RÉPUBLIQUE FRANÇAISE
Liberté
Égalité
Fraternité

ONERA
THE FRENCH AEROSPACE LAB

ISAE
Institut Supérieur de l'Aéronautique et de l'Espace
SUPAERO

Master of Science in Aerospace Engineering

END-OF-STUDY PROJECT REPORT

A Machine Learning Based Approach For Uncertainty Quantification, Application To Launch Vehicle Design

Jorge L. Valderrama

supervised by:

Dr. Loïc Brevault
DTIS, ONERA

Dr. Mathieu Balesdent
DTIS, ONERA

Starting date of project: April 6, 2021
Submission date: September 1, 2021

Acknowledgments

I would like to give my warmest thanks to Dr. Loïc Brevault and Dr. Mathieu Balesdent. Their guidance was instrumental for the main part of my master studies. First for my master's research project, and then for my internship period at ONERA.

Special thanks to my family, for their love and support that always accompanied me in this journey in France.

Thank you also to the interns and PhD students *du premier et du deuxième étage*. It was a pleasure to exchange with them during the breaks, lunch and sport activities.

Finally, thank you to my friend Tom Lawson for the fruitful and fun evening discussions about Kriging and rockets.

Declaration of authenticity

This assignment is entirely my own work. Quotations from literature are properly indicated with appropriate references in the text. All literature used in this piece of work is indicated in the bibliography placed at the end. I confirm that no sources have been used other than those stated.

I understand that plagiarism (copy without mentioning the reference) is a serious examination offense that may result in disciplinary action being taken.

Signature:  Date: 31/08/2021

Jorge Luis Valderrama

Contents

List of Figures

List of Tables

List of abbreviations

AAO	: All-At-Once
CFD	: Computational Fluid Mechanics
CMA-ES	: Covariance Matrix Adaptation Evolution Strategy
DoE	: Design of experiments
DTIS	: Département de Traitement de l'Information et Systèmes
ETO	: Elliptic Transfer Orbit
FEA	: Finite Element Analysis
GLOW	: Gross Lift-Off Weight
GP	: Gaussian Process
ISAE-SUPAERO	: Institut Supérieur de l'Aéronautique et de l'Espace
KL	: Karhunen-Loève
LEO	: Low Earth Orbit
LGL	: Legendre-Gauss-Lobatto
LOO	: Leave-one-out
MCS	: Monte-Carlo Simulation
MDA	: MultiDisciplinary Analysis
MDAO	: MultiDisciplinary Design, Analysis and Optimization
M2CI	: Méthodes Multidisciplinaires et Concepts Intégrés
NLP	: Non-Linear Programming
ONERA	: Office National d'Etudes et de Recherches Aérospatiales
ODM-CMC	: ONERA Damage Model for Composites with Ceramic Matrix
PCE	: Polynomial Chaos Expansion
RBDO	: Reliability-Based Design Optimization
RBF	: Radial Basis Function
RK	: Runge-Kutta
SVM	: Support Vector Machine
TSTO	: Two-Stage-To-Orbit
UQ	: Uncertainty Quantification

Nomenclature

A	: Confidence interval area
$\mathcal{A}$	: σ -algebra
$C_{\mathbf{XX}}(\cdot,\cdot)$	: Autocovariance of the stochastic process $\boldsymbol{X}$
$\boldsymbol{g}(\cdot)$	: Inequality constraints
$\boldsymbol{h}(\cdot)$	: Equality constraints
$J(\cdot)$	: Performance index
$k^{\theta}(\cdot,\cdot)$	: Kernel or covariance model of a Gaussian process
$\mathcal{L}_k(\cdot)$	: k^{th} eigenfunction from a Karhunen-Loève (KL) expansion
l	: Length parameter of a kernel
M	: Size of training sample
N	: Size of time mesh $\mathcal{T}_N$
N_k	: Number of modes included in the truncation of a KL expansion
$N_{GP_{traj}}$	: Number of Gaussian process trajectories
$N_{maxiter}$	: Maximum number of iterations of the active learning algorithm
$N_{maxiter_{opt}}$	: Maximum number of iterations for the optimizer
$N_{popsize}$	: Number of individuals for the CMA-ES algorithm

P : Dimension of input uncertain vector $\boldsymbol{U}$
$\mathbb{P}$: Probability measure
q : Dimension of state variable $\boldsymbol{x}$
q_η : η - quantile
Q_{X^*} : MDAO with optimal control application
$\mathcal{Q}$: Set containing $N_{GP_{traj}}$ estimated quantiles
Q_2 : Predictivity factor
$\boldsymbol{R}(\cdot)$: Disciplinary residuals
R : Size of quantile estimation sample
t : Time
$\mathcal{T}$: Time domain
$\mathcal{T}_N$: Time mesh of size N
$\boldsymbol{u}$: Realization of input uncertain vector $\boldsymbol{U}$
$\boldsymbol{U}$: Input uncertain vector
$\mathcal{U}_M = [\mathbf{u}_1, \ldots, \mathbf{u}_M]$: Training input sample of size M
$\mathcal{V}_R = [\mathbf{u}_1, \ldots, \mathbf{u}_R]$: Quantile estimation input sample of size R
$\mathcal{V}_V = [\mathbf{u}_1, \ldots, \mathbf{u}_V]$: Validation input sample of size V
V : Size of validation sample
$\boldsymbol{w}$: Trajectory control variable vector
$\boldsymbol{x}$: State variable vector
$\boldsymbol{X}$: State variable stochastic process
$\boldsymbol{X}^*$: Optimal state variable stochastic process
$\boldsymbol{X}^*_M = [\boldsymbol{X}^*_1, ..., \boldsymbol{X}^*_M]$: Training output sample of size M
$\boldsymbol{X}^*_V = [\boldsymbol{X}^*_1, ..., \boldsymbol{X}^*_R]$: Validation output sample of size R
$\boldsymbol{X}_t$: Marginal of state variable stochastic process $\boldsymbol{X}$
$\boldsymbol{X}(\boldsymbol{u})$: Trajectory of state variable stochastic process $\boldsymbol{X}$
$\tilde{\boldsymbol{X}}$: Snapshots matrix
$\boldsymbol{y}$: Coupling variables
$\mathcal{Y}_{k_M} = \xi_k(\mathcal{U}_M)$: Output sample of KL decomposition for mode $k \in [1, .., N_k]$
$\boldsymbol{z}$: Architectural design variables
α : Confidence level of quantile estimation
β : β - quantile of a given set
ϵ_{xx} : Strain for the ODM-CMC
σ : Amplitude parameter of a kernel
σ_{xx} : Stress for the ODM-CMC
λ_k : k$^{\text{th}}$ eigenvalue resulting from a KL expansion
ξ_k : k$^{\text{th}}$ uncertain variable resulting from a KL expansion
$\phi_{\mathbf{U}}(\cdot)$: Joint probability density function of $\boldsymbol{U}$
Ω : Definition domain of $\mathbf{U}$
$\psi(\cdot)$: Cumulative density function
μ_t : Mean of stochastic process marginal $\boldsymbol{X}_t$
$\mu(t)$: Mean realization of $\boldsymbol{X}$
$\mu(\cdot)$: Trend or mean of a Gaussian process
ν : Parameter of Matérn kernel
θ : Set of parameters of a kernel
Υ : Universal set

Abstract

The coupling of uncertainty quantification methodologies with multidisciplinary optimization tools for the early design phase of aerospace vehicles is computationally intensive as it also involves strategies for multidisciplinary coupling satisfaction and optimal control through the trajectory discipline. The early design phase is characterized by a high number of input uncertain variables (*e.g.,* specific impulse, drag coefficient) that render uncertain the output fields (*e.g.,* the speed profile as function of time and the pressure distribution on aerodynamic surfaces). The output fields are comprised of a high number of correlated aleatory variables that make even more daunting the uncertainty quantification task. This work presents an active learning methodology for field variable quantile estimation relying on a surrogate model to reduce the computational cost. Two refinement criteria based on the propagation of the variance of Gaussian process regressors through the Karhunen-Loève decomposition of the field are proposed. An example case is demonstrated for the quantile estimation of the resulting state variables from the multidisciplinary optimization of a two-stage-to-orbit vehicle. The two refinement criteria methodologies improve the accuracy of the predicted quantiles and outperform an aleatory enrichment strategy.

Keywords: Uncertainty quantification, Launch vehicle design, Optimal control, Karhunen-Loève, Gaussian process/Kriging, Active learning

1 Introduction

The design of an aerospace vehicle is a complex task due to the coupled interactions between multiple disciplines, like trajectory, structure, propulsion, aerodynamics, *etc.* At the early design phase, low-fidelity models are used in general to explore a broad design space where very different solutions coexist. The determination of the best concepts according to the given design specifications is perturbed by the modeling uncertainties (*e.g.,* numerical approximations, physical simplification) inherited from the low fidelity models (*e.g.,* low number of nodes used for the discretization of the trajectory) and the uncertainty on input variables (*e.g.,* strength of materials, rocket engine characteristics). For example, a launch vehicle architecture that seems like a good solution but that approaches a design specification constraint during the early design phase could happen to violate that constraint during more detailed design phases (using higher fidelity models), thus being discard as it fails to comply with the design specifications. Such constraints could be a trajectory threshold on the axial load, the dynamic pressure or visibility criteria from ground stations. Uncertainty Quantification (UQ) is used in launch vehicle design to characterize and reduce the errors associated to uncertainty modeling in the early design phases. Uncertain variables describing modeling uncertainties may be expressed using probability formalism and sampling-based techniques (*e.g.,* Crude Monte Carlo) to propagate the uncertainty through a given model and assess the overall uncertainty on its response.

Multidisciplinary Design, Analysis and Optimization (MDAO) is a methodology that is used to account for the interactions between the different disciplines involved in complex systems design, such as launch vehicles, and look for an optimal solution of the system as a whole. MDAO looks to replace traditional design approaches where individual optimization of each discipline is performed without including their interactions and resulting on a burdensome iterative process of exchanges between the different design offices in charge of each discipline. MDAO framework for launch vehicle design is computationally intensive as it not only solves for complex models for each discipline, that can include Computational Fluid Mechanics (CFD) for the estimation of aerodynamic characteristics, Finite Element Analysis (FEA) for the simulation of structural behavior and trajectory optimization, but it also looks to satisfy the couplings that represent

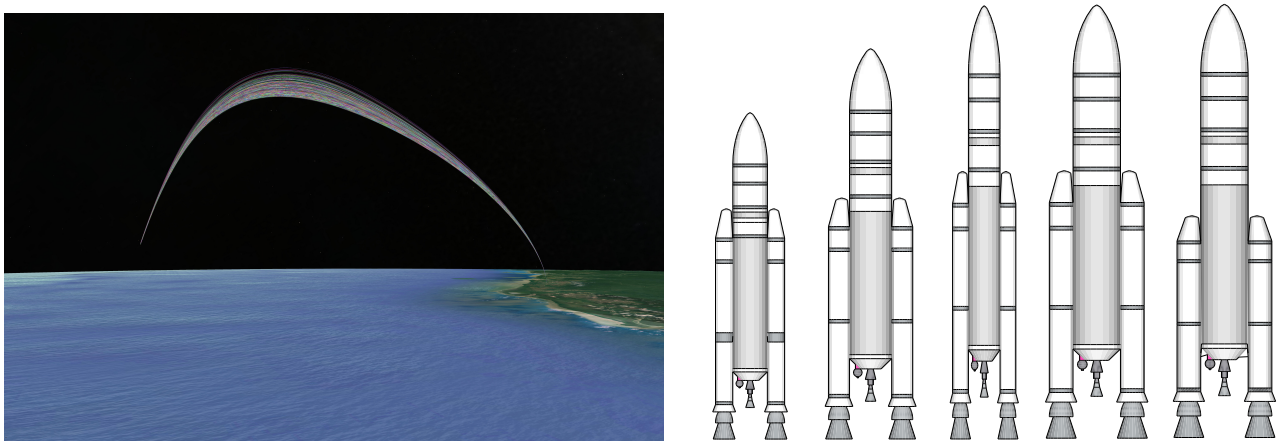


Figure 1: Representation of uncertain optimal ascent trajectory for a vehicle designed to launch from the Guiana Space Center (left) and multiple architectures of an Ariane-5-type launcher for different geometrical parameters (right).

the interactions between those disciplines. Depending on the chosen MDAO formulation, the satisfaction of the multidisciplinary couplings inside the optimization loop can require an iterative MultiDisciplinary Analysis (MDA), that includes for instance the simulation of the trajectory and the structure disciplines (the loads that the structures must withstand vary with the trajectory). In decoupled MDAO formulations, such an iterative MDA loop is not required, but a cost must be incurred at including extra optimization variables and constraints that render the optimization problem more difficult to solve.

Trajectory optimization is central to launch vehicle MDAO as the optimal control techniques that are used to find the trajectories with the best performance index represent a large portion of the computational cost. Furthermore, the trajectory discipline ensures that the solutions satisfy the mission constraints, such as the desired target orbit. Including uncertainty quantification into the assessment of launch vehicle performance is computationally challenging as it requires to combine the already expensive MDAO with the UQ methodology (Fig. 2). The aim of this work consists in developing a reduced-cost strategy for uncertainty propagation in multidisciplinary systems containing an optimal control problem. To achieve this, the following objectives are set:

- To review the state of the art of active learning and machine learning applications for the estimation of quantiles of field variables.
- To develop a methodology for the refinement of a surrogate model (active learning) for the estimation of field variable quantiles using Gaussian processes and model order reduction methods.
- To apply the methodology on a test case for the multidisciplinary optimization of a launch vehicle.

Different approaches were described in [2] to deal with uncertainty propagation in the multidisciplinary launch vehicle design problem involving optimal control and a new methodology based on model order reduction and spectral methods was introduced. This approach consists on the exploitation of a surrogate model of the MDAO problem involving optimal control. Surrogate models are advantageous because they allow to predict quantities of interest at a reduced computational cost. Nevertheless, the accuracy of the prediction for surrogate models based on

spectral methods (as Polynomial Chaos Expansions) can only be assessed when a large enough set of validation data is available or by iteratively training the model in subsets of the data and evaluating performance in the remaining samples. In a first step, this internship work intends to extend the approach presented in [2] to couple model order reduction methods and Gaussian processes. Gaussian processes (also called Kriging) are a Bayesian approach in which the predicted outputs take the form of a normal distribution instead of deterministic values. Hence, its results can be interpreted as a mean prediction with an associated variance error model. Such an error can be reduced by improving the quality of the training set (also referred as Design of experiments - DoE) that is used to create the surrogate model, and with active learning approaches used to identify the samples that can contribute the most to the error reduction when they are added to the training set. In [3] and [4], active learning techniques based on Gaussian processes were used to improve the estimation of quantiles of scalar variables. To the best of my knowledge, no information of an equivalent technique for the estimation of quantiles of field variables (such as a launcher trajectory or a pressure field on a aerodynamic surface) has been found in the literature. In a second step, this work looks to develop an active learning technique for the surrogate model, based on model order reduction methods and Gaussian processes to estimate quantiles of field variables, while controling the associated error produced by the use of the surrogate model instead of the exact model. Such a methodology enables the quantification of extreme probability quantiles of trajectory state variables at a reduced computational cost. An application case of the proposed active learning technique is done to predict extreme quantiles of the optimal trajectory resulting from the MDAO of a Two-Stage-To-Orbit (TSTO) vehicle, designed to inject 10 tons of payload into a Low Earth Orbit (LEO). The launch vehicle MDAO is performed using a variation of an All-At-Once formulation coupled with pseudospectral methods for trajectory optimization that was developed during my master's research project at ISAE-SUPAERO [5].

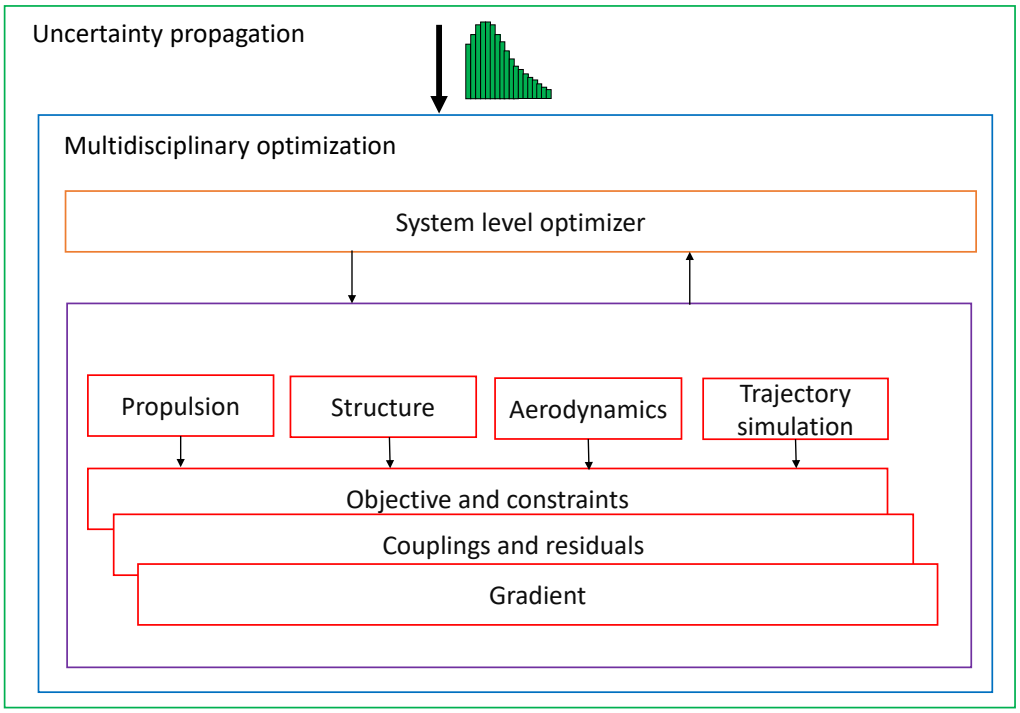


Figure 2: Scheme of uncertainty propagation for launch vehicle MDAO

This report is structured as follows: Section 2 contains a brief description of ONERA and the specific department and unit where I did my internship. Section 3 is a literature review addressing the main methodologies to perform uncertainty quantification on problems that involve optimal control, focusing on the surrogate model strategy. The description of a model order reduction method (the Karhunen-Loève expansion) is done followed by Gaussian processes for

surrogate modeling. The use of the later methodology in the context of active learning is also described. Section 4 contains the description of the launch vehicle MDAO code that was used in the test case and a five step process detailing the proposed active learning strategy for field variable quantile estimation. The results for the estimated quantiles of trajectory state variables and quantities of interest resulting from the launch vehicle MDAO are presented in section 5.

2 About ONERA and the work environment

This work is the result of my internship at ONERA (from the French name Office National d'Etudes et de Recherches Aérospatiales). This internship had place between the 6^{th} of April and the 31^{st} of August of 2021 at the research center located in Palaiseau, in the Île-de-France region of France under the guidance of Dr. Loïc Brevault and Dr. Mathieu Balesdent.

The origins of ONERA date back to 1946 when the French state decided to merge all of the aeronautical institutions being sponsored by the country into a unique research center, whose mission was to boost a modern aviation in France in response to a diminished industry that suffered during World War II. The research center was born around the operation of wind tunnels, and nowadays, it has become its largest operator in Europe. Dedicated not only to aeronautics but also to the defense and space sectors, ONERA is comprised of 7 scientific departments covering topics on aerodynamics, electromagnetism, materials science, multi-physics, optics, space instrumentation and information processing. ONERA groups around two thousand employees where the majority are engineers and researchers, working in 8 different sites distributed over the French territory.

The DTIS department (from the French name Département de Traitement de l'Information et Systèmes) is dedicated to signal and image processing, robotics, numerical simulation, design and automation of systems and processes, information systems engineering, and knowledge and cognitive engineering in the fields of aeronautics, defense and space. The M2CI research unit (from the French name Unité Méthodes Multidisciplinaires et Concepts Intégrés) is part of the 9 different units of the DTIS and is the unit that welcomed me for the development of my end-of-study internship. The M2CI unit is dedicated to the development of design strategies including Multidisciplinary Design, Analysis and Optimization (MDAO), methodologies for requirements management, systems engineering, surrogate modeling, multi-fidelity, reliability, uncertainty quantification and integrated design methods. The application of these methodologies inside the M2CI unit covers new concepts for civil aviation like blended body wings, hybrid propulsion aircraft, supersonic aircraft and boundary layer ingestion. It also wraps new concepts for launch vehicles like air-launching, and expendable and reusable winged launch vehicles.

During my internship, I worked under the guidance of Dr. Loïc Brevault and Dr. Mathieu Balesdent, two researchers from the M2CI unit with a combined experience of more than 20 years on the MDAO and UQ of launch vehicles and with whom I sustained meetings two times per week to analyze results, verify progress, troubleshoot and plan future actions. This work extends the research presented by Dr. Loïc Brevault and Dr. Mathieu Balesdent in [2], on a methodology for uncertainty quantification for multidisciplinary launch vehicle design using surrogate models and the application case is based on my master's research project at ISAE-SUPAERO (from the French name Institut Supérieur de l'Aéronautique et de l'Espace), developed under the guidance of the same supervisors and Dr. Annafederica Urbano from ISAE-SUPAERO. That work was presented in [5].

3 Literature Review

Following the work in [2], 3 main approaches can be distinguished to deal with uncertainty propagation in the multidisciplinary launch vehicle design problem involving optimal control (Fig. 3). The first one [6] [7] [8] uses Monte Carlo Simulation (MCS) on top of the MDAO problem, generating samples of the input uncertain variables (*e.g.,* engine specific impulse, mass flow rate, drag coefficient) and solving the MDAO problem for each sample. This leads to exact state variable realizations (*e.g.,* velocity, altitude) and quantities of interest (*e.g.,* dynamic pressure, heat flux, *etc.*). Such a nested approach can easily become prohibitive due to the associated computational cost. This is because MDAO tools can take several hours or days to solve for one case sample (due to repeated discipline evaluations that can involve advanced simulation tools such as FEM and CFD calculations) and the size of a UQ sample is measured in hundreds or even thousands. The second approach [9] [10] [11] [12] consists on the decomposition of the optimal control problem using spectral methods (*e.g.,* Polynomial Chaos Expansion - PCE), modifying the original MDAO problem to create an extended design space and solve a unique optimization problem. Typically this method is only used for a few uncertain variables (3 or less), given that the complexity of the optimization problem scales in a fast manner as of function of the number of uncertain variables [9] [11] [12]. The final approach was proposed in [2] and consists on the creation of a surrogate model based on model order reduction and spectral methods that can be exploited at a reduced computational cost.

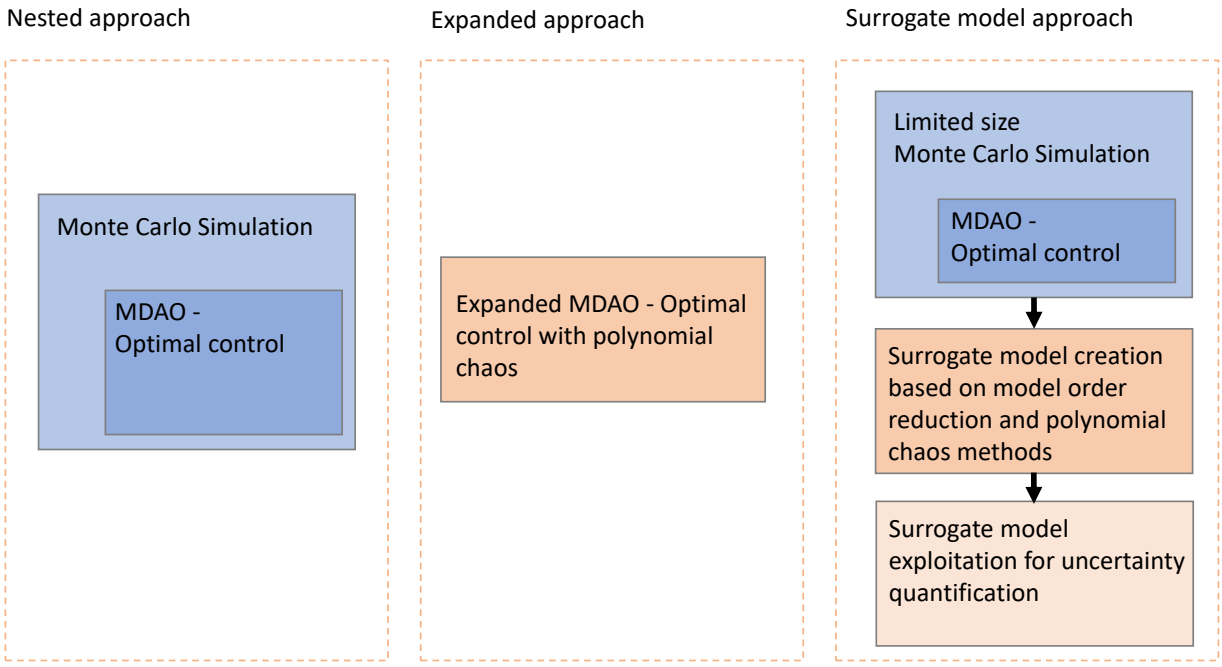


Figure 3: Existing approaches for uncertainty quantification with optimal control

Surrogate models are commonly used in design optimization applications in which a high number of evaluations of a computationally expensive function is required. They look to mimic the behavior of the original function based on the relationship between known input variables and their model responses. The surrogate model capacity to predict accurate responses depends on the quality and the quantity of information that is used to train it, and the formulation of the model itself. The quality of the predictions of techniques such as Polynomial Chaos Expansions (PCE), Neural networks and Support Vector Machines (SVM) can be assessed when a large enough amount of validation data is available or by training the same model on different subsets of a small data set and evaluating the error for each iteration (*e.g.,* Leave-one-out - LOO technique)

on the samples that do not belong to the training subset. In the launch vehicle MDAO problem, it is difficult to obtain large amounts of data due to the high computational cost of the models that are used. A different surrogate model method called Gaussian Processes (GP) or Kriging has an appealing characteristic for the context of small data that consist of an error model for its predictions, based on Gaussian variables. The output variables take the form of Gaussian random variables whose variance give a measure of how certain the surrogate model prediction is. This characteristic can be used to improve the model by identifying new samples that reduce the variance and adding them to the dataset. This is called an active learning technique. In the context of uncertainty quantification, active learning techniques based on Gaussian processes have been proposed for instance to improve the estimation of scalar variable quantiles [3] [4].

To facilitate the discussion, let us define an MDAO problem including optimal control and the probability formalism necessary to describe uncertainty propagation. In this work, the MDAO of the launch vehicle is carried out using an All-At-Once-like (AAO-like) MDAO formulation where the feed-forward couplings between the disciplines are kept and only the feed-back couplings are replaced by extra optimization variables and optimization constraints. The trajectory is an analysis discipline whose solving is carried out using an optimization problem and a transcription based on the Legendre-Gauss-Lobatto (LGL) orthogonal collocation technique of order 3. The concepts of state discretization nodes, collocation nodes, defects and defect constraints are necessary to formulate this MDAO problem and are briefly described in appendix A. A complete description can be found in [1]. The dynamics of the trajectory are ruled by a set of coupled ordinary differential equations describing the motion and mass changes of the vehicle in polar coordinates. In this approach, the architectural design variables (*e.g.,* stage diameters, chamber pressures, engine thrust at vacuum), the trajectory optimization static variables (*e.g.,* duration of the phases, parameterized pitch angle parameters) and the state discretization nodes values (*e.g.,* values of mass, velocity and altitude at every state discretization node) are handled at the same level by the same Non-Linear Programming (NLP) optimizer. The MDAO problem can be written as:

$$\text{minimize } J(\boldsymbol{X}(t), \boldsymbol{y}, \boldsymbol{z}, \boldsymbol{w}, \boldsymbol{U}) \tag{3.1}$$
$$\text{with respect to } \boldsymbol{X}(t), \boldsymbol{y}, \boldsymbol{z}, \boldsymbol{w} \tag{3.2}$$
$$\text{subject to:} \tag{3.3}$$
$$\text{inequality constraints } \boldsymbol{g}(\boldsymbol{X}(t), \boldsymbol{y}, \boldsymbol{z}, \boldsymbol{w}) \leq \boldsymbol{0} \tag{3.4}$$
$$\text{equality constraints } \boldsymbol{h}(\boldsymbol{X}(t), \boldsymbol{y}, \boldsymbol{z}, \boldsymbol{w}) \leq \boldsymbol{0} \tag{3.5}$$
$$\text{residuals } \forall i, \boldsymbol{R}_i\left(\boldsymbol{X}(t)_i, \boldsymbol{y}_i, \boldsymbol{z}_i\right) = \boldsymbol{0} \tag{3.6}$$
$$\text{coupling variables } \forall i, \forall j \neq i, \boldsymbol{y}_{ji} = c_{ji}\left(\boldsymbol{X}(t)_j, \boldsymbol{y}_j, \boldsymbol{z}_j\right) \tag{3.7}$$

where J is a performance function driving the optimization problem (*e.g.,* the Gross Lift-Off Weight - GLOW), $\boldsymbol{X}$ is the state variables vector representing the values of the states (*e.g.,* mass, speed, latitude) at the discretization nodes of the LGL transcription. These variables are optimization variables. $\boldsymbol{y}$ is the input coupling variables vector (*e.g.,* stage dry mass, specific impulse) between the i^{th} and the j^{th} disciplines, $\boldsymbol{z}$ is the architectural design variables vector, and $\boldsymbol{U} \in \mathbb{R}^P$ is considered as a constant vector for the moment and will be used to represent the uncertainties later. The problem is constrained by the set of inequality and equality constraints, $\boldsymbol{g}$ and $\boldsymbol{h}$ respectively. In this case, the residuals $\boldsymbol{R}$ for the i^{th} discipline are only defined for the trajectory discipline. They are equivalent to the defect constraints of the orthogonal collocation transcription (see appendix A). The trajectory control variables $\boldsymbol{w}$ correspond to a parameterized pitch angle guidance program and the duration of the different phases (*e.g.,* lift-off, pitch over,

exo-atmospheric). Due to the orthogonal collocation transcription, this optimization problem is high-dimensional and is a perfect suit for modern NLP solvers.

Based on the formalism introduced in [2], let us now consider $\mathbf{U}$ as an aleatory vector defined over a probability space $(\Upsilon, \mathcal{A}, \mathbb{P})$ with Υ the universal set, $\mathcal{A}$ a σ -algebra and the probability measure $\mathbb{P}$. This allows to introduce uncertainties in the launch vehicle MDAO problem by defining $\mathbf{U}$ with a joint probability density function $\phi_{\mathbf{U}}(\cdot)$ on its definition domain Ω. In early design phase for instance, the uncertain variables can include the residual mass of propellants, the mass flow rate of the engines or the drag coefficient of the vehicle. The MDAO problem is solved for different realizations of $\mathbf{U}$, implying that each solution takes different values of $\boldsymbol{X}(t), \boldsymbol{y}, \boldsymbol{z}$ and $\boldsymbol{w}$. Hence, the state variable vector becomes a stochastic process that can be noted as $\boldsymbol{X} : \mathcal{T} \times \Omega \to \mathbb{R}^q$ corresponding to a multivariate stochastic process of the state variables of dimension q with $\mathcal{T}$ the time domain and $\boldsymbol{X}(t, \mathbf{U}) \in \mathbb{R}^q$.

The concepts of marginal and trajectory or realization are useful to describe and understand stochastic processes. Let us define the additional notations and vocabulary required for the proposed approach. $\boldsymbol{X}_t : \Omega \to \mathbb{R}^q$ is the random variable vector at time $t \in \mathcal{T}$ defined by $\boldsymbol{X}_t(\mathbf{u}) = \boldsymbol{X}(t, \mathbf{u})$. If the marginal distribution of $\boldsymbol{X}_t, \forall t$ is Gaussian (Fig. 5) and the covariance function of the marginals corresponds to that of a joint Gaussian distribution, the stochastic process is called a Gaussian process. For a given realization of $\boldsymbol{U}$, a trajectory of the stochastic process (Fig. 4), $\boldsymbol{X}(\mathbf{u}) : \mathcal{T} \to \mathbb{R}^q$, is defined as $\boldsymbol{X}(\mathbf{u})(t) = \boldsymbol{X}(t, \mathbf{u})$. To define the model order reduction methods, it is convenient to address the joint variability of the marginals at two different time instants $(t_1, t_2 \in \mathcal{T}^2)$ using the autocovariance of the stochastic process $C_{\mathbf{XX}}(t_1, t_2) = \mathrm{cov}[\boldsymbol{X}_{t1}, \boldsymbol{X}_{t2}] = \mathbb{E}[(\boldsymbol{X}_{t1} - \mu_{t1})(\boldsymbol{X}_{t2} - \mu_{t2})]$ where μ_{ti} is the mean associated to the stochastic process.

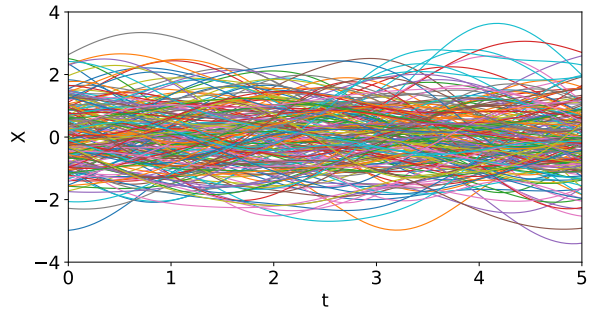


Figure 4: 200 realizations or trajectories of a Gaussian process

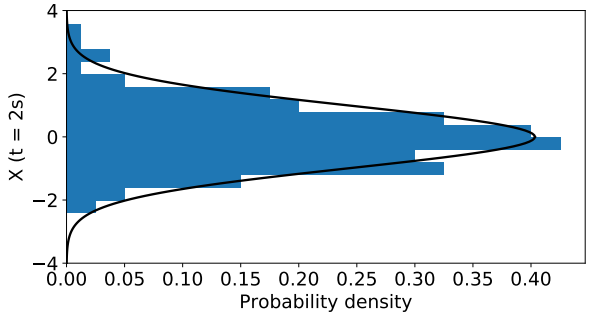


Figure 5: Marginal distribution for $t = 4$ based on 200 realizations

Given the probability density function of the input random vector $\phi_{\mathbf{U}}(\cdot)$ and an application $Q_{\boldsymbol{X}^*}$ describing the MDAO of the launch vehicle including optimal control, it is possible to obtain the stochastic process describing the optimal states $\boldsymbol{X}^*(t, \mathbf{U})$, formally the application is defined as

$$\mathbf{U} \sim \phi_{\mathbf{U}} \xrightarrow{Q_{X^*}} \boldsymbol{X}^*(t, \mathbf{U}) \tag{3.8}$$

The optimal values of the state variables (or derived quantities of interest) are obtained by solving the MDAO problem which includes the optimal control problem defined for the trajectory discipline. For simplicity, let us consider that $\boldsymbol{X}^*(t, \mathbf{U})$ represents only one state to facilitate the writing of the remaining of this report.

The surrogate model approach to perform uncertainty quantification shown in Fig. (3) requires to generate M samples of the input variable probability density function $\phi_{\mathbf{U}}$. Hence,

an input vector $\mathcal{U}_M = [\boldsymbol{u}_1, ..., \boldsymbol{u}_M]$ is created over which the MDAO application Q_{X^*} is executed to obtain the optimal state realizations $\boldsymbol{X}^*(t, \mathbf{u}_k), k \in [1, ..., M]$. The input vector $\mathcal{U}_M$ and its model responses are used to train the surrogate model(they constitute the Design of Experiments, also called training set). Each of the stochastic process realizations is discretized using a common mesh of N vertices for the time $\mathcal{T}_N$, in such a way that values of the stochastic process can be read at every $[\boldsymbol{X}(t_1, \mathbf{u}_k), \ldots, \boldsymbol{X}(t_N, \mathbf{u}_k)]$. As a result, the discretized stochastic process is comprised of N correlated random variables.

The creation of a surrogate model with a given technique can imply various underlying methodologies that lead to different characteristics of the metamodel. For example, in a Gaussian process regressor there exist different types of covariance kernels, each of them leads to a unique way of linking the training data to the predictions. In what follows of this section, some of the methodologies involved in the creation of surrogate models will be described as found in the literature. Finally, the active learning technique for quantile estimation will be addressed.

3.1 Karhunen-Loève (KL) expansion

Given that a fine enough mesh is desired to capture the dynamic behavior of the trajectory state variables, the discretization of the stochastic process leads to a high number of vertices (N around 200 in the present case, but in more complex problems it can go up to millions). The use of individual surrogate models to predict each of the N random variables would become burdensome, thus it is desire to reduce the dimensionality of the problem. In the surrogate model approach of [2], the Karhunen-Loève expansion was used for this purpose.

The Karhunen-Loève expansion of the random process $\boldsymbol{X}^*(t, \mathbf{U})$ is represented in the form of an infinite sum of orthogonal functions, similarly to a Fourier series representation. The expansion is based on the spectral decomposition of the auto-covariance function of the random process $C_{\mathbf{X}^*\mathbf{X}^*}(\cdot, \cdot)$ [13] or $C(\cdot, \cdot)$ for simplicity. In a similar way to Principal Component Analysis, the KL expansion looks to determine the most significant eigenvalues and their associated eigenfunctions to then project the information into the space that they form. The eigenvalues and eigenfunctions are determined by solving the second kind Fredholm equation:

$$\int_{\mathcal{T}} C(s,t)\mathcal{L}_k(t)\mathrm{d}t = \lambda_k \mathcal{L}_k(s) \quad \forall s \in \mathcal{T} \tag{3.9}$$

where $(\lambda_k)_{k\geq 1}$ are the eigenvalues and $(\mathcal{L}_k)_{k\geq 1}$ the associated eigenfunctions. $(\mathcal{L}_k)_{k>1}$ form a complete orthogonal basis of $L^2(\mathcal{T})$. Any realization of the stochastic process can then be represented as:

$$\boldsymbol{X}^*(t, \mathbf{u}) = \mu(t) + \sum_{k=1}^{\infty} \sqrt{\lambda_k}\xi_k(\mathbf{u})\mathcal{L}_k(t) \tag{3.10}$$

where $\mu(t)$ is the mean trajectory of $\boldsymbol{X}^*(t, \mathbf{U})$, and $\xi_k(\mathbf{u})$ corresponds to the uncertain part of the decomposition. The calculation of the eigenfunctions and eigenvalues involves a numerical solution of the second kind Fredholm equation and a generalized eigenvalue problem. For more details refer to [2] and [13]. Such a numerical estimation induces a first error in the KL expansion. A second source of error comes from the truncation to the N_k most significant eigenvalues, necessary for the

practical exploitation of this methodology. The KL expansion based on the estimated eigenvalues, eigenfunctions, and the finite truncation for one realization reads

$$\hat{\boldsymbol{X}}^*(t,\mathbf{u}) = \mu(t) + \sum_{k=1}^{N_k} \sqrt{\hat{\lambda}_k}\xi_k(\mathbf{u})\hat{\mathcal{L}}_k(t) \tag{3.11}$$

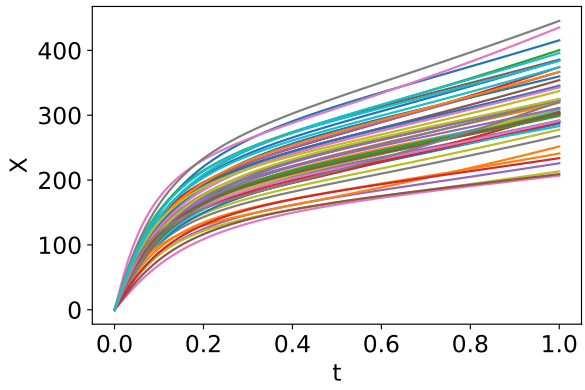


Figure 6: 50 realizations of a stochastic process

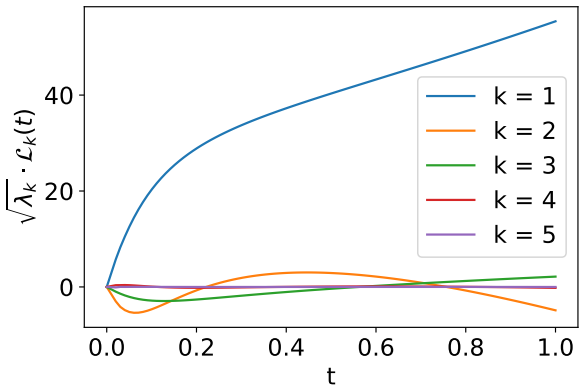


Figure 7: 5 most significant scaled modes of Karhunen-Loève decomposition

The error linked to estimation of the eigenvalues and eigenfunctions is not easy to quantify and reduce by itself. On the other hand, the choice of N_k represents a degree of freedom that controls the trade-off between accuracy and computational time. Higher values of N_k lead to a smaller truncation error, but for the desired purpose of this work, they also demand the creation of an equivalent number of surrogate models to predict for the uncertain part of the decomposition. Both components of error can be assessed in tandem by quantifying the error in the projection of a sample using Eq. (3.11), to identify the number of modes (N_k) that give an acceptable error value according to a chosen metric. It is important to remark that the only term of Eq. (3.11) that changes for two different $\boldsymbol{u}$ is $\xi_k(\mathbf{u})$. This fact will be exploited in the next sections as it means that the eigenfunctions and eigenvalues that are found for the training sample can be used in the prediction of a new stochastic process realization ($\hat{\boldsymbol{X}}^*(t,\boldsymbol{u})$). Such prediction is based on the capacity of approximating the unknown mapping $\mathbf{u} \sim \phi_{\mathbf{U}} \xrightarrow{\xi_k} \xi_k(\mathbf{u})$ with a surrogate model. In the proposed approach, such a surrogate model consists of univariate Gaussian processes. The stochastic process over which the Karhunen-Loève expansion is applied can be centered by calculating its mean realization and subtracting it from every realization. In that case, the term $\mu(t)$ of the KL expansion takes a zero value. When the autocovariance function associated to the optimal stochastic process $\boldsymbol{X}^*$ is not explicitly known, it can be accessed through the M realizations $(\boldsymbol{X}_1^*, \ldots, \boldsymbol{X}_M^*)$ that constitute the snapshots matrix $\tilde{\boldsymbol{X}}^* = (\boldsymbol{X}_1^* | \cdots | \boldsymbol{X}_M^*)$, where the symbol | indicates a collection of matrices. For a centered stochastic process the autocovariance based on the snapshots matrix reads $C = \frac{1}{M}\tilde{\boldsymbol{X}}^*\tilde{\boldsymbol{X}}^{*T}$. Is is worth mentioning that when non-linearities between the random input vector and the output stochastic process are present, the accuracy of the KL expansion is compromised. It is also worth highlighting that for the purpose of this work, the KL expansion reduces the dimension of the problem from the N correlated marginales of the time grid, to N_k uncertain variables.

In Fig. (6), 50 realizations of a stochastic process are depicted, the KL decomposition was performed to obtain its 5 most significant eigenvalues. A projected sample is expressed as a linear combination of the uncertain variables ξ_k and the scaled modes $\left(\sqrt{\hat{\lambda}_k} \cdot \mathcal{L}_k(t)\right)$ shown on Fig. (7).

3.2 Gaussian processes (GP)

After the Karhunen-Loève expansion is performed to obtain the uncertain KL variables, the eigenfunctions and the eigenvalues, a surrogate model can be trained to predict estimates of the uncertain variables ($\boldsymbol{u} \longrightarrow \hat{\xi}_k(\boldsymbol{u})$, for $k \in [1, ..., N_k]$) for new input values. Hence allowing to predict model responses $\hat{\boldsymbol{X}}^*(t, \boldsymbol{u})$. The advantage of using a surrogate model in tandem with the KL expansion is that the resulting function is cheaper to evaluate than the original model(which in our case requires to solve a full MDAO problem). Nonetheless, the obtained response has an intrinsic error due to the truncation to N_k modes of the KL expansion, its numerical approximations, and the GP error. If the error of the surrogate is known and lies under an acceptable threshold, the resulting surrogate strategy can be exploited for its intended purpose. But if the error on the predictions is large according to the predefined criteria, the surrogate model has to be retrained with additional data or with different parameters to try to improve its performance. When a large enough set of validation data of the original model is available, the error of the surrogate model can be computed by comparing its output predictions with the validation set, using for instance the generalization error metric as defined in [14]. On an opposite case, in the context of small-data where the original model is expensive to evaluate and not enough validation data is available, an error metric as Leave-one-out (LOO) can be used, it consists on training the surrogate model with all the samples but one, and then computing the error using the sample that was left out. The procedure is repeated M times by changing the sample that is left out and adding up the results.

The advantage of constructing the surrogate model using Gaussian processes is that they provide an error metric that quantifies how certain the prediction is, without the necessity of comparing the output with a large validation set nor training the surrogate model multiple times to compute error metrics. Gaussian processes, also called Kriging [15], are a Bayesian approach to machine learning where the estimated outputs are random normal variables with known parameters instead of just deterministic values. In this approach, the training data is used in the prediction step and has to be conserved along with the tuned parameters, for this reason, Gaussian process falls into the category of non-parametric machine learning algorithms.

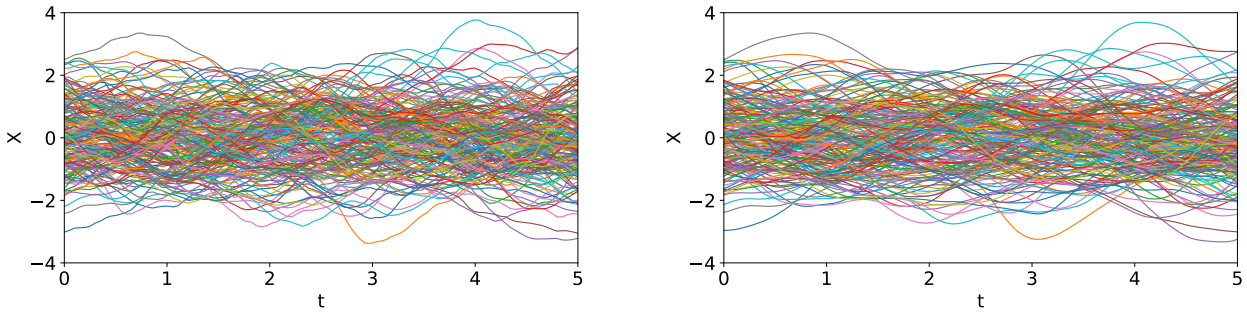
A Gaussian process is fully described by its mean or trend ($\mu(\cdot)$) and its covariance function or kernel ($k^{\boldsymbol{\theta}}(\cdot,\cdot)$) functions and is noted $\hat{f}(\cdot) \sim \mathcal{GP}\left(\mu(\cdot), k^{\boldsymbol{\theta}}(\cdot,\cdot)\right)$. Following the Bayesian approach, a Gaussian process requires a prior knowledge on $\mu(\cdot)$ and $k^{\boldsymbol{\theta}}(\cdot,\cdot)$, where θ is the set of parameters of the kernel. During the training phase, a maximization of the marginal likelihood is carried out to obtain a posterior distribution of θ, that is used later on to perform predictions. Based on the prior used to describe the mean, Gaussian process (or Kriging) are grouped into two categories: Ordinary Kriging and Universal Kriging [16]. In ordinary Kriging $\mu(\cdot)$ is described by a constant function, whereas in universal Kriging a specific form for the trend maybe consider (*e.g.,* linear, quadratic). Universal Kriging approaches may lead to more accurate surrogate models at the cost of an increased number of parameters that have to be tuned. In [14], an approach where the prior mean was modeled as a Polynomial Chaos expansion was presented. There exist multiple covariance models or kernels that help to describe how related is the change between two aleatory variables. Among the most used kernels are the Radial Basis Function (RBF) and the family of Matérn kernels, although many more exist. The choice of kernel has an important influence on the capacity of the surrogate model to fit the data and hence it is important to choose it right. Nevertheless, the active learning methodology that is the main objective of this work is independent of the kernel choice. The RBF kernel reads:

$$k\left(\boldsymbol{z},\boldsymbol{z}'\right)=\sigma^2\exp\left(-\frac{\|\boldsymbol{z}-\boldsymbol{z}'\|^2}{l}\right)=\sigma^2\exp\left(-\frac{d^2_{\boldsymbol{z},\boldsymbol{z}'}}{l}\right) \tag{3.12}$$

with $d_{\boldsymbol{z},\boldsymbol{z}'}=\|\boldsymbol{z}-\boldsymbol{z}'\|$ the Euclidean distance between $\boldsymbol{z}$ and $\boldsymbol{z}'$, σ the amplitude and l the length-scale coefficient. The set of parameters describing the kernel is $\theta=[\sigma,l]$. The family of Matérn kernels are described by:

$$k\left(\boldsymbol{z},\boldsymbol{z}'\right)=\sigma^2\frac{2^{1-\nu}}{\Gamma(\nu)}\left(\sqrt{2\nu}\frac{d_{\boldsymbol{z},\boldsymbol{z}'}}{l}\right)^{\nu}K_\nu\left(\sqrt{2\nu}\frac{d_{\boldsymbol{z},\boldsymbol{z}'}}{l}\right) \tag{3.13}$$

where ν characterizes the different kernels belonging to the family and typically takes values of $\frac{1}{2},\frac{3}{2}$ or $\frac{5}{2}$. By using the same lenght-scale and amplitude parameters it is possible to notice the differences between the kernels. To illustrate this concept, a Gaussian process prior is shown with length-scale and amplitude of 1 in Fig. (4). In Fig. (8), the same parameters are used to depict a Gaussian process prior by using Matérn kernels.



$\nu=\frac{3}{2}$ $\nu=\frac{5}{2}$

Figure 8: Gaussian process prior using Matérn Kernels for 200 trajectories

In the frame of this work, univariate Gaussian processes (*i.e.,* Gaussian processes with scalar output) are to be used to create a surrogate model of the uncertain part of the Karhunen-Loève decomposition ($\xi_k(\boldsymbol{u})$) represented by the application

$$\mathbf{u}\sim\phi_{\mathbf{U}}\xrightarrow{\xi_k}\xi_k(\mathbf{u})$$

The univariate Gaussian process metamodels are trained with the input set $\mathcal{U}_M$, and its responses $\mathcal{Y}_{k_M}=\xi_k(\mathcal{U}_M)\in\mathbb{R}^M$, $k\in[1,...,N_k]$ that correspond to the uncertain part of the Karhunen-Loève decomposition obtained after solving the generalized eigenvalue problem. After the training phase, that served to determine the posterior distribution of θ and the parameters of the trend $\mu(\cdot)$, the Gaussian process can issue predictions by conditioning the posterior distribution on the training data set. This consists in getting the conditional distribution of a joint Gaussian distribution

$$\hat{\xi}_k(\mathbf{u}|\mathcal{U}_M,\mathcal{Y}_{k_M})\sim\mathcal{N}\left(\hat{\bar{\xi}}_k(\mathbf{u}),\hat{\sigma}^2_k(\mathbf{u})\right) \tag{3.14}$$

whose mean $\hat{\bar{\xi}}_k(\mathbf{u})$ follows

$$\hat{\tilde{\xi}}_k(\mathbf{u}) = \mu(\boldsymbol{u}) + \boldsymbol{r}(\boldsymbol{u},\mathcal{U}_M)^T \boldsymbol{R}^{-1}(\mathcal{U}_M)(\mathcal{Y}_{k_M} - \boldsymbol{m}_M(\mathcal{U}_M)) \tag{3.15}$$

where

$$\boldsymbol{R}_{|ij}(\mathcal{U}_M) = k^\theta(\boldsymbol{u}_i, \boldsymbol{u}_j) \tag{3.16}$$
$$\boldsymbol{r}(\boldsymbol{u},\mathcal{U}_M) = [k^\theta(\boldsymbol{u}, \boldsymbol{u}_1), ..., k^\theta(\boldsymbol{u}, \boldsymbol{u}_M)]^T \tag{3.17}$$
$$\boldsymbol{m}_M(\mathcal{U}_M) = [m(\boldsymbol{u}_1), ..., m(\boldsymbol{u}_M)]^T \tag{3.18}$$

and whose variance $\hat{\sigma}_k^2(\mathbf{u})$ is described by

$$\hat{\sigma}_k^2(\mathbf{u}) = \sigma_\zeta^2(1 - \boldsymbol{r}(\boldsymbol{X},\mathcal{U}_M)^T \boldsymbol{R}^{-1}(\mathcal{U}_M)\boldsymbol{r}(\boldsymbol{u},\mathcal{U}_M)) \tag{3.19}$$

where σ_ζ^2 is the process variance. Gaussian process are an interpolating regressor, this means that the prediction they issue matches the exact value at the training data points. The variance of the Gaussian process prediction is zero at these points and increases as the points at which a prediction is issued go further from the training points. New points can be added in a specific area to improve the prediction accuracy and reduce the variance as illustrated in Fig. (9). On the left side of this figure the regressor is far away from the exact value around $X = 5$. On the left side $X = 5$ and its response are added to the training set to improve the accuracy in that region and decrease the variance associated to the prediction.

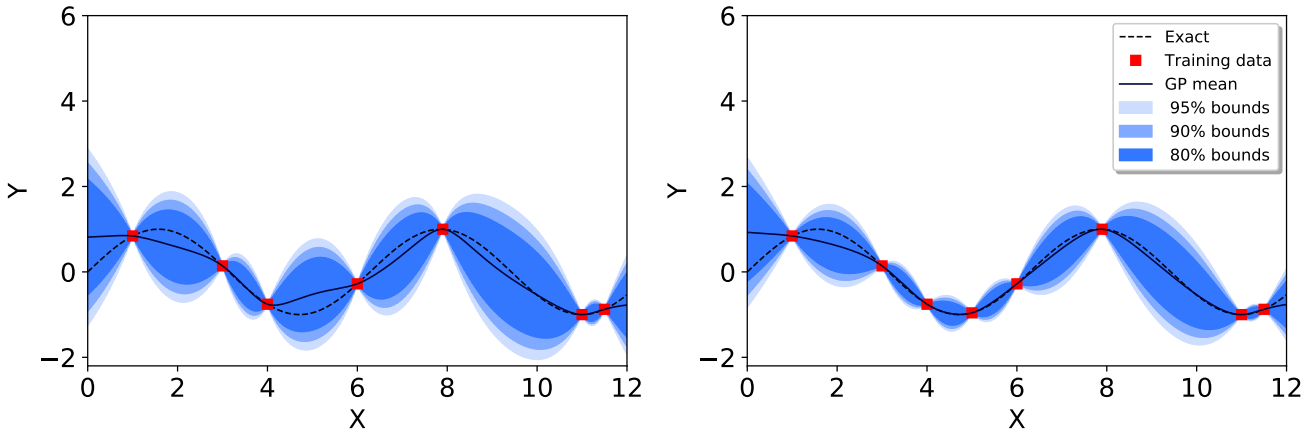


Figure 9: Illustration of Gaussian Process (GP) regression, comparison with exact function and variance reduction.

3.3 Active learning for quantile estimation

When uncertainties are taken into account in the MDAO of a launch vehicle, each realization of the uncertain input vector $\boldsymbol{U}$ leads to a different optimal trajectory. It is important to determine whether the resulting trajectories of the launch vehicle stay within the visibility zone of ground stations that relay vital information between the vehicle and the control center. For this purpose, it is useful to determine flight envelopes composed by rare probability quantiles. For instance, the 1 % and 99 % quantiles. This allows to determine for a given input vector the extreme cases of the trajectory at a given level of confidence.

Multiple realizations (in the order of hundreds or thousands) are necessary to calculate rare probability quantiles with precision and hence the interest of using cheap-to-evaluate surrogate models. In [3] and [4], Gaussian processes were used in the context of Reliability-Based Design Optimization (RBDO) to estimate quantiles, and the associated error model was utilized to improve the quality of the surrogate model in what is called an active learning strategy.

The construction of the surrogate model is based on an initial Design of Experiments (DoE) of limited size comprised of the realization of the input variable random vector $\mathcal{U}_M$ and its model responses $\boldsymbol{X}_M^*(t, \boldsymbol{U})$. The active learning technique looks to add new samples to $\mathcal{U}_M$ that improve the quality of the surrogate model. The advantage of the active learning technique with respect to an enrichment strategy in which more training points are added randomly (this corresponds simply to a larger initial DoE), consists of its capacity to locally refine the surrogate model. In other words, the active learning strategy improves the accuracy of the surrogate model prediction in the specific domain of the output space that contributes the most to the uncertainty reduction on the computation of interest (a given quantile in this work). For instance, if the surrogate model is used to estimate a 1% quantile ($q_{1\%}$), it is desired that the new samples contribute to this specific purpose without caring about the performance of the surrogate model to issue predictions near the 99% quantile ($q_{99\%}$). A simplified scheme of an active learning strategy to improve quantile estimation based on Gaussian process (GP) surrogate model is shown in Fig. (10).

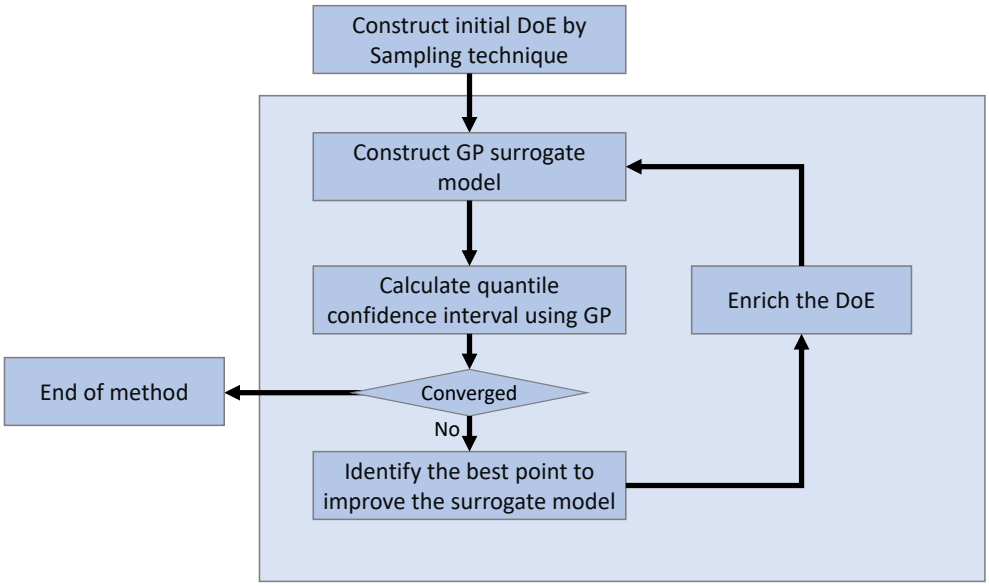


Figure 10: Active learning strategy to improve quantile estimation based on Gaussian process (GP) surrogate model

4 Proposed methods

The active learning technique for field variable quantile estimation proposed in this work can work on multi-dimensional fields (*e.g.,* a pressure distribution over an aerodynamic surface) or in uni-dimensional fields (*e.g.,* the evolution in time of a launcher state variable). Nevertheless, the methods to calculate the uncertainty on the quantile estimation change based on the dimension of the field. Following an approach of incremental complexity, the illustration cases that are

presented deal with the simpler uni-dimensional version and leave the multi-dimensional case as possible future development.

A simple test case was used to develop and test the active learning methodology. It consists of the ONERA Damage Model for Composites with Ceramic Matrix (ODM-CMC) [17]. This model allows to generate trajectories describing the stress-strain curves of composite materials under a tensile test at low computational cost. A brief description of the model and some associated results are shown in appendix B. The main application case of this work is related to the MDAO of launch vehicles. A brief description of it is given in section 4.1. The proposed active learning method is breakdown into 5 steps as shown in Fig. (11). Steps 1, 2 and 3 correspond to the construction of the surrogate model based on a limited size Monte Carlo simulation, the Karhunen-Loève expansion and Gaussian process and are described in section 4.2. The methodologies to compute the confidence interval area on the estimated quantile, corresponding to step 4, are described in section 4.3. Step 5 corresponds to the refinement criteria optimization that allows to identify the new inputs that contribute the most to the uncertainty reduction on the estimation of a given quantile and it is described in section 4.4.

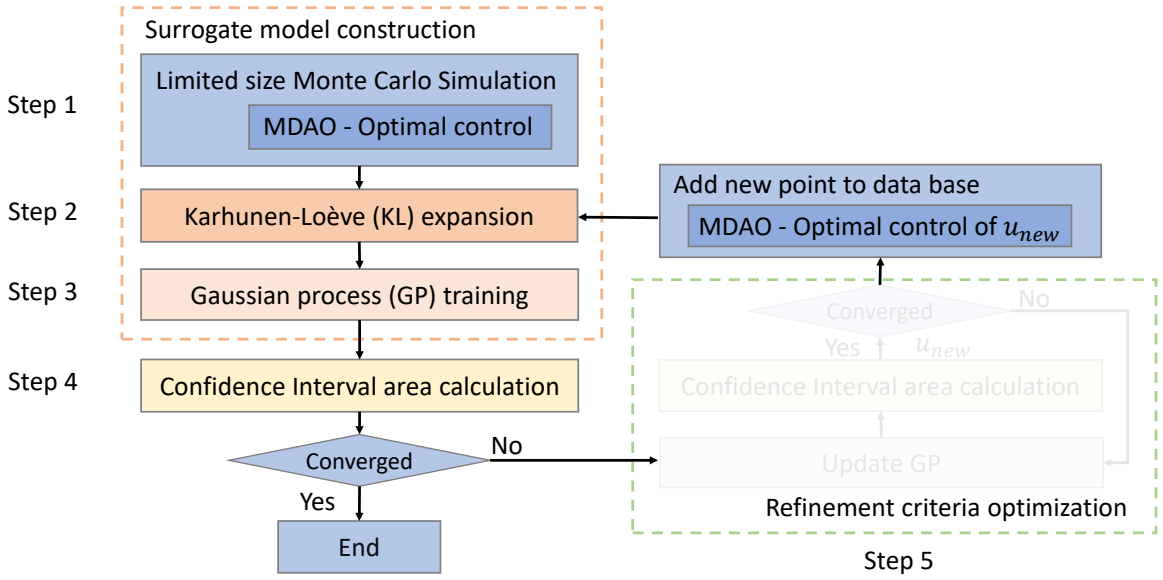


Figure 11: Breakdown of proposed active learning strategy for qunatile estimation

4.1 Launch vehicle MDAO

The launch vehicle code used for this project consists of a variant of an All-At-Once (AAO) formulation (Eqs. 3.1 to 3.7) that takes into account the propulsion, structures, aerodynamics and trajectory disciplines and that was developed during my master's research project [5]. The code accounts for dynamic pressure and heat flux constraints, performs an orbit injection using a Hohmann transfer type ascent, and considers the changes of mass due to stage and fairing jettisoning. 8 different phases are considered (*e.g.,* Lift-Off and gravity turn) during the ascent and a parameterized pitch angle guidance program conforms the vector of trajectory control variables ($\boldsymbol{w}$). To perform uncertainty propagation, a baseline mission of injecting 10 tons of payload into a 400 km altitude circular orbit was considered, using a Two-Stage-To-Orbit (TSTO) vehicle.

The architectural design variables ($\boldsymbol{z}$) corresponding to the propulsion discipline (*i.e.,* chamber pressure, mixture ratio of propellants, thrust at vacuum, nozzle exit pressure) were fixed with respect to the baseline solution and only the trajectory control variables ($\boldsymbol{w}$) and the mass of propellants were left under the control of the optimizer, with the objective of minimizing the GLOW. This is equivalent to fix a given design of a launch vehicle and only vary its length as a function of the uncertainties to accommodate the required amount of propellants so that it can accomplish its mission. For the purpose of creating the metamodel, the MDAO code is seen as a black box (Fig. 12) whose inputs are the realizations of the uncertain vector $\boldsymbol{U}$ contained in the training set $\mathcal{U}_M$ and whose output is the random process $\boldsymbol{X}^*(t, \boldsymbol{U})$ that is converted into the snapshots matrix $\tilde{\boldsymbol{X}}^*$ by the means of interpolation to ensure the all realizations are discretized according to the same time grid ($\mathcal{T}_N$). In this case the output $\boldsymbol{X}^*(t, \boldsymbol{U})$ represents the optimal trajectories issued by the MDAO code.

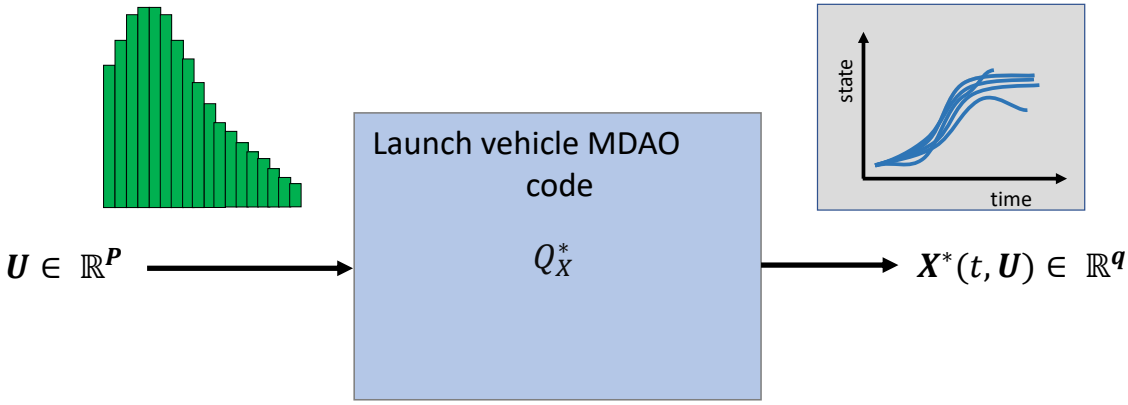


Figure 12: Launch vehicle MDAO code as a black box

The input random vector ($\boldsymbol{U}$) is composed of 7 independent random variables that are used to model uncertainties on drag coefficient, specific impulse, residual mass of propellants and mass flow rate of both stages. The probability density functions of the components of $\boldsymbol{U}$ are defined based on expert knowledge. A summary of the distributions is presented in table 2. A Monte-Carlo simulation with 1200 samples of ($\boldsymbol{U}$) was performed and the MDAO code was executed for each one of them in the High-Performance-Computing (HPC) cluster from ONERA. Each run of the code takes between 2 to 10 minutes to execute on a single processor (around 100 hours to execute the 1200 runs). Given that the MDAO code is on its early development stage, some corrupted results could be interpreted as converged solutions as the code has no easy-to-see flags to signal the problems, hence a deep understanding from the user is required to identify possible issues. Including non converged results into the training set would imply the training of a faulty surrogate model. In total 59 runs did not converge because they reached the maximum imposed number of iterations and 10 runs reached the lower optimization bound of the variable corresponding to the mass of propellants of the first stage. Hence 1131 successful runs were obtained and integrate the available sample set. The trajectories that were obtained correspond to the optimal values found by the optimizer for the state discretization nodes of the pseudo-spectral transcription. These trajectories describe the ascent phase of the launch vehicle up to the point of Elliptical Transfer Orbit (ETO) insertion (Fig. 13).

Given that all the trajectories were obtained using the same model with different input values, they all are discretized with the same number of state discretization nodes. Two issues arise when trying to compose the snapshots matrix ($\tilde{\boldsymbol{X}}^*$) out of this values. The first one is that for every optimal trajectory or realization of the output stochastic process ($\boldsymbol{X}^*(t, \boldsymbol{u}_k)$) the values of the state discretization nodes with same index take different time coordinates. The second one is

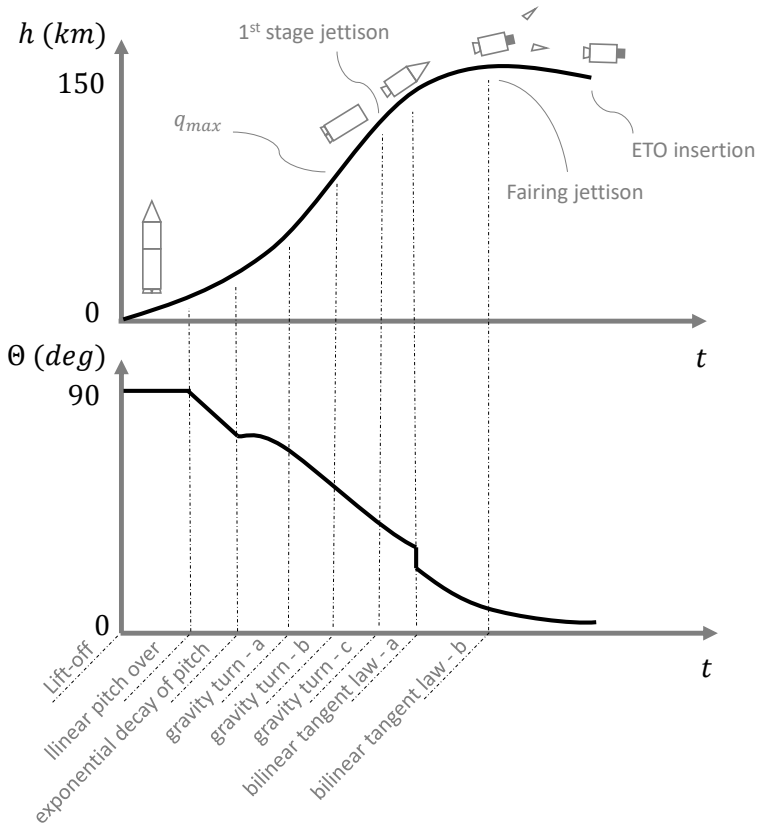


Figure 13: Phases of the launch vehicle ascent trajectory shown for the altitude (h) and the pitch angle (Θ) evolution in time

that all the trajectories have a different duration, this is, the instant of ETO insertion happens at a different time for all of them. To solve the first issue an interpolation procedure is used in order to obtain the values of the states using the same time grid ($\mathcal{T}_N$). The interpolation was performed using quadratic splines to smooth out the collocation defects of the pseudo-spectral nodes and redundant nodes corresponding to an uncompressed transcription were ignored. This is equivalent to assume that the continuity defects joining segments of the same phase take a zero value. This method works fine only for trajectories without discontinuities. The resulting stochastic process for the speed, altitude, dynamic pressure and heat flux for 1131 realizations can be seen on Figs. 14 to 17. When discontinuities are present in the trajectory, the quadratic interpolator outputs a noisy response. This can be noticed in the plot of the evolution of mass (Fig. 18) where the discontinuity after first stage separation causes a wobbly behavior and the discontinuity at the moment of fairing jettison causes a noisy response for some realizations. The MDAO code also outputs trajectories obtained as validation procedure that use the Runge-Kutta (RK) numerical integrator and the optimal values of the coupling variables ($\boldsymbol{y}$), architectural design variables ($\boldsymbol{z}$) and trajectory control variables ($\boldsymbol{w}$) to propagate dynamics from the initial boundary conditions up to 100s after the ETO insertion. In order to obtain trajectories that end at the same point in time, all the realizations were truncated at 430s and a linear interpolator was used to obtain the values of the states at the nodes of the time grid ($\mathcal{T}_N$). As the Runge-Kutta solution is smooth, the linear interpolator works just fine and overcomes the discontinuities of the mass evolution (Fig. 19). To be able to conform the snapshots matrix with the results interpolated from the pseudo-spectral nodes, the trajectories can be normalized in time without need to stop all the trajectories at 430s. The set of available trajectories is used to constitute the training set ($\boldsymbol{X}_M^*$) and the validation set ($\boldsymbol{X}_V^*$).

Table 2: Probability distribution of the components of the uncertain random vector

Name	Notation	Model (mean, standard deviation)
Specific impulse stage 1	U_{Isp_1}	$\mathcal{N}(0,1)$(additive, s)
Specific impulse stage 2	U_{Isp_2}	$\mathcal{N}(0,1)$(additive, s)
Residual mass stage 1	U_{m_1}	$\mathcal{U}(-750,750)$ (additive, kg)
Residual mass stage 2	U_{m_2}	$\mathcal{U}(-250,250)$ (additive, kg)
Mass flow rate stage 1	U_{q_1}	$\mathcal{N}(0,5)$(additive, kg/s)
Mass flow rate stage 2	U_{q_2}	$\mathcal{N}(0,5)$(additive, kg/s)
Drag coefficient	U_{CD}	$\mathcal{U}(-0.05,0.05)$(additive, $-$)

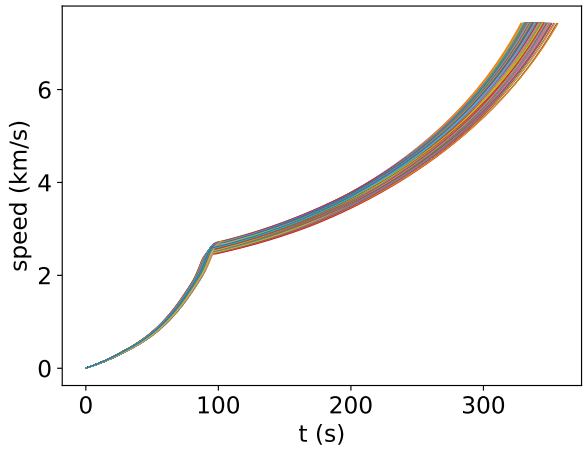


Figure 14: Uncertain speed profile

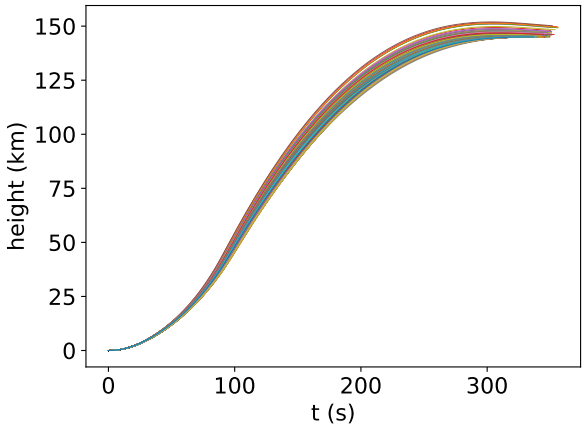


Figure 15: Uncertain altitude profile

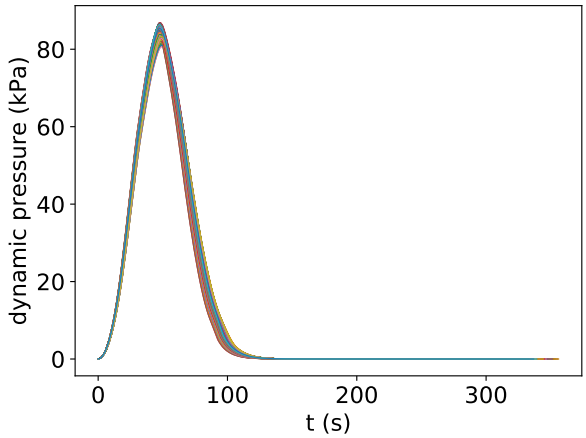


Figure 16: Uncertain dynamic pressure

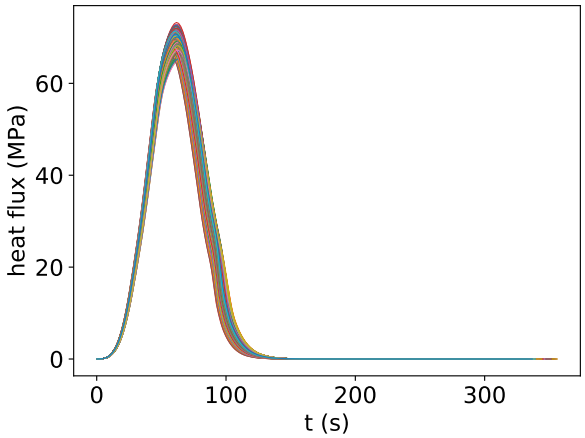


Figure 17: Uncertain heat flux

4.2 Surrogate model creation

The calculation of an extreme quantile of the output random process $\boldsymbol{X}^*(t,\boldsymbol{U})$ requires many evaluations (in the order of hundreds or thousands) of the application Q_x^* to capture the information of the low probability events. In the launch vehicle MDAO case presented in this work, the execution of Q_x^* for one thousand times takes around 4 days in a single processor. But in more complex MDAO cases including Finite Element Analyses (FEM) or Computation Fluid Dynamics (CFD) such a high number of evaluations can be prohibited given that one single evaluation of the original model can take one day or more. Hence the importance of creating a surrogate

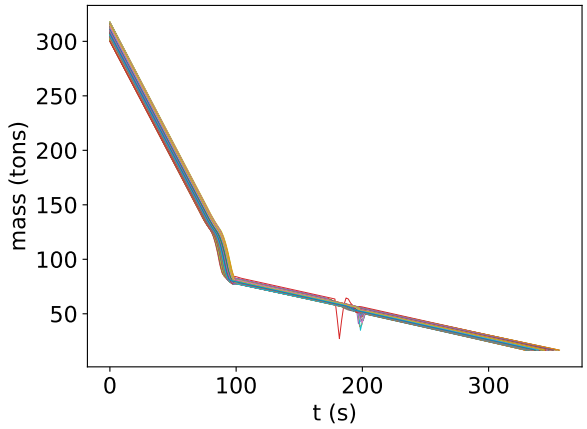


Figure 18: Uncertain mass profile

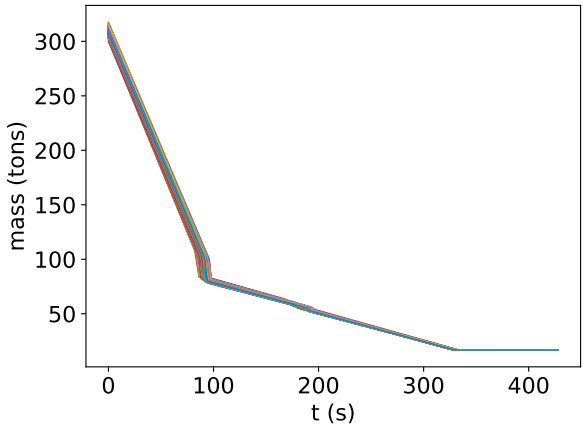


Figure 19: Uncertain mass profile - RK

model that can replicate the responses of Q_x^* at a lower computational cost. The surrogate model application is defined as:

$$\mathbf{U} \sim \phi_{\mathbf{U}} \xrightarrow{\hat{Q}_{\boldsymbol{X}^*}} \tilde{\boldsymbol{X}}^*(t, \mathbf{U}) \tag{4.1}$$

The creation of the surrogate model consists of a 3 step process (Fig. 20). On a first step, a reduced size Monte Carlo simulation is performed by obtaining M samples of the input probability density function $\phi_{\boldsymbol{U}}$ to constitue the training set $\mathcal{U}_M$, and the original model Q_x^* is executed for each realization of the sample. The snapshots matrix ($\tilde{\boldsymbol{X}}^*(t, \mathbf{U})$) is created by reading the values of the states at the common time grid ($\mathcal{T}_N$). The quality of the predictions of the surrogate model depends on the quality and the quantity of information available in $\tilde{\boldsymbol{X}}^*(t, \mathbf{U})$, as well as on the parameters used to create and train the model.

On a second step, the snapshots matrix $\tilde{\boldsymbol{X}}^*(t, \mathbf{U})$ is used to perform the Karhunen-Loève (KL) decomposition and reduce the dimensionality of the problem. Obtaining the eigenvalues ($\hat{\lambda}_k$) and the eigenfunctions ($\hat{\mathcal{L}}_k(t)$) as well as the uncertain variables of the decomposition ($\mathcal{Y}_{k_M} = \xi_k(\mathcal{U}_M)$).

The final step of the creation of the surrogate model consists in using univariate Gaussian processes (GP) for predicting the random variables associated to each KL mode. $\mathcal{U}_M$ and $\mathcal{Y}_{k_M}$ are used to train the N_k univariate Gaussian processes and predict the uncertain part of the KL decomposition for the point $\mathbf{u}$. Predictions can be issued using the conditional distribution on the desired input and the training sets, obtaining the posterior distributions ($\hat{\xi}_k(\mathbf{u}|\mathcal{U}_M, \mathcal{Y}_{k_M}) \sim \mathcal{N}\left(\hat{\bar{\xi}}_k(\mathbf{u}), \hat{\sigma}_k^2(\mathbf{u})\right)$) that are used to model uncertainty of the GP outputs. Hence, predictions on a new input $\boldsymbol{u}$ can be obtained using the full surrogate model that reads:

$$\hat{\boldsymbol{X}}^*(t, \mathbf{u}, \mathcal{U}_M, \mathcal{Y}_{k_M}) = \hat{\mu}(t) + \sum_{k=1}^{N_k} \sqrt{\hat{\lambda}_k} \hat{\xi}_k(\mathbf{u}|\mathcal{U}_M, \mathcal{Y}_{k_M}) \hat{\mathcal{L}}_k(t) \tag{4.2}$$

$\hat{\boldsymbol{X}}^*(t, \mathbf{u}, \mathcal{U}_M, \mathcal{Y}_{k_M})$ represents the predicted response in time at a new input point ($\mathbf{u}$) using a surrogate model based on Gaussian processes that were trained and issue predictions based on the training data $\mathcal{U}_M$ and $\mathcal{Y}_{k_M}$. It is very important to remember that the output variables of a Gaussian process are joint normal distributions instead of deterministic values. Hence, a prediction using Eq. (4.2) depends on the realizations of the random variables $\hat{\xi}_k(\mathbf{u}|\mathcal{U}_M, \mathcal{Y}_{k_M}) \sim$

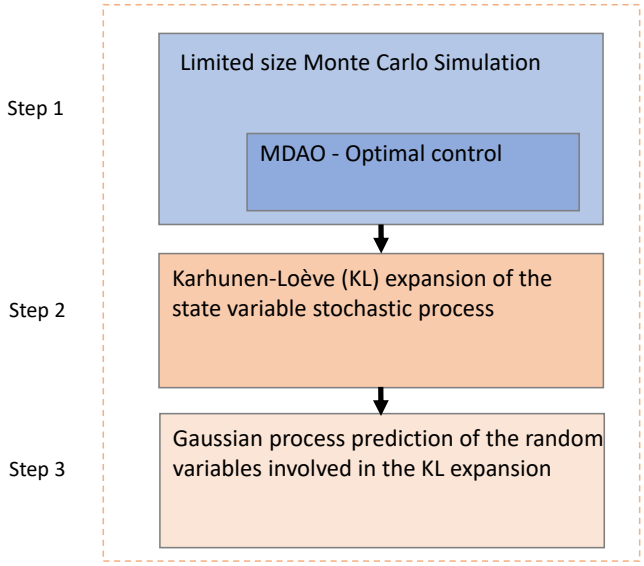


Figure 20: Proposed strategy for the creation of surrogate model using the Karhunen-Loève expansion and Gaussian processes

$\mathcal{N}\left(\hat{\bar{\xi}}_k(\mathbf{u}), \hat{\sigma}_k^2(\mathbf{u})\right)$. The most likely value for $\hat{\xi}_k(\mathbf{u}|\mathcal{U}_M, \mathcal{Y}_{k_M})$ is its mean $\hat{\bar{\xi}}_k(\mathbf{u})$ and its variance can be interpreted as an error model. This error model can be propagated through the Karhunen-Loève decomposition using the properties of the normal distribution. This means that every random process realization predicted with the strategy proposed in Fig. (20) is Gaussian variable with known mean and variance. Its variance can be interpreted as a measure of how certain the Gaussian process predictions are about the prediction.

4.3 Confidence interval area calculation of the estimated quantile

Step 4 corresponds to the criterion that drives the optimization loop of the active learning criteria. This criterion corresponds to the confidence interval area of the estimated quantile. This area comes from the error model of the Gaussian process. An analytic way of propagating the Gaussian process error through the Karhunen-Loève expansion is presented in section 4.3.1. The quantile calculation definition is presented in section 4.3.2. Finally, two different methods to calculate the confidence interval area of the estimated quantile are presented in sections 4.3.3 and 4.3.4.

4.3.1 Propagation of the error model of the Gaussian processes through the Karhunen-Loève decomposition

Let us consider a prediction of the surrogate model for a fixed time instant t_f and where the mean of the KL decomposition is null ($\mu(t) = 0$).

$$\hat{\boldsymbol{X}}^*(t = t_f, \mathbf{u}, \mathcal{U}_M, \mathcal{Y}_{k_M}) = \sum_{k=1}^{N_k} \sqrt{\hat{\lambda}_k}\hat{\xi}_k(\mathbf{u}|\mathcal{U}_M, \mathcal{Y}_{k_M})\hat{\mathcal{L}}_k(t = t_f) \tag{4.3}$$

where $\hat{\mathcal{L}}_k(t=t_f)$ and $\hat{\lambda}_k$ are scalars. Given that $\hat{\xi}_k\left(\mathbf{u}|\mathcal{U}_M,\mathcal{Y}_{k_M}\right) \sim \mathcal{N}\left(\hat{\bar{\xi}}_k(\mathbf{u}),\hat{\sigma}_k^2(\mathbf{u})\right)$ is normally distributed and following the affine property of the normal distribution, the following product

$$\sqrt{\hat{\lambda}_k}\hat{\xi}_k(\mathbf{u}|\mathcal{U}_M,\mathcal{Y}_{k_M})\hat{\mathcal{L}}_k(t=t_f) \tag{4.4}$$

is distributed according to

$$\mathcal{N}\left(\sqrt{\hat{\lambda}_k}\hat{\bar{\xi}}_k(\mathbf{u}|\mathcal{U}_M,\mathcal{Y}_{k_M})\hat{\mathcal{L}}_k(t=t_f),\hat{\lambda}_k\hat{\sigma}_k^2(\mathbf{u}|\mathcal{U}_M,\mathcal{Y}_{k_M})\hat{\mathcal{L}}_k(t=t_f)^2\right) \tag{4.5}$$

given the additive properties of the normal distributions, it can be inferred for the following sum

$$\sum_{k=1}^{N_k}\sqrt{\hat{\lambda}_k}\hat{\xi}_k(\mathbf{u}|\mathcal{U}_M,\mathcal{Y}_{k_M})\hat{\mathcal{L}}_k(t=t_f) \tag{4.6}$$

that its distribution reads

$$\mathcal{N}\left(\sum_{k=1}^{N_k}\sqrt{\hat{\lambda}_k}\hat{\bar{\xi}}_k(\mathbf{u}|\mathcal{U}_M,\mathcal{Y}_{k_M})\hat{\mathcal{L}}_k(t=t_f),\sum_{k=1}^{N_k}\hat{\lambda}_k\hat{\sigma}_k^2(\mathbf{u}|\mathcal{U}_M,\mathcal{Y}_{k_M})\hat{\mathcal{L}}_k(t=t_f)^2\right) \tag{4.7}$$

So far it has been demonstrated that the prediction of the metamodel for a fixed time instant has an uncertainty model that is normally distributed. For a prediction at a different time instant, only the value of the eigenfunctions of the KL decomposition changes. Hence, it can be generalized for any time instant

$$\begin{aligned}\hat{\boldsymbol{X}}^*(t,\mathbf{u},\mathcal{U}_M,\mathcal{Y}_{k_M}) &\sim \mathcal{N}\left(\sum_{k=1}^{N_k}\sqrt{\hat{\lambda}_k}\hat{\bar{\xi}}_k(\mathbf{u}|\mathcal{U}_M,\mathcal{Y}_{k_M})\hat{\mathcal{L}}_k(t),\sum_{k=1}^{N_k}\hat{\lambda}_k\hat{\sigma}_k^2(\mathbf{u}|\mathcal{U}_M,\mathcal{Y}_{k_M})\hat{\mathcal{L}}_k^2(t)\right) \\ &\sim \mathcal{N}\left(\hat{\bar{\boldsymbol{X}}}^*(\mathbf{u},t|\mathcal{U}_M,\mathcal{Y}_{k_M}),\Sigma(\mathbf{u},t|\mathcal{U}_M,\mathcal{Y}_{k_M})\right)\end{aligned} \tag{4.8}$$

where $\hat{\bar{\boldsymbol{X}}}^*(\mathbf{u},t|\mathcal{U}_M,\mathcal{Y}_{k_M})$ is the mean prediction and $\Sigma(\mathbf{u},t|\mathcal{U}_M,\mathcal{Y}_{k_M})$ is the variance of the full surrogate model. It is worth noticing that the mean prediction of the metamodel using the KL decomposition and the GPs is based on the KL decomposition using the mean prediction of the GPs. More importantly, a variance model for the metamodel is obtained. Analyzing the variance in a per mode way, it can be noticed that the variances of the GPs are scaled by the product of the eigenvalues and the square of the eigen functions. Meaning that the uncertainty of the GPs associated to the most significant modes have greater influence on the metamodel uncertainty. If at a given time instant a mode takes a zero value, the uncertainty of its GP does not affect the uncertainty of the metamodel.

During the exploitation of the surrogate model to issue a prediction, it is important to consider that its mean and variance are fields of the same dimension, in the case of the launch vehicle trajectories, they are 1-dimensional fields. To obtain the confidence interval of the prediction at a 3σ level it is enough to calculate:

$$\sum_{k=1}^{N_k}\sqrt{\hat{\lambda}_k}\bar{\xi}_k(\mathbf{u}|\mathcal{U}_M,\mathcal{Y}_{k_M})\hat{\mathcal{L}}_k(t) \pm 3\cdot\sqrt{\sum_{k=1}^{N_k}\hat{\lambda}_k\hat{\sigma}_k^2(\mathbf{u}|\mathcal{U}_M,\mathcal{Y}_{k_M})\hat{\mathcal{L}}_k^2(t)} \tag{4.9}$$

To illustrate the concept of this error model let us consider the following example for the centered trajectories for the altitude of the launch vehicle MDAO. The trajectories corresponding to the altitude state that are shown in Fig. (15) were centered and normalized in time. The resulting centered trajectories are shown in Fig. (21). A subset of M trajectories is randomly selected to compose the training set via the snapshots matrix and the full surrogate model as described in Fig. (20) is trained. The surrogate model ($\hat{\boldsymbol{X}}^*$) is used to issue a prediction on an input sample ($\boldsymbol{u}$) using Eq.(4.2). The resulting trajectory is itself a stochastic process depending on the realizations of the KL random variables $\hat{\xi}_k(\mathbf{u}|\mathcal{U}_M, \mathcal{Y}_{k_M}) \sim \mathcal{N}\left(\hat{\bar{\xi}}_k(\mathbf{u}), \hat{\sigma}_k^2(\mathbf{u})\right)$. One thousand realizations of $\hat{\xi}_k(\mathbf{u}|\mathcal{U}_M, \mathcal{Y}_{k_M})$ are generated to build trajectories using Eq.(4.2) and the results are shown in the right hand side of Fig. (22). In Fig. (23) the same procedure is performed for twenty thousand realizations. The error model presented in Eq. (4.8) is used to issue the mean prediction and the confidence intervals at 1, 2 and 3 standard deviations from the mean, shown with a black line and gray shadowed areas in the left hand side of Fig. (22), on the same figure, the red dotted lines correspond to the confidence intervals at 1, 2 and 3 standard deviations from the mean computed using the one thousand realizations. For the high probability events represented by the confidence intervals at 1 and 2 standard deviation from the mean the predictions using the error model of Eq. (4.8) and those obtained by brute force sampling coincide. But for the rare probability events represented by the confidence interval at 3 standard deviations from the mean the one thousand realizations obtained by brute force are not enough to capture the behavior in an accurate manner. Only by increasing the number of realizations, as with the twenty thousand represented on Fig. (23), the brute force generation matches the performance of the analytic method proposed in Eq. (4.8). The importance of this is that is illustrates the accuracy of the analytic model of Eq. (4.8) to represent the uncertainty of the surrogate model predictions. The uncertainty can be reduced by adding new training samples to the training set $\mathcal{U}_M$.

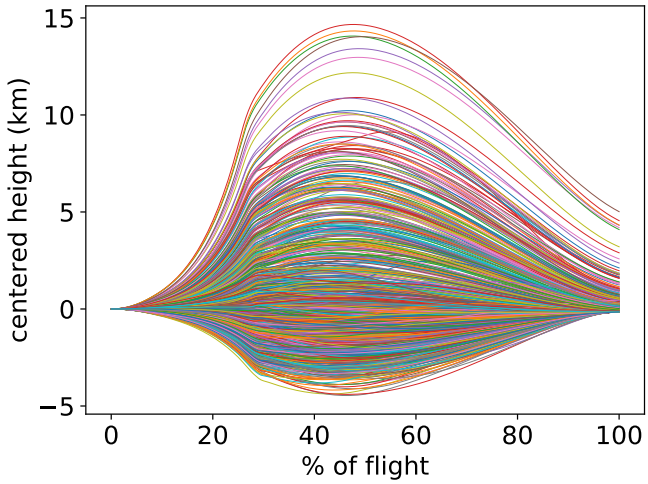


Figure 21: Centered an normalized output stochastic process for the altitude

4.3.2 Quantile estimation

In the case of the launch vehicle trajectories, the computation of a quantile is important as it allows to determine trajectory envelopes. This means that given some input uncertain variables contained in the vector $\boldsymbol{U}$ it is possible to determine for instance the 95% confidence interval for the output stochastic process $\hat{\boldsymbol{X}}(t, \boldsymbol{U})$. The importance of this notion is that it allows to calculate, for example, the evolution of the altitude of the trajectory at a certain level of confidence that can later be compared with the visibility regions of ground stations. The quantile estimation is performed by generating R samples using the input probability density function $\phi_{\boldsymbol{U}}$ to constitute the quantile

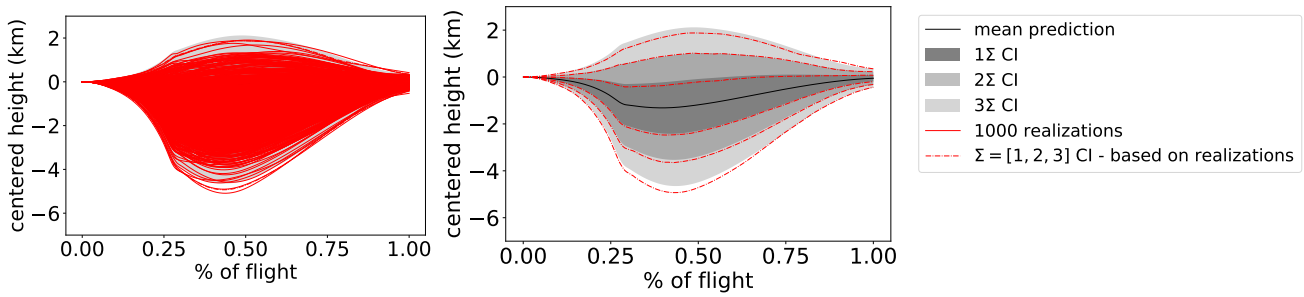


Figure 22: Calculation of confidence interval for 1 predicted sample using 1 000 GP trajectories. Training set of 50 samples.

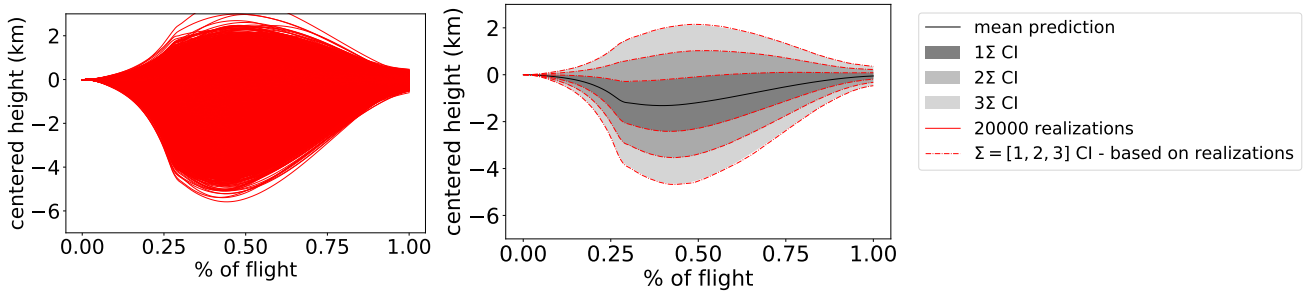


Figure 23: Calculation of confidence interval for 1 predicted sample using 20 000 GP trajectories

estimation input sample ($\mathcal{V}_R = [\boldsymbol{u_1}, ..., \boldsymbol{u_R}]$). The later is used to calculate the surrogate responses (approximated stochastic process trajectories) using the model $\hat{\boldsymbol{X}}^*(t, \boldsymbol{u})$. Each of the realizations of the resulting stochastic process is a random variable itself, whose probability density function is dictated by the error model of the Gaussian processes. This concept is illustrated in Fig. (24) for $R = 3$ and where the error model described in Eq. (4.8) is used to obtain the mean predictions and the mean plus 2 standard deviation predictions.

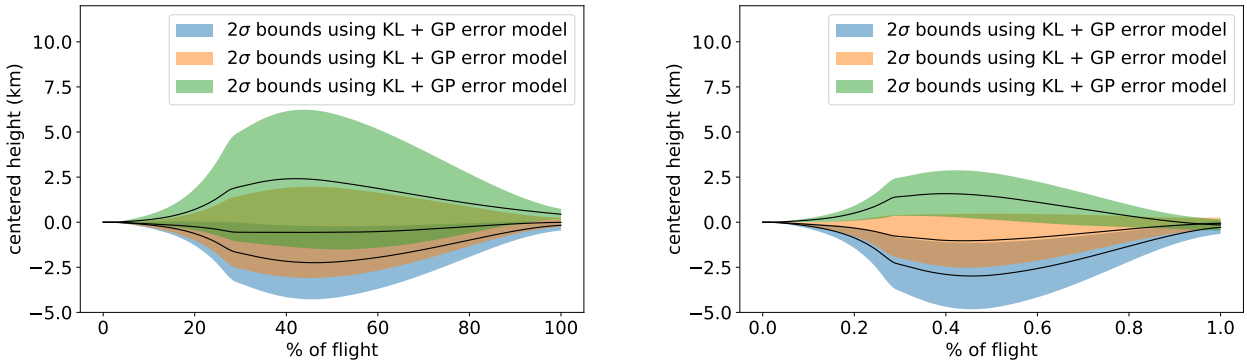


Figure 24: 2Σ confidence interval for 3 predicted samples using 50 training samples on the left and 100 training samples on the right. Mean predictions represented with black line.

The calculation of a quantile of the stochastic process is based on a realization of each of the random variables that compose it. Hence, the quantile is itself uncertain. The way in which those realizations are chosen is important as it leads to two different methods to assess the uncertainty of the quantile that will be explained in detail later and whose computational cost differs in two orders of magnitude.

To focus on the quantile computation, let us assume that a realization for each of the trajectories composing the stochastic process is available. At each point in time of the mesh $\mathcal{T}_N$ it is possible to reconstruct the empirical marginal distribution of $\hat{\boldsymbol{X}}_t^*$ and to compute the

$\eta-$quantile q_η for $\hat{\boldsymbol{X}}_t^*$ and $\eta \in [0,1]$. The definition q_η reads

$$q_\eta = \inf_{v\in\mathbb{R}} \left\{ \mathbb{P}\left[\hat{\boldsymbol{X}}_t^* \leq v\right] \geq \eta \right\} = \inf_{v\in\mathbb{R}} \left\{ \Psi_{\hat{\boldsymbol{X}}_t^*}(v) \geq \eta \right\} \tag{4.10}$$

where $\Psi_{\hat{\boldsymbol{X}}_t^*}(v)$ is the cumulative density function of $\hat{\boldsymbol{X}}_t^*$. Using this definition it is possible to compute the $\eta-$quantile of the stochastic process $\hat{\boldsymbol{X}}^*(t,\boldsymbol{U})$ to obtain for instance the trajectory envelopes considering a 95 % confidence interval. Such quantile is time dependent and can be written as $q_\eta(t)$.

4.3.3 Confidence interval area calculation based on Gaussian process trajectories (CI Area -A)

The first method to compute the uncertainty on the estimation of the quantile based on the Gaussian processes prediction consists of the generation of GP trajectories. As described in previous sections, univariate Gaussian processes are used to create a surrogate model $\mathbf{u} \sim \phi_{\mathbf{U}} \xrightarrow{\hat{\xi}_k} \hat{\xi}_k(\mathbf{u}|\mathcal{U}_M, \mathcal{Y}_{k_M})$ for each of the Karhunen-Loève modes. Following the Bayesian nature of Gaussian process, the resulting prediction $(\hat{\xi}_k(\mathbf{u}|\mathcal{U}_M, \mathcal{Y}_{k_M}))$ is a random Gaussian variable with known mean and variance. A Gaussian process trajectory consists in obtaining the i^{th} realization of this random variable. In Fig. (25), an illustration of this concept is depicted by considering only one dimension of the input random vector $(\boldsymbol{U})$. The GP trajectory ensures the spatial correlation of the predictions issued to compute a given quantile using the set $\mathcal{U}_R$ according to the chosen mean and covariance models and their trained parameters (θ). The surrogate model responses based on a unique GP trajectory for $\mathcal{U}_R$ are shown in Fig. (26) along with the computed quantile (q_η).

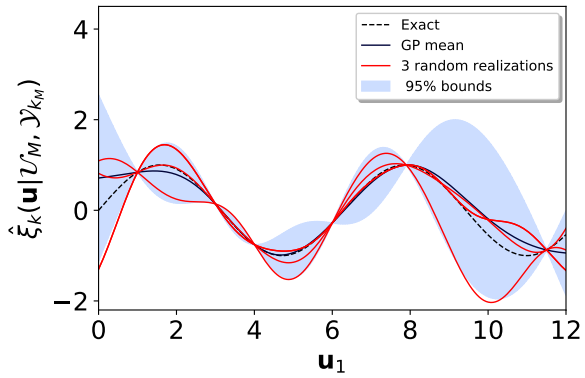


Figure 25: Illustration of realizations of the surrogate model $\hat{\xi}_k^{(i)}$ for $i = [1,2,3]$

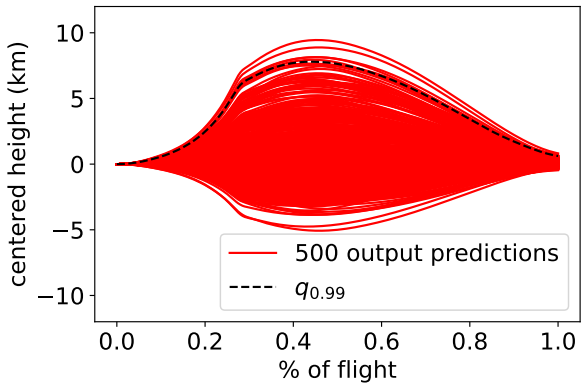


Figure 26: Responses $\hat{\boldsymbol{X}}^*$ based on 1 trajectory of $\hat{\xi}_k^{(i)}$

To model the uncertainty of the quantile estimation, this method generates $N_{GP_{traj}}$ trajectories, obtains the model responses $\hat{\boldsymbol{X}}^{*(i)}(t,\mathbf{u},\mathcal{U}_M,\mathcal{Y}_{k_M})$ using Eq. (4.2) $\forall \boldsymbol{u} \in \mathcal{U}_R$ and stores them in the set $\hat{\boldsymbol{X}}_R^*$. The responses contained in $\hat{\boldsymbol{X}}_R^*$ correspond to the i^{th} Gaussian process trajectory. Next, this method computes the β-quantile (q_β) of $\hat{\boldsymbol{X}}_R^*$ and stores it in $\mathcal{Q}_\beta$. After the procedure is repeated for all the generated GP trajectories, the set $\mathcal{Q}_\beta$ contains $N_{GP_{traj}}$ quantiles (see Fig. 27). The confidence interval at a level $\alpha \in [0,1]$ on the estimation of the quantile is obtained by computing the quantiles $\frac{1-\alpha}{2}$ and $\frac{1-\alpha}{2}+\alpha$ of the set $\mathcal{Q}_\beta$. A metric for the uncertainty on the estimation of the quantile is defined by calculating the area between $q_{\frac{1-\alpha}{2}}$ and $q_{\frac{1-\alpha}{2}+\alpha}$. Hence, the objective of the active learning technique is to reduce this area. The main interest of the CI Area - A method is that it respects the spatial correlation given a posterior distribution of

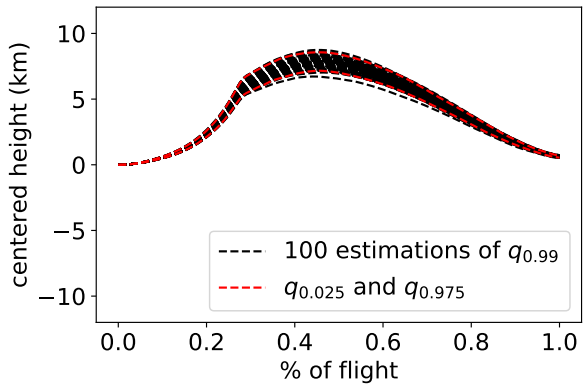


Figure 27: Quantiles computed based on GP trajectories of $\hat{\xi}_k^{(i)}$ for $i = [1, ..., 100]$

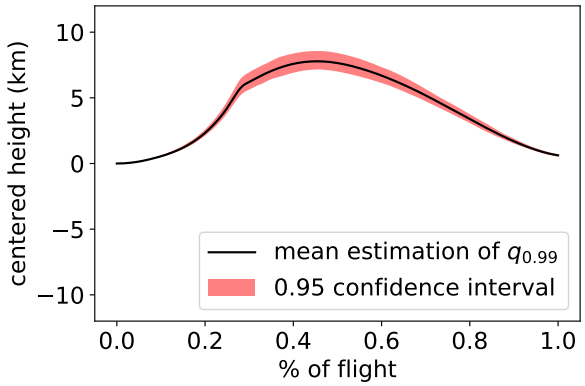


Figure 28: Confidence interval on the quantile estimation and mean prediction of the quantile

the covariance kernel by issuing all the predictions contained in $\hat{\boldsymbol{X}}_R^*$ based on a Gaussian process trajectory. It also leads to a known level for the confidence interval on the estimation of the desired quantile. This method requires the evaluation of the full metamodel $N_{GP_{traj}} \times R$ times, hence it is computationally expensive.

Algorithm 1: Confidence interval area calculation based on GP trajectories - CI Area A

Inputs : $t, \mathcal{V}_R, \mathcal{U}_M, \mathcal{Y}_{k_M}$
Initialization: $\alpha, \beta, N_k, N_{GP_{traj}}$
for $i = 1 : N_{GP_{traj}}$ **do**
  generate new $\hat{\xi}_k^{(i)}(\mathbf{u}|\mathcal{U}_M, \mathcal{Y}_{k_M})$ for $k = 1 : N_k$
  for $\mathbf{u}$ *in* $\mathcal{V}_R$ **do**
    calculate $\hat{\boldsymbol{X}}^{*(i)}(t, \mathbf{u}, \mathcal{U}_M, \mathcal{Y}_{k_M})$ using Eq. (4.2) and store it in $\hat{\boldsymbol{X}}_R^*$
  end
  calculate $q_\beta(t)$ of $\hat{\boldsymbol{X}}_R^*$ and store it in $\mathcal{Q}_\beta$
end
calculate $q_{\left(\frac{1-\alpha}{2}\right)}(t)$ and $q_{\left(\frac{1-\alpha}{2}+\alpha\right)}(t)$ of $\mathcal{Q}_\beta$
calculate $A = \int \left(q_{\left(\frac{1-\alpha}{2}+\alpha\right)}(t) - q_{\left(\frac{1-\alpha}{2}\right)}(t) \right) dt$
return A

4.3.4 Confidence interval area calculation based on the propagation of the Gaussian process error through the Karhunen–Loève expansion (CI Area - B)

The second approach to assess the uncertainty of the quantile estimation is based on the propagation of the error model of the Gaussian processes through the Karhunen–Loève decomposition (see Sec. 4.3.1). This method takes advantage of the fact that each trajectory issued by the surrogate model $\hat{Q}^*$ is a normal distributed variable with known mean and variance. Eq. (4.8) allows to compute extreme realizations of the predicted outputs, using a level γ it can be written $\hat{\boldsymbol{X}}^*(\mathbf{u}, t|\mathcal{U}_M, \mathcal{Y}_{k_M}) + \gamma \cdot \Sigma(\mathbf{u}, t|\mathcal{U}_M, \mathcal{Y}_{k_M})$. The 95.45% confidence interval can be calculated by estimating two predictions, one with $\gamma = +2$ and another with $\gamma = -2$. The confidence interval area calculation consists in obtaining the uppermost realization at a level γ for each of the outputs, storing the results in a set $\hat{\boldsymbol{X}}_R^{*+}$. An illustration of this is given on Fig. (24), where the trajectories at a level $\gamma = 2$ are represented by the upper bound of the shaded areas. The procedure

continues by the calculation of the ψ - quantile (q_ψ) of $\hat{\boldsymbol{X}}_R^{*+}$. The ψ - quantile of the CI Area - B method is equivalent to the β - quantile of the CI Area - A method. The procedure is repeated for $-\gamma$ to obtain the quantile of $\hat{\boldsymbol{X}}_R^{*-}$ (see Fig. 29). The area between both quantiles is finally calculated using a numerical integrator and used as a measure of uncertainty (see Fig. 30). This method ignores the spatial correlation of the Gaussian process trajectories that rely on the choice of the mean and covariance kernel and the level of the confidence interval depends on the size of the sample $\mathcal{U}_R$. Nevertheless, the computational cost is reduced by a factor of $\frac{N_{GP_{traj}}}{2}$ compared to the CI Area - A method. The level of the confidence interval could be calculated by using recursive methods to find the K^{th} order-statistic that contains a desired quantile, as the problem consists on computing quantiles of a set of normal variables [18]. However, this would not solve the spatial correlation of the kriging trajectories. If the CI Area - A method with $\alpha = 0.95$ and the CI Area - B method with $\gamma = 2$ (equivalent to 0.954 confidence interval) are used within the active learning technique to estimate the same quantile (*i.e.*, $\psi = \beta$), the confidence interval area of the CI Area - B method over estimates its confidence level w.r.t. to the CI Area - A. This is linked to the fact that each of the trajectories involved in the confidence interval estimation in the CI Area - B method was assumed to takes its mean plus/minus two standard deviation values, which is an unlikely scenario in the CI Area - A due to the spatial correlation ensured by the GP trajectories. As a consequence, the area of the CI Area - B has a larger magnitude than that of the CI Area - A method.

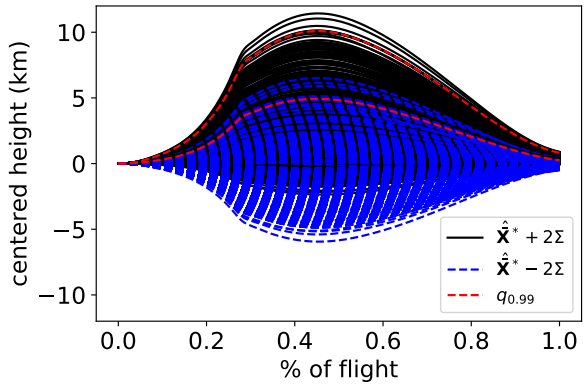


Figure 29: $\hat{\boldsymbol{X}}_R^{*+}$ and $\hat{\boldsymbol{X}}_R^{*-}$ sets with their respective quantiles

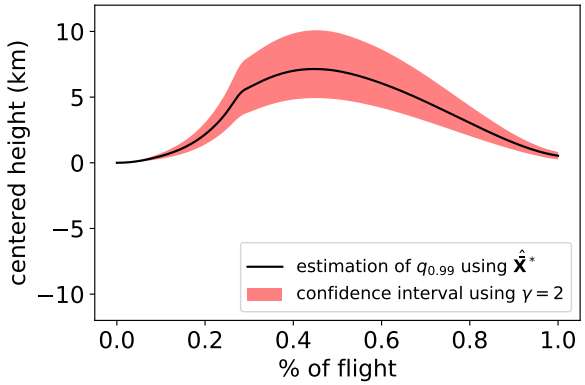


Figure 30: confidence interval on the quantile estimation and prediction using $\hat{\boldsymbol{X}}^*$

Algorithm 2: Confidence interval area calculation based on GP trajectories - CI Area B

Inputs : $t, \mathcal{V}_R, \mathcal{U}_M, \mathcal{Y}_{k_M}$
Initialization: γ, ψ, N_k
for u *in* $\mathcal{V}_R$ **do**
  calculate $\hat{\bar{\boldsymbol{X}}}^*(\mathbf{u}, t|\mathcal{U}_M, \mathcal{Y}_{k_M}) + \gamma \cdot \Sigma(\mathbf{u}, t|\mathcal{U}_M, \mathcal{Y}_{k_M})$ (Eq. 4.8) and store it in $\hat{\boldsymbol{X}}_R^{*+}$
  calculate $\hat{\bar{\boldsymbol{X}}}^*(\mathbf{u}, t|\mathcal{U}_M, \mathcal{Y}_{k_M}) - \gamma \cdot \Sigma(\mathbf{u}, t|\mathcal{U}_M, \mathcal{Y}_{k_M})$ (Eq. 4.8) and store it in $\hat{\boldsymbol{X}}_R^{*-}$
end
calculate $q_\psi(t)$ of $\hat{\boldsymbol{X}}_R^{*-}$ and $\hat{\boldsymbol{X}}_R^{*+}$
calculate $A = \int \left(q_\psi[\hat{\boldsymbol{X}}_R^{*+}](t) - q_\psi[\hat{\boldsymbol{X}}_R^{*-}](t) \right) dt$
return A

4.4 Refinement criteria optimization

Until this point, the procedure to construct a surrogate model of the orginal application based on the Karhunen-Loève expansion and Gaussian processes has been described and the uncertainty quantification case of computing a quantile of the surrogate model responses has been defined. Two different methods for the computation of the uncertainty on the estimation of the quantile due to the use of the Gaussian processes surrogate models have been also proposed. Step 5 of the proposed methodology corresponds to the refinement criteria optimization. The objective of the active learning technique is to improve the accuracy of the surrogate model for the calculation of a given quantile by improving the quality of the Gaussian process models (both in terms of prediction and error model). To accomplish this mission, a strategy is envisioned to find a new input point $\boldsymbol{u}_{new}$, which together with its original model response $\boldsymbol{X}^*_{new}$, minimizes the uncertainty of the quantile estimation when added to the training sets $\mathcal{U}_M$ and $\boldsymbol{X}^*_M$. This allows to add meaningful data to the training set, therefore improving the estimation of the quantile while using computational resources in an efficient manner. This is because the new samples are specially selected based on their contribution to the uncertainty reduction. The quantile refinement criteria optimization problem can be defined as

$$\begin{aligned}
&\text{minimize } A_{virtual}\left(t, \mathcal{V}_R, [\mathcal{U}_M, \boldsymbol{u}_{new}], [\mathcal{Y}_{k_M}, \mathcal{Y}_k(\boldsymbol{u}_{new})]\right) && (4.11)\\
&\text{with respect to } \boldsymbol{u}_{new} && (4.12)\\
&\text{subject to:} && (4.13)\\
&\text{upper/lower bounds } \underline{\boldsymbol{u}_{new}} \leq \boldsymbol{u}_{new} \leq \overline{\boldsymbol{u}_{new}} && (4.14)\\
& && (4.15)
\end{aligned}$$

where the cost function to be minimized is $A_{virtual}$, that should be distinguished from a standard area confidence interval area evaluation in that the training sets are augmented. The input training set $[\mathcal{U}_M, \boldsymbol{u}_{new}]$ corresponds to the union of the original training set $\mathcal{U}_M$ and the new sample proposed by the optimizer $\boldsymbol{u}_{new}$. The output training set $[\mathcal{Y}_{k_M}, \mathcal{Y}_k(\boldsymbol{u}_{new})]$ is comprised of the original training output set $\mathcal{Y}_{k_M}$ and the GP response for $\boldsymbol{u}_{new}$, that is obtained using the current GP prediction $\hat{\xi}_k(\mathbf{u}_{new}|\mathcal{U}_M, \mathcal{Y}_{k_M})$ to calculate $\mathcal{Y}_k(\boldsymbol{u}_{new})$. The upper and lower bounds $(\underline{\boldsymbol{u}_{new}}, \overline{\boldsymbol{u}_{new}})$ are defined differently for each component of $\boldsymbol{u}_{new}$ as a function of the probability distribution family. For the launch vehicle design case, the probability distributions of the components of $\boldsymbol{u}_{new}$ were defined in table 2. For the uniform distribution, the lower and upper bounds of the optimization correspond the minimum and maximum values of the distribution. For the normal distributions, the lower and upper bounds for the optimization were defined with the values of the mean plus/minus four standard deviations of the distributions.

The active learning strategy uses an optimizer to identify $\boldsymbol{u}_{new}$. For this work, the evolutionary optimizer CMA-ES [19] is used, although the methodology is compatible with other types of optimization methods and algorithms. CMA-ES stands for Covariance Matrix Adaptation Evolution Strategy, it is a stochastic global search algorithm that uses multivariate normal distributions to generate individuals and evaluate their fitness, at each iteration, these individuals are recombined to try to find the one with the best fitness. This optimizer was used in [3]. The recommended size of the popluation for CMA-ES is a function of the dimension of $\boldsymbol{U}$

$$N_{popsize} = 4 + \lfloor 3 \cdot \log P \rfloor \tag{4.16}$$

In a first step, the active learning strategy creates the surrogate model as presented in sec-

tion 4.2 and evaluates the confidence interval area for a given quantile using one of the methods presented in sections 4.3.3 and 4.3.4. During the optimization stage, the optimizer looks for a new point $\boldsymbol{u}_{new}$ that reduces the uncertainty on the quantile estimation linked to the Gaussian process error. This uncertainity is represented by $A_{virtual}$. The error induced by the truncation to N_k modes of the Karhunen-Loève expansion cannot by evaluated during the optimization stage and should rather be analyzed in a previous step before running the active learning algorithm. Each time that the optimizer evaluates an auxiliary candidate $\boldsymbol{u}_{aux}$, the N_k Gaussian process surrogate models $(\hat{\xi}_k(\boldsymbol{u}|\mathcal{U}_M), \mathcal{Y}_{k_M})$ used to predict the uncertain variables of the Karhunen-Loève decomposition are updated with this point. For this purpose, the mean evaluation of $\hat{\xi}_k(\boldsymbol{u}_{aux}|\mathcal{U}_M, \mathcal{Y}_{k_M})$ for $k \in [1, ..., N_k]$ is used to predict the response $\mathcal{Y}_k(\boldsymbol{u}_{aux})$ and the set of observations employed to issue the conditional predictions are augmented in such a way that the new predictions of the surrogate model follow $\hat{\xi}_k(\boldsymbol{u}|[\mathcal{U}_M, \boldsymbol{u}_{aux}], [\mathcal{Y}_{k_M}, \mathcal{Y}_k(\boldsymbol{u}_{aux})])$ for $k \in [1, ..., N_k]$. It is important to highlight that the Gaussian processes are not retrained during this stage, and the parameters of the kernel (θ) and the trend $(\mu(\cdot))$ are kept invariant. Then, the surrogate model using the augmented sets is used to calculate the confidence interval area $(A_{virtual})$ for a given quantile using one of the methods presented in sections 4.3.3 and 4.3.4.

The convergence criteria of the optimizer can follow different metrics, like a tolerance on the change of the area or the input $(\boldsymbol{u}_{aux})$ or a given maximum number of iterations $(N_{maxiter_{opt}})$. Once the optimizer converges, giving as output $(\boldsymbol{u}_{new})$, the high computational cost exact function (in our case the launch vehicle MDAO problem) is called to solve for $\boldsymbol{u}_{new}$. This new point and its response are added to the training sets, with the likely outcome that the uncertainty on the estimation of the quantile coming from the Gaussian processes will be reduced. Although this is not always guaranteed. The procedure of adding new inputs $(\boldsymbol{u}_{new})$ is repeated multiple times until a convergence criteria is reached. This limit can be defined with a maximum number of iterations based on the affordable computational cost $(N_{maxiter})$ or when the area of the confidence interval of the quantile estimation (A) falls under a desired threshold $(A_{desired})$. A scheme of the described methodology is shown in Fig. (31).

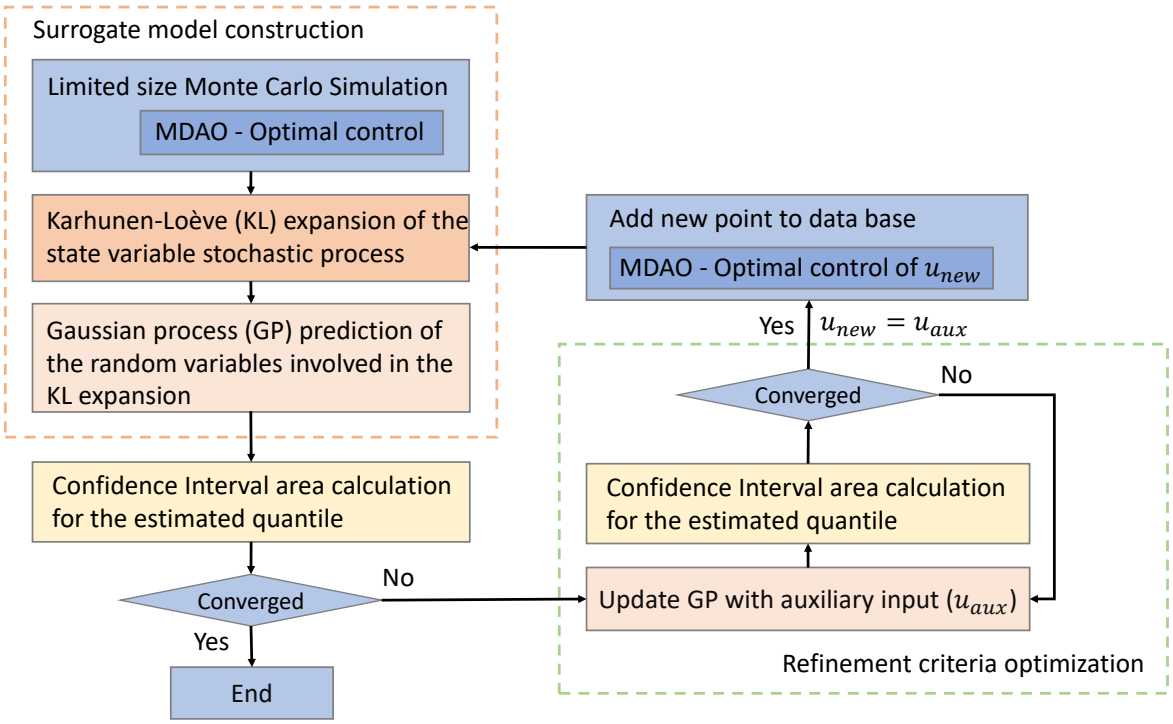


Figure 31: Proposed active learning strategy for quantile estimation

Algorithm 3: Active learning for quantile confidence interval refinement

Inputs : $t, \mathcal{V}_R, \mathcal{U}_M, \boldsymbol{X}^*_M$
Initialization: $\alpha, \beta, \gamma, \psi, N_k, N_{GP_{traj}}, A_{desired}, N_{maxiter}, i = 0$
- Perform KL decomposition of $\boldsymbol{X}^*_M$ truncating at N_K modes to obtain $\mathcal{Y}_{k_M}$ for $k \in [1, ..., N_k]$
- Train N_k univariate GP ($\hat{\xi}_k(\mathbf{u}|\mathcal{U}_M, \mathcal{Y}_{k_M})$) surrogate models based on $\mathcal{U}_M$ and $\mathcal{Y}_{k_M}$
- Build full surrogate model $\hat{\boldsymbol{X}}^*(t, \mathbf{u}, \mathcal{U}_M, \mathcal{Y}_{k_M})$
- Obtain $A(t, \mathcal{V}_R, \mathcal{U}_M, \mathcal{Y}_{k_M})$ using CI Area A or CI Area B methods
while $A > A_{desired}$ *and* $i \leq N_{maxiter}$ **do**
  while *optimizer criteria is not reached* **do**
    - optimizer provides $\mathbf{u}_{aux}$
    - evaluate $\hat{\xi}_k(\mathbf{u}_{aux}|\mathcal{U}_M, \mathcal{Y}_{k_M})$ to obtain $\mathcal{Y}_k(\boldsymbol{u}_{aux})$
    - Obtain $A_{virtual}$ $(t, \mathcal{V}_R, [\mathcal{U}_M, \boldsymbol{u}_{aux}], [\mathcal{Y}_{k_M}, \mathcal{Y}_k(\boldsymbol{u}_{aux})])$ using CI Area A or CI Area B methods
  end
  return $\mathbf{u_{new}} = \mathbf{u}_{aux}$ *that minimizes* $A_{virtual}$
  - evaluate the original model using $\mathbf{u_{new}}$ to obtain $\boldsymbol{X}^*(\mathbf{u_{new}})$
  - add $\mathbf{u_{new}}$ to $\mathcal{U}_M$ and $\boldsymbol{X}^*(\mathbf{u_{new}})$ to $\boldsymbol{X}^*_M$
  - Perform KL decomposition of $\boldsymbol{X}^*_M$ truncating at N_K modes to obtain $\mathcal{Y}_{k_M}$ for $k \in [1, ..., N_k]$
  - Train N_k univariate GP ($\hat{\xi}_k(\mathbf{u}|\mathcal{U}_M, \mathcal{Y}_{k_M})$) surrogate models based on $\mathcal{U}_M$ and $\mathcal{Y}_{k_M}$
  - Build full surrogate model $\hat{\boldsymbol{X}}^*(t, \mathbf{u}, \mathcal{U}_M, \mathcal{Y}_{k_M})$
  - Obtain A $(t, \mathcal{V}_R, \mathcal{U}_M, \mathcal{Y}_{k_M})$ using CI Area A or CI Area B methods
  - i +=1
end

5 Results and discussion

On a first step, this section focuses on analyzing the results for the active learning technique applied to the altitude profile trajectories of the launch vehicle MDAO case. A comparison between the two different methodologies for the assessment of the quantile prediction uncertainty is made. An aleatory enrichment strategy is also tested and compared against both methodologies. More importantly, the link between the reduction of the quantile uncertainty and the quantile estimation error is assesed by comparing the results with the quantile of the validation set. Secondly, the results of the active learning technique for the quantile estimation of the speed, and heat flux profiles are also presented.

5.1 Active learning for the altitude profile trajectory of a launch vehicle MDAO

The uncertain trajectory for the TSTO vehicle with mission of injecting 10 tons of payload into a circular orbit of 400km altitude was used in this case. The uncertain input vector $\mathbf{U}$ was described in table 2 and 1131 optimal trajectories were obtained by evaluating the original model Q^*_X. They were centered and shown in Fig. (21). 200 hundred realizations of $\mathbf{U}$ and their responses were randomly chosen to constitute the input training set $\mathcal{U}_{M=200}$ and the output training set $\boldsymbol{X}^*_{M=200}$. The quantile estimation input set was composed with 500 realizations ($\mathcal{V}_{R=500}$), belonging to the sample set of 1131 realizations described in section 4.1. The input validation set ($\mathcal{V}_{V=500}$) was

composed with the same sample, and the validation output sample ($\boldsymbol{X}^*_{V=500}$) is comprised of the responses of $\mathcal{V}_V$. The code was run to add 10 training samples to the initial training sets ($N_{maxiter} = 10$) without specifying a target area ($A_{desired}$). The CMA-ES optimizer was utilized with a population of 9 individuals and a maximum number of iterations of $N_{maxiter_{opt}} = 50$. A study to find the number of Karhunen-Loève modes that are used to truncate the decomposition is first shown, followed by the results of the active learning technique.

5.1.1 KL modes study

Some information is lost when the Karhunen-Loève expansion is truncated to (N_k) modes. The larger N_k, the better the projected samples represent the original samples, but in the case of the surrogate model presented in this work, this comes at the price of having to train a larger number of univariate Gaussian processes (GP) as each KL mode requires one GP. If the number of modes is small, the information that is lost in the projection of the samples increases. A study using different N_k values was made to assess the trade-off between the accuracy of the projected samples and the number of modes (see Fig. 32). The information that is lost during the projection is calculated in the form of a predictivity factor that approaches to 1 when the projected samples approach to the original samples. This represents a meaningful metric for the projected training samples if the training set is large enough, or if a validation set can be projected. Although validation data is not always available.

The training samples $\boldsymbol{X}^*_M$ were projected by using their uncertain variables $\xi_k(\boldsymbol{u})$ for $k \in [1, ..., N_k]$ and the scaled modes $\sqrt{\hat{\lambda}_k} \cdot \hat{\mathcal{L}}_k(t)$ to obtain the projected set $\boldsymbol{X}^*_{M_{proj}}$. The predictivity factor reads:

$$Q_2 = 1 - \frac{\sum_{i=1}^{M} \left(\boldsymbol{X}^*_M(t, \boldsymbol{u_i}) - \boldsymbol{X}^*_{M_{proj}}(t, \boldsymbol{u_i}) \right)^2}{\mathbb{V}[\boldsymbol{X}^*_M(t, \boldsymbol{u_i})]} \tag{5.1}$$

For $N_k = 2$ the predictivity factor ($Q_2 = -42.9737$) reflects the bad quality of the prediction that can also be evaluated by the amplitude of the residuals on the top left of Fig. (32). The residuals decrease and Q_2 augments as the number of modes are increased, and for $N_k = 30$ the predictivity factor reaches its threshold of 1 to a precision of 4 decimal places. The choice of the KL modes was based on the predictivity factor for the projected training samples as the validation is used to asses the accuracy of the complete strategy as a whole and it was assumed that no information about it was known to make the choice. Nevertheless, the predictivity factor and residuals for the projection of the validation set are displayed in Fig. (33). It can be noticed that Q_2 behaves in a similar way to the case of the projection of the training set. $N_k = 30$ was chosen as truncation parameter for the active learning technique.

With the purpose of assessing the effect of the KL expansion error on the estimation of a given quantile, the $q_{0.99}$ was calculated for a different validation set $\boldsymbol{X}^*_{V=900}$, that is to be be used as reference. This quantile was compared, using the Root Mean Squared Error (RMSE), with the quantile of the projection of $\boldsymbol{X}^*_{V=900}$ into the base of eigenvalues and eigenfunctions obtained from the KL decomposition of the training set $\boldsymbol{X}^*_{M=200}$. The RMSE using $N_k = 30$ was found to be 0.2m, which is almost negligible for this application.

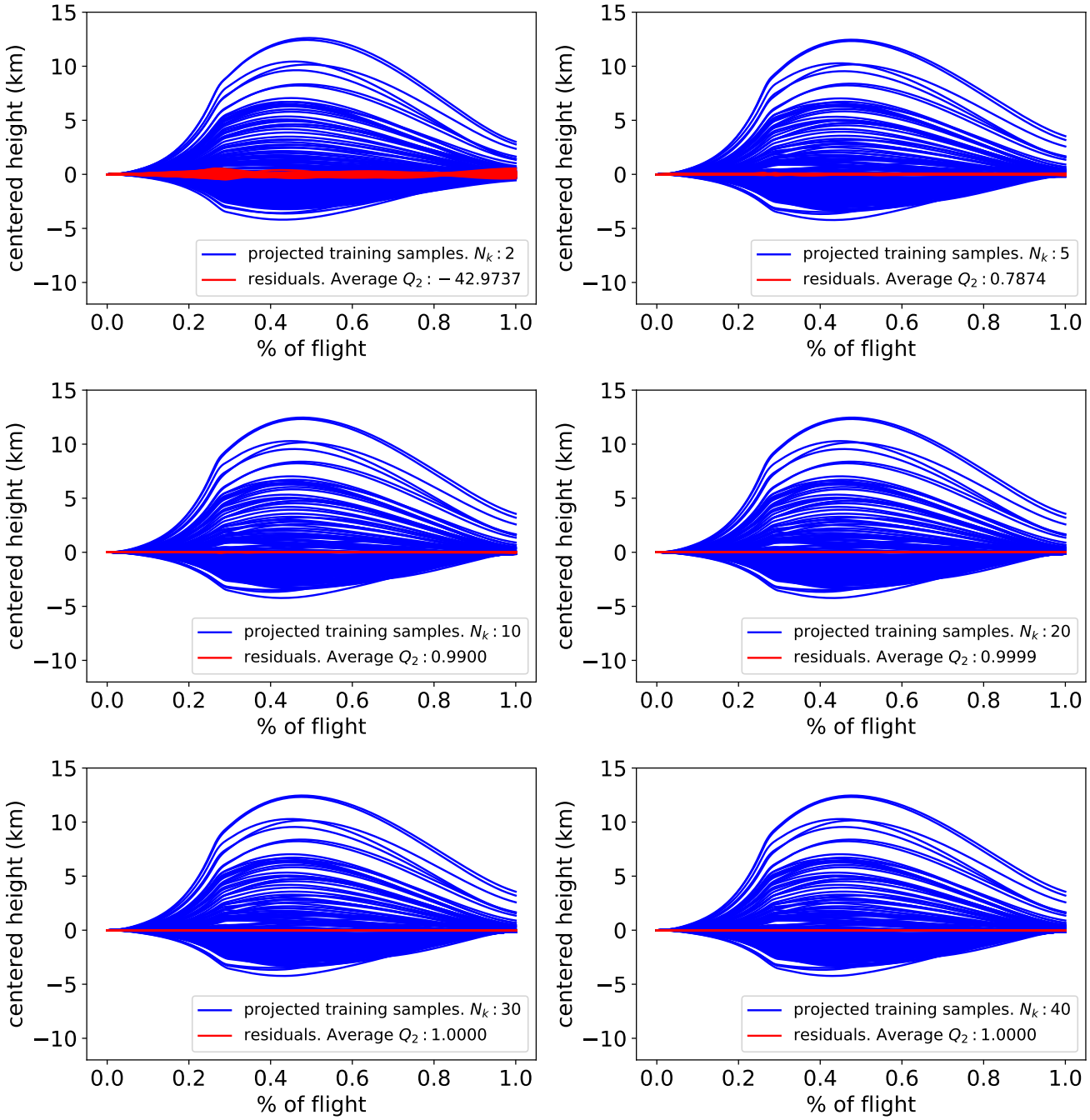


Figure 32: Evolution of residuals and predictivity factor (Q_2) as a function of the number of KL modes (N_k) that are used to truncate the expansion

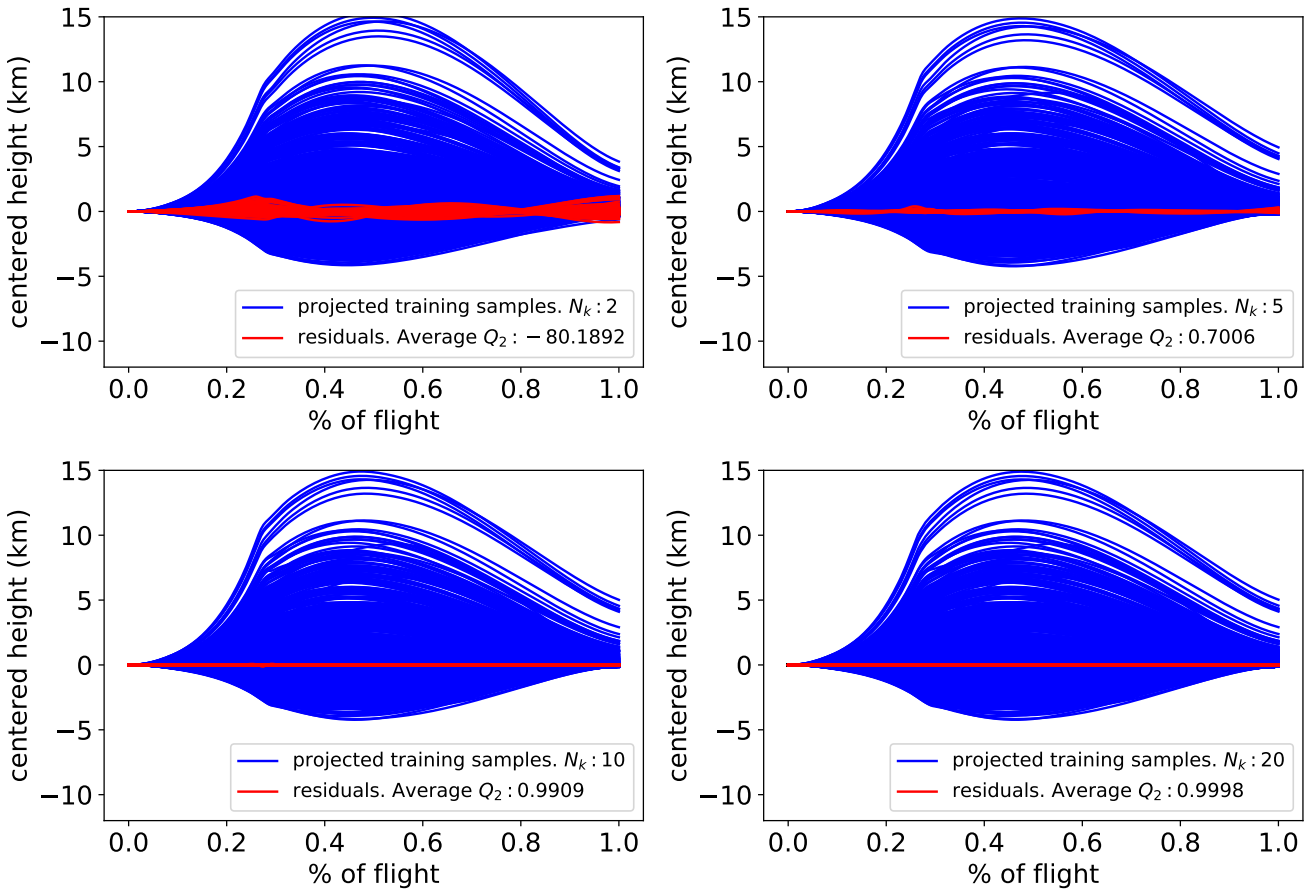


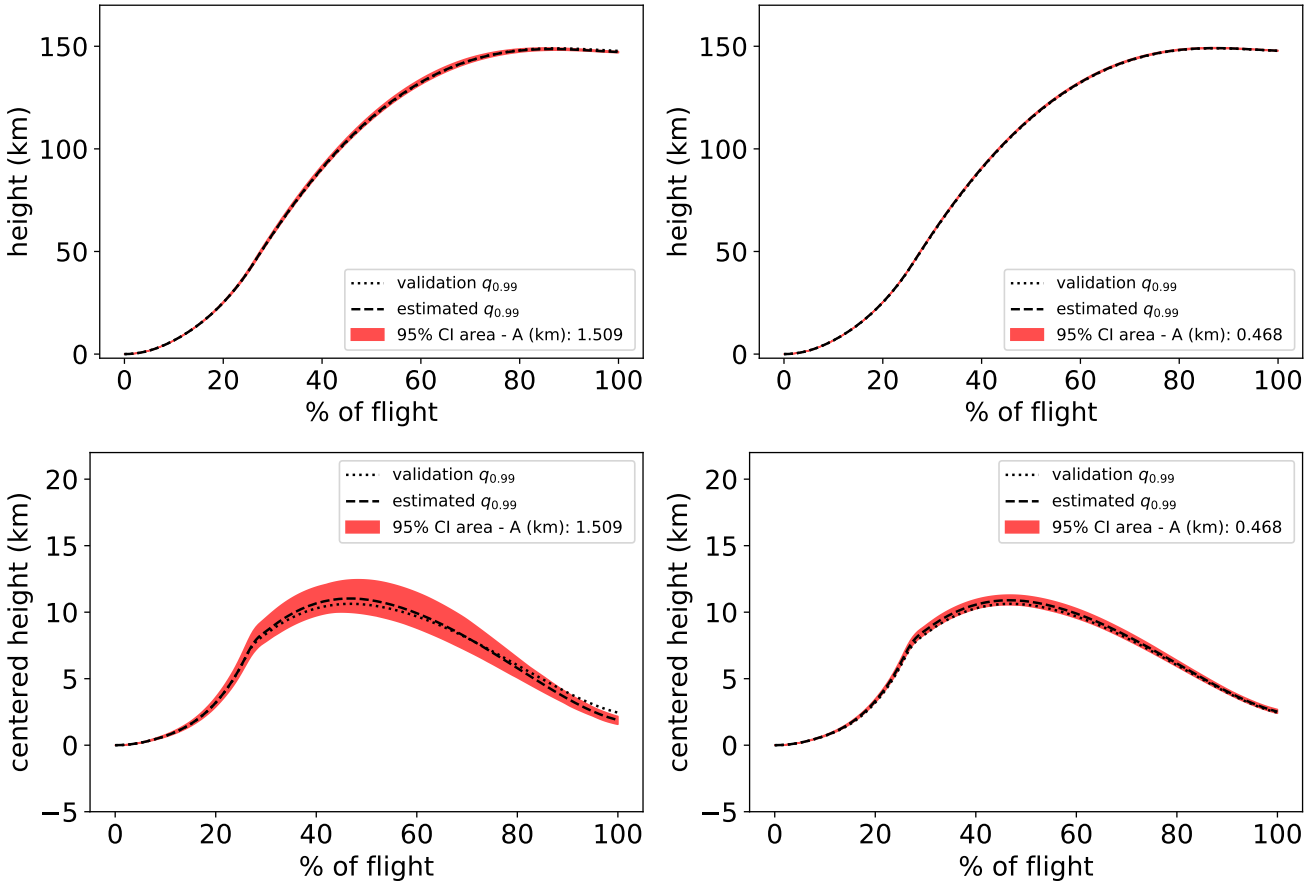
Figure 33: Evolution of residuals and predictivity factor (Q_2) as a function of the number of KL modes (N_k) for the valdiation samples

5.1.2 $q_{0.99}$ using active learning enrichment with CI Area - A method

The active learning technique was executed with the purpose of reducing the uncertainty on the quantification of the 0.99 - quantile using the CI Area - A method with $\mathcal{V}_{R=500}$ and $N_{GP_{traj}} = 100$. $R = 500$ was chosen because it makes likely that 5 trajectories appear above the 0.99 - quantile. In a similar way, the choice of $N_{GP_{traj}} = 100$ leads to the likely event that 5 trajectories are outside the 95% confidence interval of the estimation of the quantile. A higher number of realizations for $N_{GP_{traj}}$ and $\mathcal{V}_R$ is desired but the computational cost increases linearly as a function of each parameter. The 95% confidence interval area of the estimation of the quantile based on the original training set ($\mathcal{U}_{M=200}$) was found to be 1.509 km and it is displayed to the right of Fig. (34). It is important to highlight that the confidence interval area of the estimation of the quantile takes the value of the state or quantity of interest when the time is normalized. In this case the units are in km. On the left side of the same figure, the 95% confidence interval for the quantile estimation obtained with the enriched set with 10 extra samples added with the active learning technique is displayed. The area was reduced more than 3 times, taking a value of 0.468 km.

An important feature on Fig. (34) can be noticed when analyzing the centered trajectories shown in the lower part. The uncertainty on the quantile estimation is bigger between the 30% and the 80% of the flight duration where the centered altitude profile takes its widest amplitude. Although the CI Area - A method is not based on Eq. (4.8), the definition of the variance ($\sum_{k=1}^{N_k} \hat{\lambda}_k \hat{\sigma}_k^2(\mathbf{u}|\mathcal{U}_M, \mathcal{Y}_{k_M}) \hat{\mathcal{L}}_k^2(t)$) obtained from the propagation of the Gaussian process error through the Karhunen-Loève expansion shows that the uncertainty for predicting one trajectory grows as a function of the square of the scale modes ($\hat{\lambda}_k \cdot \hat{\mathcal{L}}_k^2(t)$), as a consequence, the

uncertainty of the set of trajectories is greater where the magnitude of the given state or quantity of interest is greater. The 10 trajectories that were added with the active learning procedure can be noticed in Fig. (35). They all are near the $q_{0.99}$ that is to be estimated, meaning that the active learning technique refines locally the surrogate model, it adds the new samples near the quantile of interest. Thanks to this, the new samples contribute mostly to the precise estimates of trajectories in the region of interest. The number of calls to the surrogate model (Eq. 4.2) that was utilized to identify and add the 10 samples is the product of the maximum number of iterations ($N_{maxiter} = 10$), the optimizer iterations ($N_{maxiter_{opt}} = 50$), the number of individuals ($N_{popsize} = 9$), the size of the quantile estimation sample ($R = 500$) and the number of GP trajectories ($N_{GP_{traj}} = 100$). In this case, 225 million surrogate model calls were necessary.



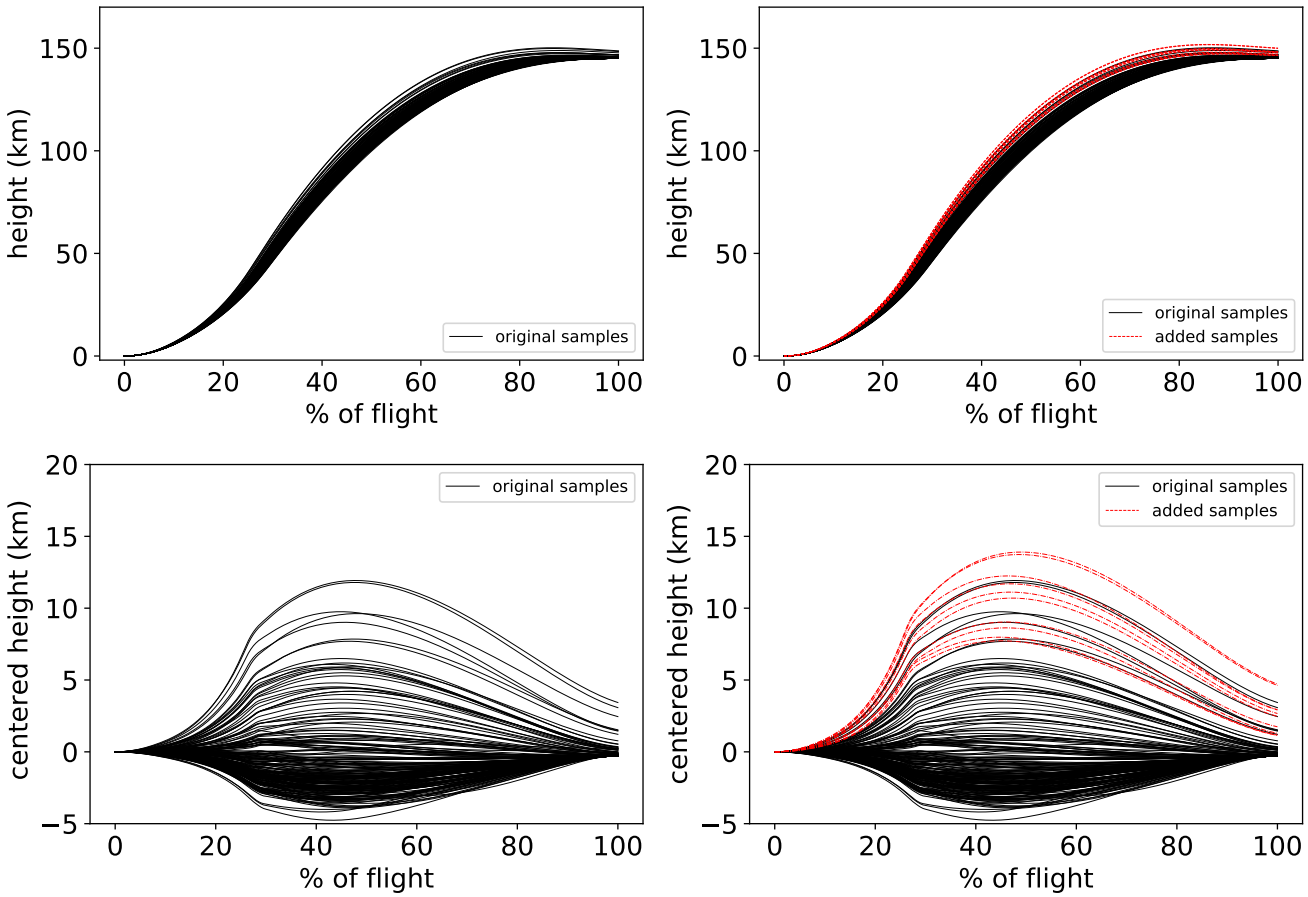
Estimation using the original set

Estimation using the enriched set

Figure 34: Quantile estimation variation after 10 active-learning-based enriched samples using the CI area - A method

5.1.3 $q_{0.99}$ using active learning enrichment with CI Area - B method

The interpretation of the resulting confidence interval of the quantile estimation using the CI Area - B method differs from that of the CI Area -A method on the fact that no confidence level is easily attributed to it. Eq. (4.8) was used to issue predictions at the $\pm 2 \cdot \Sigma$ level for each trajectory and the 0.99 -quantiles of the two resulting sets were calculated. The resulting confidence interval area is called 95% CI Area - B as it is linked to the $\pm 2 \cdot \Sigma$ prediction that corresponds to the 95.45% confidence interval. Although, in reality, the area of the quantile confidence interval is higher than 95% and approaches 100% as the size of the quantile estimation sample $\mathcal{U}_R$ increases. The confidence interval area using the initial training set was found to be 2.972 km, as expected,



Original samples | Original and added samples

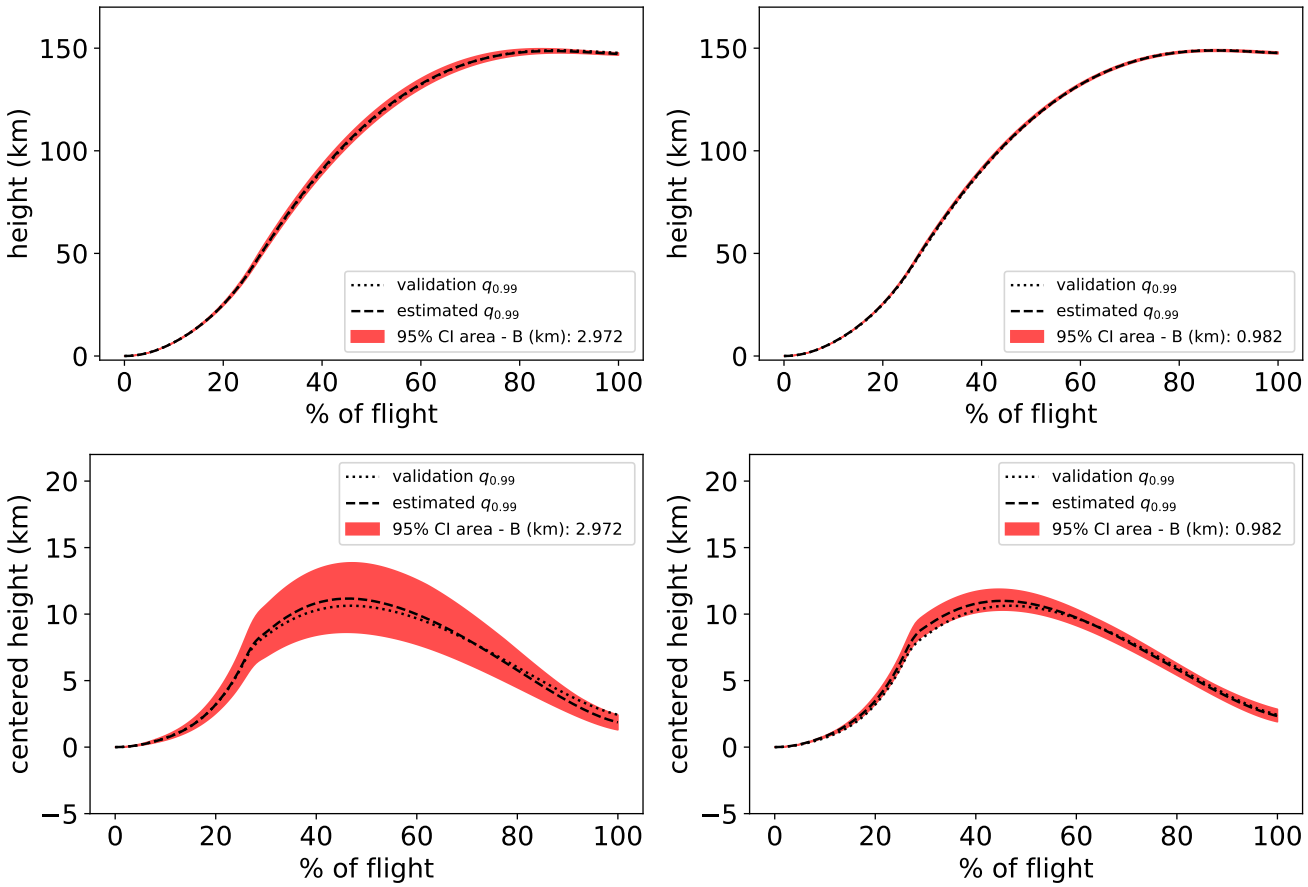
Figure 35: altitude profile trajectories used for surrogate training using CI area - A method

is higher than the initial area that was found with the CI Area -A method. After the active learning technique was used to add 10 samples to the original training set, the confidence interval area was reduced more than 3 fold, down to 0.982 km. The results are displayed in Fig. (36). In a similar fashion to the CI Area -A method, the 10 added samples are in the region that is close the quantile of interest ($q_{0.99}$) that in this case is calculated with the mean realization of Eq. (4.8). The added samples can be seen in Fig. (37). This method requires 50 times less calls to the surrogate model, reaching 4.5 million evaluations as the GP trajectories necessary for the CI Area - A method ($N_{GP_{traj}} = 100$) are replaced by just 2 evaluations ($\pm 2 \cdot \Sigma$).

A comparison of the new samples that were added using the CI Area - A and the CI Area - B methods is displayed in the input space in Fig. (38). The enriched samples added with both methods fall over the same regions. Some clear clusters can be observed in the figure, leading to the conclusion that the samples that improve the estimation $q_{0.99}$ tend to higher values of Isp_2, lower values of m_1 and m_2 and low values of C_D. The values of Isp_1 tend to have positive values. The remaining components of the input uncertain vector (q_1 and q_2) show no special trend and are more evenly spread over their domain.

5.1.4 $q_{0.99}$ estimation - comparison with aleatory enrichment

In the previous sections, it was demonstrated that the active learning technique reduces the confidence interval area representing the uncertainty on the quantile estimation with both methods (CI Area A and B). The active learning technique is based on an optimization loop that performs a

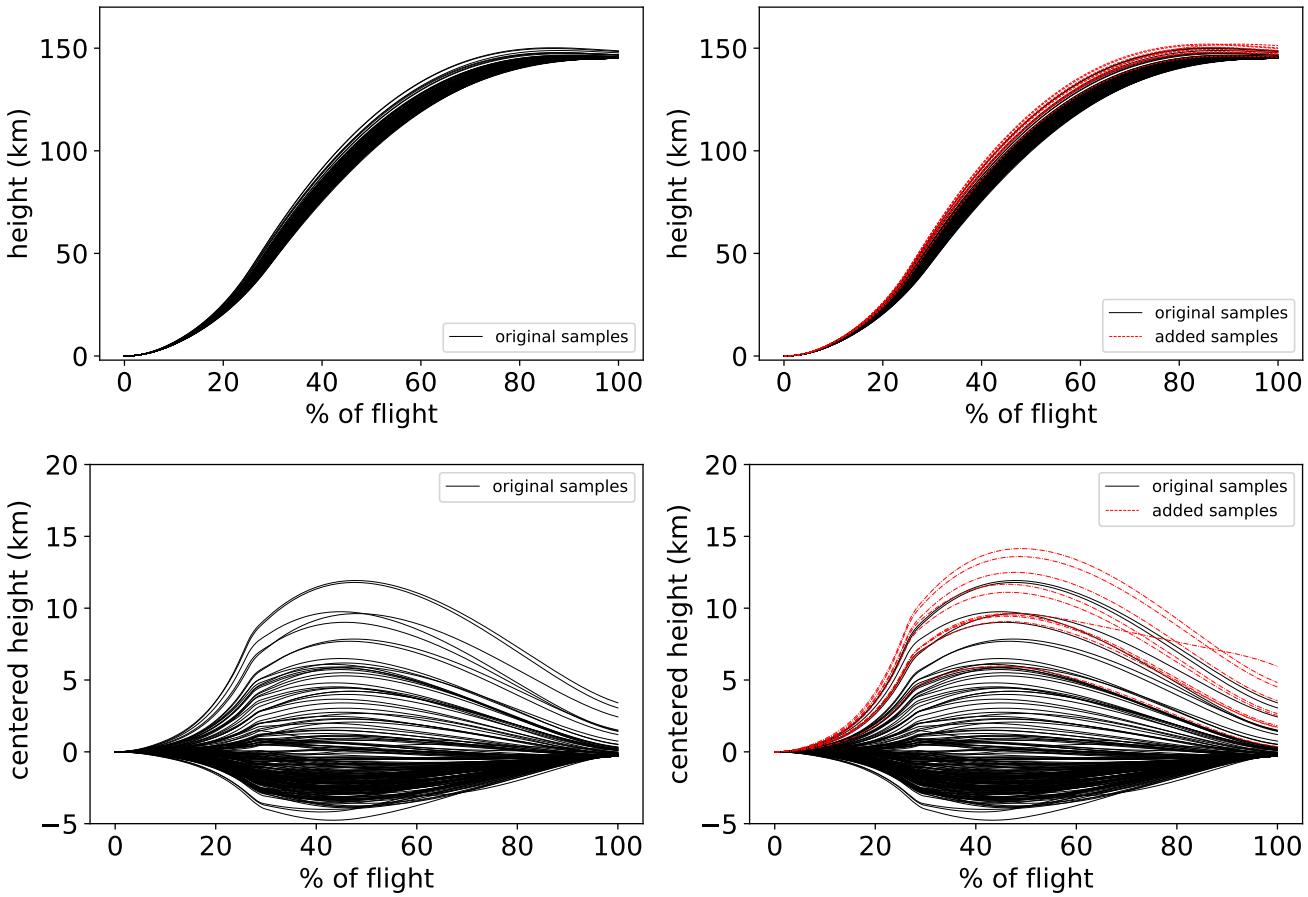


Estimation using the original set

Estimation using the enriched set

Figure 36: Quantile estimation variation after 10 optimization-based enriched samples using the CI area - B method

high number of surrogate model evaluations to try to identify the samples that contribute the most to the reduction of the uncertainty. As a result, the surrogate model is locally refined by adding new training samples close to the quantile to be estimated. Both methods were compared against an aleatory enrichment strategy, which consists in adding new training samples by sampling in an aleatory manner the distribution ϕ_U. In this case, the new samples are randomly distributed all over the domain and are not necessarily close to the quantile to be estimated. To prove that the results are independent of the initial training set ($\mathcal{U}_{M=200}$), 10 repetitions of the entire process were performed by choosing the initial training set in aleatory fashion among the available 1131 samples. For each repetition, 10 samples were added using the active learning technique and the aleatory enrichment strategy. The results using the CI Area - A method are displayed on Fig. (39). Using the active-learning-based enrichment the confidence interval area was reduced from an initial average of around 950 m to a final average after 10 enriched samples were added of around 400 m. The aleatory enrichment strategy stays around 950 through the 10 enrichment iterations. A similar analysis with using the CI Area - B method is displayed on Fig. (40). The average initial confidence interval area of around 2500 m was reduced down to around 800m after 10 enrichment iterations using the active learning strategy. The aleatory method also stays around the initial confidence interval average value of 2500 after the 10 enrichment iterations. Failing to reduce the confidence interval area.



Original samples | Original and added samples

Figure 37: altitude profile trajectories used for surrogate training using CI area - B method

5.1.5 $q_{0.99}$ estimation - validation set comparison

So far it has been demonstrated that the active learning strategy based on the CI Area - A and CI Area - B methods reduces the uncertainty on the quantile computation linked to the use of Gaussian processes. Furthermore, it has been shown that both strategies locally refine the same zone of the samples domain. This zone is close to the desired quantile to be estimated. In this section, we compare the estimated quantile using the surrogate model prediction with the quantile calculated using the validation set $\mathcal{V}_{V=500}$. Again, using the results from the 10 different repetitions of the whole procedure, and adding 10 samples per repetition, the distance between the estimated quantiles using the surrogate model and the validation set is assessed via the Root Mean Square Error (RMSE). For each repetition a RMSE value is obtained by comparing the two quantiles at each node of the time grid ($\mathcal{T}_N$).

The results for the CI Area - A method are displayed on Fig. (41) and those using the CI Area - B method on Fig. (42). It can be noticed that the RMSE for both methods gets reduced from its initial mean value of around 600m to 200m when the active learning technique is used. It is specially important to notice than the outlier data point of Fig. (41) corresponding to an RMSE above 2600 m for the quantile estimation using the initial training set got drastically reduced to below 500 m after the 10 enrichment iterations. Demonstrating the correction capacity of the active learning technique. For the aleatory enrichment strategy, the RMSE stays around its initial value and no clear improvement tendency is observed as more random samples are added. The former results for the active learning technique show its effectiveness to reduce the error

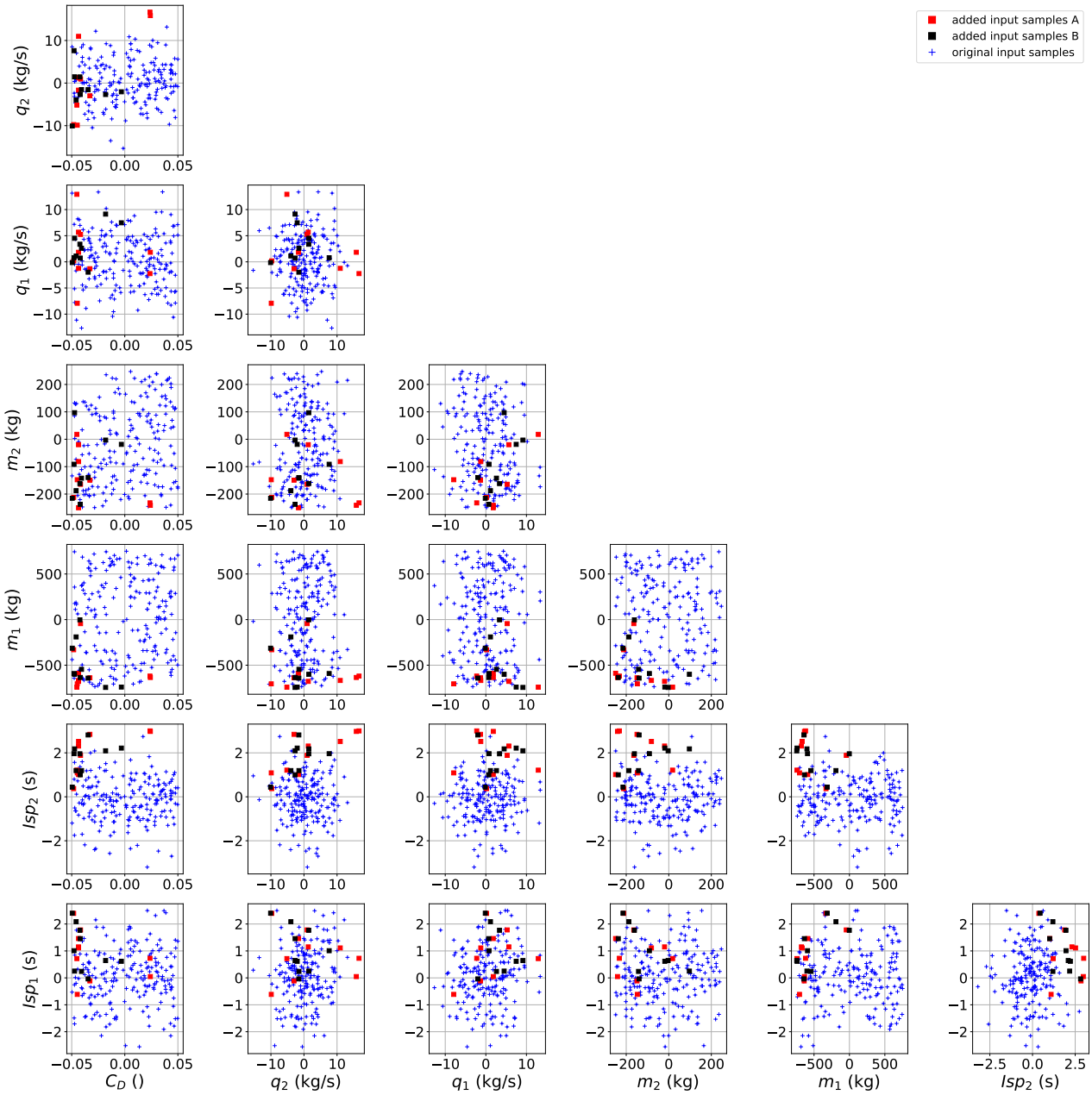
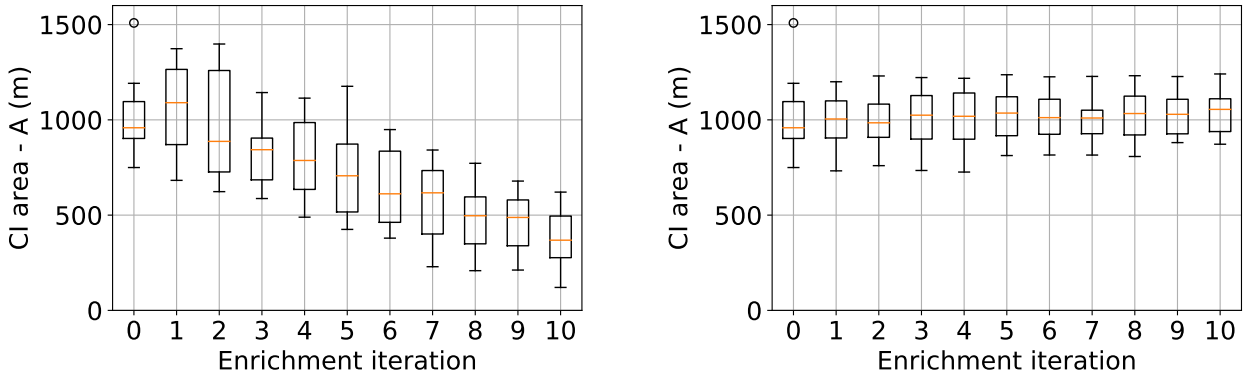


Figure 38: Enriched input training samples using CI area - A method and CI area - B method

estimation with respect to a validation set comprised of the same input sample and its associated responses. Nevertheless, it is important to remember that the validation quantile based on the sample $\mathbf{X}_{\mathbf{V}}^{*}$ is an imperfect estimation of the exact quantile, and that its error decreases as V increases. In a similar way, the quantile estimated using the active learning technique approaches the exact quantile as the size of $\mathcal{V}_R$ increases.

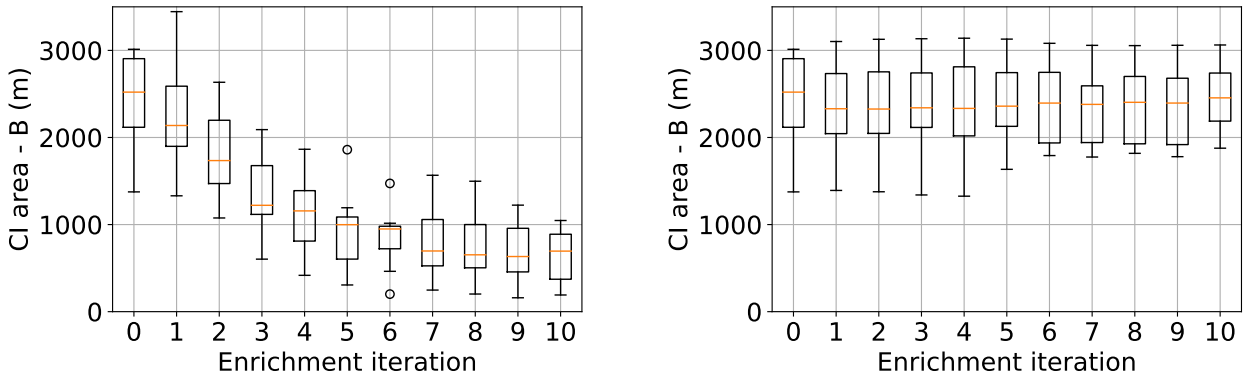
5.1.6 $q_{0.09}$ using active learning enrichment hybrid methodology

As it was shown that the CI Area - A and CI Area - B methods lead to the refinement of the surrogate model in the same zones of the input and output spaces, their advantages can be combined in a hybrid methodology that employs the CI Area - A method to calculate the confidence interval area at a known 95% level and respecting the spatial correlation imposed by the covariance kernel at every enrichment step of the active learning methodology. This means that the CI Area - A method is called only after a new sample is added to the training set. During



Active-learning-based enrichment Aleatory enrichment

Figure 39: Enrichment strategies using CI area - A method



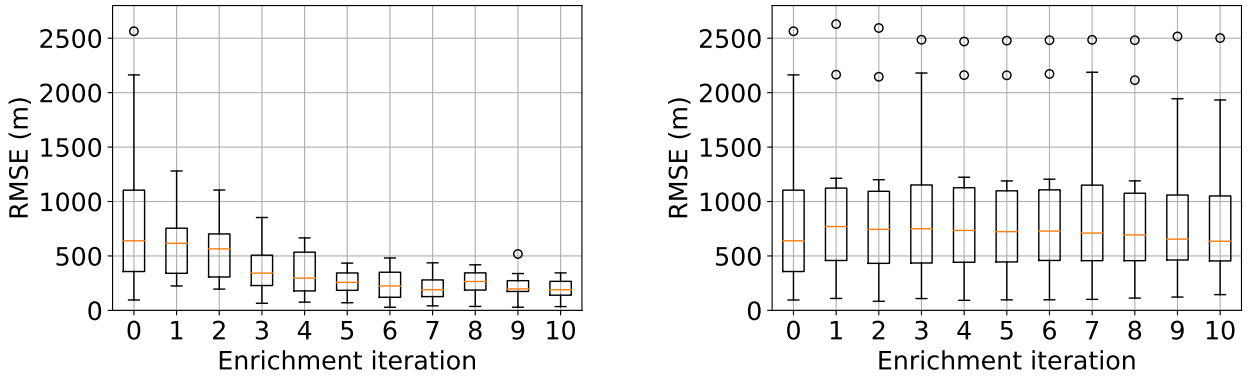
Active-learning-based enrichment Aleatory enrichment

Figure 40: Enrichment strategies using CI area - B method

the optimization stage, the CI Area - B method is used as it is computationally cheaper and the optimization stage represents the biggest computational cost of the algorithm. The confidence interval area reduction and the error reduction with respect to the validation set can be seen in Figs. (43) and (44). The area and the RMSE get reduced to similar average values of the ones reached using the active learning strategy with only the CI Area - A method.

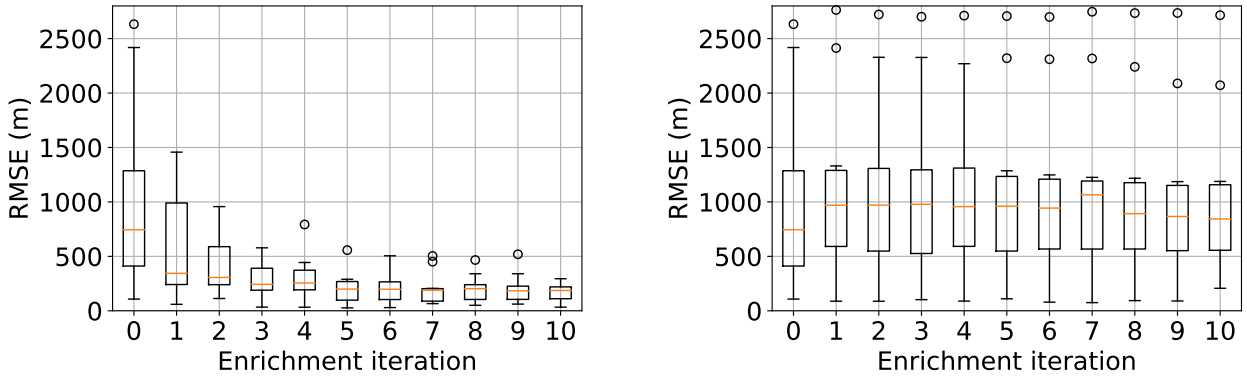
5.1.7 $q_{0.01}$ using active learning enrichment with CI Area - A method

A lower bound quantile for the altitude profile was also estimated using the active learning strategy with the same parameters used for $q_{0.99}$. The results are displayed in Figs. 45 and 46. The 95% confidence interval area is reduced after 10 enriched samples although the improvement is less than for the upper quantile $q_{0.99}$. This can be explained by the fact that the initial training set already had a high density of samples around the quantile area, hence the new samples have a less remarkable effect. In other words, for the initial training set, the estimations around the $q_{0.99}$ had more uncertainty, and hence, more room for improvement. In this analysis is important to highlight that even though the input space is randomly sampled, the output space uncertainty can be distributed in non-homogeneous ways, based on the effects of the original model. It can also be noticed in the figures that the validation quantile is outside the confidence interval between the 70% and 100% portions of the flight. This highlights the fact that the Gaussian process error gives a measure of how certain the prediction is but this measure does not replace the validation with actual model responses. The samples that were obtained with the active learning



Optimization-based enrichment Aleatory enrichment

Figure 41: Quantile RMSE using CI area - A method



Optimization-based enrichment Aleatory enrichment

Figure 42: Quantile RMSE using CI area - B method

technique for the quantiles $q_{0.01}$ and $q_{0.99}$ were added two an initial training set of 200 random samples. A metamodel was trained using the 220 samples an a prediction was issued for the quantile estimation set $\mathcal{V}_{R=500}$. The quantiles $q_{0.01}$ and $q_{0.99}$ of the output stochastic process were computed to create the flight envelope shown in Fig. (47).

5.2 Active learning for trajectory states of a launch vehicle MDAO

In a similar manner to the altitude state, the speed and heat flux evaluations in time were used within the active learning technique to improve the estimation of a given quantile. The hybrid methodology combining the CI Area - A and CI Area - B methods was used as it proved to work in a satisfactory manner for the $q_{0.99}$ computation of the altitude state.

The 95% confidence interval area for the estimation of the $q_{0.99}$ of the speed was reduced from 4.795 km to 3.971 km after 10 samples were added using the active learning technique with the hybrid methodology and the interpolation values of the Runge-Kutta solution obtained from the MDAO code. As a consequence, time is truncated at 430 s and not normalized (see Fig. 48). The CI Area is expressed in Km as it results from the integration of speed units in the non-normalized time domain. The estimated quantiles do not correspond to a unique trajectory, on the contrary, they are composed of different trajectories that change the order in which they are arranged as a function of time. This leads to interesting phenomena as the one that can be seen in Fig. (49) where the enriched samples that were added to refine the $q_{0.99}$ estimation

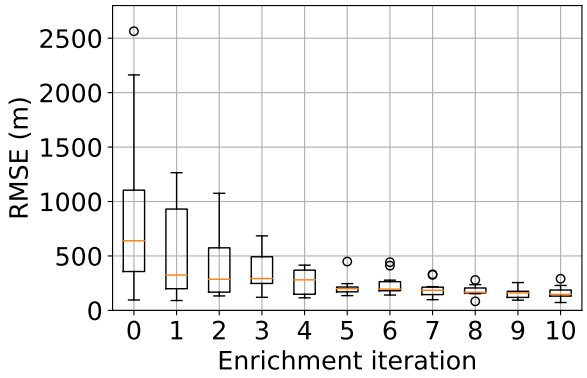


Figure 43: RMSE for $q_{0.99}$ estimation using hybrid method

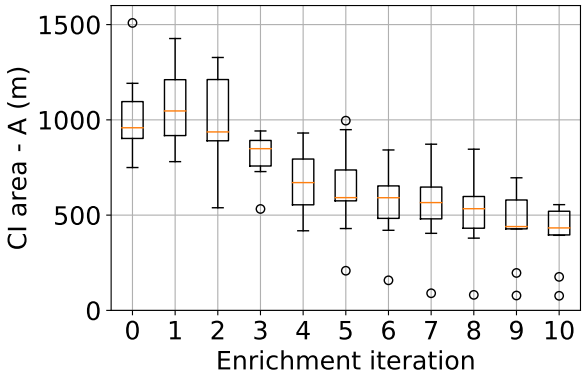
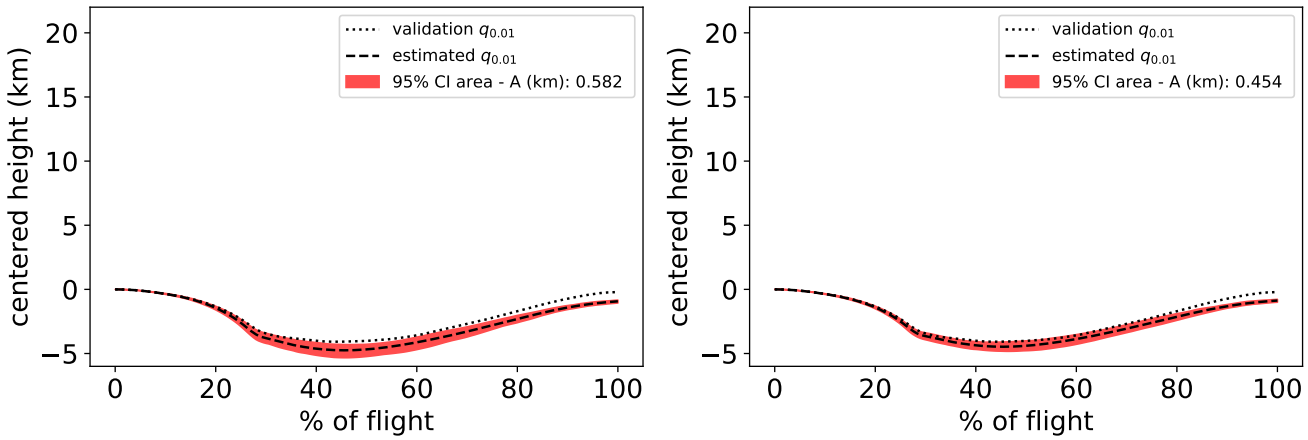


Figure 44: 95% CI Area - A for $q_{0.99}$ estimation using hybrid method



Estimation using the original set

Estimation using the enriched set

Figure 45: $q_{0.01}$ estimation variation after 10 active-learning-based enriched samples using the CI area - A method

between 0 and 90 s, are not near that quantile between 90 and 340 s. Three different sections of the output space can be identified where the added samples refine locally the given subdomain. They comprise the intervals [0,90]s , [90, 290]s and [290, 340]s.

The 95% confidence interval area for the estimation of the $q_{0.99}$ of the heat flux was found to be 0.301 MW/m^2. After 10 enrichment runs it was reduced down to 0.233 MW/m^2 (see Fig. 50). A similar feature to that of the speed plots can be spotted on the new samples that were added to refine the $q_{0.99}$ of the heat flux, where the samples that refine the estimation of the surrogate model between 0 and 15 % of the flight, refine a complete different zone between 15 and 40 % of the flight. Nevertheless, samples near the $q_{0.99}$ were added all over the domain.

6 Conclusions and perspectives

In this work, an active learning technique for quantile field variable estimation was developed. It relies on a surrogate model comprised by a model order reduction method (Karhunen-Loève expansion) and Gaussian process to reduce the computational cost of uncertainty propagation through multidisciplinary optimization processes involving optimal control. The methodology can be applied for example for the estimation of flight envelopes of states (*e.g.,* velocity, altitude) and

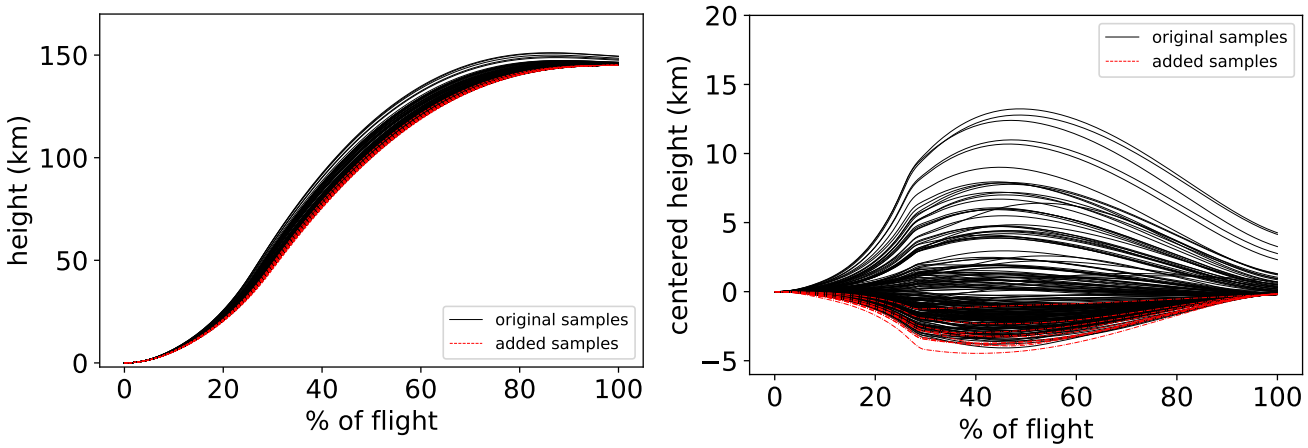


Figure 46: altitude profile trajectories used for surrogate training using CI area - A method

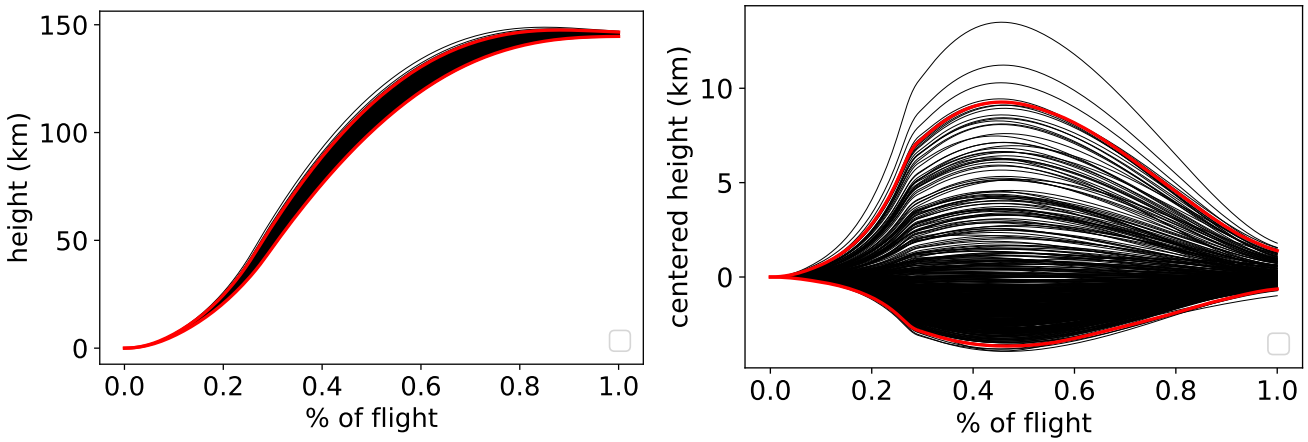
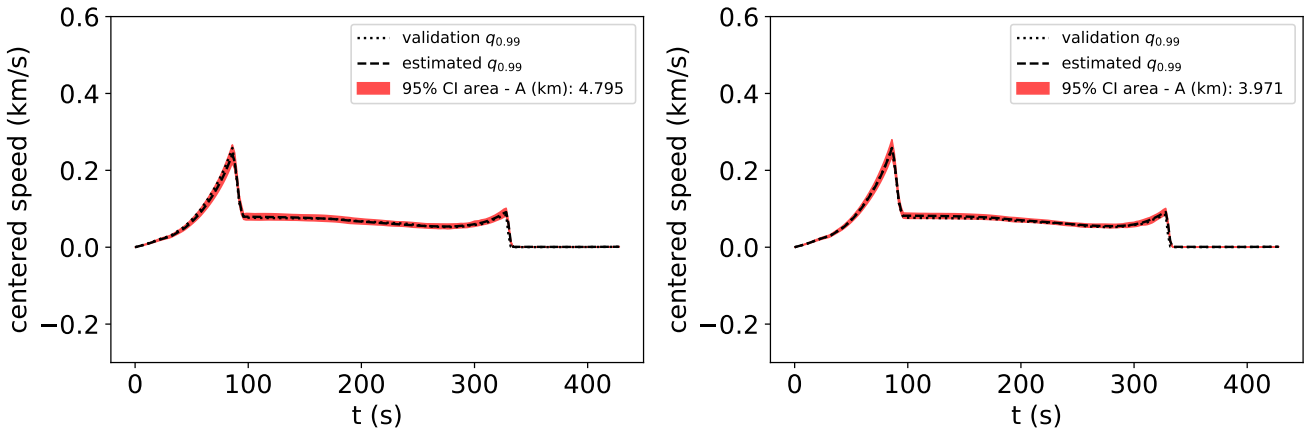


Figure 47: Flight envelope using the quantiles (red curves) $q_{0.01}$ and $q_{0.99}$. Estimated samples shown in black.

quantities of interest (*e.g.,* heat flux, dynamic pressure). Two different methodologies to calculate the uncertainty of an estimated quantile were proposed, namely the CI Area A and B methods. The first method is based on the generation of random Gaussian process trajectories and leads to the calculation of a known confidence interval for the quantile estimation. The second method is based on the analytic propagation of the variance model of the Gaussian processes through the Karhunen-Loève expansion and is less expensive computationally, nevertheless, it does not follow the spatial correlation imposed by the covariance kernel and the level of confidence of its prediction varies as a function of the quantile sample size. An example case on the optimal states quantile estimation resulting from the uncertain multidisciplinary optimization of a two-stage-to-orbit vehicle was demonstrated. A comparison of the active learning technique based on the CI Area A and B methods was done and it was shown that both methods refine the same zones of the input and output spaces. It was also shown that both methods lead to the reduction of the quantile uncertainty and the quantile estimation error by more than twofold when the active learning technique is used to enrich with 10 samples an original training set of 200 samples. No uncertainty or error reduction was achieved when the same number of samples were added randomly. This illustrates the advantage of using the active learning methodology to select in an intelligent manner the samples that improve the predictions of the surrogate model in a specific region. A hybrid methodology combining the CI Area - A and CI Area - B methods was also studied for the prediction of quantiles of the speed and heat flux trajectories.

Future work could include an attempt to link the CI Area A and B methods. It is desired



Estimation using the original set

Estimation using the enriched set

Figure 48: Quantile estimation variation after 10 active-learning-based enriched samples using the hybrid method for the speed state

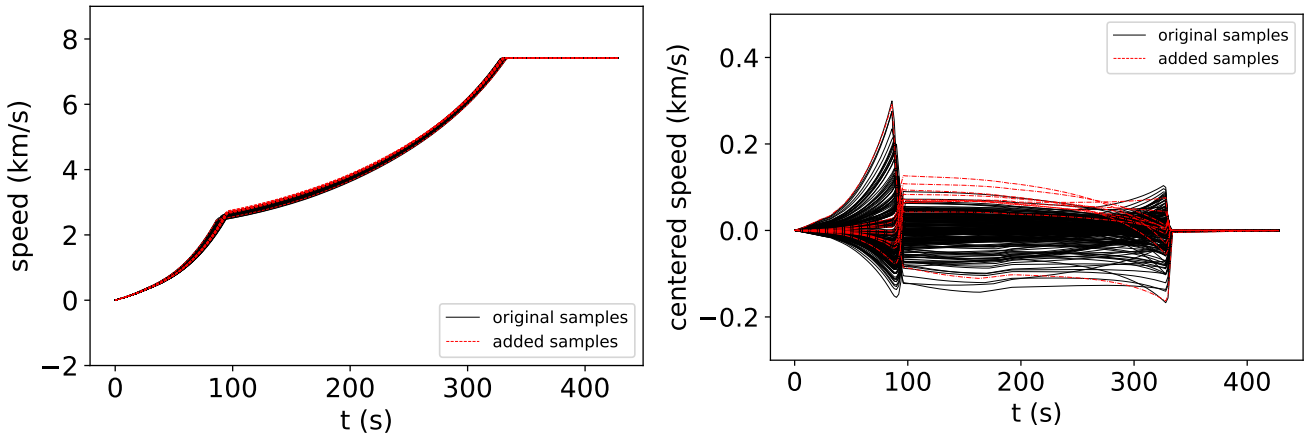
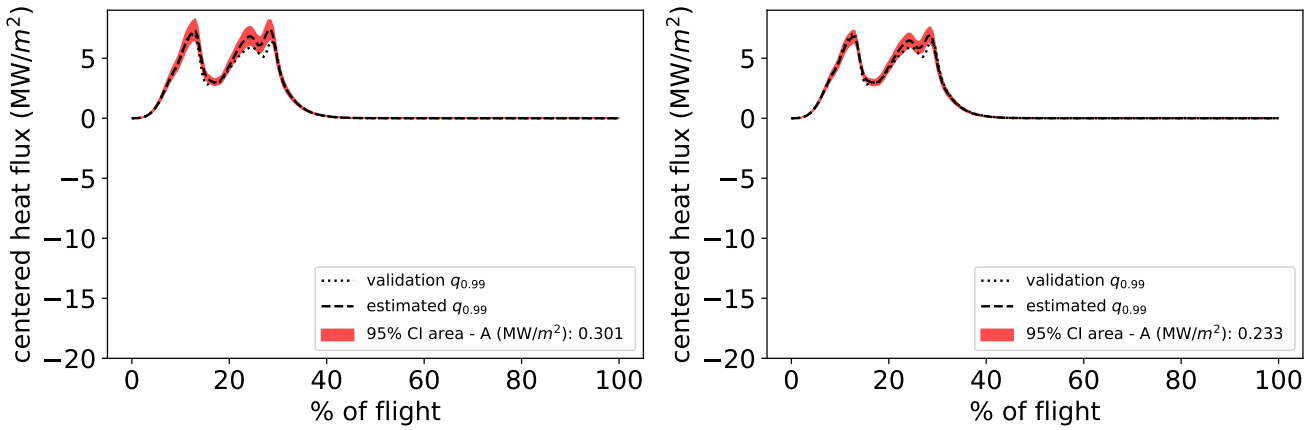


Figure 49: Speed profile trajectories used for surrogate training using hybrid method

to devise a strategy that allows to know the confidence level of the estimation using the CI Area - B method. A possible path to reach this goal would be order statistics. Recurrence relations can be used to calculate the K^{th} order statistic of a set of normally distributed variables [18] and an associated quantile could be found. Recurrence relations allow to compute extreme order statistics quite simply, but complexity increases when going further from the extremes. In a more complex approach, joint distributions could be device to express the spatial correlation of the quantile computation sample ($\boldsymbol{X}_R^*$) and finding the K^{th} order statistics of a joint normally distributed variables. Another future development could be in the context of trajectory envelopes, where two quantiles enclose a portion of the trajectories, it could be analyzed which one of the two quantiles representing the upper and lower bounds of the envelope should be refined the most if a fixed budget of enrichment runs is given.

An extension of the active learning methodology to multi-dimensional fields would also be desired. In principle, the mathematical development would be similar but the programming code and the refinement criterion would need to be modified. This would allow to calculate quantiles of for example the pressure distribution over an aerodynamic surface. The coupling with multi-fidelity techniques is also a perspective development. This would allow to optimize the use of computational resources to refine the surrogate model in critical zones using different fidelity models. An example case with the launch vehicle MDAO problem would consist of the call to a high fidelity version of the code, with a high number of Legendre-Gauss-Lobatto (LGL)



Estimation using the original set Estimation using the enriched set

Figure 50: Quantile estimation variation after 10 active-learning-based enriched samples using the hybrid method for the heat flux

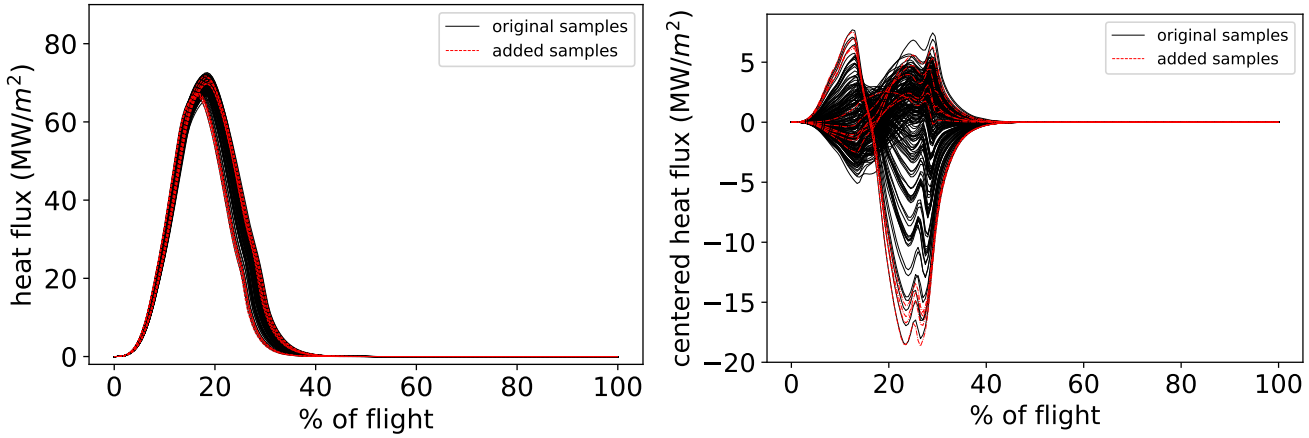


Figure 51: Heat flux trajectories used for surrogate training using hybrid method

nodes for the discretization of the trajectory, to add a high quality sample to the training set that would allow to refine a zone of the output space that is fundamental for the quantile computation. Coupled with a low fidelity version, with less LGL nodes, to refine the model in a fast manner in the zones that have less influence in the accuracy of the computed quantile.

References

[1] R. D. Falck and J. S. Gray, "Optimal Control within the Context of Multidisciplinary Design, Analysis, and Optimization," in *AIAA Scitech 2019 Forum*, American Institute of Aeronautics and Astronautics, 2019.

[2] L. Brevault and M. Balesdent, "Uncertainty quantification for multidisciplinary launch vehicle design using model order reduction and spectral methods," *Acta Astronautica*, vol. 187, pp. 295–314, Oct. 2021.

[3] M. Moustapha, B. Sudret, J.-M. Bourinet, and B. Guillaume, "Quantile-based optimization under uncertainties using adaptive Kriging surrogate models," *Structural and Multidisciplinary Optimization*, vol. 54, pp. 1403–1421, Dec. 2016.

[4] J. Kim and J. Song, "Quantile surrogates and sensitivity by adaptive Gaussian process for efficient reliability-based design optimization," *Mechanical Systems and Signal Processing*, vol. 161, p. 107962, Dec. 2021.

[5] J. Valderrama, L. Brevault, M. Balesdent, and A. Urbano, "All-At-Once MDO formulation for coupled optimization of launch vehicle design and its trajectory using a pseudo spectral method," June 2021.

[6] L. Brevault and M. Balesdent, "Uncertainty-Based Multidisciplinary Design Optimization (UMDO)," in *Springer Optimization and Its Applications*, vol. 156 of *Aerospace System Analysis and Optimization in Uncertainty*, pp. pp.235–292, 2020.

[7] L. Brevault, M. Balesdent, N. Bérend, and R. L. Riche, "Decoupled Multidisciplinary Design Optimization Formulation for Interdisciplinary Coupling Satisfaction Under Uncertainty," *AIAA Journal*, vol. 54, no. 1, p. 186, 2016.

[8] R. Zardashti, M. Jafari, S. M. Hosseini, and S. A. S. Arani, "Robust Optimum Trajectory Design of a Satellite Launch Vehicle in the Presence of Uncertainties," *Journal of Aerospace Technology and Management*, vol. 12, Aug. 2020. Publisher: Departamento de Ciência e Tecnologia Aeroespacial.

[9] J. Fisher and R. Bhattacharya, "Optimal Trajectory Generation With Probabilistic System Uncertainty Using Polynomial Chaos," *Journal of Dynamic Systems, Measurement, and Control*, vol. 133, Nov. 2010.

[10] X. Li, P. Nair, Z. Zhang, L. Gao, and C. Gao, "Aircraft Robust Trajectory Optimization Using Nonintrusive Polynomial Chaos," *Journal of Aircraft*, vol. 51, pp. 1592–1603, Sept. 2014.

[11] F. Wang, Y. Shuxing, F. Xiong, Q. Lin, and J. Song, "Robust trajectory optimization using polynomial chaos and convex optimization," *Aerospace Science and Technology*, vol. 92, June 2019.

[12] F. Xiong, Y. Xiong, and B. Xue, "Trajectory Optimization under Uncertainty based on Polynomial Chaos Expansion," 2015.

[13] B. Sudret, "Stochastic Finite Element Methods and Reliability A State-of-the-Art Report," p. 190.

[14] R. Schoebi, B. Sudret, and J. Wiart, "Polynomial-Chaos-based Kriging," *arXiv:1502.03939 [stat]*, Feb. 2015. arXiv: 1502.03939.

[15] G. Matheron, "Principles of geostatistics," *Economic Geology*, vol. 58, pp. 1246–1266, Dec. 1963.

[16] M. Balesdent and L. Brevault, "MULTIDISCIPLINARY DESIGN OPTIMIZATION, APPLICATION TO AEROSPACE VEHICLE DESIGN," 2020.

[17] F. Laurin, N. Tableau, M. Kaminski, Z. Aboura, and F. Bouillon, "Validation of the onera damage model through comparisons with multi-instrumented structural tests on interlock woven ceramic matrix composites," in *16th European Conference on Composite Materials*, (SEVILLE, Spain), June 2014.

[18] G. Cao and M. West, "COMPUTING DISTRIBUTIONS OF ORDER STATISTICS," p. 11.

[19] N. Hansen, "The CMA Evolution Strategy: A Tutorial," Apr. 2016. arXiv: 1604.00772.

A Legendre-Gauss-Lobatto (LGL) orthogonal collocation

In orthogonal collocation methods a discretization of the states in time is done in order to transcribe the optimal control problem into a Non-Linear Programming (NLP) problem. The LGL transcription requires a 2 step evaluation of the ODE per optimizer iteration. The first evaluation is used to fit an Hermite interpolating polynomial on a set of nodes used to discretize the states and the second one to assess how well the polynomial approximates the dynamics of the system.

For the launch vehicle MDAO code, the optimal control is divided in the 8 phases defined in Fig. (13). At the same time, each phase is divided into a different amount of segments that can be specified by the user. The LGL transcription of order 3 was used. In this case, a segment has a state discretization node at the each of its two extremes and a collocation node at the center as shown in figure 52. In an uncompressed transcription, two adjacent segments are bounded by using a continuity constraint,in such a way that the state discretization nodes at the junction point for each segment take the same values. Continuity constraints are also used to bound two adjacent phases.

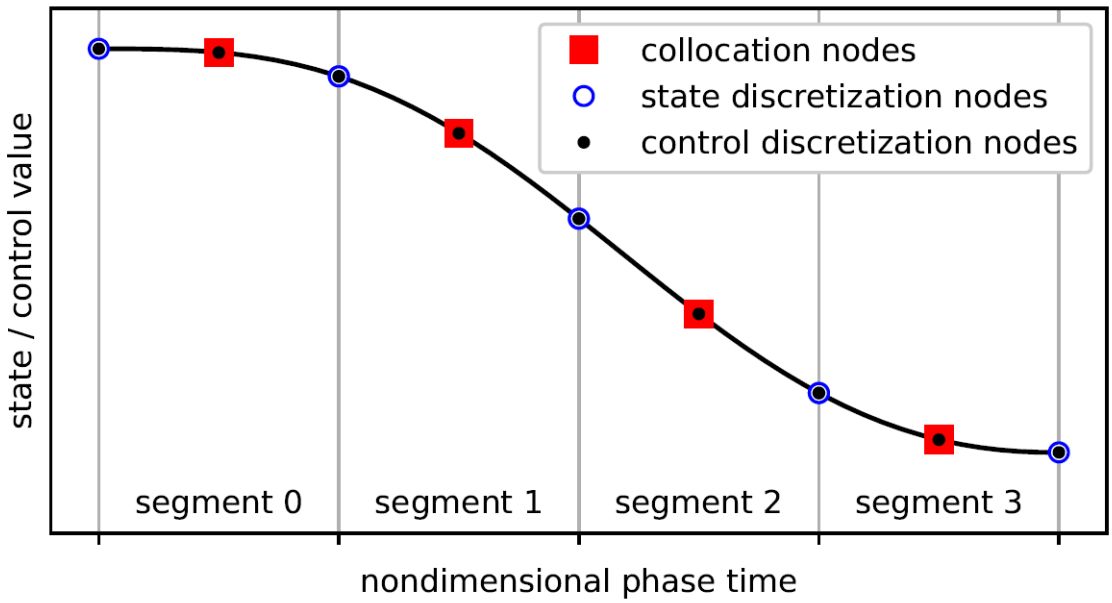


Figure 52: Legendre-Gauss Lobatto transcription of order 3. Taken from [1]

The LGL transcription allows the optimization of dynamic controls via the control discretization nodes. This nodes are defined at the same locations in time as the state discretization nodes and the collocation nodes, resulting in 3 control discretization nodes per segment. Dynamic controls are not implemented for the launch vehicle MDAO code, nonetheless, the models are built up to account for throttle as a dynamic control so that it can be easily modifiable in the future.

Hence, the vector of state variables $\boldsymbol{x}$ is transformed into a discrete set, $\boldsymbol{x}_{t_a}$, containing the state discretization nodes $\boldsymbol{x}_{t_d}$ and the collocation nodes $\boldsymbol{x}_{t_c}$. The time is also discretized to form t_a, this set is completely defined by the initial time t_{p_0} and the phase duration Δt_p of all phases.

As defined in [1], the first step for the optimization consists in an input from the user providing an initial guess for $\boldsymbol{x}_{t_d}$, the time parameters controlling t_a, and the design variables $\boldsymbol{z}_t$. Then, the dynamics are evaluated to obtain the state rates at the state discretization nodes following

$$\dot{\boldsymbol{x}}_{t_d} = \boldsymbol{f}_{ode}[\boldsymbol{x}_{t_d}, t_d, \boldsymbol{z}_t] \tag{A.1}$$

Using $\dot{\boldsymbol{x}}_{t_d}$ and $\boldsymbol{x}_{t_d}$, an Hermite polynomial is fitted so that the value of the states and the state rates can be approximated at the collocation nodes by using

$$\boldsymbol{x}_{t_c} = \frac{2}{t_{seg}}[A_d]\boldsymbol{x}_{t_d} + [B_d]\dot{\boldsymbol{x}}_{t_d} \tag{A.2}$$

$$\boldsymbol{x}'_{t_c} = [A_i]\boldsymbol{x}_{t_d} + \frac{t_{seg}}{2}[B_i]\dot{\boldsymbol{x}}_{t_d} \tag{A.3}$$

Where $[A_i]$ and $[B_i]$ are the Hermite interpolation matrices, $[A_d]$ and $[B_d]$ are the Hermite differentiation matrices and t_{seg} is the duration of the segment of the current node. A value for the state rate at the discretization node can also be obtained by evaluating the dynamics (set of coupled ordinary differential equations) with the approximated state values $\dot{\boldsymbol{x}}_{t_c}$

$$\dot{\boldsymbol{x}}_{t_c} = \boldsymbol{f}_{ode}[\boldsymbol{x}_{t_c}, t_c, \boldsymbol{z}_t] \tag{A.4}$$

The difference between the derivative at the collocation node obtained by evaluating the dynamics, $\dot{\boldsymbol{x}}_{t_c}$, and the approximated derivative obtained with the Hermite polynomial $\dot{\boldsymbol{x}}'_{t_c}$ represents the collocation defect. This defect is formulated as an equality constraint following

$$\boldsymbol{\Delta} = \boldsymbol{x}'_{t_c} - \frac{t_{seg}}{2}\dot{\boldsymbol{x}}_{t_c} = \boldsymbol{0} \tag{A.5}$$

Considering the segment i of a state with N segments, the continuity constraints for the state values read $\boldsymbol{C}_x(\boldsymbol{x}_d) = \boldsymbol{0}$. And considering the phase p of a state with P phases the time continuity constraints to link the time read $\boldsymbol{C}_t(t_d) = \boldsymbol{0}$.

B ONERA Damage Model for Composites with Ceramic Matrix (ODM-CMC)

The ODM-CMC [17] was used to test and develop the active learning strategy as it produces trajectories at a cheap computational cost. The trajectories describe the evolution of the stress σ_{xx} in MPa as function of the normalized strain ϵ_{xx} in a tensile test of a probe made out of a composite material with ceramic matrix. The components of the input uncertain that were used are described in table 3. 500 trajectories obtained with the ODM-CMC model are shown in Fig. (53).

Table 3: Probability distribution of the components of the uncertain random vector ODM-CMC

Name	Notation	Model (mean, standard deviation)
Young modulus	E_0	$\mathcal{N}(180, 36)$(additive, GPa)
Damage evolution celerity	ycs	$\mathcal{N}(5, 0.144)$(additive, Pa)
Damage threshold	$y0s$	$\mathcal{N}(4.27, 1.08)$(additive, MPa)
Damage saturation	dc	$\mathcal{N}(5.33, 0.36)$(additive, $-$)

The active learning technique using the CI Area - B method was used to add 10 samples to an initial training set of 4 samples, with the purpose of estimating the $q_{0.99}$. The quantile estimation sample $\mathcal{V}_R$ was comprised of 1000 realizations. The evolution of the area of the quantile estimated with the original set and the quantile estimated with the original set plus 10 enriched samples can be seen in

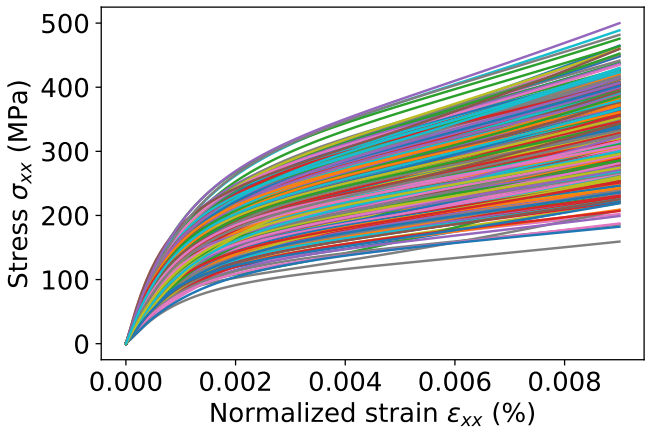


Figure 53: Output stochastic process of size 500 for the ODM-CMC

Fig. (54). The new inputs found with the active learning strategy are close to the desired quantile $q_{0.99}$ and are depicted in Fig. (55). The same samples in the input space are show in Fig. (57). The whole procedure was repeated 10 times to assess the generalized behavior of the confidence interval area on the estimated quantile. In Fig. (56), it can be seen that the active learning strategy effectively reduces the area after 10 iterations.

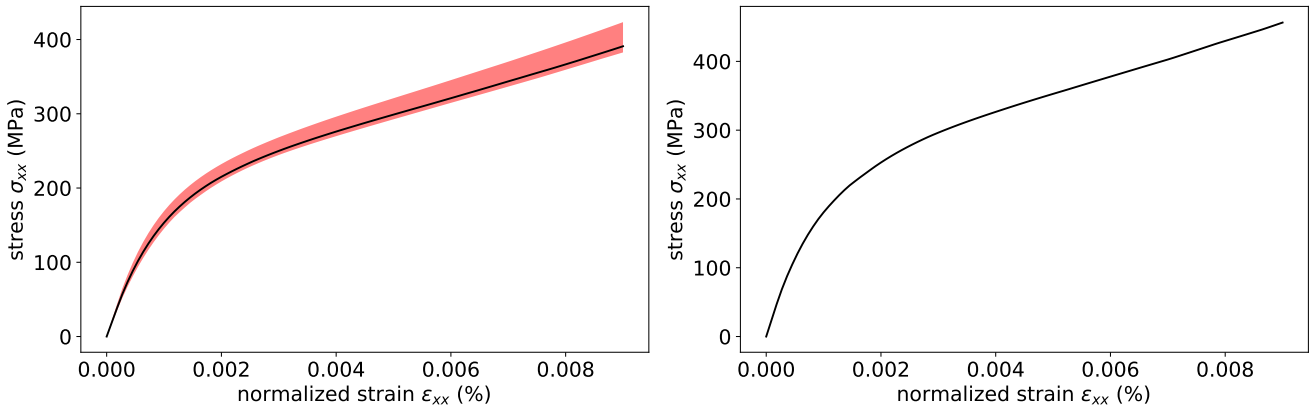


Figure 54: Confidence interval evolution on the estimated quantile for the ODM-CMC

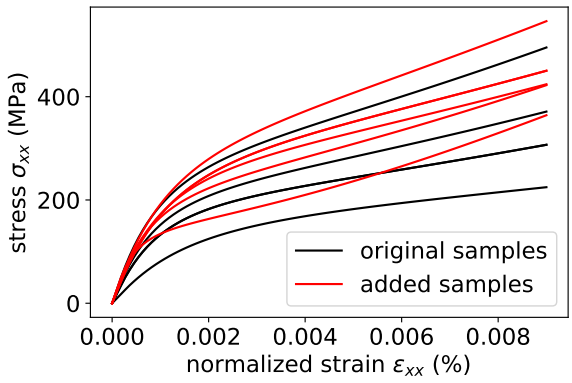


Figure 55: Added output samples for the ODM-CMC

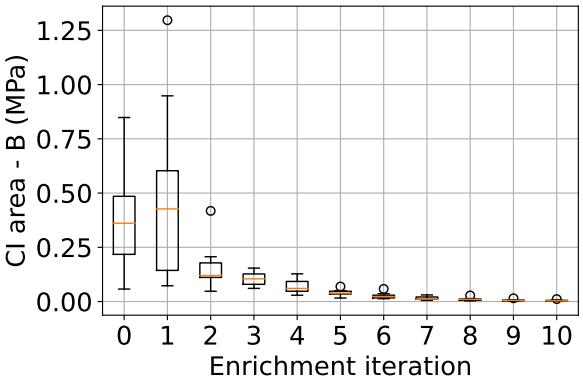


Figure 56: Confidence interval area evolution on the estimated quantile for 10 repetitions of the ODM-CMC

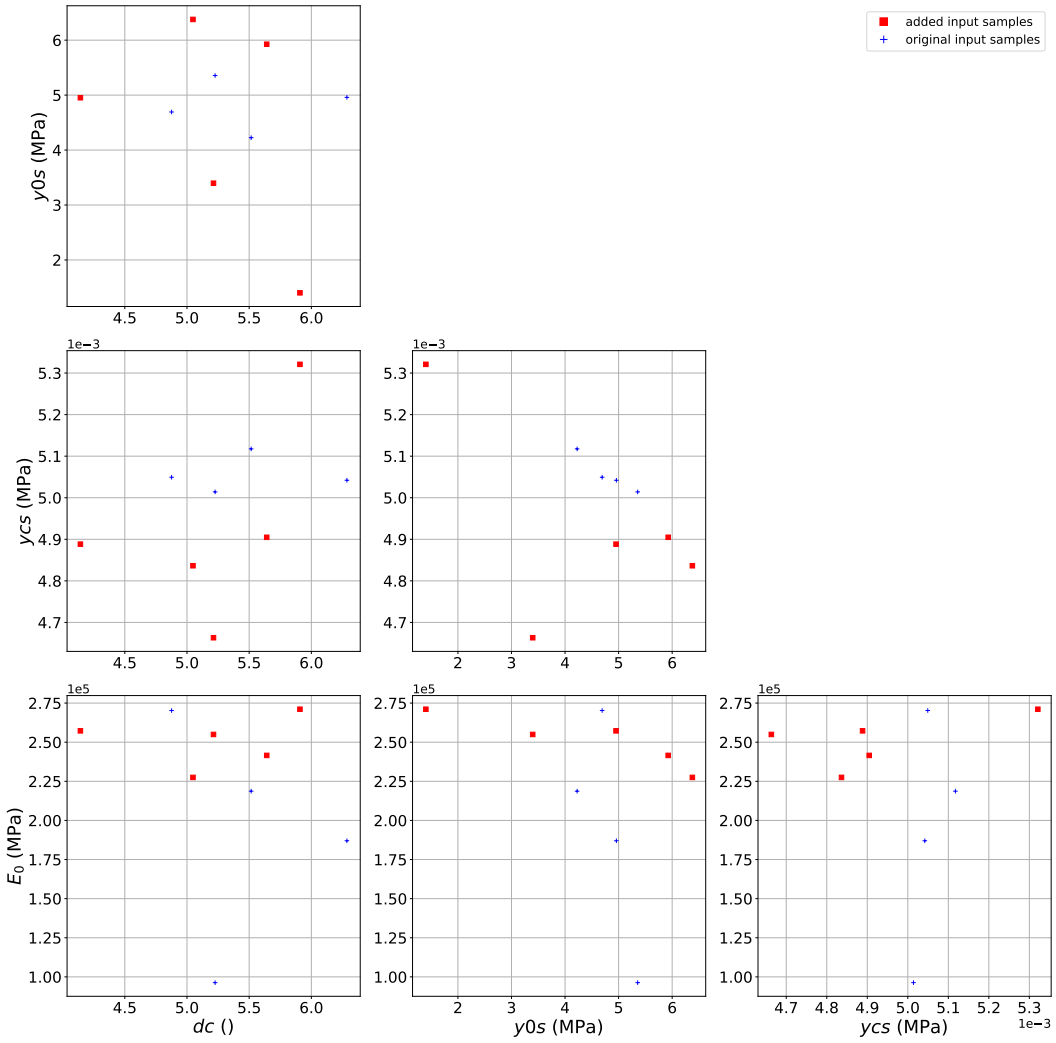


Figure 57: Enriched input samples for the ODM-CMC