Multi-fidelity Bayesian optimization strategy applied to Overall Drone Design

Rémy Charayron, Thierry Lefebvre, Nathalie Bartoli, Joseph Morlier

Multi-fidelity Bayesian optimization strategy applied to Overall Drone Design

Rémy Charayron *
ONERA/DTIS, ISAE-SUPAERO, Université de Toulouse, Toulouse, France

Thierry Lefebvre † and Nathalie Bartoli ‡
ONERA/DTIS, Université de Toulouse, Toulouse, France

Joseph Morlier §
ICA, Université de Toulouse, INSA, CNRS, MINES Albi, UPS, Toulouse, France

Nowadays, drones can be developed for a wide range of use cases, from infrastructure monitoring to sea rescue, urban mobility or military purposes. Which drone design is best suited for a specific mission? To answer this question, we need to solve a constrained optimization problem based on a multi-disciplinary design model that takes the mission into account. The model generally being a computationally expensive numerical model whose gradients are not available all the time encourages us to consider a Bayesian optimization approach. Such strategy is well known to achieve a trade-off between exploitation and exploration in order to find interesting minimal area with a reduced number of function evaluations. A multi-fidelity approach can improve even more the computational efficiency of the Bayesian optimization strategy. In this work, we aim at designing a fixed-wing drone (fully electric) for long range surveillance mission. Two fidelity level electric drone models are developed. For a given mission requirement, the final battery state of charge is optimized with respect to drone design variables. Optimizations are performed on several missions using both a mono and a multi-fidelity Bayesian optimization strategy. The interest of using a multi-fidelity method for overall drone design has been assessed. The multi-fidelity super-efficient global optimization algorithm (MFSEGO) appeared to need less budget to reach convergence than the mono-fidelity algorithm and to be more robust to the initial design of experiments.

I. Nomenclature

C_D	=	Drag coefficient
C_L	=	Lift coefficient
D	=	Drag force (N)
e_b	=	Battery specific energy (Wh.kg^{-1})
ESC	=	Electronic speed controller
η_{ESC}	=	ESC efficiency
η_m	=	motor efficiency
η_p	=	propeller efficiency
η_{tot}	=	total efficiency
g	=	Gravity acceleration (m.s^{-2})
γ	=	Flight path angle
L	=	Lift force (N)
m	=	Total drone mass (kg)
m_{batt}	=	Battery mass (kg)
MDA	=	Multidisciplinary Analysis

*PhD Student, Information Processing and Systems Department & Aeronautical and Space Superior Institute
†Research Engineer, Information Processing and Systems Department, thierry.lefebvre@onera.fr, AIAA Member.
‡Senior researcher, Information Processing and Systems Department, nathalie.bartoli@onera.fr, AIAA MDO TC Member.
§Professor, Structural Mechanics Department, joseph.morlier@isae-supaero.fr, AIAA Member.

MDO = Multidisciplinary optimization
ODE = Ordinary differential equation
P_{batt} = Battery output power
P_{ESC} = ESC output power
P_m = Motor output power
P_{prop} = Propeller output power
Q_{batt} = Capacity of the battery (Coulomb)
SOC = State of charge (%)
T = Thrust force (N)
TAS = True airspeed (s)
UAV = Unmanned aerial vehicle
U_{batt} = Total electric tension of the battery (V)
U_{cell} = Electric tension of a single cell of the Lithium-Polymer (LiPo) battery (V)
W = Weight force (N)

II. Introduction

Optimization of systems with strong multidisciplinary interactions is one of the current challenges in the aircraft design community. This context of multidisciplinary design analysis and optimization (MDAO) [1] is reflected in this work. Of the various disciplines that could fit into a multidisciplinary Unmanned Aerial Vehicle (UAV) model, the mission discipline is important but not the easiest to consider at a detailed level. In this paper, we have endeavored to achieve a high fidelity mission discipline. Regarding the optimization strategy, Bayesian technique [2] was chosen to solve the problem, it is a well-known approach in the field of expensive black box optimization as it does not need the gradient information. In this work, we focus on the value of multi-fidelity. When multiple sources of information are available, we can take advantage of the characteristics of each source. In general, cheap but inaccurate information sources are used to explore the design space, while expensive and accurate sources are used for exploitation. First, a brief state of the art on multi-fidelity Kriging and Bayesian optimization is provided in Section III. Next, Section IV describes the two developed high-fidelity (HF) and low-fidelity (LF) UAV models that take into account the mission. Section V presents the four test cases experimented in this paper. Each test case corresponds to a geometric drone design for a specific mission. In Section VI, the results for each test case are outlined. Finally, the conclusion and perspectives are given in Section VII.

III. State of the art

This section presents the multi-fidelity framework used in this paper. We want to solve an optimization problem with inequality or/and equality constraints:

$$\min_{x\in\Omega} \; f(x) \qquad \text{such that} \begin{cases} g_i(x) \leq 0 \\ g_e(x) = 0 \end{cases} \tag{1}$$

where the objective function f, the inequality constraint functions g_i and the equality constraint functions g_e are evaluated with a Multidisciplinary Design Analysis (MDA) costly to evaluate. In a multi-fidelity context, the f, g_i and g_e can be obtained by different MDAs associated to different fidelity levels. To solve this multidisciplinary optimization (MDO) problem with expensive black-box functions, Bayesian optimizers [2, 3] are used to minimize the number of function evaluations. First Section III.A shows how to construct a multi-fidelity Gaussian process surrogate model. Then, in Section III.B, the current Bayesian optimization approach is adapted to multi-fidelity by defining a criterion to choose the fidelity level to query. We assume here that the reader is familiar both with Gaussian processes (GP) interpolation also denoted Kriging models [4–6] and with the classical mono-fidelity Kriging based Bayesian optimization methods like the Efficient Global Optimization (EGO) [7] algorithm for unconstrained problems and the Super Efficient Global Optimization (SEGO) algorithm [8–10] for constrained problems.

A. Multi-fidelity Kriging

The first extension of the Kriging model for multivariate functions, called co-Kriging, was first developed in geostatistics (see [11, 12]). Next, [13] figured that making assumptions to relate different levels of fidelity is a way to simplify multi-fidelity problems. In this section, focus will be made on Le Gratiet's [14] recursive formulation that we used to construct multi-fidelity Kriging models. Let's suppose that $L+1$ fidelity levels $f_0, ..., f_L$ sorted from the lowest to the highest are available to approach the function f (the same philosophy could be applied to the constraint functions g_i and g_e) and that we dispose of $L+1$ design of experiments (DoE), one for each fidelity level, denoted $(D_i)_{i=0...L}$ with $D_i = ((x_0, f_i(x_0)), ..., (x_j, f_i(x_j)), ..., (x_{N_i}, f_i(x_{N_i})))^T$, N_i being the number of points in D_i. We assume that we build some nested DoE: each point evaluated at a fidelity level is also evaluated at all the lowest fidelity levels: $D_L^x \subseteq D_{L-1}^x \subseteq \ldots \subseteq D_0^x$ where $D_i^x = (x_0, ..., x_j, ..., x_{N_i})^T$ corresponds to the inputs part of D_i. The nested DoE provides some properties useful to express the surrogate model variance in closed form. The following assumption first introduced by [13] in the bi-fidelity case is used to link the fidelity levels:

$$f_{l+1}(x) = \rho_l f_l(x) + \delta_{l+1}(x) \quad \text{such that} \quad f_l \perp \delta_{l+1} \quad \forall l = 0, ..., L-1 \tag{2}$$

where $\delta_{l+1}(x)$ is a discrepancy function that captures the difference between the $l+1$-th and the l-th fidelity levels while ρ_l is a scaling factor applied to f_l.
Le Gratiet [14] proposed to add the lowest fidelity function to the basis function set $(h_i)_{i=1...p}$ used in the universal Kriging regression term to get:

$$\mu(x) = \sum_{i=1,...,p} \left(\beta_i h_i(x)\right) + \beta_{\rho_0} f_0(x) \tag{3}$$

where β_{ρ_0} is an estimation of ρ_0 and $(\beta_i)_{i=1...p}$ is a set of unknown coefficients to be multiplied by the basis functions. Estimations of these coefficients are done by maximizing the likelihood [15, 16]. Since we used a nested DoE structure, the independence between fidelity levels of the surrogate model is assumed. Then the following recursive formulation [14] for the mean ($\hat{\mu}_{l+1}$) and variance ($\hat{\sigma}^2_{l+1}$) of each fidelity level GP surrogate model can be written:

$$\forall l = 0, ..., L-1 \qquad \begin{cases} \hat{\mu}_{l+1} = \rho_l \hat{\mu}_l + \hat{\mu}_{\delta_{l+1}} \\ \hat{\sigma}^2_{l+1} = \rho_l^2 \hat{\sigma}^2_l + \hat{\sigma}^2_{\delta_{l+1}} \end{cases} \tag{4}$$

In this case, ρ_l is considered as a constant but it can depend on x and we have $\rho_l : x \mapsto \rho_l(x)$. The learning process is the following, the lowest fidelity level is first learnt, then the relationships (scaling factor ρ_l and discrepancy function δ_{l+1} with $l = 0, ..., L-1$) between each fidelity level are successively learnt. Since the variances can be expressed in a closed form, the contribution of each fidelity level to the total variance of the multi-fidelity model can also be expressed. This is the most remarkable advantage of Le Gratiet's recursive formulation. Denoting $\sigma^2_{cont}(l, x)$ the variance contribution of the l^{th} fidelity level at the point x, with the notation $\sigma^2_{\delta_0} = \sigma^2_0$, we have:

$$\forall l = 0, ..., L-1 \quad \sigma^2_{cont}(l, x) = \sigma^2_{\delta_l}(x) \prod_{j=l}^{L-1} \rho_j^2 \quad \text{and} \quad \sigma^2_{cont}(L, x) = \sigma^2_{\delta_L}(x) \tag{5}$$

B. Multi-fidelity Bayesian optimization

With a multi-fidelity Bayesian optimization process, one must decide the most promising point and the fidelity level at which to evaluate it. It has been proposed in [14] to solve the problem of finding the point and the fidelity level in two successive steps. First the point is found using a classical acquisition function as in mono-fidelity Bayesian optimization. Then the knowledge of the variance contribution to each fidelity level gives some information to smartly decide which fidelity level to choose. Let $c_0, ..., c_L$ be respectively the querying costs of all the fidelity levels $f_0, ..., f_L$. Let us denote $\sigma^2_{red}(l, x^*)$ the variance reduction of the high fidelity model when the point x^* is evaluated with all the fidelity levels $\leq l$ to ensure the nested DoE structure

$$\sigma^2_{red}(l, x^*) = \sum_{i=0}^{l} \sigma^2_{\delta_i}(x^*) \prod_{j=i}^{L-1} \rho_j^2 \tag{6}$$

A criterion to choose the level of enrichment can then be written as:

$$l^* = \underset{l \in 0,...,L}{\arg\max} \ \frac{\sigma^2_{red}(l, x^*)}{(\sum_{i=0}^{l} c_i)^2} \tag{7}$$

One can remark that Eq (7) is a trade off between variance reduction and computational cost. Then the cost ratios between the different fidelity levels are important parameters to the method. This two step approach from [17, 18] is described in Fig. 11 of Appendix **??**. It is refered in the following as the Multi-Fidelity Super Efficient Global Optimization (MFSEGO). Now that we have detailed the multi-fidelity model and the optimization technique that we will use in this work, we describe the electric drone models we have developed for the purposes of this work.

IV. Electric drone models developed

The increasing use of drones and in particular electric drones for various missions is accompanied by a need to model these drones and the missions they are assigned. In this paper, we focus on a power line monitoring mission. We will consider that the mission is a succession of checkpoints along the power line through which the drone must pass. Two models of electric drones have been developed in this work: a high fidelity model and a low fidelity one. Both of them couple different disciplines (aerodynamics, structure, masses, geometry, mission, propulsion, ...) which are computed with more or less accuracy. The purpose of the models is to estimate different quantities of interest: the final state of charge of the drone's battery (SOC) at the end of the mission as well as the mission time, and two other outputs that indicate if the wing or the tail can be broken during the mission. Both models must be able to consider the UAV geometrical parameters such as wing span, root chord, taper ratio and wing and tail sweep angle, dihedral angle and spanwise twist control points as variables. In addition, structural parameters like wing and tail spanwise spar thickness control points are also considered as variables. Finally we want to model a function f depending on 19 design variables and described in the following equation.

$$f : \left| \begin{array}{rcl} \Omega \subset \mathbb{R}^{19} & \longrightarrow & \mathbb{R}^4 \\ x & \longmapsto & y \end{array} \right. \tag{8}$$

All the models have been developed in Python using OpenMDAO, an open source MDO framework [19] from NASA Glenn. The Extended Design Structure matrix (XDSM) [1] diagrams of our OpenMDAO analyses have been made thanks to WhatsOpt [20] which is a web application developed at ONERA to define and share multidisciplinary analyses in terms of disciplines and data exchange. Figure 1 is an example of XDSM that represents the LF model workflow. The aerostructural discipline relies on a tool called OpenAeroStruct (OAS) [21–23]. It is an open source lightweight tool that performs aerostructural analysis and optimization using OpenMDAO. It couples a vortex-lattice method (VLM) and a 6 degree of freedom 3-dimensional spatial beam model to simulate aerodynamic and structural analyses using lifting surfaces. OpenAeroStruct is developed by the NASA and the University of Michigan. In the following, two fidelity levels are presented to model the drone. For the aerostructural discipline OpenAeroStruct is used for both fidelity levels with different meshes. To deal with the mission in the multidisciplinary process, the low fidelity model relies on the electrical Breguet range equation while the high fidelity model requires to solve an optimal control problem which is tackled by the DYMOS software [24]. DYMOS is a framework for the simulation and optimization of dynamical systems within the OpenMDAO environment. It has two primary objectives: provide a generic ordinary differential equation (ODE) integration interface that allows for the analysis of dynamical systems and allow the user to solve optimal control problems involving dynamical multidisciplinary systems. In Section IV.A the low fidelity drone model is described and in Section IV.B the high fidelity one is detailed.

A. Low fidelity electrical drone model

The low fidelity model tries to approach the final SOC, the mission time, and the wing and tail failure constraints with a cheap computational cost. Figure 1 depicts the associated workflow with the different disciplinary modules involved in the low fidelity model and their associated number of inputs and outputs variables. Regarding performance during mission computation, this model relies on an adaptation of the Breguet range equation to electrical aircraft. The classic fuel Breguet range equation is recalled in Appendix VII.B. The electric adaptation is given by:

$$R = \frac{C_L}{C_D} \eta_p \eta_m \eta_{ESC} \frac{e_b}{g} \frac{m_{\text{batt}}}{m} \tag{9}$$

with η_p, η_m and η_{ESC} being respectively the efficiency coefficients for the propeller, the motor and the electronic speed controller. e_b is the battery specific energy, g the gravitational acceleration, m_{batt} the battery mass, and m is the total mass of the aircraft. Typically, for our study, we choose $e_b = 250$ Wh.kg^{-1}. This equation is valid in the steady level flight case and its demonstration is given in Appendix VII.C.

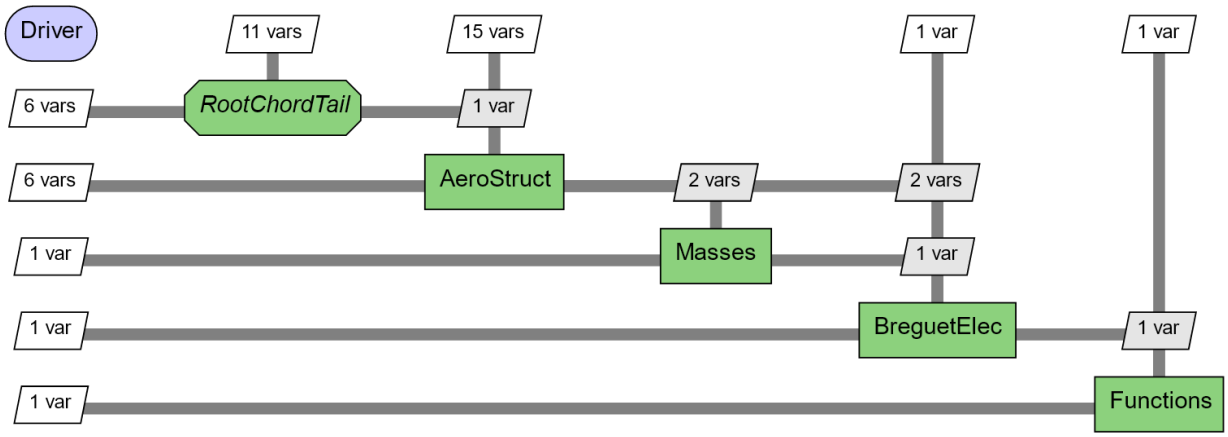


Fig. 1 XDSM for the low fidelity drone model with the different disciplinary modules involved.

The low fidelity workflow is therefore the following. First, the RootChordTail group (see Fig. 1) balances the tail root chord in order to ensure longitudinal stability for the drone. Figure 2 shows the XDSM of the RootChordTail group. Note that in order to ensure the longitudinal stability of the drone, we use a rule of thumb. The moment arm between the tail and the wing is equal to 3 meters. The tail root chord is balanced so that the tail surface is equal to 0.15 times the wing surface. In order to obtain a realistic tail root chord, we impose that the taper ratios of the wing and tail are equal and that the wing span is equal to 3 times the tail span. The tail span, taper ratio and root chord are not input variables as they depend on the geometric properties of the wing. Next, the aerostructure group (see Fig. 1) uses OpenAeroStruct

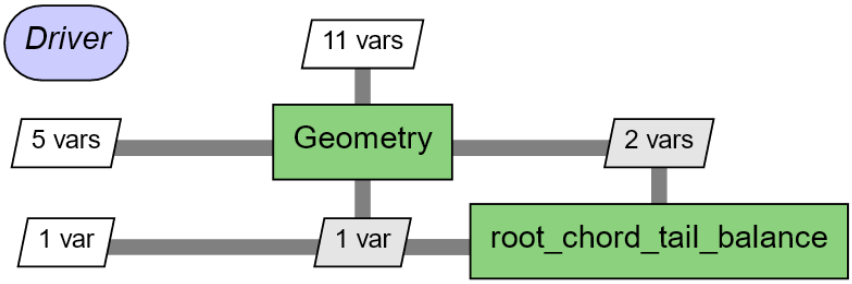


Fig. 2 XDSM of the RootChordTail group.

to compute the lift and drag coefficient required for the electrical Breguet component, the wing and tail failures that will be used as constraints by the optimizer and the wing and tail masses. The material used for the wing and tail spars is carbon fiber, and the following properties have been defined in the aerostructure model: the Young modulus of the spars is equal to $85e9$Pa, the shear modulus of the spars is equal to $25e9$Pa, the maximum authorized yield stress is set to $\frac{350e6}{m}$Pa with $m = 2.5$ being a safety margin and the material volumic mass is equal to $1.6e3$ kgm^{-3}. In addition, we choose a tubular spar for both the wing and the tail. The radius of these spars is considered constant in the spanwise direction with a radius of 0.044m for the wing tube and a radius of 0.02m for the tail tube. The thickness to chord ratio of the wing and of the tail are chosen to be constant in the spanwise direction and equal to 0.12. Note that this value of thickness to chord ratio ensures that the tubular spars can be effectively contained within the wing and the tail. It is also important to remark that we have chosen a fixed discretization of the wing and tail meshes in the OpenAeroStruct model. Indeed the number of points in the spanwise direction is equal to 9 for the wing and to 7 for the tail while the number of points in the chordwise direction is equal to 7 for the wing and to 5 for the tail. The wing and tail masses from the aerostructure component are passed to the masses component which deduces the overall drone mass. Finally the electrical Breguet range equation is applied to find the maximum range and the functions component (see Fig. 1) is

used to make a rough approximation of the SOC at the end of the mission estimated by the following formula:

$$SOC = (1 - \frac{R_{\text{target}}}{R_{\text{max}}})100 \tag{10}$$

where R_{target} is the targeted range of the mission and R_{max} is the maximum range that the drone can perform (computed with the electrical Breguet range equation Eq (9)).

B. High fidelity electrical drone model

The high fidelity model also attempts to approximate the final SOC and the failure constraints for the wing and the tail, but with greater accuracy. Figure 3 shows the XDSM diagram of the high fidelity drone model with the main disciplinary modules involved.

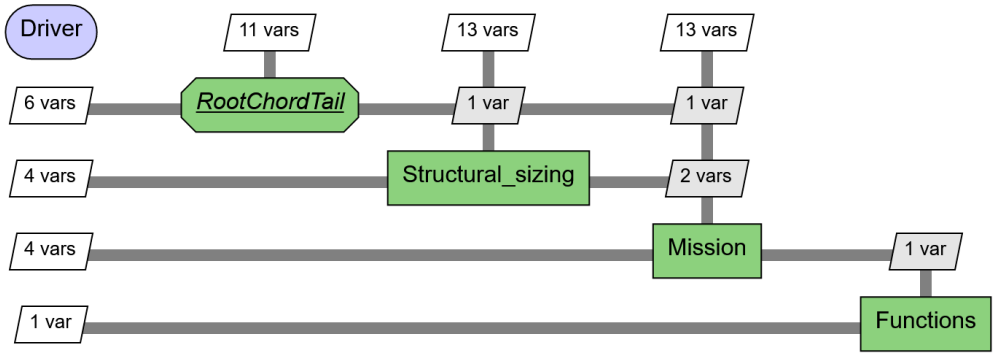


Fig. 3 XDSM for the high fidelity drone model with the different disciplinary modules involved.

The strategy involved computes the optimal trajectory in terms of energy consumption and then returns the final SOC passing through this optimal path. The workflow of the high fidelity model is as follows. First, as with the low fidelity model, the tail root chord is balanced to ensure the longitudinal stability of the UAV. Next, we perform the structure sizing using a call to OpenAeroStruct at the most critical flight point: maximum speed and maximum angle of attack. This component computes the wing and tail failure constraints and the wing and tail masses. Note that, as for the low fidelity model, the wing and tail spars are considered tubular. The spar radius, the spar material properties, the thickness to chord ratio, and mesh discretization are also considered to be the same as in the low fidelity model for the wing and the tail. Finally, the mission is completed and the corresponding component provides the mission time and the battery state of charge. The main part of the model is encapsulated in the mission component which calls DYMOS [24], the OpenMDAO package for optimal control. The mission component has n control points of the trajectory through which the drone must pass and which define the mission requirements. Each control point is composed of 4 values: altitude, range, Mach number, and climb rate. We then build $n - 1$ phases between each consecutive control point. To be consistent with the low fidelity model, the trajectory control points are chosen so that the sum of each of the distances of the $n - 1$ phases is equal to the target range R_{target} of the low fidelity model. DYMOS is used to determine the detailed optimal trajectory that minimizes the final SOC on each phase knowing the set of design variables. In short, DYMOS looks for the optimal control variables (climb rate and Mach number) that maximize the final SOC for a fixed UAV design with the constraint of passing through the trajectory control points on each phase. A constraint is added to the optimal control problem in order to ensure that the maximum possible thrust is not exceeded during the mission. Denoting n_{phases} the number of phases, the optimal control problem solved with DYMOS on each phase is described by

the formula:

$$\forall i \in 0, \ldots, n_{\text{phases}}, \quad u_i^* = \max_{u_i \in U_i} \int_{t_i}^{t_{i+1}} \frac{\partial SOC}{\partial t}(s_i(t), u_i(t))dt$$

$$\text{such that} \begin{cases} \text{Thrust}(t) \leq \text{Thrust}_{\max} \quad \forall t \in [t_i, t_{i+1}] \\ \text{alt}(t_i) = \text{alt}_i \\ \text{range}(t_i) = \text{range}_i \\ \text{Mach}(t_i) = \text{Mach}_i \\ \text{climb rate}(t_i) = \text{climb rate}_i \\ \text{alt}(t_{i+1}) = \text{alt}_{i+1} \\ \text{range}(t_{i+1}) = \text{range}_{i+1} \\ \text{Mach}(t_{i+1}) = \text{Mach}_{i+1} \\ \text{climb rate}(t_{i+1}) = \text{climb rate}_{i+1} \end{cases} \tag{11}$$

where s_i and u_i are respectively states and controls over the i-th phase and $(\text{alt}_i, \text{range}_i, \text{Mach}_i, \text{climb rate}_i)$ is the i-th trajectory control point. Note that we perform a maximization since the state of charge time derivative is always negative. We consider the control system associated with our optimal control problem. It can be written as a system of differential equations of the following form:

$$\dot{s} = H(s, u) \tag{12}$$

The control variables u are the climb rate and the Mach number and the state variables s are the SOC, the altitude and the range. H is a function of the state and control variables which computes the time derivatives $\dot{s}$ of the state variables. To solve the optimal control problem, the time derivatives of all the state variables must be computed. To do so DYMOS allows the time derivatives to be provided either as control variables or as outputs of an OpenMDAO model able to compute them by approximating the H function defined by Eq. (12). To approximate the state variables time derivatives $\dot{s}$, we proposed a model denoted by $\hat{H}(.)$, coded in OpenMDAO, and depicted in Fig. 4.

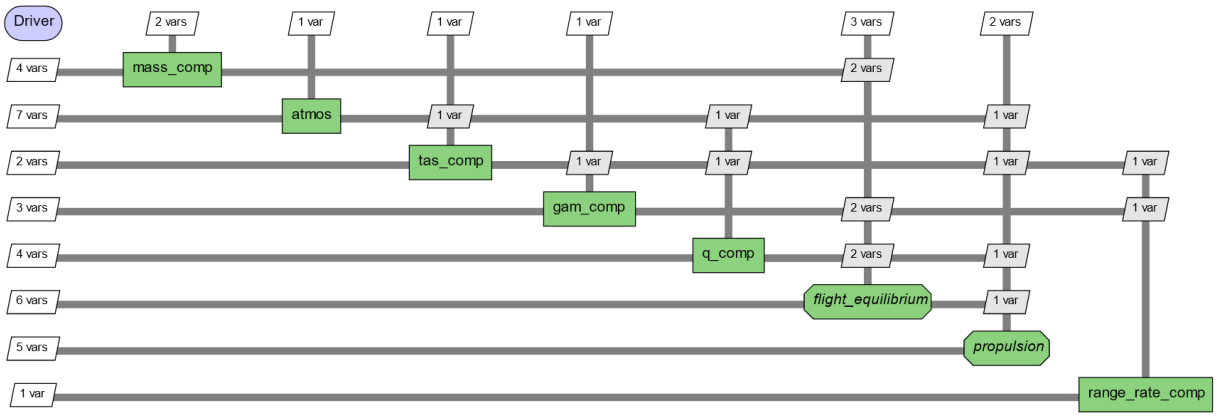


Fig. 4 XDSM of the model used to compute the optimal control problem state variable time derivatives. It is used in the mission component which relies on DYMOS.

The altitude time derivative is the climb rate which is given as a control variable. To compute the range time derivative, we first compute the altitude dependent characteristics of the atmosphere. Then, we compute the true air speed (TAS) using the Mach number and the speed of sound in the air in these atmospheric conditions. Knowing the TAS and the climb rate, we can apply elementary trigonometric considerations to compute the flight path angle γ and the range rate which is the speed component along the x-axis. The SOC rate ($\dot{SOC}$) is the most complex part of the model $\hat{H}(.)$ to compute. It is computed in the propulsion group whose XDSM diagram is given in Fig. 5. The SOC rate is the opposite of P_{batt} the power output of the battery at each instant scaled by the product between the battery capacity Q_{batt} (Coulomb) and the battery voltage U_{batt} (Volt). We express $\dot{SOC}$ in percentages:

$$\dot{SOC}(t) = \frac{\partial SOC}{\partial t}(t) = -P_{\text{batt}} \frac{100}{Q_{\text{batt}} U_{\text{batt}}} \tag{13}$$

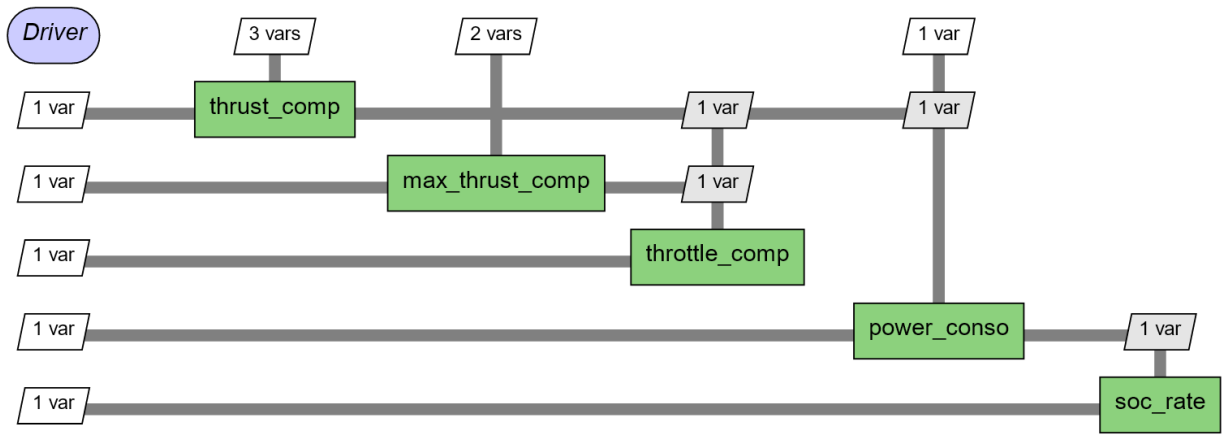


Fig. 5 XDSM of the propulsion group.

where $U_{\text{batt}} = U_{\text{cell}} \times k$, with U_{cell} being the electric tension of a single cell in the on-board Lithium-Polymer (LiPo) battery and k being the number of cells in the battery. Typically, $U_{\text{cell}} = 3.7V$ and $k = 3$. P_{batt} is necessarily greater than the propeller output power P_{prop} required for flight equilibrium due to the efficiencies of the propeller, motor and electronic speed controller (ESC) being less than 1. These efficiency losses passing through the propeller, motor and ESC are modeled using efficiency coefficients $\in [0; 1[$, respectively $\eta_p = 0.7$, $\eta_m = 0.95$ and $\eta_{ESC} = 0.98$ and we have:

$$P_{\text{prop}} = P_{\text{batt}} \eta_p \eta_m \eta_{ESC} \tag{14}$$

The required power output of the propeller depends on the required thrust T for flight balance and the true air speed TAS

$$P_{\text{prop}} = T \times TAS \tag{15}$$

with the thrust T being equal to the product of dynamic pressure $q = \frac{1}{2}\rho_{\text{air}} TAS$, the reference area $S = S_{\text{wing}} + S_{\text{tail}}$ and the thrust coefficient C_T

$$T = qSC_T \tag{16}$$

The dynamic pressure q is computed in a dedicated component inside the $\hat{H}$ model while the reference surface is given to the $\hat{H}$ model as an input coming from the structure component. The C_T is an output of the steady flight equilibrium group of $\hat{H}$. Note that the trajectories considered are simple, so not dealing with acceleration phases and only assuming steady flight phases is accurate enough for us. Within the steady flight equilibrium group, we first call the aerodynamic component. Using OpenAeroStruct simulations directly at this stage to compute the aerodynamics is very expensive in term of computational cost because a very large number of simulations must be performed. Then, for the UAV defined by the MDA inputs variables, two surrogate models of the OpenAeroStruct aerodynamic model are built to solve the optimal control problem at a reasonable cost. The same radii, material properties, the same thickness to chord ratios and same mesh discretizations as in the low fidelity model are considered. These surrogate models are Kriging models built at the very beginning of the mission component from a 90 point Latin Hypercube Sampling (LHS) design of experiments. Once built the surrogate models are passed to $\hat{H}$ through an option dictionary to be used in it. They take three inputs (altitude, Mach number and angle of attack). The first surrogate provides the lift coefficient C_L while the second provides the drag coefficient C_D.

$$C_D = C_{D,GP}(\text{AoA}, \text{Mach}, \text{alt}) \tag{17}$$

$$C_L = C_{L,GP}(\text{AoA}, \text{Mach}, \text{alt}) \tag{18}$$

Next we use the C_D to compute the C_T required for steady flight equilibrium before computing the C_L required for steady flight equilibrium. We denote by W_{total} the total weight of the UAV. It is computed in a previous component of $\hat{H}$ the time derivative model and is equal to the sum of the battery weight (fixed equals to 15 kg), the payload weight (fixed equals to 10 kg), the fuselage weight (fixed equals to 20 kg) and the lifting surface structural weight (computed by the

structural component of the model).

$$C_T = W_{\text{total}} \frac{\sin(\gamma)}{\cos(AoA) qS} + \frac{C_D}{\cos(AoA)} \tag{19}$$

$$C_{L_{eq}} = W_{\text{total}} \frac{\cos(\gamma)}{qS} - C_T \sin(AoA) \tag{20}$$

Finally we use a Newton solver to find the angle of attack that balances the lift coefficient from the aerodynamic surrogate model and the one that allows steady flight equilibrium. In other words we want to solve the following non linear coupled system:

$$\begin{aligned} &\text{find} \quad AoA \in [-15; 15] \\ &\text{such that} \quad C_{L_{eq}}(C_D(\text{AoA, Mach, alt})) = C_L(\text{AoA, Mach, alt}) \end{aligned} \tag{21}$$

The thrust coefficient C_T used in Eq. (16) is the one of Eq. (19) once the Newton solver which balances the lift coefficient has converged. The workflow of the steady flight equilibrium group is described in Fig. 6. Figure 7 shows an optimal

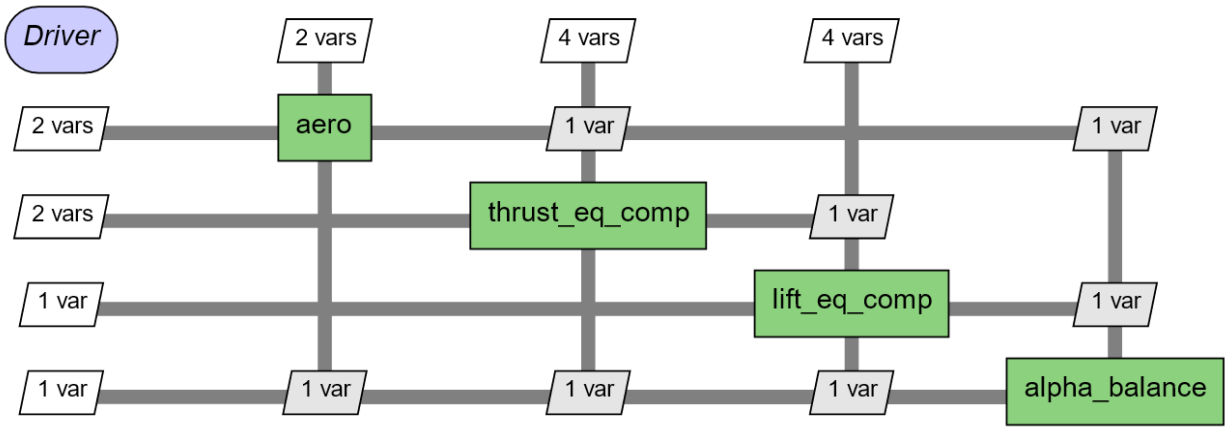


Fig. 6 XDSM of the steady flight equilibrium group which is a sub analysis of Fig. (6).

trajectory obtained on a climb phase between 2 control points. The phase begins at an altitude of 6 kft and a range of 0 NM and ends at 6.4 kft and 2 NM. The optimal trajectory in terms of energy consumption appears to be linear for this phase. The blue dots representing the DYMOS solution match perfectly with the blue dotted line representing the simulation of the dynamical system evolution when subjected to the controls computed with DYMOS. This match ensures that the optimal control solution is dynamically realistic.

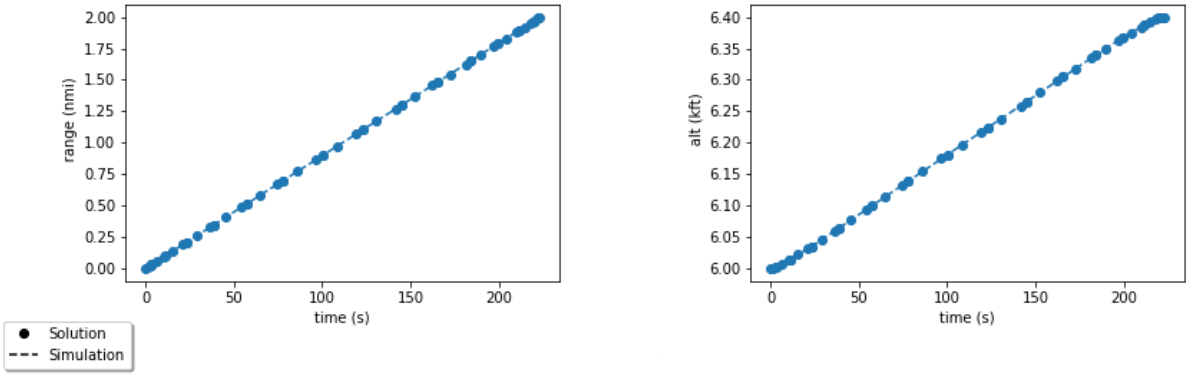
V. Mission test cases

In this section, the overall methodology adopted for the optimization and the mission test cases are described. Four test cases are considered (Mission 1 to Mission 4), each corresponding to a different mission of gradual complexity. Regardless of the test case considered, the optimization problem we want to solve is to maximize the mission end SOC with respect to 19 design variables $x \in \Omega$ where the design space $\Omega = ([0, 45] \times [-20, 20] \times [0, 1.2] \times [-5, 5]^3 \times [0.001, 0.01]^3)^2 \subset \mathbb{R}^{19}$ and satisfying two structural constraints. The design variables are listed in Table 1 and the constraints in Table 2.

The high-level optimization problem is expressed as follows:

$$\begin{aligned} x^* = \underset{x \in \Omega}{\arg\max} \;\; & \text{SOC}_{\text{final}}(x) \\ \text{such that} \begin{cases} \text{wing}_{\text{failure}}(x) \leq 0 \\ \text{tail}_{\text{failure}}(x) \leq 0 \end{cases} & \end{aligned} \tag{22}$$

In this paper we want to compare the performance of the Super Efficient Global Optimization (SEGO) algorithm [8], a mono fidelity Kriging based Bayesian optimization technique that only uses the high fidelity model with the performance



(a) Range vs time. **(b) Altitude vs time.**

Fig. 7 Illustration of an optimal trajectory obtained on a climb phase.

Table 1 Definition of the 19 design variables.

variables	design space	unit	quantity
wing span	$[5, 6]$	m	1
root chord wing	$[0.9, 1.2]$	m	1
wing taper ratio	$[0.6, 1]$	no unit	1
wing sweep angle	$[0, 5]$	degree	1
tail sweep angle	$[0, 5]$	degree	1
wing dihedral angle	$[-3, 3]$	degree	1
tail dihedral angle	$[-3, 3]$	degree	1
structural tube thickness control points along wing span	$[0.001, 0.01]^3$	m	3
structural tube thickness control points along tail span	$[0.001, 0.01]^3$	m	3
twist control points along wing span	$[1, 1.5]x[0.5, 1]x[0, 0.5]$	degree	3
twist control points along tail span	$[1, 1.5]x[0.5, 1]x[0, 0.5]$	degree	3
Total number of design variables			**19**

Table 2 Definition of the constraints.

name	type	bound	dimension	tolerance
$\text{wing}_{\text{failure}}$	$\leq$	0	1	$1e-7$
$\text{tail}_{\text{failure}}$	$\leq$	0	1	$1e-7$
Total number of constraints			**2**	

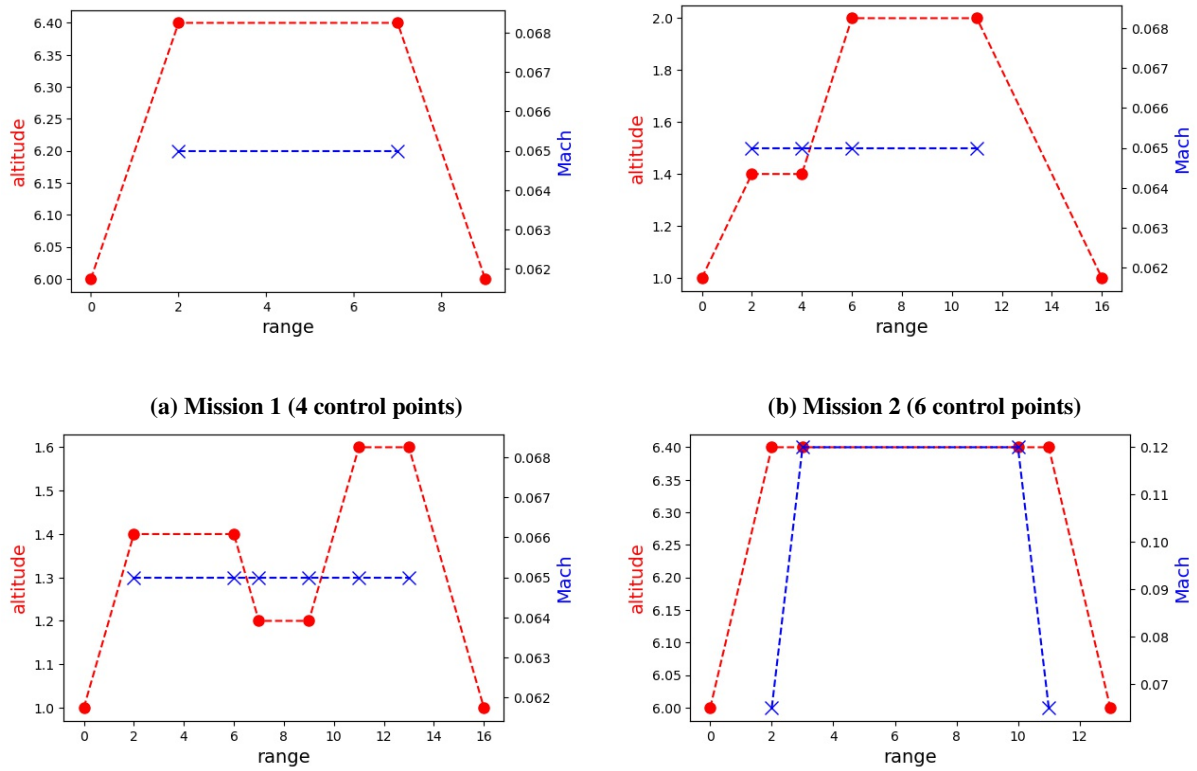
of the Multi-Fidelity Super Efficient Global Optimization (MFSEGO) algorithm [17, 18], the multi-fidelity co-Kriging based Bayesian optimization method described in Section III.B which takes advantage of both the high and the low fidelity electric drone models. The squared exponential kernel and a constant trend are chosen to build the Kriging (if mono fidelity) or the co-Kriging (if multi-fidelity) models at each iteration of the Bayesian optimizer. The Super Efficient Global Optimization Mixture Of Experts (SEGOMOE) framework [9, 10, 25, 26] is selected to perform the Bayesian optimizations. SEGOMOE is a Bayesian optimization internal ONERA & ISAE-SUPAERO toolbox. It relies on the Surrogate Modelling Toolbox (SMT) introduced in Section III.A. Note that [27] implemented MFSEGO in SEGOMOE, [18] adapted this implementation to fit the SMT format. It is expected that the use of multi-fidelity will decrease the total optimization cost by reducing the number of calls to the high-fidelity model. One can remark that since the number of design variables is relatively high in our problem it is useful to use a dimension reduction technique in order to save computational cost. To do so we choose to use the partial least squares method (PLS) technique [28, 29]. The PLS method finds a linear relationship between input variables and the output variable by projecting input variables

onto a new space of lower dimension formed by newly chosen variables called latent variables. PLS has been integrated into the SEGOMOE and can be coupled with SEGOMOE's multi-fidelity kriging method. In our tests, we realized that 4 latent variables were enough to capture most of the information. Adding a fifth latent variable does not improve the precision since almost all of the variance is captured by the 4 first latent variables. On the other hand, removing one of them will strongly reduce the prediction capability without reducing the computational cost that much. Then, we decided to set the dimension of latent space to 4.

For the MFSEGO method, the cost ratio between the HF and LF models must be defined. Monte Carlo simulations of the HF and LF models on 100 randomly drawn points according to a uniform distribution are used to estimate the computational costs for all mission test cases. The cost ratios are given by the CPU time ratios. They are summarized in Table 3. The control points defining the four different missions are shown in Fig. 8 where the altitude and Mach values are given as a function of range.

Table 3 Cost ratio for each mission test case.

	Mission 1	Mission 2	Mission 3	Mission 4
cost ratio	140	124	168	164



(a) Mission 1 (4 control points)

(b) Mission 2 (6 control points)

(c) Mission 3 (8 control points)

(d) Mission 4 (6 control points)

Fig. 8 Range, altitude (red dot) and Mach values (blue cross) at the control points of the four missions.

In order to compare the performances of the mono and multi-fidelity methods, we performed 10 runs for the single-fidelity Bayesian optimization using SEGO and 10 runs for the multi-fidelity optimization using MFSEGO, respectively. The mono and multi-fidelity runs share the same HF DoE. Table 4 details the sizes of the initial DoE for each mission test case. For each multi-fidelity run, the size of the initial LF DoE is twice the size of the initial HF initial DoE and contains, among other points, the points of the initial HF DoE in order to respect the nested structure of the DoE. The COBYLA [30] optimizer which stands for Constrained Optimization By Linear Approximation is used to determine a reference solution for each test case. COBYLA is an implementation of Powell's nonlinear derivative–free

Table 4 Initial DoE size for each mission test case.

	Mission 1	Mission 2	Mission 3	Mission 4
HF	10	10	5	5
LF	20	20	10	10

constrained optimization that uses a linear approximation approach. The algorithm is a sequential trust–region algorithm that employs linear approximations to the objective and constraint functions. The obtained optimal designs for each mission test case are shown in Fig. 9. Now that a reference solution is available we can propose a convergence criterion

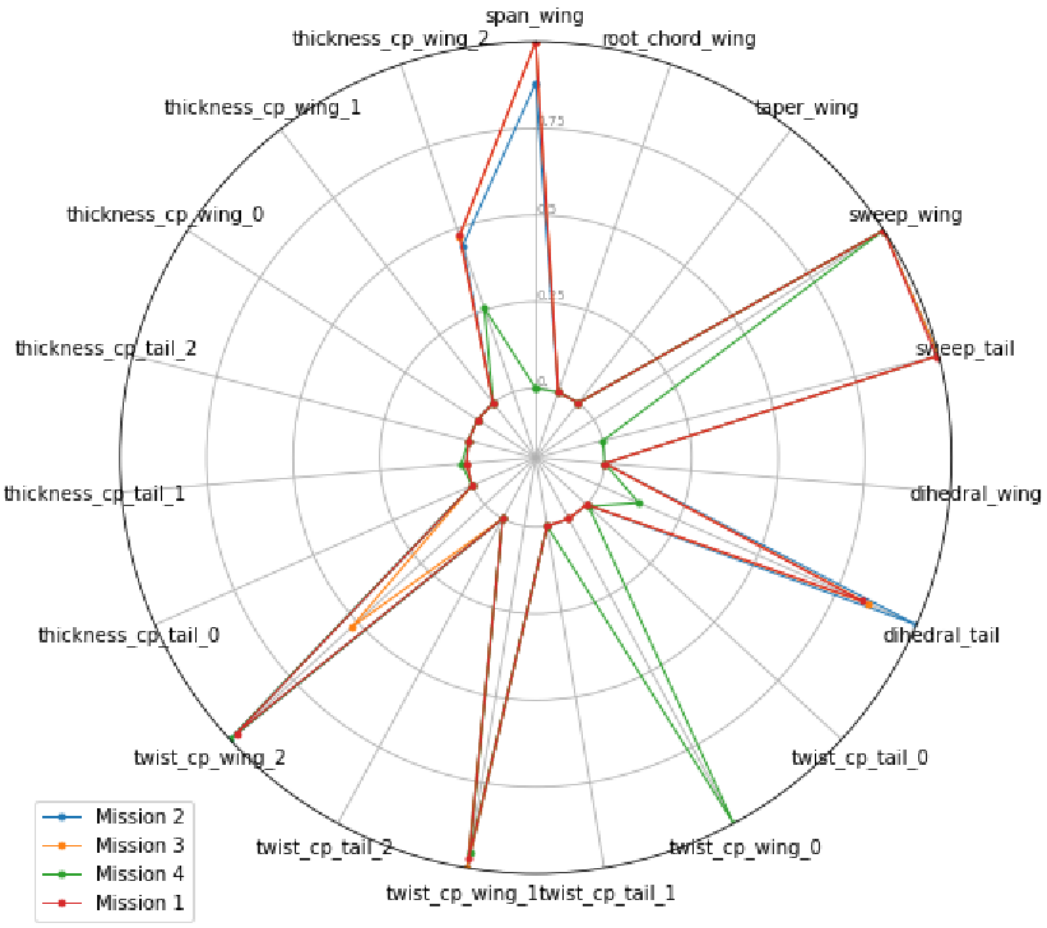


Fig. 9 Radar plot of the optimal drone design for each mission test case.

for our methods. Knowing that a point is considered feasible if the constraints are satisfied at this point, we denote by $y_{best}^{feasible}$ the best feasible objective value at a given iteration of the Bayesian optimization and $\bar{y}_{best}^{feasible}$ the respective mean value over the 10 runs. We can define the following convergence criterion method:

$$|\bar{y}_{best}^{feasible} - f_{\text{HF}}(\text{ref}_{sol})| \leq \epsilon_0 |f_{HF}(\text{ref}_{sol})| + \epsilon_1 \tag{23}$$

where ϵ_0 is a relative tolerance chosen equal to 0.001 while ϵ_1 is an absolute tolerance chosen equal to 0.05. The four different missions that the drone must satisfy are detailed in the following sections.

A. Mission 1

Mission 1 is defined by the set of four waypoints detailed in the Table 5. The values in this table are the values imposed on the control points. This first mission consists in a climb phase followed by a cruise phase and a descent

phase.

Table 5 Range, altitude, Mach and climb rate values at the Mission 1 waypoints.

	Range	alt	Mach	Climb rate
control point 0	0 NM	6.0 kft		
control point 1	2 NM	6.4 kft	0.065	0 $\frac{\text{ft}}{\text{min}}$
control point 2	7 NM	6.4 kft	0.065	0 $\frac{\text{ft}}{\text{min}}$
control point 3	9 NM	6.0 kft		

B. Mission 2

Mission 2 consists of by six waypoints (see Table 6). The specificity of this mission is that it carries out a level climb. Moreover, the flight altitude is much lower.

Table 6 Range, altitude, Mach and climb rate values at the Mission 2 waypoints.

	Range	alt	Mach	Climb rate
control point 0	0 NM	1.0 kft		
control point 1	2 NM	1.4 kft	0.065	0 $\frac{\text{ft}}{\text{min}}$
control point 2	4 NM	1.4 kft	0.065	0 $\frac{\text{ft}}{\text{min}}$
control point 3	6 NM	2.0 kft	0.065	0 $\frac{\text{ft}}{\text{min}}$
control point 4	11 NM	2.0 kft	0.065	0 $\frac{\text{ft}}{\text{min}}$
control point 5	16 NM	1.0 kft		

C. Mission 3

Table 7 shows the 8 waypoints that define Mission 3. The flight is performed at low altitude and includes two cruise phases.

Table 7 Range, altitude, Mach and climb rate values at the Mission 3 waypoints.

	Range	alt	Mach	Climb rate
control point 0	0 NM	1.0 kft		
control point 1	2 NM	1.4 kft	0.065	0 $\frac{\text{ft}}{\text{min}}$
control point 2	6 NM	1.4 kft	0.065	0 $\frac{\text{ft}}{\text{min}}$
control point 3	7 NM	1.2 kft	0.065	0 $\frac{\text{ft}}{\text{min}}$
control point 4	9 NM	1.2 kft	0.065	0 $\frac{\text{ft}}{\text{min}}$
control point 5	11 NM	1.6 kft	0.065	0 $\frac{\text{ft}}{\text{min}}$
control point 6	13 NM	1.6 kft	0.065	0 $\frac{\text{ft}}{\text{min}}$
control point 7	16 NM	1.0 kft		

D. Mission 4

The 6 waypoints of Mission 4 are detailed in Table 8. The main difference with previous missions is that the cruise phase must be performed at a highest minimal speed. A higher speed is then imposed. This mission consists in a climb phase, an acceleration phase, a cruise phase performed at least at $Mach = 0.1$, a deceleration phase and finally a descent phase.

VI. Results

For the four test cases, the MFSEGO method outperforms the SEGO method. Multi-fidelity appears to improve the budget required for convergence. Figures 10a, 10b, 10c and 10d show, respectively for Missions 1, 2, 3, 4, the

Table 8 Range, altitude, Mach and climb rate values at the Mission 4 waypoints.

	Range	alt	Mach	Climb rate
control point 0	0 NM	6.0 kft		
control point 1	2 NM	6.4 kft	0.065	0 $\frac{\text{ft}}{\text{min}}$
control point 2	3 NM	6.4 kft	0.12	0 $\frac{\text{ft}}{\text{min}}$
control point 3	10 NM	6.4 kft	0.12	0 $\frac{\text{ft}}{\text{min}}$
control point 4	11 NM	6.4 kft	0.065	0 $\frac{\text{ft}}{\text{min}}$
control point 5	13 NM	6.0 kft		

evolution of the mean and 1σ confidence interval over the 10 runs for the objective value at the best current feasible point. The budget required to achieve the convergence criterion of Eq. (23) for both the mono-fidelity and multi-fidelity methods is presented in Table 9 for each mission test case. In this table, "Not reached" means that the convergence criterion was not met with the allowed budget of 100 computational units. One computational unit is the cost of one HF evaluation. For the first 3 missions, Fig. 10a, 10b and 10c show that compared to the SEGO method, the MFSEGO method approaches the optimum faster and eventually reaches the convergence criterion (see Eq. (23)) faster too. For Mission 4, at the beginning, both SEGO and MFSEGO methods approach the optimum at the same speed but at the end the convergence criterion (see Eq. (23)) is reached with the MFSEGO method and not with SEGO. Finally, for Mission 1, 2 and 3, the use of the bi-fidelity MFSEGO method instead of SEGO reduces the computational budget to reach convergence respectively by a factor 2.24, 2.21, 1.32 respectively. For Mission 4, the mono-fidelity method does not even converge for each run. Only 6 runs out of 10 reached the optimum with the allowed budget in the case of the mono-fidelity method, while all runs reached the optimum in the case of the multi-fidelity method. The multi-fidelity method is therefore much more robust.

Table 9 Needed budget to reach convergence criterion for each method and each mission test case.

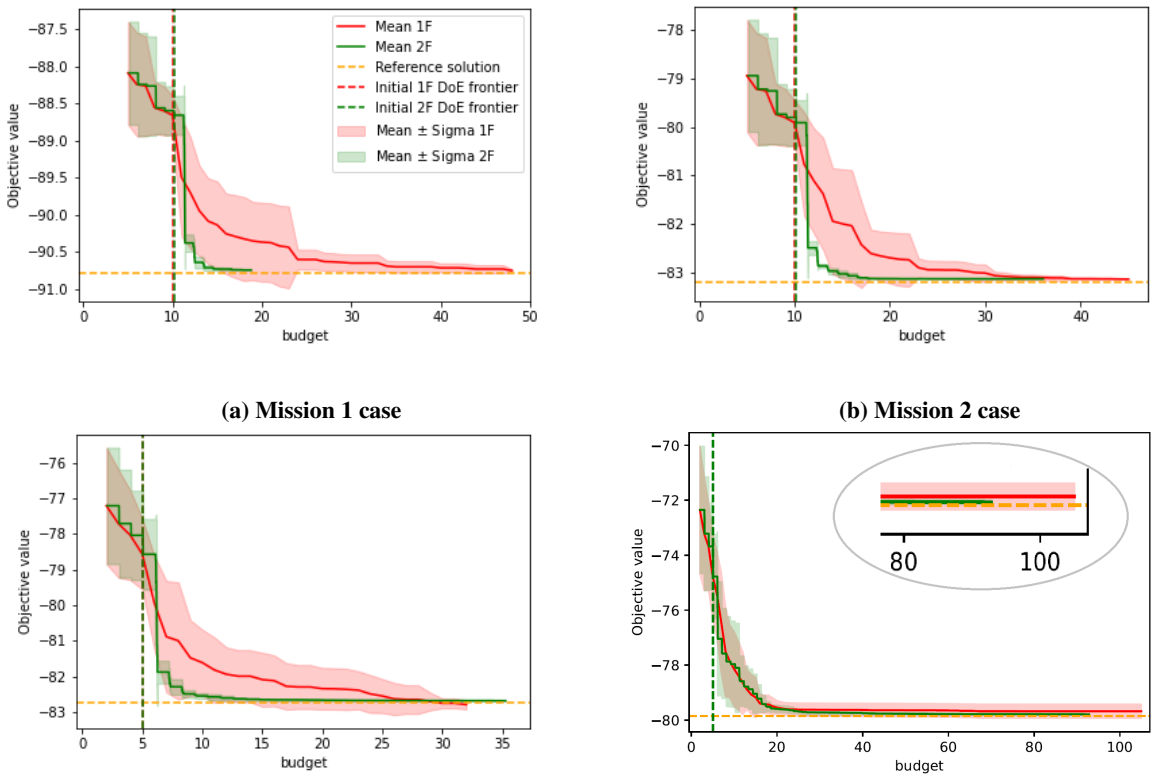
	Mission 1	Mission 2	Mission 3	Mission 4
Mono-fidelity (1F)	30	39	29	Not reached
Multi-fidelity (2F)	13.39	17.65	21.90	30.61

VII. Conclusion

In this paper, we addressed the problem of designing an electric fixed-wing drone for long range surveillance missions. We recall that the methodology applied to solve the optimization problem relies on a multi-fidelity Bayesian optimization. We developed two drone models of different fidelities. The main contribution here is the addition of an electric propulsion module and of a mission module to these drone models. For the low fidelity drone model, a mission component based on the electrical Breguet range equation has been developed while in the high fidelity model, the mission discipline has been tackled using DYMOS, performing optimal control on the mission trajectory. The use of metamodels to approximate the lift and drag coefficients provides sufficient accuracy while greatly reducing the cost of calling Dymos since we no longer have to call OpenAeroStruct many times when solving the optimal control problem. In the end we used both of these models to perform mono and multi-fidelity Bayesian optimizations on four different missions that lead to different drone designs. Finally, to solve the optimization problem, the MFSEGO multi-fidelity algorithm overtakes SEGO mono-fidelity algorithm in terms of needed convergence budget and robustness.

Appendix

In this appendix, Section VII.A presents the MFSEGO methodology in a diagrammatic form. Then, the classical Breguet fuel autonomy equation is recalled in Section VII.B and Section VII.C details how to find the expression of the electric Breguet range equation.



(a) Mission 1 case

(b) Mission 2 case

(c) Mission 3 case

(d) Mission 4 case with a zoom for the last iterations

Fig. 10 Mean and 1-sigma confidence interval over 10 runs of the objective value at the best HF feasible point against budget.

A. Diagram of the multi-fidelity Bayesian optimization methodology based on Le Gratiet's recursive formulation.

Figure 11 is a diagram that explains the multi-fidelity Bayesian optimization methodology with the Le Gratiet's recursive formulation.

B. Classic fuel Breguet range equation

The classic fuel Breguet range equation is given by:

$$R = -\frac{h}{g}\frac{L}{D}\eta_{\text{overall}} \ln \frac{W_{\text{final}}}{W_{\text{init}}} \tag{24}$$

where h is the fuel energy per unit of mass, g is the gravity acceleration, L is the lift force, D the drag force, η_{overall} is the overall efficiency (ie: propulsive power over fuel power), W_{final} is the aircraft final weight and W_{init} is the initial aircraft weight.

C. Electric Breguet range equation demonstration

First note that the range R can be expressed with the speed and the time flying at that speed: $R = V \times t$. Under ideal conditions, the time needed to empty a battery is given by:

$$t = \frac{m_{\text{batt}} e_b}{P_{\text{batt}}} \tag{25}$$

where e_b is the specific energy density of the battery (unit: J.kg^{-1}), m_{batt} is the battery mass and P_{batt} is the battery power. Lets define the overall propulsion system efficiency $\eta_{tot} = \frac{P_{\text{prop}}}{P_{\text{batt}}}$ with P_{prop} is the required propulsive power to

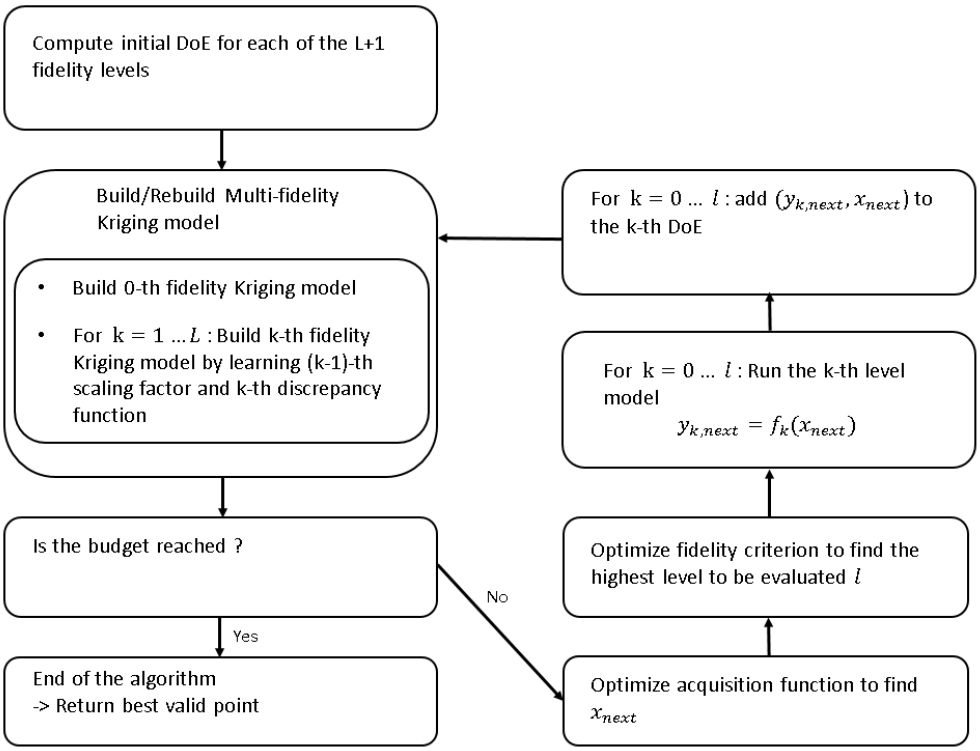


Fig. 11 MFSEGO methodology diagram.

reach the equilibrium. η_{tot} can be expressed as the product of 3 individual efficiencies: η_p the propeller efficiency, η_m the motor efficiency and η_{ESC} the electric speed controller efficiency. $P_{\text{prop}} = TV$ with T the required thrust. Assuming the steady level flight hypothesis, we have

$$\begin{cases} T = D \\ L = W \end{cases} \tag{26}$$

where L is the lift force, D is the drag force and W is the weight. Equation (26) implies

$$T = \frac{L}{\frac{L}{D}} = \frac{W}{\frac{L}{D}} = \frac{mg}{\frac{L}{D}}$$

with g being the gravity acceleration. A relation is now available for the speed:

$$V = \frac{P_{\text{prop}}}{T} = \frac{P_{\text{batt}}\,\eta_{tot}}{\frac{mg}{\frac{L}{D}}} \tag{27}$$

Then the final expression for the range is given by:

$$R = V \times t = \frac{m_{\text{batt}} e_b}{P_{\text{batt}}} \times \frac{P_{\text{batt}}\,\eta_{tot}}{\frac{mg}{\frac{L}{D}}} = \frac{L}{D}\eta_p\eta_m\eta_{ESC}\frac{e_b}{g}\frac{m_{\text{batt}}}{m} \tag{28}$$

Note that the $\frac{L}{D}$ ratio can be replaced by the ratio of lift and drag coefficient $\frac{C_L}{C_D}$. Indeed, $L = qSC_L$ and $D = qSC_D$ with S being the lifting surface area and q being the dynamic pressure.

Acknowledgments

The PhD is funded by the defense innovation agency (AID) and by the french Directorate General of Armaments (DGA) as part of the CONCORDE project. This work is part of the activities of ONERA - ISAE - ENAC joint research group.

References

[1] Martins, J. R. R. A., and Ning, A., *Engineering Design Optimization*, Cambridge University Press, 2020. (to be published).

[2] Frazier, P. I., "A tutorial on Bayesian optimization," *arXiv preprint arXiv:1807.02811*, 2018.

[3] Archetti, F., and Candelieri, A., *Bayesian optimization and data science*, Springer, 2019.

[4] Rasmussen, C. E., and Williams, C., "Gaussian processes for machine learning, vol. 1," , 2006.

[5] Krige, D. G., "A statistical approach to some basic mine valuation problems on the Witwatersrand," *Journal of the Southern African Institute of Mining and Metallurgy*, Vol. 52, No. 6, 1951, pp. 119–139.

[6] Matheron, G., de Géostatistique Appliquée, T., and Tome, I., "Mémoires du Bureau de Recherche Géologiques et Minières, n. 14," *Ed. Technip, Paris*, 1962.

[7] Jones, D. R., Schonlau, M., and Welch, W. J., "Efficient global optimization of expensive black-box functions," *Journal of Global optimization*, Vol. 13, No. 4, 1998, pp. 455–492.

[8] Sasena, M. J., "Flexibility and efficiency enhancements for constrained global design optimization with kriging approximations," Ph.D. thesis, Citeseer, 2002.

[9] Bartoli, N., Lefebvre, T., Dubreuil, S., Olivanti, R., Bons, N., Martins, J. R., Bouhlel, M.-A., and Morlier, J., "An adaptive optimization strategy based on mixture of experts for wing aerodynamic design optimization," *18th AIAA/ISSMO Multidisciplinary Analysis and Optimization Conference*, 2017, p. 4433.

[10] Bartoli, N., Bouhlel, M.-A., Kurek, I., Lafage, R., Lefebvre, T., Morlier, J., Priem, R., Stilz, V., and Regis, R., "Improvement of efficient global optimization with application to aircraft wing design," *17th AIAA/ISSMO Multidisciplinary analysis and optimization conference*, 2016, p. 4001.

[11] Chiles, J.-P., and Delfiner, P., *Geostatistics: modeling spatial uncertainty*, Vol. 497, John Wiley & Sons, 2009.

[12] Wackernagel, H., *Multivariate geostatistics: an introduction with applications*, Springer Science & Business Media, 2003.

[13] Kennedy, M. C., and O'Hagan, A., "Bayesian calibration of computer models," *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, Vol. 63, No. 3, 2001, pp. 425–464.

[14] Le Gratiet, L., "Multi-fidelity Gaussian process regression for computer experiments," Ph.D. thesis, Université Paris-Diderot-Paris VII, 2013.

[15] Bachoc, F., "Cross validation and maximum likelihood estimations of hyper-parameters of Gaussian processes with model misspecification," *Computational Statistics & Data Analysis*, Vol. 66, 2013, pp. 55–69.

[16] Pavlyuk, D., "Computing the maximum likelihood estimates: concentrated likelihood, EM-algorithm," ????

[17] Charayron, R., Lefebvre, T., Bartoli, N., and Morlier, J., "Multi-fidelity constrained Bayesian optimization, application to drone design," *CAID 2021 Forum*, 2021.

[18] Meliani, M., Bartoli, N., Lefebvre, T., Bouhlel, M.-A., Martins, J. R., and Morlier, J., "Multi-fidelity efficient global optimization: Methodology and application to airfoil shape design," *AIAA aviation 2019 forum*, 2019, p. 3236.

[19] Gray, J. S., Hwang, J. T., Martins, J. R., Moore, K. T., and Naylor, B. A., "OpenMDAO: An open-source framework for multidisciplinary design, analysis, and optimization," *Structural and Multidisciplinary Optimization*, Vol. 59, No. 4, 2019, pp. 1075–1104.

[20] Lafage, R., Defoort, S., and Lefebvre, T., "WhatsOpt: a web application for multidisciplinary design analysis and optimization," *AIAA Aviation 2019 Forum*, 2019, p. 2990. https://doi.org/10.2514/6.2019-2990, URL https://doi.org/10.2514/6.2019-2990.

[21] Jasa, J. P., Hwang, J. T., and Martins, J. R., "Open-source coupled aerostructural optimization using Python," *Structural and Multidisciplinary Optimization*, Vol. 57, No. 4, 2018, pp. 1815–1827.

[22] Chauhan, S. S., and Martins, J. R., "Low-fidelity aerostructural optimization of aircraft wings with a simplified wingbox model using OpenAeroStruct," *International Conference on Engineering Optimization*, Springer, 2018, pp. 418–431.

[23] Chaudhuri, A., Jasa, J., Martins, J. R., and Willcox, K. E., "Multifidelity optimization under uncertainty for a tailless aircraft," *2018 AIAA Non-Deterministic Approaches Conference*, 2018, p. 1658.

[24] Falck, R., Gray, J. S., Ponnapalli, K., and Wright, T., "dymos: A Python package for optimal control of multidisciplinary systems," *Journal of Open Source Software*, Vol. 6, No. 59, 2021, p. 2809. https://doi.org/10.21105/joss.02809, URL https://doi.org/10.21105/joss.02809.

[25] Priem, R., Bartoli, N., Diouane, Y., Lefebvre, T., Dubreuil, S., Salaün, M., and Morlier, J., "SEGOMOE: Super Efficient Global Optimization with Mixture of Experts," *Workshop CIMI Optimization & Learning*, 2018.

[26] Bartoli, N., Lefebvre, T., Dubreuil, S., Olivanti, R., Priem, R., Bons, N., Martins, J. R., and Morlier, J., "Adaptive modeling strategy for constrained global optimization with application to aerodynamic wing design," *Aerospace Science and technology*, Vol. 90, 2019, pp. 85–102.

[27] Vauclin, R., "Développement de modèles réduits multifidélité en vue de l'optimisation de structures aéronautiques," *Rapport Institut Supérieure de l'Aéronautique et de l'Espace–École Nationale Supérieure des Mines de Saint-Etienne*, 2014.

[28] Bouhlel, M. A., Bartoli, N., Otsmane, A., and Morlier, J., "Improving kriging surrogates of high-dimensional design models by Partial Least Squares dimension reduction," *Structural and Multidisciplinary Optimization*, Vol. 53, No. 5, 2016, pp. 935–952.

[29] Ng, K. S., "A simple explanation of partial least squares," *The Australian National University, Canberra*, 2013.

[30] Powell, M. J., "A direct search optimization method that models the objective and constraint functions by linear interpolation," *Advances in optimization and numerical analysis*, Springer, 1994, pp. 51–67.