



Efficient Hypersonic Guidance Through Neural Network-Based Trajectory Generation

Prince Edorh¹, Bruno Herisse²

Abstract

Hypersonic re-entry guidance poses significant challenges due to the sensitivity of optimal trajectories to dispersions in initial conditions and vehicle dynamics. Traditionally, guidance systems rely on precomputed optimal reference trajectories to mitigate these dispersions, employing on-line tracking algorithms to track to the nominal path. However, for large dispersions in initial conditions, this approach necessitates to embark either extensive databases of representative trajectories or robust online trajectory re-planning capabilities, both of which entail high computational or storage demands. Recent approaches have explored data compression techniques—such as Bézier curves or optimal shooting points — to reduce the storage required for representing optimal trajectories and commands. In this paper, we demonstrate that the relationship between stored representative trajectory data and initial condition dispersions can be effectively learned and subsequently leveraged onboard. Artificial neural networks have been trained offline using a limited number of optimal trajectories within an initial dispersion box. The trained model is then used online to quickly recompute an initial reference trajectory suitable for guidance algorithms such as Bézier curves-based guidance (BCBG) or Proportional Navigation (PN). This approach enables coverage of a large dispersion box in Monte Carlo simulations while satisfying precision requirements and various path constraints significantly reducing computational and storage demands while maintaining robustness to dispersions.

Keywords: *Optimal guidance, Bezier curves, Trajectory generation, Surrogate modeling*

Nomenclature

Vehicle States and Dynamics *Latin*

a – Acceleration vector
 a_n – Normal (centripetal) acceleration
 g – Gravitational acceleration
 L, D – Lift and drag forces
 m – Vehicle mass
 r – Position vector
 v – Velocity vector

Angles and Coordinates *Greek*

γ – Flight path angle
 χ – Velocity azimuth (heading angle)
 α – Angle of attack
 μ – Bank angle
 λ, φ – Longitude and latitude

Bézier and Geometric Notation *Latin*

B_i^n – i -th Bernstein polynomial of degree n
 N – Number of Bézier segments

P_i – Bézier control point of segment i
 $P_{i,k}$ – k -th control point of the i -th Bézier arc
Greek

$\kappa(\tau)$ – Curvature vector of the reference path
 τ – Normalized curve parameter
 $\theta_i(\tau)$ – Physical time along Bézier segment i

Mission Parameters and Scaling *Latin*

R_E – Mean Earth radius
 t_{ref} – Reference time for velocity scaling
 Ma – Mach number

Indexing and Operators *Superscripts*

ref,0 – Superscripts denoting values on the nominal offline reference trajectory

Subscripts

i, k – Segment and local control-point indices
 $0, f$ – Initial and final conditions

¹DTIS, ONERA, Université Paris Saclay, F-91123 Palaiseau, France, prince.edorh@onera.fr

²DTIS, ONERA, Université Paris Saclay, F-91123 Palaiseau, France, bruno.herisse@onera.fr

1. Introduction

Hypersonic atmospheric re-entry guidance remains a challenging problem due to the strong sensitivity of the vehicle trajectory to initial condition dispersions, model uncertainties, and environmental disturbances [1, 2].

Conventional approaches [3] rely on precomputed optimal reference trajectories generated offline, which are then tracked onboard using feedback control laws to mitigate deviations. While effective for small dispersions, this strategy becomes inadequate when large off-nominal conditions occur: the reference trajectory no longer represents a feasible optimal path, and simply tracking it can lead to violations of mission constraints or significant loss of accuracy.

Two main strategies are available to address this issue:

- **Onboard re-optimization** of the trajectory, which can restore optimality but is often computationally prohibitive in a real-time, resource-constrained environment [4, 5].
- **Adaptation or reconstruction** of the stored reference trajectory based on the current state, which is generally faster but requires a compact, easily adjustable representation [6, 7].

The latter strategy motivates the use of low-dimensional trajectory representations, where the complete trajectory is encoded by a reduced set of parameters. Bézier curves are a particularly attractive option: they can accurately approximate smooth flight paths using only a few control points [8, 9], are well-suited for storage, and allow for efficient interpolation or modification in flight. Moreover, they can be directly integrated into certain guidance laws, such as the BCBG scheme [10], which computes the control commands required to track the curvature of a Bézier-defined reference path.

However, in the presence of large dispersions, even the parameters of this reduced representation must be updated. In this work, we propose an *offline learning approach* that maps initial condition dispersions to the parameters of a compact trajectory representation. A neural network [11] is trained using a set of optimal trajectories generated within a predefined dispersion domain. At runtime, the trained model predicts the parameters corresponding to the current state, enabling the onboard reconstruction of a reference trajectory suitable for immediate use by any path-following algorithm.

While the principle is generic and could be applied to alternative representations and tracking laws (e.g., proportional navigation [12], sliding-mode control [13]), this study focuses on Bézier curves and evaluates the approach within the Bézier Curves-Based Guidance (BCBG) framework [10]. Through Monte Carlo simulations, we assess the ability of our surrogate model to provide accurate, constraint-compliant trajectories over a wide dispersion domain, while significantly reducing onboard computational and storage requirements.

2. Bézier curves and guidance derivation

2.1. Mathematical definition and properties

Bézier curves are polynomial parametric curves defined by a set of control points $P_0, P_1, \dots, P_n$ and Bernstein polynomials [8]. Given a dimensionless curve parameter $\tau \in [0, 1]$ (also referred to as the normalized curvilinear abscissa), an n -th order Bézier curve is expressed as

$$P(\tau) = \sum_{i=0}^n P_i B_i^n(\tau) \quad (1)$$

where $B_i^n(\tau) := \binom{n}{i} \tau^i (1 - \tau)^{n-i}$ is the n -th order Bernstein polynomial.

The initial and final positions are exactly the first and last control points ($P(0) = P_0, P(1) = P_n$). The derivatives at the endpoints are directly related to the vectors $P_1 - P_0$ and $P_n - P_{n-1}$, namely,

$$P'(0) = \frac{dP}{d\tau}(0) = n(P_1 - P_0), \quad P'(1) = \frac{dP}{d\tau}(1) = n(P_n - P_{n-1}), \quad (2)$$

which makes it straightforward to enforce boundary conditions on position, velocity direction, or flight path angle by appropriately setting these points. These properties make Bézier curves particularly appealing for trajectory modelling.

Moreover, Bézier curves provide a compact representation, since smooth trajectories can be defined with only a small number of control points. Intermediate states can be computed analytically from these points, which greatly simplifies interpolation. Furthermore, the *De Casteljau* algorithm [9] allows any portion of a Bézier curve to be expressed as another Bézier curve, enabling straightforward updates of the path during flight without recomputing the entire trajectory.

In aerospace applications, 3rd- or 4th-order curves are often used to represent the reference path in geocentric coordinates (r, λ, ϕ) , where r is the radius, λ longitude, and ϕ latitude [4, 6, 7, 10, 14].

For this application, 3rd-order Bézier curves are used, therefore

$$P(\tau) = (1 - \tau)^3 P_0 + 3(1 - \tau)^2 \tau P_1 + 3(1 - \tau) \tau^2 P_2 + \tau^3 P_3. \quad (3)$$

The derivatives at the endpoints follow directly from this definition: at $\tau = 0$,

$$P'(0) = 3(P_1 - P_0), \quad P''(0) = 6(P_0 + P_2 - 2P_1) \quad (4)$$

while at $\tau = 1$,

$$P'(1) = 3(P_3 - P_2), \quad P''(1) = 6(P_1 + P_3 - 2P_2). \quad (5)$$

2.2. Piece-wise interpolation of a reference trajectory

In many guidance scenarios, a single global polynomial or spline is insufficient to accurately capture long and dynamically complex trajectories. Variations in curvature, speed, and flight conditions are more effectively represented with a Piece-wise Interpolation, where the path is decomposed into N cubic Bézier curves, each defined by its own control points and timing.

This piecewise construction improves local fidelity — for instance during high dynamic pressure phases or terminal corrections — and enables efficient mid-course updates by recomputing only the remaining segments.

Let $x(t) = (r(t), v(t))$, $t \in [t_0, t_f]$ denote a reference trajectory. Each segment $P_i(\tau)$, $\tau \in [0, 1]$, satisfies

$$\begin{cases} P_i(0) = P_{i,0} = r(t_{i-1}), \\ P_i(1) = P_{i,3} = r(t_i), \\ P'_i(0) = 3(P_{i,1} - P_{i,0}) = \rho_{i,1} v(t_{i-1}), \\ P'_i(1) = 3(P_{i,3} - P_{i,2}) = \rho_{i,2} v(t_i), \end{cases} \quad (6)$$

ensuring positional and tangent continuity at the junctions t_k . The free parameters $(\rho_{i,1}, \rho_{i,2})$ are directly related to the curvature at the segment endpoints [8]. The optimization problem therefore consists in selecting the set of curvature parameters $\rho_{i,1}, \rho_{i,2}$ together with the intermediate switching times t_k (with t_0 and t_f fixed), so as to maximize guidance performance [10]. A common criterion is thus to minimize the deviation from the reference:

$$\min \sum_{i=1}^N \int_0^1 \|r(\theta_i(\tau)) - P_i(\tau)\|^2 d\tau, \quad (7)$$

where the so-called *true time* $\theta_i(\tau)$ is defined as the solution of

$$\begin{cases} \frac{d\theta_i}{d\tau} = \frac{\|P'_i(\tau)\|}{\|v(\theta_i(\tau))\|}, \\ \theta_i(0) = t_{i-1}. \end{cases} \quad (8)$$

2.3. From trajectory representation to guidance law

A Bézier curve is, by construction, a purely geometric object and does not inherently encode vehicle dynamics. In a guidance context, the curve is interpreted as the desired reference path in space, parameterized by $\tau(t)$, where τ is the normalized curvilinear abscissa along the currently tracked segment. The guidance task consists of computing control inputs that drive the actual vehicle state toward this path while satisfying operational constraints.

The process involves three main steps:

1. **Kinematic mapping:** From the Bézier formulation, the first and second derivatives of position with respect to the Bézier parameter τ are computed analytically. Using the chain rule, these are converted into time derivatives as

$$\frac{d}{dt} = \dot{\tau} \frac{d}{d\tau}, \quad (9)$$

for first derivatives, and

$$\frac{d^2}{dt^2} = \dot{\tau}^2 \frac{d^2}{d\tau^2} + \ddot{\tau} \frac{d}{d\tau}, \quad (10)$$

for second derivatives. This yields the instantaneous velocity and acceleration vectors along the reference path.

2. **Inverse dynamics:** Using the time derivatives from the kinematic mapping, the commanded accelerations (normal, tangential, or full 3D) are derived so as to reproduce the curvature and progression rate of the reference trajectory. In the case of an unpowered glider, these accelerations correspond to aerodynamic lift and drag components; for a powered vehicle, they include thrust contributions [14].
3. **Control allocation:** The commanded accelerations are converted into physical control variables (e.g., angle-of-attack α and bank angle μ for an unpowered glider, or thrust magnitude and direction for a powered stage) through the relevant aerodynamic or propulsion model, while enforcing actuator saturations, aerodynamic constraints, and other operational limits.

Since τ is not directly related to physical distance or elapsed time, a monotonic mapping $t(\tau)$ is here introduced. For any smooth function f ,

$$f'(\tau) = \dot{f}(t(\tau)) t'(\tau), \quad f''(\tau) = \ddot{f}(t(\tau)) t'(\tau)^2 + \dot{f}(t(\tau)) t''(\tau), \quad (11)$$

ensuring the reparameterization is valid even for complex geometries.

A derivation of the dynamics and associated control laws for a Bézier-based reference path in geocentric coordinates was presented in [10, 14]. In the present work, we adopt a formulation expressed directly in cartesian coordinates, where the total acceleration is defined as

$$a = \frac{L}{m} - \frac{D}{m} - g, \quad (12)$$

where L and D are the lift and drag forces, and g the gravitational acceleration. For an unpowered glider, the tangential acceleration is essentially dictated by drag and gravity and cannot be actively increased. The controllable part of the motion comes from the lift force L , which acts nearly perpendicular to the velocity vector. By adjusting the bank angle μ , the lift vector is rotated within the plane normal to the velocity, thereby shaping the trajectory curvature. It is therefore natural to formulate the guidance problem in terms of a commanded normal acceleration, which directly controls how closely the vehicle tracks the Bézier reference path.

In this framework, the guidance command is derived from the curvature of the reference trajectory $P(\tau)$, which specifies the lift acceleration to be produced. The curvature vector is given by

$$\kappa(\tau) = \frac{(P'(\tau) \wedge P''(\tau)) \wedge P'(\tau)}{|P'(\tau)|^4}, \quad (13)$$

and the required normal acceleration can then be expressed directly as

$$a_n = \frac{\|v\|\|\kappa\|}{\|v \wedge \kappa\|} (\|v\|^2 \kappa - \langle v, \kappa \rangle v). \quad (14)$$

This formulation automatically corrects for potential misalignment between the velocity and curvature vectors. By construction, the commanded normal acceleration satisfies

$$\|a_n\| = \|v\|^2 \|\kappa\|, \quad (15)$$

which reduces to the classical centripetal acceleration in the planar case where $v \perp \kappa$. Therefore, the lift acceleration command reads

$$\frac{L}{m} = a_n + \frac{(v \wedge g) \wedge v}{\|v\|^2}. \quad (16)$$

This lift acceleration L/m is expressed in the vehicle body frame, whose z -axis is aligned with the velocity-normal direction in the vertical plane, and whose y -axis is lateral, perpendicular to both the velocity vector and the z -axis. Projecting L/m onto these axes gives

$$a_y = \frac{1}{m} \langle L, e_y \rangle, \quad a_z = \frac{1}{m} \langle L, e_z \rangle, \quad (17)$$

so that the commanded bank and angle of attack can be computed as

$$\mu = \arctan\left(\frac{a_y}{a_z}\right), \quad \alpha = F^{-1}(|L|, \rho, \text{Ma}), \quad (18)$$

where F^{-1} denotes the inverse aerodynamic lift model, mapping lift magnitude, atmospheric density ρ , and Mach number Ma to the corresponding angle of attack. Finally, the obtained commands are saturated according to actuator authority and operational load-factor limits

Remark. In practice, the BCBG algorithm is executed in a shrinking-horizon fashion, with updates performed at discrete times t_k [10]. At each step, the current Bézier segment being tracked is indexed by i_k , and the active target point along this segment is denoted P_{c_k} . Between updates, commands are evaluated from precomputed look-up tables indexed by time $t(\tau)$.

When a new update occurs, the remaining portion of the trajectory from the vehicle's current state to P_{c_k} is reparameterized using the De Casteljau algorithm [9]. The target point is switched to the next waypoint when the remaining curvilinear distance to it drops below a threshold $L_{\min}$, or fixed to the final waypoint when $i_k = N$. An alternative strategy is to select the active target point based on a fixed look-ahead distance along the reference path, rather than locking it to predefined waypoints. This approach can smooth the guidance response and avoid abrupt target switches, particularly when disturbances or modelling errors cause deviations from the nominal trajectory.

2.4. Guidance limitations under large initial errors

A limitation of guidance strategies based on reference following arises when the initial dispersion in position and heading is large. In such cases, the nominal reference trajectory $P(\tau)$ may become too costly or even impossible to track directly, as the required corrections would exceed the vehicle's maneuvering capabilities. A natural idea is then to modify the reference itself so as to improve its followability.

A simple approach would be to translate the reference trajectory so that it originates at the current initial state. This guarantees immediate consistency at launch but carries the risk of "propagating" the initial dispersions to the final state, thereby degrading terminal accuracy. More elaborate compromise strategies can be envisioned, for example by progressively morphing the translated trajectory back towards the nominal one as a function of the trajectory progress (e.g. percentage of flight completed, or current altitude).

In the present work, rather than relying on ad-hoc transformations, we adopt a data-driven strategy: neural networks are trained to generate a modified reference trajectory directly, conditioned on the dispersed initial state. This enables rapid adaptation while preserving feasibility with respect to terminal conditions and operational constraints.

Recent works have even sought to associate state-action mappings with dispersions along the entire trajectory, enabling purely neural-network-based guidance laws [15]. In contrast, the approach considered here focuses on predicting the parameters of a compact reference trajectory representation, while the online path-following itself can be governed by any suitable guidance law (e.g., BCBG, proportional navigation, or sliding-mode schemes), rather than being embedded in the neural network. Beyond its computational efficiency, this strategy also provides an explicit reference trajectory, which can be readily exploited a posteriori for mission analysis.

3. Neural-network based generation of reference trajectories

3.1. Motivation and principle

In general, trajectory-tracking guidance frameworks allow the reference path to be reconfigured at the beginning of a given flight phase, based on the actual initial state — possibly dispersed from nominal — either through geometric adjustments or by solving an optimization problem.

When additional constraints must be handled — for instance, exclusion zones, desired final state — it becomes necessary to ensure that any trajectory deformation remains consistent with these requirements, or that the optimization procedure is sufficiently robust to deliver a feasible solution within the allotted time. To avoid relying on such a reconfiguration step, the objective here is instead to exploit the framework, which enables the rapid computation of a relevant reference trajectory directly adapted to the dispersed initial state.

The generation process relies on a *surrogate model* trained on a large database of pre-optimized trajectories. Each trajectory in the database results from solving an offline optimal control problem under realistic aerodynamic, structural, and operational constraints. Once trained, the meta-model provides the Bézier parameters directly from the measured initial state

First, because the trajectories used for training are obtained from offline optimization under realistic aerodynamic and operational constraints, the generated paths are feasible by construction and automatically satisfy the required mission constraints.

Second, the method maintains a high degree of robustness to large initial dispersions, ensuring that the final state remains close to the target even when the initial conditions are significantly perturbed.

Finally, since the output is simply a dynamically consistent reference trajectory, it can be followed by a variety of guidance laws — such as BCBG, Proportional Navigation (PN) [12], or alternative path-following schemes [16] — without the need for re-tuning.

3.2. Case study and dataset generation

The proposed method is assessed on a simplified hypersonic vehicle model inspired by a high-lift Common Aero Vehicle (CAV) configuration [17], representative of an unpowered glider during atmospheric entry. In the nominal mission scenario, the vehicle begins its descent from 60 km altitude with an initial velocity of about 3 km s^{-1} (roughly Mach 10 at this altitude). The trajectory subsequently features an aerodynamic pull-up maneuver, followed by a glide phase that continues down to 20 km altitude at approximately Mach 3. In addition, mission objectives impose specific constraints on the final position and velocity azimuth, along with operational limits on load factor and angle-of-attack (see table 1).

To train the surrogate model, the initial state of the vehicle is dispersed around nominal conditions using *Latin Hypercube Sampling* (LHS) with a maximin criterion, ensuring a uniform coverage of the operational envelope [18]. The dispersions apply to the key components of the initial kinematic state as summarized in table 2, and are defined from mission tolerances and uncertainty analyses to ensure physical relevance.

Table 1. Nominal initial/final conditions and mission constraints

Parameter	Symbol	Value	Unit
Initial altitude	h_0	60.0	km
Initial velocity	v_0	10.0	Mach
Initial longitude	λ_0	0.0	deg
Initial latitude	φ_0	0.0	deg
Initial slope	γ_0	-30.0	deg
Initial latitude	χ_0	-75.0	deg
Final altitude	h_f	20.0	km
Final longitude	λ_f	-14.0	deg
Final latitude	φ_f	0.0	deg
Load factor constraint	$\eta_{\max}$	100.0	m/s ²
Final azimuth constraint	χ_f	-90.0	deg
Angle-of-attack constraint	$\alpha_{\max}$	15.0	deg

Table 2. Initial condition dispersion ranges for LHS sampling (relative to nominal)

Variable	Min	Max	Unit
Longitude	-0.5	0.5	deg
Latitude	-0.5	0.5	deg
Velocity magnitude	-100.0	100.0	m/s
Flight path angle	-2	2	deg
Azimuth	-5.0	5.0	deg

For each sampled initial condition, an offline trajectory optimization is performed using a parametric direct method [19]. The guidance laws for $(\alpha(t), \mu(t))$ are defined as piecewise polynomials, with switching times between phases such as pull-up, glide, and terminal approach. The optimization seeks to maximize the final velocity while satisfying:

- the final position and azimuth constraints,
- operational bounds on load factor and α ,
- keep-out zone avoidance.

The trajectory must also avoid a Keep-Out Zone (KOZ) defined in the ground track (longitude-latitude plane) [17]. In this work, the zone is modeled as a vertically unbounded cylinder, with an elliptical cross-section in the longitude-latitude plane. The corresponding parameters (center longitude, latitude, and semi-axes) are listed in table 3.

Table 3. Keep-out zone parameters

Parameter	Symbol	Value	Unit
KOZ center	$(\lambda_{KOZ}, \varphi_{KOZ})$	(-7.0, 0)	deg
KOZ semi-axes	(a, b)	(2.0, 1.0)	deg

For the direct optimization problem, the constraint is enforced by a penalty method: the ellipse is inflated by 5.0% on each semi-axis (a, b) , and any trajectory entering this boundary incurs an added cost, discouraging penetration while ensuring robustness.

The optimized trajectories are expressed in Cartesian coordinates and interpolated by $N = 9$ cubic Bézier curves, as described in section 2.2. For the nominal mission profile, as few as four segments suffice to provide an accurate representation. Nevertheless, a larger number of segments is deliberately employed in all cases to guarantee sufficient interpolation fidelity, particularly for highly perturbed trajectories that may exhibit irregular or complex shapes.

Each segment is defined by 4 control points, yielding a total output vector of size $3 \times 4 \times N = 108$. The control points are obtained by solving the interpolation problem formulated in section 2.2. This nonlinear optimization (eq. (7)) is solved using a Quantum-behaved Particle Swarm Optimization (QPSO) algorithm [20], which efficiently explores the search space of curvature and timing parameters and provides the corresponding set of Bézier points $\{P_{i,k}\}$ that best approximate the optimal trajectory. This vector of control points $\{P_{i,k}\}$ constitutes the output of the learning problem, while the corresponding input is the 6D initial state vector (r_0, v_0) . The final dataset comprises 10000 trajectories spanning the full dispersion envelope. By construction, each trajectory is dynamically feasible and satisfies all mission constraints. This guarantees that the surrogate model, once trained, generalizes reliably within the operational domain while preserving feasibility.

3.3. Learning architecture and training setup

The surrogate model is implemented as a regressor chain [21] with a Multilayer Perceptron (MLP) [22] as its base estimator. In this setup, the prediction of each Bézier control point component depends not only on the input state but also on previously predicted outputs. This sequential dependency allows the model to exploit correlations between control points, which is particularly advantageous for Bézier curves where geometric continuity naturally couples adjacent segments.

Prior to training, both input and output spaces are normalized.

- On the input side, the initial state vector is first scaled to remove unit disparities: positions are non-dimensionalized by the Earth's radius R_E and velocities by a reference time $t_{\text{ref}} = 1000$ s. A subsequent standardization (zero mean, unit variance) is then applied, following common practice in machine learning for physical systems.
- On the output side, the 108 Bézier control point coordinates are similarly standardized and then reduced by Principal Component Analysis (PCA). Twenty-four principal components are retained, capturing 99.029% of the total variance. This dimensionality reduction step both accelerates convergence and mitigates overfitting by eliminating noise and redundancy in the highly correlated outputs [11, 23].

The base estimator has four hidden layers of 16 neurons each, using hyperbolic tangents as activation functions. This choice provides smooth nonlinear mappings bounded in $[-1, 1]$ and helps maintain stable gradients across a wide input range, which makes it well suited for regression tasks involving continuous physical quantities such as trajectory coefficients [22].

The overall learning pipeline is illustrated in Figure 1. It not only summarizes the preprocessing and dimensionality reduction steps (normalization and PCA), but also highlights surrogate model is trained and subsequently used for trajectory reconstruction.

Training is carried out with the Adaptive Moment Estimation (Adam) optimizer [24], which adaptively combines gradient descent with momentum and per-parameter learning-rate scaling. This often yields faster and more stable convergence than vanilla stochastic gradient descent, particularly on noisy or ill-conditioned problems such as high-dimensional trajectory regression.

A benchmark with alternative regressors — including polynomial ridge regression, partial least squares (PLS), extra trees, gradient boosting, and stacking — showed that while tree-based methods performed competitively on lower-dimensional subsets, the MLP regressor chain consistently achieved the best trajectory-level accuracy. This strength can be attributed to its ability to capture nonlinear

interactions and output correlations in structured-output regression tasks [25]. Furthermore, PCA was found to accelerate convergence and reduce variance across folds, confirming its benefit for this application [26].

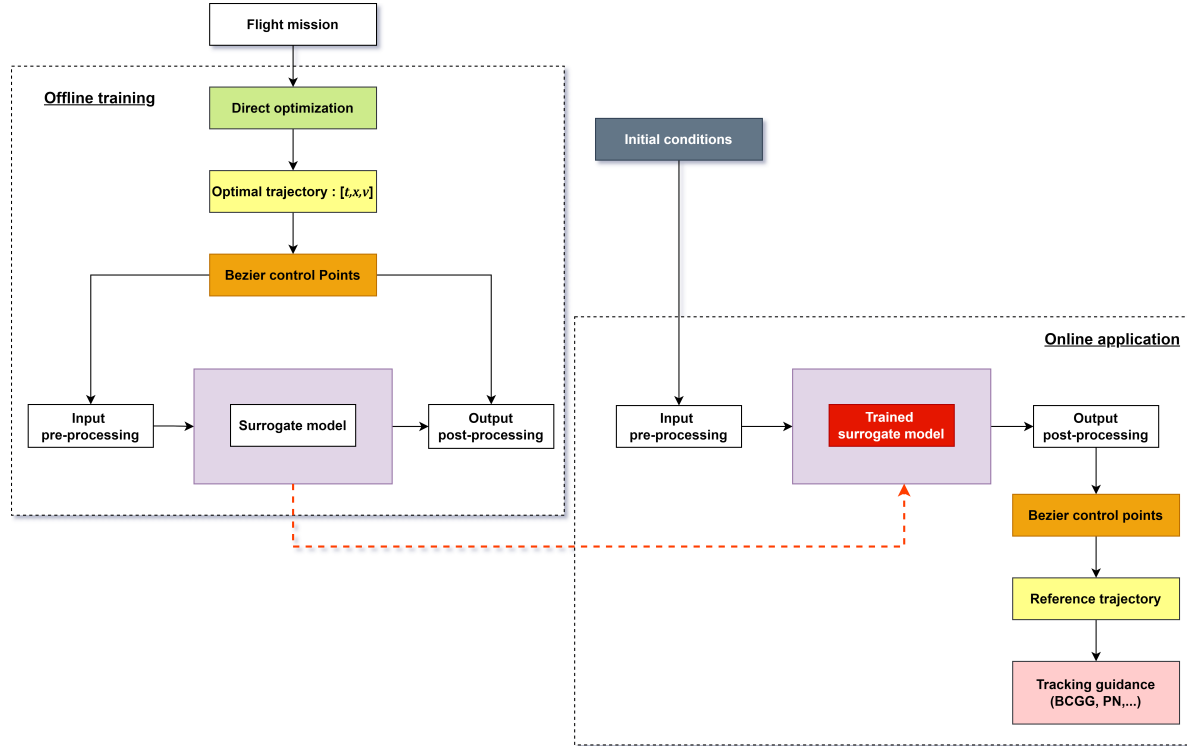


Fig 1. Learning pipeline with preprocessing, regressor chain, and MLP base estimator.

4. Results and discussion

4.1. Analysis of Network Prediction Results

The performance of the surrogate model was first examined using classical regression metrics. Over the validation folds, the mean coefficient of determination is $R^2 \approx 0.75$, where R^2 quantifies the proportion of variance in the target data explained by the model. Although such a score may appear modest when judged against conventional machine learning standards, it does not directly reflect the quality of the reconstructed trajectories. The regression targets the full set of 108 Bézier control point coordinates, and even small discrepancies in intermediate control points — which strongly affect R^2 — often have negligible consequences on the resulting geometric path. For this reason, the mean absolute error (MAE) was adopted as the training loss, since it offers a more faithful measure of predictive accuracy for this type of highly correlated, structured outputs.

A more relevant evaluation is obtained by focusing on the geometric fidelity of the reconstructed paths rather than on raw regression scores. For each prediction, the surrogate-generated control points were re-interpolated into smooth Bézier polylines using 100 samples per segment, and these were compared with the corresponding reference trajectories from the optimized dataset. Here, a polyline denotes the piecewise curve obtained by connecting successive sampled points, which provides a continuous geometric representation of the predicted trajectory. This procedure allows one to assess how closely the surrogate reproduces the actual *shapes* of trajectories in the testing set. Three complementary metrics were employed:

- Root-mean-square error (RMSE) measures the average Euclidean distance between predicted and reference polylines after reparametrization by arc length. For the present study, typical

values remain below 4 km over the full entry profile, which is relatively small. compared with the overall trajectory scale [27].

- Hausdorff distance [28] measures the maximum discrepancy between two curves, i.e. the largest point-to-curve deviation. This reflects a worst-case separation and reaches about 7 km in the most unfavorable instances.
- Chamfer distance [29] computes, in both directions, the average distance from each point in one set to its nearest neighbor in the other. This yields a symmetric and balanced indicator of global similarity. Reported values are close to 3 km, consistent with the RMSE.

It is worth stressing that these comparisons are purely geometric: they do not involve dynamic simulation or guidance tracking, but only assess how faithfully the surrogate reproduces the optimized reference shapes. Despite the relatively modest R^2 , the agreement at the trajectory level is strong, with deviations remaining well within margins acceptable for guidance applications.

A closer inspection of the control-point predictions reveals a heterogeneous accuracy. The segment endpoints, $P_{i,0}$ and $P_{i,3}$, which ensure continuity between successive Bézier curves, are generally predicted with high reliability. By contrast, the intermediate points $P_{i,1}$ and $P_{i,2}$ — which govern local curvature and indirectly influence speed distribution — are more challenging to approximate. The largest errors occur near phases of high curvature, such as the aerodynamic pull-up or sharp lateral turns, where trajectory variability is greatest and interpolation of synthetic trajectories is less robust [8].

This observation highlights the intrinsic difficulty of predicting local geometric features while still ensuring overall path consistency. It is worth noting that many of the discrepancies responsible for the reduced R^2 score are associated with *degenerate cases*. In nearly straight trajectory segments, the internal Bézier nodes $P_{i,1}$ and $P_{i,2}$ become weakly sensitive: small prediction errors on these control points produce negligible changes in the actual curve. As a result, coordinate-level regression metrics penalize the model more severely than trajectory-level geometric measures.

Beyond the aggregate metrics discussed above, it is instructive to visualize the behavior of the surrogate at the level of individual trajectories. Figure 2 illustrates the predicted versus optimized Bézier control points for two randomly selected validation cases, while Figure 3 shows the corresponding reconstructed polylines. These comparisons highlight that, although small deviations appear on the intermediate control points, the overall polygonal structure is well preserved, leading to geometric paths that remain very close to the optimized references.

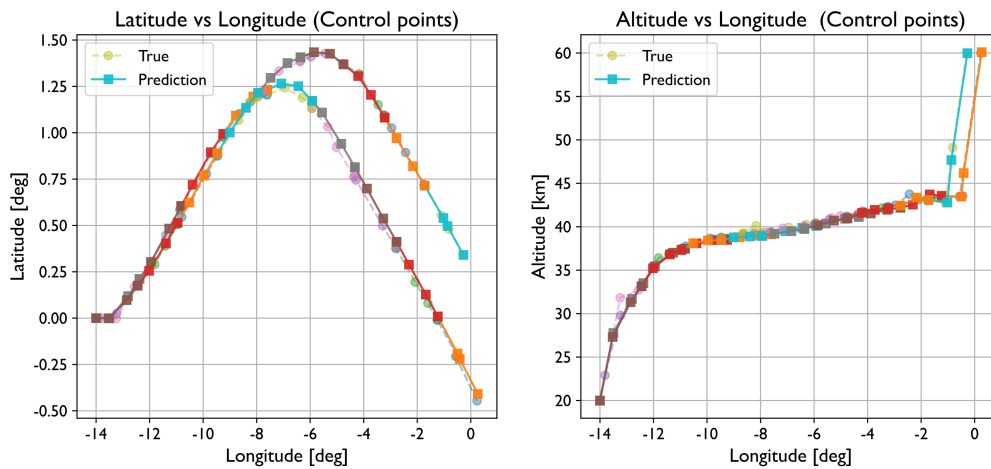


Fig 2. Comparison of predicted versus optimized Bézier control points for two validation trajectories.

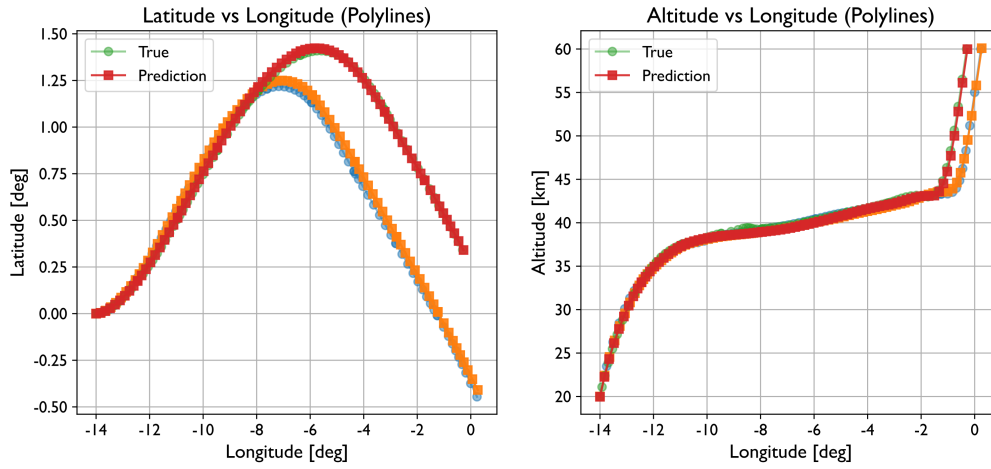


Fig 3. Reconstructed Bézier polylines for the same validation cases.

4.2. Trajectory Simulation and Guidance

To assess the practical usefulness of the surrogate model, trajectory-following simulations were carried out using the BCBG algorithm. Although other schemes such as proportional navigation could also be employed, BCBG was selected here as a representative closed-loop law, since the objective is not to test a specific controller but to evaluate the quality of the surrogate-generated references.

The simulations compare two initialization strategies:

- **offline optimized reference** where a trajectory optimized under nominal conditions is used directly as the reference for all dispersed cases.
- **Surrogate generation** where a new reference trajectory is produced by the surrogate model, consistent with the dispersed initial conditions.

In both cases, BCBG algorithm is applied to track the supplied reference.

For robustness, dispersions were introduced not only in the initial position and velocity, sampled from Gaussian distributions whose 3σ intervals match the ranges defined in the LHS dataset (see table 2), but also in the aerodynamic coefficients and atmospheric density profiles. Specifically, the vehicle drag and lift coefficients, together with the atmospheric density, were perturbed according to Gaussian laws with a standard deviation of $\pm 10\%$ around their nominal values, in order to emulate realistic uncertainties in the entry conditions.

A campaign of 1000 Monte Carlo simulations was conducted, both with and without activation of the predictive model. The outcomes are illustrated in Figures 4 and 5. In the ground track (longitude-latitude, fig. 4b), surrogate-generated trajectories remain closely aligned with the optimized reference path and consistently avoid the keep-out zone, even under large dispersions. In contrast, when relying on the single offline reference, the guidance law drives the vehicle back towards the optimized path too aggressively. This results in strong control actions during the early atmospheric phase, excessive energy dissipation, and ultimately a reduced energy budget for the subsequent glide. The degraded state may shorten the glide phase and, in extreme cases, cause premature descent.

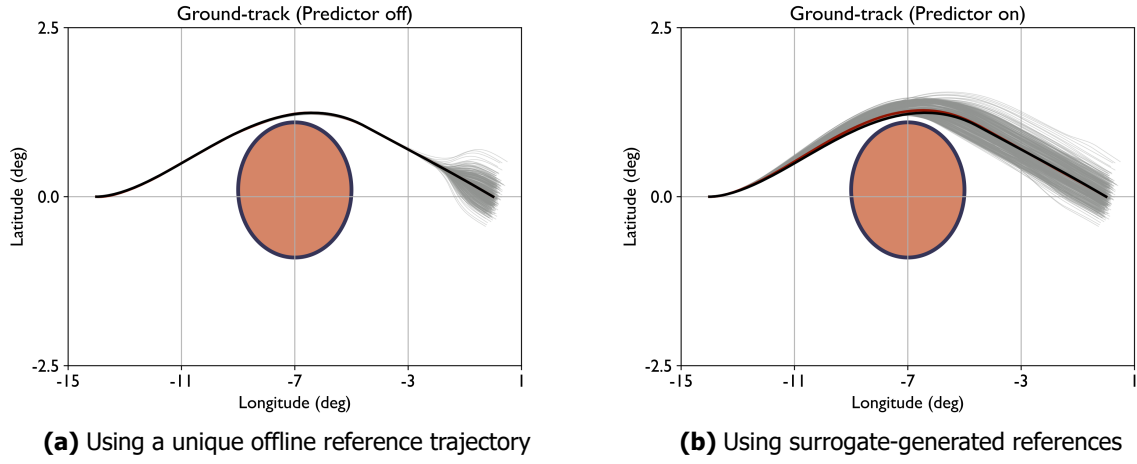


Fig 4. Ground track comparison under dispersed conditions. *Left: trajectories obtained by tracking a unique optimized reference trajectory. Right: trajectories obtained from surrogate-predicted references. Thin grey lines correspond to dispersed runs, while the thick solid red line denotes the trajectory under nominal conditions (identical in both cases, with or without surrogate initialization).*

In the altitude profile (see fig. 5b), the surrogate-based strategy naturally guides the vehicle towards the optimized solution throughout the trajectory. The characteristic pull-up and glide structure is preserved, while convergence to the final position is ensured. In both plots, the thick solid red line represents the trajectory under nominal conditions (with or without surrogate initialization), and thin grey lines correspond to dispersed runs. Overall, the surrogate model maintains feasibility and proximity to the desired reference, whereas reliance on the nominal path alone leads to energy losses and may compromise the glide phase.

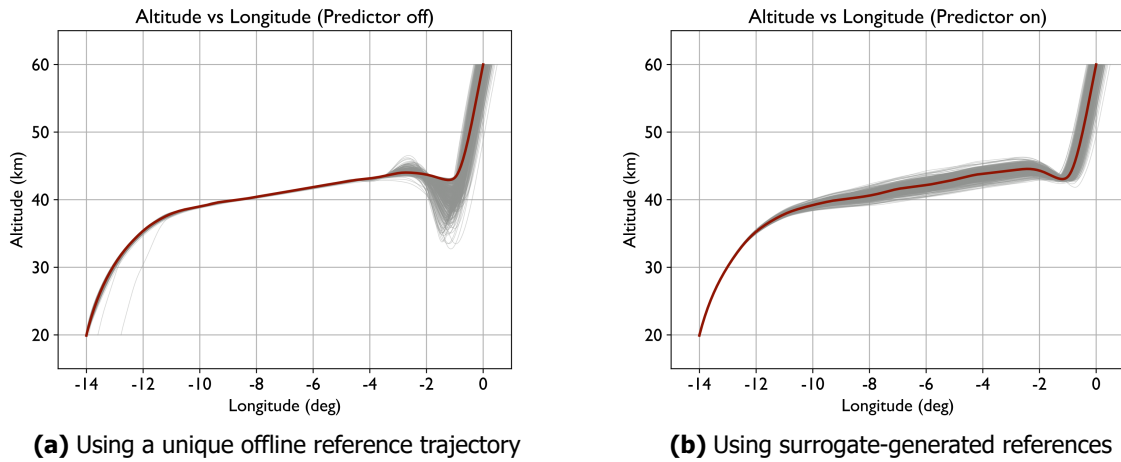


Fig 5. Altitude profile comparison under dispersed conditions. *Left: trajectories obtained by tracking a unique optimized reference trajectory. Right: trajectories obtained from surrogate-predicted references. Thin grey lines correspond to dispersed runs, while the thick solid red line denotes the trajectory under nominal conditions (identical in both cases, with or without surrogate initialization).*

It is worth noting that simply shifting the offline reference to match the dispersed initial conditions would

have avoided the premature loss of altitude observed with the offline reference strategy. However, such rigid translations provide no guarantee of constraint satisfaction, since the shifted path may intersect restricted zones. Designing meaningful deformations would therefore be necessary, explicitly accounting for environmental and operational constraints.

To complement this geometric analysis of simulated trajectories, it is equally important to assess the guidance effort. Figures 6 and 7 display the corresponding angle-of-attack and bank-angle histories, providing additional insight into the control activity required to track the trajectories.

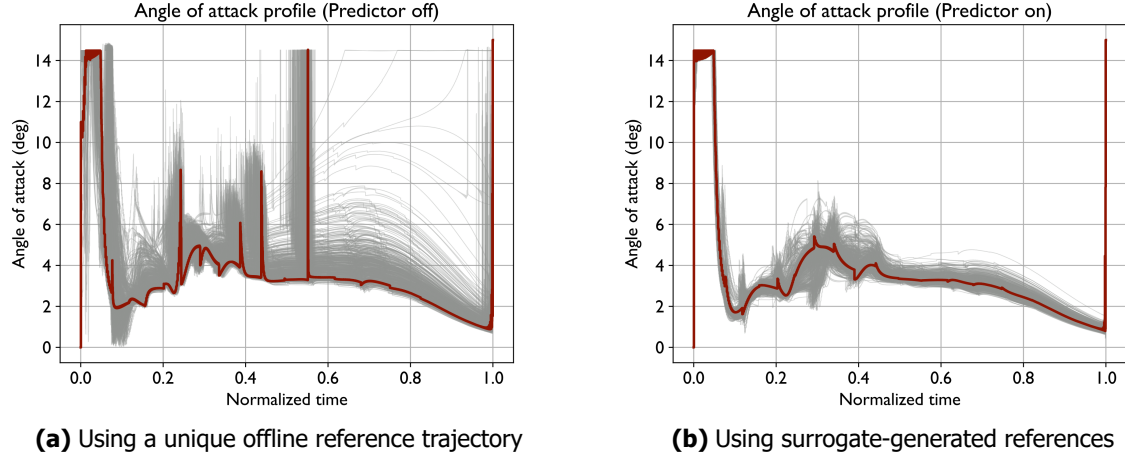


Fig 6. Angle of attack profiles. *Thin grey lines correspond to dispersed runs, while the thick solid red line denotes the trajectory under nominal conditions.*

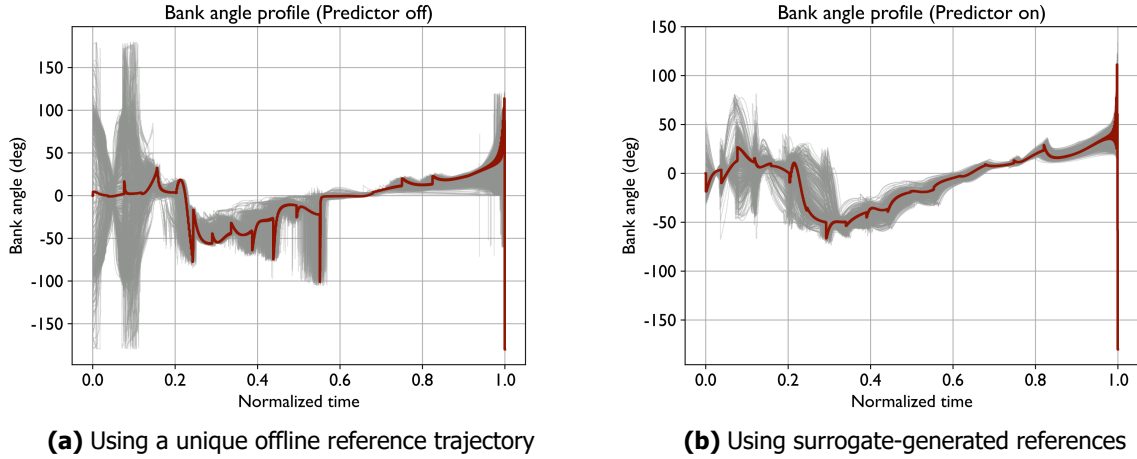


Fig 7. Bank angle profiles. *Thin grey lines correspond to dispersed runs, while the thick solid red line denotes the trajectory under nominal conditions.*

With the offline reference strategy, the vehicle attempts to rejoin the optimized trajectory as early as possible. This induces strong control actions in both bank angle and angle of attack, leading to excessive energy dissipation during the initial pull-up. As a result, the subsequent glide phase is entered with reduced velocity, forcing the vehicle to fly at a higher incidence to sustain lift. This degraded energy state shortens the glide and may even result in a premature descent.

By contrast, surrogate-generated references provide a trajectory already adapted to the dispersed initial state. The corrective maneuver remains modest and coherent, preserving both energy and the nominal pull-up glide structure. The guidance remains smooth and dynamically consistent, without the overcompensation and degraded glide characteristic of the offline reference strategy.

The violin plots in Figure 8 provide a statistical view of how initial dispersions are propagated to the terminal state under the two strategies.

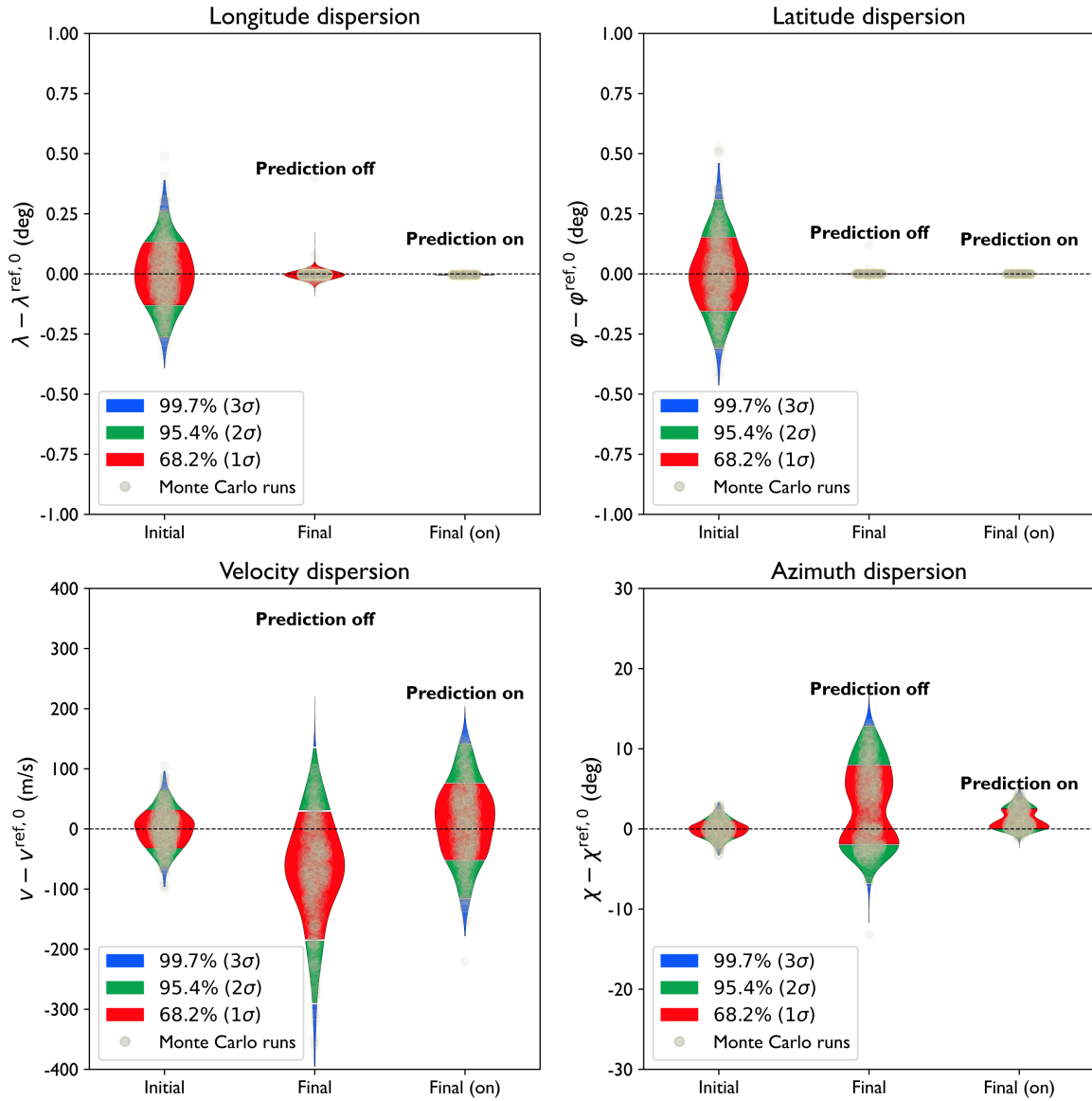


Fig 8. Violin plots of initial and final state dispersions (longitude, latitude, velocity, azimuth). Each violin shows the probability distribution of the corresponding variable, with shaded bands indicating the 1σ, 2σ, and 3σ confidence intervals. The left set represents the initial dispersions, the central set corresponds to final states obtained with a nominal offline reference trajectory, and the right set to surrogate-based reference generation. Superscripts ref,0 denote normalization with respect to the nominal offline reference.

Each violin represents the probability distribution of trajectory dispersions for the initial conditions (left), the final solutions obtained with a nominal offline reference (center), and the final solutions obtained with the surrogate-based reference (right). Shaded bands correspond to the 1σ (68.3%), 2σ (95.4%), and 3σ (99.7%) confidence intervals, thus highlighting both the concentration of trajectories around the mean and the presence of outliers. All metrics are normalized with respect to the nominal offline reference, emphasizing the systematic reduction in terminal dispersion achieved through surrogate-based initialization.

Both the surrogate model and the offline reference approaches reduce final dispersion in longitude and latitude. The surrogate model, however, consistently yields a tighter clustering of trajectories around the target. In the offline reference case, a few outliers persist, corresponding to trajectories that dissipate excessive energy during the early atmospheric phase and terminate prematurely. By adapting the reference to the dispersed initial state, the surrogate strategy mitigates this behavior and generates a coherent bundle of trajectories that progressively converge toward the target.

For velocity, the surrogate approach achieves a lower final standard deviation (approximately 60 m/s versus 100 m/s while using an nominal reference), indicating improved preservation of the entry energy budget despite the corrective maneuvers required for obstacle avoidance and terminal constraint satisfaction. In addition, the mean final velocity with the surrogate remains close to that of the offline optimized solution, providing a clear performance advantage.

Regarding azimuth, both strategies reduce dispersion, but the surrogate model again produces a substantially tighter distribution. Although terminal dispersion is slightly larger than the initial spread — mainly due to guidance-law saturation in the final phase — the time history (see fig. 9) shows that surrogate-based trajectories progressively converge toward the desired heading, with dispersion steadily decreasing as the flight evolves.

Overall, surrogate-based initialization offers a robust correction mechanism that ensures compliance with mission constraints (e.g., keep-out zone avoidance, azimuth accuracy) while maintaining terminal dispersions within acceptable tolerances. In contrast, reliance on a fixed nominal reference, although computationally simpler, fails to compensate for inherited initial errors and thus lacks robustness under large dispersions. These results demonstrate the advantages of surrogate-based reference adaptation in enhancing robustness and dynamic consistency, making it a promising solution for precision-demanding guidance applications.

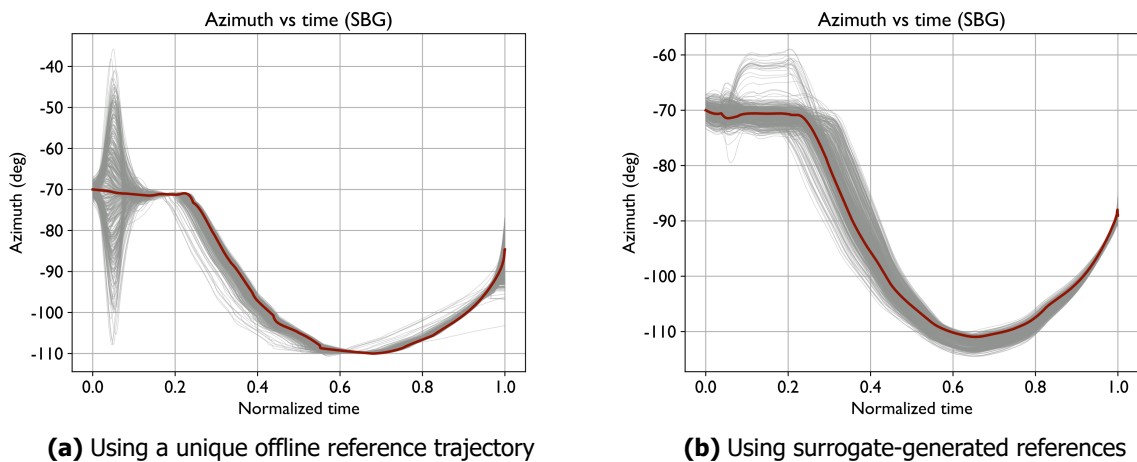


Fig 9. Azimuth evolution under BCBG guidance. *Thin grey lines correspond to dispersed runs, while the thick solid red line denotes the trajectory under nominal conditions.*

5. Conclusion

This work investigated a compact representation of hypersonic entry trajectories based on cubic Bézier curves, enabling efficient storage, interpolation, and online adaptation. A predictive surrogate model was developed to map initial condition dispersions directly to the control-point parameters of this reduced representation. By training on a large dataset of dynamically feasible trajectories, the surrogate provides reference paths that are both accurate and compliant with mission constraints.

The proposed approach is naturally coupled with a trajectory-following guidance law, which exploits the Bézier representation to compute commands (e.g., angle of attack and bank angle) ensuring adherence to the reconstructed path. In this context, the surrogate does not replace the guidance algorithm itself, but rather serves as an initializer, providing feasible and well-shaped references that can be tracked in real time. Monte Carlo campaigns demonstrated that this combination preserves terminal accuracy, geometric feasibility, and constraint satisfaction more reliably than rigid-body deformations of a nominal trajectory.

Beyond the Bézier framework, the same learning-based methodology could be extended to indirect optimal control formulations. In such cases, trajectories are parameterized by selected states and costates at shooting nodes [2, 5]. A predictive model trained to map initial dispersions to these quantities could enable rapid reconstruction of reference solutions while retaining the accuracy and optimality properties of indirect methods. This perspective opens the way to hybrid strategies combining analytical optimal control with data-driven surrogates for onboard implementation.

References

- [1] Ashok Joshi, K Sivan, and S Savithri Amma. "Predictor-corrector reentry guidance algorithm with path constraints for atmospheric entry vehicles". In: *Journal of Guidance, Control, and Dynamics* 30.5 (2007), pp. 1307–1318.
- [2] Prince Edorh, Bruno Hérisse, and Eric Bourgeois. "Glide back recovery of a winged reusable launch vehicle with wind estimate". In: *9th European Conference for Aeronautics and Space Sciences (EUCASS)*. Lille, France, 2022.
- [3] Ping Lu, Stephen Forbes, and Morgan Baldwin. "Gliding guidance of high L/D hypersonic vehicles". In: *AIAA Guidance, Navigation, and Control (GNC) Conference*. AIAA, 2013, pp. 4648–4660. DOI: 10.2514/6.2013-4648.
- [4] Suwon Lee and Youdan Kim. "Optimal output trajectory shaping using Bézier curves". In: *Journal of Guidance, Control, and Dynamics* 44.5 (2021), pp. 1027–1035. DOI: 10.2514/1.G005265.
- [5] Arthur Earl Bryson. *Applied optimal control: optimization, estimation and control*. Routledge, 2018.
- [6] Tawfiqur Rahman, Zhou Hao, and Wanchun Chen. "Bézier approximation based inverse dynamic guidance for entry glide trajectory". In: *2013 9th Asian Control Conference (ASCC)*. IEEE, 2013, pp. 1–6. DOI: 10.1109/ASCC.2013.6606269.
- [7] Usman Fareed, Ke Zhang, and Mir Soban Ahmed. "Trajectory optimization of common aero vehicle through genetic algorithm aided with rational Bézier curve". In: *7th European Conference for Aeronautics and Aerospace Sciences (EUCASS)*. Milan, Italy, 2017, pp. 1–15.
- [8] Gerald Farin. *Curves and surfaces for CAGD: a practical guide*. Morgan Kaufmann / Elsevier, 2001. ISBN: 9781558607378.
- [9] George M Phillips. "A de Casteljau algorithm for generalized Bernstein polynomials". In: *BIT Numerical Mathematics* 37.1 (1997), pp. 232–236.
- [10] Philippe Vernis and Jean-Baptiste Carroy. "Hypersonic Entry Guidance and Bézier Curves, EUCASS2019". In: *8th European Conference for Aeronautics and Space Sciences (EUCASS)*. Madrid, Spain, 2019.
- [11] Alexander Forrester, Andras Sobester, and Andy Keane. *Engineering design via surrogate modelling: a practical guide*. John Wiley & Sons, 2008.
- [12] Paul Zarchan. "Proportional navigation and weaving targets". In: *Journal of Guidance, Control, and Dynamics* 18.5 (1995), pp. 969–974.
- [13] Xiang-yu Zhang et al. "Hypersonic sliding target tracking in near space". In: *Defence technology* 11.4 (2015), pp. 370–381. DOI: 10.1016/j.dt.2015.08.003.

- [14] P Vernis and A Passeron. "Landing a Propulsive Stage with Bézier Curves". In: *9th European Conference for Aeronautics and Space Sciences (EUCASS)*. Lille, France, 2022.
- [15] Haochen Li et al. "Fast trajectory generation with a deep neural network for hypersonic entry flight". In: *Aerospace* 10.11 (2023), p. 931.
- [16] Ashwini Ratnoo et al. "Path following using trajectory shaping guidance". In: *Journal of Guidance, Control, and Dynamics* 38.1 (2015), pp. 106–116.
- [17] Timothy R Jorris. *Common aero vehicle autonomous reentry trajectory optimization satisfying waypoint and no-fly zone constraints*. Air Force Institute of Technology, 2007.
- [18] Michael D Shields and Jiaxin Zhang. "The generalization of Latin hypercube sampling". In: *Reliability Engineering & System Safety* 148 (2016), pp. 96–108.
- [19] John T Betts. *Practical methods for optimal control and estimation using nonlinear programming*. SIAM, 2010.
- [20] Jun Sun, Choi-Hong Lai, and Xiao-Jun Wu. *Particle swarm optimisation: classical and quantum perspectives*. Crc Press, 2011.
- [21] Gabriella Melki et al. "Multi-target support vector regression via correlation regressor chains". In: *Information Sciences* 415 (2017), pp. 53–69.
- [22] Kurt Hornik, Maxwell Stinchcombe, and Halbert White. "Multilayer feedforward networks are universal approximators". In: *Neural networks* 2.5 (1989), pp. 359–366.
- [23] G Gary Wang and Songqing Shan. "Review of metamodeling techniques in support of engineering design optimization". In: *International design engineering technical conferences and computers and information in engineering conference*. Vol. 4255. 2006, pp. 415–426.
- [24] Diederik P Kingma and Jimmy Ba. "Adam: A method for stochastic optimization". In: *arXiv preprint arXiv:1412.6980* (2014).
- [25] Eleftherios Spyromitros-Xioufis et al. "Multi-target regression via input space expansion: treating targets as inputs". In: *Machine Learning* 104.1 (2016), pp. 55–98.
- [26] Ian T Jolliffe and Jorge Cadima. "Principal component analysis: a review and recent developments". In: *Philosophical transactions of the royal society A: Mathematical, Physical and Engineering Sciences* 374.2065 (2016), p. 20150202.
- [27] Danlei Hu et al. "Spatio-temporal trajectory similarity measures: A comprehensive survey and quantitative study". In: *IEEE Transactions on Knowledge and Data Engineering* 36.5 (2023), pp. 2191–2212.
- [28] Daniel P Huttenlocher, Gregory A. Klanderman, and William J Rucklidge. "Comparing images using the Hausdorff distance". In: *IEEE Transactions on pattern analysis and machine intelligence* 15.9 (2002), pp. 850–863.
- [29] Haoqiang Fan, Hao Su, and Leonidas J Guibas. "A point set generation network for 3d object reconstruction from a single image". In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2017, pp. 605–613.