DOI: 10.13009/EUCASS2025-597
11TH EUROPEAN CONFERENCE FOR AERONAUTICS AND SPACE SCIENCES (EUCASS)

MDO COUPLED WITH RANDOM FOREST SENSITIVITY ANALYSIS ON AN INTERCEPTOR MISSILE MISSION

Vassilios Silaidis†, Stefania Carlotti*, Elbano De Nuccio+ and Filippo Maggi**
* *Politecnico di Milano*
+*MBDA Italia*
vassilios.silaidis@polimi.it – stefania.carlotti@polimi.it – elbano.de-nuccio@mbda.it – Filippo.maggi@polimi.it
† Corresponding Author

Abstract
This work presents a novel framework integrating evolutionary algorithms and machine learning regression to support early-phase missile design. The approach employs a genetic algorithm for multidisciplinary optimization across disciplines like aerodynamics, propulsion, and trajectory: as the optimization converges, a database with the evaluated designs is build and a Random Forest regression is applied to the resulting dataset to identify critical design variables, affecting mission performance. This method provides both optimization capabilities and variable sensitivity insights, enabling design space reduction without compromising solution quality. The results show a 2-second reduction in intercept time compared to the baseline design and identify thrust, burning time, diameter, and tail surface span as key performance drivers. The proposed framework enhances decision-making in early missile design phases by integrating optimization and variable importance analysis.

1. Introduction

In missile system design, early development stages are critical, as decisions made in those stages significantly influence performance and cost. Multidisciplinary Design Optimization (MDO) methods are especially suited to this phase, as they allow integrated trade-off analysis across aerodynamics, propulsion, guidance, and other domains [1]. Although MDO has been applied in missile design with promising results [2][3][4], most studies focus on finding optimal solutions without exploring variable interactions or enabling model simplification through sensitivity analysis [5]. This study applies a framework [6] that combines evolutionary optimization with sensitivity analysis in a unified loop: the genetic algorithm stores a solution archive during optimization, which is then analyzed using Random Forest regression to uncover relationships between design inputs and performance metrics. This approach supports robust optimization and reveals key variable dependencies governing system behavior. The article is structured as follows. Section 2 introduces the fundamental concepts of genetic algorithms and random forests, followed by a description of the proposed optimization framework. Section 3 presents the simplified models employed in this study, including the three-degree-of-freedom simulator, providing an overview of its architecture and its role in performance evaluation. Section 4 outlines the initial design parameters defining the optimization problem. Finally, Section 5 discusses the results, offering a critical interpretation of the findings.

Genetic Algorithms for MDO: GAs are a class of optimization techniques inspired by the principles of natural selection and evolutionary biology. These algorithms aim to identify optimal solutions to complex design problems by iteratively evolving a population of potential candidates - referred to as chromosomes - each encoding a set of design variables. Through successive generations, candidate solutions are evaluated using a fitness function that quantifies their performance relative to the design objectives. Selection, crossover, and mutation operators are applied to progressively refine the population, guiding the search toward improved solutions. The decision variables, or genes, often exhibit non-linear interactions, which makes GAs particularly suitable for solving multi-variable, non-convex problems. Comprehensive descriptions of the algorithm's structure and implementation are widely available in the literature [7][8]. Genetic algorithms have demonstrated robust performance across a broad range of aerospace engineering applications, including trajectory optimization [9][10], propulsion system design [11][12], missile configuration [13], launch vehicle development [14], and more. Their ability to handle multidisciplinary problems stems from their stochastic nature, which reduces the risk of premature convergence to local optima. This characteristic

DOI: 10.13009/EUCASS2025-597

is especially valuable when dealing with trade-offs between conflicting objectives, as is common in multidisciplinary design environments [15] [16].

Random Forest: is a supervised machine learning algorithm used for both regression and classification tasks, known for its robustness and accuracy in handling complex, nonlinear datasets. It operates by constructing an ensemble of decision trees during the training phase [17]. Each tree is built from a random subset of the original dataset, both in terms of samples and input features - a process that introduces diversity among the trees and improves generalization. For regression problems, the final output of a random forest is obtained by averaging the predictions of all individual trees, while in classification tasks, the output corresponds to the majority vote across the trees. This ensemble approach reduces variance compared to a single decision tree and makes the model inherently resistant to overfitting, particularly when applied to noisy or high-dimensional data.
Random forests are particularly effective when dealing with large numbers of input features and are less sensitive to multicollinearity or non-linear relationships. Unlike linear models, they do not require explicit assumptions about the form of data dependencies, and they can perform well even in cases with a relatively small number of observations [18]. Moreover, the method scales efficiently with dataset size and requires minimal hyperparameter tuning, which has contributed to its widespread adoption as a versatile tool for predictive modeling [19].
Decision trees that compose forest are hierarchical models composed of internal (decision) nodes and terminal (leaf) nodes, interconnected by branches that represent decision paths. The process of forming a tree begins at the *root node*, which poses a question derived from one of the dataset's input features. Based on the outcome, the data is split into subsets, and the tree continues branching recursively by applying additional decision rules at each level, thereby segmenting the data space based on feature values. A widely used method for constructing decision trees is the CART (Classification and Regression Trees) algorithm. It recursively evaluates all possible splits across all predictors to identify the division that yields the best binary partition, based on a *splitting criterion* that quantifies node impurity. In regression trees, node impurity is typically measured using the Mean Squared Error (MSE), defined as:

$$MSE = \frac{1}{n}\sum_{i=1}^{n}(y_i - \bar{y})^2 \text{ where } \bar{y} = \frac{1}{n}\sum_{i=1}^{n} y_i$$

Here, y_i are the response values at the node, and $\bar{y}$ is their mean. The algorithm selects the predictor and split point that produce the lowest weighted sum of impurity across the resulting child nodes. The tree continues to grow until a predefined stopping criterion is met, such as a minimum number of observations in a node or a maximum tree depth.
Random Forest builds upon this foundation by combining the predictions of multiple decision trees in an ensemble, using a technique known as bootstrap aggregating, or *bagging*. Given a training dataset $\{(x_1, y_1), \dots, (x_n, y_n)\}$, the algorithm generates *B* bootstrap samples -i.e., datasets of size *n* sampled with replacement from the original data—and trains a separate decision tree $\hat{T}_b$ on each sample. The procedure is as follows:

- For each $b = 1, \dots, B$: 1) Draw a bootstrap sample (X_b, Y_b) from the training set; 2) Train a decision tree $\hat{T}_b$ on X_b, Y_b

Once all trees are trained, their predictions are aggregated. For classification tasks, the final prediction $\hat{f}(x)$is the majority vote among all trees:

$$\hat{f}(x) = argmax\, y \sum_{b=1}^{B} I\left(y = \hat{T}_b(x)\right)$$

For regression tasks, the ensemble prediction is the average of the individual tree outputs:

$$\hat{f}(x) = \frac{1}{B}\sum_{b=1}^{B} \hat{T}_b(x)$$

This ensemble approach reduces the variance of the model without increasing bias, as the averaging process mitigates the high sensitivity of individual trees to noise in the training data. Simply training multiple trees on the same dataset would result in highly correlated models, reducing the benefit of aggregation. However, bootstrap sampling introduces variability in the training data seen by each tree, thus promoting decorrelation and enhancing generalization performance. Figure 1 illustrates the bootstrap process used in Random Forests. Figure 2 outlines the three-step framework: 1) optimization, 2) data handling, and 3) post-processing with machine learning regression.

DOI: 10.13009/EUCASS2025-597

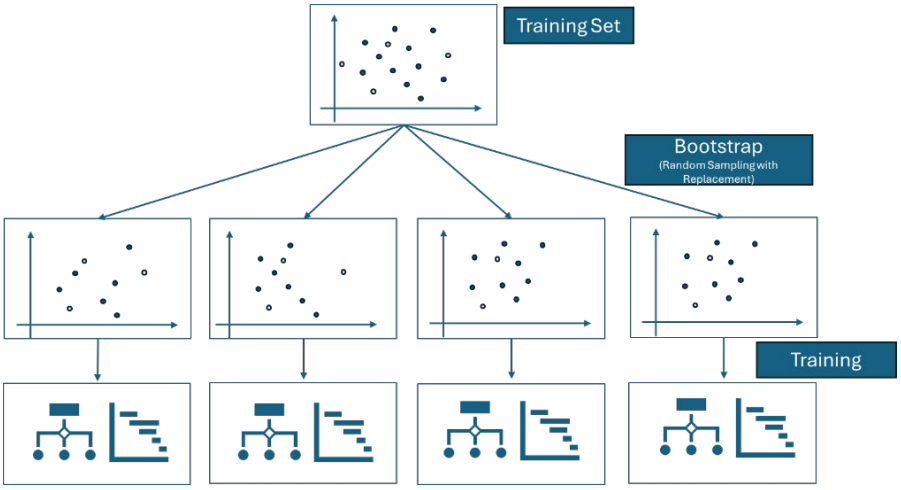
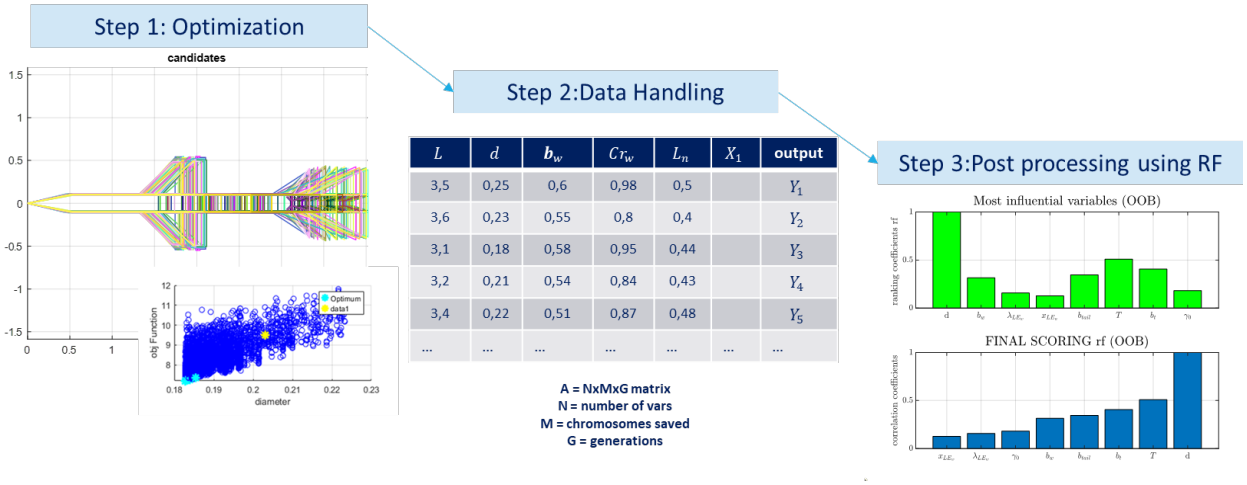


Figure 1: Bootstrap procedure of Random Forest: given an initial set of data, N trees are built (using CART algorithm), randomly sampling points from the dataset and making predictions. The random forest prediction is the average of the predictions of all the trees.



L	d	b_w	Cr_w	L_n	X_1	output
3,5	0,25	0,6	0,98	0,5		Y_1
3,6	0,23	0,55	0,8	0,4		Y_2
3,1	0,18	0,58	0,95	0,44		Y_3
3,2	0,21	0,54	0,84	0,43		Y_4
3,4	0,22	0,51	0,87	0,48		Y_5
...	...	...	...	...	...	...

Figure 2: The proposed framework can be seen as three step process: 1) the optimization, 2) data saving and handling, 3) post processing and machine learning regression

2. Simulation architecture and modelling

The simulation model, implemented in Simulink, was made for intercept mission optimization using Proportional Navigation (PN) guidance. A MDO framework is applied to the AIM-7 Sparrow III as a baseline: once the preliminary design models were validated against AIM-7 characteristics, the GA is used to optimize an intercept mission against a non-manoeuvring target.

Design Process: The MDO process begins by defining the missile's external geometric parameters, selected within a feasible range derived from the baseline study. Given a set of geometric inputs, the design of the propulsion system is performed, constrained by the required total impulse and the maximum allowable body length. Once the propulsion system is dimensioned in terms of geometry, mass, and performance, the remaining subsystems - payload and guidance, navigation, and control (GNC) - are pre-dimensioned by allocating internal volume to each. The mass of the subsystems is estimated using Mass Estimation Relationships (MERs) sourced from literature, based on the allocated volumes. For the aerodynamic characterization, once the body geometry is fixed, a database is generated as a function of Mach number, angle of attack, altitude, and mass. This database provides estimates of the aerodynamic coefficients C_L and C_D, and evaluates maneuverability across the mission. With the missile geometry and subsystem characteristics defined and given the initial mission conditions (initial position x_0, velocity v_0, and launch angle γ_0, an intercept simulation is carried out, and performance data is collected. The MDO optimization is performed using MATLAB's Optimization

DOI: 10.13009/EUCASS2025-597

Toolbox. Finally, a post-processing analysis of the optimization archive is conducted to extract correlation relationships between design variables and performance metrics.

Assumption and modelling notes:

The simulation framework is structured into four primary blocks: Missile, GNC, Engine, and Target Kinematics, each responsible for a distinct portion of the missile guidance and control loop. The integration of these components ensures a coherent flow of information from guidance generation to actuation and dynamic response.
The ***Missile block*** implements the missile's equations of motion expressed in the wind-aligned reference frame, thus enabling the computation of the missile trajectory. Within this block, the seeker module is embedded, which receives both the missile and target position and velocity vectors. It calculates the corresponding relative kinematic quantities and forwards them to the GNC block. The commands generated by the GNC subsystem are then passed through an ideal actuator (actuator dynamics are not modelled) - before affecting the missile dynamics. The block requires as input the aerodynamic coefficients, the thrust magnitude, the current missile mass, and the direction of the aerodynamic normal force. It outputs the relative position and velocity with respect to the target. An internal condition is set such that the simulation is terminated if the missile impacts the ground, which is interpreted as a failure to intercept the target. The ***GNC block***, which receives relative kinematic data from the seeker, is responsible for the generation and management of guidance and control commands. The Proportional Navigation (PN) law computes the required lateral accelerations to achieve target interception. These accelerations are processed in the Commanded Actions module, which evaluates the necessary lift coefficient and its orientation in the body frame and derives the flight path angles needed to support the commanded maneuver. The feasibility of the commanded aerodynamic conditions is verified using a pre-generated aerodynamic database that assesses whether the commanded lift coefficient is within allowable limits; if not, the command is saturated to the maximum admissible value. This module also returns the corresponding drag coefficient and the total effective angle of attack. Finally, the Attitude Angles module calculates both the aerodynamic angles (angle of attack and sideslip) and the missile's attitude angles (pitch and yaw), based on the missile's absolute and relative kinematic state. The result of this process is the set of control surface coefficients that are required to actuate the desired maneuver. The ***Engine block*** models the propulsion system using a simplified representation, assuming a constant thrust level throughout the burn phase until the propellant is depleted. Following burnout, the missile transitions to a coasting flight phase. The thrust is not actively modulated during the burn but remains fixed according to initial design parameters. The inputs to this block consist of the motor parameters - such as total impulse, burn time, and initial mass - and the current simulation time. Based on these inputs, the block provides the instantaneous thrust magnitude and the corresponding mass of the missile, updated over time to reflect propellant consumption. The ***Target Kinematics block*** governs the motion of the target, which is assumed to follow a prescribed trajectory defined by its initial kinematic state. The block receives as input the initial position, velocity, and acceleration of the target, and propagates these through the simulation according to a predetermined model, which in this study assumes non-maneuvering behavior. As output, it provides the target's instantaneous position and velocity at each time step, which are required by both the seeker and guidance logic to compute relative kinematics

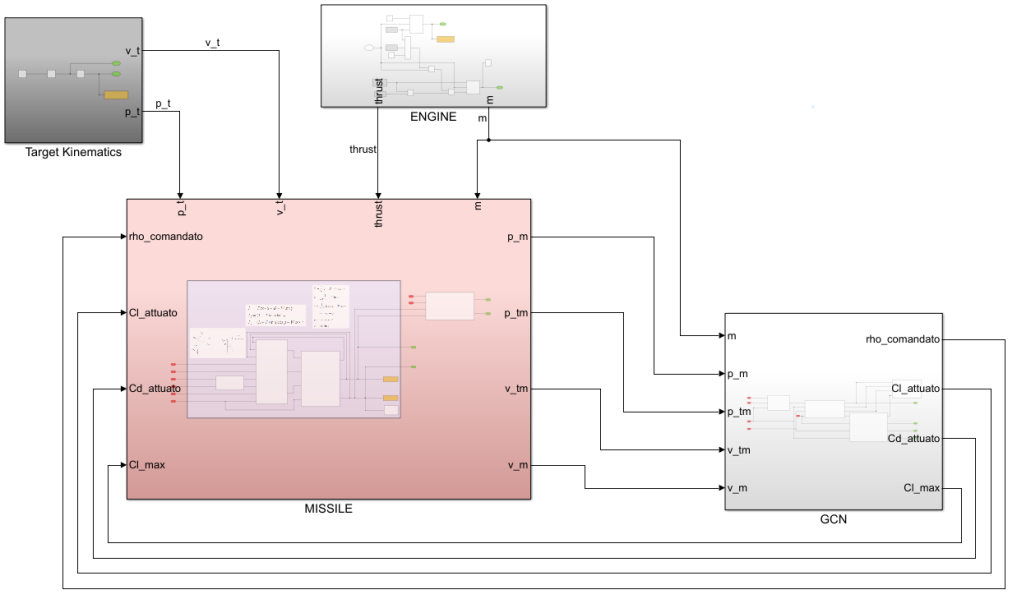


Figure 3: Simulink main blocks: Missile, GCN, Engine, Target Kinematics

DOI: 10.13009/EUCASS2025-597

The Simulink-based missile model adopts the following assumptions and simplifications:

1. *Point-Mass Dynamics*: The missile is modeled as a point mass. Aerodynamic coefficients are computed using a 2D missile representation under trimmed flight conditions.
2. *Inertial Earth-Fixed Reference Frame*: The simulation uses an inertial frame of reference aligned with Earth-fixed axes.
3. *Proportional Navigation (PN) Guidance Law*: The PN guidance logic commands lateral acceleration vectors that are always perpendicular to the missile's velocity vector. No commanded acceleration is applied along the missile's longitudinal axis.
4. *Ideal Seeker and Actuator Model*: An idealized seeker block is included to isolate and analyze interface behavior without introducing sensor inaccuracies. Similarly, an ideal actuator block is used.
5. *Small Angle Assumption for Angle of Attack and Sideslip* (α, β): It is assumed that the body axes approximately align with the wind axes. Consequently, the missile's longitudinal axis is aligned with the velocity vector, and the attitude angles coincide with the flight path angles. Specifically: Pitch angle $\theta \approx \gamma$; Yaw angle $\psi \approx \chi$; Roll angle $\phi = 0$. This implies that the rotation matrix between the Earth and body frame (R_{EW}) is equivalent to the Earth-wind frame (R_{EW}).
6. *Angle Derivations*: The angles of incidence (α, β) are computed a posteriori based on the total effective angle of attack α_{TOT} and the direction of the normal force vector ϕ_A. Given known values of heading angles (between Earth axes and wind axes) and aerodynamic angles (between body and wind axes), the attitude angles (between body and Earth axes) can also be derived.

2.1 Trajectory and dynamics

The equations describing the dynamics of the chaser are reported in the next set of equations :

$$\begin{cases} \dot{V} = \dfrac{1}{m}(T - D - mg\sin\gamma) \\ \dot{\psi} = \dfrac{1}{mV\cos\gamma}\, L\sin\varphi \\ \dot{\gamma} = \dfrac{1}{mV}(L\cos\varphi - mg\cos\gamma) \\ \dot{x} = V\cos\gamma\cos\psi \\ \dot{y} = V\cos\gamma\sin\psi \\ \dot{z} = V\sin\gamma \\ \dot{m} = -\dfrac{T}{I_s g_0} \end{cases}$$

The three dynamics equations are expressed in a missile-based reference frame, the wind axes system $F_w(O, x_w, y_w, z_w)$, usually x_w coincident with the velocity vector. The three kinematic equations are expressed in a ground-based reference frame, the Earth reference frame $F_e(O_e, x, y, z)$ and are usually referred to as down range, cross range, and altitude, respectively. x, y and z denote the components of the center of gravity of the missile, the radio vector $\vec{r}$, expressed in an Earth reference frame. φ, ψ and γ are the bank angle, the heading angle, and the flight-path angle, respectively. The navigation is imposed by the guidance law given by :

$$a_n = -N\frac{|\overrightarrow{V_r}|\,\overrightarrow{V_m}}{|\overrightarrow{V_m}|} \times \overrightarrow{\Omega}$$

Where a_n is the acceleration perpendicular to the missile's instantaneous velocity vector, N is the proportionality constant generally having an integer value 3-5 (dimensionless), $\overrightarrow{V_r}$ is the target velocity relative to the missile $\overrightarrow{V_r} = \overrightarrow{V_t} - \overrightarrow{V_m}$ and $\overrightarrow{\Omega}$ is the rotation vector of the line of sight : $\overrightarrow{\Omega} = \frac{\vec{R}\times\overrightarrow{V_r}}{\vec{R}\cdot\vec{R}}$.

2.2 Propulsion estimation : SRM engine sizing

Designing the shape of a solid rocket motor grain is a complex process involving multiple parameters. The primary goal in this study was to size the grain while ensuring feasibility and achieving the required thrust profile. A key assumption to simplify the initial modelling of the thrust profile is that the chamber pressure remains constant, and the burning area remains stable throughout the mission. A star-shaped grain is typically selected, as it provides a nearly constant thrust profile and can be analysed using a simplified approach.

Step 1: Performance Estimation**:** Starting with the required thrust T and burn time t_b for the motor, a suitable solid propellant formulation (e.g., Al/AP/HTPB) is selected. Using NASA's Chemical Equilibrium with Applications (CEA)

DOI: 10.13009/EUCASS2025-597

code, key combustion parameters are obtained: Chamber temperature (T_c), Chamber pressure (P_c), Molecular weight (M_{mol}), Specific impulse (I_{sp}), Expansion ratio (ε). Performance values such as characteristic velocity (C*), thrust coefficient (C_T), and specific impulse (I_{sp}) are analytically verified. Based on these values, the nozzle's throat and exit areas are determined, leading to the calculation of the exit diameter.

Step 2: Propellant Mass Calculation**:** Given the mass flow rate ṁ, the total impulse I_{tot}, and the propellant mass m_{prop} are computed as follows:

$$\dot{m} = \frac{T}{C^* C_T}; \qquad I_{tot} = I_{sp}\, \dot{m}\, g_0\, t_b; \qquad m_{prop} = \dot{m}\, t_b$$

Step 3: Grain Geometry Estimation**:** The required motor length is then estimated to accommodate the necessary propellant mass. By considering the burning rate r_b and the density ρ of the chosen propellant, the required burning area A_b is derived as: $A_b = \frac{\dot{m}}{r_b \rho}$; since a star-shaped grain is assumed, its initial burning area must be equal to its final burning area to maintain a stable thrust output: $A_b^{in} = A_b^{end}$.

The feasibility and dimensions of the grain are controlled through a convergence loop defined as $f(P, FF, r_b)$. This iterative process ensures that the required A_b, derived from the mass flow demand, converges to the geometric burning area A_b, calculated based on the designed grain volume:

$$V_{prop} = \frac{m_{prop}}{\rho} \qquad h_{prop} = \frac{V_{prop} * FF}{D^2 \pi / 4} \qquad A_b = h_{prop} \pi D_m$$

where: FF (Filling Factor) is related to propellant inside the chamber, typically ranging from 0.75 to 0.85 for a star grain and D_m represents the mean grain diameter. The burning rate r_b is determined using empirical relations specific to the selected propellant composition (Al/AP/HTPB).

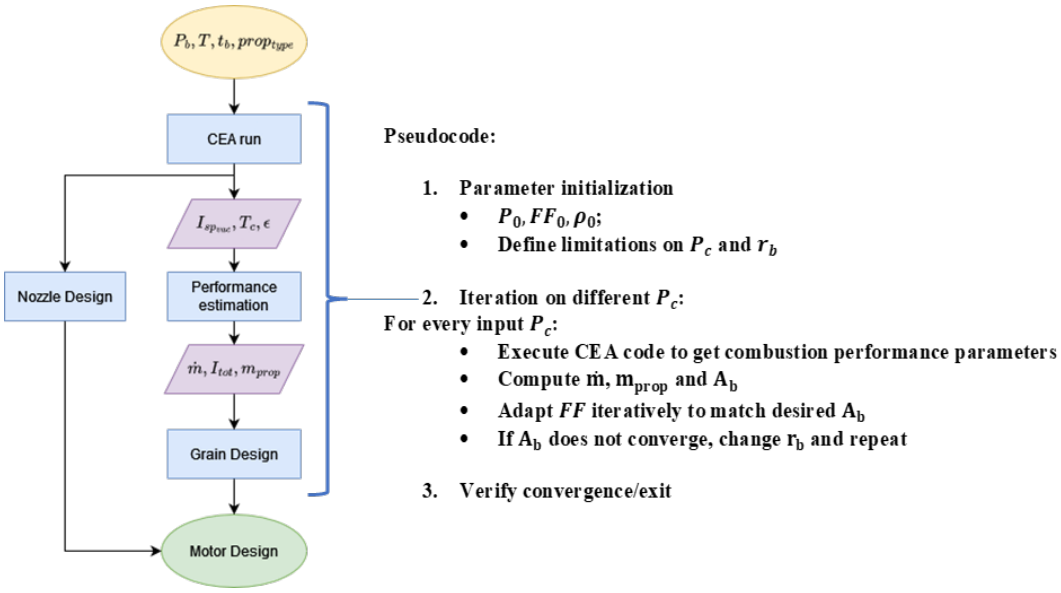


Figure 4: The preliminary grain design involves an iterative process where thrust, mass flow, propellant volume, and geometric constraints are balanced to achieve a feasible and efficient motor design. The convergence loop ensures that the computed burning area

2.3 Aerodynamic predictions

The missile aerodynamics were estimated using semi-empirical models suitable for preliminary design. The approach follows Fleeman's component build-up method [20], briefly summarized here. The total drag coefficient is calculated as:

$$C_{D_0} = C_{D_W} + C_{D_{SF}} + C_{D_B}$$

Where the three components can be computed with the following expressions:

DOI: 10.13009/EUCASS2025-597

$$C_{D_w} = \left(1.586 + \frac{1.834}{M^2}\right)\left(\frac{\tan(0.5)}{\frac{ln}{d}}\right) \qquad \begin{cases} C_{D_B} = \left(1 - \frac{A_e}{S_{ref}}\right)\frac{0.25}{M} \;\; if\ M < 1 \\ C_{D_B} = \left(1 - \frac{A_e}{S_{ref}}\right)(0.12 + 0.13M^2)\; if\ M > 1 \end{cases}$$

$$C_{D_{SF}} = 0.031\frac{l}{d}\left(\frac{M}{qL}\right)^{0.2} \;\; if\ M < 0.2$$

The normal force coefficient is computed as : $C_N = C_{N_{body}} + C_{N_{wing}} + C_{N_{tail}}$

The contribute due to wings and tail surfaces is computed by means of equation X. Those relations are based on the Newtonian impact theory : if the Mach number is lower than a certain critical value that theory is augmente using slendet body theory, if higher the linear wing theory is used.

$$\begin{cases} C_{N_{wing}} = \left(\frac{\pi A_r}{2}\,|\sin\alpha_w \cos\alpha_w| + 2\sin^2\alpha_w\right)\frac{S_{surf}}{S_{ref}} \;\; if\ M < M_{cr_{wing}} \\ C_{N_{wing}} = \left(\frac{4}{(M^2-1)^{\frac{1}{2}}}\,|\sin\alpha_w \cos\alpha_w| + 2\sin^2\alpha_w\right)\frac{S_{surf}}{S_{ref}} \;\; if\ M > M_{cr_{wing}} \end{cases}$$

Where the critical mach number for the surface is computed by : $M_{cr_{wing}} = \left(1 + \left(\frac{8}{\pi A_r}\right)^2\right)^{\frac{1}{2}}$

Finally, the friction drag and the wave drag coefficients associated to the surfaces are computed ccording to the following equations:

$$C_{d_{wing,friction}} = 0.0078\left(\frac{M}{qc_{mac}}\right)^{0.2}\frac{2\,S_{surf}}{S_{ref}}$$

$$C_{d_{wing,wave}} = \frac{1.429}{M_{\Lambda_{LE}}^2}\left(\left(1.2M_{\Lambda_{LE}}^2\right)^{3.5}\left(\frac{2.4}{2.8\,M_{\Lambda_{LE}}^2 - 0.4}\right) - 1\right)\frac{(\sin^2\delta_{LE}\, cos\Lambda_{LE} t_{mac} b_e)}{S_{ref}}$$

Where, t_{mac} is the thickness of the wing in the mean chord position, δ_{LE} is the leading edge angle in the same position, and $M_{\Lambda_{LE}}$ is the Mach number component perpendicular to the leading edge.

3. Baseline and MDO scheme

To validate the preliminary design framework, a reproduction of the AIM-7 Sparrow configuration was conducted as a reference case. The AIM-7 Sparrow (Airborne Interception Missile) is a medium-range, radar-guided air-to-air missile historically deployed alongside the AIM-9 Sidewinder. The key physical and performance parameters of the AIM-7 considered in this study are as follows: a total mass of approximately 197 kg, a solid rocket propulsion system (Rocketdyne MK 38/MK 52) providing 3447 kg of thrust sustained over 2.9 seconds, and a 30 kg payload. The missile is capable of sustaining maneuver loads up to 25g and achieves launch velocities up to Mach 2.2.

Starting from the propulsion subsystem, the engine configuration was reconstructed using known thrust and burn time values, as previously described in the propulsion modeling methodology. Given the known motor length and total missile geometry, the internal volume was used as a basis for allocating the remaining subsystems. The payload volume was estimated as a fixed fraction of the missile volume according to: $V_{PL} = 0.15 \cdot V_{missile}$

where V_{PL} is the volume allocated to the payload. The volume allocated to the guidance and control (G&C) subsystem was derived by subtracting the volumes of the propulsion and payload sections from the total missile volume:

$$V_{GC} = V_{missile} - V_{PL} - V_{prop}$$

With the subsystem volumes determined, corresponding mass estimates were obtained using Mass Estimation Relationships (MERs) sourced from the literature [21], which relate component mass to geometric and volumetric parameters, and results of the preliminary design were validated by comparison with specifications of the AIM-7.

DOI: 10.13009/EUCASS2025-597

Table 1: Error between preliminary designed components and baseline

Component	Baseline	Design	Err
GNC system	59,8 kg	64.97 kg	8,6%
$Prop$	68,2	65,37	4,15%
Payload	39 kg	33.13	15%
Wings	29 kg	26.33 kg	9,2%
TOTAL	197 kg	189,8 kg	3,65 %

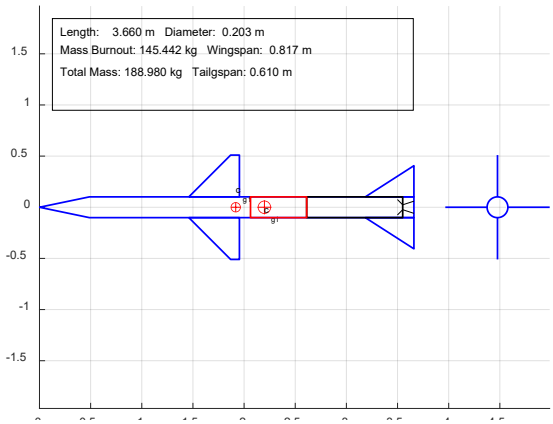


Figure 5: missile initial configuration

The intercept simulation was conducted using predefined initial conditions for both the interceptor (missile) and the target. The scenario assumes a launch with an initial velocity of 40 m/s, a ramp (pitch) angle of 45°. The target is positioned at coordinates $\vec{r} = [5000,0,2000]$ meters, traveling with a velocity vector of $\vec{v} = [-100,0,0]$m/s, corresponding to a straight, non-maneuvering trajectory. With this scenario in place, the Multidisciplinary Design Optimization (MDO) framework - as illustrated in the reference figure - was employed to identify an optimal missile configuration that minimizes the intercept time. The problem is defined as:

$$f(x) = \min t_f$$

$$c_{eq} = \left[bit_{miss} - 1; bit_{sat} - 1; check_{grain} - 1\right]$$

$$c_{neq} = \left[V_{prop} < 0.5V_{missile}; x_{LE_{tail}} + b_{tail} < l; \frac{l}{d} < 22\right];$$

where t_f denotes the final time at which interception occurs. The design space is defined by varying the primary control parameters within a ±10% range around their nominal values. The optimization problem includes both equality and inequality constraints. Equality constraints ensure mission feasibility by enforcing successful target interception (bit_miss), actuator effectiveness (bit_sat), and manufacturability of the designed grain geometry (check_grain). If any of these are not satisfied, the solution is considered unfeasible. The inequality constraints, on the other hand, address design limitations related to geometry and physical implementation of the chaser: these limit the maximum volume allocated to the propulsion system, enforce the correct placement of the tail leading edge within the body length, and cap the length-to-diameter ratio for structural integrity considerations.

An initial parametric analysis was carried out to tune key hyperparameters such as population size, crossover fraction, and the type of crossover operator. Based on this sensitivity analysis, a configuration yielding the best trade-off between convergence speed and solution quality was selected. The final settings used for the GA are detailed in Table 2.

Table 2: GA settings for optimization

Option	Value	Option	Value
PopulationSize	200	*FunctionTolerance*	1e-3
CreationFcn	@gacreationnonlinearfeasible	*ConstraintTolerance*	1e-3
CrossoverFcn	@crossoverscattered	*NonLinearConstraintAlgorithm*	penalty
EliteCount	20	*MaxStallGenerations*	10
CrossoverFraction	0.8	*SelectionFcn*	selectiontournament
MutationFcn	@mutationadaptfeasible	*UseParallel*	true

DOI: 10.13009/EUCASS2025-597

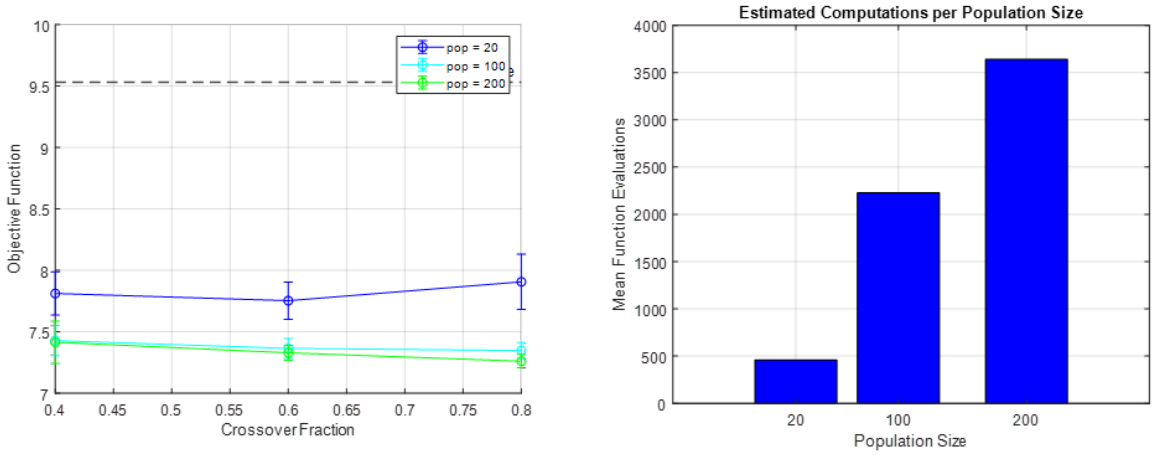


Figure 6: crossover fraction and population sensitivity with respect to objective function (left); mean function evaluations per optimization vs population size

Figure 7 presents the results of a sensitivity analysis performed on the GA hyperparameters, specifically crossover fraction and population size. Each data point represents the average outcome of ten independent optimization runs for a given parameter combination. The left plot shows the variation in the objective function value (intercept time in seconds) as a function of the crossover fraction, for three different population sizes: 20, 100, and 200. It is evident that increasing the population size results in improved optimization performance. The objective function value decreases from approximately 9.5 s (for population equal to 200) to a stable value around 7.3 s at higher population sizes. In contrast, the crossover fraction appears to have a limited effect on the optimization outcome, although the best results are observed for a fraction of 0.8, suggesting that increased genetic recombination slightly enhances convergence.
The right plot quantifies the computational cost in terms of mean number of function evaluations required for each population size. As expected, the computational effort increases with population size, ranging from below 1000 evaluations for a population of 20, to nearly 4000 for a population of 200. This trade-off highlights the balance between solution quality and computational expense when tuning evolutionary algorithm parameters.

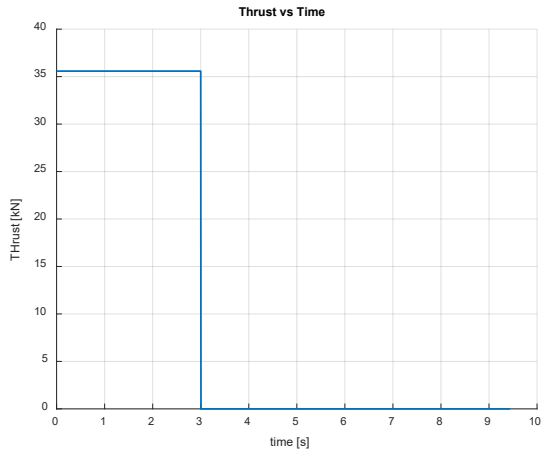


Figure 8: thrust profile of initial configuration

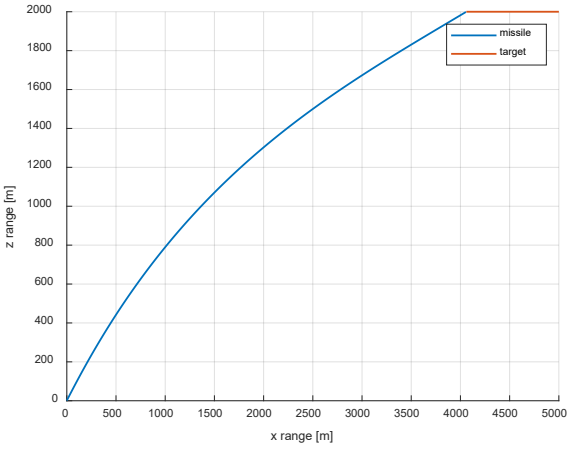


Figure 7: 2D interception trajectory of initial configuration

Figure 10 illustrates the optimization flowchart implemented in this study. The process is driven by a genetic algorithm, which manages a set of control variables acting as chromosomes in the evolutionary loop: each unique combination of these control variables defines a complete missile design configuration and triggers a corresponding mission simulation. The control parameters - grouped into aerodynamic (U_{aero}), propulsion (U_{prop}), and trajectory (U_{traj}) inputs - are fed into subsystem models responsible for geometry definition, propulsion design, and aerodynamic database generation. Subsystem masses are estimated via Mass Estimation Relationships (MERs), based on geometry and allocated volumes. The resulting configuration is then simulated through the Simulink model, which evaluates the missile's dynamic behavior and intercept capability under the specified initial conditions. The performance metric of each simulation is used by the genetic algorithm to evaluate the fitness of the chromosome, and the process iterates, evolving the population toward improved solutions by selecting, recombining, and mutating control variables, until convergence is achieved or stopping criteria are met.

DOI: 10.13009/EUCASS2025-597

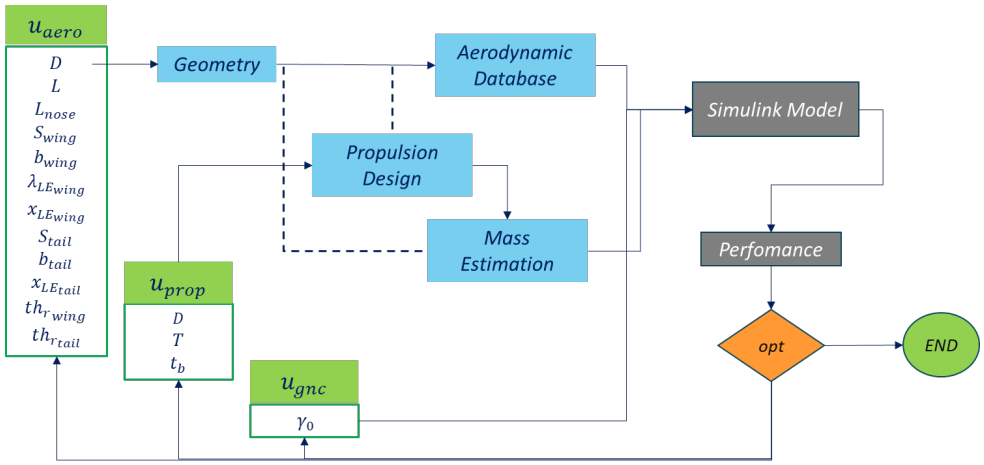


Figure 9: MDO design, simulation and optimization scheme

4. Results of Optimization and Ranking with Random Forest

The results of the optimization procedure yield to a missile configuration capable of intercepting the target approximately two seconds earlier than the initial (baseline) design. The optimized missile is slightly longer and features a smaller diameter, with a propulsion system that occupies relatively larger volume compared to the original configuration. Despite these geometric adjustments, the total mass is reduced by approximately 30 kg.
The optimized solution approaches the upper limit of the defined boundaries for both thrust and burn time, indicating that the algorithm favors designs with enhanced propulsive performance; simultaneously, the missile diameter reaches its lower bound, suggesting a design trend toward reduced frontal area and lower aerodynamic drag. This decrease in caliber contributes directly to the overall mass reduction. Additionally, since a larger proportion of the missile's internal volume is allocated to the propulsion system (while maintaining a fixed relative volume for the payload), the remaining volume available for the guidance and control subsystem is reduced. Given that the Mass Estimation Relationships (MERs) for electronics and GNC components are among the most mass-sensitive, this reallocation contributes significantly to the observed decrease in total system mass. The trajectory plots confirm improved performance, with the missile reaching intercept conditions earlier.

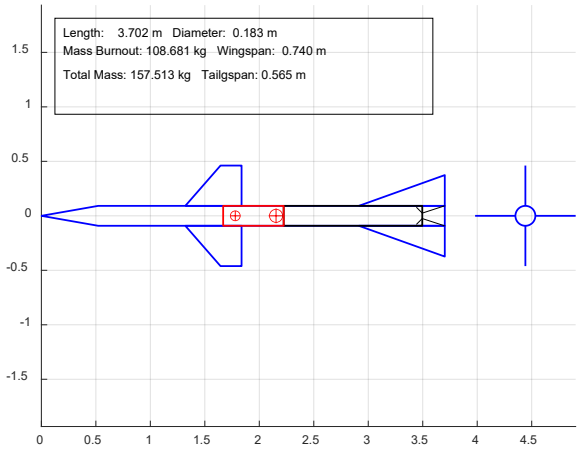


Figure 11: optimal concept missile configuration for selected mission

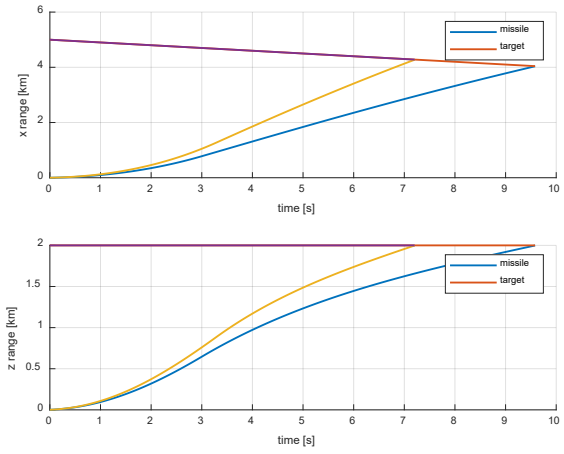


Figure 10: comparison between initial (blue line) and optimized (yellow line) mission trajectory

DOI: 10.13009/EUCASS2025-597

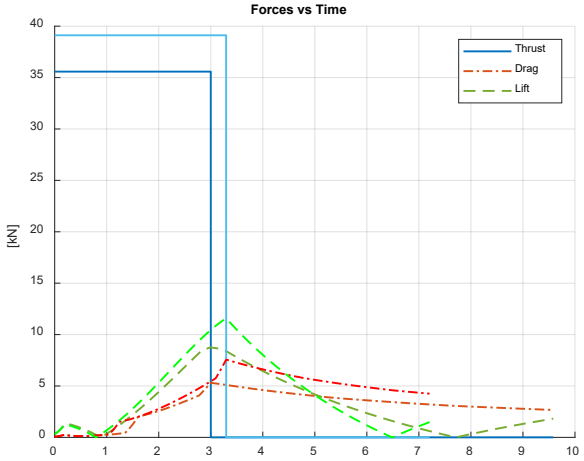


Figure 12: comparison between initial and optimal configuration

To analyze the obtained database, we used MATLAB's *TreeBagger* algorithm, which follows the method described by Breiman and was introduced earlier. This section presents a quantitative approach to selecting the optimal parameters for database analysis and subsequent ranking classification. The main hyperparameters influencing the performance of the algorithm include the number of trees (*M*), which determines how many decision trees are included in the forest; the minimum leaf size *(MinLeafSize*), which controls the minimum number of observations required in each terminal node of a tree; and the number of predictors *(NumPredictors*), which specifies how many features are considered for splitting at each node when constructing individual trees. To evaluate the quality of our results, we use the coefficient of determination, R^2, which is defined as:

$$R^2 = 1 - \frac{\sum(Y_{true} - Y_{pred})^2}{\sum(Y_{true} - \bar{Y})^2}$$

where: Y_{true} is the actual values of the target variable, Y_{pred} is the predicted values from the model, $\bar{Y}$ is the mean of Y_{true}. In figure 10 the R^2 is computed each time for different values of *NumPredictors*: results show how increasing this value from its original (default) value of p/3, increases the predictive performance of the algorithm. However, after a certain value, it reaches a plateau of validation MSE ~= 0.023: this corresponds to a mean error of about 0.15s on mission time of flight. The same procedure has been applied to tune the *MinLeafSize.*

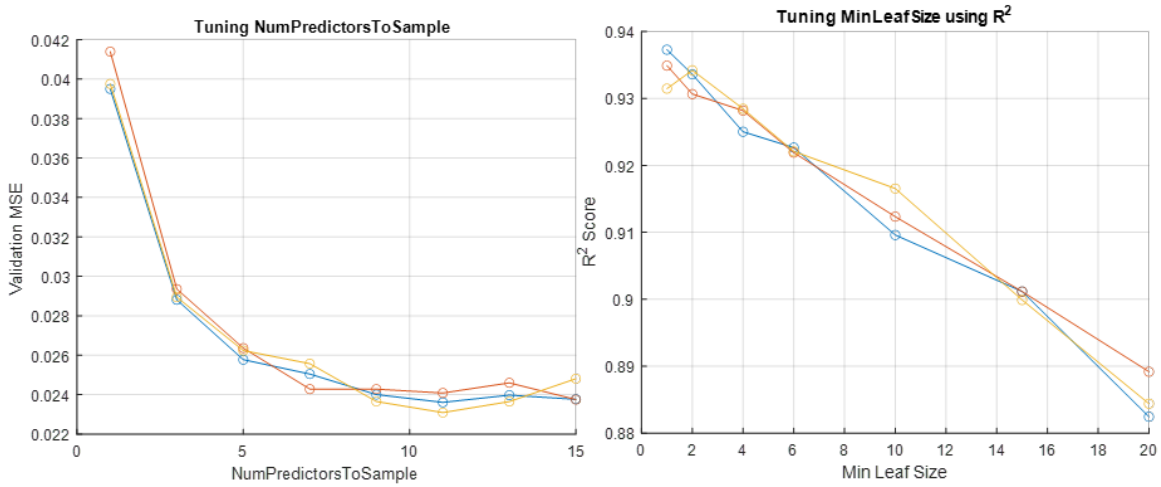


Figure 13: tuning of hyperparameters: number of predictors (left) and minimum leaf size (right)

DOI: 10.13009/EUCASS2025-597

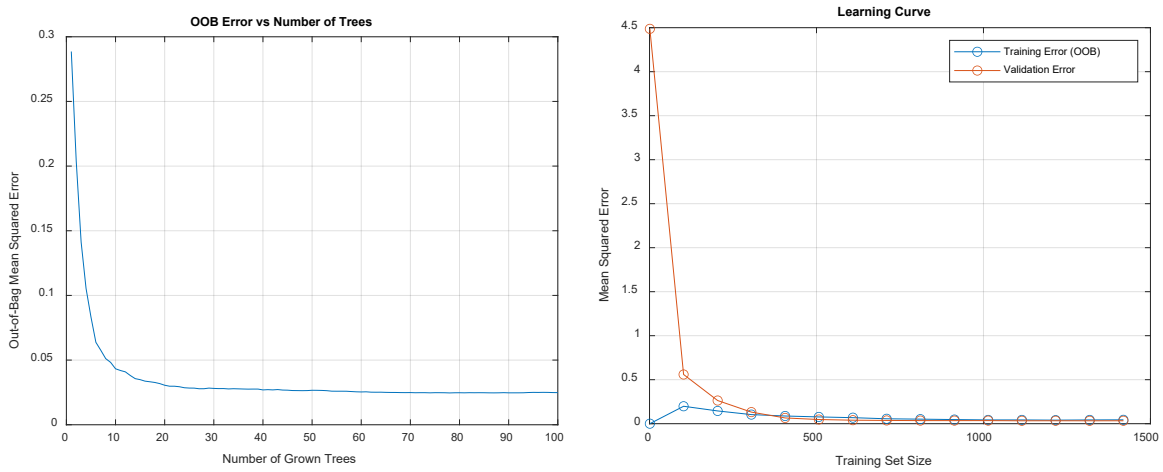


Figure 14: tuning of number of trees (left) and validation curve of learning of the forest (right)

For each instance in the dataset, the model prediction is made using only the trees for which that instance was OOB. The error between the predicted and actual values is then computed and averaged across all OOB samples, yielding an unbiased estimate of model performance. Effectively, the OOB error acts as a form of built-in cross-validation, fully integrated into the Random Forest training process [17],.

As shown in Figure 15, the OOB error decreases steadily as the number of trees increases, before converging to a stable value once approximately 50–60 trees are reached. This behaviour confirms that the model benefits from ensemble averaging and that adding more trees beyond this point does not substantially improve predictive accuracy. In addition to OOB validation, we also computed the Validation Error, which measures the model's performance on an entirely unseen subset of data. This external validation set consisted of 20% of the original dataset, withheld during training. Predictions on this set were made using the full trained ensemble (all 100 trees). The resulting validation error reflects the model's true generalization ability. The learning curve in the figure captures the behaviour of the model as the training set size increases. The stabilization of both the OOB error and the Validation Error - with both remaining low and close in magnitude - indicates that: the model is not overfitting the training data, the training is sufficiently robust and the selected hyperparameters result in a well-generalizing model. Based on the combined results of hyperparameter tuning and the convergence of error metrics, the final hyperparameters adopted for the Random Forest model are:

- Number of Trees (NumTrees): 100
- Minimum Leaf Size (MinLeafSize): 2
- Number of Predictors to Sample (NumPredictorsToSample): 7

These parameters offer a good trade-off between computational efficiency and model performance, achieving an R^2 of approximately 0.94.

Following model training, variable ranking was performed using two methods:

1. Permutation-based Mean Decrease in Accuracy (MDA)
2. OOB Error-based Variable Shuffling

Figure 15 illustrates the rankings produced by both methods. The most influential variables identified consistently across both techniques are thrust, missile diameter, burn time, tail surface span. The OOB-based method exhibits less contrast in importance scores among variables, while the MDA-based ranking shows a more distinct separation, clearly identifying the top contributors: propulsive variables like thrust and burning time are directly linked to flight time due to their impact on acceleration and speed, the missile diameter is strongly correlated with system mass - affecting both the motor and subsystems - while tail span is associated with control authority and guidance feasibility.

Additional ranking tests were conducted using identical procedures to validate robustness and reproducibility and are reported in the appendix. The MDA-based ranking consistently identified the same top variables in the same order. In contrast, the OOB method showed slight variability, with one test identifying an additional variable among the top four. Overall, MDA proved to be more robust and reliable for sensitivity analysis in this application.

DOI: 10.13009/EUCASS2025-597

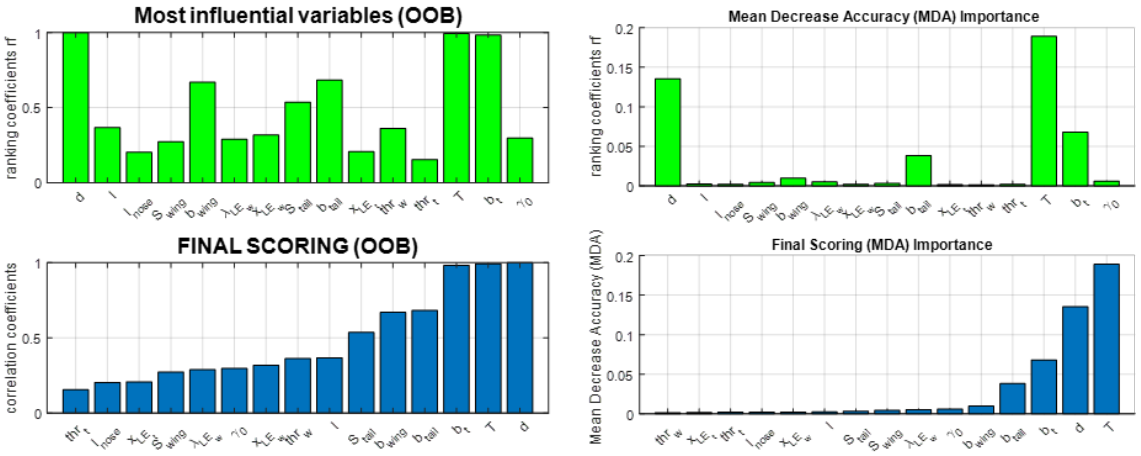


Figure 15: ranking of variables with OOB error (left) and MDA importance (right)

Conclusions

The proposed framework demonstrates the potential for a novel and efficient methodology for conducting sensitivity analysis within the context of preliminary mission design. By integrating a Random Forest surrogate modeling approach, the methodology offers a data-driven means to both predict mission outcomes and assess the relative influence of design variables. The model can capture both linear and nonlinear dependencies, offering interpretable importance metrics through techniques such as permutation-based feature importance (MDA) and OOB-based error rankings: a key advantage of the method lies in the fact that Random Forest training is intrinsically linked to data generated from optimized scenarios. This ensures that the model learns only from feasible design points, where mission constraints (e.g., performance, structure, guidance) are already satisfied. As a result, the regression focuses exclusively on the relevant region of the design space, preserving physical realism and reducing the need for excessive filtering or post-processing. Results indicate that for time-of-flight minimization mission, the thrust, diameter, burning time and tail span are the most important variables. Future research should address the potential for dimensionality reduction by excluding less significant variables to decrease computational load, while preserving optimization quality. Moreover, integrating surrogate modeling into real-time design loops could enhance responsiveness during trade-off analysis in early design stages.

Appendix

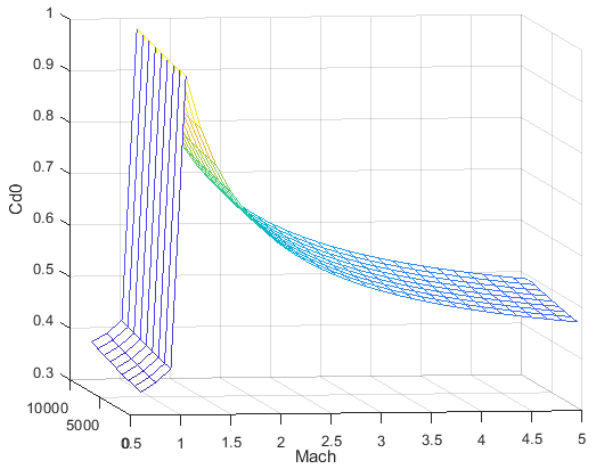


Figure 16: Drag coefficient Cd vs Mach and height curve

DOI: 10.13009/EUCASS2025-597

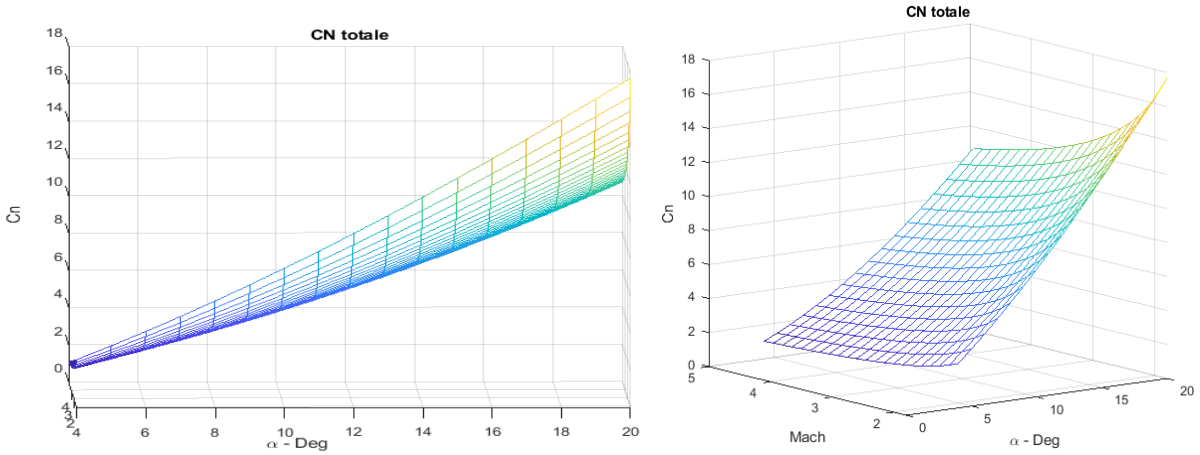


Figure 17: normal coefficient Cn vs AoA and Mach

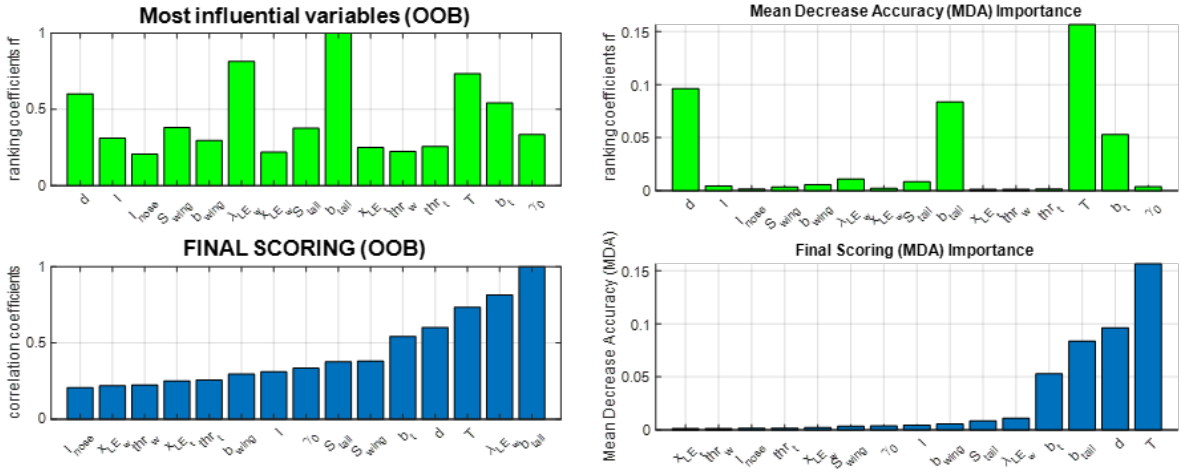


Figure 18: ranking of variables with OOB error (left) and MDA importance (right). test 2

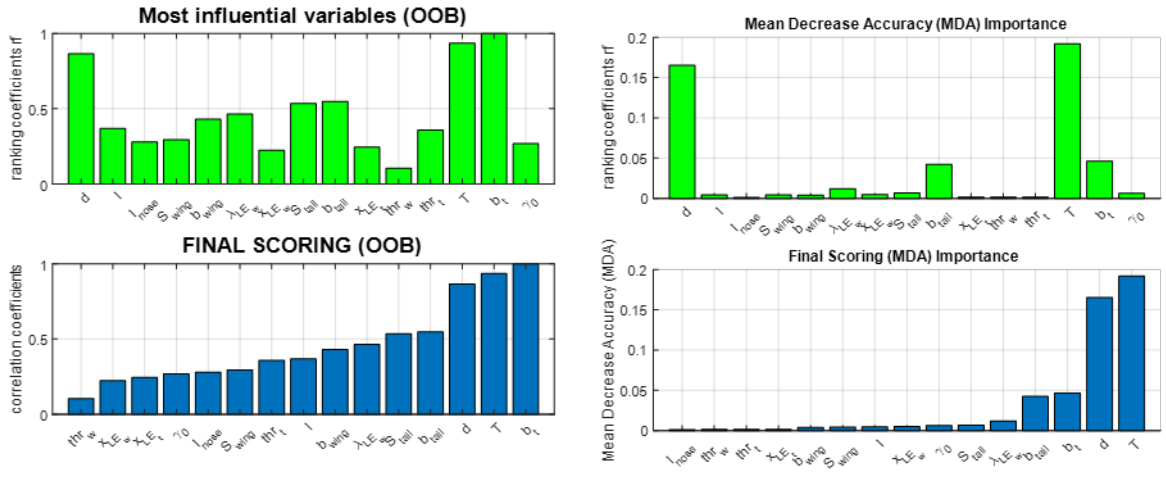


Figure 19: rankning of variables with OOB error (left) and MDA importance (right). test 3

Aknowledgements

DOI: 10.13009/EUCASS2025-597

This work is output of PhD student program, sponsored by Politecnico di Milano and MBDA Italia. The first author would like to thank F.Bianchini (francesco.bianchini@mbda.it) for his support in conceptualizing and debugging the simulink model associated with this study.

References

[1] Simpson, T. W., and Martins, J. R. R. A., "The Future of Multidisciplinary Design Optimization (MDO): Advancing the Design of Complex Engineered Systems," Arlington, VA, 2010.

[2] Khalil, M., Hashish, A., and Abdalla, H. M., "A Preliminary Multidisciplinary Design Procedure for Tactical Missiles," *Proceedings of the Institution of Mechanical Engineers, Part G: Journal of Aerospace Engineering*, Vol. 233, No. 9, 2019, pp. 3445–3458. https://doi.org/10.1177/0954410018797882

[3] Lesieutre, D., Dillenius, M., and Lesieutre, T., "Multidisciplinary Design Optimization of Missile Configurations and Fin Planforms for Improved Performance."

[4] Zheng, X., Gao, Y., Jing, W., and Wang, Y., "Multidisciplinary Integrated Design of Long-Range Ballistic Missile Using PSO Algorithm," *Journal of Systems Engineering and Electronics*, Vol. 31, No. 2, 2020, pp. 335–349. https://doi.org/10.23919/JSEE.2020.000011

[5] Afilipoae, T. P., Neculăescu, A. M., Onel, A. I., Pricop, M. V., Marin, A., Persinaru, A. G., Cismilianu, A. M., Munteanu, C. E., Toader, A., Sirbi, A., Bennani, S., and Chelaru, T. V., "Launch Vehicle - MDO in the Development of a Microlauncher," Vol. 29, 2018, pp. 1–11. https://doi.org/10.1016/j.trpro.2018.02.001

[6] Vassilios Silaidis, F. M. S. C. A. D. R. and E. D. N., "Sensitivity Analysis of Parameters on Multi-Disciplinary Design and Optimization Approach for Air-Launched Mission," 2025.

[7] Holst, T. L., "Genetic Algorithms Applied to Multi-Objective Aerodynamic Shape Optimization," Moffett Field, California, February 2005.

[8] Holland, J. H., "Genetic Algorithms," *Scientific American*, Vol. 267, No. 1, 1992, pp. 66–73. https://doi.org/10.2307/24939139

[9] Dancila, R. I., and Botez, R. M., "New Flight Trajectory Optimisation Method Using Genetic Algorithms," *Aeronautical Journal*, Vol. 125, No. 1286, 2021, pp. 618–671. https://doi.org/10.1017/aer.2020.138

[10] Salvador, R., Patrón, F., Kessaci, A., and Mihaela Botez, R., "Flight Trajectories Optimization under the Influence of Winds Using Genetic Algorithms," 2013.

[11] Shafaee, M., Mohammadzadeh, P., Elkaie, A., and Abbasi, S., "Layout Design Optimization of a Space Propulsion System Using Hybrid Optimization Algorithm," *Proceedings of the Institution of Mechanical Engineers, Part G: Journal of Aerospace Engineering*, Vol. 231, No. 2, 2017, pp. 338–349. https://doi.org/10.1177/0954410016636914

[12] Fatehi, M., and Hadi Taherzadeh, S. M., "Designing Space Cold Gas Propulsion System Using Three Methods: Genetic Algorithms, Simulated Annealing and Particle Swarm," *International Journal of Computer Applications*, Vol. 118, No. 22, 2015, pp. 975–8887.

[13] Anderson, M. B., Burkhalter, J. E., and Jenkins, R. M., "Design of a Guided Missile Interceptor Using a Genetic Algorithm," *Journal of Spacecraft and Rockets*, Vol. 38, No. 1, 2001, pp. 28–35. https://doi.org/10.2514/2.3668

[14] Bayley, D. J., Hartfield, R. J., Burkhalter, J. E., and Jenkins, R. M., "Design Optimization of a Space Launch Vehicle Using a Genetic Algorithm," *Journal of Spacecraft and Rockets*, Vol. 45, No. 4, 2008, pp. 733–740. https://doi.org/10.2514/1.35318

[15] Anderson, M. B., "Genetic Algorithms In Aerospace Design: Substantial Progress, Tremendous Potential," 2002.

[16] Dupont, C., Tromba, A., and Missonnier, S., "New Strategy to Preliminary Design Space Launch Vehicle Based on a Dedicated MDO Platform," *Acta Astronautica*, Vol. 158, 2019, pp. 103–110. https://doi.org/10.1016/j.actaastro.2018.04.013

[17] Breiman, L., "Random Forests," *Machine learning*, Vol. 45, 2001, pp. 5–32.

[18] Menze, B. H., Kelm, B. M., Splitthoff, D. N., Koethe, U., and Hamprecht, F. A., "On Oblique Random Forests," 2011.

[19] Biau, G., and Scornet, E., "A Random Forest Guided Tour," *Test*, Vol. 25, No. 2, 2016, pp. 197–227. https://doi.org/10.1007/s11749-016-0481-7

[20] Fleeman, E. L., "Tactical Missile Design," American Institute of Aeronautics and Astronautics, 2006.

[21] Nowell, J. B., "Missile Total and Subsection Weight and Size Estimation Equations," Naval Postgraduate School, Monterey, California, 1992.