, Eq. (23), but with no limit on flight time. Instead, three cylindrical obstacles straddle the straight-line path, adding the path constraints

$$(s_x - x_{c,i})^2 + (s_y - y_{c,i})^2 - r^2 \geq 0, \quad i \in \{1, 2, 3\}, \tag{24}$$

where $r = 250\,\text{km}$ and the obstacle centers are listed in Table 4. Avoiding them requires the vehicle to bank, exercising its maneuverability.

Table 4: Obstacle center coordinates for the obstacle mission

Obstacle	x_c, km	y_c, km
1	500	150
2	1000	-150
3	1500	150

5. Low-Fidelity Co-Design Results

The three design problems of Section 4 are solved using the local surface inclination methods of Section 3.4 to evaluate the aerodynamics. The low cost of sweeping the flight envelope with this approach keeps each outer iteration inexpensive, allowing for rapid development of preliminary designs.

5.1. Optimized Vehicle Geometries

All three vehicle geometries are presented in Fig. 4, with their cross-sections compared on common axes in Fig. 5, showing that each vehicle has a distinct design. This uniqueness of geometry is evidence of the coupling between mission profile-driven dynamics and the vehicle geometry optimization. $\mathcal{V}_{\text{sp}}$ presents a slender shape overall, with near-symmetric upper and lower surfaces and a diamond-like aft cross-section. $\mathcal{V}_{\text{ob}}$ is much wider and flatter in comparison to $\mathcal{V}_{\text{sp}}$, while still attaining the same length, volume, and stability. $\mathcal{V}_{\text{pd}}$ is even wider and flatter than $\mathcal{V}_{\text{ob}}$. Unlike the two mission-designed vehicles, $\mathcal{V}_{\text{pd}}$'s aft section does not curve downwards; instead, the bottom surface remains close to the $z = 0$ plane throughout.

Several static properties of the vehicles are listed in Table 6. All vehicles sit at the minimum usable volume constraint and reach the maximum length. Maximizing the length is an easy way for the vehicles to increase the usable volume, as there is no viscous drag penalty associated with the length of the vehicle in the aerodynamic analysis. The heights and widths of the vehicles vary greatly, and this is also quantified by the sphericity. Sphericity is a scaled measure of the volumetric efficiency, with a maximum of unity achieved for a sphere,

$$\text{Sphericity} \coloneqq \pi^{1/3} 6^{2/3} \frac{V^{2/3}}{A_{\text{surface}}}. \tag{25}$$

As observed for similar volume-to-area measures [58], the maximum lift-to-drag ratio of the optimized vehicles decreases with increasing sphericity, with $\mathcal{V}_{\text{pd}}$ achieving the highest lift-to-drag and lowest sphericity. All three vehicles have a maximum static margin that is within the optimizer tolerance (10^{-4}) of the upper constraint.

5.2. Mission Performance

5.2.1. Sprint Mission

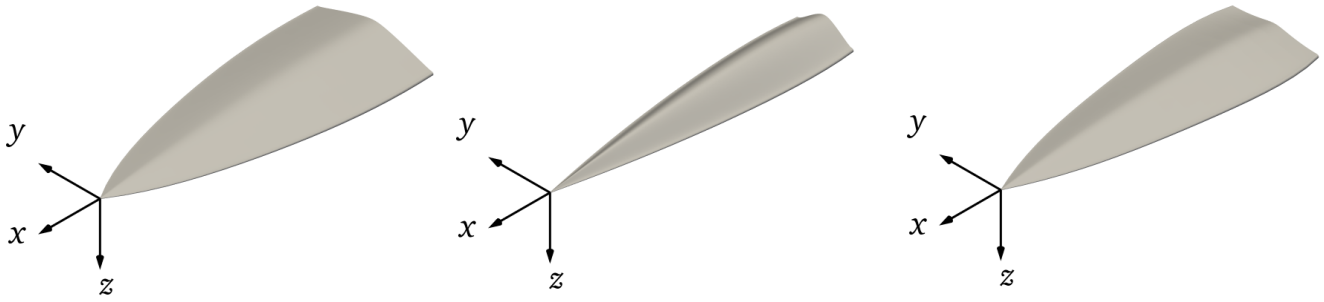
For the sprint mission, the target objective was to maximize x-distance, so none of the optimum trajectories utilized bank angle or obtained any y-displacement, and the corresponding plots are omitted as the values remain constant at zero. The optimum trajectories for the sprint mission are shown in Fig. 6 for all three vehicles. The sprint co-designed vehicle $\mathcal{V}_{\text{sp}}$ achieves the greatest distance of the three vehicles, as is expected from the co-design process. The profiles for the altitude H as a function of time for the vehicles are quite similar, due to the similar L/W values obtained by the vehicles throughout. However, the sprint vehicle is able to minimize the skipping behavior more than the other vehicles, helping it to maintain a more steady dynamic pressure. $\mathcal{V}_{\text{sp}}$ has the highest drag-to-weight ratio of the three vehicles for most of the trajectory, and therefore the greatest deceleration. Despite this, $\mathcal{V}_{\text{sp}}$ travels further because of its higher initial velocity, which it maintains throughout the trajectory and which the optimizer obtains by reducing the vehicle mass. As the mass scales with surface area and the usable volume is constrained, a lighter vehicle requires a higher sphericity, at a cost to aerodynamic performance; the design of $\mathcal{V}_{\text{sp}}$ reflects the optimizer's trade-off between the two.

The point-designed vehicle performed the worst at this mission, owing to its larger mass and lower initial velocity. Mission performance could be improved by changing the reference mass, altering the balance of the aerodynamic and mass contributions, but this requires knowledge of the trajectory performance and how it depends upon the mass and aerodynamics. $\mathcal{V}_{\text{pd}}$ also remained above its Mach 6 design point throughout; an iterative point design that shifts the target Mach number and angle of attack to the trajectory's average could improve performance. Both adjustments can narrow the gap but are not

Table 5: Summary of the three design problems

Vehicle	Design problem	Objective, $\mathcal{J}$	Distinguishing constraint
$\mathcal{V}_{\text{pd}}$	Point design	$-\tilde{C}_L/\tilde{C}_D + m/m_{\text{ref}}$[a]	–
$\mathcal{V}_{\text{sp}}$	Sprint mission	$-s_x(t_f)$	$t_f \leq 420\,\text{s}$
$\mathcal{V}_{\text{ob}}$	Obstacle mission	$-s_x(t_f)$	Three obstacles, $r = 250\,\text{km}$

[a]Evaluated at Ma = 6, $\alpha = 7.5°$, with $m_{\text{ref}} = 500\,\text{kg}$.



(a) $\mathcal{V}_{\text{pd}}$ – Point design (b) $\mathcal{V}_{\text{sp}}$ – Sprint co-design (c) $\mathcal{V}_{\text{ob}}$ – Obstacle co-design

Figure 4: Isometric view of vehicles

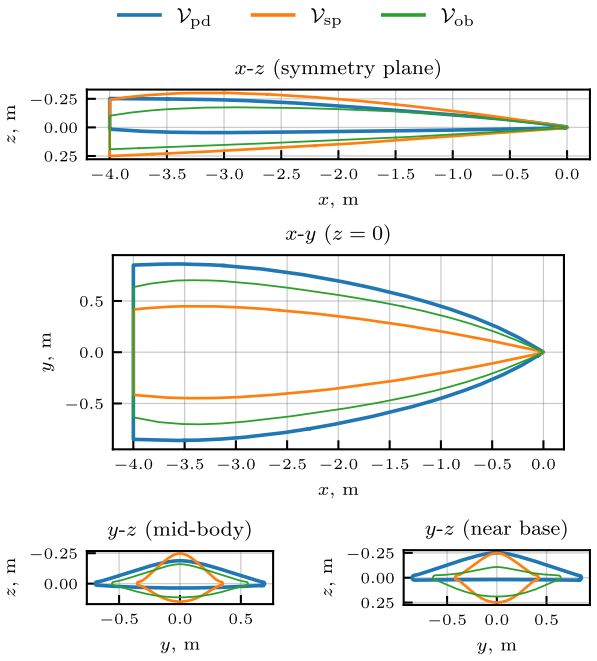


Figure 5: Cross-section outlines of $\mathcal{V}_{\text{pd}}$, $\mathcal{V}_{\text{sp}}$ and $\mathcal{V}_{\text{ob}}$ overlaid on common axes: the x-z symmetry plane, the x-y planform at $z = 0$, and y-z cuts at mid-body and near the base. The nose is at $x = 0$ and z is positive down.

Table 6: Vehicle properties

Property	$\mathcal{V}_{\text{pd}}$	$\mathcal{V}_{\text{sp}}$	$\mathcal{V}_{\text{ob}}$
Length, m	4	4	4
Width, m	1.72	0.899	1.41
Height, m	0.298	0.553	0.367
Max SM[a], $\times 10^{-5}$	2.43	6.94	−2.65
Max L/D	6.41	5.22	6.36
Mass, kg	614	483	556
Usable Vol., m^3	0.45	0.45	0.45
Sphericity	0.368	0.566	0.432

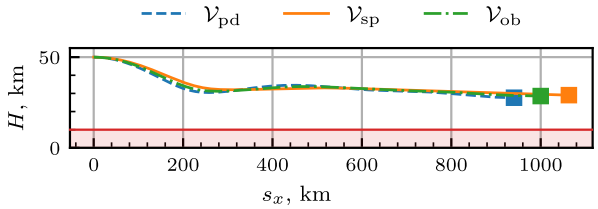
[a]Evaluated at points in $\mathcal{G}_{\text{SM}}$.

guaranteed to match the performance of the sprint co-designed vehicle.

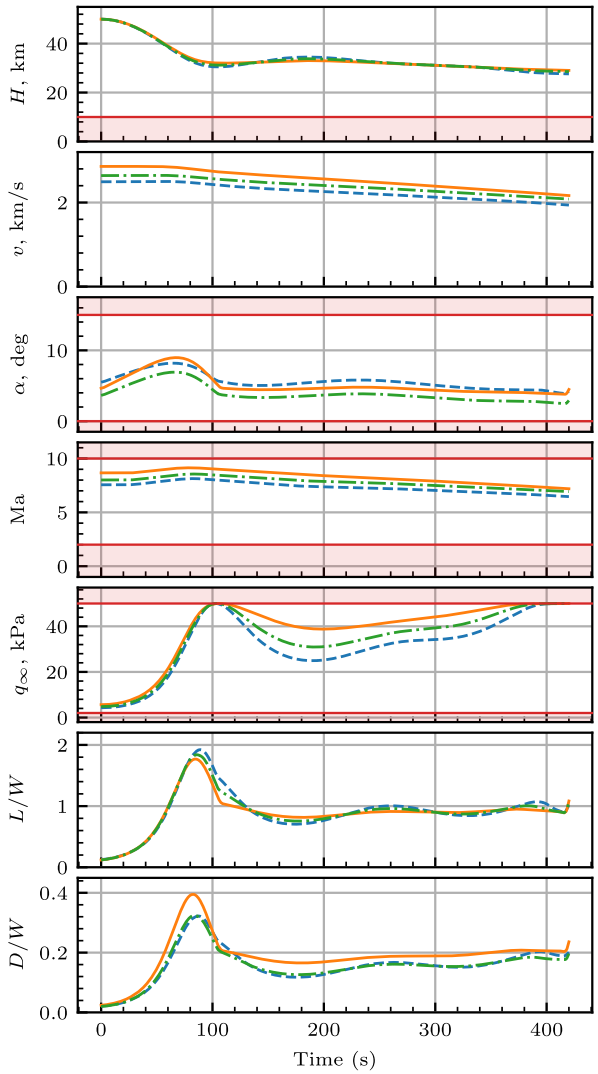
The only trajectory constraint which affected the trajectory of the vehicles in the sprint mission was the upper dynamic pressure limit. The vehicles reach the dynamic pressure limit at the initial pull-up ($t \approx 100\,\text{s}$) and at the end of the trajectory. The vehicles are initialized with insufficient speed to maintain their altitude and quickly descend, which increases the vehicles' speed as well as the atmospheric density, causing the dynamic pressure to increase. The optimal control for the vehicles shows that this is mitigated by increasing the angle of attack to perform a pull-up maneuver, increasing both lift and drag to minimize altitude loss and decrease velocity, respectively.

5.2.2. *Obstacle Mission*

The obstacle mission places three obstacles in the x-y plane, which are visible in the spatial plots in Fig. 7. To maneuver around the obstacles, the vehicles must control their bank angle to adjust their heading angle (not plotted). The obstacle co-design vehicle $\mathcal{V}_{\text{ob}}$ is able to maneuver around all three obstacles and achieve the greatest downrange distance. $\mathcal{V}_{\text{sp}}$ and



(a) Spatial profile



(b) Temporal profile

Figure 6: Trajectories for the sprint mission. Lift and drag are normalized by vehicle weight W.

$\mathcal{V}_{\text{pd}}$ both clear the first two obstacles but end their trajectories alongside the third, with $\mathcal{V}_{\text{pd}}$'s distance falling between those of the two mission co-designed vehicles.

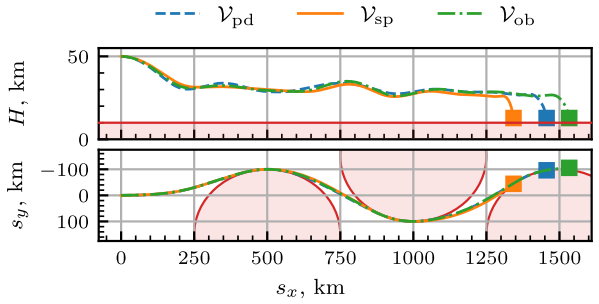
With the lower mass, $\mathcal{V}_{\text{sp}}$ starts the obstacle mission with the highest velocity, but this decreases with time due to higher operating drag. Over the longer time duration, $\mathcal{V}_{\text{sp}}$'s aerodynamics cause it to perform worse than the other two vehicles: operating at similar L/W but higher D/W than the other vehicles has reduced the maximum distance $\mathcal{V}_{\text{sp}}$ can glide. $\mathcal{V}_{\text{ob}}$ maintained a slightly higher velocity than $\mathcal{V}_{\text{pd}}$ throughout the trajectory, allowing it to travel further and for a longer duration. As with $\mathcal{V}_{\text{sp}}$ on the sprint mission, the co-design optimization is balancing $\mathcal{V}_{\text{ob}}$'s mass and aerodynamic performance to attain the optimal mission performance.

All vehicles start at a larger angle of attack than they did for the sprint mission, but this is also accompanied by a bank angle of greater than 90°. At these angles, the lift force is pushing the vehicle downward, allowing all three vehicles to dive slightly faster while also moving laterally to maneuver around the first obstacle. The bank angles when passing the first and second obstacles are smaller in magnitude, as the lateral acceleration required for a turn of a given radius, v^2/R, falls as the vehicles slow. After the initial pull-up maneuver, the maximum dynamic pressures are achieved during the apex of these turns, when the bank angles are at a local maximum magnitude. This higher dynamic pressure provides the additional lift needed to both support the vehicle against gravity and turn it, at the cost of increased drag. The loads during the initial pull-up maneuver are more intense in the obstacle mission than for the sprint mission, achieving larger L/W and reaching the maximum load factor constraint. The trajectory for the vehicles ends once the vehicles have exhausted their energy, with all three vehicles riding along the lower Mach number boundary at the end. The vehicles are observed to invert at the end to help dive and maintain their Mach number. This dive maneuver eventually reaches the dynamic pressure constraint and causes the trajectory to end, as no further path is feasible.

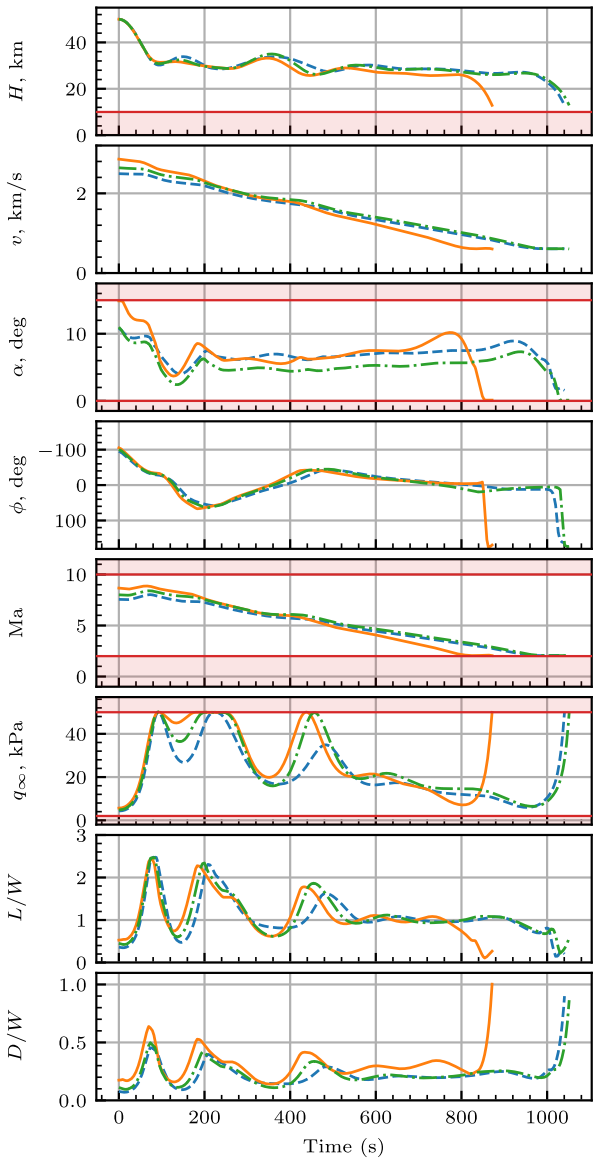
5.2.3. *Aerodynamic Operating Regimes*

Figure 8 shows the lift-to-drag ratio of each vehicle over the Mach number–angle-of-attack envelope, with the optimal sprint and obstacle trajectories overlaid. Both trajectories for a vehicle start at the same Mach number, as the missions share the same initial states, but at different angles of attack, which are free initial controls. All three vehicles show similar profiles for the mission trajectories. The sprint missions start at an angle of attack that nearly achieves the maximum lift-to-drag ratio for that Mach number. The angle of attack then increases to generate lift for the pull-up maneuver, before decreasing to a lower angle of attack to minimize drag. To achieve the maximum distance in the finite time the vehicles primarily operate at an angle of attack below the maximum L/D for the given Mach number, favoring the lower drag over pure aerodynamic efficiency. The obstacle missions differ in behavior and span a much larger Mach number range. The most prominent feature of the obstacle trajectories for all vehicles is that the angle of attack is chosen to stay at the maximum lift-to-drag for any given Mach number. The optimal trajectory is unable to follow this ridge line perfectly due to other trajectory constraints.

Comparing the lift-to-drag contours between the vehicles, the sprint vehicle has the lowest lift-to-drag values across the envelope. The point-design vehicle has both the highest and the lowest lift-to-drag values. It is optimized for the middle of the flight envelope, and the lift-to-drag values in the low-α region are noticeably worse than the other vehicles. As the angle-of-



(a) Spatial profile



(b) Temporal profile

Figure 7: Trajectories for the obstacle mission. Lift and drag are normalized by vehicle weight W.

attack is an open-loop control variable, $\mathcal{V}_{\text{pd}}$ is able to avoid the poor performing low-α region. Across the obstacle trajectory, the time-averaged lift-to-drag for $\mathcal{V}_{\text{pd}}$ (L/D = 5.0) is higher than that for $\mathcal{V}_{\text{ob}}$ (L/D = 4.9) or $\mathcal{V}_{\text{sp}}$ (L/D = 3.8). This is further evidence of the optimizer trading aerodynamic efficiency for mass minimization to improve the performance of $\mathcal{V}_{\text{ob}}$.

5.2.4. Summary

Table 7 summarizes the performance of all three vehicles across both missions. In each case the co-designed vehicle outperforms the alternatives on its own mission, demonstrating the benefit of jointly optimizing vehicle design and trajectory. $\mathcal{V}_{\text{ob}}$ performs better than $\mathcal{V}_{\text{pd}}$ for both missions, by 6.4% on the sprint mission and 5.3% on the obstacle mission, while $\mathcal{V}_{\text{sp}}$ is only better than $\mathcal{V}_{\text{pd}}$ for the sprint mission.

Table 7: Vehicle performance across both missions. Bold indicates the best performance for the mission. Percentages are the change relative to the point design $\mathcal{V}_{\text{pd}}$ flying the same mission.

	Distance, km	
Vehicle	Sprint mission	Obstacle mission
$\mathcal{V}_{\text{pd}}$	940	1456
$\mathcal{V}_{\text{sp}}$	**1063** (+13%)	1342 (-7.8%)
$\mathcal{V}_{\text{ob}}$	1000 (+6.4%)	**1533** (+5.3%)

The results presented here rely on low-fidelity aerodynamics for expediency. Whether the resulting designs and conclusions survive a higher-fidelity aerodynamic model, and at what computational cost, is the subject of the following section.

6. High-Fidelity Co-Design Results

To better reflect the realistic performance of these vehicles, this section repeats the sprint mission co-design using inviscid CFD, in place of the local surface inclination method, to calculate the aerodynamic forces. The aims are to show that co-design remains tractable with CFD and the approximate flow gradients, and to investigate the utility of the low-fidelity solution for initializing the high-fidelity optimization. Two CFD co-design optimizations are performed, differing only in their initial design parameters. $\mathcal{V}^{\text{CFD}}_{\text{sp};c}$ is cold-started from the midpoint of the design parameter ranges, as for the low-fidelity optimizations, while $\mathcal{V}^{\text{CFD}}_{\text{sp};w}$ is warm-started from the optimal low-fidelity sprint design, $\mathcal{V}_{\text{sp}}$.

6.1. Design Comparison

The geometries of the optimal sprint vehicles, presented in Fig. 9, are visually similar, particularly between $\mathcal{V}_{\text{sp}}$ and $\mathcal{V}^{\text{CFD}}_{\text{sp};w}$. The cross-sections of Fig. 10 show that $\mathcal{V}^{\text{CFD}}_{\text{sp};c}$ differs mainly in position rather than shape: with a similar planform, its body sits higher toward the base. Because the nose and leading edges are fixed at $z = 0$ m in the geometry generation, an offset that grows toward the base is equivalent to pitching the vehicle about its nose, changing its built-in incidence.

To quantitatively evaluate the similarity of the vehicle design parameters, a design parameter differential is defined. The design parameter vector of a vehicle $\mathcal{V}$ is denoted $\boldsymbol{x}_{\text{v}}(\mathcal{V}) \in \mathbb{R}^{N_d}$,

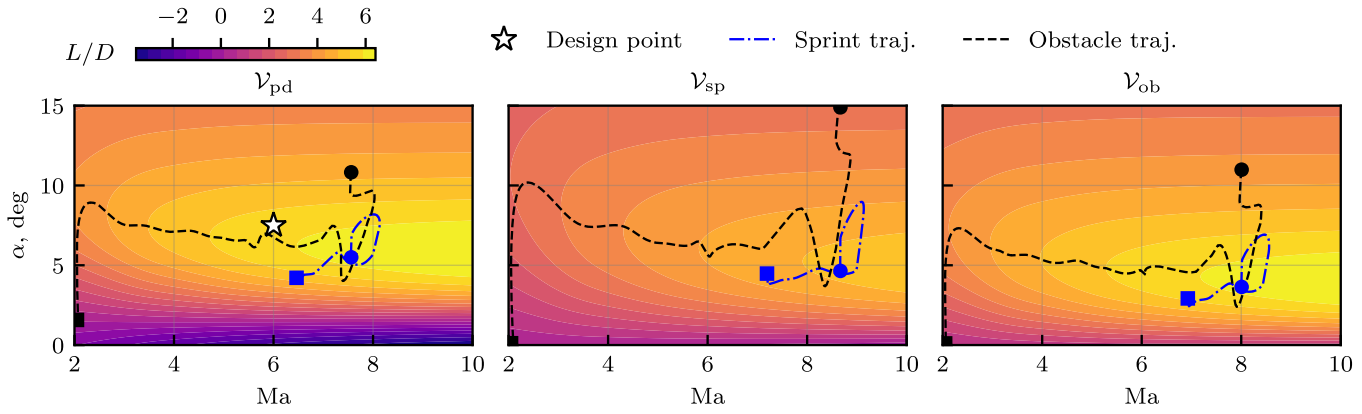
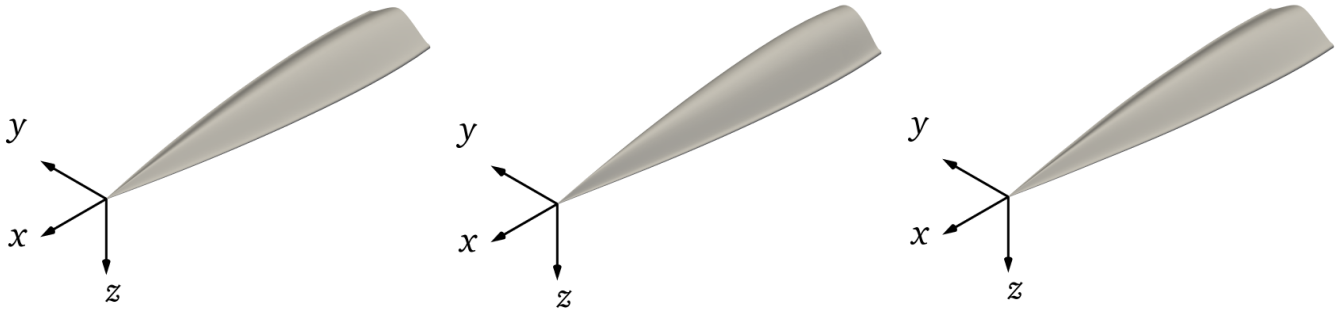


Figure 8: Mach–angle-of-attack L/D contour maps for $\mathcal{V}_{\text{pd}}$, $\mathcal{V}_{\text{sp}}$, and $\mathcal{V}_{\text{ob}}$, with mission trajectories overlaid. Circles mark the start of each trajectory and squares the end.



(a) $\mathcal{V}_{\text{sp}}$ (b) $\mathcal{V}^{\text{CFD}}_{\text{sp};c}$ (c) $\mathcal{V}^{\text{CFD}}_{\text{sp};w}$

Figure 9: Isometric views of the sprint vehicles

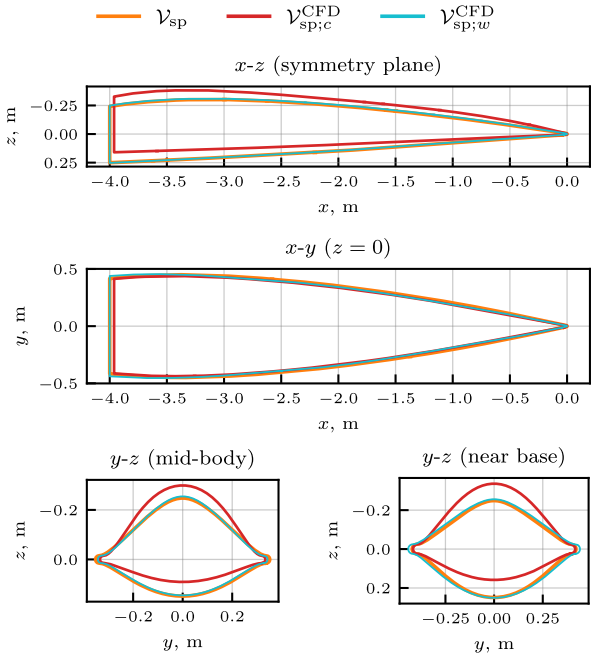


Figure 10: Cross-section outlines of $\mathcal{V}_{\text{sp}}$, $\mathcal{V}^{\text{CFD}}_{\text{sp};c}$ and $\mathcal{V}^{\text{CFD}}_{\text{sp};w}$ overlaid on common axes: the x-z symmetry plane, the x-y planform at $z = 0$, and y-z cuts at mid-body and near the base. The nose is at $x = 0$ and z is positive down.

and is normalized according to the range of each design variable,

$$\tilde{\boldsymbol{x}}_{\text{v}}(\mathcal{V}) \coloneqq \frac{\boldsymbol{x}_{\text{v}}(\mathcal{V}) - \boldsymbol{x}_{\text{v}_{\text{min}}}}{\boldsymbol{x}_{\text{v}_{\text{max}}} - \boldsymbol{x}_{\text{v}_{\text{min}}}}, \tag{26}$$

such that $\tilde{\boldsymbol{x}}_{\text{v}}(\mathcal{V}) \in [0,1]^{N_d}$. The difference between the design parameter vectors of two vehicles, $\mathcal{V}_a$ and $\mathcal{V}_b$, is then given by a dimensionally scaled root-mean-square (RMS) difference between the vectors,

$$\sigma_d(\mathcal{V}_a, \mathcal{V}_b) \coloneqq \frac{1}{\sqrt{N_d}} \left\| \tilde{\boldsymbol{x}}_{\text{v}}(\mathcal{V}_a) - \tilde{\boldsymbol{x}}_{\text{v}}(\mathcal{V}_b) \right\|_2 , \tag{27}$$

with the resulting value of σ representing the average difference of the design parameters, with $\sigma \in [0,1]$. A value of $\sigma = 0$ represents two vehicles with identical parameters, and $\sigma = 1$ is obtained when two vehicles are at opposite corners of the hypercube design space. The difference parameter is calculated between all vehicle designs and is presented in the upper triangle of Fig. 11. The lower triangle presents the geometric counterpart, the RMS displacement between corresponding nodes on the vehicle surfaces,

$$\sigma_g(\mathcal{V}_a, \mathcal{V}_b) \coloneqq \frac{1}{\sqrt{N_{\text{nodes}}}} \left\| \boldsymbol{X}(\mathcal{V}_a) - \boldsymbol{X}(\mathcal{V}_b) \right\|_F , \tag{28}$$

where $\|\cdot\|_F$ is the Frobenius norm, so that σ_g is the RMS distance, in meters, between corresponding nodes. Every vehicle's surface is evaluated at the same interpolation coordinates on each surface patch, so node k has a corresponding node for every vehicle, and its position differs between vehicles only through the design variables. Each pair of vehicles is therefore compared in both design space and geometric space. The main diagonal is left blank, as these cells compare each vehicle with itself.

The upper-left 3×3 cells are a comparison of the low-fidelity optimized vehicles with one another, representing the inter-mission differences. These values compare how the optimized designs vary for different problems or missions. The differences between the three panel-optimized vehicles, representing inter-mission differences, are large in both the parametric and geometric space, matching the observation from the visual differences (see Fig. 4 and 5).

The bottom-right 3×3 cells compare the different sprint vehicles. The largest design parameter difference is $\sigma_d(\mathcal{V}_{\text{sp}},\mathcal{V}^{\text{CFD}}_{\text{sp};c})= 0.106$, comparable to the inter-mission differences. The geometric differences between the sprint vehicles are much smaller though: the largest sprint vehicle σ_g is less than half of the smallest inter-mission σ_g. Part of the difference between the sprint vehicles is a rigid pitch: rotating $\mathcal{V}^{\text{CFD}}_{\text{sp};c}$ nose-up by 0.8° about the nose tip reduces the largest σ_g to the other sprint vehicles to 0.033 m. The sprint vehicles are therefore physically similar, even where they differ in design space. The two CFD-optimized vehicles have nonetheless converged to different solutions, as the different starting points led the gradient-based optimizer along different search paths. The observed tilt suggests there is a direction in the design space corresponding to body pitch, along which the vehicle and the trajectory's angle of attack can trade against each other. Despite these differences, the geometric similarity of the sprint vehicles shows that, for this vehicle parameterization, the mission has a larger effect on the vehicle design than the choice of aerodynamic fidelity. Larger physical variations between fidelity levels may be observed in designs where capturing the shock structure and shock interactions is more important [37]. The largest difference in Fig. 11 is between $\mathcal{V}_{\text{pd}}$ and $\mathcal{V}^{\text{CFD}}_{\text{sp};c}$, in both design and physical space. This separation is driven by the point-design formulation rather than by fidelity, since $\mathcal{V}_{\text{pd}}$ is almost as far from the low-fidelity $\mathcal{V}_{\text{sp}}$ (0.231 m) as from $\mathcal{V}^{\text{CFD}}_{\text{sp};c}$ (0.238 m).

6.2. *Aerodynamic and Mission Performance*

The similarity of the sprint vehicles' designs raises the question of how similar their aerodynamics and mission trajectories are. The Mach–angle-of-attack contour plots in Fig. 12 allow for a comparison of the aerodynamics for each vehicle. The contours are very similar, but it can be observed that the contours for $\mathcal{V}^{\text{CFD}}_{\text{sp};c}$ appear to be shifted up by approximately 1°, matching the observed tilt. Likewise, the trajectory through the Mach–angle-of-attack space for the three vehicles appears nearly identical, just with $\mathcal{V}^{\text{CFD}}_{\text{sp};c}$ operating at a slightly higher angle of attack. Evaluating the time-averaged angle of attack, $\mathcal{V}^{\text{CFD}}_{\text{sp};c}$ sits at 6.34° whereas $\mathcal{V}^{\text{CFD}}_{\text{sp};w}$ is 5.34°. The average L/D over the trajectory for each vehicle is similar with $L/D = 4.65$ for $\mathcal{V}^{\text{CFD}}_{\text{sp};c}$ and $L/D = 4.63$ for $\mathcal{V}^{\text{CFD}}_{\text{sp};w}$, reinforcing their physical aerodynamic similarities.

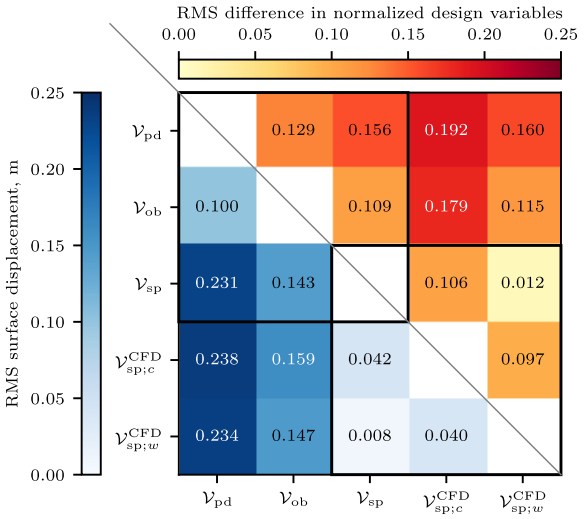


Figure 11: Differences between all vehicle pairs: design parameter differential σ (upper triangle) and RMS surface displacement (lower triangle).

Also included in the plots are the locations of the aerodynamic evaluations, with the CFD cases using a sparser set to save on compute cost. The sprint trajectory occupies only a small region of the Mach–angle-of-attack space, but the cubic-spline surrogates require at least four points in each direction across the full envelope, so many evaluations lie in regions the trajectory never visits. A more efficient methodology would be to utilize a multi-fidelity approach that uses higher fidelity data near the trajectory path and lower fidelity methods further away [59, 10].

The mission performance of the three vehicles, listed in Table 8, is likewise almost identical: the distances differ by at most 1.6 km (0.15 %). Although the two CFD optimizations started from different points, followed approximate gradients, and converged to different designs, they reached the same mission performance, which supports the robustness of the optimization methodology. This difference is small compared with the effect of the modeling assumptions, such as the omission of viscous forces and of the control surface deflections needed to trim the vehicle, either of which would change the predicted range by considerably more.

Table 8: Downrange distances of the sprint co-designed vehicles.

Vehicle	Aerodynamics	Distance, km
$\mathcal{V}_{\text{sp}}$	Surface inclination	1063.1
$\mathcal{V}^{\text{CFD}}_{\text{sp};c}$	CFD	1063.7
$\mathcal{V}^{\text{CFD}}_{\text{sp};w}$	CFD	1062.1

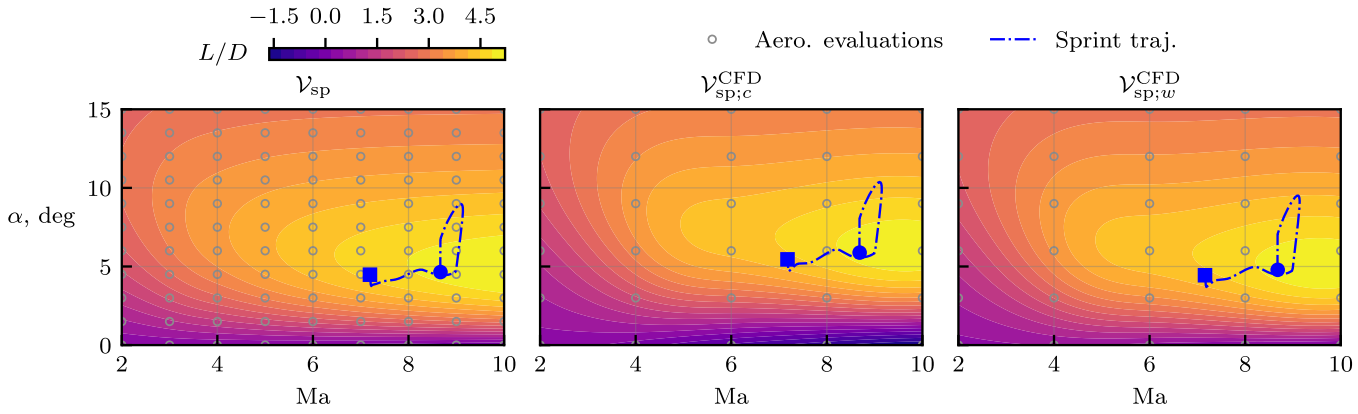


Figure 12: Mach–angle-of-attack L/D contour maps for $\mathcal{V}_{\text{sp}}$, $\mathcal{V}_{\text{sp;}c}^{\text{CFD}}$, and $\mathcal{V}_{\text{sp;}w}^{\text{CFD}}$, with sprint mission trajectories overlaid. Solid circles mark the start of each trajectory and squares the end.

6.3. Computational Cost

The computational cost of all the optimization cases is presented in Table 9. The core-hours were calculated using the University of Southern Queensland's *Fawkes* high-performance computing cluster. *Fawkes* is equipped with AMD EPYC 7702 processors, which contain 64 cores. While the low-fidelity optimization problems are performed using a single core, the high-fidelity cases use all 64 to run 8 CFD simulations in parallel, each simulation with 8 cores. The point-design optimization is the cheapest option, as there is no trajectory optimization performed. The trajectory optimization is the most computationally demanding component of the low-fidelity co-design optimizations, increasing the average time per iteration to two minutes for the sprint mission and four minutes for the obstacle mission. The obstacle problem takes longer per iteration than the sprint problem due to the additional spatial constraints in the problem requiring more control input, as well as the large flexibility in the value of t_f. For the high-fidelity aerodynamic modeling, the optimization takes around 3 h to 3.5 h per iteration, dominated by the CFD simulations. $\mathcal{V}_{\text{sp;}c}^{\text{CFD}}$ has the largest average iteration clock time, as the starting design vehicle is large, increasing the computational cost to simulate with CFD. $\mathcal{V}_{\text{sp;}c}^{\text{CFD}}$ requires slightly more iterations than the lower-fidelity approaches. This is in part due to the approximate gradients used for the aerodynamics, which slows down the convergence. Using the *Fawkes* cluster, $\mathcal{V}_{\text{sp;}c}^{\text{CFD}}$ requires 10 270 core-hours, taking nearly seven days. Faster results are obtained by warm-starting. Owing to the similarity between $\mathcal{V}_{\text{sp}}$ and $\mathcal{V}_{\text{sp;}w}^{\text{CFD}}$, $\mathcal{V}_{\text{sp;}w}^{\text{CFD}}$ requires only 9 iterations and 1.1 days on *Fawkes*. This is still a dramatic reduction compared to $\mathcal{V}_{\text{sp;}c}^{\text{CFD}}$. Accounting for the time required for the low-fidelity optimization, the total time comes to 28 hours for the warm-start optimization, less than a fifth of the time of the cold-start CFD optimization. In Appendix B, two additional panel schemes are used for the warm start, showing a consistent benefit in warm starting.

Table 9: Optimization computational cost

Vehicle	Outer iterations	Wall time per iteration, min	Core-hours
$\mathcal{V}_{\text{pd}}$	32	0.38	0.2
$\mathcal{V}_{\text{sp}}$	31	2.2	1.1
$\mathcal{V}_{\text{ob}}$	22	4.1	1.5
$\mathcal{V}_{\text{sp;}c}^{\text{CFD}}$	46	209	10270
$\mathcal{V}_{\text{sp;}w}^{\text{CFD}}$	9	181	1733

7. Conclusion

The optimization of hypersonic vehicles requires the evaluation of several disciplines, each with unique constraints on the dynamic system. This work presents a gradient-based, nested control co-design framework for the optimization of hypersonic vehicles and their trajectories for specified missions. The nested co-design layout is ideal for problems where the trajectory optimization is a minor cost compared to evaluating the vehicle's physics and constructing surrogates, as is the case for systems that rely on expensive computational fluid dynamics simulations to evaluate the aerodynamics. The nested approach reduces the dimensionality of the optimization problem, reducing the number of iterations and expensive computations needed to be performed. The nested co-design approach is also amenable to gradient-based local optimization. By propagating sensitivities, obtained here by automatic differentiation, surrogate differentiation, or approximation, the optimizer can efficiently search the design space, allowing for a higher dimensional design space to remain tractable. The modularity of the framework allows for additional physics and models to be incorporated into the design process, as long as they are differentiable or reasonable derivatives can be approximated efficiently.

The framework is applied to the optimization of a vehicle for three separate missions, using low-fidelity local surface inclination methods for the aerodynamics. The first vehicle, $\mathcal{V}_{\text{pd}}$, is a point-designed vehicle with an objective based on mass and lift-to-drag at a single design point. The second vehicle, $\mathcal{V}_{\text{sp}}$, is

co-designed for a sprint mission to achieve the maximum distance in a finite amount of time. The third vehicle, $\mathcal{V}_{\text{ob}}$, is co-designed for an obstacle mission to achieve the maximum range while maneuvering around the spatial constraints. Evaluation of the vehicles across the three missions shows that the co-designed vehicles perform the best at their respective mission, with $\mathcal{V}_{\text{sp}}$ beating $\mathcal{V}_{\text{pd}}$ by 13% for the sprint mission and $\mathcal{V}_{\text{ob}}$ beating $\mathcal{V}_{\text{pd}}$ by 5% for the obstacle mission.

The modularity of the control co-design framework allows different physics models to be swapped out for higher fidelity versions. Swapping out the low-fidelity aerodynamic model for higher-fidelity inviscid CFD, the sprint mission is optimized using a cold-start (starting from the middle of the design space) and a warm-start (starting from the low-fidelity solution). While the two CFD-optimized vehicles converged to slightly different solutions, the three sprint vehicles are more physically similar to each other than they are to $\mathcal{V}_{\text{pd}}$ or $\mathcal{V}_{\text{ob}}$. This shows that escalating fidelity is a viable approach for vehicle design, with the warm-start solution also requiring less than 20 % of the computational time to converge.

The co-designed vehicles are specialized to their missions, as shown by $\mathcal{V}_{\text{sp}}$ performing the worst on the obstacle mission. Co-design is therefore well suited to maximizing performance for a single mission, but further work is needed where multi-mission performance is desired. A weighted sum of the mission objectives is one option, although the resulting design is sensitive to the chosen weights and normalizations. Varying these traces out a Pareto front, which could also be identified directly with a multi-objective optimizer; however, many such optimizers are gradient-free evolutionary algorithms that scale poorly with the number of design variables. Alternatively, an ϵ-constraint approach optimizes one mission while constraining the performance on the others to remain above specified thresholds. Starting from an existing feasible design, this improves one mission while at least maintaining the others, but without a feasible starting design the thresholds may not all be achievable.

The presented optimization of hypersonic glide vehicles has several assumptions and limitations. Thermal effects, structural requirements, control surfaces, and viscous aerodynamic forces are not included in the analysis. However, these models can be included to improve the realism of the modeling process. Likewise, propulsion, multi-stage, and subsonic behavior can be incorporated. The development and incorporation of these additional physics modules is ongoing work within the research group [60, 61, 21, 62]. Even with the simplified analysis, the control co-design framework demonstrates utility for the optimization of hypersonic glide vehicles.

Funding Sources

Funding was received from the Australian Government through the Australian Research Council grant ARC LP190100104. This material is based upon work supported by the Air Force Office of Scientific Research under award number FA2386-22-1-4074. Any opinions, findings and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the U.S. Department of Defense.

CRediT authorship contribution statement

Bradley Pascoe: Writing - original draft, Software, Visualization, Formal Analysis. **Andrew Lock**: Conceptualization, Software, Writing – review & editing, Methodology. **Jack Williamson**: Software, Writing – review & editing. **Robert Watt**: Software, Writing – review & editing. **Ingo Jahn**: Conceptualization, Writing – review & editing, Supervision, Resources, Funding acquisition.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Appendix A. Local Surface Inclination Methods

The four schemes used to evaluate the pressure coefficient $C_p(\delta, \text{Ma}, \gamma)$ are described here, together with their domains of applicability. A representative comparison of C_p as a function of δ across a range of Mach numbers is shown in Fig. A.13. Definitions of these schemes can be found in Anderson [63].

Windward panels ($\delta > 0$)

The primary scheme for windward-facing panels is the *tangent cone* method, which estimates the surface pressure by assuming each panel lies on a cone with the same inclination angle to the flow as the local panel angle, with an attached oblique shock. This scheme was chosen for two reasons. For blunt bodies, the *Modified Newtonian* scheme is typically used; however, the leading edge and nose radius was considered sufficiently small that the vehicle was not considered a blunt body [64]. Due to the three-dimensional characteristics of the vehicles, the tangent cone was predicted to be more accurate than the *oblique shock* scheme which has been previously used [56, 21]. The tangent cone scheme is not valid for all positive deflection angles: beyond a Mach-dependent maximum cone/inclination angle the theoretical shock detaches. For panels exceeding this limit, a *normal shock* (detached shock) scheme is applied as a fallback. While the usage of the normal shock does not achieve the maximum stagnation pressure at $\delta = \pi/2$, it does maintain monotonic behavior which would not be achieved if the Modified Newtonian scheme was used as the high-δ fallback scheme.

Rearward panels ($\delta < 0$)

The primary scheme for rearward-facing panels is the *Prandtl–Meyer* expansion, which assumes an isentropic expansion fan from the freestream conditions to the local panel angle. The Prandtl–Meyer scheme is similarly not valid for all negative deflection angles. For panels beyond the applicable range, a *rear-face* scheme is used, which sets the pressure equal to zero. This treatment is consistent with the limiting value of the Prandtl–Meyer expansion, as seen in Fig. A.13.

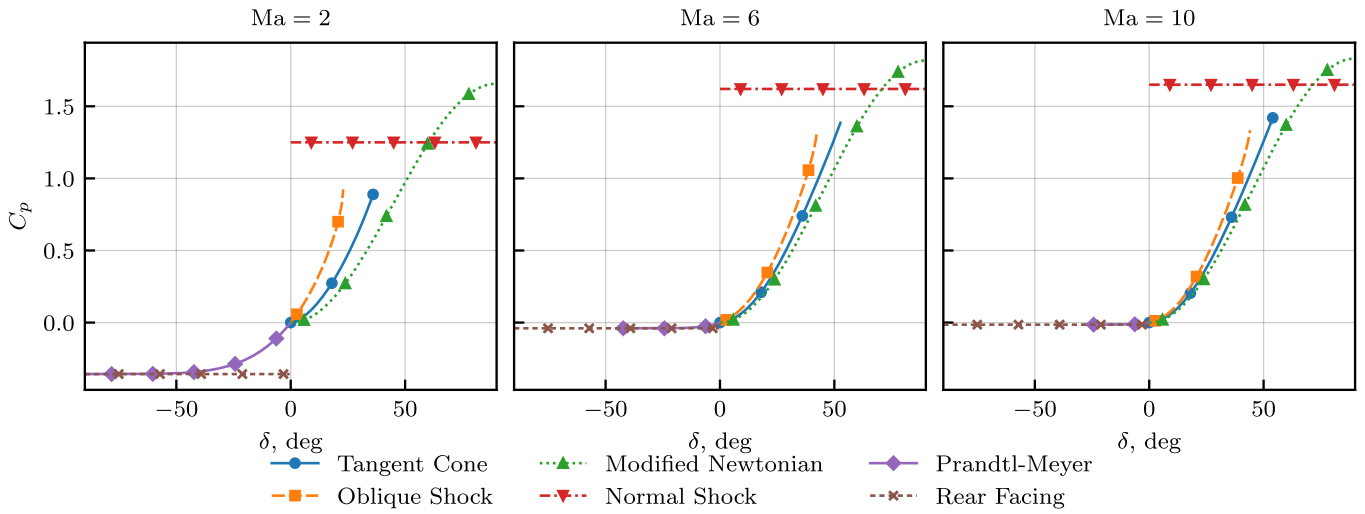


Figure A.13: Pressure coefficient as a function of panel inclination angle for each local surface inclination scheme, at Mach 2, 6 and 10 with $\gamma = 1.4$. Each scheme is drawn only over the inclination range where it is applicable.

Sensitivity of C_p with respect to δ

For the tangent cone and Prandtl–Meyer schemes, C_p is not an explicit function of the surface inclination and is instead obtained through an iterative search. To reduce the cost of the evaluations, for each (Ma, γ) pair, $C_p(\delta; \text{Ma}, \gamma)$ is sampled as a function of δ and fitted with a cubic spline interpolator. Although $\partial C_p/\partial\delta$ could be obtained with implicit differentiation at the converged state, the analytic derivative of the splines is used. These derivatives are exactly consistent with the C_p function used in the solver and are inexpensive to evaluate at runtime, requiring no further iterations. For the normal shock and rear-face schemes, C_p is independent of the surface inclination, providing $\partial C_p/\partial\delta = 0$. Panels which switch between these schemes and an inclination-dependent scheme experience a discontinuity in $\partial C_p/\partial\delta$, which can degrade the convergence of gradient-based methods.

Appendix B. Warm start comparison

It is expected that the efficacy of the warm-start optimization for high-fidelity is dependent upon the quality of the low-fidelity model. Section 6 calculated the computational savings obtained by using the tangent cone solution for warm-starting the CFD optimization problem. To further show the benefit of warm-starting, the warm-start optimization is performed using two additional low-fidelity schemes: *Modified Newtonian* and *Oblique Shock/Normal Shock*, both of which are discussed in Appendix A. These two runs are otherwise identical to the warm start of Section 6, except for starting from a different vehicle design.

The results are listed in Table B.10. Every warm start converges in fewer than a fifth of the 46 outer iterations required by the cold-started $\mathcal{V}^{\text{CFD}}_{\text{sp};c}$, so the saving is not specific to the tangent cone seed. The two new seeds in fact converge marginally faster than the featured one, in 8 and 6 iterations against 9, despite both predicting the performance of their own designs less accurately than the tangent cone scheme. The iteration count is therefore not simply dependent upon the accuracy of the low-fidelity model relative to the higher-fidelity method.

The three low-fidelity schemes obtain different optimum designs at low fidelity, with a range of 13.1 km in their obtained values. After fidelity escalation with identical numerics, that spread collapses to 0.4 km, from 1061.7 km to 1062.1 km. While the tangent cone seed is the most accurate in matching the corresponding CFD-optimized vehicle's distance, the Modified-Newtonian-seeded vehicle moves the smallest distance from its seed vehicle. In this regard, all three low-fidelity methods can be justified for use as the low-fidelity method.

The three warm-started designs lie within $\sigma = 0.044$ of one another, whereas each lies $\sigma = 0.095$ to 0.106 from $\mathcal{V}^{\text{CFD}}_{\text{sp};c}$, and all three fall 1.6 km to 2.0 km short of its 1063.7 km. The warm starts therefore share a common solution that is distinct from the cold-started one. The distinction is largely one of incidence rather than shape. Evaluated with their CFD surrogates at Mach 8.3, all three warm-started vehicles have a zero-lift angle of attack of 0.0° and fly the sprint mission at a time-averaged angle of attack of 5.3°, while $\mathcal{V}^{\text{CFD}}_{\text{sp};c}$ has a zero-lift angle of attack of 0.9° and flies at 6.3°, along a near-identical Mach–altitude profile. Warm starting therefore reduces the cost of reaching a converged CFD design to a fraction of the cold start's, at the price of a performance shortfall of under 0.2 %.

Acknowledgments

This research was undertaken using the University of Southern Queensland (UniSQ) Fawkes HPC.

Table B.10: Cost and outcome of the warm-started CFD sprint co-designs. Core-hours are quoted for the 64 cores used per optimization.

				Sprint distance, km		Design differential σ_d	
Seed method	Iterations	Wall time, h	Core-hours	Low fidelity	CFD	From seed	To $\mathcal{V}_{\mathrm{sp};c}^{\mathrm{CFD}}$
Tangent cone	9	27.1	1733	1063.1	1062.1	0.012	0.097
Modified Newtonian	8	25.0	1598	1057.1	1061.7	0.011	0.095
Oblique shock	6	21.9	1399	1070.2	1062.0	0.017	0.106

References

[1] T. R. F. Nonweiler, Aerodynamic problems of manned space vehicles, The Aeronautical Journal 63 (585) (1959) 521–528. doi:10.1017/S0368393100071662.

[2] J. Jones, K. Moore, J. Pike, P. Roe, A method for designing lifting configurations for high supersonic speeds, using axisymmetric flow fields, Ingenieur-Archiv 37 (1) (1968) 56–72. doi:10.1007/BF00532683.

[3] H. Sobieczky, F. C. Dougherty, K. Jones, Hypersonic waverider design from given shock waves, in: First International Waverider Symposium, University of Maryland, College Park, MD, 1990.

[4] T.-t. Zhang, Z.-g. Wang, W. Huang, L. Yan, Parameterization and optimization of hypersonic-gliding vehicle configurations during conceptual design, Aerospace Science and Technology 58 (2016) 225–234. doi:10.1016/j.ast.2016.08.020.

[5] M. Yin, E. He, Y. Li, C. Zhang, Aerodynamic shape optimization of hypersonic vehicle based on improved class-shape-transformation method, Advances in Space Research 76 (9) (2025) 5518–5534. doi:10.1016/j.asr.2025.08.015.

[6] J. Liu, S. Xu, Y. Han, Y. Deng, Y. Wang, Parametric modeling and optimization of hypersonic glide vehicles under stringent loading constraints, Physics of Fluids 38 (8) (2026) 086112. doi:10.1063/5.0339067.

[7] F. Ding, J. Liu, C.-b. Shen, Z. Liu, S.-h. Chen, X. Fu, An overview of research on waverider design methodology, Acta Astronautica 140 (2017) 190–205. doi:10.1016/j.actaastro.2017.08.027.

[8] Z.-t. Zhao, W. Huang, L. Yan, Y.-g. Yang, An overview of research on wide-speed range waverider configuration, Progress in Aerospace Sciences 113 (2020) 100606. doi:10.1016/j.paerosci.2020.100606.

[9] A. M. Oliveira, C. S. Cerqueira, A systematic review of MDO methods applied to launch vehicle design and their contributions to concurrent engineering, Discover Mechanical Engineering 4 (1) (2025) 19. doi:10.1007/s44245-025-00104-8.

[10] K. Haque, A. Meo, A. Meini, M. Panesi, Multi-fidelity surrogate-based trajectory-aware shape optimization of hypersonic vehicles, in: AIAA SCITECH 2026 Forum, American Institute of Aeronautics and Astronautics, Orlando, FL, 2026, p. 2026–2301. doi:10.2514/6.2026-2301.

[11] L. Iuspa, A. Aprovitola, G. Pezzella, V. Cristillo, A. Viviani, Multi-disciplinary optimization of a space reentry vehicle using skeleton-based integral soft objects, Aerospace Science and Technology 131 (2022) 107996. doi:10.1016/j.ast.2022.107996.

[12] N. Montella, L. Iuspa, G. A. Vio, A. Aprovitola, G. Pezzella, A. Viviani, Aeroshape design and low speed aerodynamic analysis of a crew return vehicle, Aerospace Science and Technology 158 (2025) 109876. doi:10.1016/j.ast.2024.109876.

[13] H. Fathy, J. Reyer, P. Papalambros, A. Ulsoy, On the coupling between the plant and controller optimization problems, in: Proceedings of the 2001 American Control Conference. (Cat. No.01CH37148), Vol. 3, 2001, pp. 1864–1869. doi:10.1109/ACC.2001.946008.

[14] J. T. Allison, D. R. Herber, Multidisciplinary design optimization of dynamic engineering systems, AIAA Journal 52 (4) (2014) 691–710. doi:10.2514/1.J052182.

[15] D. R. Herber, J. T. Allison, Nested and simultaneous solution strategies for general combined plant and control design problems, Journal of Mechanical Design 141 (1) (2019) 011402. doi:10.1115/1.4040705.

[16] B. G. Coulter, D. Huang, Z. Wang, Geometric design of hypersonic vehicles for optimal mission performance with high-fidelity aerodynamic models, Journal of Aircraft 60 (3) (2023) 870–882. doi:10.2514/1.C036980.

[17] J. Cangelosi, M. Heinkenschloss, J. T. Needels, J. J. Alonso, Simultaneous design and trajectory optimization for boosted hypersonic glide vehicles, in: AIAA SCITECH 2024 Forum, American Institute of Aeronautics and Astronautics, Orlando, FL, 2024, pp. 2024–0375. doi:10.2514/6.2024-0375.

[18] S. Ma, Y. Yang, Y. Chen, H. Yang, W. Chen, Multidisciplinary design optimization of reentry-powered hypersonic vehicles based on surrogate model, International Journal of Aerospace Engineering 2024 (1) (2024) 5557153. doi:10.1155/2024/5557153.

[19] X. Pan, L. Jiang, Y. Wang, T. Wang, G. Chen, Aerodynamic–control–trajectory collaborative optimization methods for hypersonic reusable vehicles, Aerospace Science and Technology 179 (2026) 113598. doi:10.1016/j.ast.2026.113598.

[20] C. van der Heide, A. Lock, V. Bone, K. Mackle, I. H. Jahn, P. M. Dower, C. Manzie, Multi-mission codesign of a hypersonic vehicle leading edge with heat flux constraints, in: AIAA SCITECH 2025 Forum, AIAA SciTech Forum, American Institute of Aeronautics and Astronautics, Orlando, FL, 2025, pp. 2025–0955. doi:10.2514/6.2025-0955.

[21] J. Williamson, B. Pascoe, A. Lock, I. H. Jahn, Simultaneous vehicle design and trajectory optimisation of a multi-stage hypersonic boost-glide system, in: 27th AIAA International Space Planes and Hypersonic Systems and Technologies Conference, American Institute of Aeronautics and Astronautics, Naples, Italy, 2026, p. 2026–5003. doi:10.2514/6.2026-5003.

[22] C. van der Heide, V. Bone, P. M. Dower, C. Manzie, I. Jahn, Hypersonic vehicle co-design for multi-stage mission planning, in: AIAA SCITECH 2026 Forum, American Institute of Aeronautics and Astronautics, Orlando, FL, 2026, p. 2026–0369. doi:10.2514/6.2026-0369.

[23] C. van der Heide, P. Cudmore, I. Jahn, V. Bone, P. M. Dower, C. Manzie, Feasibility detection for nested codesign of hypersonic vehicles, in: 2023 62nd IEEE Conference on Decision and Control (CDC), 2023, pp. 632–637. doi:10.1109/CDC49753.2023.10384006.

[24] S. Kaneko, J. R. R. A. Martins, Simultaneous design and trajectory optimization strategies for computationally expensive models, AIAA Journal 63 (2) (2025) 420–438. doi:10.2514/1.J063976.

[25] A. B. Lambe, J. R. R. A. Martins, Extensions to the design structure matrix for the description of multidisciplinary design, analysis, and optimization processes, Structural and Multidisciplinary Optimization 46 (2) (2012) 273–284. doi:10.1007/s00158-012-0763-y.

[26] J. R. R. A. Martins, A. Ning, Engineering Design Optimization, Cambridge University Press, Cambridge, UK, 2021. doi:10.1017/9781108980647.

[27] Y. Yu, Z. Lyu, Z. Xu, J. R. R. A. Martins, On the influence of optimization algorithm and initial design on wing aerodynamic shape optimization, Aerospace Science and Technology 75 (2018) 183–199. doi:10.1016/j.ast.2018.01.016.

[28] J. A. E. Andersson, J. B. Rawlings, Sensitivity analysis for nonlinear programming in CasADi, IFAC-PapersOnLine 51 (20) (2018) 331–336. doi:10.1016/j.ifacol.2018.11.055.

[29] D. Kraft, A software package for sequential quadratic programming, Forschungsbericht DFVLR-FB 88-28, Deutsche Forschungs- und Versuchsanstalt für Luft- und Raumfahrt (DFVLR), Köln, Germany (1988).

[30] D. Kraft, Algorithm 733: TOMP–Fortran modules for optimal control calculations, ACM Trans. Math. Softw. 20 (3) (1994) 262–281. doi:10.1145/192115.192124.

[31] A. J. Joshy, J. T. Hwang, PySLSQP: A transparent Python package for the SLSQP optimization algorithm modernized with utilities for visualization and post-processing, Journal of Open Source Software 9 (103) (2024) 7246. doi:10.21105/joss.07246.

[32] K. Mackle, I. Jahn, Efficient and flexible methodology for the aerodynamic shape optimization of hypersonic vehicle concepts in a high-dimensional design space, in: AIAA SCITECH 2024 Forum, American Institute of Aeronautics and Astronautics, 2024, pp. 2024–2838. doi:10.2514/6.2024-2838.

[33] N. N. Gibbons, K. A. Damm, P. A. Jacobs, R. J. Gollan, Eilmer: An open-source multi-physics hypersonic flow solver, Computer Physics Communications 282 (2023) 108551. doi:10.1016/j.cpc.2022.108551.

[34] C. C. Margossian, A review of automatic differentiation and its efficient implementation, WIREs Data Mining and Knowledge Discovery 9 (4) (2019) e1305. doi:10.1002/widm.1305.

[35] K. Mackle, A. Lock, I. Jahn, C. van der Heide, Developing a co-design framework for hypersonic vehicle aerodynamics and trajectory, in: AIAA SCITECH 2024 Forum, AIAA SciTech Forum, American Institute of Aeronautics and Astronautics, 2024, pp. 2024–0238. doi:10.2514/6.2024-0238.

[36] ANSYS, Inc., ANSYS Fluent User's Guide, ANSYS, Inc., Canonsburg, PA, USA (2025).
URL `https://www.ansys.com/products/fluids/ansys-fluent`

[37] J. Jessen, J. Larsson, C. Brehm, Comparative optimization of hypersonic waveriders using analytical and computational methods, Aerospace Science and Technology 172 (2026) 111703. doi:10.1016/j.ast.2026.111703.

[38] J. Nassif, J.-J. O. Hoste, T. I. Józsa, Multi-point surrogate-based aerodynamic optimisation of a generic hypersonic waverider, Aerospace Science and Technology 175 (2026) 111978. doi:10.1016/j.ast.2026.111978.

[39] Z. J. Wang, A fast nested multi-grid viscous flow solver for adaptive Cartesian /Quad grids, International Journal for Numerical Methods in Fluids 33 (5) (2000) 657–680. doi:10.1002/1097-0363(20000715)33:5<657::AID-FLD24>3.0.CO;2-G.

[40] M.-S. Liou, A Sequel to AUSM: AUSM+, Journal of Computational Physics 129 (2) (1996) 364–382. doi:10.1006/jcph.1996.0256.

[41] I. Jahn, Towards an efficient method for calculating design parameter derivatives in super- and hypersonic flow, in: Proceedings of the 24th Australasian Fluid Mechanics Conference (AFMC), Australasian Fluid Mechanics Society (AFMS), Canberra, Australia, 2024, p. 31. doi:10.5281/zenodo.14213419.

[42] I. H. Jahn, Approximate surface pressures and heat load gradients for the optimisation of hypersonic vehicles, in: AIAA SCITECH 2025 Forum, AIAA SciTech Forum, American Institute of Aeronautics and Astronautics, Orlando, FL, 2025, pp. 2025–0952. doi:10.2514/6.2025-0952.

[43] J. T. Parker, A. Serrani, S. Yurkovich, M. A. Bolender, D. B. Doman, Control-oriented modeling of an air-breathing hypersonic vehicle, Journal of Guidance, Control, and Dynamics 30 (3) (2007) 856–869. doi:10.2514/1.27830.

[44] C. E. Rasmussen, C. K. I. Williams, Gaussian Processes for Machine Learning, The MIT Press, 2005. doi:10.7551/mitpress/3206.001.0001.

[45] V. Bone, C. van der Heide, K. Mackle, I. Jahn, P. M. Dower, C. Manzie, Gradient-enhanced deep Gaussian processes for multifidelity modeling, Journal of Computational Physics 520 (2025) 113474. doi:10.1016/j.jcp.2024.113474.

[46] B. Etkin, L. D. Reid, Dynamics of Flight: Stability and Control, 3rd Edition, Wiley, New York, 1996.

[47] D. K. Schmidt, D. Lisoski, B. P. Danowsky, Two-dimensional static margin for three-dimensional aircraft, Journal of Aircraft 62 (3) (2025) 748–750. doi:10.2514/1.C038107.

[48] A. Miele, Flight Mechanics: Theory of Flight Paths, Dover Publications, 2016.

[49] NOAA, NASA, USAF, U.S. Standard Atmosphere, 1976, Tech. Rep. NOAA-S/T 76-1562, U.S. Government Printing Office, Washington, D.C. (Oct. 1976).
URL https://ntrs.nasa.gov/citations/19770009539

[50] A. Wächter, L. T. Biegler, On the implementation of an interior-point filter line-search algorithm for large-scale nonlinear programming, Mathematical Programming 106 (1) (2006) 25–57. doi:10.1007/s10107-004-0559-y.

[51] J. A. E. Andersson, J. Gillis, G. Horn, J. B. Rawlings, M. Diehl, CasADi: a software framework for nonlinear optimization and optimal control, Mathematical Programming Computation 11 (1) (2019) 1–36. doi:10.1007/s12532-018-0139-4.

[52] P. Dai, D. Feng, W. Feng, J. Cui, L. Zhang, Entry trajectory optimization for hypersonic vehicles based on convex programming and neural network, Aerospace Science and Technology 137 (2023) 108259. doi:10.1016/j.ast.2023.108259.

[53] X. Hui, C. Guangbin, Z. Shengxiu, Y. Xiaogang, H. Mingzhe, Hypersonic reentry trajectory optimization by using improved sparrow search algorithm and control parametrization method, Advances in Space Research 69 (6) (2022) 2512–2524. doi:10.1016/j.asr.2021.12.030.

[54] Y. Wu, J. Deng, L. Li, X. Su, L. Lin, A hybrid particle swarm optimization-gauss pseudo method for reentry trajectory optimization of hypersonic vehicle with navigation information model, Aerospace Science and Technology 118 (2021) 107046. doi:10.1016/j.ast.2021.107046.

[55] J. Carvalho, E. Mooij, Integrated shape and trajectory optimisation of hypersonic waveriders, in: AIAA SCITECH 2024 Forum, American Institute of Aeronautics and Astronautics, Orlando, FL, 2024, p. 2024–0156. doi:10.2514/6.2024-0156.

[56] A. Lock, G. Oberman, I. H. Jahn, C. van der Heide, V. Bone, P. M. Dower, C. Manzie, Hypersonic glide vehicle shape and trajectory co-design, in: AIAA SCITECH 2025 Forum, American Institute of Aeronautics and Astronautics, Orlando, FL, 2025, pp. 2025–1337. doi:10.2514/6.2025-1337.

[57] A. J. Eggers, Jr., H. J. Allen, S. E. Neice, A comparative analysis of the performance of long-range hypervelocity vehicles, NACA Report 1382, National Advisory Committee for Aeronautics (1958).
URL https://ntrs.nasa.gov/citations/19930092363

[58] K. G. Bowcutt, Physics drivers of hypersonic vehicle design, in: 22nd AIAA International Space Planes and Hypersonics Systems and Technologies Conference, AIAA SPACE Forum, American Institute of Aeronautics and Astronautics, Orlando, FL, 2018, pp. 2018–5373. doi:10.2514/6.2018-5373.

[59] J. T. Needels, J. J. Alonso, Trajectory-informed sampling for efficient construction of multi-fidelity surrogate models for hypersonic vehicles, in: AIAA SCITECH 2024 Forum, American Institute of Aeronautics and Astronautics, Orlando, FL, 2024, pp. 2024–1013. doi:10.2514/6.2024-1013.

[60] A. Mittelman, K. Mackle, I. Jahn, R. Gollan, Sensitivity derivatives of a low-order integrated scramjet propulsion model for gradient-based co-design mdao, in: 3rd International Conference on High-Speed Vehicle Science & Technology, Busan, Korea, 2024, p. 2024–0323. doi:10.82241/ceas-hisst-2024-323.

[61] M. van Hoffen, B. Pascoe, J. Moran, D. Buttsworth, I. H. Jahn, Optimisation of vehicle geometry and control for dynamic manoeuvres at mach 7, in: 27th AIAA international space planes and hypersonic systems and technologies conference, American Institute of Aeronautics and Astronautics, Naples, Italy, 2026, p. 2026–5050. doi:10.2514/6.2026-5050.

[62] S. B. Koirala, I. H. Jahn, J. Williamson, A. Lock, D. Buttsworth, Trajectory optimization with thermal dynamic constraints in hypersonics, in: 27th AIAA international space planes and hypersonic systems and technologies conference, American Institute of Aeronautics and Astronautics, Naples, Italy, 2026, p. 2026–5092. doi:10.2514/6.2026-5092.
URL https://arc.aiaa.org/doi/abs/10.2514/6.2026-5092

[63] J. D. Anderson, Hypersonic and High-Temperature Gas Dynamics, 3rd Edition, AIAA education series, American Institute of Aeronautics and Astronautics, Reston, Va, 2019.

[64] D. Dirkx, E. Mooij, Conceptual Shape Optimization of Entry Vehicles, Springer Aerospace Technology, Springer, 2017.