Rotorcraft low-noise trajectories design: black-box optimization using surrogates

Pierre Dieumegard, Sonia Cafieri, Daniel Delahaye, R John Hansman

Rotorcraft low-noise trajectories design: black-box optimization using surrogates

Pierre Dieumegard Sonia Cafieri Daniel Delahaye

R. John Hansman

Abstract

This paper addresses the noise-minimal trajectory optimization problem for a specific type of aircraft: rotorcraft. It relies on a realistic noise footprint computation software provided by industry that is black-box. Locally optimal trajectories are computed through a tailored solution approach based on the Mesh-Adaptive Direct Search algorithm. We propose multiple surrogates defined according to our knowledge of the problem, including a surrogate relying on the physics of the problem (approximating the rotorcraft noise model), and another based on a machine learning (neural network) method. The proposed solution approach is further enhanced by the computation of an appropriate starting guess through a path planning algorithm tailored to the problem, and by the reduction of the variable space domain. The performance of the proposed methodology both in terms of quality of the solutions (trajectories exhibiting significant noise reduction compared to those currently flown in practice) and computing time is illustrated through numerical experiments on real-world case studies.

1 Introduction

Rotorcraft play an essential role in the aviation landscape, thanks to their specific maneuvering abilities, which enable them to conduct specific missions that other types of aircraft cannot accomplish. This includes performing Emergency Medical Services (EMS), civil Search & Rescue (SAR), as well as carrying out logistic missions in remote areas. Rotorcraft environmental impact (CO_2 and noise emissions) could be considered negligible compared to that of fixed-wing traffic, as the number of rotorcraft operations is significantly smaller. However, the specific missions of rotorcraft lead them to operate at low altitudes and in close proximity to populated areas. Therefore, rotorcraft environmental impact is significant in terms of noise annoyance. Taking into account the foreseen growth in helicopter operations (Grand View Research, 2022), and even more the emergence of Urban Air Mobility (UAM), that also relies on the use of some kind of rotorcraft, the noise reduction of rotorcraft operations is urgently needed (ICAO, 2008). A way to reduce emitted noise is based on modifying the structural design of rotorcraft (e.g. Rauch et al., 2011), but this usually implies some compromise in vehicle performances, besides requiring a long-time development process. Another promising way to address the mitigation of noise, that we consider in this paper, is the design of noise-abatement operations, and more specifically the design of rotorcraft trajectories whose associated noise impact is minimized.

The problem at hand thus belongs to the framework of aircraft trajectory optimization. The objective function to be minimized is evaluated by computer simulations through an industrial software, that is therefore a *black-box*. The problem we address is then an emerging application in

air transportation, which is challenging as it combines both the difficulties of optimal trajectory design and of black-box optimization.

Aircraft trajectory optimization has been widely addressed in the literature (Betts, 1998; Delahaye et al., 2014). This problem has been studied for different types of aircraft: airplanes (Hagelauer and Mora-Camino, 1998; Sridhar et al., 2011; Rodionova et al., 2014), rotorcraft (Guntzer et al., 2014; Morris et al., 2016) and more recently UAVs (Unmanned Aerial Vehicles) (Raap et al., 2017; Coelho et al., 2017; Dasdemir et al., 2020). Several objectives to be minimized are considered, such as fuel consumption and operational costs (Hagelauer and Mora-Camino, 1998), traffic congestion (Rodionova et al., 2014), contrails formation (Sridhar et al., 2011) and noise annoyance (Prats et al., 2010; Guntzer et al., 2014). The design of rotorcraft trajectories whose associated noise footprint is minimized relies on optimization levers usually including the position and the velocity of the rotorcraft. In the literature, many works focus on optimizing rotorcraft straight approach profiles, such as in (Padula et al., 2009; Guntzer et al., 2014). In these works, the horizontal path is fixed, letting the altitude and speed of the rotorcraft be the only optimization variables. Significant noise reduction (up to 6 dB) have been obtained by Guntzer et al. (2014). However, the increase of operations in (more constrained) urban areas makes it needed to investigate the possibility to modify additionally the horizontal path. More recent research by Greenwood (2019, and references therein) have considered more complex rotorcraft trajectories. The trajectory is optimized in 3D, but it relies on a coarse discretization of the space, with a limited choice for heading changes. Another work by Greenwood (2017) takes into account both time and 3D positioning of the rotorcraft in trajectory modeling. However, only some simplified trajectories (e.g. straight approach, level turn) are optimized. In our work, both the horizontal and vertical paths, along with speed, are degrees of freedom in the optimization. As regards to the solution of aircraft trajectory optimization problems, many different approaches can be found in the literature, including integer linear programming (Raap et al., 2017), optimal control (Prats et al., 2010; Sridhar et al., 2011) and heuristic methods (Rodionova et al., 2014; Dasdemir et al., 2020). For the considered problem of rotorcraft noise abatement trajectory design, population-based heuristic methods are typically used (Padula et al., 2009; Guntzer et al., 2014). However, in the framework of black-box optimization, these methods remain computationally very expensive. As an example, Padula et al. (2009) fix a maximum number of function evaluations to 300 to limit the computing time. More efficient heuristic methods based on space discretization, such as A*, have also been used (Morris et al., 2016; Greenwood, 2019) but without any guarantee of optimality in the continuous space. Additionally, Morris et al. (2016) assume that noise is factorable (i.e. the noise impact associated to two joined segments of trajectory equals the sum of the noise impact of each individual segment), while in practice it is not. The approach that we propose in this paper takes into account the relevant characteristics of the real-world problem at hand while handling the computational complexity of the black-box.

Black-box optimization is receiving an increasing attention for efficiently addressing problems arising in engineering and economy, where the evaluation of the objective and/or the constraints is obtained through a computer simulation (e.g. Audet et al., 2008; Manno et al., 2020). In this context, the use of surrogate models (sometimes referred to as *meta-models*) has proved to be effective in enhancing the performance of black-box optimization, in particular for problems where computer simulation is expensive, and derivatives are costly or not available (Sóbester et al., 2014; Vu et al., 2017; Xia and Shoemaker, 2021).

In this paper, we address the problem of designing minimal-noise rotorcraft trajectories. We propose to design 4D trajectories, where both the rotorcraft 3D position and speed are optimized. The computation of the associated noise footprint, which is minimized, relies on an accurate noise footprint simulator provided by industry. The design of realistic 4D rotorcraft trajectories combined to accurate noise predictions is rarely present in the literature and appears to be of

particular interest in the considered application framework. We propose a solution approach relying on a state-of-the-art method for black-box optimization. Despite the complexity of the problem, we show that we are able to provide good quality (locally) optimal solutions exhibiting significant noise reduction compared to trajectories currently flown in practice.

Another main contribution of this paper relies on the definition of multiple surrogate models of the original black-box simulator, based on our knowledge of the problem. On the one hand, we propose some surrogates that are defined from a simplification of the original black-box, while keeping the black-box nature. They include in particular a surrogate based on the physics (acoustics principles) of the problem. On the other hand, we propose a surrogate relying on a machine learning process, where the knowledge of the problem plays again an important role in its construction. We propose an algorithmic scheme relying on these surrogates and on a state-of-the-art algorithm for black-box problems. Numerical results on real-world problem instances show significant benefits from the proposed approach in terms of both quality of the computed solution and computing time.

The paper is structured as follows. Section 2 presents the mathematical modeling of the problem. Section 3 details the algorithmic scheme proposed to solve the problem. Section 4 introduces the different surrogate models that we propose to enhance the optimization and how to use them in the proposed algorithmic scheme. Section 5 presents and discusses the numerical results obtained with the proposed approaches on real-world instances. Section 6 draws conclusions and gives some perspectives for future work.

2 Modeling the minimum-noise rotorcraft trajectory optimization problem

This section gives first a brief description of the model that we consider for rotorcraft trajectories. Then, it presents the mathematical formulation of the problem at hand, comprising the definition of the decision variables, the objective function and the constraints.

2.1 Trajectory modeling

In this paper, a trajectory is modeled as a succession of *waypoints*. A (flyable) trajectory is determined afterwards, as the result of an interpolation of these points, as depicted in Figure 1. A *waypoint* is generally defined by coordinates in the 3-dimensional space identifying the physical position of the rotorcraft. They are widely used in the aeronautical domain, according to the International Civil Aviation Organization (ICAO, 2006), to define the flight path of a rotorcraft employing area navigation.

In this work, we consider a set $\mathcal{W}$ of waypoints in a 4-dimensional space. Each element w of this set is a quadruplet $(x_w, y_w, z_w, v_w) \in \mathbb{R}^4$, where the first three coordinates, x_w, y_w, and z_w, identify the rotorcraft position in the 3-dimensional space, and the fourth one, v_w, sets its speed. More precisely, v_w denotes the rotorcraft true air speed (TAS), which is defined as the speed of the rotorcraft relative to the air mass in which it is flying. Ground speed (relative to the Earth's surface) is calculated as the sum of the TAS and the wind speed. All speeds are expressed in knots (kt). The number of waypoints $N_{\mathcal{W}} = |\mathcal{W}|$ is a parameter, whose value depends on the problem instance.

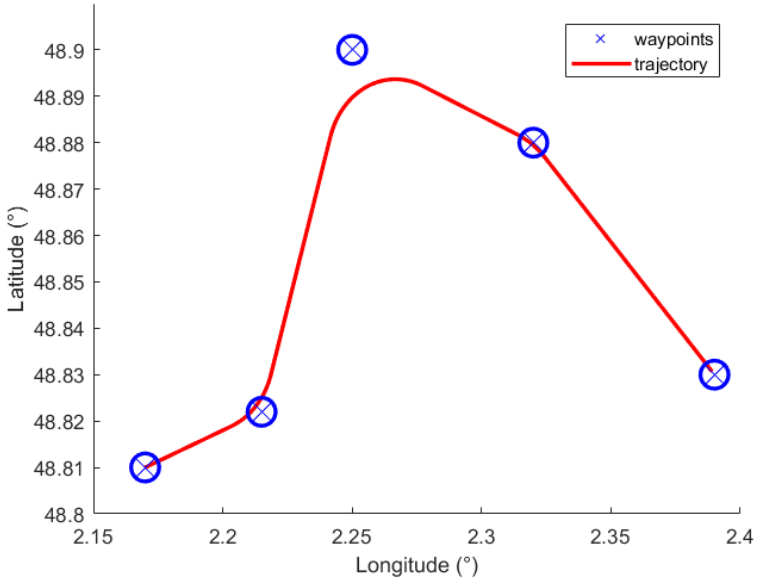


Figure 1: A trajectory (in red) is built by interpolation of waypoints (in blue).

2.2 Optimization problem formulation

The problem of designing rotorcraft trajectories so as to minimize the associated noise footprint takes the general form in (1):

$$\begin{aligned} \min_{\mathbf{x}\in\Omega} \quad & f(\mathbf{x}) \\ \text{subject to} \quad & g_j(\mathbf{x}) \leq 0, \quad j = 1, \ldots, m \end{aligned} \tag{1}$$

As detailed in the following, the vector $\mathbf{x}$ of decision variables follows directly from the trajectory model. The objective function f is derived from typical noise metrics, whose computation is based on noise values resulting from computer simulations through the considered industrial black-box simulator. Finally, the constraints g_j are mainly related to operational requirements.

Decision variables. The decision variables are the degrees of freedom used to minimize the noise impact associated to the trajectory. They derive directly from the trajectory modeling introduced in Section 2.1 and correspond to the 4-dimensional coordinates defining the waypoints that can be optimized. The set of decision variables is defined as follows:

$$\mathbf{x} = \{x_i, y_i, z_i, v_i : i = 1, \ldots, N\} \tag{2}$$

where $N = N_{\mathcal{W}} - 2$ is the number of variable waypoints. Indeed, there are two particular waypoints in the set $\mathcal{W}$: the starting waypoint (x_0, y_0, z_0, v_0) and the end waypoint $(x_{N+1}, y_{N+1}, z_{N+1}, v_{N+1})$. Their coordinates do not belong to $\mathbf{x}$, since they are data of the problem and remain fixed during the optimization.

Constraints. The constraints of the addressed problem are mainly related to operational requirements. The trajectory must respect rotorcraft performance limitations (e.g. maximum

rate of turn), flyability constraints (ensuring pilot and passengers comfort) and Air Traffic Management (ATM) regulations (e.g. obstacle avoidance). All these constraints can be represented through non-linear functions g_j, satisfying:

$$\forall j \in \{1, \ldots, m\}, \quad g_j(\mathbf{x}) \leq 0, \quad \mathbf{x} \in \mathbb{R}^{4N} \tag{3}$$

First, all variables are bounded. The lower and upper bounds for the position variables (x, y, and z) are set in such a way that the variable space domain remains out of obstacles. A minimum speed, v_{min}, is set to avoid hazardous flying conditions at low speed, while a maximum speed, v_{max}, is fixed by rotorcraft performance limitations. Contrary to the bounds on the position variables, those on velocity are the same for all the waypoints:

$$\forall\ i \in \{1, \ldots, N\}, \begin{cases} \underline{x_i} \leq x_i \leq \overline{x_i} \\ \underline{y_i} \leq y_i \leq \overline{y_i} \\ \underline{z_i} \leq z_i \leq \overline{z_i} \\ v_{min} \leq v_i \leq v_{max} \end{cases}$$

Second, the main constraints that characterize the problem involve the rotorcraft flight path angle γ (expressed in degrees, $^\circ$), its vertical speed V_z (expressed in feet per minute, ft.min^{-1}) and its heading Ψ (expressed in degrees, $^\circ$). They are defined in terms of the decision variables as follows:

$$\forall\ i \in \{0, \ldots, N\}, \begin{cases} \gamma_i = \arctan\left(\dfrac{\Delta_{z_i}}{dist_{i,i+1}}\right) \\ V_{z,i} = \sin(\gamma_i)\, v_i \\ \Psi_i = \dfrac{\pi}{2} - \text{arctan2}\left(\dfrac{\mathrm{d}y_i}{\mathrm{d}t}, \dfrac{\mathrm{d}x_i}{\mathrm{d}t}\right) \end{cases}$$

where $\Delta_{z_i} = z_{i+1} - z_i$ is the altitude change and $dist_{i,i+1}$ is the Euclidean ground distance between successive waypoints i and $i+1$.
The flight path angle and the vertical speed are lower bounded by γ_{min} and $V_{z,min}$, respectively, to ensure passenger comfort and to limit pilot workload. They are also upper bounded by respectively γ_{max} to avoid blade stall and $V_{z,max}$ due to rotorcraft performance limitations. Operational limitations also impose additional constraints such as a maximum acceleration, a_{max} (expressed in knots per second, kt.s^{-1}), and a maximum rate of turn, $\dot{\Psi}_{max}$ (expressed in degrees per second, $^\circ$.s^{-1}). The latter represents the maximum number of degrees of heading change allowed per second. Note that in practice most of the turns are performed at a standard rate turn, which corresponds to $\dot{\Psi} = 3^\circ.\text{s}^{-1}$.

The formulation of the above constraints reads:

$$\forall\ i \in \{0, \ldots, N\}, \begin{cases} \gamma_{min} \leq \gamma_i \leq \gamma_{max} \\ V_{z,min} \leq V_{z,i} \leq V_{z,max} \\ \left|\dfrac{\mathrm{d}v_i}{\mathrm{d}t}\right| \leq a_{max} \\ \left|\dfrac{\mathrm{d}\Psi_i}{\mathrm{d}t}\right| \leq \dot{\Psi}_{max} \end{cases}$$

We remark that in practice the lower bound on the vertical speed is the most binding constraint in most cases. Indeed, on the one hand, flying as high as possible seems to be a good way to reduce noise on the ground. This would indeed allow to benefit from a larger attenuation of noise resulting from the increased propagation distance compared to that of a lower altitude

flight. On the other hand, it seems also interesting to fly as fast as possible to reduce the population overflight duration. The combination of these two intuitive ways to reduce noise during the approach phase of the flight would lead to a steep and fast descent with a very low, thus non-flyable, vertical speed.

Finally, obstacles such as forbidden areas, terrain (hills, mountains) and buildings represent also some kind of additional constraints. Indeed, as obstacles have to be avoided to ensure flight safety, the feasible domain is restricted. Unlike the preceding ones, the constraints associated to obstacles are not described by an analytical expression. We instead take them into account through a definition of the feasible space for the decision variables that does not include points of the 3D-space corresponding to obstacles. Further details about the management of the obstacles are given in Section 3.2.2.

Objective function. We address the minimization of the noise perceived on the ground, associated to the rotorcraft trajectory to be designed. Since we consider the optimization of a single trajectory, its associated noise impact is typically measured by single-event noise metrics, such as Perceived Noise Level (PNL), Sound Exposure Level (SEL) or maximum A-weighted sound level ($L_{A,max}$) (Watson and Downey, 2008). These noise metrics are defined taking into account the time evolution of noise at different ground locations. In this paper, we consider the SEL metric, which is widely used in the application framework we address. SEL values are the result of a computationally expensive computer simulation performed by an industrial software from Airbus Helicopters. The objective function to be minimized relies on these SEL values, and is defined as follows.

Let N_m be the number of relevant ground locations, where the noise has been computed. For all $k \in \{1, \ldots, N_m\}$, p_k is a problem data denoting the number of people at ground location k, and SEL(k) denotes the Sound Exposure Level at ground location k, computed through the black-box computer simulation. SEL noise contours are defined as areas where the predicted SEL is higher than given values. Let $\mathcal{L} = \{\mathcal{L}_0, \mathcal{L}_1, \ldots, 100\}$ dB(A) be the set of values defining the noise contours. In this paper, we consider 5 dB(A) contours, hence $\mathcal{L}_j = \mathcal{L}_0 + 5j$ dB(A) for all $j \in \{0, \ldots, |\mathcal{L}|-1\}$, $|\mathcal{L}|$ being the cardinality of the set $\mathcal{L}$. The lowest noise contour threshold, denoted as $\mathcal{L}_0$, is a user-defined parameter typically set to 70 dB(A). The total population considered exposed to noise, P_{tot}, is then defined as:

$$P_{tot} = \sum_{\substack{k \in \{1,\ldots,N_m\}: \\ \mathcal{L}_0 \leq \mathrm{SEL}(k)}} p_k \tag{4}$$

Let $\mathcal{M}_j$ be the subset of ground locations whose SEL value is between $\mathcal{L}_{j-1}$ and $\mathcal{L}_j$:

$$\forall j \in \{1, \ldots, |\mathcal{L}| - 1\}, \quad \mathcal{M}_j = \{k \in \{1, \ldots, N_m\}, \mathcal{L}_{j-1} \leq \mathrm{SEL}(k) < \mathcal{L}_j\},$$

and let P_j denote the number of people associated with the set of ground locations $\mathcal{M}_j$:

$$\forall j \in \{1, \ldots, |\mathcal{L}| - 1\}, \quad P_j = \sum_{k \in \mathcal{M}_j} p_k.$$

The population exposure to rotorcraft noise, denoted as f_{noise}, is defined as a normalized weighted sum of the population in the different SEL contours. It is defined as follows:

$$f_{noise} = \sum_{j=1}^{|\mathcal{L}|-1} \alpha_j \cdot \frac{P_j}{P_{tot}} \tag{5}$$

The weighting factors values, $\alpha_j \geq 0$, are user-defined parameters used to give more or less weight to the different noise exposure levels. In this paper, these weighting factors are set according to dose-response relationships for aircraft noise defined by the World Health Organization (WHO, 2018). Finally, we consider the following objective function f to be minimized, accounting for both distance travelled and population exposure to rotorcraft noise:

$$f = \lambda \cdot f_{dist} + (1 - \lambda) \cdot f_{noise} \tag{6}$$

where $\lambda \in [0, 1)$ is a user-defined weighting parameter, that can be adjusted to reach a good compromise between trajectory length and population exposure to noise, and f_{dist} is a normalized measure of the trajectory length.

The two terms, f_{dist} and f_{noise}, of the objective function depend on the decision variables x, y, z and v defined above. The length of the trajectory is directly related to the 3D position of the waypoints, which is defined through the variables x, y and z. As for the f_{noise} term, first remark that the population exposure to noise strongly depends on the population overflown by the horizontal path of the rotorcraft, defined through variables x and y. In addition, the SEL perceived on the ground closely depends on the propagation distance between the noise source and the receiver. The height of the rotorcraft, controlled by variable z, thus plays a major role: the higher the rotorcraft, the lower the noise level on the ground. The rotorcraft speed, controlled by variable v, also has a strong influence on the objective function. First, the noise emitted by the rotorcraft depends on its speed. In addition, as the SEL is a time-integrated metric, the speed also has a strong influence on the objective function: the faster the rotorcraft, the lower the population overflight duration, hence the lower the SEL value.

As the computation of the objective function values relies on an industrial black-box simulator, we do not have access to an analytical expression of f_{noise}, nor to its derivatives in function of the decision variables. Derivatives can neither be approximated through finite differences due to the computational cost involved. We consequently use solution algorithms for black-box optimization, which do not rely on gradient information to compute descent directions.

3 Optimizing rotorcraft trajectories

In this section, we first present methods typically used for the resolution of black-box optimization (BBO) problems. Then, we describe the chosen method and the dedicated algorithmic scheme proposed to solve the problem at hand.

3.1 Literature review on BBO methods

Computational approaches to solve BBO problems are based on methods that do not need any gradient information. Such methods can be classified in different categories.

First, meta-heuristic methods, and especially population-based algorithms, are widely used to solve BBO problems (Padula et al., 2009; Guntzer et al., 2014). Such methods manage a collection of individual candidate solutions, forming what is called a population. They iteratively update the population keeping the most promising individuals and generating new individuals. To proceed this way, these methods evaluate the black-box at each iteration on each individual of the population, and as a consequence they are generally computationally expensive. The computational cost of a single run of the black-box simulator that we use is already too high to consider such population-based methods to solve our problem.

Statistical-based methods can also be used to address black-box optimization problems (Jones et al., 1998). Among them, Bayesian Optimization (Mockus, 1989; Frazier, 2018) is a class of

global optimization methods, which is particularly adapted to problems with expensive black-box functions. It benefits from the use of a statistical model, typically a Gaussian Process. However, statistical-based methods, as well as population-based ones mentioned above, do not provide any guarantee of optimality for the computed solutions.

In this paper, we focus on deterministic derivative-free optimization (DFO) methods. They can be essentially divided into two major categories: *model-based* and *direct-search* (Audet and Hare, 2017; Larson et al., 2019).

On the one hand, model-based methods typically use a model as a surrogate of the objective function, to determine candidate solutions and guide the optimization. Polynomials or radial basis functions are widely used surrogate models in the literature (Gutmann, 2001; Costa and Nannicini, 2018; Xia and Shoemaker, 2021). These models can then be used within several methods. As an example, in line-search methods, the descent directions are determined according to the gradient information of the model. In trust-region methods, the next iterate is determined by minimizing the model over a *trusted* region, where it is assumed to be a faithful approximation of the objective function. These model-based methods are particularly appropriate when the objective function is assumed to be smooth, that cannot be intuitively stated for our problem.

On the other hand, direct-search methods only rely on objective function evaluations of candidate points (solutions). At each iteration k, these methods generate a set of *poll* points around the current candidate point $\mathbf{x}_k$ according to a finite set of predefined directions of search, denoted as $\mathbf{D}_k$. Given a step size α_k and a *polling* direction $\mathbf{d} \in \mathbf{D}_k$, a poll point is obtained as $\mathbf{x}_k + \alpha_k \mathbf{d}$. The objective function f is then evaluated at the poll points in order to determine the next candidate point $\mathbf{x}_{k+1}$. If a poll point providing a lower (in case of a minimization problem) value of the objective function is found, $\mathbf{x}_{k+1}$ is set to that point and the step size is possibly increased. Otherwise, $\mathbf{x}_{k+1}$ is set to $\mathbf{x}_k$ and the step size is decreased. This local exploration around the current candidate point is called the *poll step*. Most of direct-search methods also include a *search step*, which allows for a wider exploration in the variables space. This step is very important in practice as it can guide the method to promising regions and help to escape local solutions. In the *search step*, the objective function is evaluated at any finite set of points $\mathbf{T}_k$ that can be generated by any user-defined procedure. If this step fails to find a better candidate solution, then the *poll step* is performed. A general description of direct-search methods is given in Algorithm 1.

There are in the literature several direct-search methods that mainly differ from each other on how to generate the set of polling directions. One of the first approaches is the Coordinate Search (CS) (Fermi and Metropolis, 1952), for which the set $\mathbf{D}_k$ of polling directions is composed of the positive and negative coordinate directions. The pattern-search methods advance the CS in that they include a *search step*, as described above. In addition, they allow for a broader finite set of directions than the coordinate ones in the *poll step*. The Brute Force Optimizer (BFO), introduced by Porcelli and Toint (2018) is an example of a recent algorithm implementing a pattern-search method. The Mesh Adaptive Direct Search (MADS) method (Audet and Dennis (2006)) further improves pattern-search methods in that it allows for an infinite set of polling directions whose union across the iterations is asymptotically dense in the search space. In addition, convergence to a locally optimal solution can be proved for both unconstrained and constrained problems.

Some more recent works (Conn and Le Digabel, 2013; Manno et al., 2020) propose methods that cannot be clearly classified as direct-search or model-based methods as they combine some characteristics of both. Conn and Le Digabel (2013) introduced an adaptation of the MADS algorithm using quadratic models. They propose to use the models whether directly in the search step or for ordering the candidate points in both the search and poll steps. Another method, RQLIF, is defined by Manno et al. (2020) as an hybrid method which consists of the

Algorithm 1 Direct-search method

Choose an initial point $\mathbf{x}_0$ and a step size α_0
$\mathbf{x}_0^+ \leftarrow \mathbf{x}_0, \quad k \leftarrow 0$
while stopping criterion not reached **do**
  Generate a finite set $\mathbf{T}_k$ ▷ *search step*
  if $f(\mathbf{t}) < f(\mathbf{x}_k)$ for some $\mathbf{t} \in \mathbf{T}_k$ **then**
    $\mathbf{x}_k^+ \leftarrow \mathbf{t}$
  end if
  if $\mathbf{x}_k^+ = \mathbf{x}_k$ **then**
    Generate the set of polling directions $\mathbf{D}_k$ ▷ *poll step*
    if $f(\mathbf{t}) < f(\mathbf{x}_k)$ for some $\mathbf{t} \in \{\mathbf{x}_k + \alpha_k \mathbf{d} : \mathbf{d} \in \mathbf{D}_k\}$ **then**
      $\mathbf{x}_k^+ \leftarrow \mathbf{t}$
    end if
  end if
  if $\mathbf{x}_k^+ = \mathbf{x}_k$ **then** ▷ *update*
    Decrease(α_k)
  else
    Increase(α_k)
  end if
  $\mathbf{x}_{k+1} \leftarrow \mathbf{x}_k^+, \quad k \leftarrow k+1$
end while

alternation of direct-search and model-based steps.

In this paper, we consider direct-search methods, and in particular the MADS method. It features indeed some characteristics that makes it particularly appropriate for the problem we address. First, it uses an infinite set of polling directions, which can be a key element to promote efficiency in the presence of multiple local optima. Furthermore, it provides a guarantee of local convergence to an optimal solution for constrained optimization, where the constraints are handled through a penalization approach. Finally, its performance on the problem considered in this paper have been compared, using the NOMAD solver, against a well-known pattern search algorithm, BFO (Porcelli and Toint, 2018) and the above-mentioned hybrid method RQLIF (Manno et al., 2020). The results presented in Section 5.3 show the interest in the use of MADS for our problem.

3.2 Solution approach

In this section, we present the algorithmic scheme that we propose to solve the problem of designing a minimal-noise rotorcraft trajectory. Our approach is based on a local direct-search method, and specifically on MADS. In order to enhance the performance of MADS and more generally of any local direct-search method, we rely on different levers, which include the computation of a suitable starting guess for the optimization, the reduction of the space of variables and the use of surrogate models. The two former levers are introduced in subsections 3.2.1 and 3.2.2, while different types of surrogates, appropriate to the problem at hand, are discussed in Section 4. The proposed algorithmic scheme is sketched in Algorithm 2.

Algorithm 2

1: Compute an **initial solution $\mathbf{X_0}$** through a dedicated path planning algorithm **(FMT*)**.
▷ See Subsection 3.2.1
2: Define an appropriate **variable space domain** according to obstacles.
▷ See Subsection 3.2.2
3: **while stopping** criterion not reached **do**
4: Run one iteration (*search* and *poll* steps) of the **MADS** algorithm.
▷ See Algorithm 1
where all function evaluations are performed by the **black-box**.
5: **end while**
6: Return a (locally) optimal trajectory $\mathbf{X_{opt}}$.

3.2.1 Providing a good starting point

It is widely known that for local optimization methods, the computed optimal solution is closely dependent on the choice of the initial candidate solution. Therefore, it is important to provide such methods with a good starting guess. In the considered context, the aim is to define an initial trajectory, referred to as $\mathbf{X_0}$ in the following, which is feasible with respect to the problem constraints. Furthermore, one expects the design of such initial trajectory being not time consuming within the whole trajectory optimization process.

In this paper, the initial trajectory is determined by solving an auxiliary optimization problem, that differs from the one defined in Section 2 on two main points. First, in order to save computing time, the criterion to be minimized does not rely on the costly black-box. Its expression is similar to the one given in Equation (6), except that f_{noise} is replaced by a measure of the population overflown, as defined in the following. Second, the auxiliary problem also differs from the one defined in Section 2 in that it is a path planning problem, where the decision variables are only the 3-dimensional x, y and z variables identifying the rotorcraft position. There are no variables depending on time, as it is the case in trajectory optimization problems. A pre-defined speed profile is assigned to the obtained solution afterwards to get the initial trajectory design.

This auxiliary problem is solved using a sampling-based path planning algorithm: the Fast Marching Tree algorithm (FMT*) (Janson et al., 2015). The authors (Janson et al.) show that FMT* is asymptotically optimal and outperforms state-of-the-art path planning algorithms such as RRT* and PRM* (Karaman and Frazzoli, 2011). Like other sampling-based path planning algorithms, FMT* performs a dynamic programming recursion: it generates a tree in the search space and builds an optimal path on this tree, from the starting to the destination point. Its efficiency is due to several reasons. First, it operates in a pre-sampled search space. Therefore, it can perform concurrently the graph construction and the search. At each iteration, the tree is extended towards the destination point by building the locally optimal connection (i.e. with the minimal cost) between one of the neighbor points (i.e., which have not been connected yet to the tree) of the current most promising node (tree leaf) and one of its neighbor node. FMT* takes also advantage, as detailed in the following paragraph, of a restricted selection of neighbors within a given radius r. In addition, FMT* performs a "lazy" computation in that it ignores the obstacles for the evaluation of the potential connections. Then, if the locally optimal connection crosses an obstacle, it is simply skipped instead of looking for other connections to extend the tree. More details about the algorithm are given in (Janson et al., 2015) and an illustration of a generated tree is given in Figure 2a.

In this paper, FMT* has been tailored to our problem. As the aim is to build a flight path, the starting point and the destination point are obviously respectively the starting waypoint and

the end waypoint of the trajectory. As regards the neighbors selection of a given node, FMT* considers that the neighbor is defined as a sphere centered on that node. The value of the sphere radius, r, depends on the predetermined number of sampled points, that in turns depends on the granularity of the search space discretization. The higher the number of sampled points, the closest these points each other, the smaller the radius r. In this paper, some points located in that sphere are omitted when looking for locally optimal connections because the resulting connections would violate the flyability constraints (e.g., maximum turn rate, maximum flight path angle).

The function c, evaluating the cost of a potential connection between a node of the tree $\mathbf{u}$ and one of its neighbors $\mathbf{v}$, is similar to the objective function introduced in Equation (6), but does not relies on the black-box simulator. Its formulation is the following:

$$c(\mathbf{u}, \mathbf{v}) = \lambda \cdot \frac{dist(\mathbf{u}, \mathbf{v})}{r} + (1 - \lambda) \cdot \frac{pop(\mathbf{u}, \mathbf{v})}{p(\mathbf{u}, r)}$$

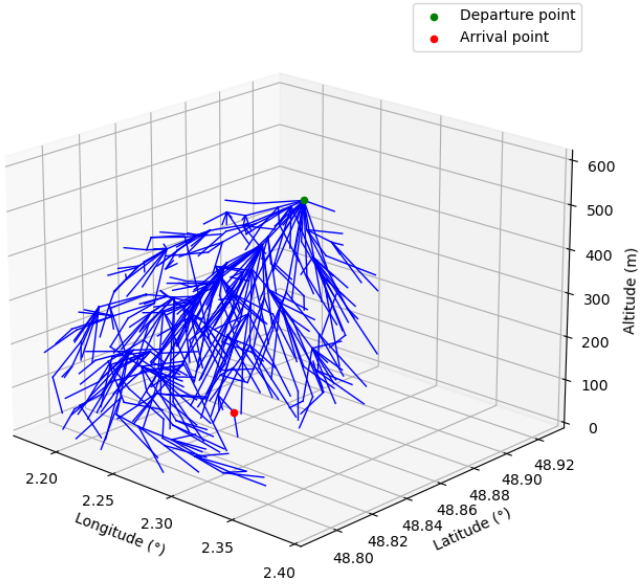
where $\lambda \in [0, 1]$ is an user-defined parameter, $dist(\mathbf{u}, \mathbf{v})$ is the Euclidean distance between $\mathbf{u}$ and $\mathbf{v}$, $p(\mathbf{u}, r)$ is the population located in the r-radius sphere centered on the tree node $\mathbf{u}$, and $pop(\mathbf{u}, \mathbf{v})$ is the population overflown from $\mathbf{u}$ to $\mathbf{v}$ as defined hereafter. Population data is available in a grid format, and each grid cell is associated with the number of people lying in it. The value of $pop(\mathbf{u}, \mathbf{v})$ is computed summing up the number of people in every grid cell crossed by the straight line between $\mathbf{u}$ and $\mathbf{v}$ and in all the surrounding cells lying within a predetermined noise spanning distance value. We typically consider a noise spanning distance of 1000 m.

The weighting factor λ allows the user to set the expected balance between the reduction of the population overflown and the increase in the travelling distance. The role of λ is illustrated in Figure 2b. For low values (e.g. $\lambda = 0.1$), the focus is put on reducing population overflown, which typically results in an increased path length due to the detours taken to avoid high-densely populated areas (see the blue path in Figure 2b). On the contrary, for high values ($\lambda = 0.9$), the travelling distance takes a greater importance and the resulting path is more straight to the destination (see the black path in Figure 2b).

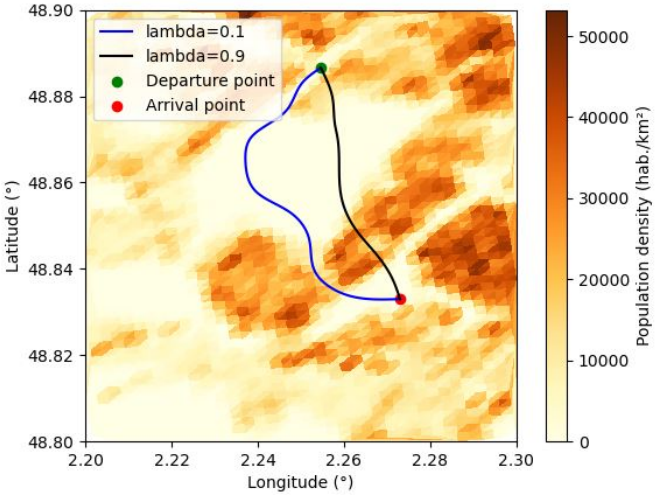
3.2.2 Reducing the variables space domain

The second lever, enhancing the efficiency of the problem resolution, consists in reducing the variable space domain. The interest of this lever is twofold. On the one hand, the reduction of the space of variables usually helps any optimization method to converge faster. On the other hand, the constraint related to obstacle avoidance is handled directly through the use of this restricted space of variables, which will be referred to as *corridor* in the following.

The corridor is defined as an area around the initial trajectory computed through the algorithm presented in the previous subsection. It must be as large as possible to allow variables to vary (within their bound-constrained domain) along the optimization process, but always keeping the associated flight path collision-free with respect to the obstacles. In practice, the corridor is built piecewise linearly from the starting waypoint to the end waypoint of the trajectory. It connects pairwise special points that can whether be defined according to the lower and upper bounds of each decision variable x and y representing the rotorcraft position, or added around the obstacles in order to get collision-free connections between two successive points. An illustration of a corridor is given in dashed black in Figure 3.



(a) Illustration of some connections of the tree of paths generated by the FMT* algorithm.



(b) Optimized path (2D top view) over population data without obstacles for different values of the weighting parameter λ.

Figure 2: FMT* algorithm representation on a given instance of our problem.

4 Surrogates

In this section, we first introduce the surrogate models that we propose for the considered black-box. Then, we propose an effective algorithmic scheme, based on Algorithm 2 and relying on the use of these surrogates within a direct-search method.

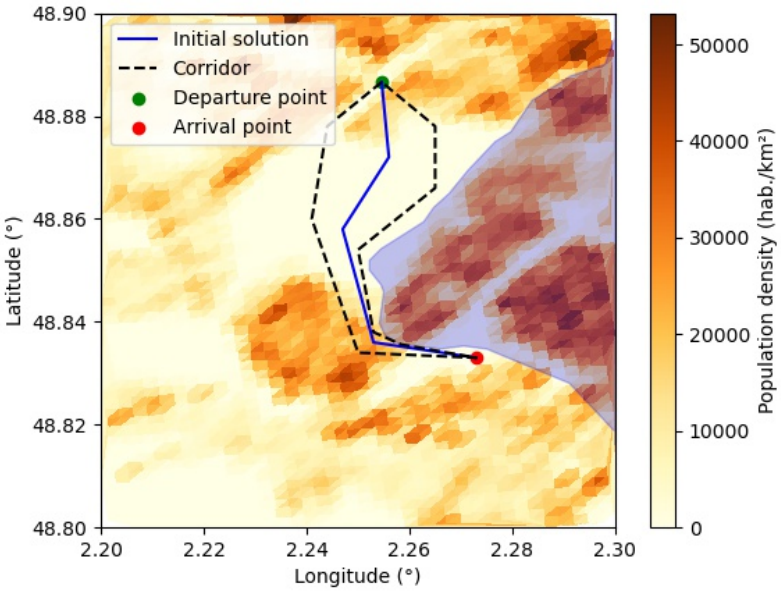


Figure 3: Illustration of a corridor (black dashed lines) around an initial trajectory, avoiding obstacles. In this example, there is only one obstacle, represented in blue, corresponding to a prohibited area (no-fly zone).

4.1 Designing surrogates

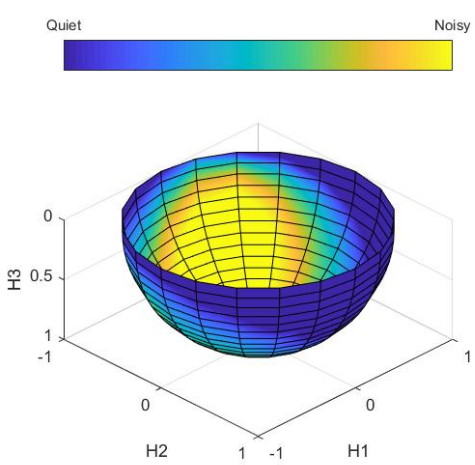
The goal is to design surrogates of the original black-box function that are cheaper to evaluate. We build on our knowledge of the addressed problem to propose different types of surrogates tailored to such a problem. First, we introduce surrogates which are based on different simplifications of the original black-box. Then, we present another type of surrogate model based on machine learning.

4.1.1 Physics-based surrogates

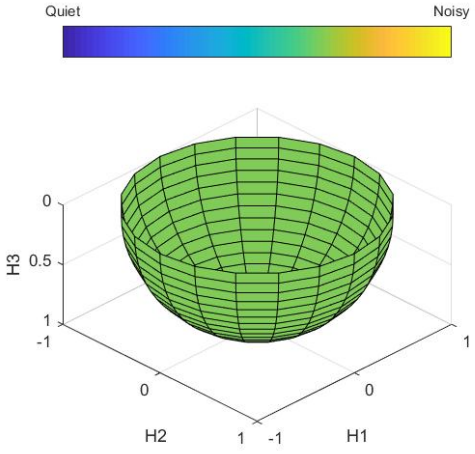
We present in the following two surrogates that are based on simplifications of the original black-box. While keeping their black-box nature, these surrogates are less costly to evaluate than the original function, thanks to their reduced complexity.

The first surrogate that we propose relies on an approximation of the physical noise model on which the noise footprint computation, thus the black-box, is based. It is widely agreed in the literature (Schmitz, 1995; Gopalan et al., 2003; Guntzer et al., 2014) that for steady-state flight and moderate acceleration/deceleration, the noise emitted by a rotorcraft can be parametrized by two flight parameters: the true air speed v and the flight path angle γ. In addition, due to their rotary parts (e.g. the main and tail rotors for an helicopter), rotorcraft do not emit the same noise in all directions, they have their own sound directivity patterns. As a consequence, for a given flight condition determined by the couple (v,γ), the rotorcraft noise source is typically represented by an hemisphere of noise levels centered on the rotorcraft (Padula et al., 2009; Guntzer et al., 2014; Morris et al., 2016). More details about the rotorcraft noise modeling

approach used by the original black-box simulator considered in this paper can be found in (Guntzer et al., 2014; Dieumegard et al., 2022). The approximation that we propose consists in considering that, rather than presenting some sound directivity patterns, such as the hemisphere represented in Figure 4a, the helicopter noise source is isotropic. In this case, the total emitted noise is equally distributed over the whole hemisphere, as illustrated in Figure 4b.



(a) Realistic rotorcraft noise source presenting specific sound directivity patterns.



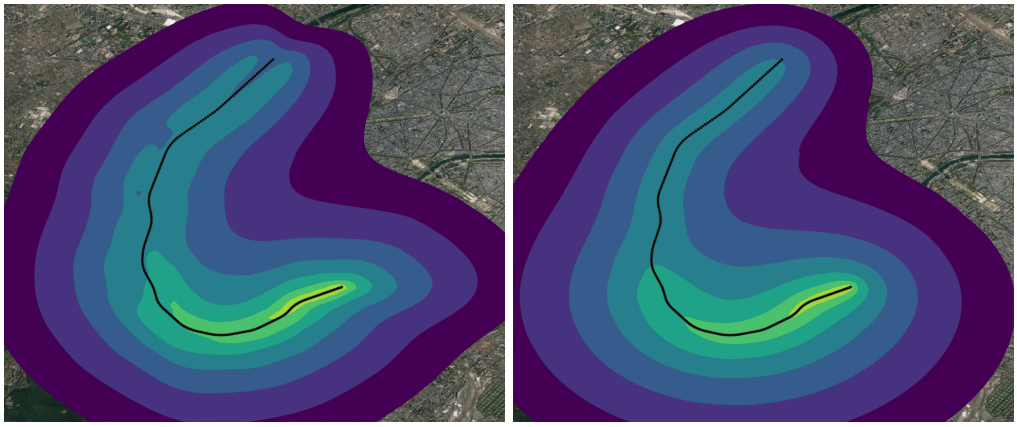
(b) Isotropic approximation of the rotorcraft noise source. The emitted noise is equally spread in all directions.

Figure 4: Definition of the surrogate model considering an approximation of the noise source as isotropic.

The noise footprint associated to a given representative trajectory, resulting from this surrogate model, is shown in Figure 5b. We can see that the noise contours are smoother than those resulting from the computation with the original black-box simulator (Figure 5a). However, the general pattern remains the same, as expected since we chose to keep constant the total noise emitted.

The second surrogate that we propose is based on a simplification of the black-box-based noise footprint computation. The noise footprint results from the computation of the time evolution of the noise at different ground locations, which are typically placed on a regular grid with a user-defined resolution. The finer the grid, the more accurate the noise footprint but the higher the computational cost. The simplification that we propose consists in evaluating the noise on a coarse grid. As a consequence, the computation may not accurately capture the rotorcraft noise footprint, especially in areas where the noise is expected to vary significantly, e.g. in the vicinity of the landing point.

An example of the noise footprint computed for the same trajectory on a fine (Fig. 6a) and on a coarse (Fig. 6b) grid is given in Figure 6. The color scale is the same than in Figure 5: the yellower the contour, the noisier. We can observe some differences in the noise contours in Fig. 6a and Fig. 6b, in particular nearby the end of the trajectory: the yellowest contour is not captured in Figure 6b. Yet, the degraded noise footprint evaluated on a coarse grid keeps an overall shape similar to the more accurate one. In addition, its computing time is significantly



(a) Noise footprint computed by the original black-box simulator.

(b) Noise footprint computed by the isotropic surrogate model.

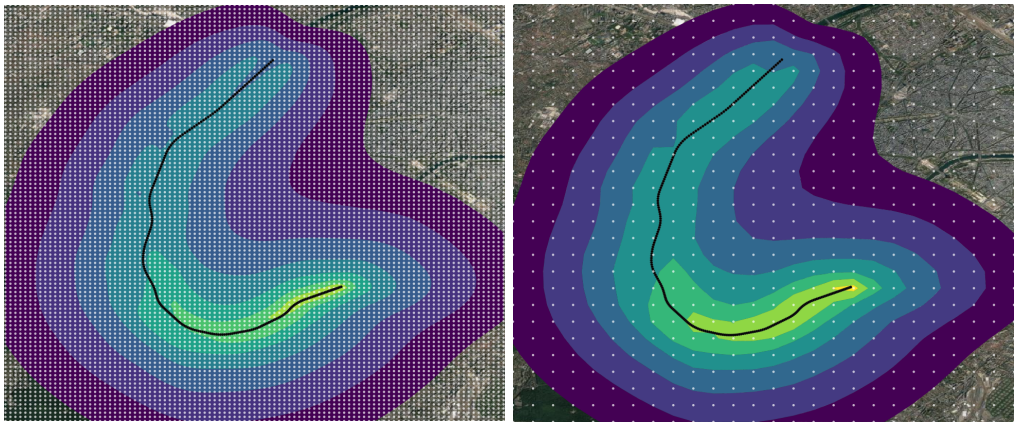
Figure 5: Comparison of the noise footprints, represented by 5 dB(A) SEL contours, associated to the same trajectory evaluated by the original black-box simulator and with the isotropic surrogate model.

reduced. More details about the computing time savings are given in Section 5.3.

4.1.2 Machine-Learning based surrogate

Contrary to the above presented ones, the surrogate model detailed in this section is not based on a simplification of the original black-box. We propose an application of a widely used machine learning algorithm to build a surrogate model tailored to our problem. Machine learning techniques are more and more used to generate surrogates, in particular in applications where the problem relies on the computation of costly black-boxes (Mengistu and Ghaly, 2008; Kocuk et al., 2015; Wang et al., 2019). Furthermore, previous works (Gervais and Schmitz, 2005; Greenwood, 2018) have shown the interest of using machine learning for the prediction of rotorcraft noise.

In this paper, the aim is to learn how the black-box works, i.e. to learn the rotorcraft noise emission and noise propagation models. The learning consists in finding a relationship between some inputs and a single output. In order to simplify the process, we consider that the rotorcraft emits noise at given time instants. Thus, the trajectory is sampled in successive emission instants (typically every second) and the noise is estimated at each prescribed ground location for each emission instant. The inputs of the learning process, which will be referred to as *input features*, are the flight parameters that affect rotorcraft noise emission (i.e. true air speed, v, and flight path angle, γ), the direction and the length of the acoustic ray between the rotorcraft and the position on the ground where the noise is to be estimated. Since the noise source is represented as an hemisphere centered on the rotorcraft (see Fig. 4a), the direction of the acoustic ray (i.e. of the noise propagation) is given by two angles (θ, ϕ). These two angles describe the coordinates of the intersection point of the acoustic ray with the noise hemisphere. The length of the acoustic ray (i.e. the propagation distance) is denoted as *dObs*. All the input features are illustrated in Figure 7. The output is a single value representing the predicted noise level for a given ground location and emission instant. Data considered for the machine learning process has thus a simple shape: a one-dimensional array gathers all the input features and a single value represents the output. Feed-forward artificial neural networks (multi-layer perceptrons) (Hertz et al., 1991;



(a) Noise footprint computed by the original black-box simulator on a *fine* observer grid (∼100m step).

(b) Noise footprint computed by the original black-box simulator on a *coarse* observer grid (∼500m step).

Figure 6: Comparison of the noise footprints, represented by 5 dB(A) SEL contours, associated to the same trajectory evaluated on the original black-box simulator with different resolutions of the observer grid.

Haykin, 2009), like the one chosen to address the considered learning problem, are widely used for similarly-shaped data.

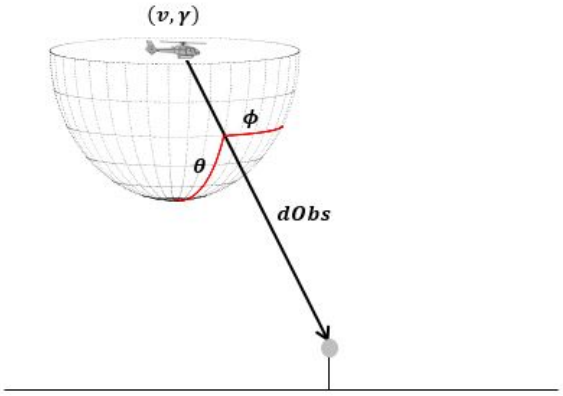


Figure 7: Illustration of the input features for our machine learning-based surrogate model.

A neural network, like the one that we build, is a collection of connected nodes, which are typically organized in layers. The *input layer* gathers all the input features and the *output layer* provides the final output of the network (a noise level in our case). In between, there is at least one *hidden layer*. Each node of a hidden layer behaves like a *neuron*: it receives a signal (i.e. a processed information) from the nodes of the preceding layer, processes and transmits it to those of the following layer. The hidden layers are fully connected, meaning that each node is connected to every node of the following layer, with a specific connection weight. The processing

of each node consists in the computation of the weighted sum of all the input signals coming from the nodes of the preceding layer. Then, a (non-linear) *activation function* is applied to this weighted sum and the result is transmitted to every neuron of the following layer. The learning process, i.e. the *training* of the neural network, consists in solving another optimization problem. This problem aims at minimizing a *loss function*, which generally measures the error between the output predicted by the network and a target output. This is done by setting the weights of the neuron connections.

In this paper, the considered loss function (MSE) is the mean sum of squares of the difference between the predicted and the target noise level as defined below. This function has been chosen in order to have a strong penalization of large errors:

$$MSE = \frac{1}{n}\sum_{k=1}^{n}(L_k - \tilde{L_k})^2$$

where n is the number of examples considered for the evaluation of the loss function, L_k is the target noise value computed by the original black-box and $\tilde{L_k}$ is the noise value predicted by the neural network.

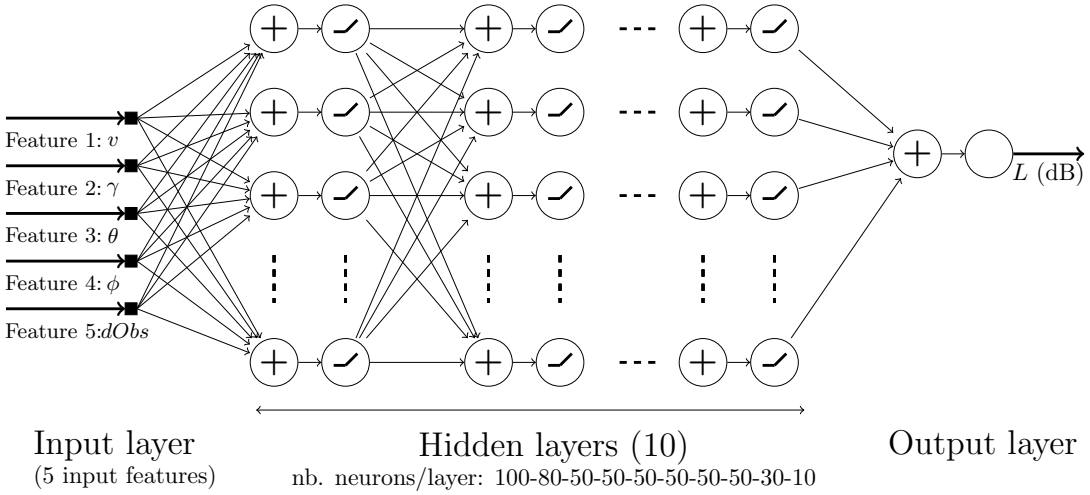


Figure 8: Feed-forward neural network architecture for our problem.

The architecture of the neural network that we build to predict rotorcraft noise is represented in Figure 8. It is known that the prediction accuracy of a neural network depends on several *hyperparameters*, such as the number of hidden layers, the number of neurons per layer, and the activation function, whose values are fixed at the beginning of the learning process. We have carried out some parametric studies to investigate the influence of the values of these hyperparameters, assuming that a prediction accuracy of approximately $RMSE = \sqrt{MSE} =$ 1 dB is very acceptable as it is below typical noise measurement uncertainty and because a 1 dB sound variation is almost imperceptible for the human ear. As a result, we set the number of hidden layers of the multi-layer perceptron to 10, with a decreasing number of neurons per layer, as detailed in Figure 8. For every neuron, the same activation function is applied: the Rectified Linear Unit (ReLu) function, which is defined by $a : x \rightarrow \max(0, x)$ and widely used for multi-layer perceptrons. It is particularly appropriate for the output layer, where it is expected to behave like a linear activation ($a(x) = x$), since the final output is a noise level, positive by

definition. Finally, the optimization algorithm chosen to minimize the considered loss function (MSE) is a stochastic gradient method (Kiefer and Wolfowitz, 1952), which is typically used in the case of large datasets.

In this paper, the dataset is composed of fifty million examples, which have been generated according to specific probabilistic distributions for each input feature. These distributions have been defined according to real flight data recordings. Direction angles ($\theta \in [0, 90]^\circ$ and $\phi \in [0, 360]^\circ$) follow uniform distributions within their whole range since we want the model to be able to capture all the sound directivities of the rotorcraft noise source. The propagation distance ($dObs \in [15, 3000]m$) also follows a uniform distribution. We do not consider propagation distances below 15m because the assumption of the rotorcraft as a point source is not valid for distances smaller than 15m. Above 3000m, it is assumed that the noise attenuation is such that on the ground the rotorcraft noise is not perceptible. Noise-dependent flight parameters ($v \in [0, 130]$ kt, $\gamma \in [-15, 15]^\circ$) follow a normal distribution for each of three different flight phases (take-off/climb, cruise and approach). Mean and standard deviation for the different phases have been estimated from real flight data. All the examples in the dataset have been associated to a noise value, calculated offline using the original black-box simulator. The dataset has been split into three distinct subsets. The training set used to fit the model comprises 64% of the examples. A validation dataset, comprising 16% of the examples, is used for the evaluation of the model at different phases of the training process. Finally, the test set, comprising 20% of the examples, is used for the evaluation of the final model trained on the training set.

The neural network has been developed using the Keras Python library (Chollet et al., 2015), which runs on top of TensorFlow backend. The CPU time required for training the neural network is 41595 seconds (11 hours 33 minutes). This seems relatively long compared to the running of the original black-box, but it is done offline only once. Afterwards, evaluating a trajectory through the neural network based surrogate model is very fast ($\sim$0.5 seconds). After the training, the model reaches a root mean squared error of 0.27 dB on the test set, which is very accurate when considering noise levels. A comparison between the noise footprints computed by this surrogate model and by the original black-box is given in Figure 9 for a representative problem instance. The figure shows that the absolute difference in SEL levels is lower than 0.5 dB, which is much lower than typical noise measurement uncertainty as mentioned previously. This fully validates the accuracy of the proposed neural network model.

4.2 Algorithmic scheme using surrogates

In this section, we propose an optimization algorithmic scheme for the problem at hand, embedding the above defined surrogates in order to enhance the efficiency of MADS. We recall (see Section 3.1) that NOMAD proposes several ways to embed surrogate models in the implemented MADS algorithm (Conn and Le Digabel, 2013). The surrogate can be used either in the *search step* or in the *poll step* to order the candidate points. These candidate points are then evaluated by the original black-box simulator in an opportunistic way, i.e., the most promising one is evaluated first and so on, according to their ranking, until a candidate point improving the objective function value is found. Some points are thus evaluated twice (first with the surrogate model and then with the original black-box).

In this paper, we propose a different framework to use surrogate models. The surrogate is called for the evaluation of the candidate points during both the *search* and *poll* steps. Moreover, we resort to the surrogate during a first phase of the optimization process, while, to refine the search until reaching the local optimal solution, we propose to *switch* from the surrogate model to the original black-box. This allows the use of the more accurate noise footprint computation in the second phase of the optimization process, to get the optimal solution. A sketch of the

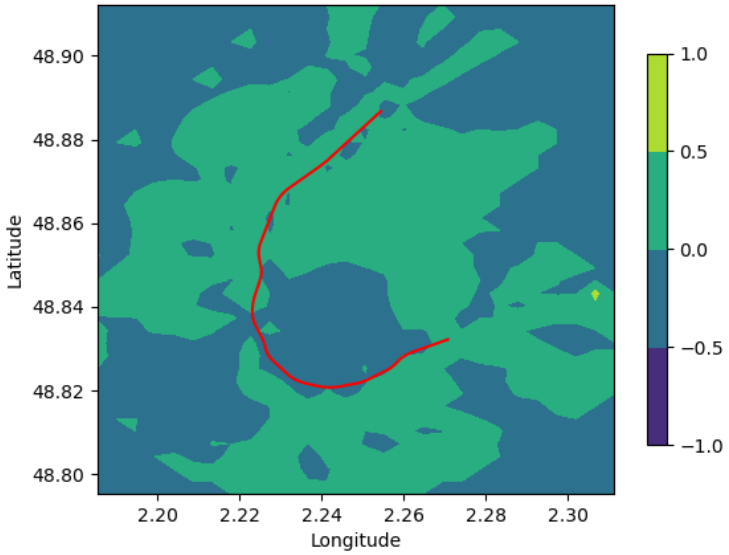


Figure 9: Comparison between the noise footprint computed by the original black-box and by the neural network based surrogate model for the same trajectory (plotted in red): absolute difference of SEL values.

proposed algorithmic scheme embedding surrogates is given in Algorithm 3.

In this algorithmic scheme, it is crucial to decide when the algorithm has to switch between the surrogate model and the original black-box. This choice results from a compromise. In the case of early switching, one would loose the advantage of using a surrogate, as the computing time reduction would not be significant. In the case of late switching, the algorithm could get stuck in a local minimum specific to the surrogate. This might lead to a degraded (local) solution, or could require a high number of iterations with the original black-box, subsequent to the switch, to escape from that local minimum, loosing the benefit of using the surrogate. In this work, the switch is done as soon as one iteration of the MADS algorithm (in the first phase, using the surrogate) has failed, i.e., no improving solution has been found neither in the *search* nor in the *poll* step. The final stopping criterion (of the second phase, using the original black box simulator) is similar to the switching one, apart from the number of failed iterations: the MADS algorithm stops as soon as it reaches n consecutive failed iterations, n being the number of decision variables.

Another important algorithmic choice occurs after the switch, regarding the mesh parameters. On the one hand, they could keep their current values. It means that, after the switch, the search continues on the same mesh but using the original black-box simulator. On the other hand, after the switch, the search could be relaunched on a completely new, reinitialized mesh. This choice is discussed in Section 5.3.

Algorithm 3

Compute an **initial solution $\mathbf{X_0}$** through a dedicated path planning algorithm **(FMT*)**.
▷ See subsection 3.2.1
Define an appropriate **variable space domain** according to obstacles.
▷ See subsection 3.2.2
while *switch* criterion not satisfied **do**
    Run one iteration (*search* and *poll* steps) of the **MADS** algorithm,
    ▷ See Algorithm 1
    where all function evaluations are performed using a **surrogate model**.
    ▷ See subsection 4.1
end while
while stopping criterion not satisfied **do**
    Run one iteration (*search* and *poll* steps) of the **MADS** algorithm,
    ▷ See Algorithm 1
    where all function evaluations are performed using the **original black-box**.
end while
Return a (locally) optimal trajectory $\mathbf{X_{opt}}$.

5 Numerical results

In this section, we first present the computational setting and the main characteristics of the considered problem instances used for our numerical experiments, then numerical results are presented and discussed.

5.1 Computational setting

We first give the numerical values of the parameters appearing in the constraints formulation. These values depend on the type of considered rotorcraft. Each type of rotorcraft has indeed different performance limitations and has its own acoustic signature. In this paper, numerical tests are run for a single type of rotorcraft: a H130 helicopter. For this particular rotorcraft, minimum and maximum speed values are imposed by its performance limitations:

$$v_{min} = 40 \text{ kt} \quad \text{and} \quad v_{max} = 125 \text{ kt}$$

In addition, we assume that the rotorcraft is operated under Visual Flight Rules (VFR), which means that the safety of the flight relies on visual reference rather than on navigation instruments (Instrument Flight Rules (IFR)). In this context, the vertical speed is limited to:

$$V_{z,min} = -1200 \text{ ft.min}^{-1}$$

Finally, to ensure passenger comfort and manageable pilot workload, the rate of turn and the maximum acceleration are bounded by the following values:

$$\dot{\Psi}_{max} = 9 \text{ }^{\circ}.\text{s}^{-1} \quad \text{and} \quad a_{max} = 3 \text{ kt.s}^{-1}$$

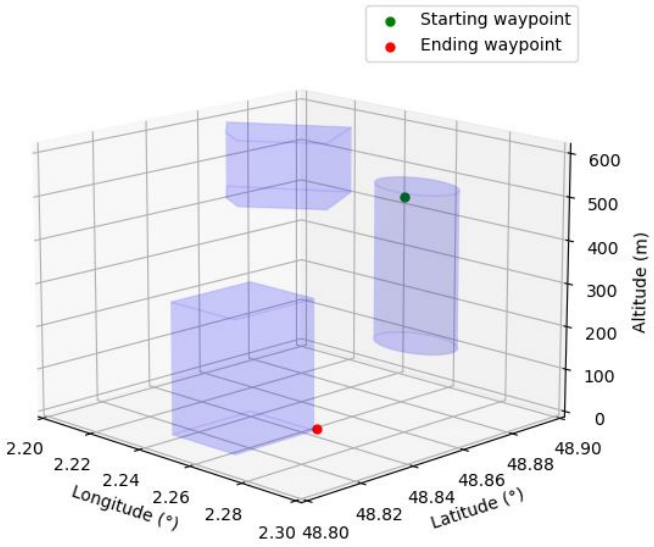
Contrary to the previous ones, these last two values are applicable to any instance, since they do not depend on the rotorcraft type.

5.2 Problem instances

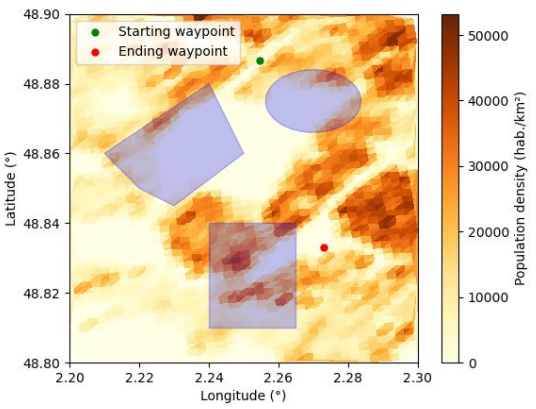
Most of the instances considered in the following correspond to real-world rotorcraft trajectory problems. For each of these instances, the optimal trajectories computed through the proposed approaches (embedding or not the surrogate models) are compared to really-flown rotorcraft trajectories, referred to as *reference* trajectories in the following. These reference trajectories are typically retrieved from rotorcraft-radar communications or from rotorcraft flight recorder. Each instance is characterized by the following elements, which are illustrated in Figure 10:

- **A search space.** It has generally a cube shape and corresponds to a small portion of the Earth surface (see Fig. 10a). It is defined around the reference trajectory by latitude/longitude boundaries and a height ceiling.
- **Obstacles.** They can be of different types (buildings, forbidden areas...) and can take different shapes. In this paper, obstacles are described by either prisms or cylinders in the 3-dimensional space. As an example, obstacles are represented in blue in Figure 10a.
- **Population data.** For the instances corresponding to real-world test cases, population density is associated to the latitude/longitude coordinates of the considered search space. Such data are publicly available for France, published by INSEE (National Institute of Statistics and Economic Studies), with a precision of 200 m (INSEE, 2022). A 2D representation of population density is given in Figure 10b. The browner the area the higher the population density.
- **Terrain elevation data.** As for population data, terrain elevation can be associated to the considered search space. In this paper, we used DTED1 (Digital Terrain Elevation Data), which is available for the whole Earth surface with a precision of 90 m (NGA, 1990). A representation of terrain elevation data is given in Figure 10c. The yellower the area, the higher the terrain.
- **A trajectory starting waypoint.** In this paper, we focus on cruise and approach flight phases of the trajectory. Therefore, the starting waypoint of the trajectory is a waypoint of the cruise phase, represented by a green dot in Figure 10. It is defined by four coordinates: latitude, longitude, altitude above mean sea level (AMSL) and speed. We use typical values for rotorcraft altitude and speed in cruise flight, that are respectively 1500 ft and 115 kt for the H130 helicopter considered in this paper, while latitude and longitude depend on the problem instance.
- **A trajectory ending waypoint.** It is the last considered waypoint of the trajectory, represented by a red dot in Figure 10. In practice, it is a *point in space* (PinS), where the pilot decides to perform landing or to go-around. The final landing procedure, coming after this PinS, is not considered in the optimization because it is strongly constrained in order to ensure a safe landing. The optimization is done up to this PinS, which is defined with four coordinates, like the starting waypoint. Typical values for rotorcraft height above ground level (AGL) and speed at this waypoint are 100 ft and 50 kt, respectively.

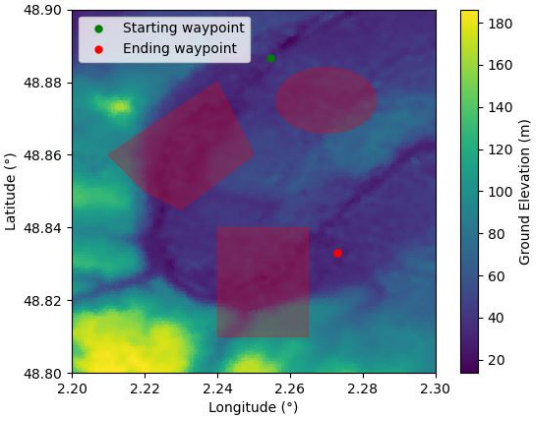
The results of our numerical experiments are in particular illustrated and discussed in the following on five instances. The first instance is the only one that does not correspond to a real-world traffic test case. Indeed, it has been generated in order to assess the performances of the different proposed approaches with respect to state-of-the-art results from the literature. This instance is defined as a 2D approach flight, where the horizontal path is considered fixed (i.e. x and y are not optimization variables anymore). In this case, only the vertical (z) and speed (v)



(a) Example of a 3D view of the search space. Obstacles are represented in blue.



(b) Population density distribution. Obstacles are represented in blue (2D upper view).

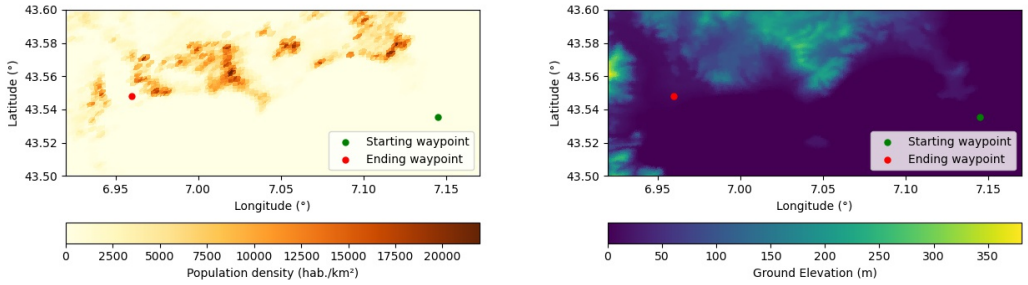


(c) Terrain elevation data. Obstacles are represented in red (2D upper view).

Figure 10: Illustration of the main characteristics of problem instances on a representative one (instance n°2)

profiles can vary. As the instance is not associated to a specific location, a uniform population density and a flat terrain are assumed. All the other instances correspond to real-world traffic examples. The second and third instances both address a cruise and approach flight to an heliport located in a very high densely populated area. They only differ on the cruise altitude: 1500ft and 2000ft, respectively. The environmental characteristics of these two instances are illustrated in Figure 10: the population density and the terrain elevation corresponding to the considered area are shown in Figure 10b and 10c, respectively. The fourth instance addresses a cruise and approach flight to a landing point that is located on a sea coast. The considered area is associated with a particularly uneven population distribution comprising both very high densely populated

zones (on the coast) and uninhabited areas (above the sea), as depicted in Figure 11a. Terrain elevation is represented in Figure 11b. The fifth instance is similar to the fourth instance in terms of population distribution while referring to a different geographic location.



(a) Population density distribution.

(b) Terrain elevation data.

Figure 11: Illustration of the environmental characteristics of instance n°4.

5.3 Results

In this section, we present numerical results obtained by applying the proposed approaches on the previously described problem instances. We first compare three well-known local DFO algorithms applied to multiple instances of the problem at hand. Then, we select the algorithm that performs best on our problem and keep it for the numerical experiments presented in the remaining of this section. The purpose of such numerical experiments is twofold.

First, we show that our proposed approach, relying on a state-of-the-art DFO algorithm and taking advantage from the tailored computation of the initial trajectory candidate and from the definition of the *corridor* to handle obstacles, allows (local) optimal solutions to be computed, featuring a significant reduction of the noise impact on population.

Second, we show the benefits of using the proposed surrogate models within an appropriate algorithmic scheme. The different surrogates are first compared to the original black-box simulator in terms of computational efficiency. Then, setting and handling the mesh parameters of the search within the proposed algorithmic scheme (Algorithm 3) is discussed. We show afterwards that this algorithmic scheme embedding the surrogates is able to provide good quality solutions within a reduced computing time compared to the algorithm based on the original black-box simulator alone (Algorithm 2). Finally, we present the results of the optimization relying only on the surrogate based on the neural network, where it totally replaces the original black-box simulator over the whole computation.

All the numerical tests have been carried out on a Linux platform with 48 CPUs (3.00 GHz) and 130 GB of RAM, and using NOMAD solver version 3.9.1.

5.3.1 Comparing DFO algorithms

In this section, we assess the performance, on several instances of the problem at hand, of three well-known local DFO algorithms: MADS (Audet and Dennis, 2006), RQLIF (Manno

et al., 2020) and BFO (Porcelli and Toint, 2018). They have been selected among other local DFO algorithms because their efficiency has been proven on many standard DFO problems (Rios and Sahinidis, 2013). The selected algorithms have been compared on 55 problem instances, among which 5 instances correspond to the ones described previously (Section 5.2) and the 50 others have been randomly generated around several heliports, considering local population density, terrain elevation and obstacles.

In order to carry out a fair comparison, the three tested algorithm are provided, for each problem instance, with the same initial trajectory. In addition, the same value is set for the user-defined parameters for the three algorithms. In particular, the values of the reduction and expansion factors of the mesh are set to 0.08 and 1.5, respectively (default RQLIF values). The maximum number of function evaluations is set to 1000 and is used as the stopping criterion for the three algorithms. The MADS algorithm is applied through its NOMAD (Le Digabel, 2011) implementation.

To compare these three algorithms, we use performance profiles introduced by Dolan and Moré (2002). In this paper, we define the performance of an algorithm on a given problem instance as the objective function value obtained after a fixed number (1000) of function evaluations. Figure 12 shows the performance profiles for the three tested algorithm on the 55 problem instances. For $\tau \geq 1$, $\rho_a(\tau) \in [0, 1]$ is the percentage of problem instances for which the performance of algorithm a is within a factor τ of the best one. Note that the higher the profile, the better the algorithm performs. We can see that, for the considered problem instances, the NOMAD implementation of the MADS algorithm outperforms both BFO and RQLIF. In particular, $\rho_a(1)$ is interesting in that it gives the fraction of problems for which algorithm a is better (i.e., in our case, provides a lower objective function value) than all the other tested algorithms. The value $\rho_a(1)$ is 64.0% for NOMAD, 18.0% for RQLIF and 18.0% for BFO.

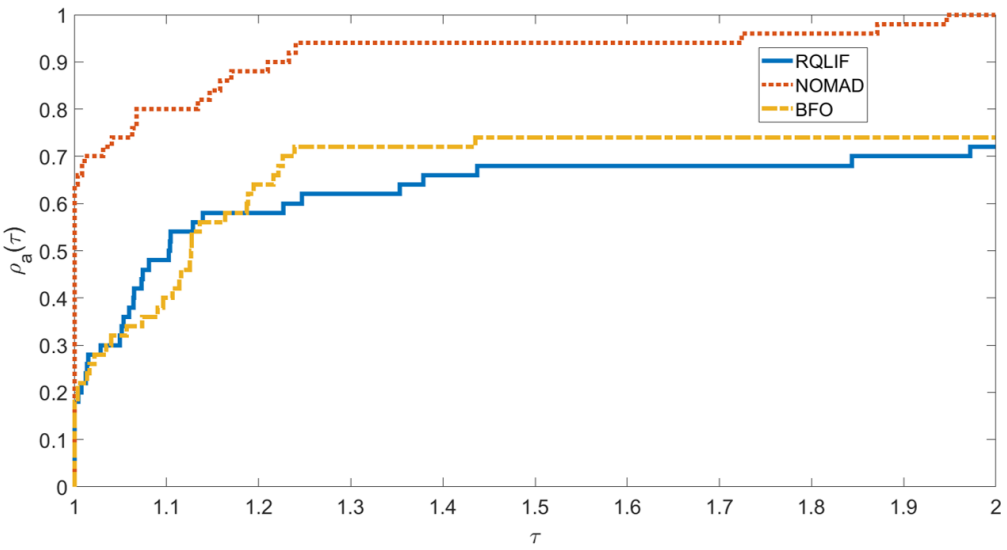


Figure 12: Performance profiles comparing the quality of the solution obtained after a fixed number of function evaluations by three well-known DFO algorithms on a set of 55 problem instances.

All the following numerical experiments have been carried out using the NOMAD implementation of the MADS algorithm within the different approaches proposed in the paper. The

results are detailed for the five instances described in Section 5.2, representative of the problem at hand.

5.3.2 Computing local optimal solutions of the black-box optimization problem

We show in the following that we are able to obtain good quality solutions by applying the first proposed solution approach (Algorithm 2), which does not make use of surrogate models. Table 1 reports, for the 5 real-world instances, the objective function values of the reference trajectory $\mathbf{X_{ref}}$, the initial trajectory $\mathbf{X_0}$, and the computed optimal trajectory $\mathbf{X_{opt}}$, respectively $f(\mathbf{X_0})$, $f(\mathbf{X_{ref}})$ and $f(\mathbf{X_{opt}})$. The required CPU time (in seconds) for the computation of $\mathbf{X_0}$ and $\mathbf{X_{opt}}$ is also given. In addition, the table gives the percentage reduction of the objective function value for the initial and the optimal trajectories with respect to that of the reference trajectory.

Problem instance	Reference trajectory ($\mathbf{X_{ref}}$)	Initial trajectory ($\mathbf{X_0}$)		Computed optimal trajectory ($\mathbf{X_{opt}}$)	
	$f(\mathbf{X_{ref}})$	$f(\mathbf{X_0})$	CPU Time (s)	$f(\mathbf{X_{opt}})$	CPU Time (s)
N°1	6794.1	6794.1 (0.0%)	N/A	6064.5 (10.8%)	15251
N°2	755116.6	527396.4 (30.2%)	6.8	429709.3 (43.1%)	10327
N°3	762795.4	528115.7 (30.8%)	7.3	437270.7 (42.7%)	9350
N°4	35190.2	24700.4 (29.8%)	5.4	5160.7 (85.3%)	8687
N°5	39064.5	7374.8 (81.1%)	5.8	3732.7 (90.4%)	13625

Table 1: Results for the optimal trajectories computed by the proposed solution approach (Algorithm 2) using the original black-box simulator.

On the one hand, we observe that the initial trajectory, which is used as a starting guess for the black-box optimization, shows already a significant improvement compared to the reference trajectory. This was expected as such initial trajectory is not randomly chosen but computed through a dedicated path planning algorithm, as described in Section 3.2.1. We notice that, for the first instance only, the initial trajectory does not improve the reference one. This is due to the specific characteristics of that instance: it addresses a straight approach flight, i.e., only the altitude and speed of the rotorcraft can vary. As the horizontal path is fixed and the path planning algorithm aims at minimizing population overflown and distance travelled, no improvement can be obtained at this stage. For the other instances, we get at least a 30% reduction in the objective function value compared to that of the reference trajectory. This shows that this first step is effective in finding a good initial trajectory candidate, while being computationally inexpensive compared to the whole optimization process.

On the other hand, Table 1 shows that the optimal trajectory $\mathbf{X_{opt}}$ computed through the first proposed solution approach always exhibits a reduction in the objective function value with respect to the trajectory currently flown in practice. We recall that the objective function measures the population exposure to the noise generated by a rotorcraft flying the considered trajectory. Its value is thus strongly correlated to the population density in the area above which the trajectory goes through. As the instances are characterized by very different population distributions (see the color scales in Figures 10b and 11a), the objective function values and their associated percentage reductions vary widely amongst them. In particular, we see that the improvement is the least significant (only 10.8%) for the first instance. First, as explained above, this is due to the fact that the horizontal path is not optimized. Then, for this particular instance, a uniform population distribution is considered. Therefore, the population density, that appears

in the definition of the objective function, takes the same value over all the considered area and does not actually guide the optimization. The second and third instances feature a reduction of approximately 43% in the objective function value. In terms of population exposure to noise, this is equivalent to a reduction of approximately 230000 people exposed to a SEL $\in [75, 80]$ dB(A). The percentage reduction is even more significant for the fourth and fifth instances (>85%). This can be explained by observing that there is a greater flexibility to reduce the population exposure to noise in sparsely populated areas rather than in high densely ones. Yet, for these instances, the improvement brought by the computed optimal trajectories is equivalent to a reduction of only about 21200 people exposed to a SEL $\in [75, 80]$ dB(A). This number is very low compared to that of second and third instances, again mainly due to a much lower population density. From the application standpoint, obtaining such good quality locally optimal solutions, i.e. rotorcraft trajectories whose noise footprint is significantly reduced compared to reference trajectories, is valuable for practitioners.

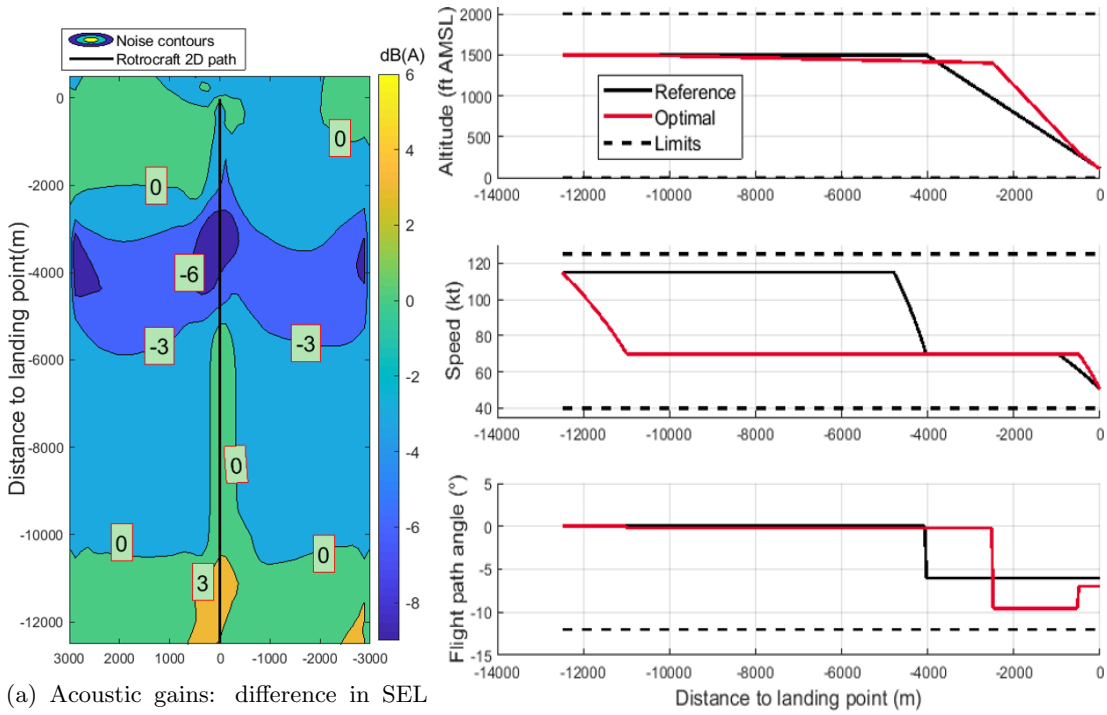
In the following, we further detail the results for the first two instances: the first one because of its particular characteristics, and the second one, as it is representative of the instances based on real traffic.

As regards the first instance, the results are illustrated in Figure 13. In particular, Figure 13a shows the difference in terms of SEL between the noise footprints associated to the computed optimal trajectory and the reference one. Significant reductions, up to -6 dB(A), can be obtained, that is consistent with the results of previous work in the literature (Padula et al., 2009; Guntzer et al., 2014). These reductions are very much related to the flight parameters variations, illustrated in Figure 13b. The computed optimal trajectory features a much steeper descent slope ($\gamma = -10°$) than the reference one ($\gamma = -6°$). As a consequence, the speed profile is also very different: a strong deceleration occurs from the cruise phase in order to be able to conduct such a steep descent, while sticking to the constraint on the vertical speed.

As regards the second instance, Figure 14 gives a 2D upper view of both the reference trajectory (plotted in black) and the computed optimal trajectory (plotted in red). On top of that, their associated SEL noise footprints are shown in Fig.14a and Fig.14b, respectively. We remark that the computed optimal trajectory comes with a much smaller noise footprint compared to that of the reference one. This strongly correlates with the reduced noise impact on population discussed previously. Again, reductions of noise exposure are directly related to changes in flight parameters during the flight, shown in Figure 15. This time, the altitude profile of the computed optimal trajectory remains similar, in terms of descent slope, to that of the reference one. However, the speed profile greatly differs: in the computed solution, the deceleration occurs as late as possible to reduce the overflight duration, thus the time-integrated SEL value on the ground. In addition to the flight parameters variations, we notice that, for the second instance, the reduced noise footprint is also very much related to the reduction in the trajectory length (see Fig. 14 and Fig. 15).

As regards the computing time, from the considered application standpoint optimal trajectory planning is done at a strategic level, that means from some days up to some hours in advance with respect to the flight. As a consequence, short computing times are not necessarily expected and even a magnitude of several minutes up to a few hours can still be considered acceptable. Table 1 first shows, as expected, that the computation of the initial trajectory $\mathbf{X_0}$ through the dedicated path planning algorithm is very efficient. The second phase of the optimization using the MADS algorithm is much more computationally expensive. This is due to the computational complexity of the black-box simulator.

Even though computing time is not critical it is still of interest to find ways to reduce it. This motivates the proposed approach relying on surrogate models, whose associated numerical results are discussed in the next subsection.



(a) Acoustic gains: difference in SEL between the noise footprint of the computed optimal trajectory and the reference one.

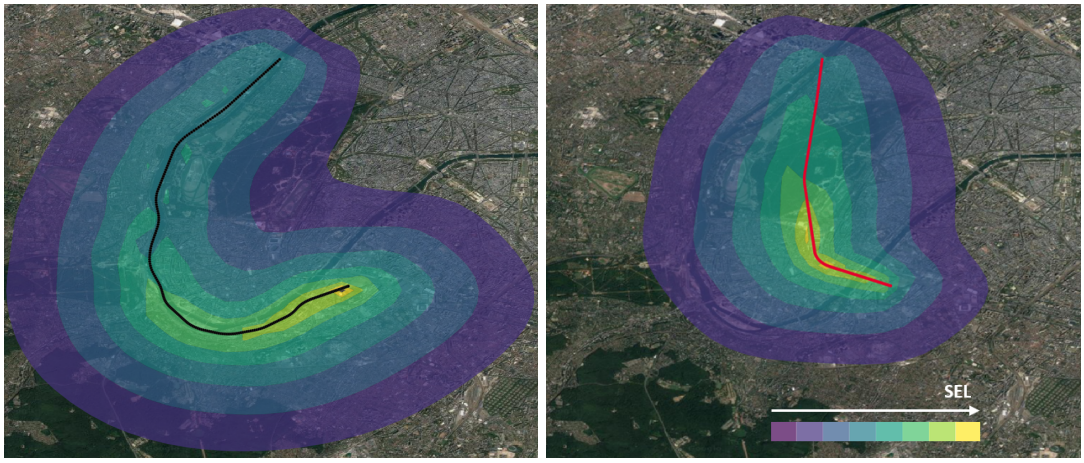
(b) Flight parameters variations (altitude z, speed v, and flight path angle γ) in function of the remaining distance to the landing point.

Figure 13: Results for instance n°1. Comparison in terms of both SEL and flight parameters variations between the computed optimal and the reference trajectories.

5.3.3 Investigating the impact of surrogate models

We focus hereafter on the usage of the different proposed surrogates introduced in Section 4.1. Prior to presenting the detailed results on all the instances, we recall that the surrogate models considered in this study are of different nature. In the following, they will be referred to as *isotropic approximation*, *microphone grid simplification* and *neural network*. The results obtained making use of these surrogates are compared to the ones obtained with the original black-box simulator.

First, we give an insight on their computational performance, without considering their embedding in any optimization algorithm. Table 2 reports the CPU time required to evaluate the trajectory represented in Figure 14a with the original black-box simulator and with the proposed surrogate models. As expected, the surrogates are much cheaper to evaluate than the original black-box simulator. The required CPU time is reduced by a factor of 4 with the isotropic approximation, 10 with the microphone grid simplification, and up to 130 with the neural network. In addition, regarding the shape of the computed noise footprint, the surrogates share strong similarities with the original black-box as discussed previously in Section 4.1. While illustrated here on one trajectory only for the sake of simplicity, a similar behavior can be observed on



(a) Noise footprint associated to the reference trajectory.

(b) Noise footprint associated to the computed optimal trajectory.

Figure 14: Results for instance n°2. Noise footprints associated to the computed optimal and the reference trajectories, illustrated by 5 dB(A) SEL noise contours. The considered rotorcraft (H130) flies from North to South.

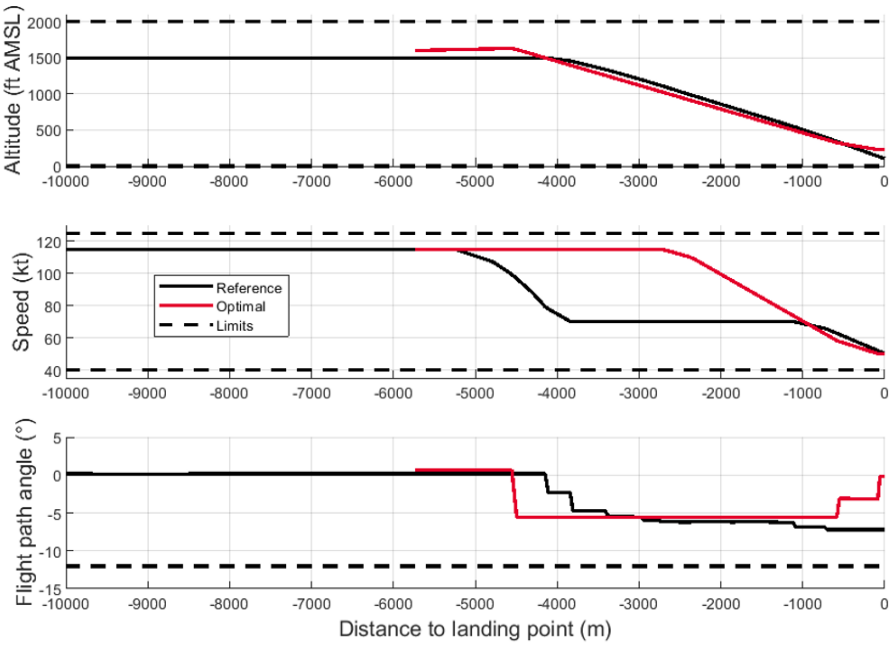


Figure 15: Results for instance n°2. Flight parameters variations (altitude z, speed v, and flight path angle γ) in function of the remaining distance to the landing point for the computed optimal and the reference trajectories.

other trajectories. This justifies the use of the proposed surrogates in combination to the MADS

Surrogate type	Original black-box	Isotropic approximation	Microphone grid simplification	Neural network
CPU Time (s)	68.2	16.4	7.0	0.5

Table 2: CPU time required for the evaluation of a single trajectory (see Figure 14a) with the original black-box simulator and with the proposed surrogate models.

algorithm through the algorithmic scheme (Algorithm 3) introduced in Section 4.2.

Then, the results in terms of both quality of the computed solutions and computational savings, with respect to the use of the original black-box only, are shown in Tables 3–7 for the five problem instances considered. The structure of the tables is the following. The first column indicates whether the original black-box is used or which surrogate is used among the ones defined in Section 4.1. In the following columns, $\mathbf{X_{switch}}$ denotes the current trajectory candidate when the switch criterion has been reached and $f(\mathbf{X_{switch}})$ the corresponding value of the objective function. The CPU time (expressed in seconds) up to the switch criterion is satisfied is denoted as t_{switch}. Similarly, $\mathbf{X_{opt}}$ denotes the computed optimal trajectory, $f(\mathbf{X_{opt}})$ the corresponding objective function value, and t_{opt} the CPU time. Finally, the total number of function evaluations is reported in the last column.

Surrogate type	$f(\mathbf{X_{switch}})$	CPU Time t_{switch} (s)	Mesh parameters setting	$f(\mathbf{X_{opt}})$	CPU Time t_{opt} (s)	Number of evaluations
Original black-box	N/A	N/A	N/A	6064.5	15251	1368
Isotropic approximation	6762.6	2656	Keep	5815.5	9732	891
			Reinitialize	5755.7	10809	981
Microphone grid simplification	6117.5	2861	Keep	5985.4	11375	1050
			Reinitialize	5934.7	13371	1232
Neural network	6561.6	610	Keep	5931.7	10970	1392
			Reinitialize	6054.0	11932	1514

Table 3: Numerical results obtained, for instance n°1, with the original black-box simulator and with the proposed surrogates, in terms of objective function value and of CPU time. A comparison is also done, for each surrogate, between the performances using two mesh parameters settings. For comparison purpose, the objective function value of the initial trajectory is $f(\mathbf{X_0}) = 6794.1$

In Tables 3 and 4 two lines are reported for each surrogate model. We recall that the MADS algorithm relies on a discretization of the search space as a mesh. In our proposed algorithmic scheme embedding surrogates (Algorithm 3), the MADS algorithm switches between the surrogate and the original black-box for the evaluation of the trajectory candidates. We investigate whether, when performing the switch, it is better to keep the mesh parameters to their current values or to reinitialize them to their initial values. We can see in Tables 3 and 4 that computed optimal solutions of a better quality are generally obtained when the mesh parameters are reinitialized to their initial values at the switch. This could be expected, as a reinitialization may help the MADS algorithm to escape from a local minimum of the surrogate model. However, this setting often comes with a higher number of function evaluations, thus a higher computing

Surrogate type	$f(\mathbf{X_{switch}})$	CPU Time t_{switch} (s)	Mesh parameters setting	$f(\mathbf{X_{opt}})$	CPU Time t_{opt} (s)	Number of evaluations
Original black-box	N/A	N/A	N/A	429709.3	10327	929
Isotropic approximation	523310.8	3445	Keep	430852.5	6419	577
			Reinitialize	425625.4	7652	688
Microphone grid simplification	542028.3	3656	Keep	430170.2	8341	750
			Reinitialize	426719.4	9076	816
Neural network	470560.3	255	Keep	449878.7	5280	475
			Reinitialize	430852.5	5311	478

Table 4: Numerical results obtained, for instance n°2, with the original black-box simulator and with the proposed surrogates. A comparison is also done, for each surrogate, between the performances using two mesh parameters settings. For comparison purpose, the objective function value of the initial trajectory is $f(\mathbf{X_0}) = 755116.6$

time compared to that obtained with the other setting (i.e. when the mesh parameters are kept to their actual values at the switch). As such computing time increase remains limited, we decided to reinitialize the mesh parameters after the switch for all the instances.

Surrogate type	$f(\mathbf{X_{switch}})$	CPU Time t_{switch} (s)	$f(\mathbf{X_{opt}})$	CPU Time t_{opt} (s)	Number of evaluations
Original black-box	N/A	N/A	437270.7	9350	848
Isotropic approximation	511361.1	1153	460817.2	6160	560
Microphone grid simplification	511361.1	1852	460817.2	6835	624
Neural network	486633.1	156	436066.2	9185	1032

Table 5: Numerical results obtained, for instance n°3, with the original black-box simulator and with the proposed surrogates. For comparison purpose, the objective function value of the initial trajectory is $f(\mathbf{X_0}) = 762795.4$

In the following, we discuss the benefits of using the proposed surrogates through the numerical results reported in Tables 3–7.

First, we focus on the quality of the different computed optimal solutions. We notice for all instances that different optimal solutions are computed (and so different values of the objective function are obtained) when using the surrogates. This can be explained by the fact that the problem at hand exhibits a large number of local minima. One could therefore expect MADS, being a local optimization method, to converge to different local optimal solutions. For the first three instances (Tables 3–5), some better quality solutions are obtained when using the

Surrogate type	$f(\mathbf{X_{switch}})$	CPU Time t_{switch} (s)	$f(\mathbf{X_{opt}})$	CPU Time t_{opt} (s)	Number of evaluations
Original black-box	N/A	N/A	5160.7	8687	785
Isotropic approximation	6232.8	1199	5198.1	6062	546
Microphone grid simplification	23037.2	689	5529.9	2975	271
Neural network	6986.6	333	5198.1	4045	579

Table 6: Numerical results obtained, for instance n°4, with the original black-box simulator and with the proposed surrogates. For comparison purpose, the objective function value of the initial trajectory is $f(\mathbf{X_0}) = 24700.4$

Surrogate type	$f(\mathbf{X_{switch}})$	CPU Time t_{switch} (s)	$f(\mathbf{X_{opt}})$	CPU Time t_{opt} (s)	Number of evaluations
Original black-box	N/A	N/A	3732.7	13625	1233
Isotropic approximation	7081.4	6372	4088.4	9080	834
Microphone grid simplification	6619.8	2632	4100.3	7782	710
Neural network	4216.5	848	4008.2	4988	997

Table 7: Numerical results obtained, for instance n°5, with the original black-box simulator and with the proposed surrogates. For comparison purpose, the objective function value of the initial trajectory is $f(\mathbf{X_0}) = 7374.8$

surrogates. Nevertheless, there is no surrogate that generally outperforms the others. On the contrary, for the last two instances (Tables 6 and 7), the optimal solutions computed when using the surrogates are of lower quality than those computed using the original black-box all over the optimization. Even though using the surrogates might lead to solutions of lower quality, it turns out that the solutions remain very close to each other both in terms of corresponding objective function values and trajectory shape. The objective function values of the computed solutions using surrogates differ from that of the computed solution using the original black-box simulator by less than 3% for all the considered instances. Furthermore, the third and fourth instances reveal the presence of equivalent local optimal solutions. In addition, the local solutions computed using the surrogates exhibit similar values of the decision variables, in particular in

terms of altitude and speed. Figure 16 shows, for a representative instance, that the gap in altitude, respectively in speed, between the optimal trajectories computed using the surrogates and using the original black-box is lower than 100ft, respectively 5kt.

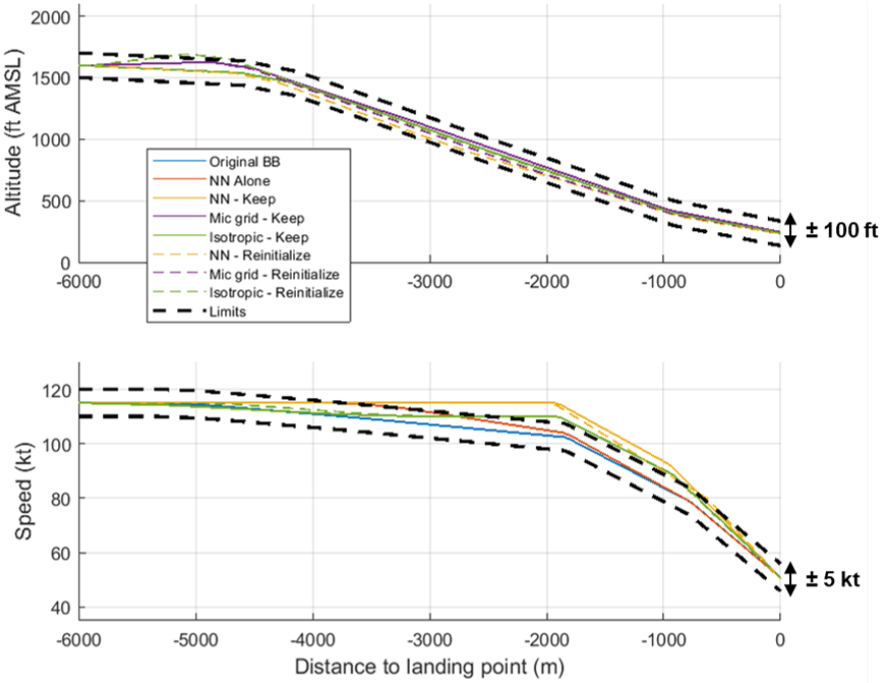


Figure 16: Altitude and speed profiles of the local optimal trajectories computed using the different surrogates and the original black-box simulator on instance n°2.

Then, as regards the computing time, we recall that short computing times are not necessarily expected. Tables 3-7 show that the CPU time is significantly reduced (up to 50%) for the computation using the different surrogates within Algorithm 3 in comparison to the computation relying on the original black-box only. The reason for this computing time reduction is twofold. The CPU time reduction is closely linked, on the one hand, to the reduction in the number of function evaluations to reach the optimum, and on the other hand to a more efficient evaluation of the candidate trajectories through the proposed surrogates. We observe a reduction in the number of function evaluations, and thus a reduction in CPU time, when using the surrogates built from an approximation of the original black-box (i.e. the isotropic approximation or the microphone grid simplification). When using the surrogate based on the neural network, a CPU time reduction is observed even though the number of function evaluations required to reach the optimum is higher than that with the original black-box. This is due to its very high efficiency in the evaluation of candidate trajectories, as discussed previously (see Table 2). While computational savings are observed for all the proposed surrogate models, we can see that there is no surrogate that outperforms the others over all the considered instances. This computing time reduction is interesting, as it may allow practitioners to move from a strategic to a pre-tactical level of decision making, that means only less than one hour prior to the flight.

Recall that surrogate based on the neural network looks very promising as it has been shown able to predict accurately the noise footprint of rotorcraft trajectories (see Fig. 9), and with a

reduced computational cost with respect to that of the original black-box simulator. Indeed, as detailed in Section 4.1.2, all the computational effort related to the neural network is done during the training, which is performed only once off-line, prior to the optimization. Taking into account such characteristics, one could consider the neural network totally replacing the original black-box simulator for an entire run of the MADS algorithm (not switching to the original black-box anymore). We investigate this possibility in Table 8 for all the problem instances considered. The second column gives the value of the objective function of the optimal solution computed by the neural network. The associated percentage reduction with respect to the solution computed with the original black-box simulator is given in parenthesis. The third column gives the CPU time and its associated reduction compared to that of the optimization with the original black-box. As regards the solutions, their quality remains very satisfactory compared to the ones computed with the original black-box: even though a reduced solution quality is naturally observed when using the surrogate, the gaps remain very low ($\leq 2.5\%$). As regards the CPU time, as expected, huge reductions in the CPU time are obtained ($> 90\%$) when the neural network replaces the original black-box simulator for all the evaluations.
Thanks to such huge CPU time reductions while keeping good quality solutions, practitioners could consider to use this approach, based on the use of the neural network surrogate over all the optimization process, for pre-tactical trajectory planning even up to a few minutes before flight.

Problem instance	$f(\mathbf{X_{opt}})$	CPU Time (s)
N°1	6088.4 (0.3%)	871 (94.3%)
N°2	440308.5 (1.4%)	441 (95.7%)
N°3	454442.3 (2.3%)	519 (94.5%)
N°4	5601.7 (1.3%)	839 (90.3%)
N°5	4138.8 (1.0%)	1042 (92.4%)

Table 8: Numerical results relying on the neural network replacing the original black-box simulator throughout the optimization.

6 Conclusion

In this paper, we address an emerging application of Operations Research focusing on rotorcraft minimal-noise trajectory design. We propose an algorithmic scheme, based on the MADS algorithm and relying on an effective computation of a good-quality initial candidate trajectory through a FMT* algorithm as well as on a tailored definition of the search space, allowing to reduce the variable domain while computing obstacle collision-free trajectories. We show that the locally optimal rotorcraft trajectories computed through the proposed approach relying on an industrial black-box simulator exhibit a significant noise reduction with respect to trajectories flown in practice.

We then propose multiple surrogate models to be embedded in such algorithmic scheme. They are defined building on our knowledge of the problem and relying, on the one hand, on the physics of the problem, and on the other hand on a machine learning model. The numerical results, obtained for representative real-world instances, show the interest of using surrogate models defined from the knowledge of the problem to enhance the optimization process and obtain good quality (i.e., associated to significantly reduced noise impacts) solutions, within reduced computing times. The proposed surrogate based on a neural network model exhibits,

in particular, very significantly reduced computing times, that appear promising for pre-tactical decision making for the considered application.

As the addressed problem typically exhibits a large number of local minima, future work will address global approaches to be combined with the presented algorithmic scheme. Defining multiple feasible trajectories representing possible alternatives in case of huge traffic appears also interesting to investigate.

References

Audet C, Dennis JE (2006) Mesh Adaptive Direct Search Algorithms for Constrained Optimization. SIAM Journal on Optimization 17(1):188–217. https://doi.org/10.1137/040603371

Audet C, Hare W (2017) Derivative-Free and Blackbox Optimization. Springer, https://doi.org/10.1007/978-3-319-68913-5

Audet C, Béchard V, Chaouki J (2008) Spent potliner treatment process optimization using a MADS algorithm. Optimization and Engineering 9:143–160. https://doi.org/10.1007/s11081-007-9030-2

Betts JT (1998) Survey of Numerical Methods for Trajectory Optimization. Journal of Guidance, Control, and Dynamics 21(2):193–207. https://doi.org/10.2514/2.4231

Chollet F, et al. (2015) Keras. URL https://github.com/fchollet/keras

Coelho BN, Coelho VN, Coelho IM, Ochi LS, Zuidema RHKD, Lima MS, da Costa AR (2017) A multi-objective green UAV routing problem. Computers & Operations Research 88:306–315. https://doi.org/10.1016/j.cor.2017.04.011

Conn A, Le Digabel S (2013) Use of quadratic models with mesh-adaptive direct search for constrained black box optimization. Optimization Methods and Software 28:139–158. https://doi.org/10.1080/10556788.2011.623162

Costa A, Nannicini G (2018) RBFOpt: an open-source library for black-box optimization with costly function evaluations. Mathematical Programming Computation 10:597–629. https://doi.org/10.1007/s12532-018-0144-7

Dasdemir E, Köksalan M, Öztürk DT (2020) A flexible reference point-based multi-objective evolutionary algorithm: An application to the UAV route planning problem. Computers & Operations Research 114(104811). https://doi.org/10.1016/j.cor.2019.104811

Delahaye D, Puechmorel S, Tsiotras P, Feron E (2014) Mathematical Models for Aircraft Trajectory Design: A Survey. In: Air Traffic Management and Systems, Springer Japan, Tokyo, pp 205–247. https://doi.org/10.1007/978-4-431-54475-3_12

Dieumegard P, FGuntzer, Caillet J, Cafieri S (2022) A Realistic Rotorcraft Noise Footprint Computation for Low-Noise Trajectory Optimization. In: American Helicopter Society 78$^{\text{th}}$ Annual Forum, Fort Worth, TX

Dolan ED, Moré JJ (2002) Benchmarking optimization software with performance profiles. Mathematical Programming 91:201–213. https://doi.org/10.1007/s101070100263

Fermi E, Metropolis N (1952) Numerical solution of a minimum problem. Tech. Rep. LA-1492, Los Alamos Scientific Laboratory of the University of California, https://doi.org/10.2172/4377177

Frazier PI (2018) A Tutorial on Bayesian Optimization. https://doi.org/10.48550/ARXIV.1807.02811

Gervais M, Schmitz F (2005) Neural Network Modeling of Measured Tiltrotor Acoustics for Designing Low-Noise Approach Profiles. In: American Helicopter Society 61st Annual Forum, Grapevine, TX

Gopalan G, Xue M, Atkins EM, Schmitz FH (2003) Longitudinal-Plane Simultaneous Non-Interfering Approach Trajectory Design for Noise Minimization. In: American Helicopter Society 59th Annual Forum, Phoenix, AZ

Grand View Research (2022) Commercial Helicopter Market Size, Share & Trends Analysis Report by Type (Light, Medium, Heavy), by Application, by Region, and Segment Forecasts, 2022-2030. GVR-3-68038-826-8, Grand View Research

Greenwood E (2017) Helicopter Flight Procedures for Community Noise Reduction. In: American Helicopter Society 73rd Annual Forum, Fort Worth, TX

Greenwood E (2018) Estimating Helicopter Noise Abatement Information with Machine Learning. In: American Helicopter Society 74th Annual Forum, Phoenix, AZ

Greenwood E (2019) Dynamic Replanning of Low Noise Rotorcraft Operations. In: American Helicopter Society 75th Annual Forum, Philadelphia, PA

Guntzer F, Gareton V, Gervais M, Rollet P (2014) Development and testing of optimized Instrument Flight Rules (IFR) noise abatement procedures on EC155. In: American Helicopter Society 70th Annual Forum, Montreal, Quebec, Canada

Gutmann HM (2001) A Radial Basis Function Method for Global Optimization. Journal of Global Optimization 19:201–227. https://doi.org/10.1023/A:1011255519438

Hagelauer P, Mora-Camino F (1998) A soft dynamic programming approach for on-line aircraft 4D-trajectory optimization. European Journal of Operational Research 107(1):87–95. https://doi.org/10.1016/S0377-2217(97)00221-X

Haykin S (2009) Neural Networks and Learning Machines. Pearson

Hertz J, Krogh A, Palmer RG (1991) Introduction to the Theory of Neural Computation. CRC Press, https://doi.org/10.1201/9780429499661

ICAO (2006) Procedures for Air Navigation Services. Aircraft Operations, vol 1. International Civil Aviation Organization

ICAO (2008) Doc 9829 AN/451. Guidance on the Balanced Approach to Aircraft Noise Management. ICAO, Montreal, Quebec, Canada

INSEE (2022) Revenus, pauvreté et niveau de vie en 2017 - Données carroyées. URL https://www.insee.fr/fr/statistiques/6215138?sommaire=6215217

Janson L, Schmerling E, Clark A, Pavone M (2015) Fast marching tree: A fast marching sampling-based method for optimal motion planning in many dimensions. The International Journal of Robotics Research 34(7):883–921. https://doi.org/10.1177/0278364915577958

Jones DR, Schonlau M, Welch WJ (1998) Efficient Global Optimization of Expensive Black-Box Functions. Journal of Global Optimization 13:455–492. https://doi.org/10.1023/A:1008306431147

Karaman S, Frazzoli E (2011) Sampling-based Algorithms for Optimal Motion Planning. The International Journal of Robotics Research 30(7):846–894. https://doi.org/10.48550/arXiv.1105.1186

Kiefer J, Wolfowitz J (1952) Stochastic Estimation of the Maximum of a Regression Function. The Annals of Mathematical Statistics 23:462–466. https://doi.org/10.1214/aoms/1177729392

Kocuk B, Altinel K, Aras N (2015) Approximating the objective function's gradient using perceptrons for constrained minimization with application in drag reduction. Computers & Operations Research 64:139–158. https://doi.org/10.1016/j.cor.2015.05.012

Larson J, Menickelly M, Wild S (2019) Derivative-free optimization methods. Acta Numerica 28:287–404. https://doi.org/10.1017/S0962492919000060

Le Digabel S (2011) Algorithm 909: NOMAD: Nonlinear Optimization with the MADS Algorithm. ACM Transactions on Mathematical Software 37(4):1–15. https://doi.org/10.1145/1916461.1916468

Manno A, Amaldi E, Casella F, Martelli E (2020) A local search method for costly black-box problems and its application to CSP plant start-up optimization refinement. Optimization and Engineering 21:1563–1598. https://doi.org/10.1007/s11081-020-09488-w

Mengistu T, Ghaly W (2008) Aerodynamic optimization of turbomachinery blades using evolutionary methods and ANN-based surrogate models. Optimization and Engineering 9:239–255. https://doi.org/10.1007/s11081-007-9031-1

Mockus J (1989) Bayesian Approach to Global Optimization. Mathematics and its Applications, Springer Dordrecht

Morris R, Johnson M, Venable KB, Lindsey J (2016) Designing Noise-Minimal Rotorcraft Approach Trajectories. ACM Transactions on Intelligent Systems and Technology 7(4):1–25. https://doi.org/10.1145/2838738

NGA (1990) Digital Terrain Elevation Data. URL https://earth-info.nga.mil/index.php?dir=elevation&action=elevation

Padula SL, Burley CL, Boyd Jr DD, Marcolini MA (2009) Design of Quiet Rotorcraft Approach Trajectories. Tech. Rep. TM-2009-215771, NASA

Porcelli M, Toint P (2018) BFO, A Trainable Derivative-free Brute Force Optimizer for Nonlinear Bound-constrained Optimization and Equilibrium Computations with Continuous and Discrete Variables. ACM Transactions on Mathematical Software 44:1–25. https://doi.org/10.1145/3085592

Prats X, Puig V, Quevedo J, Nejjari F (2010) Multi-objective optimisation for aircraft departure trajectories minimising noise annoyance. Transportation Research Part C: Emerging Technologies 18(6):975–989. https://doi.org/10.1016/j.trc.2010.03.001

Raap M, Zsifkovits M, Pickl S (2017) Trajectory optimization under kinematical constraints for moving target search. Computers & Operations Research 88:324–331. https://doi.org/10.1016/j.cor.2016.12.016

Rauch P, Gervais M, Cranga P, Baud A, Hirsch JF, Walter A, Beaumier P (2011) Blue Edge™: The Design, Development and Testing of a New Blade Concept. In: American Helicopter Society 67th Annual Forum, Fairfax, VA, pp 542–555

Rios LM, Sahinidis NV (2013) Derivative-free optimization: A review of algorithms and comparison of software implementations. Journal of Global Optimization 56:1247–1293. https://doi.org/10.1007/s10898-012-9951-y

Rodionova O, Sbihi M, Delahaye D, Mongeau M (2014) North Atlantic Aircraft Trajectory Optimization. IEEE Transactions on Intelligent Transportation Systems 15(5):2202–2212. https://doi.org/10.1109/TITS.2014.2312315

Schmitz F (1995) Reduction of Blade-Vortex Interaction (BVI) Noise through X-Force Control. Technical Memorandum 110371, NASA

Sridhar B, Ng HK, Chen NY (2011) Aircraft Trajectory Optimization and Contrails Avoidance in the Presence of Winds. Journal of Guidance, Control and Dynamics 34(5). https://doi.org/10.2514/1.53378

Sóbester A, Forrester AIJ, Toal DJJ, Tresidder E, Tucker S (2014) Engineering design applications of surrogate-assisted optimization techniques. Optimization and Engineering 15:243–265. https://doi.org/10.1007/s11081-012-9199-x

Vu KK, d'Ambrosio C, Hamadi Y, Liberti L (2017) Surrogate-based methods for black-box optimization. International Transactions in Operational Research 24(3):393–424. https://doi.org/10.1111/itor.12292

Wang L, Yang G, Sun Q, Ge J (2019) An uncertain optimization method for overall ballistics based on stochastic programming and a neural network. Engineering Optimization 51:663–679. https://doi.org/10.1080/0305215X.2018.1484122

Watson R, Downey O (2008) The Little Red Book of Acoustics: A Practical Guide. Blue Tree Acoustics

WHO (2018) Environmental Noise Guidelines for the European Region. World Health Organization Regional Office for Europe

Xia W, Shoemaker C (2021) GOPS: efficient RBF surrogate global optimization algorithm with high dimensions and many parallel processors including application to multimodal water quality PDE model calibration. Optimization and Engineering 22:2741–2777. https://doi.org/10.1007/s11081-020-09556-1