

IAC-24,D2,5,8,x87239

Genetic Programming Guidance for the Reentry Trajectory of the ReFEx Vehicle

Francesco Marchetti^{a,*}, Jose Luis Redondo Gutierrez^a, David Seelbinder^a

^a Department of Guidance, Navigation and Control Systems, German Aerospace Centre (DLR), Institute of Space Systems, Bremen, Germany E-mail: francesco.marchetti@dlr.de, jose.redondogutierrez@dlr.de, david.seelbinder@dlr.de

* Corresponding author

Abstract

In this work Genetic Programming (GP) is used to obtain an alternative guidance law for online trajectory adaptation of the Reusability Flight Experiment (ReFEx) reentry vehicle. GP is an Evolutionary Algorithm (EA) capable of producing interpretable mathematical models that satisfies user defined objectives and constraints. It can be applied in a guidance setting, through direct interaction with the environment, learning to evolve a the guidance law to steer the vehicle towards the successful satisfaction of the mission. To the best of the authors' knowledge, the application of GP to a flight-ready vehicle represents a first in this domain and can help in increasing the TRL of this technique for Guidance and Control (G&C) applications. Furthermore, it produces interpretable models, in contrast to other Machine Learning (ML) approaches, and it can be applied to nonlinear models, in contrast with traditional methods.

In this study, GP is applied offline to design an feedback guidance law to guide the ReFEx vehicle towards a desired final position in the presence of uncertainties in the physical models, navigation and control signals. The performance of the GP-based guidance law is compared against the current design, which defines the trajectory correction as an optimal control problem and reduces it to a nonlinear unconstrained optimization problem, which is then solved using a successive linearization strategy. The performance comparison between the two methods utilizes high fidelity 6-DoF Monte Carlo simulations. Capabilities, advantages and disadvantages of both methods are discussed and conclusions on the viability of the investigated GP approach are drawn.

Keywords: Genetic Programming, Reusable Launch Vehicle, Guidance, Machine Learning

Acronyms/Abbreviations

CDR	Critical Design Review
DLR	Deutsches Zentrum für Luft- und Raumfahrt
EA	Evolutionary Algorithm
ECEF	Earth Centered Earth Fixed
ECI	Earth Centered Inertial
EI	Entry Interface
EoE	End of Experiment
G&C	Guidance and Control
GNC	Guidance, Navigation, and Control
GP	Genetic Programming
IGP	Inclusive Genetic Programming
MC	Monte Carlo
ML	Machine Learning
NN	Neural Network

PDF	Probability Density Function
RCS	Reaction Control System
ReFEx	Reusability Flight Experiment
RL	Reinforcement Learning
RLV	Reusable Launch Vehicle

1. Introduction

Thanks to the successes of companies like Space X and to advantages such as reduced cost and environmental impact, the aerospace industry is moving towards a greater usage of Reusable Launch Vehicles (RLVs). This paradigm shift currently involve only classical launch vehicles, such as rockets. However, several companies and research institutes are developing RLVs with spaceplanes

configurations^{1,2}, i.e. with lifting aerodynamic surfaces, since these vehicles can be fully reusable, possess greater control capabilities and can perform a greater variety of reentry trajectories than their rocket counterparts. One of such vehicles is the Reusability Flight Experiment (ReFEx) vehicle currently in development at the Deutsches Zentrum für Luft- und Raumfahrt (DLR) Institute of Space Systems in Bremen. ReFEx is part of the DLR effort to develop future winged reusable launcher stages and vehicles with reentry capabilities. The mission focuses on the reentry phase of a Vertical Takeoff and Horizontal Landing (VTHL) vehicle. The vehicle relies on its aerodynamic properties to reduce the kinetic energy during the reentry phase. One of the main objectives of the mission is to demonstrate autonomous GNC capabilities for aerodynamically controlled reentry vehicles [1]. The Critical Design Review (CDR) of the project was conducted in 2021. Within the mission, the current focus is now in V&V activities. The flight is planned for 2025 from the Koonibba Test Range in Australia.

Several approaches were developed to perform the reentry guidance of these vehicles. Traditional ones focus on finding offline the set of optimal guidance commands, i.e. an optimal trajectory, using direct optimal control techniques, and then tracking such trajectory during flight. These techniques rely on high-fidelity models and computationally expensive optimization approaches to find the optimal set of guidance commands. However, the optimization output is an open-loop guidance law and a robust controller must be used to track the desired trajectory when disturbances arise. An example of such approach can be found in the work of Bollino et al. [2], where a direct pseudospectral method is used to compute an optimal trajectory offline, and a PI-controller designed using dynamic inversion is used to track such a trajectory.

Other strategies focus on recomputing the guidance commands online to adapt to disturbances. When using these approaches, the optimization problem is simplified in order to be solved online during flight. An example of such approach can be found in the work of Sagliano et al. [3], where the authors formulate the guidance problem using a loss-less convexification approach to solve the optimal control problem online and generate guidance commands for a reentry vehicle. Their approach is tested with a Monte Carlo (MC) analysis where dispersions on the initial states are considered. Also the ReFEx vehicle currently employs an approach belonging to this group of

techniques, as described in [4, 5]. This approach is based on a two step algorithm: 1) trajectory planning, correcting for the dispersion in position and velocity after separation from the launcher, by generating once an updated trajectory leading from the estimated state to the target [4], and 2) trajectory tracking, correcting for the accumulating effect of modeling uncertainties and control and navigation errors, by periodically updating the trajectory so that it leads from the most recent estimated state to the target [5]. In both cases the algorithm consists in a simplification of the optimal control problem. First of all, the control variables are expressed as a function of a set of parameters. These parameters, whose definitions are critical as they define the solution space, define both the short term behaviour and the overall profile of the control variables. By combining this parameterization with a function that propagates the trajectory while accounting for constraints, the original optimal control problem can be transformed into a non-linear unconstrained optimization problem.

A last group of techniques to generate robust guidance commands focus on generating offline a feedback guidance law, capable of adapting the desired trajectory online, hence relieving the load on the control system. In recent years, ML has been applied to design such guidance laws. These approaches often involve Reinforcement Learning (RL) such as in the works of Gaudet et al. [6–8]. In particular, in [8] they applied reinforcement meta learning to optimize the adaptive guidance system of an hypersonic glider, to output the angle of attack and bank angle rates according to the current flight conditions. Also more traditional feedforward Neural Networks (NNs) have been used for this purpose. An example can be found in [9], where the authors created a real-time guidance system for the landing phase of a reusable rocket by integrating two NNs: one for classification and another for regression. The classification network classifies the initial conditions of landing flights into various groups, each linked to a specific thrust profile. Following this, the regression network provides the precise thrust values needed to ensure a successful landing.

This work aims at further exploring the application of another ML technique: Genetic Programming (GP). GP is an EA that can be used to generate guidance laws in a similar fashion to what is done with RL. Traditionally used for regression tasks it can also be applied in a guidance setting, where, through direct interaction with the environment, it learns how to evolve a guidance law to steer the vehicle towards the successful satisfaction of the mission in the presence of disturbances or uncertainties. The use of a GP-based guidance algorithm can lead to greater

¹<https://www.space.com/polaris-spaceplanes-mira-lig ht-flight-test-campaign-complete>

²<https://www.dawnaerospace.com/spacelaunch>

robustness against uncertainties w.r.t traditional methods, by exploiting the nonlinearities of the used models. In fact, the application of GP does not require linearization, and can easily include nonlinear constraints. Moreover, in comparison to other ML approaches, GP is interpretable, which makes it particularly interesting for G&C applications. Despite examples of GP applied to perform the aforementioned task can be found in the literature, to the best of the authors knowledge it was never tested on a flight-ready vehicle. Although GP will not be the algorithm used during the real mission, its application and testing on a realistic model in the loop simulator represents an innovative application to assess its feasibility for this type of vehicles and missions and to increase its TRL. Regarding examples of the GP application in a guidance context, in [10] GP is used to design the feedback guidance law of a reentry vehicle considering uncertainties. In this particular applications, only one set of uncertainties is considered during the GP training, hence the produced guidance law is not robust against unforeseen uncertainties. However, several guidance laws are produced considering increasingly more demanding uncertainties and GP proves capable of producing good performing guidance laws even when extremely prohibitive uncertainties are considered. In contrast to the previous application, where the GP evolutionary process is performed offline, in [11] GP is used online during flight effectively generating a new guidance law when necessary. The GP objective is to produce a guidance law for a Goddard rocket during its ascent phase, to cope with disturbances injected in the vehicle and environment models. This application shows the potential of GP for online usage, i.e. in an intelligent control setting. However, to generate the guidance laws online, the considered dynamical system must possess a low degree of nonlinearity in order to minimize the computational time required by the trajectory propagation.

The approach presented in this manuscript employs the same methodology described in [11], i.e. GP is used offline to produce a static feedback guidance law but, in contrast to the latter, considering multiple uncertainty configurations during the evolutionary process, hence producing a guidance law robust against unforeseen perturbations. Moreover, the chosen test case is a vehicle currently under development, whose 6DoF simulator is used to test the obtained guidance laws considering all the effects that will be present during the real flight, i.e. vehicle deformation, wind variation, navigation errors, etc. To the best of the authors' knowledge, this work represent the first application of GP to generate the guidance laws of a real vehicle and its outcome sheds light on the applicability of this ap-

proach for real missions, hence increasing the TRL of this technique.

The reminder of this paper is structured as follows: Section 2 provides a brief description of the ReFEx vehicle and the performed trajectory. Section 3 describes the guidance problem. Section 4 describes the GP algorithm and how to employ it for the guidance law generation. Section 5 presents the obtained results. The work is concluded by Section 6 that contains the drawn conclusions and future work directions.

2. Reusability Flight Experiment (ReFEx) Reentry Vehicle

ReFEx is a demonstrator mission that succeeds the DLR SHEFEX (Sharp Edge Flight Experiment) II, which was launched in 2012 [12]. ReFEx aims at performing an autonomously controlled and guided flight, following a trajectory representative for a winged RLV first stage. Therefore the vehicle will transition from hypersonic speeds down to a subsonic regime. This section introduces the vehicle and the trajectory for the mission. A more complete overview of the mission can be found in [13–15].

2.1 Trajectory

Figure 1 shows the sequence of events for the mission with some preliminary details. Several phases can be identified from this figure: 1) launch phase, 2) experimental phase until Entry Interface (EI), 3) experimental phase between EI and End of Experiment (EoE), and 4) experimental phase after EoE.

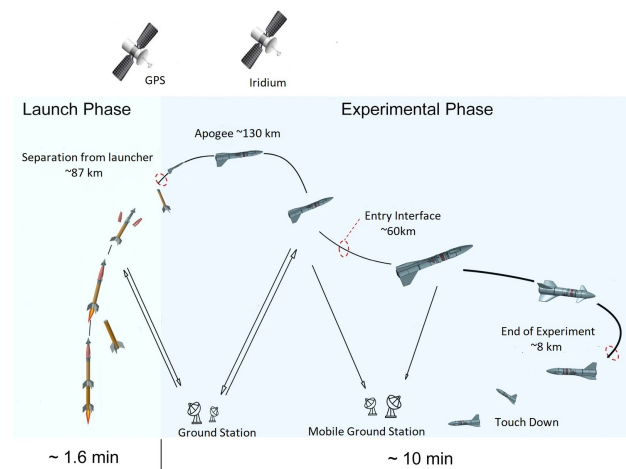


Fig. 1: Mission architecture & flight events [13].

A Brazilian solid propellant two-stage rocket VSB-30

will be used for the launch. This rocket is unguided and only passively stabilized, leading to a considerable state uncertainty at separation. During the launch phase most of the re-entry segment is covered using a hammerhead fairing to minimize its aerodynamic effect. After the two stages are burned out, the rocket is spun down using a Yo-Yo system and both fairing and rocket are separated. The current launch site is the Koonibba test range in Australia and the flight is planned for 2025. After separation and until EI the atmosphere is not dense enough to meaningfully affect the motion of the vehicle, leading to a ballistic trajectory. The aerodynamic actuators are also not effective and, thus, the Reaction Control System (RCS) is used. The objectives of this phase are: 1) to detumble the vehicle after separation, 2) to acquire the sun to enable a convergence of the navigation solution, 3) to reach EI with the desired attitude, and 4) to update the reference trajectory.

When EI is reached the dynamic pressure is already high enough to: 1) generate aerodynamic forces able to influence the trajectory and 2) control the attitude of the vehicle using the aerodynamic actuators. During this phase the objective is to use the aerodynamic forces to correct the trajectory, compensating for the state dispersion at separation, control and navigation errors and modelling uncertainties. This correction of the trajectory in order to reach the target with the required accuracy is the focus of the algorithm presented in this paper. The target point to be reached at EoE is defined in position and velocity. After EoE the vehicle aims at dissipating as much energy as possible to minimize the ground impact and ensuring a ground landing.

2.2 Vehicle

The re-entry segment of ReFEx has approx. 400 kg of mass and a longitudinal length of 2.7 m. The wingspan is 1.1 m and the diagonal terms of the moment of inertia are approx. 15 kg·m² for the longitudinal axis and 240 kg·m² for the other two axes. A section view of the vehicle is shown in Figure 2.

Several iterations on the shape of the vehicle have been undertaken through the history of the mission in order to fulfill aerodynamic requirements for the planned range of velocities, i.e. from hypersonic to subsonic. The current design consists of two fixed wings located at the back of the vehicle, a fin with an attached vertical tail actuator (rudder) and two canards positioned close to the nose of the vehicle. As the translational motion is controlled by aerodynamic forces, the actuators are used only for controlling the attitude of the vehicle. Two sets of actuators are used: 1) RCS, with 8 thrusters located at the back of the vehicle,

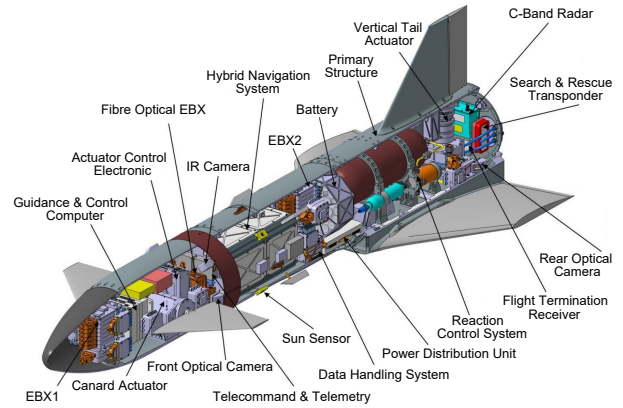


Fig. 2: Section view of the Re-Entry segment of ReFEx [13].

and 2) aerodynamic actuators, with the canards and rudder previously mentioned.

2.3 Equations of motion

The motion of the vehicle is formulated using the state vector $\mathbf{x} = [r, \phi, \lambda, v_G, \gamma_G, \chi_G]$ where r is the radial distance to Earth's center, ϕ the latitude, λ the longitude, v_G the norm of the velocity, γ_G the flight path angle and χ_G the heading angle. All variables are expressed with respect to a geocentric reference model. The velocity variables represent the velocity w.r.t. ground, i.e. Earth Centered Earth Fixed (ECEF) reference frame.

The time derivatives of the position can be expressed as in Equations 1

$$\begin{aligned} \dot{r} &= v_G \sin \gamma_G \\ \dot{\phi} &= \frac{v_G \cos \gamma_G \cos \chi_G}{r} \\ \dot{\lambda} &= \frac{v_G \cos \gamma_G \sin \chi_G}{r \cos \phi} \end{aligned} \quad (1)$$

When obtaining the expressions for the time derivatives of the velocity, it is important to keep in mind that ECEF is a rotating reference frame w.r.t. Earth Centered Inertial (ECI) reference frame. Therefore the apparent acceleration of the vehicle contains external forces (F), Coriolis acceleration due to the motion of the vehicle in the rotating frame and transport acceleration due to the angular rate of the rotating frame [16]. The resulting equations are listed in Equation 2.

$$\begin{aligned}
 \dot{v}_G &= \frac{F_v}{m} + \omega^2 r \cos \phi (\sin \gamma_G \cos \phi + \\
 &\quad - \cos \gamma_G \sin \phi \cos \chi_G) \\
 \dot{\gamma}_G &= \frac{F_\gamma}{m} + \frac{v_G}{r} \cos \gamma_G + 2\omega \cos \phi \sin \chi_G + \\
 &\quad + \omega^2 \frac{r}{v_G} \cos \phi (\cos \phi \cos \gamma_G - \sin \gamma_G \sin \phi \cos \chi_G) \\
 \dot{\chi}_G &= \frac{F_\chi}{m} + \frac{v_G}{r} \cos \gamma_G \tan \phi \sin \chi_G + \\
 &\quad - 2\omega (\tan \gamma_G \cos \phi \cos \chi_G - \sin \phi) + \\
 &\quad + \omega^2 \frac{r}{v_G \cos \gamma_G} \cos \phi \sin \phi \sin \chi_G
 \end{aligned} \tag{2}$$

The two contributions to the external forces that are considered are the gravity and aerodynamic forces. The gravity acceleration is obtained based in a model including the J2 term. The aerodynamic acceleration depends on: 1) the aerodynamic properties of the vehicle, 2) the atmosphere (i.e. temperature and density) and 3) the velocity w.r.t. the local atmosphere, i.e. including the wind. The aerodynamic properties of the vehicle were determined by the Institute of Aerodynamics and Flow Technology, also part of DLR, using Computational Fluid Dynamics simulations. More information about the generation of this database and its validation can be found in [17] and [18]. For the atmosphere, the NRLMSISE-00 Atmosphere Model is used. The wind model is obtained based on HWM14 (Horizontal Wind Model 14). The evolution of the attitude and angular rate is modelled using a simplified model that imposes a limitation on the angular acceleration and angular rate. For more information on the used models and assumptions, please refer to [5].

3. Problem description

The guidance algorithms are developed within the Guidance, Navigation, and Control (GNC) subsystem. This subsystem is responsible for maintaining stability and steering the vehicle to achieve a target state with a certain accuracy [1]. This objective can be traced back to GNC and mission requirements regarding the target accuracy at EoE. The target is defined taking into account other mission requirements such as reaching a subsonic speed and ensuring a minimum heading change. This accuracy requirement can be developed into two functional requirements for the guidance algorithms:

- Plan the trajectory after separation. Due to the considerable state uncertainty at separation, it may be necessary to modify the trajectory severely [4].

	Separation State	Target State	Tolerance at Target
r	76995.14 m	1513.25 m	± 1000 m
ϕ	133.44 deg	132.81 deg	± 0.074 deg
λ	-31.13 deg	-29.16 deg	± 0.069 deg
v_G	1315.93 m/s	190.55 m/s	± 50 m/s
γ_G	43.15 deg	/	/
χ_G	-1.48 deg	178.24 deg	± 20.05 deg

Table 1: Mission's initial and final conditions (target)

- Regularly update the trajectory throughout the reentry. The errors in navigation, control and modelling can accumulate leading to a divergence from the planned trajectory. Therefore, periodic limited corrections to the trajectory are required [5].

This paper presents an alternative algorithms for performing the latter. Regarding the first requirement, no dispersion on the initial conditions is considered, hence one successfully planned trajectory for the unperturbed case is used as a reference. The uncertainty on the initial conditions will be considered in a future work.

The vehicle's state at separation from the launcher and the desired target with relative tolerance are summarized in Table 1.

3.1 Challenges

Among the main challenges when designing the guidance strategy are: 1) the strong effect of control and navigation errors and model uncertainties on the trajectory, and 2) particularities of the flight constraints which are described in [5].

When planning the trajectory after separation, there are several unknown uncertainties and errors that can affect the flight, including:

- Model uncertainties. The trajectory is planned using nominal values to model the environment and the vehicle. These model parameters include a wide range of parameters, such as the aerodynamics properties of the vehicle, its mass properties, the velocity of the wind or the atmospheric density. Each of these variables is subject to uncertainties, which reduce the representativity of the planning. Figure 3 shows the profile of the wind velocity in North and West directions as function of altitude for 100 samples. Even though at high altitudes the effect of the wind on the trajectory of the vehicle is negligible, below 30 km altitude it can have a considerable impact. This fig-

ure also presents the dispersion of 100 samples of the mass, which is modelled as normal distribution with σ equal to 0.33 kg.

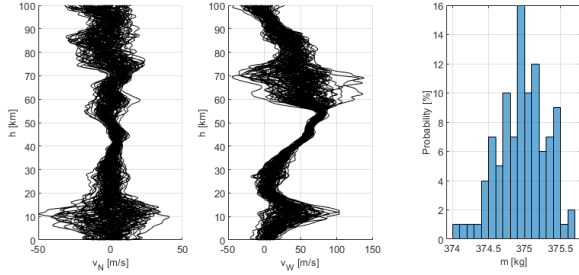


Fig. 3: Perturbed wind velocity profiles (left) and mass (right).

- Navigation errors. For the planning of the trajectory the navigation error is not known, leading to further inaccuracies. Figure 4 shows the evolution of the navigation error in angle of attack (α), angle of sideslip (β) and bank angle (μ) for a set of trajectories. During the ascent phase the attitude error can become considerably high, reaching up to 100 deg.) due to the fast rotations. During the exoatmospheric phase, i.e. until approx. 320 seconds, the error is reduced (to below 10 deg) by the combined use of magnetometer and sun sensor measurements. Once the vehicle reenter the atmosphere, the error in each angle remains around or below 0.5 degrees. This seemingly small errors can have considerable effects in the motion of the vehicle if sustained in time. Further description of the evolution of the attitude error can be found at [19].

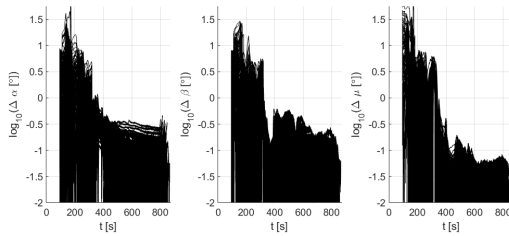


Fig. 4: Navigation error.

- Control errors. Even though the guidance ensures that the trajectory is feasible from a control perspective (e.g. by imposing limits in the maximum angular rates and angular accelerations), the existence of control errors is unavoidable. Figure 5 shows the

evolution of the control error for α , β and μ during the atmospheric phase for a set of trajectories. During the initial reentry (between 280 and 320 seconds) the error is considerably high, caused by fast improvements in the navigation solution and limited control authority due to the thin atmosphere. The bank angle error shows several peaks above 1 degree at lower layers of the atmosphere, which are caused by wide rotation maneuvers around this axis. The control errors have qualitatively similar effects on the accuracy of the trajectory planning as the navigation errors.

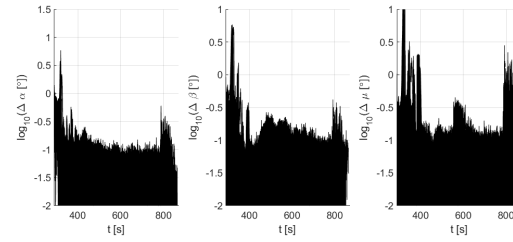


Fig. 5: Control error.

Due to the limitations described above, following the control profile defined in the initial trajectory planning would incur in a non-negligible error at EoE. Figure 6 shows the results of a MC campaign of 100 runs in which the trajectory is not periodically updated. This means that the controller aims at following the control profiles (in α and μ) that were defined in the initial trajectory planning. In the figure the blue color identifies the runs that met the requirements and the black color those that did not. The first gray segment contains the ballistic trajectory, until the dynamic pressure reaches 1000 Pa. The second gray segment contains the trajectory after the End of Experiment. In this campaign 26% of the cases failed to meet the requirements.

The flight constraints that need to be enforced during the mission can be divided in two main groups: 1) constraints to ensure the physical integrity of the vehicle and 2) constraints to ensure the controllability of the vehicle [4]. The first group includes limitation in variables such as the load, dynamic pressure, heat flux and heat load. The vehicle is designed such that these constraints are always met (for the envelope of trajectories considered) and, therefore, they are not considered when designing the trajectory update. The controllability of the vehicle is ensured via three constraints: 1) limitation in the angular rate, 2) limitation in the angular acceleration and 3) imposition of a flying corridor in the Mach- α domain. The reason for a flying corridor is that, due to the aerodynamic properties of the

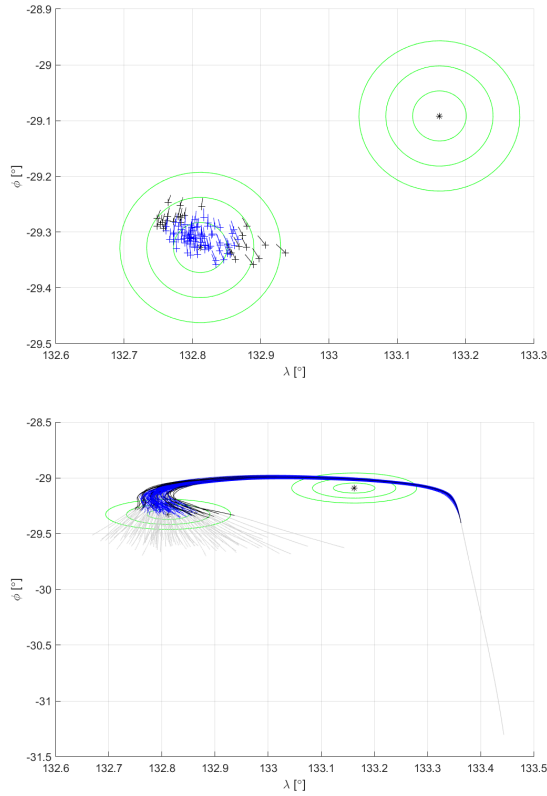


Fig. 6: Error at EoE (above) and trajectory (below) following initial trajectory planning.

vehicle, there are combinations of Mach and α where the vehicle is underactuated. In [20] a detailed analysis of the causes and consequences of this phenomenon can be found.

4. Genetic Programming

Genetic Programming (GP) [21] is an EA capable of autonomously evolve computer programs to achieve a user-defined objective. Compared to other EAs, GP's peculiarity is the tree-based structure adopted to represent the individuals. As an example, Figure 7 depicts a GP individual that encodes the mathematical expression $x * 10 + y + 3$.

As for other EAs, the core of the GP algorithm is represented by the evolutionary process, whose general formulation is illustrated in Algorithm 1. Here, crossover consists in swapping two randomly selected genes - substructures within a GP tree - of two randomly chosen individuals, while mutation consists in randomly varying a randomly selected gene in a randomly chosen individual. Finally,

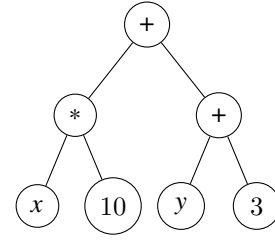


Fig. 7: Depiction of a GP individual.

selection consists in choosing a set of well performing individuals from the global population composed by parents P and offspring O to create the population for the next generation of the evolutionary process.

Algorithm 1 Pseudocode of GP evolutionary process

- 1: Initialize population P randomly
 - 2: Compute fitness of individuals in P
 - 3: Store best individual
 - 4: **while** Termination criteria is not met **do**
 - 5: Perform crossover and mutation to generate offspring O
 - 6: Evaluate fitness of individuals in O
 - 7: Apply selection to generate new population P'
 - 8: Update best individual
 - 9: **end while**
-

Several different approaches to perform crossover, mutation and selection can be found in the literature, and it is not the aim of this work to discuss them. The reader is referred to [21, 22] for more details. The particular GP formulation chosen for this work is the Inclusive Genetic Programming (IGP) [23], because of its ability to handle complex models efficiently and to produce multiple mathematical models simultaneously.

When employed for G&C applications, GP can be incorporated as depicted in Figure 8, where $\mathbf{x}$ are the states, $\mathbf{u}$ the guidance or control signals and $\mathbf{u}_{Ref}$ the reference guidance or control signal. The GP models are formulated as mathematical equations that output the desired signals. These are selected by the user and can be angle of attack and bank angle, such as in the case of a guidance application. But they can also be the actuators' deflection angles such as in a control application.

In the context of this work, GP is applied to generate corrections of the guidance commands, such as $\mathbf{u} = \mathbf{u}_{Ref} + \mathbf{u}_{GP}$. This way of applying the GP output facilitates the convergence of the GP evolutionary process. However, if

these are not available, GP can be applied also directly to steer the plant towards the desired trajectory, although a fine tuning of the algorithm might be required to help the convergence of the evolutionary process.

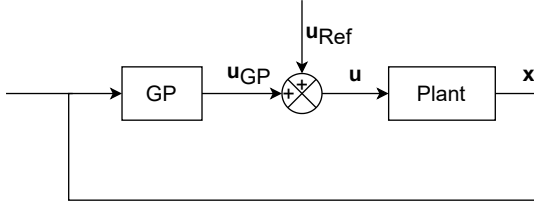


Fig. 8: Schematic application of GP in a G&C scheme

Once the GP models are incorporated in the control loop as depicted in Figure 8, the output of the GP models is computed for each simulation step or at the onboard computer frequency when used during flight. The computational cost associated with the GP models evaluation is the cost of one function evaluation.

When considering the application of GP in a G&C context, the fitness function is computed by testing the produced GP guidance laws through a direct interaction with the environment. This consists in propagating the plant's dynamics for every fitness evaluation, potentially resulting in a high computational load if the treated problem is highly nonlinear.

To limit the computational burden, a simplified configuration of the ReFEx simulator is used during the GP training phase. In this simulator, the state is propagated using the equations of motion shown in Section 2.3. This choice limits the type of uncertainties that can be considered during training. As explained in Section 3, three main groups of uncertainties affects the vehicle performance: model, navigation and control uncertainties. As a consequence of the usage of the simplified simulator, only models' uncertainties can be considered. However, during the test phase the complete simulator is used to consider the effects of all three groups of uncertainties, hence more realistic simulations.

The incorporation of uncertainties during training is necessary to obtain a GP guidance law robust against unforeseen uncertainties. For exposition's sake, from this point onward a combination of uncertainties is defined as uncertainty scenario. The more uncertainty scenarios are considered during training, the more robust the produced GP models will be against unseen uncertainties. However, the more uncertainty scenarios are considered, the greater

the computational burden will be. The user must define a number N_U of uncertainty scenarios to be considered during training, in order to obtain robust models without incurring in an excessive computational cost.

Algorithm 2 Pseudocode of GP fitness evaluation

- 1: Load N_U uncertainty scenarios
 - 2: **for** $i = 1 \rightarrow N_U$ **do**
 - 3: Set uncertainties as in $i - th$ uncertainty scenario U_i
 - 4: Propagate system's dynamics to obtain $i - th$ trajectory
 - 5: Evaluate total error e_T using Equation 3
 - 6: Store whether the trajectory was successfully propagated or not in the vector $\mathbf{S}$ (1=yes, 0=no)
 - 7: **end for**
 - 8: Compute the total fitness F_f using Equation 4
-

Uncertainty scenarios are considered during training as summarized in Algorithm 2: the trajectory propagation is repeated for each uncertainty scenario, resulting in an total error e_T computed using Equation 3, where $\hat{\mathbf{e}}$ is the scaled error between the states and their reference values. The scaling is performed by dividing the errors by the tolerance on the states at EoE, as reported in Table 1.

$$e_T = \sqrt{\sum |\hat{\mathbf{e}}|^2} \quad (3)$$

Each total error e_T is representative of the behaviour of the considered GP guidance law on that particular uncertainty scenario. Once all the uncertainty scenarios are considered, all the obtained global errors are stored in the vector $\mathbf{F}_s = [e_{T_1}, e_{T_2}, \dots, e_{T_{N_U}}]$. The number of successful propagations among those performed is stored in the vector $\mathbf{S}$, as 1 representing a success and 0 a failure. The definition of success or failure is defined by the user. More on the setting used in this work in Section 5.

Finally, the fitness of the individuals is computed using Equation 4. Here, *max* and *median* are used to take into account both the average behaviour across different uncertainty scenarios and the behaviour in the worst case, so to produce a GP guidance law that is generally well behaving and it has good performance even in the worst case.

$$F_f = \frac{\max(\mathbf{F}_s) + \text{median}(\mathbf{F}_s)}{\max(\sum \mathbf{S}, 1)} \quad (4)$$

5. Results

To assess the statistical reliability of GP, 10 runs, i.e. evolutionary processes, were performed. Each obtained

GP guidance law was tested by performing a MC composed of 100 runs. Each MC was set with a different combination of uncertainties and each of the 10 GP guidance law was tested on the same combinations of uncertainties. A trajectory is considered successfully propagated if the vehicle is within the hypercube defined in Table 1 at EoE. As mentioned above, during the training phase, a simplified simulator was used considering only the models uncertainties, while for the MC analyses a complete 6DoF simulator was used.

Two uncertainties scenario configurations were tested, one comprising only the models' uncertainties, referred to as "Limited Uncertainties", and the other considering model, navigation and control uncertainties, referred to as "All Uncertainties". These two configurations of uncertainties were considered because the GP models were generated using only the models uncertainties, therefore the configuration with limited uncertainties reflects the behaviour of the GP models on uncertainty scenarios of the same type of those used at training. While when using all the uncertainties, it is possible to test the GP models' robustness against unseen types of uncertainties. Moreover, the configuration with all uncertainties reflects the real environment during flight, hence the GP models must be tested with this configuration to assess the applicability of this technique in a real scenario.

5.1 Algorithm Setting

The IGP algorithm used to generate the guidance laws was set as in Table 2.

Regarding the parameters listed in Table 2, the input features correspond to the states and the energy linearly mapped from $[x_{min}, x_{max}]$ with $x = r, \phi, \lambda, v, \gamma, \chi, E$ to $[0, 1]$. The energy is computed as $E = E_{kinetic} - E_{potential}$, where $E_{kinetic} = \frac{1}{2}v_G^2$ and $E_{potential} = \frac{\mu}{r}$, where μ is Earth's standard gravitational parameter. Among the listed primitives, *add3* is a ternary addition, while *psqrt*, *plog* and *pexp* are the protected version of the *sqrt*, *log* and *exp* functions to avoid numerical errors. Regarding the mutation mechanisms, the percentage beside each of them represents the probability of selecting that particular mechanism when mutation is applied.

The fitness is computed using Equations 3 and 4, where the scaled errors $\hat{e}$ are computed only on the final point of the trajectory and considering all the states except γ . Five uncertainty scenarios are considered during the training phase.

Input features	$\hat{r}, \phi, \lambda, \hat{v}_G, \hat{\gamma}_G, \hat{\chi}_G, \hat{E}$
Population Size	300 individuals
Maximum Generations	300
Stopping criteria	Reaching the maximum number of generations
Number of niches	10
Crossover probability	0.2 (+0.01 at every generation if $P_f = 0$) $\rightarrow$ 0.65
Mutation probability	0.7 (-0.01 at every generation if $P_f = 0$) $\rightarrow$ 0.25
1:1 Reproduction probability	0.1
Crossover selection criteria	Best, random
Mutation selection criteria	Random
1:1 Reproduction selection criteria	Best
Evolutionary strategy	$M + \Lambda$
M	Population size
Λ	Population Size $\times$ 1.2
Number of Ephemeral constants	5
Limit Height	40
Limit Size	40
Selection Mechanism	Inclusive Tournament
Double Tournament fitness size	2
Double Tournament parsimony size	1.6
Tree creation mechanism	Ramped half and half
Mutation mechanisms	Uniform (55%), Shrink (5%), Insertion (25%), Mutate Ephemeral (15%)
Crossover mechanism	One point crossover
Primitives Set	$+, -, *, add3, tanh, psqrt$ $plog, pexp, sin, cos$
Fitness measure	F_f

Table 2: IGP setting.

5.2 Applied Perturbations

Five uncertainty scenarios were considered to compute the individuals' fitness. The process employed to select these scenarios is described hereafter: The simplified simulator was considered, hence only model uncertainties were applied. There were 500 MCs performed, each one set with a different uncertainty scenario and using the reference guidance commands. The resulting trajectories were classified by computing their total error using Equation 3. Also here, the errors $\hat{e}$ were computed considering only the last point of these trajectories. Figure 9 depicts the results of this analysis. Each point represents the total error e_T resulting from a trajectory propagation, and this is related to the severity of the considered uncertainty scenario. These are sorted in ascending order for ease of visualization, and the red circles highlight the trajectories, i.e. the uncertainty scenarios, used in the GP training phase. The trajectories are selected by linearly dividing the interval between the maximum and the minimum error by N_U . This methodology allows to select the least and most severe uncertainty scenarios plus other three in between that uniformly cover the range of possible uncertainty combinations.

5.3 Monte Carlo Analysis

The performance of the proposed approach was assessed by performing 100 MC runs. The same number of runs, with the same combinations of uncertainties, were performed using: 1) only the reference guidance commands; 2) the guidance commands generated by the online op-

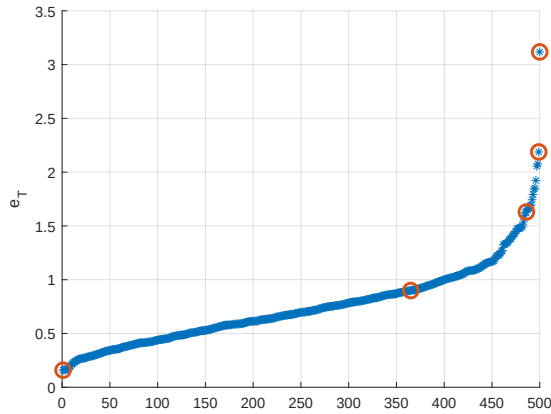


Fig. 9: Total errors of all trajectories produced as training data, sorted in ascending order. The red circles highlight those selected for the GP training.

timization routine presented in [5]; and 3) the guidance commands generated by the GP models.

For each of the three guidance approaches, two sets of uncertainties were considered: a limited one, comprising only the models' uncertainties, and another one considering all uncertainties, i.e. navigation and control uncertainties as well. These uncertainties are described in Section 3.

The results of the performed MCs are listed in Table 3. Here, the results are expressed in term of success rate, meaning how often the considered guidance law lead the vehicle in the desired final position at EoE. The results of all the 10 GP models are presented, along with the resulting median and standard deviation. "Reference" refers to the guidance commands planned after separation and not updated online, while "Online Optimization" refers to the guidance update performed with the optimization routine described in [5].

The same results are summarized in Figure 10, using boxplots to represent the GP's statistical distribution.

According to the obtained results, the GP proves capable of generating guidance laws that improve the performance of the reference guidance commands. Moreover, it achieves a success rate comparable to the optimization approach currently employed for the ReFEx guidance for both configurations of uncertainties. This also means, that the guidance laws produced considering limited uncertainties during training are still robust against a wider set of uncertainties. From a statistical perspective the GP success rate is around 95%, hence with slightly worse performance than

Run ID	Limited Uncertainties	All Uncertainties
1	95%	94%
2	95%	91%
3	99%	97%
4	96%	96%
5	95%	98%
6	98%	99%
7	86%	78%
8	96%	94%
9	71%	78%
10	96%	99%
Median	95.5%	95%
Std Dev	8.4%	8%
Other Approaches		
Reference	71%	74%
Online Optimization	100%	99%

Table 3: Success rates of the GP models and those of the reference guidance commands, for comparison. The values in bold corresponds to the results plotted in Figures 11 and 12

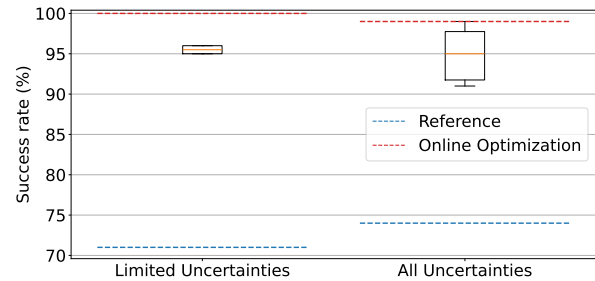


Fig. 10: Comparison between the GP and reference results

the optimization approach. This suggests that, during the design phase, multiple GP simulations should be conducted to achieve optimal results. However, these are preliminary findings based on a statistical sample composed of 10 GP evolutionary processes. A statistical sample composed by more than 10 GP runs should be considered to better assess the reliability of the GP algorithm. This point will be addressed in future works.

Figures 11 and 12 depict the Probability Density Functions (PDFs) of the reference guidance, the online optimization and the GP guidance for the configurations with limited uncertainties and with all uncertainties respectively. These PDFs are obtained by considering only the successful MC simulations. These plots depict the probability of having small errors in the selected quantities at EoE, i.e. a smaller distance from the desired target. In Figure 12 two lines are plotted for the best GP, since two simulations

got the highest success rate (99%), as reported in Table 3. From these results it can be observed how the best GP models tend to have bigger errors at EoE compared to the optimization approach. This can be explained by how the GP guidance laws are applied. They are defined by a single mathematical models that is applied as a correction for the reference guidance commands, as depicted in Figure 8. Therefore, one model must be capable of successfully guiding the vehicle when multiple uncertainties configurations are applied. For this reason, the GP evolutionary process favor those models that do not overfit the training data, meaning that they lead the vehicle towards the desired target but without exceedingly reducing the final accuracy that would result in overfitting and a loss of robustness.

These considerations are also highlighted in Figures 13 and 14. These plots depict the 100 trajectories resulting from the MC analyses, considering the online optimization and the best GP model (Simulation 6) respectively. The MC analyses were performed considering all uncertainties. As can be observed from these plots, the optimization approach tend to greatly reduce the error at the EoE, since that is the optimization objective. After EoE, other mission objectives are prioritized (i.e. reducing kinetic energy and ensuring land landing), and the vehicle's heading is heavily modified (see [5]). The GP model shows a wider distribution of the positions at EoE. As the after-EoE objectives are out of the scope of the GP model, the trajectories after reaching the target do not diverge and continue following their path until they reach the ground.

Lastly, the convergence of the GP evolutionary process is analyzed in Figure 15. Here, the mean (solid line) and standard deviation (shaded areas) of the fitness value across the performed generations is showed. The results are depicted using the logarithmic scale to highlight the small differences towards the end of the evolutionary process. Mean and standard deviations are computed considering the 10 GP evolutionary processes performed. From these results it can be observed how the evolutionary processes converge towards similar fitness values. Therefore, the fitness value itself is not directly indicative of the performance of the obtained GP models, since, with similar fitness values, models with 71% and 99% success rates were obtained. This suggests that the fitness functions should be formulated differently to further consider the different behaviours on the considered uncertainty scenarios during training, or to consider more uncertainty scenarios during training. This would improve the robustness of the obtained models.

6. Conclusions

This work presented the application of GP to generate feedback guidance laws for the ReFEx RLV. The guidance strategy implemented for this missions consists of two steps: 1) correcting once for dispersion after separation from the launcher (trajectory planning), and 2) correcting periodically for the effect of modelling uncertainties and control and navigation errors (trajectory tracking). Both algorithms are based on a simplification of the optimal control problem into a non-linear unconstrained optimization problem.

In this work, GP is proposed as an alternative to the aforementioned approach. The main advantage of GP compared to other optimization-based approaches, is that the GP algorithm can be applied across different test cases, by adjusting the definition of the vehicle's dynamics and properties, while the algorithm's inner working does not require fine tuning for each different test case. On the other hand, optimization-based approaches require the formulation of the optimization problem that is specific for each considered test case. Moreover, to the best of the authors' knowledge this is the first application of the GP algorithm to an accurate model of a real vehicle, therefore this work contributes in increasing the TRL of the GP technique for guidance design.

10 GP evolutionary processes are performed, producing 10 different guidance models. During the evolutionary process, a simplified plant model was used for propagating the equations of motion and obtaining the terminal condition achieved by the evolving guidance law. For testing the guidance law finally obtained at the end of the evolutionary process, a complete 6DoF simulator with the controller in the loop was considered, and two uncertainties settings were considered, one with only models uncertainties and the other with uncertainties applied to the models, navigation and control signals.

The results obtained by the GP models are comparable to those of the current ReFEx guidance. Statistically, the success rate improves by 24.5% and 21% for the configuration with limited and all uncertainties, if compared with the reference (not updated) guidance commands. A peak of 99% success rate was observed in both uncertainties configurations, while the current guidance approach shows 100% and 99% success rates for the configurations with limited and all uncertainties respectively. Statistically, GP is slightly worse with a median success rate of 95% with a standard deviation of 8%. Therefore, at design phase, multiple GP evolutionary processes should be performed in order to find the best performing one. These results show GP's promising potential. At this stage it could be

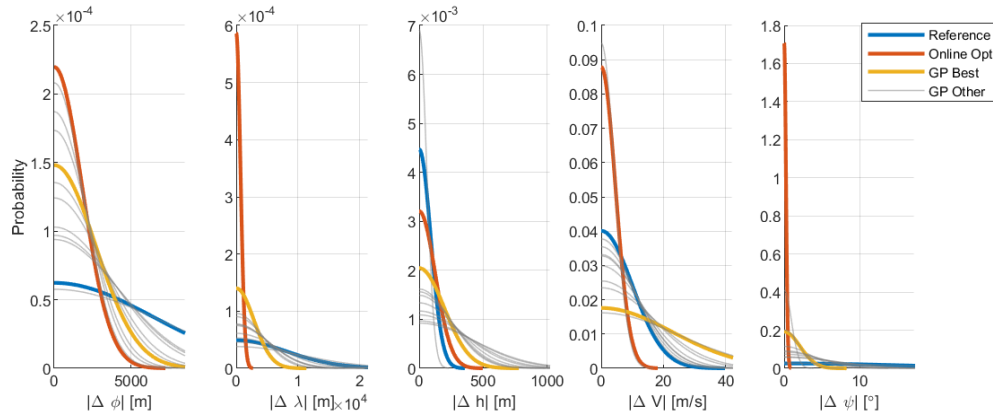


Fig. 11: PDFs of the considered guidance approaches with limited uncertainties

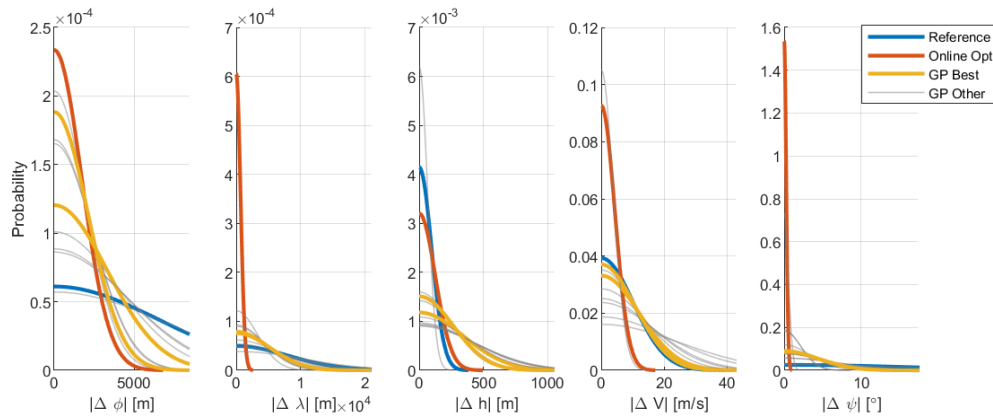


Fig. 12: PDFs of the considered guidance approaches with all uncertainties

suitable as a method that is cheap to evaluate during real-time closed loop and could be used e.g. as backup or initial guess for other, more computationally expensive guidance methods.

Three main future work directions can be identified: first, the number of GP evolutionary processes should be increased to better assess its statistical reliability; second, the uncertainty on the initial position should be included, since it is a problem affecting the ReFEx mission; and finally, the fine tuning of the GP algorithm should be investigated to improve its performance.

References

- [1] Jose Luis Redondo Gutierrez, David Seelbinder, Stephan Theil, Pablo Bernal Polo, Björn Gäßler, Johannes Robens, and Paul Acquatella B. Refex:

Reusability flight experiment - architecture and algorithmic design of the gnc subsystem. In *Aerospace Europe Conference 2023 - 10th EUCASS - 9th CEAS*, 2023.

- [2] Kevin P. Bollino, Michael W. Oppenheimer, and David D. Doman. Optimal guidance command generation and tracking for reusable launch vehicle reentry. *Collection of Technical Papers - AIAA Guidance, Navigation, and Control Conference 2006*, 7:4614–4636, 2006.
- [3] Marco Sagliano and Erwin Mooij. Optimal drag-energy entry guidance via pseudospectral convex optimization. *Aerospace Science and Technology*, 117:106946, 2021.
- [4] Jose Luis Redondo Gutierrez, David Seelbinder, and Stephan Theil. Design of refex guidance: Trajectory correction after ascent. In *AIAA Science and Tech-*

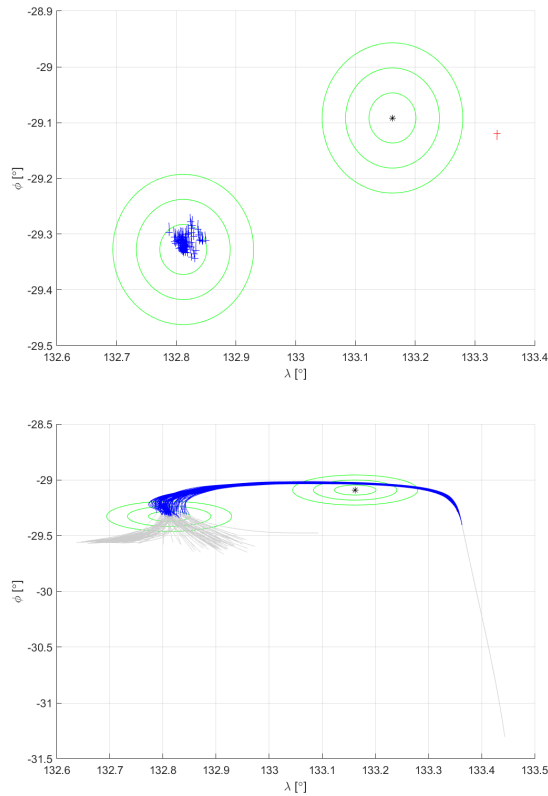


Fig. 13: Error at EoE (top) and trajectory (bottom) following initial trajectory planning. The plots depict the configuration with all uncertainties and using the on-line optimization routine.

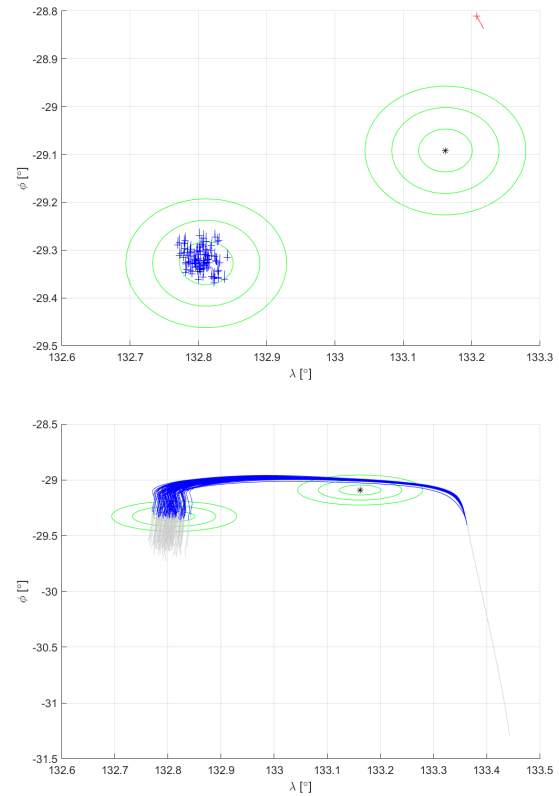


Fig. 14: Error at EoE (top) and trajectory (bottom) following initial trajectory planning. The plots depict the configuration with all uncertainties and using the best GP model obtained in simulation 6.

nology Forum and Exposition, AIAA SciTech Forum 2023. American Institute of Aeronautics and Astronautics, Inc., 2023.

- [5] Jose Luis R. Redondo Gutierrez, David Seelbinder, and Stephan Theil. Design of reflex guidance: Periodic on-board trajectory updates. In *AIAA SCITECH 2024 Forum*. American Institute of Aeronautics and Astronautics, 1 2024.
- [6] Brian Gaudet, Richard Linares, and Roberto Furfaro. Deep reinforcement learning for six degree-of-freedom planetary landing. *Advances in Space Research*, 65:1723–1741, 2020.
- [7] Brian Gaudet, Richard Linares, and Roberto Furfaro. Adaptive guidance and integrated navigation with reinforcement meta-learning. *Acta Astronautica*, 169:180–190, 2020.

- [8] Brian Gaudet, Kris Drozd, and Roberto Furfaro. Adaptive approach phase guidance for a hypersonic glider via reinforcement meta learning. In *AIAA SCITECH 2022 Forum*. American Institute of Aeronautics and Astronautics, 1 2022.
- [9] Jinbo Wang, Hongjun Ma, Huixu Li, and Hongbo Chen. Real-time guidance for powered landing of reusable rockets via deep learning. *Neural Computing and Applications*, 35:6383–6404, 2023.
- [10] Francesco Marchetti and Edmondo Minisci. Genetic programming guidance control system for a reentry vehicle under uncertainties. *Mathematics*, 9:1–17, 2021.
- [11] Francesco Marchetti, Edmondo Minisci, and Annalisa Riccardi. Towards intelligent control via genetic programming. *Proceedings of the International Joint Conference on Neural Networks*, 2020.

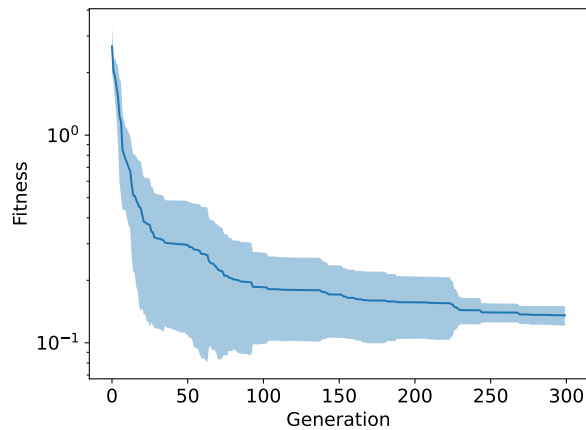


Fig. 15: Fitness evolution. The mean is the solid line, while the shaded areas represent the standard deviation.

- [12] Thino Eggers. The shefex ii experimental re-entry vehicle: Presentation of flight test results. In *28th International Congress of the Aeronautical Science (ICAS)*, 2012.
- [13] Waldemar Bauer, Peter Rickmers, Alexander Kallenbach, Sven Stappert, Viola Wartemann, Clemens Hans-Joachim Merrem, René Schwarz, Marco Sagliano, Jan Thimo Grundmann, Andreas Flock, et al. Dlr reusability flight experiment refex. *Acta Astronautica*, 168:57–68, 2020.
- [14] Peter Rickmers, Waldemar Bauer, Martin Sippel, and Sven Stappert. Refex: Reusability flight experiment—a flight experiment to demonstrate controlled aerodynamic flight from hypersonic to subsonic velocities with a winged rlv. In *7th European Conference for Aeronautics and Space Sciences (EUCASS)*, July, pages 3–6, 2017.
- [15] Peter Rickmers, Sebastian Kottmeier, Guido Wübbels, and Waldemar Bauer. Refex: Reusability flight experiment - a demonstration experiment for technologies for aerodynamically controlled rlv stages. In *Aerospace Europe Conference 2023 – 10TH EUCASS – 9TH CEAS*, July 2023.
- [16] E Mooij. The motion of a vehicle in a planetary atmosphere. *Delft University of Technology, Faculty of Aerospace Engineering, Report LR-768*, 1994.
- [17] Clemens Hans-Joachim Merrem, Viola Wartemann, and Thino Eggers. Aerodynamic data set generation for the experimental vehicle refex. *Notes on Numer-*

ical Fluid Mechanics and Multidisciplinary Design, pages 132–140, 2021.

- [18] Viola Wartemann, Ntarinai Konosidou, Andreas K. Flock, and Clemens Hans-Joachim Merrem. Contribution of numerical and experimental flow simulations to the aerodynamic data base of the dlr reusable flight experiment refex. *Notes on Numerical Fluid Mechanics and Multidisciplinary Design*, 151, 2021.
- [19] Joao Dias Basto da Silva, Jose Redondo, and Pablo Bernal. Refex navigation design: Improvements to the navigation filter. In *75th International Astronautical Congress (IAC)*, October 2024.
- [20] Jose Luis Redondo Gutierrez, David Seelbinder, Johannes Robens, and Paul Acquatella. Design of a control allocation solution for the winged reusable launch vehicle refex. In *AIAA Scitech 2022 Forum*, page 1842, 2022.
- [21] John R Koza. *Genetic programming: on the programming of computers by means of natural selection*. MIT Press, 1992.
- [22] Riccardo Poli, W.B. Langdon, and N.F. McPhee. *A Field Guide to Genetic Programming*. 2008.
- [23] Francesco Marchetti and Edmondo Minisci. Inclusive Genetic Programming. In Ting Hu, Nuno Lourenço, and Eric Medvet, editors, *Genetic Programming*, pages 51–65, Cham, 2021. Springer International Publishing.