Multi-Objective Trajectory Optimisation under Epistemic Uncertainty

PhD Thesis

Simão da Graça Marto

Aerospace Centre of Excellence

Mechanical and Aerospace Engineering

University of Strathclyde, Glasgow

January 29, 2026

This thesis is the result of the author's original research. It has been composed by the author and has not been previously submitted for examination which has led to the award of a degree.

Abstract

This thesis proposes innovative methodologies for the many-objective optimisation of spacecraft trajectories under epistemic uncertainty, which can also have application outside the domain of trajectory optimisation. Epistemic uncertainty refers to a type of uncertainty that is due to a lack of knowledge, and which cannot be adequately quantified using standard probability theory, making it more challenging to quantify in practice, as computational costs can become prohibitively high, in particular when the number of uncertain variables is increased. Thus, the main challenge addressed by this thesis was reducing the computational cost of quantifying epistemic uncertainty in the context of trajectory optimisation. Several strategies were developed and used in combination.

Since the quantification of epistemic uncertainty is an integral part of the objectives being optimised, strategies are developed that reduce the number of evaluations of these objectives. A surrogate model of the lower expectation is combined with a dimensionality reduction technique to contain the computational cost. These dimensionality reduction techniques take the form of control maps, which map the space of control laws to a lower dimensional space, reducing the dimensionality of the search space. Several control maps are tested and compared.

The second approach is to reduce the computational cost of quantifying the epistemic uncertainty. Techniques are proposed for some general formulations of epistemic uncertainty. These are accompanied with studies of the scalability of their computational cost with the number of epistemic variables.

To optimise trajectories while accounting for the responsiveness of the control law to observations of the state during the trajectory, sampling the uncertain variables and

finding the optimal control law for each case can become prohibitively expensive, even in the context of standard probability theory. Dynamic programming strategies have been applied, in the literature, to stochastic optimal control. A key innovation in this work is a dynamic programming formulation for epistemic uncertainty. This allows designing trajectories where the control law responds to observations and corrects for deviations, while still efficiently quantifying the epistemic uncertainty. Additionally, by splitting the trajectory into segments affected by different epistemically uncertain variables, this limits the growth of the computational cost with the number of epistemic variables. Additional key contributions were theoretical proofs that bound the value of this estimate with respect to other epistemic uncertainty quantifications of interest, in a way which is applicable to many commonly used formulations of epistemic uncertainty.

These methodologies are incorporated into algorithms which are applied to test cases consisting of spacecraft equipped with low thrust engines, whose parameters are affected by epistemic uncertainty. These consist of an asteroid rendezvous mission, and an asteroid fly-by tour of four asteroids in the asteroid belt. For these, a set of control laws are obtained, describing the evolution of the thrust magnitude and direction over time, and launch parameters, which are robust with respect to epistemic uncertainty in the system parameters. The algorithms used are empirically and theoretically shown to scale well with the number of uncertain variables.

Contents

List of Figures

List of Tables

Nomenclature

Symbols used in this paper are listed in this section, ordered alphabetically, with Latin letters, Greek letters, and acronyms in separate groups.

We mostly follow the convention that lower-case letters represent scalar quantities, bold lower case letters represent vector quantities, upper-case letters represent infinite sets, and calligraphic letters, such as $\mathcal{Q}$, represent finite sets. Exceptions to these rules include, but are not limited to, E, $\underline{\mathrm{E}}$, I_{sp}, P_1, P_2, Q_1, Q_2, and T which are scalars despite being upper-case letters, and $\mathbf{F}$ which is a vector quantity. Every bold letter represents a vector quantity.

When a variable appears both in normal and bold form, such as ν and $\boldsymbol{\nu}$, the latter is a vector of the former, so only the normal letters are shown in this list. Upper case letters will generally be used to represent the infinite sets containing their lower case versions such as $u \in U$ and $\nu \in N$, and calligraphic versions are the same for when the set is finite, e.g. $u \in \mathcal{U}$ and $\nu \in \mathcal{N}$. Apart from a few exceptions for symbols that occur more often, these set variants won't be repeated when the symbol for their elements is already listed.

Likewise, symbols where diacritics, subscripts, or superscripts are added to distinguish from another symbol to represent different quantities of the same type are not usually repeated in this list either, unless the distinction is significant.

Variables used only in a single section are defined locally and are not listed here.

Latin symbols

$\mathbf{a}$	Acceleration vector
a	Semi-major axis
AU	Astronomical unit, equal to approximately 149.6×10^9 m
$\mathbf{b}$	System dynamics depending on $\mathbf{x}$ only
$b_{j;q}$	Univariate Bernstein basis polynomial
$B_{\mathbf{j};\mathbf{q}}$ or $B_{\mathbf{j}}$	Multivariate Bernsten basis polynomial
$\mathbf{c}$	Coefficient vector
$\mathbf{d}$	Controllable system dynamics
$\underline{\mathrm{E}}$	Lower expectation
E	Mathematical expectation
e	Orbital eccentricity
$\mathbf{F}$	Stochastic process affecting system dynamics
$\mathbf{f}$	System dynamics depending on $\mathbf{x}$, $\mathbf{u}$ and $\boldsymbol{\xi}$
$\mathbf{f}_{\mathrm{CMap}}$	Control map. For variants, see Section 2.7.
$\mathbf{g}$	Vector of path constraints
$\mathbf{h}$	Quantity of interest
$\mathbf{h}_K$	Vector with samples of quantity of interest
I_{sp}	Specific impulse
$\mathcal{J}$	Set of multi-indices (vectors of indices)
j	Index of basis function

$\mathcal{K}$	Set with n samples of Ξ
L	True longitude
$\mathcal{M}_k[p]$	Marginal distribution of $\boldsymbol{\xi}_k$ when $\boldsymbol{\xi} = [\boldsymbol{\xi}_1, \boldsymbol{\xi}_2, \ldots]$ follows distribution p
M	Number of segments a trajectory is split into
m	System mass
m_p	Propellant mass spent
N	Set of thresholds $\boldsymbol{\nu}$
n	Number of samples or kernels used to estimate and expectation or lower expectation [1]
N_k^B	Multi-index neighbourhood defined in Eq. (3.18)
N_k^S	Multi-index neighbourhood defined in Eq. (3.19)
n_ξ	Number of uncertain variables
N_a	Size of the archive used by MACS
N_E	Number of evaluations of E performed to estimate $\underline{\mathrm{E}}$.
n_f	Number of surrogate model evaluations by MACS
N_p	Number of agents used by MACS
N_s	Number of points returned by MACS that are evaluated and used to re-train the model at each iteration
N_{init}	Number of initial training points for the surrogate model
$N_{\text{re-train}}$	Number of re-training iterations, i.e., number of times MACS is run and the surrogate model is re-trained

[1] n also refers to the mean motion, but this use is restricted to Section 2.3 and explicitely stated therein.

n_{LT}	Number of coast/thrust arcs
$\mathcal{P}$	Set of all valid probability density functions
p	Probability density function
P_1, P_2	In-plane equinoctial elements
$\mathcal{Q}$	Subset of $\mathcal{P}$ that defines what is known about the distribution of the uncertain variables
q	Polynomial degree
Q_1, Q_2	Out-of-plane equinoctial elements
$\mathbf{r}$	Cartesian position vector in a heliocentric frame
r	Heliocentric distance
s	Auxiliary variable
T	Thrust magnitude
t	Time
t_D	Departure time
$\mathbf{u}$	Control law
U	Space of control laws
$\mathbf{v}$	Cartesian velocity vector in a heliocentric frame
v_∞	Magnitude of the hyperbolic excess velocity
$\mathbf{x}$	State vector
$\mathbf{y}$	Control law transcription
Y	Set of control law transcriptions
$\mathbf{z}$	Proxy control vector, which is passed to a control map to obtain a control vector $\mathbf{y}$.

Greek symbols

α	Thrust azimuth
β	Thrust declination
γ	Azimuth of the hyperbolic excess velocity
Δr	Distance to target
Δv	Relative speed to target
δ	Declination of the hyperbolic excess velocity
$\mathbf{\Delta L^{OFF}}$	True longitude length of a coast arc
$\mathbf{\Delta L^{ON}}$	True longitude length of a thrust arc
$\boldsymbol{\epsilon}$	Vector of constants
ζ_i	Kernel functions used to approximate a probability distribution
η_i	Auxiliary functions used to model stochastic processes
θ	True anomaly
λ	Auxiliary variable
μ	Standard gravitational parameter
ν	Threshold used to define a target set
$\overline{\nu}_{\hat{\mathrm{E}}}(h)$	Upper Quantile - threshold ν such that $\underline{\mathrm{E}}\left(h < \nu\right) = \hat{\mathrm{E}}$
$\boldsymbol{\xi}$	Uncertain vector
σ	Standard deviation
Ξ	Space of uncertain vectors
$\xi_{I,i}$	Uncertain parameter in the model for the specific impulse I_{sp}

$\xi_{T,i}$	Uncertain parameter in the model for the thrust at 1 AU, T_{AU}
$\boldsymbol{\tau}$	Linear function such that $\boldsymbol{\tau}(\boldsymbol{\xi}) \in [0,1]^{n_\xi}$
Φ	State transition function
ϕ	Scalar cost function
$\phi_L(L)$	Auxiliary function equal to $1 + P_1 \sin L + P_2 \cos L$
$\boldsymbol{\psi}$	Vector of boundary conditions
Ω	Right ascension of the ascending node
ω	Argument of pericentre

Acronyms

LHSU	Latin Hypercube Sampling from a Uniform distribution
MACS	Multi Agent Collaborative Search - A many-objective optimisation algorithm
PDF	Probability density function
QoI	Quantity of Interest

Acknowledgements

During my PhD I counted with the support of colleagues, friends, and family, without whom it would have been much harder, if not impossible, to have completed it. Firstly, I would like to thank my supervisor, Massimiliano Vasile, for his guidance and continuous feedback. I acknowledge also the support of the National Centre for Space Studies (CNES), namely through the grant R-S17/BS-0005-034, and Richard Epenoy's assistance.

Marilena Di Carlo and Lorenzo Ricciardi were very helpful, especially at the start of my PhD, by patiently explaining to me concepts that were useful for my work, and helping me with programming and technical issues.

I must also thank my friends, Aurelio Kaluthantridge, Catarina Garbacz, Duarte Santos, Fábio Morgado, Gianluca Philippi, Luis Sanchez, Maria Anna Laino, Michael McGurk, Paul Darm, Pedro Oliveira, Zita Aranka Tóth, among others, who have helped me preserve my sanity and generally function as a human being through my PhD, by being present in my life, and helping me throughout. I thank also my cousins, in particular Marta Oliveira, for being someone I could trust, and with whom I could vent about my problems. Finally, I thank my parents, Vítor Marto and Isabel Graça, who always insisted that I do what's right for me, and for that gave me their persistent support and advice.

Chapter 1

Introduction

In trajectory optimisation problems, generally the goal is to find an admissible control law that allows achieving a given set of mission objectives while minimising a cost function. If some system parameters, such as engine thrust or efficiency, are not known exactly, the trajectory that results from a specific control law cannot be known exactly either. To obtain trajectories that are robust to the uncertainty in system parameters, one has to quantify the resulting uncertainty in the achievement of mission objectives and satisfaction of mission constraints which results from that uncertainty.

The majority of existing methods for optimisation under uncertainty, and for uncertainty quantification in general, use standard probability theory which can only describe aleatory uncertainty. Aleatory uncertainty refers to systems that are inherently random, and thus this uncertainty cannot be reduced. In this PhD, the aim was to consider instead epistemic uncertainty, which results from a lack of information, and is such that if additional information was available, it could be reduced or eliminated. This characterises, for example, the uncertainty one has on the system parameters at the early phase of the design of a space mission [67]. It is argued in the literature that standard probability theory can be inadequate for handling epistemic uncertainty [33, 50]. For example, in [33], the authors point out that the typical way of representing a completely unknown variable as a uniform distribution, which follows from Laplace's principle of indifference, cannot distinguish scenarios where some parameter is completely unknown and those where it is known to follow a uniform distribution.

Using a classic example from introductory classes on probability and statistics, standard probability theory does not allow distinguishing the scenario of taking a ball from an urn containing the same number of black and white balls, versus the scenario where the number of each type of balls is unknown. In the first scenario there is aleatory uncertainty, while in the second there is epistemic uncertainty. In addition, a uniform distribution is not invariant under reparameterisation, so, for example, a lack of knowledge of the side length of a cube should lead to a lack of knowledge of the volume of the cube, but if the former is represented with a uniform distribution, the latter will have a non-uniform distribution, which wrongly suggests some degree of information about the volume [33]. In [50], the authors also argue that for a case with many epistemically uncertain variables that are only known to lie in certain intervals, characterising each of them with uniform distributions will lead to very low probabilities given for combinations of parameters near the ends of each interval, which is not a conclusion that can be drawn given the lack of knowledge on the probability distribution of the parameters. For a rigorous and complete explanation of the rationale behind the need for specific methods to handle epistemic uncertainty, the interested reader can find more information e.g. in [33, 50, 71].

This thesis considers the problem of trajectory optimisation subject to epistemic uncertainty in the control action and initial conditions. In particular, the uncertainty is considered to be time dependent, and an optimal control law is sought that is robust against multiple realisations of a set of uncertain system model parameters. This type of problem is applicable to the preliminary design of small-size low-cost space missions, where the uncertainty on the performance of the spacecraft is expected to be large, especially in the early stage of the system definition.

To account for aleatory uncertainty affecting the objectives one wants to optimise, one approach is to optimise their expected value, which is estimated by modelling the probability distribution of these objectives, starting from the knowledge of the probability distributions of the uncertain parameters affecting the trajectory. One can then use general purpose optimisation methods with these new objective functions based on mathematical expectation. For quantifying epistemic uncertainty, in this work

the lower expectation is used in place of the expectation. The uncertain parameters, instead of having a known probability distribution, have a set of possible probability distributions, referred to as a family of distributions. The lower expectation is then the minimal value of the expected value of the objective that can result from any set of distributions in these families. The detailed definition and computation of the lower expectation, as well as its use in many-objective trajectory optimisation problems, is the subject of Chapter 3.

In addition, a key novelty of this work was a dynamic programming based approach for the computation of epistemic uncertainty in the context of multi-segment trajectory optimisation problems where at each segment system parameters are affected by epistemic uncertainty and nearly exact observations of the system state are available. Rather than computing the lower expectation by minimising over the joint uncertainty space of all segments, this new dynamic programming approach allows separating the computation of the lower expectation at each segment, thus reducing the computational complexity of the overall uncertainty quantification. Since this is a multi-segment approach, it is directly applicable to feedback control problems, where the control law is a function of sparse observations. Note that in the general case, observations are also affected by a mix of epistemic and aleatory uncertainty, but this work focuses on epistemically uncertain systems and nearly exact observations. Other key contributions developed during this PhD were theoretical proofs for lower and upper bounds on the estimate of the lower expectation obtained by this dynamic programming based method, as well as a study of its computational efficiency and accuracy.

In addition to quantifying and optimising with respect to epistemic uncertainty, other contributions of this work relate to optimisation under many objectives. For the design of the trajectory of a space mission, a number of objectives are relevant, such as minimising propellant mass spent, or the distance to possibly several targets. It may be that there is no immediate way of combining these into a single objective, especially in the early segments of the design process, and these objectives may also be in conflict with each other. In this work, problems with more than 3 objectives are studied, in which case they are referred to as many-objective problems. Such problems are tackled

by estimating the set of non-dominated solutions, which are those for which no other solution exists that is better or as good in every objective. Such solutions are also termed Pareto-optimal, and their set is known as the Pareto set for a problem. [35]

The minimisation of the lower expectation is a hard global optimisation problem, in the general case, while the search for an approximation to the Pareto set in high dimensions requires a high number of calls to the objective functions. In order to make the many-objective optimisation tractable, this work elaborates on the idea, proposed in [22], of using a surrogate model that directly maps the decision variables into the value of the lower expectation. This mapping completely bypasses the computation of the lower expectation during the search for the Pareto optimal set. However, building this mapping, and validating the approximation to the Pareto set, requires evaluating the lower expectation a number of times, which is why, as previously mentioned, a significant fraction of the research effort in this PhD was dedicated to the efficient computation of this quantity. Finally, because the accuracy of the surrogate representation depends on the number of decision variables, this work also proposes dimensionality reduction techniques that map a subset of the space of admissible controls into a lower-dimensional space of proxy decision variables.

The approaches herein developed are applied to test cases consisting of missions to asteroids in the solar system, for spacecraft equipped with low-thrust engines, whose system parameters are known with epistemic uncertainty, and for which multiple objectives are optimised. Namely, these test cases are a rendezvous mission to asteroid Apophis, and an asteroid tour mission that visits four asteroids.

1.1 Aim, Objectives, and Research Questions

The aim of this thesis is to develop methods for the optimisation of spacecraft trajectories under epistemic uncertainty. Considering the challenges of undertaking this task, which often revolve around the quantification of epistemic uncertainty, the main research objectives can be summarised as:

- Optimisation of multiple objectives while minimising the number of times the

objective function is computed, which, since it involves quantifying epistemic uncertainty, tends to be computationally expensive;

- Efficient methods for the quantification of epistemic uncertainty, ideally limiting the growth of the computational cost with the number of uncertain variables, since some methods can exhibit exponential growth such as those based on evidence theory [13];

- An efficient way to quantify epistemic uncertainty that is compatible with corrections based on observations, i.e. feedback control, while also meeting the previous objective. A related objective is to develop a methodology that permits separating the computation into different segments of a trajectory, for trajectories where each segment is affected by different epistemic variables;

- Validation of previous approaches by testing them on realistic trajectory optimisation problems, and quantifying their accuracy and computational cost.

These research objectives can also be summarised into the following research questions, corresponding to the main areas where progress was made in this thesis:

- Can existing quantifications of epistemic which currently require computational complexity that is exponential with the number of uncertain variables instead be estimated in polynomial time? In this thesis, more efficient methods were developed for the quantification of epistemic uncertainty which can be applied to a very general class of problems. These were empirically shown to run in polynomial time. The dynamic programming algorithm developed to answer the following research question also addressed this one.

- Can the principles of dynamic programming be applied to epistemic uncertainty quantification, and can this be done in a way that allows accounting for feedback control in a computationally efficient manner? This thesis develops a novel dynamic programming based quantification of the epistemic uncertainty, including theoretical and experimental results. This approach achieves the desired effect of allowing for feedback control, while limiting the growth in computational cost

to polynomial with the number of segments a trajectory is divided into. The aforementioned theoretical results also give lower and upper bounds to this novel quantification of the epistemic uncertainty defined in terms of more conventional definitions.

1.2 Journal Publications

- **S. da G. Marto** and M. Vasile, 'Many-objective robust trajectory optimisation under epistemic uncertainty and imprecision', Acta Astronautica, vol. 191, pp. 99–124, 2022, doi: https://doi.org/10.1016/j.actaastro.2021.10.022.
- M. Di Carlo, **S. da Graça Marto**, and M. Vasile, 'Extended analytical formulae for the perturbed Keplerian motion under low-thrust acceleration and orbital perturbations', Celest Mech Dyn Astr, vol. 133, no. 3, p. 13, Mar. 2021, doi: 10.1007/s10569-021-10007-x.
- **S. da G. Marto** and M. Vasile, 'A Dynamic Programming Approach to Many-Objective Optimal Control Under Epistemic Uncertainty', Journal of Guidance, Control, and Dynamics (submitted)

1.3 Conference Papers and Presentations

- **S. G. Marto**, M. Vasile, and R. Epenoy, 'Multi-Objective Robust Trajectory Optimisation Under Epistemic Uncertainty and Imprecision', 70th International Astronautical Congress, p. 13, 2019.
- **S. da Graça Marto** and M. Vasile, 'Multi-Objective Robust Trajectory Optimization of Multi-Asteroid Fly-By Under Epistemic Uncertainty', in Advances in Uncertainty Quantification and Optimization Under Uncertainty with Aerospace Applications, M. Vasile and D. Quagliarella, Eds., Cham: Springer International Publishing, 2021, pp. 209–230.
- **S. Marto** and M. Vasile, 'Constrained multi-objective space trajectory optimisation under severe system and operational uncertainty: 72nd International

Astronautical Congress', Oct. 2021.

1.4 Structure of the Thesis

After this introduction, the thesis is separated into two parts. Part I describes the novel methods developed, starting with Chapter 2, where, after the general formulation of the many-objective trajectory optimisation problems tackled in this work, methods for their optimisation are described. Chapter 3 then introduces epistemic uncertainty, how it is quantified in theory and in practice, and how this quantification can be integrated into the optimisation problems to obtain trajectories that are robust to epistemic uncertainty. While the preceding two chapters already introduce novel concepts and methods, Chapter 4 introduces an approach which, to my knowledge, is unique, by applying the concept of dynamic programming to the computation of epistemic uncertainty, with the benefits of splitting the computation into segments of the trajectory, and of allowing handling of feedback control.

Part II then tests these methods by applying them to trajectory optimisation problems of spacecraft equipped with low thrust engines, for a rendezvous mission with Apophis, and a fly-by tour visiting four asteroids. These tests are split into Chapter 5, which focuses on applying and testing the methods of Chapters 2 and 3, and Chapter 6 which focuses on applying and testing the methods of Chapter 4.

Part I

Methodology

Chapter 2

Many-Objective Trajectory Optimisation

This chapter describes in general terms the type of problem that is the focus of this thesis, which are many-objective (MO) robust trajectory optimisation problems, as well as the methodology used to efficiently find solutions. Robust here means that the mission objectives should be met to some degree even in the presence of uncertainty in the dynamics. This means the objectives will include a quantification of the uncertainty, which is to be explained in more detail in the subsequent chapters.

2.1 Related Work

The literature on trajectory optimisation under uncertainty presents a number of techniques and approaches. A traditional approach, at preliminary mission design stage, is to use resource margins, which are excess amounts of some resource beyond the requirement to complete the mission under nominal (deterministic) conditions. Examples of this approach, applied to mission design, include propellant margins and missed-thrust margins, the latter being the amount of time the thrusters can fail without compromising the mission [40, 51, 55]. The addition of such margins is often mandatory as per ESA guidelines, see for example [23]. These approaches, however, may lead to non-robust trajectories with over-conservative margins [28]. In [70], the authors propose a

method to optimise backup trajectories, to be followed by the spacecraft in the event of permanent engine failure. If some of the engines fail, the spacecraft takes on an alternative trajectory focusing on different targets than had been planned for the nominal mission. Other methods for dealing with this type of uncertainty model the outcomes of the uncertain system in some way, and use this model to carry out the optimisation. Examples are the works by Bernelli-Zazzera et al. [6] using Taylor Algebra and Greco et al. [29] using a set-oriented approach based on generalised polynomial algebra (GPA). In both cases, polynomial expansions are used to relate the final state with a set of uncertain variables. In [53], Ozaki et al. use the unscented transform [36] to develop a method based on differential dynamic programming, optimising a feedback control law. Other approaches are designed to correct the trajectory once new observations of the state of the system are available. In model predictive control (MPC), at each segment of the trajectory, an optimal control law is found using an open-loop control based on the current state, and a known dynamical model of the system, which predicts its evolution over time [10]. An analogous approach is used in this work to simulate trajectory corrections based on observations of the state vector. In MPC, generally the cost function is the weighted sum of the distance to a reference trajectory and a measure of the control effort. As the trajectory unfolds, the state is measured, and the control law is updated. Since the control law is updated according to measurements of the state, it is responsive to deviations from the reference trajectory, making it a type of feedback control. An example application of robust MPC can be found in [58], wherein a linear time-invariant discrete-time system subject to an unknown but bounded disturbance is optimised, using a constraint tightening approach.

All of the works hitherto mentioned have in common that they deal only with aleatory uncertainty. Aleatory uncertainty refers to systems that are inherently random, and thus this uncertainty cannot be reduced. Epistemic uncertainty is uncertainty resulting from a lack of information, and is such that if additional information was available, it could be reduced or eliminated. Chapter 3 will go into the details of epistemic uncertainty, how it is defined, and how one can optimise under epistemic uncertainty, while in this chapter, these details are temporarily abstracted away.

2.2 Problem Formulation

In the most general form, the MO problems studied in this work can be written as

$$\begin{aligned} &\max_{\mathbf{u}\in U}[\underline{\mathrm{E}}(\phi_1), \underline{\mathrm{E}}(\phi_2), ..., \underline{\mathrm{E}}(\phi_m)]^T \\ s.t.& \\ &\dot{\mathbf{x}} = \mathbf{b}(\mathbf{x}) + \mathbf{f}(\mathbf{x}, \mathbf{u}, \boldsymbol{\xi}(t)) \\ &\underline{\mathrm{E}}(\mathbf{g}(\mathbf{x}, \mathbf{u}, \boldsymbol{\xi}, t)) \geq 1 - \boldsymbol{\epsilon}_g \\ &\underline{\mathrm{E}}(\boldsymbol{\psi}(\mathbf{x}_i, \mathbf{x}_f, \boldsymbol{\xi}_0, \boldsymbol{\xi}_f, t_i, t_f)) \geq 1 - \boldsymbol{\epsilon}_\psi, \end{aligned} \tag{2.1}$$

where $\boldsymbol{\phi} = [\phi_1, \phi_2, .., \phi_m]^T$ is a vector of scalar cost functions, $\boldsymbol{\psi}$ is a vector of boundary conditions and $\mathbf{g}$ is a vector of the path constraints. The quantities $\boldsymbol{\xi}_0, \boldsymbol{\xi}_f \in \Xi$ are vectors of uncertain parameters, $\boldsymbol{\xi}(t)$ is a multi-variate stochastic process, $\mathbf{u}$ is the control law, U is the space of control laws, and $\boldsymbol{\epsilon_g}$ and $\boldsymbol{\epsilon_\psi}$ are two vectors of constants.

The function $\underline{\mathrm{E}}$ is the lower expectation, and it is a generic way to quantify epistemic uncertainty [71]. A more detailed explanation and examples of what type of uncertainty quantification methods can be used, and how they are used, is the topic of Chapter 3. For now, it suffices to say that $\underline{\mathrm{E}}$ is a conservative estimate of the expected value of a quantity of interest, or of our degree of belief that it will be in some desired set, accounting for epistemic uncertainty. It is the use of $\underline{\mathrm{E}}$ in the objective function that makes the solutions to Eq. (2.1) robust to epistemic uncertainty.

The following chapters concern the practical aspects of obtaining an estimate of the Pareto front of Eq. (2.1). First, the dynamics model used throughout this work, without loss of generality, is described in Section 2.3. Section 2.4 discusses how the stochastic process $\boldsymbol{\xi}(t)$ is handled. Section 2.5 describes the transcription of the problem, i.e., the conversion of the problem one wishes to solve into one that can be handled by numerical methods. Section 2.6 provides a brief introduction to the solution of many-objective optimisation problems, and the algorithm used. The solution of many-objective optimisation problems can be computationally expensive, as it tends to require large numbers of evaluations of the objectives which, because they integrate epistemic uncertainty

quantification, are themselves expensive to compute, as will be discussed in the next chapters. Section 2.7 describes the use of control mapping to reduce the dimensionality of the search space, and Section 2.8 describes the use of surrogate models to further alleviate computational costs.

2.3 Orbital dynamics

This work will focus on heliocentric trajectories to visit asteroids, even though the overall methodology, in particular that for uncertainty quantification, is more generally applicable. The spacecraft will be subject to gravitational and thruster accelerations $\mathbf{a}$, given as

$$\ddot{\mathbf{r}} = -\frac{\mu}{r^3}\mathbf{r} + \mathbf{a} \ . \tag{2.2}$$

Because the focus of this work is not on the astrodynamics, but rather in obtaining optimal robust trajectories, to save time developing and running the various algorithms, perturbations were not considered.

The state $\mathbf{x}$ is modelled using equinoctial elements in a heliocentric frame. This element set has the benefit of being almost singularity free (the only singularities are $i = \pi$ and the rectilinear orbit) [4]. Therefore, $\mathbf{x} = [a, P_1, P_2, Q_1, Q_2, L, m]^T$, which can be written as a function of the classical orbit elements as

$$\begin{aligned} a &= a \\ P_1 &= e \sin(\omega + \Omega) \\ P_2 &= e \cos(\omega + \Omega) \\ Q_1 &= \tan\frac{i}{2}\sin\Omega \\ Q_2 &= \tan\frac{i}{2}\cos\Omega \\ L &= \Omega + \omega + \theta \ , \end{aligned} \tag{2.3}$$

in addition to the mass m, which decreases as propellant is spent.

The control law $\mathbf{u}(t)$ is a normalised thrust, such that $\|\mathbf{u}\| \leq 1$. The search space U is therefore the set of functions $\mathbf{u} = \mathbf{u}(t) : \mathbb{R} \to \mathbb{R}^3$ such that $\|\mathbf{u}(t)\| \leq 1 \ \forall t$. The

thrust is modelled as being inversely proportional to the squared heliocentric distance, following [19]. Thus, the acceleration $\mathbf{a}$ is written as:

$$\mathbf{a} = \frac{T(\boldsymbol{\xi}, r)}{m}\mathbf{u} = \frac{T_{\mathrm{AU}}(\boldsymbol{\xi}) r_{\mathrm{AU}}^2}{m r^2}\mathbf{u} \ , \tag{2.4}$$

where $T(\boldsymbol{\xi}, r)$ is the magnitude of the thrust at distance r, and $T_{\mathrm{AU}}(\boldsymbol{\xi})$ is the magnitude of the thrust at distance $r = r_{AU} := 1\ AU$, AU being the Astronomical Unit (around 1.5×10^{11} m) and $r = aB^2/\phi_L(L)$ is the heliocentric distance of the satellite.

This approximately models a scenario where the low thrust engine is being powered by solar panels, and the power that is made available to the engine is proportional to the amount of sunlight that hits the panels, as opposed to assuming constant thrust. The fact that the engines are solar powered does not necessarily imply that the thrust will follow an inverse square law, but this assumption also leads to simpler formulae when integrating the Gauss variation of parameter equations (given by the equation for $\dot{\mathbf{x}}$ in Eq. (2.1) using $\mathbf{b}$ and $\mathbf{f}$ in Eqs. (2.7) and (2.8)) using a first order method of perturbations [19], when compared to equivalent approximations assuming constant thrust [73].[1] The variation of the specific impulse I_{sp} that might accompany this variation in engine power was not modelled because it could not be as easily treated with this approach.[2] Consulting [26], one can infer Eqs. (2.5) and (2.6), where I_b and V_b are the beam current and beam voltage, respectively. These imply that one can keep the I_{sp} approximately constant, and vary the input power, approximately $P = I_b V_b$, by varying only I_B, which results only in a variation on the thrust T.

$$T \propto I_b \sqrt{V_b} \tag{2.5}$$

[1] The formulae for low thrust following an inverse square law are Eqs (54)-(56) of [19]. Compare to their constant thrust equivalents in Eqs (17)-(20) of [73] which use many quantities that had to be defined in an appendix. This difference in complexity also translates to a difference in computational cost.

[2] For a brief overview of this method of perturbations, in addition to the cited papers, Appendix C contains an introduction to the method of perturbations in addition to my contribution in developing some formulae for perturbations (third body) that were not included in the main thesis work, but which were published in [19]. Note that my contributions did not include the low thrust acceleration following the inverse square law that is used in this thesis, so this is not included in Appendix C, but it is also published in [19].

$$I_{\text{sp}} \propto \sqrt{V_b} \tag{2.6}$$

For the full equations the reader should consult [26] as these details are outside the scope of this work.

In Section 5.1.1, the test cases for this work are defined, which includes selecting values for I_{sp} and T_{AU}. This selection is justified by consultation of a data sheet for the NEXT ion engine and, alongside this justification, the reader can also find a justification for modelling the I_{sp} as constant based on the same data sheet.

The dynamics in Eq. (2.2) are defined by writing the vector $\mathbf{b}$ from Eq. (2.1) as

$$\mathbf{b} = \left[0, 0, 0, 0, 0, n\frac{\phi_L^2(L)}{B^3}, 0\right]^T \ , \tag{2.7}$$

and the function

$$\mathbf{f}(\mathbf{x}, \mathbf{u}, \boldsymbol{\xi}) = \begin{bmatrix} \frac{2}{Bnm}\left[\left(P_2 \sin L - P_1 \cos L\right) a_R + \phi_L(L) a_T\right] \\ \frac{B}{nam}\left[-a_R \cos L + \left(\frac{P_1 + \sin L}{\phi_L(L)} + \sin L\right) a_T - P_2 \frac{Q_1 \cos L - Q_2 \sin L}{\phi_L(L)} a_N\right] \\ \frac{B}{nam}\left[a_R \sin L + \left(\frac{P_2 + \cos L}{\phi_L(L)} + \cos L\right) a_T + P_1 \frac{Q_1 \cos L - Q_2 \sin L}{\phi_L(L)} a_N\right] \\ \frac{B}{2nam}\left(1 + Q_1^2 + Q_2^2\right) \frac{\sin L}{\phi_L(L)} a_N \\ \frac{B}{2nam}\left(1 + Q_1^2 + Q_2^2\right) \frac{\cos L}{\phi_L(L)} a_N \\ -\frac{B}{nam}\left(Q_1 \cos L - Q_2 \sin L\right) a_N \\ -\|\mathbf{u}\| \frac{T_{\text{AU}}(\boldsymbol{\xi}) r_{AU}^2}{I_{sp}(\boldsymbol{\xi}) g_0 r^2} \end{bmatrix} , \tag{2.8}$$

where a_R, a_T, and a_N are the components of the acceleration $\mathbf{a}$ in the radial-transverse-normal (RTN) reference frame [65], and n is the mean motion,

$$B = \sqrt{1 - e^2} = \sqrt{1 - P_1^2 - P_2^2} \ , \tag{2.9}$$

$$\phi_L(L) = 1 + P_1 \sin L + P_2 \cos L \ , \tag{2.10}$$

and I_{sp} is the specific impulse. See [4,38] for explanations of these formulae above, which correspond to the Gauss variation of parameters equations for equinoctial elements. In addition to the thrust acceleration, this dynamics model could have also included, in $\mathbf{b}(\mathbf{x})$, the effects of perturbations. Since accurate modelling of orbital dynamics was not the focus of this work, in the interest of simplification and saving time, perturbations were not included. However, this PhD work did include the integration of the dynamics equations of a satellite subject to third body perturbations, as part of a journal paper that deals with this and other perturbations, published in the Journal of Celestial Mechanics [19]. For completion, this work is included in Appendix C.

As mentioned, the formulae derived in [19] are used to propagate the orbital dynamics described in this section. These formulae are derived specifically for low thrust, which is characterised by the ratio between $\mathbf{a}$ and the gravitational acceleration being much lower than 1. Note that because of the assumption that this value is inversely proportional to the squared heliocentric distance, this ratio will stay constant, as the gravitational acceleration follows the same rule.

2.4 Stochastic processes

The theory of stochastic processes and stochastic differential equations (SDE) has received extensive study, and for this reason, it is worthwhile to include some discussion on how this work relates to this theory and some of its theoretical results. This theory is not the focus of this work, which instead is focused on handling uncertainty in systems with finite numbers of unknown variables. This section is meant to illustrate how such an approach can nonetheless be applied to handle SDEs, starting by writing an SDE that is similar to the problem under study.

The variables T_{AU} and I_{sp} in Eqs. (2.4) and (2.8) are defined as stochastic processes, i.e. functions of time for which each evaluation is a random variable. Stochastic processes can be written as white noise driven processes,

$$\dot{\mathbf{x}} = \mathbf{b}(\mathbf{x}, t) + L(\mathbf{x}, t)\mathbf{w}(t) \tag{2.11}$$

where $\mathbf{w}(t)$ is the white noise process, a process where each $w(t_1)$ is a Gaussian random variable independent of $w(t_2)$ for any $t_1 \neq t_2$, $\mathbf{b}$ is known as the drift term and L as the dispersion matrix. Other driving stochastic processes can also be used, such as Levy processes and Poisson processes, although this appears to be less common. [59]

The formulation in Eq. (2.1) would not fit into Eq. (2.11) even if $\boldsymbol{\xi}$ were to be replaced by a white noise process, because $\mathbf{f}$ depends on $\boldsymbol{\xi}$ in a possibly non-linear manner. In that case, as said by Särkkä and Solin in [59], "... white noise should not be thought of as an entity as such, but it only exists as the formal derivative of Brownian motion. Therefore, only linear functions of white noise have a meaning, whereas nonlinear functions do not". It is also for this reason that many works in the literature choose to write $\mathrm{d}\boldsymbol{\beta}(t)$ instead of $\mathbf{w}(t)$, where $\boldsymbol{\beta}(t)$ is Brownian motion, also known as the Wiener process. In that case, Eq. (2.11) is re-written as

$$\mathrm{d}\mathbf{x} = \mathbf{b}(\mathbf{x}, t)\,\mathrm{d}t + L(\mathbf{x}, t)\,\mathrm{d}\boldsymbol{\beta}(t)\ . \tag{2.12}$$

The dynamic equation in Eq. (2.1) can also be written as

$$\dot{\mathbf{x}} = \mathbf{b}(\mathbf{x}) + \mathbf{d}(\mathbf{x}, \mathbf{u})\mathbf{F}(\boldsymbol{\xi})\ , \tag{2.13}$$

which is closer to Eq. (2.11), but $\mathbf{F}(\boldsymbol{\xi})$ is the vector

$$\begin{bmatrix} T_{\mathrm{AU}} \\ \dfrac{T_{\mathrm{AU}}}{I_{sp}} \end{bmatrix}\ , \tag{2.14}$$

which is not a standard stochastic process like white noise, since F is a nonlinear function of the stochastic processes involved, and so Eq. (2.13) is not an SDE.

The issue is that I_{sp} is a stochastic process, but it affects the dynamics non-linearly, which is incompatible with an SDE formulation. The same is not the case with T_{AU}. A solution is to consider I_{sp} as an additional state with dispersion matrix L_I, and write

the thrust as $T_{\mathrm{AU}} = \overline{T}_{\mathrm{AU}} + L_T \, \mathrm{d}\boldsymbol{\beta}$, resulting in an augmented version of the dynamics,

$$\mathrm{d}\mathbf{x}^+ = \mathbf{b}^+(\mathbf{x}^+, \mathbf{u}) \, \mathrm{d}t + \mathbf{d}^+(\mathbf{x}^+, \mathbf{u}) \, \mathrm{d}\boldsymbol{\beta} \tag{2.15}$$

where

$$\mathbf{x}^+ = \begin{bmatrix} \mathbf{x} \\ I_{\mathrm{sp}} \end{bmatrix} \ , \ \mathbf{b}^+(\mathbf{x}^+, \mathbf{u}) = \begin{bmatrix} \mathbf{b}(\mathbf{x}) + \tilde{\mathbf{f}}(\mathbf{x}^+, \mathbf{u}) \overline{T}_{\mathrm{AU}} \\ 0 \end{bmatrix} \ , \ \mathbf{d}^+(\mathbf{x}^+, \mathbf{u}) = \begin{bmatrix} \tilde{\mathbf{f}}(\mathbf{x}, \mathbf{u}) L_T & \mathbf{0} \\ \mathbf{0} & L_I \end{bmatrix} . \tag{2.16}$$

Now that an SDE formulation has been obtained that is compatible with the problem formulation in the previous section, let us investigate how the approaches in this work, which always consider a finite number of uncertain variables, may be adapted even to this approach.

One method to approximate a stochastic process is to use a pathwise approximation [44, 59]. Such an approximation consists in writing a stochastic process $\mathbf{X}(t)$ as

$$\mathbf{X}(t) = \sum_{i=1}^{\infty} \mathbf{z}_i \phi_i(t) \ , \tag{2.17}$$

where $\mathbf{z}_i$ are zero mean random variables and $\phi_i(t)$ are basis functions. The Karhunen–Loève (KL) theorem provides the best choice of functions $\phi_i(t)$ for convergence for a stochastic process, i.e. its KL expansion. This expansion also makes the variables $\mathbf{z}_i$ uncorrelated. For example, for Brownian motion $\beta(t)$ with diffusion matrix Q, the KL expansion for $t \in [0, T]$ is

$$\mathrm{d}\boldsymbol{\beta}(t) = \sqrt{Q} \sum_{n=1}^{\infty} z_i \phi_i(t) \ , \tag{2.18}$$

where $\mathbf{z}_i$ are independent standard Gaussian random variables, and [59]

$$\phi_i(t) = \left(\frac{2}{T} \right)^{1/2} \cos \left(\frac{(2i-1)\pi}{2T} t \right) \ . \tag{2.19}$$

Other examples of pathwise expansions exist, such as Haar wavelets or the Wiener chaos expansion.

In Eq. (2.15), if L_I is a constant matrix, I_{sp} can be solved independently of other

variables, resulting in

$$I_{\mathrm{sp}}(L,\ \boldsymbol{\xi}) = \sum_{i}^{N_I} \eta_i(L)\xi_{I,i}\ , \tag{2.20}$$

and $\xi_{T,n}$ are the new uncertain variables. Similarly, if L_T is a constant matrix, the thrust $T_{\mathrm{AU}} = \overline{T}_{\mathrm{AU}} + L_T\,\mathrm{d}\boldsymbol{\beta}$ can also be written

$$T_{\mathrm{AU}}(L,\ \boldsymbol{\xi}) = \sum_{i}^{N_T} \eta_i(L)\xi_{T,i}\ , \tag{2.21}$$

where $\xi_{T,i}$ and $\xi_{I,i}$ are parameters of the epistemically uncertain stochastic processes that define the value of T and I_{sp} over time.

A reading of the work by Pierret [54] may suggest that this approach is flawed, as the second order terms on the thrust therein developed, and referred to as "supplementary terms", are absent in this formulation. They result from an application of Itô's formula (also known as Itô's lemma) [59], which can be used to obtain the SDE of a transformed process $\mathbf{Y} = \mathbf{g}(\mathbf{X})$,

$$\mathrm{d}Y_k = \frac{\partial g_k}{\partial t}(t,\mathbf{X})dt + \sum_i \frac{\partial g_k}{\partial x_i}(t,\mathbf{X})dX_i + \frac{1}{2}\sum_{i,j}\frac{\partial^2 g_k}{\partial x_i \partial x_j}(t,\mathbf{X})dX_i\,\mathrm{d}X_j\ . \tag{2.22}$$

The last term, with second order derivatives, is the source of those supplementary terms. The work in Pierret considers an Itô SDE, which is the most common type of SDE, and for which Itô's formula applies. A pathwise approximation, on the other hand, converges to the Stratonavitch integral of the SDE [59]. To indicate that this is the definition in use, Eq. (2.12) can be written with a $\circ$ between the dispersion matrix and the driving process, i.e., the following denotes a Stratonavitch SDE:

$$\mathrm{d}\mathbf{x} = \mathbf{b}(\mathbf{x},t)\,\mathrm{d}t + L(\mathbf{x},t) \circ \mathrm{d}\boldsymbol{\beta}\ . \tag{2.23}$$

Before explaining what this difference means, let us review the formalism behind SDEs.

The integration of an SDE such as the one in Eq. (2.12) requires a formalism that differs from the Riemann, Stieltjes or Lebesgue formalisms used in Calculus. The reasoning given in [59] is as follows. Stieltjes or Lebesgue integrals are defined essentially

as the limit

$$\int_{t_0}^{t} \mathbf{L}(\mathbf{x}(t), t)\mathrm{d}\boldsymbol{\beta} = \lim_{n\to\infty} \sum_{k} \mathbf{L}\left(\mathbf{x}\left(t_k^*\right), t_k^*\right)\left[\boldsymbol{\beta}\left(t_{k+1}\right) - \boldsymbol{\beta}\left(t_k\right)\right], \tag{2.24}$$

where $t_0 < t_1 < \ldots < t_n$ and $t_k^* \in [t_k, t_{k+1}]$. A similar definition exists for Riemann integrals. In both cases, the limit must be independent of the choice of t_k^* within its interval, which does not happen in general when $\mathrm{d}\boldsymbol{\beta}$ is a Wiener process. The most common way to handle this issue is to choose $t_k^* = t_k$, which results in an Itô integral. These have the advantage of resulting in Martingales, which are useful for theoretical analysis, but have the consequence that the chain rule does not apply, instead one must use Itô's formula. If instead the choice is the midpoint, $t_k^* = (t_k + t_{k+1})/2$, the limit results in a Stratonovich integral. These are not martingales, but instead of Itô's formula, the normal chain rule applies, resulting in

$$\mathrm{d}Y_k = \frac{\partial g_k}{\partial t}(t, \mathbf{X})dt + \sum_i \frac{\partial g_k}{\partial x_i}(t, \mathbf{X})dX_i \ , \tag{2.25}$$

which is Eq. (2.22) without the second order terms.

These facts do not limit the applicability of the proposed method, since an Itô SDE can be converted to a Stratonovich one by transforming the drift term [59],

$$\tilde{b}_i = b_i(\mathbf{x}, t) + \frac{1}{2}\sum_{j,l} \frac{\partial L_{ij}(\mathbf{x}, t)}{\partial x_l} L_{lj}(\mathbf{x}, t) \ . \tag{2.26}$$

For the rest of this document, the preceding discussion will be set aside, it being sufficient to remember that a representation with a finite number of uncertain variables can also represent a white noise driven stochastic process.

The stochastic processes T_{AU} and I_{sp} will be represented in a decomposition of the form of Eqs. (2.20) and (2.21). In this work, functions η_i, and uncertain variables $\xi_{T,i}$ and $\xi_{I,i}$, are not found by analysing an SDE, as that is outside the scope of this work. Instead, simpler methods for discretising the stochastic processes are used, as explained in the test cases defined in Chapter 5 of Part II.

As a final note, model (2.21) is not ideal to represent uncertainty in the thrust

magnitude due to engine outage events because it would require a very high order expansion. Thus, when studying the optimal response to engine outage, the following piece-wise constant model is introduced, where the true longitude L at the start and end of the outage are uncertain variables:

$$T_{\mathrm{AU}}(\boldsymbol{\xi}) = \begin{cases} \xi_{\delta,n}\hat{T}_{\mathrm{AU}} & if \;\; L \in [\xi_{L0,n}, \xi_{L0,n} + \xi_{\Delta L,n}] \\ \hat{T}_{\mathrm{AU}} & \text{otherwise} \end{cases} \tag{2.27}$$

where $\xi_{\delta,n} \in [0,1]$, $\xi_{\Delta L,n} > 0$, $\xi_{L0,n} \in [L_0, L_f]$, and $\hat{T}_{\mathrm{AU}}$ is the reference value of the thrust without outage. This is still a stochastic process, although not one that can be written as a solution of an SDE in the form of Eq. (2.12).

From now on, the vector $\boldsymbol{\xi}$ will denote an array containing every uncertain variable in the pathwise representation $\xi_{T,n}$, $\xi_{I,n}$, (if outages are being considered, this includes $\xi_{L0,n}$, $\xi_{\Delta L,n}$, and $\xi_{\delta,n}$), in addition to $\boldsymbol{\xi}_0$.

2.5 Transcription

The search space in Eq. (2.1) is a Hilbert space. To optimise the problem in that equation numerically, it is necessary to transcribe the problem, i.e., to devise a representation $\mathbf{y} \in Y \subseteq \mathbb{R}^k$ for finite integer k, such that there is a transformation $\mathbf{u} : Y \to U$, and perform the search over Y. Furthermore, instead of constraints, the optimisation objectives include lower expectations $\underline{\mathrm{E}}$ on indicator functions of the uncertain and control variables, which are one when these constraints are met, and zero otherwise, see Chapter 3. The result is a box constrained problem of the form:

$$\min_{\mathbf{y}\in Y, \boldsymbol{\nu}\in N} [-\underline{\mathrm{E}}(\boldsymbol{\phi}(\mathbf{y}), \boldsymbol{\nu}), -\underline{\mathrm{E}}(\mathbf{g}(\mathbf{y}), \boldsymbol{\nu}), -\underline{\mathrm{E}}(\boldsymbol{\Psi}(\mathbf{y}), \boldsymbol{\nu}), \boldsymbol{\nu}]^T \tag{2.28}$$

The quantity $\boldsymbol{\nu} \in N$ is a vector of parameters (called thresholds in the remainder of this thesis), that are used to scale the Lebesgue measure of the target sets. More detail on the meaning of the thresholds ν is given in Chapter 3.

Problem (2.28) recasts the boundary and path constraints in (2.1), into objective functions and adds a measure of the target sets as an additional vector of objectives.

This formulation is more amenable to a solution when little is known about the target sets. In fact, problem (2.1) might not have any admissible solution for a given choice of $\boldsymbol{\epsilon}_g$ and $\boldsymbol{\epsilon}_\Psi$. On the contrary, problem (2.28) always has a solution and provides a useful trade-off between constraint satisfaction and cost.

Two other variants of problem (2.28) will also be considered. One variant involves fixed thresholds, where the set N is reduced to a single point, and another variant involves fixing the lower expectations $\underline{\mathrm{E}}$, in which case the ν are referred to as upper quantiles $\overline{\nu}$ [71]. This is described in more detail in Chapter 3.

Transcribing problem (2.1) into problem (2.28) requires solving the differential constraints first and then calculating boundary and path constraints. In the following section, the transcription method is briefly presented. Note that although the methodology proposed in this work is conceived to work with direct transcription methods, it does not rely on the specific transcription approach defined in this Section. This means that in principle one could use any direct transcription approach without modification to the overall methodology for the treatment of epistemic uncertainty.

2.5.1 Transcription of the Differential Constraints

In this study, differential, path and boundary constraints are transcribed using a variant of the direct analytical multi-shooting algorithm proposed in [74] and implemented in the software code FABLE (FAst Boundary-value Low-thrust Estimator)[3]. The transfer leg is split into a predefined sequence of n_{LT} finite coast and thrust arcs. [20] Over each arc the differential constraints are solved analytically with a first order expansion in the small quantity $\|\mathbf{a}\|$, which corresponds to the first order method of perturbations solution mentioned previously and discussed in detail in [19]. Note that because of the first order nature of the solution, any second order term in $\mathbf{a}$ is neglected. Thus, the second order "supplementary terms" developed by Pierret [54] and discussed in the previous section would not be included by this approximation. The motion is assumed purely Keplerian along coast arcs.

An additional detail is that for ease of analytical integration, the dynamics are

[3] https://github.com/strath-ace/smart-o2c

integrated over the true longitude L instead of time t, which becomes a state. The time and true longitude swap places in this analysis [74]. Therefore, the length of these arcs, and the "time" they take place in are defined in terms of true longitude.

The i-th coast arc is characterised by its length in longitude ΔL_i^{OFF}, i.e., the difference between the true longitude at the end of the arc and at the beginning, while the i-th thrust arc is characterised by its length ΔL_i^{ON}, and by the azimuth α_i and declination β_i that the spacecraft engine is pointing towards, and by the throttle value τ_i. The division of the trajectory into arcs is illustrated in Fig. 2.1, while the azimuth and declination angles are illustrated in Fig 2.2, and Eq. (2.29) shows how to obtain the acceleration in the RTN frame given those angles.

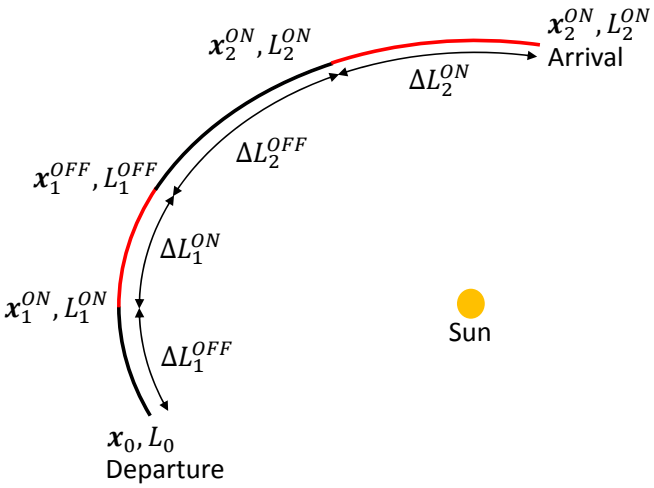


Figure 2.1: Segmentation of the trajectory into coast arcs (black) and thrust arcs (red). "On" nodes define the switching point from a coast to a thrust arc and "Off" nodes define the switching point from a thrust to a coast arc.

$$\mathbf{a} = \frac{T(\boldsymbol{\xi}, r)}{m} \begin{bmatrix} \cos\beta\cos\alpha \\ \cos\beta\sin\alpha \\ \sin\beta \end{bmatrix} \tau_i \tag{2.29}$$

The low thrust trajectories considered in this work consist of an injection into a heliocentric transfer orbit by a conventional launcher, followed by a number of alter-

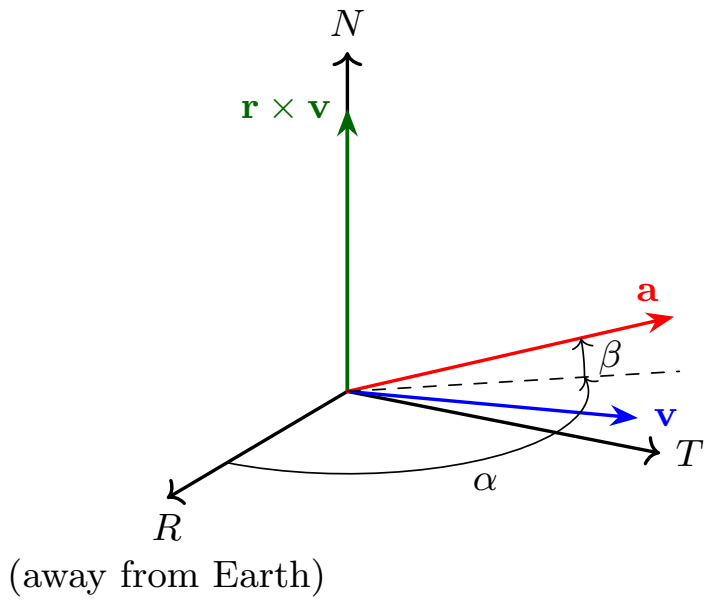


Figure 2.2: The acceleration vector **a** and the α and β angles, in the RTN frame.

nating coast and thrust arcs with ion propulsion. The injection is characterised by the departure time t_D, and the magnitude v_∞, azimuth γ and declination δ of the hyperbolic excess velocity relative to the Earth in a heliocentric reference frame. These four parameters define the initial conditions.

The variables described above form the solution vector:

$$\mathbf{y} = \left[\mathbf{\Delta L^{OFF}}, \mathbf{\Delta L^{ON}}, \boldsymbol{\alpha}, \boldsymbol{\beta}, \boldsymbol{\tau}, \gamma, \delta, t_D\right]. \tag{2.30}$$

Variables $\mathbf{\Delta L^{OFF}}$, $\mathbf{\Delta L^{ON}}$, $\boldsymbol{\alpha}$, $\boldsymbol{\beta}$, $\boldsymbol{\tau}$ are each a vector with n_{LT} elements, each element corresponding to each arc. When executing the algorithms described in this work on a computer, the control laws $\mathbf{u} \in U$ are represented internally with a vector $\mathbf{y} \in Y$ as above. When defining a function that accepts $\mathbf{u}$, it will also be written as accepting $\mathbf{y}$ where the mapping $Y \to U$ is implicit.

Variable v_∞ is considered uncertain, and since it refers to the initial conditions, it forms $\boldsymbol{\xi}_0$. Vector $\boldsymbol{\xi}$, as discussed previously, will thus include this variable in addition to the parameters of the discretization of the stochastic processes T and I_{sp}, namely

$\xi_{T,n}$, $\xi_{I,n}$ and, if engine outage is being modelled, $\xi_{L0,n}$, $\xi_{\Delta L,n}$, and $\xi_{\delta,n}$. The exact formulation of the uncertainty in these parameters will be described in more detail in following chapters.

2.5.2 Transcription of Cost Functions and Algebraic Constraints

Since Problem (2.1) was transcribed into Eq. (2.28), any integral path constraints or point constraints are turned into objectives. While the theoretical development can apply to generally any type of objectives and constraints, in our applications, in Part II, only the following objectives considered:

- propellant mass $\phi = \Delta m$: the total propellant mass consumed is formulated as a path cost;
- distance to i-th target $\Psi_{i,\Delta r} = \Delta \mathbf{r}_i$ is the distance to the i-th target at the time of rendez-vous or flyby, and is formulated as a point function cost
- the relative speed to the i-th target $\Psi_{i,\Delta r} = \Delta \mathbf{v}_i$, similar to the distance, is formulated as a point function cost

By point function cost it is meant a cost function dependent on the states at given times. [7] The only constraints that are not turned into objectives are the algebraic constraints. In our application, the only such constraint is $\|\mathbf{u}\| \leq 1$, which is equivalent to limiting the maximum thrust of the engine.

In the general case, algebraic or patch constraints can be applied to the elements of vector $\mathbf{y}$ on each segment. The constraint on the thrust magnitude is achieved by making $\tau_i \in [0, 1]$, which enforces $\|\mathbf{u}\| \leq 1$ without the need for additional constraints besides those implicit in the definition of the box-shaped search space Y. The thrust level is an uncertain quantity that is constrained to follow the models in Eqs. (2.21) or (2.27), while the specific impulse follows Eq. (2.20).

Path cost functions are computed by adding up the contribution of each segment. In particular, in the following we will be interested in the total propellant mass and transfer time. For our applications, the propellant mass is the sum of the propellant

mass over each segment, computed as the integral:

$$\Delta m_i = -\int_{L^{ON}}^{L^{OFF}} \frac{T_{\mathrm{AU}}(\boldsymbol{\xi}) r_{AU}^2}{I_{sp}(\boldsymbol{\xi}) g_0 r^2} \frac{dt}{dL} dL \ , \tag{2.31}$$

and as such $\phi = \Delta m = \sum_i \Delta m_i$.

The integration of the differential equations is carried over the true longitude L instead of the time. As such, to compute the point function constraints on $\Delta \mathbf{r}_i$ and $\Delta \mathbf{v}_i$, one needs to compute the time length of each segment:

$$\Delta t = \int_{\Delta L} \left(\sqrt{\frac{\mu}{a}} \frac{\phi_L^2(L)}{B^3} - \sqrt{\frac{a}{\mu}} \frac{B}{\phi_L(L)} a_N \left(Q_1 \cos L - Q_2 \sin L \right) \right) dL \ . \tag{2.32}$$

The positions and velocities of the targets are then computed using Keplerian dynamics at the times given by that equation.

As mentioned previously, constraints are turned into objectives by writing them as the lower expectation that these constraints are being met, as represented in Eq. (2.28). The transcription approach, including analytical expressions, accuracy and computational cost, has been published in [74], [73], [21] and [19].

2.6 Solving a Many-Objective Problem

Having cast the problem into a box constrained many-objective problem in Eq. (2.28), its solution is estimated using a memetic algorithm called Multi Agent Collaborative Search (or MACS for short) [56], which was used in this work without significant modifications.

To simplify the following discussion, the objectives are written as a single vector function $\mathbf{F}(\mathbf{p}) = [F_1(\mathbf{p}) \dots F_M(\mathbf{p})]$ of the design points $\mathbf{p} \in D$. The general many-objective optimisation (MOO) problem, of which Eq. (2.28) is an example, can then be written as

$$\min_{\mathbf{p} \in D} \mathbf{F}(\mathbf{p}) \tag{2.33}$$

In general, there is no single solution to such a problem. If two design points $\mathbf{p}^{(\mathbf{i})}$ and

$\mathbf{p^{(j)}}$, are such that for every objective F_k it holds that

$$F_k(\mathbf{d}^{(i)}) \leq F_k(\mathbf{d}^{(j)}) \ \forall \ k \ , \tag{2.34}$$

which is also written as $F_k(\mathbf{d}^{(i)}) \preceq F_k(\mathbf{d}^{(j)})$, it is said that point $\mathbf{d}^{(i)}$ *dominates* point $\mathbf{d}^{(j)}$. Clearly if one has found $\mathbf{d}^{(i)}$, one has no interest in point $\mathbf{d}^{(j)}$, since it is no better than $\mathbf{d}^{(i)}$ on every objective.

Solving Eq. (2.28) consists in finding the set of non-dominated solutions, also known as the Pareto front [8]:

$$\left\{\mathbf{F}(\mathbf{d}) : \ \forall \ \mathbf{d} \ , \ \nexists \ \hat{\mathbf{d}} \in D \ , \ \mathbf{F}(\hat{\mathbf{d}}) \preceq \mathbf{F}(\mathbf{d})\right\} \tag{2.35}$$

In practice this is achieved with a combination of techniques, the simplest and most often used being scalarisation. This consists in turning the problem above into a set of single-objective problems, optimising some combination $F_\lambda = F_\lambda(\mathbf{F}, \boldsymbol{\lambda}^{(i)})$ of the objectives, where $\boldsymbol{\lambda}^{(i)}$ is a search direction. Each one of these is then optimised separately using numerical single objective optimisation methods, for which there has been a large amount of research and software development. The solutions to these, after removing dominated points, form a sample approximation of the Pareto front.

The naive scalarisation method is to consider linear combinations of the objectives, i.e., $\boldsymbol{\lambda}^{(i)} \cdot \mathbf{F}$ where $\boldsymbol{\lambda}^{(i)}$ is a vector of real numbers the same dimension as $\mathbf{F}$. It can be shown [37], however, that this does not allow sampling the whole Pareto front, i.e., there may be points in the Pareto front that are not optimal with respect to $\boldsymbol{\lambda}^{(i)} \cdot \mathbf{F}$ for any $\boldsymbol{\lambda}$. Instead, alternatives like Chebyshev or Pascoletti Serafini are used.

MACS is a population based memetic many-objective optimisation algorithm that implements a combination of Chebyshev and Pascoletti-Serafini scalarisations with an energy-based management of the archive containing an approximation of the Pareto set, along with several other optimisation techniques. The MACS algorithm contains many more techniques than those mentioned here, such as combinations of local and social search actions, which include differential evolution techniques. For more details on MACS, its performance versus other optimisers and its use for the solution of many-

objective optimal control problems, the interested reader can refer to [56] and [57].

The archival technique, however, warrants some extra explanation, since it is used for other purposes in this work. If an MOO algorithm stored every single non-dominated point it finds during its search, since the Pareto front can be a continuous set, its storage requirements may grow unbounded. Therefore, there must be a limit to how many solutions are kept. The archival strategy employed by MACS consists in finding the subset of non-dominated solutions that minimise a cost function based on the static electric potential:

$$E = \sum_{i=1}^{N_{\text{arch}}} \sum_{j=i+1}^{N_{\text{arch}}} \frac{1}{\left(\mathbf{F}^{(i)} - \mathbf{F}^{(j)}\right)^T \left(\mathbf{F}^{(i)} - \mathbf{F}^{(j)}\right)} \tag{2.36}$$

where $\mathbf{F}^{(i)}$ is the value of the objectives for the i-th element of the archive and N_{arch} is the number of elements in the archive. Whenever, in the course of this work, a set of points needs to be downsampled while preserving variety, this algorithm can also be used, even outside of the original intended application of maintaining a fixed archive of solutions.

2.7 Control Maps

The vast majority of the control laws defined by the elements $\mathbf{y}$ of set Y result in trajectories that are very far from meeting the constraints to any reasonable level of satisfaction. If that is the search space for MACS, some of the function evaluations will be wasted on such trajectories. This work investigates the use of control maps to produce a reduced search space $\hat{Y}$, that is lower-dimensional and is aimed at containing a higher proportion of viable control laws.

A control map is a function $\mathbf{f}_{\text{CMap}} : Z \to \hat{Y}$ such that $\hat{Y} \subset Y$ and where the dimensionality of Z is much lower than Y. When such control map is in use, Problem (2.28) becomes a search over set Z, and the control map must be computed when

evaluating the objectives,

$$\min_{\mathbf{z}\in Z,\boldsymbol{\nu}\in N}[-\underline{\mathrm{E}}(\boldsymbol{\phi}(\mathbf{f}_{\mathrm{CMap}}(\mathbf{z})),\boldsymbol{\nu}),-\underline{\mathrm{E}}(\mathbf{g}(\mathbf{f}_{\mathrm{CMap}}(\mathbf{z})),\boldsymbol{\nu}),-\underline{\mathrm{E}}(\boldsymbol{\Psi}(\mathbf{f}_{\mathrm{CMap}}(\mathbf{z})),\boldsymbol{\nu}),\boldsymbol{\nu}]^T \ . \quad (2.37)$$

The following subsections describe the control maps that were developed in this work.

Deterministic Control Map

In this control map, the search space is restricted to control laws that are optimal for a deterministic setting, targeting positions and velocities in the vicinity of the targets. This is the control map proposed in [45]. Thus the control mapping $\mathbf{y} = \mathbf{f}^*_{\mathrm{CMap}}(\boldsymbol{\xi}, \mathbf{D}_r, \mathbf{D}_v)$ is defined as the vector $\mathbf{y}$ that solves the optimal control problem

$$\begin{aligned} \mathbf{f}^*_{\mathrm{CMap}}(\boldsymbol{\xi}, \mathbf{D}_r, \mathbf{D}_v) &= \underset{\mathbf{y}}{\operatorname{argmin}}\, m_p(\mathbf{y}, \boldsymbol{\xi}) \\ s.t. \quad \mathbf{r}(\mathbf{y}, \boldsymbol{\xi}, t_{T,k}) &= \mathbf{r}_{T,k}(t_{T,k}) + \mathbf{D}_r^{(k)} \\ \mathbf{v}(\mathbf{y}, \boldsymbol{\xi}, t_{T,k}) &= \mathbf{v}_{T,k}(t_{T,k}) + \mathbf{D}_v^{(k)} \text{ if target is not fly-by only.} \end{aligned} \quad (2.38)$$

where the k-th target position $\mathbf{r}_{T,k}$ and velocity $\mathbf{v}_{T,k}$ are calculated using Keplerian propagation for the target's fly-by time $t_{T,k}$, and the spacecraft position at flyby $\mathbf{r}(\mathbf{y}, \boldsymbol{\xi}, t_{T,k})$ and $\mathbf{v}(\mathbf{y}, \boldsymbol{\xi}, t_{T,k})$ are calculated for $t_{T,k}$, by propagating the trajectory as described in Section 2.3. While not explicit in Eq. (2.38), multiple-shooting is used to improve the convergence of the solver. FABLE [20, 74] is used to solve Eq. 2.38.

The state at departure $\mathbf{x}_0$ is defined by the departure conditions v_∞, γ, δ, the last two being part of $\mathbf{y}$ as mentioned. The vectors $\mathbf{D}_r^{(k)}$ and $\mathbf{D}_v^{(k)}$ are the displacements in position and velocity for the r-th target, and are collected into arrays $\mathbf{D}_r = \left[\mathbf{D}_r^{(1)}, \mathbf{D}_r^{(2)}, \ldots\right]$ and $\mathbf{D}_v = \left[\mathbf{D}_v^{(1)}, \mathbf{D}_v^{(2)}, \ldots\right]$. If the k-th target is fly-by only, the constraint on velocity is not applied and $\mathbf{D}_v^{(k)}$ is not a part of $\mathbf{z}$. If, on the other hand, the target is a rendezvous target, the constraints are in fact applied by equating the equinoctial elements, instead of position and velocity, for simplicity, as this is equivalent.

This restricts the search to the set $Y^* \subset Y$, given as

$$Y^* = \left\{ y \in Y : y = \mathbf{f}^*_{\text{CMap}}(\boldsymbol{\xi}, \mathbf{D}_r, \mathbf{D}_v) \ \forall \xi \in \Xi, \mathbf{D}_r \in R_r, \mathbf{D}_v \in R_v \right\}, \tag{2.39}$$

where R_r and R_v are 3D box sets.

A further restricted search space $Y^- \subset Y^* \subset Y$ is also considered, where the displacements $\mathbf{D}_r$ and $\mathbf{D}_v$ are fixed to zero,

$$Y^- = \left\{ y \in Y : y = \mathbf{f}^*_{\text{CMap}}(\boldsymbol{\xi}, \mathbf{0}, \mathbf{0}) \ \forall \xi \in \Xi \right\}. \tag{2.40}$$

For control maps Y^* and Y^-, the proxy control variable $\mathbf{z}$ is $\mathbf{z} = [\boldsymbol{\xi}, \mathbf{D}_r, \mathbf{D}_v] \in Z^* = \Xi \times R_r \times R_v$ and $\mathbf{z} = \boldsymbol{\xi} \in Z^- = \Xi$, respectively.

Multiple-shooting

Multiple-shooting [7] is used to improve the convergence of deterministic single objective trajectory optimisations, and is applied to the solution of Eq. (2.38). The optimisation variable for such a problem includes the states at the beginning of each arc, i.e. at the nodes, $\mathbf{x}_i^{ON}$ for ON nodes and $\mathbf{x}_i^{OFF}$ for OFF nodes. The states at a node that are obtained by propagating from the previous node are written as $\left(\mathbf{x}_i^{ON}\right)^+$ and $\left(\mathbf{x}_i^{OFF}\right)^+$. Figure 2.3 illustrates these quantities. When optimising with multiple shooting, additional constraints must be added of the form $\mathbf{x}_s^{ON} = \left(\mathbf{x}_s^{ON}\right)^+$ and $\mathbf{x}_s^{OFF} = \left(\mathbf{x}_s^{OFF}\right)^+$, which enforce that the propagation of a node must match the subsequent node.

Max-Min Control Map

In order to reduce the search space even further, and remove the dependency of the dimensionality of the search space Z on the number of uncertain variables n_ξ, another control map, $\mathbf{f}^B_{\text{CMap}}$, is tested, where the search is restricted to the control laws that correspond to a worst case scenario, that is, to the uncertain vector $\boldsymbol{\xi}$ for which the optimal control law is worst, such that the proxy control variable is given as $\mathbf{z} =$

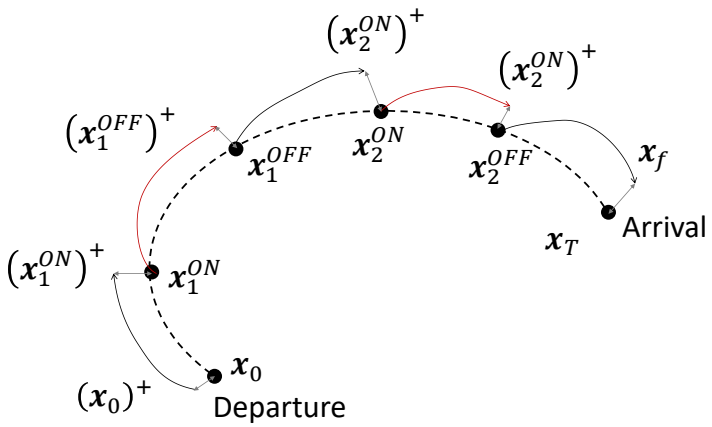


Figure 2.3: The trajectory split in thrust and coast arcs, showing the nodes. Black and red arrows represent the propagation of coast arcs and thrust arcs, respectively. The grey arrows represent the defects.

$[\mathbf{D_r}, \mathbf{D_v}]$. It is defined as

$$\begin{aligned} \mathbf{f}^B_{\text{CMap}}(\mathbf{D}_r, \mathbf{D}_v) &= \mathbf{f}^*_{\text{CMap}}(\boldsymbol{\xi}_M, \mathbf{D}_r, \mathbf{D}_v) \\ s.t.\ \boldsymbol{\xi}_M &= \underset{\boldsymbol{\xi}}{\text{argmax}}\, m_p(\mathbf{f}^*_{\text{CMap}}(\boldsymbol{\xi}, \mathbf{D}_r, \mathbf{D}_v), \boldsymbol{\xi}). \end{aligned} \tag{2.41}$$

The value of $\boldsymbol{\xi}_M$ is defined through a max-min problem, a bilevel problem. As written in Eq. (2.41), the minimisation is hidden in the computation of $\mathbf{f}^*_{\text{CMap}}$, which is the inner problem. To make evident the bilevel nature of this problem, the definition of $\boldsymbol{\xi}_M$ can be written, by combining with Eq. (2.38), as

$$\begin{aligned} \boldsymbol{\xi}_M(\mathbf{D}_r, \mathbf{D}_v) &= \underset{\boldsymbol{\xi}}{\text{argmax}} \min_{\mathbf{y}} m_p(\mathbf{y}, \boldsymbol{\xi}) \\ s.t. \quad \mathbf{r}(\mathbf{y}, \boldsymbol{\xi}, t_{T,k}) &= \mathbf{r}_{T,k}(t_{T,k}) + \mathbf{D}_r^{(k)} \\ \mathbf{v}(\mathbf{y}, \boldsymbol{\xi}, t_{T,k}) &= \mathbf{v}_{T,k}(t_{T,k}) + \mathbf{D}_v^{(k)} \text{ if target is not fly-by only.} \end{aligned} \tag{2.42}$$

Similarly to the work in [62], a surrogate model is used for the inner problem output, i.e. $\tilde{m}_p(\boldsymbol{\xi}) \approx m_p(\mathbf{f}^*_{\text{CMap}}(\boldsymbol{\xi}, \mathbf{D}_r, \mathbf{D}_v), \boldsymbol{\xi})$, and then the outer problem is solved using this surrogate,

$$\boldsymbol{\xi}_M = \underset{\boldsymbol{\xi}}{\text{argmax}}\, \tilde{m}_p(\boldsymbol{\xi}). \tag{2.43}$$

A Kriging model is used and MP-AIDEA [11] is the optimisation algorithm. In the optimisation of $\mathbf{y}^*$, the starting points are the previously obtained solution that is closest to satisfying the constraints, thereby reducing the computational time of this method.

This control map restricts the search space to Y^B, defined as

$$Y^B = \left\{\mathbf{y} \in Y : \mathbf{y} = \mathbf{f}^B_{\text{CMap}}(\mathbf{D}_r, \mathbf{D}_v)\ \forall \mathbf{D}_r \in R_r, \mathbf{D}_v \in R_v\right\}, \tag{2.44}$$

where $Y^B \subset Y$, and R_r and R_v are box sets. For this control map, the proxy control variable $\mathbf{z}$ is $\mathbf{z} = [\mathbf{D}_r, \mathbf{D}_v] \in Z^B = R_r \times R_v$. Because this control map does not depend on $\boldsymbol{\xi}$, its size is decoupled from the number of uncertain variables n_ξ. This makes the surrogate modelling and MO optimisation process more scalable with n_ξ.

Min-Max Control Map

An additional control map, $\mathbf{f}^M_{\text{CMap}}$, is now proposed which defines $\mathbf{z}$ the same way as $\mathbf{f}^B_{\text{CMap}}$, but is formulated as a min-max problem where the constraint is applied on both the minimisation and maximisation subproblems:

$$\begin{aligned} \mathbf{f}^M_{\text{CMap}}(\mathbf{D}_r, \mathbf{D}_v) = &\underset{\mathbf{y}\in\mathbf{Y}}{\operatorname{argmin}} \max_{\boldsymbol{\xi}\in\Xi} m_p(\mathbf{y}, \boldsymbol{\xi}) \\ &s.t. \quad max_{\boldsymbol{\xi}\in\Xi}\ C(\mathbf{y}, \boldsymbol{\xi}) \leq \epsilon \end{aligned} \tag{2.45}$$

where the constraint $C(\mathbf{y}, \boldsymbol{\xi})$ is defined as

$$C(\mathbf{y}, \boldsymbol{\xi}) = \sum_k \left\|\mathbf{r}(\mathbf{y}, \boldsymbol{\xi}, t_{T,k}) - \mathbf{r}_{T,k}(t_{T,k})\right\|^2, \tag{2.46}$$

for flyby constraints, and as

$$C(\mathbf{y}, \boldsymbol{\xi}) = \sum_k \left\|\mathbf{x}(\mathbf{y}, \boldsymbol{\xi}, t_{T,k}) - \mathbf{x}_{T,k}(t_{T,k})\right\|^2, \tag{2.47}$$

for the rendezvous constraints, where $\mathbf{x}$ are the equinoctial elements, with the semi-major axis a in astronomical units.

Equation (2.45) is solved using *MacMinMax* [25], a solver for Min-Max problems with strict constraints. Because the constraints cannot be met for all $\boldsymbol{\xi}$, the constraint threshold $\epsilon \in \mathbb{R}$ is found by *MacMinMax* via its iterative constraint relaxation process. This algorithm works by solving outer, inner, and constraint subproblems, which correspond, to put it in somewhat simplistic terms, to the minimisation of the objective function as a function of $\mathbf{y}$, the maximisation of the objective function as a function of $\boldsymbol{\xi}$, and the maximisation of the constraints as a function of $\boldsymbol{\xi}$, respectively [25].

The *MacMinMax* algorithm allows choosing the algorithms used for the outer, inner, and constraint subproblems. The outer and constraint subproblems are run using MATLAB's *fmincon* ®, while the inner subproblem is solved using MP-AIDEA [11].

This control map defines the following space of control vectors,

$$Y^M = \left\{ \mathbf{y} \in Y : \mathbf{y} = \mathbf{f}^M_{\mathrm{CMap}}(\mathbf{D}_r, \mathbf{D}_v) \; \forall \mathbf{D}_r \in R_r, \mathbf{D}_v \in R_v \right\}, \tag{2.48}$$

which does not necessarily satisfy $Y^M \subset Y^*$, unlike Y^B, since there is no guarantee that the resulting control laws satisfy the constraints for any specific values of $\boldsymbol{\xi}, \mathbf{D}_r, \mathbf{D}_v$. The proxy control variable here is the same as for Y^B, i.e. $\mathbf{z} = [\mathbf{D}_r, \mathbf{D}_v] \in Z^B = R_r \times R_v$, but here $\mathbf{z}$ is interpreted differently, resulting in different control laws.

2.8 Surrogate Model

A MO optimisation problem requires a large number of objective function evaluations, which can be very computationally expensive when epistemic uncertainty is being modelled. Chapter 3, which describes how $\underline{\mathrm{E}}$ are computed, will also make it clearer why this is a computationally expensive task. Furthermore, the computation of the control map adds to the computational cost of evaluating the objectives.

To deal with this problem, surrogate models are used. Instead of the real many-objective function $\mathbf{F}$ being used by MACS, MACS operates instead on a surrogate model $\tilde{\mathbf{F}}(\mathbf{d}) \approx \mathbf{F}(\mathbf{d})$. For problem (2.37), the lower expectations are approximated by surrogate models $\underline{\tilde{\mathrm{E}}}_{\boldsymbol{\phi}}(\mathbf{z}, \boldsymbol{\nu}) \approx \underline{\mathrm{E}}(\boldsymbol{\phi}(\mathbf{f}_{\mathrm{CMap}}(\mathbf{z})), \boldsymbol{\nu})$, likewise for $\mathbf{g}$ and $\boldsymbol{\Psi}$, and MACS is

tasked with solving

$$\min_{\mathbf{z},\boldsymbol{\nu}}[-\underline{\tilde{\mathrm{E}}}_{\boldsymbol{\phi}}(\mathbf{z},\boldsymbol{\nu}),\ -\underline{\tilde{\mathrm{E}}}_{\mathbf{g}}(\mathbf{z},\boldsymbol{\nu}),\ -\underline{\tilde{\mathrm{E}}}_{\boldsymbol{\Psi}}(\mathbf{z},\boldsymbol{\nu}),\ \boldsymbol{\nu}]^T \tag{2.49}$$

This allows sidestepping the computation of the control map and the lower expectation within MACS. After MACS runs, for the sample points it returns of $\mathbf{z}, \boldsymbol{\nu}$, the exact values of the objectives are computed and added to the training data of the surrogate model, which is re-trained. Note that, in this formulation, there is no need to model $\boldsymbol{\nu}$ with a surrogate model. More details on this process are in the subsequent chapters.

In this work, this surrogate is built with the Kriging model implemented in the DACE toolbox [49]. Other choices are possible but the main reason for using Kriging and DACE is the simplicity of implementation and the flexibility offered by this model when it comes to the choice of the samples required to build the surrogate. Its predictor is given by [43]

$$\tilde{F}_k(\mathbf{p}) = \mathbf{b}_{Fk}(\mathbf{p})\boldsymbol{\beta}_{Fk} + \boldsymbol{\rho}_{Fk}(\mathbf{p};\boldsymbol{\theta}_{Fk})^T\boldsymbol{\gamma}_{Fk}\ . \tag{2.50}$$

The parameters optimised during training are $\boldsymbol{\beta}_{Fk}$, $\boldsymbol{\theta}_{Fk}$, and $\boldsymbol{\gamma}_{Fk}$. The function $\mathbf{b}_{Fk}$ is the regression function, $\boldsymbol{\beta}_{Fk}$ the regression coefficients, $\boldsymbol{\rho}_{Fk}$ is the correlation function, parameterised by $\boldsymbol{\theta}_{Fk}$, which produces a matrix with the correlation coefficients between the input and each training point, and $\boldsymbol{\gamma}_{Fk}$ is a weight vector for the correlations. The training process used to find the aforementioned parameters is described in [43].

The regression $\mathbf{b}_{Fk}$ and correlation $\boldsymbol{\rho}_{Fk}$ functions are chosen by the user. In this work, the regression function used is linear, so that $\mathbf{b}_K(\mathbf{p}) = [1\ \mathbf{p}^T]$, and the correlation function is exponential,

$$\rho_{Fk}^{(i)}(\mathbf{p};\boldsymbol{\theta}_{Fk}) = \prod_j \exp\left(-\theta_{Fk}^{(i)}\left|p_j - p_j^{(i)}\right|\right)\ , \tag{2.51}$$

where $\mathbf{p}^{(i)}$ is the i-th training point.

In the application in this work, the design point $\mathbf{p}$ contains the proxy control variable $\mathbf{z}$ and the vector of thresholds $\boldsymbol{\nu}$, $\mathbf{p} = [\mathbf{z}^\mathbf{T}, \boldsymbol{\nu}^\mathbf{T}]^\mathbf{T}$. There are two noteworthy implementation details regarding the use of these surrogate models in this work.

The first is that often some of the variables in $\mathbf{p}$ cannot influence the objective being approximated, namely only one of the thresholds ν in $\boldsymbol{\nu}$ refer to each objective. When this happens, these variables are not considered in the surrogate model defined for that particular objective. To allow this, the surrogate models are trained on each objective function independently, which has the side-effect that a different $\boldsymbol{\theta}_{Fk}$ is found for the surrogate of each objective.

Secondly, when computing upper quantiles, see Chapter 3, it is necessary to evaluate a surrogate model many times with the same value of $\mathbf{y}$ but varying ν. The predictor in Eq. (2.50), with our choice of regressor $\mathbf{b}_{Fk}$ and correlation ρ_{Fk} functions, can also be written as

$$\begin{aligned}\tilde{F}_k(\mathbf{p}) = \beta_{Fk;0} + \mathbf{b}_{Fk}(\mathbf{z})\boldsymbol{\beta}_{Fk;z} + \mathbf{b}_{Fk}(\nu)\beta_{Fk;\nu} + \\ \left(\boldsymbol{\rho}_{Fk}(\mathbf{z};\boldsymbol{\theta}_{Fk;z}) \odot \boldsymbol{\rho}_{Fk}(\nu;\theta_{Fk;\nu})\right)^T \boldsymbol{\gamma}_{Fk} \ ,\end{aligned} \tag{2.52}$$

where $\odot$ is the Hadamard (i.e. element-wise) product. As ν is varied while keeping $\mathbf{z}$ constant, the values of $\mathbf{b}_{Fk}(\mathbf{z})\boldsymbol{\beta}_{Fk;z}$ and $\boldsymbol{\rho}_{Fk}(\mathbf{z};\boldsymbol{\theta}_{Fk;z})$ do not need to be recomputed each time. Instead, they are computed only once for each estimation of the upper quantile, which is an example of memoisation.

2.9 Conclusion

This chapter presented the formulation of the optimisation problems that are to be tackled in the rest of the thesis. The objectives to be optimised were defined, alongside the constraints. One important aspect is the fact that these objectives all involve the quantification of epistemic uncertainty. While this was only mentioned in this chapter, the next two chapters will be mostly focused on the quantification of epistemic uncertainty. After the formulation, the practical aspects of the optimisation of such a problem were discussed.

The formulation of the problem presented initially involves searching through the space of all possible trajectories, which is impractical since this is a space of functions, which cannot be easily searched with numerical methods. To solve it computationally,

it is first transcribed, i.e., a version of the problem that can be passed to an existing algorithm, in this case MACS, is formulated. In the transcription process, the constraints, that are also affected epistemic uncertainty, were turned into objectives, to ensure that the problem always has a solution, since it may not be generally known a-priori if the problem is feasible. This results in a many-objective optimisation problem.

The rest of the chapter discussed the practical aspects of optimising this problem. The first necessary step is to introduce the many-objective search algorithm, MACS, which was developed by other researchers and applied to this problem. In addition, two other tools were used to complement this algorithm, the control maps and the surrogate model. The control maps map the search space of all control laws that can be represented by the transcription method onto a lower dimensional search space. This has the purpose of simplifying the search process by reducing its dimensionality, but also to focus the search on subsets of the original space that are more likely to contain robust solutions. The surrogate models are used to speed up the search, since even with the control maps, solving a many-objective problem requires a large number of function evaluations.

Chapter 3

Epistemic Uncertainty Quantification

As discussed in Chapter 1, uncertainty can be divided into two major types: aleatory and epistemic. Aleatory uncertainty is that which cannot be reduced, i.e., it is an inherent property of the uncertain system, which is why it is also sometimes called irreducible uncertainty. This is the most commonly used type of uncertainty, and it corresponds to that which is normally taught in probability and statistics courses. The unknown quantities in a system subject to aleatory uncertainty are modelled with probability distributions, which are assumed to be known exactly. Some uncertainty, however, is due to a lack of knowledge, the so called epistemic uncertainty. In this case, it may happen that one is not even able to attribute a probability distribution to a set of possible events. This work focuses on this type of uncertainty. In this Chapter, the theoretical and practical aspects of quantifying epistemic uncertainty are discussed.

3.1 Related Work

While it is far more common to consider aleatory uncertainty only, some examples of trajectory optimisation under epistemic uncertainty can be found in the literature. In [13], low-thrust trajectories are optimised under epistemic uncertainty, defined using Dempster Shafer theory of evidence, also referred to as evidence theory. With evidence

theory, experts assign a probability mass to subsets, called focal elements, of the set of possible realisations of the uncertain variables, representing the degree of belief these experts have that the variables will lie in that focal element, without specifying exact probability values for the realisations within that set. One of the challenges of this approach is the exponential growth of focal elements with the number of epistemically uncertain variables. Another issue is the non-differentiability of the belief function. [13] In [22], an approximation of the belief function is proposed for faster computational time and better differentiability. A different approach was followed in [30], which considers distributions with interval valued parameters. In [30], the authors propose a filter based approach for epistemic uncertainty propagation. To reduce the computational cost, a Gaussian mixture model with interval valued weights is used, and at the time of inference from observations, variational inference is used to restart the Gaussian mixture, to keep the number of kernels from increasing.

This work does not delve significantly into evidence theory. However, the theoretical developments that support some of the methodology herein proposed are briefly considered in the context of evidence theory, for the sake of completion. Instead, the focus is on other approaches, based on imprecise probabilities [71]. In this formulation of epistemic uncertainty, since the lack of knowledge has the consequence that one is not able to attribute an exact probability distribution to the uncertain variables, instead one considers a set of possible distributions of these variables. To quantify epistemic uncertainty, the lower expectation, defined as the minimum expected value of some quantity of interest with distributions within such a set, is used.

One approach, proposed in [68], defines the set of distributions using Bernstein polynomials, a formulation which makes the lower expectation obtainable as the solution of a linear programming problem. In this chapter, this approach is expanded upon, as also published in [14], by developing a more efficient algorithm for the estimation of the lower expectation for this formulation of the epistemic uncertainty.

3.2 Definitions

When uncertainty is epistemic, the probability distribution followed by the uncertain variables $\boldsymbol{\xi} \in \Xi$ is often unknown or known with a degree of imprecision. Multiple sources of information may suggest different distributions. Thus, by combining these sources of information, one obtains a set $\mathcal{Q} \subset \mathcal{P}(\Xi)$ that contains all distributions compatible with the available information, where $\mathcal{P}(\Xi)$ is the set of all valid probability distributions over Ξ, henceforth written just as $\mathcal{P}$. It is assumed that sets $\mathcal{Q}$ are always non-empty. The lower expectation $\underline{\mathrm{E}}$ captures this imprecision by measuring the lowest value of the expectation obtainable with distributions in $\mathcal{Q}$, i.e., the lower expectation of quantity h, $\underline{\mathrm{E}}(h)$, see Definition 3.2.2. Using this definition, the upper quantile [71] is also defined in Definition 3.2.3.

Definition 3.2.1. The expected value of a function $h(\boldsymbol{\xi}) : \Xi \to \mathbb{R}$, under a distribution with probability density function given by p, is given by

$$\mathrm{E}(h; p) = \int_{\Xi} h(\boldsymbol{\xi}) p(\boldsymbol{\xi}) \, \mathrm{d}\boldsymbol{\xi}. \tag{3.1}$$

Definition 3.2.2. If $h(\boldsymbol{\xi}) : \Xi \to \mathbb{R}$ is a generic quantity of interest, the lower expectation operator $\underline{\mathrm{E}}$ is defined as follows:

$$\underline{\mathrm{E}}(h) = \min_{p \in \mathcal{Q}} \mathrm{E}(h; p) = \min_{p \in \mathcal{Q}} \int_{\Xi} h(\boldsymbol{\xi}) p(\boldsymbol{\xi}) d\boldsymbol{\xi} \ , \tag{3.2}$$

where the function $p(\boldsymbol{\xi})$ is a probability distribution belonging to a family of distributions $\mathcal{Q} \subset \mathcal{P}(\Xi)$, $\mathcal{P}(\Xi)$ is the family of all possible probability distributions over Ξ, and Ξ is the space of the uncertain variables $\boldsymbol{\xi}$. When written as $\underline{\mathrm{E}}(h < \nu)$, this refers to the lower expectation of the indicator function which is 1 if $h(\boldsymbol{\xi}) < \nu$ and 0 otherwise.

Just as the expected value of an indicator function such as $\phi_1 < \nu_1$ is the probability of $\phi_1 < \nu_1$ being verified, the lower expectation of the same quantity can be seen as its lower probability. With this rationale, the quantities of the form $\underline{\mathrm{E}}(\boldsymbol{\phi}(\mathbf{y}), \boldsymbol{\nu})$ in Eq. (2.28) are formulated as $[\underline{\mathrm{E}}(\phi_1 < \nu_1), \underline{\mathrm{E}}(\phi_2 < \nu_2), \ldots]$, representing a lower-bound on the probability of these chance constraints being met. If a quantity of interest is

a function of variables other than $\boldsymbol{\xi}$, e.g. $h(\mathbf{y}, \boldsymbol{\xi})$, writing $\underline{\mathrm{E}}(h(\mathbf{y}))$ denotes the lower expectation $\underline{\mathrm{E}}(h^*)$ of a function $h^*(\boldsymbol{\xi}) = h(\mathbf{y}, \boldsymbol{\xi})$, for a specific value of $\mathbf{y}$.

Definition 3.2.3. For a scalar quantity of interest $h(\boldsymbol{\xi})$, the upper quantile [71] $\overline{\nu}_{\hat{\mathrm{E}}}$ is defined as the threshold ν for which the lower expectation equals $\hat{\mathrm{E}}$,

$$\overline{\nu}_{\hat{\mathrm{E}}}(h) = \nu : \underline{\mathrm{E}}(h < \nu) = \hat{\mathrm{E}} \ , \tag{3.3}$$

where $h < \nu$ is an indicator function which is 1 if $h(\boldsymbol{\xi}) < \nu$ and 0 otherwise. For all probability distributions in $\mathcal{Q}$, the $\hat{\mathrm{E}}$-quantile of h is at most $\overline{\nu}_{\hat{\mathrm{E}}}(h)$.

The definition of the family of distributions $\mathcal{Q}$ is crucial as it defines both the applicability of this uncertainty quantification as well as the efficiency of its estimate. If the family of distributions is the set $\mathcal{P}$, the lower expectation $\underline{\mathrm{E}}(h)$ will be the minimum of h. To get more useful, less pessimistic uncertainty quantification, some information must be used to constrain the family of distributions. Types of information used to constrain the set $\mathcal{Q}$, as found in the literature, include [24] upper and lower bounds of $\boldsymbol{\xi}$, moments such as mean and variance, correlations between variables, and basic probability assignments (evidence theory). There are many epistemic uncertainty quantification methods, but the formalism outlined herein is applicable to a large variety, including all these examples. The bounds are typically considered implicitly, by only considering distributions defined over Ξ, as done in this work. The next sections describe some ways of modelling epistemic uncertainty, how they can be formulated using the terminology in this work, i.e. what is $\mathcal{Q}$, and how $\underline{\mathrm{E}}$ and $\overline{\nu}$ are calculated in practice.

3.3 Lower Expectation based on Bernstein polynomials

Following [68], the family of probability distributions $\mathcal{Q}$ is defined using Bernstein polynomials. These Bezier curves are a linear combination of positive basis functions. If the coefficients are positive, the resulting function is guaranteed to be positive, which makes them ideal to represent probability distributions.

Let $b_{j;q}(x)$ be the j^{th} univariate Bernstein basis functions of order $q \in \mathbb{N}$, scaled so that they are probability density functions (PDF), that is, they integrate to 1 in their domain $[0, 1]$:

$$b_{j;q}(x) = (q+1)\binom{q}{j} x^j (1-x)^{q-j},\ j \in \{0, \ldots, q\}\,. \tag{3.4}$$

The multivariate basis functions B are simply the products of univartiate basis functions:

$$B_{\mathbf{j};\mathbf{q}}(\mathbf{x}) = \prod_{k=1}^{n_\xi} b_{j_k;q_k}(x_k), \tag{3.5}$$

where $q_k \in \mathbb{N}$ indicates the degree of the polynomial along dimension k, and $\mathbf{j} = (j_1, \ldots, j_{n_\xi})$ is a multi-index, i.e., a tuple of n_ξ indices, each of which is between 0 and q_k. Bernstein basis functions are only defined for $x \in [0, 1]$, so a linear function $\boldsymbol{\tau}$ is applied so that $\boldsymbol{\tau}(\boldsymbol{\xi}) \in [0\ 1]^{n_\xi}\ \forall \boldsymbol{\xi} \in \Xi$.

The family of distributions could be defined as a linear combination of multi-variate Bernstein basis functions, as in Eq. (3.6). This allows representing multi-variate distributions (see for example Section 2.4 of [32]), including those of correlated variables. The problem of finding the lower expectation is a linear program, but the complexity is exponential with n_ξ, as the number of coefficients is $(q+1)^{n_\xi}$, q being the degree of the polynomial.

$$\mathcal{Q}_m = \left\{ \begin{array}{l} p_m(\boldsymbol{\xi}, \mathbf{c}) = \displaystyle\sum_{\mathbf{j} \in \mathcal{J}} c_{\mathbf{j}} B_{\mathbf{j};\mathbf{q}}\left(\tau(\boldsymbol{\xi})\right) \\ \forall \mathbf{c} > 0 : \displaystyle\sum_{\mathbf{j} \in \mathcal{J}} c_{\mathbf{j}} = 1 \end{array} \right\}, \tag{3.6}$$

where $\mathcal{J} = \{0, \ldots, q_1\} \times \ldots \times \{0, \ldots, q_{n_\xi}\} \subset \mathbb{N}^{n_\xi}$ represents the set of multi-indices $\mathbf{j}$.

An alternative is to have the distributions be the product of univariate Bernstein polynomials, as in equation (3.7). In this new family of distributions only independent variables are possible, since the distributions are defined as the product of univariate functions. The number of coefficients is now $(q+1) \times n_\xi$, which no longer grows exponentially with the number of uncertain variables n_ξ. However, Problem (3.2) is no

longer a linear program.

$$\mathcal{Q}_u = \left\{ \begin{array}{l} p_u(\boldsymbol{\xi}; \mathbf{c}) = \prod_{k=1}^{n_\xi} \sum_{j=0}^{q_k} c_j^{(k)} b_{j;q_k}(\tau_k(\xi_k)) \\ \forall \mathbf{c} > 0 : \sum_j c_j^{(k)} = 1 \ \forall k \end{array} \right\}. \tag{3.7}$$

Using either of these families of distributions in the definition of the lower expectation produces the same results. In fact, the discrete set of distributions $\mathcal{Q}_j$ in Eq. (3.8) is also equivalent to either of the other two in terms of the resulting value of the lower expectation. Since this set only requires n_ξ integer coefficients, it is preferable to perform the optimisation in this set.

$$\mathcal{Q}_j = \{ B_{\mathbf{j};\mathbf{q}}(\tau(\boldsymbol{\xi})) \ \ \forall \mathbf{j} \in \mathcal{J} \subset \mathbb{N}^{n_\xi} \}. \tag{3.8}$$

Furthermore, solving for family $\mathcal{Q}_m$ requires evaluating all elements of the set $\mathcal{Q}_j$ just to calculate the parameters for the linear program. Therefore it is less efficient than performing a brute-force search on this discrete set. Algorithms that are less computationally demanding will be described in Section 3.3.2, using $\mathcal{Q}_j$ as the family of distributions.

The lemmas that follow formally demonstrate that both families $\mathcal{Q}_m$ or $\mathcal{Q}_u$ are equivalent to $\mathcal{Q}_j$ when computing an approximation to the lower expectation $\underline{\mathrm{E}}$. Their proofs are in Appendix A.

Lemma 1. *If $\underline{\mathrm{E}}$ is defined using the family $\mathcal{Q}_m$, there is always at least one distribution $\hat{p} \in \mathcal{Q}_j$ such that $\underline{\mathrm{E}}(h) = \mathrm{E}(h; \hat{p})$.*

Lemma 2. *If $\underline{\mathrm{E}}$ is defined using the family $\mathcal{Q}_u$, there is always at least one distribution $\hat{p} \in \mathcal{Q}_j$ such that $\underline{\mathrm{E}}(h) = \mathrm{E}(h; \hat{p})$.*

Henceforth, the family of distributions considered is always $\mathcal{Q}_j$ defined in Equation (3.8). For ease of notation, $B_{\mathbf{j};\mathbf{q}}$ will be written as $B_{\mathbf{j}}$. The optimisation problem, in Eq. (3.2), is now written as

$$\underline{\mathrm{E}}(h) = \min_{\mathbf{j} \in \mathcal{J}} \mathrm{E}(h; B_{\mathbf{j}}). \tag{3.9}$$

3.3.1 Estimating Expectation

Given the difficulty of calculating the integral in Eq. (3.2) analytically, a quasi-Monte Carlo approach is employed. This is identical to a Monte Carlo approach, but instead of selecting the samples randomly, a deterministic, low discrepancy sequence $\boldsymbol{\xi}^{(1)}, \ldots, \boldsymbol{\xi}^{(n)}$ is used. These sequences are specifically designed to have low integration errors, by attempting to spread the points uniformly over the space. A random sample, on the other hand, tends to result in some regions having a higher concentration of sample points than others. This is supported by theoretical proofs that quasi-Monte Carlo integration methods have better asymptotic convergence, as there is an error upper-bound for this integration scheme, the Koksma-Hlawka inequality, that is proportional to $1/N$, while for traditional Monte Carlo, the average error is proportional to $1/\sqrt{n}$ [48]. It should be noted that the inequality itself does not actually produce useful bounds on the integration error for practically realizable values of n when a large number of variables is used, as discussed later in this section, but the fact that the sequence is designed with integration in mind still leads to it being preferable to a random sequence. The Halton sequence [31] is chosen for this purpose, due to its simplicity compared to other low discrepancy sequences. The expectation $\mathrm{E}(h; p)$ can be estimated as

$$\hat{E}_U(h; p) = \frac{1}{n} \sum_{i=1}^{n} h(\boldsymbol{\xi}^{(i)}) p(\boldsymbol{\xi}^{(i)}) \ , \quad \boldsymbol{\xi}^{(i)} \sim U. \tag{3.10}$$

Here, $\boldsymbol{\xi}$ is not a random variable, but a deterministic sequence, the Halton sequence. The expression $\boldsymbol{\xi} \sim U$ means that $\boldsymbol{\xi}$ has uniform distribution. This corresponds to taking the Halton sequence as is, with no transformation. Later we will consider modifying the Halton sequence so that its distribution is not uniform.

A well known upper bound on the absolute error resulting from quasi-Monte Carlo is given by the Koksma-Hlawka inequality [48]. However, this upper bound cannot be used for the purposes of this work. As explained in [48], it requires an upper bound on the discrepancy of the sequence of samples. By definition, the discrepancy must be lower than 1, but for $n_\xi = 10$, the value of the upper bounds in [48], which decrease

proportionally to $1/n$, are lower than the trivial value of 1 only when $n > 10^{19}$, a consequence of the greater-than-exponential growth of these bounds with the number of variables n_ξ.

Therefore, the analysis of the error was performed using statistical methods instead [31], as if $\boldsymbol{\xi}$ were randomly sampled, and afterwards an experimental analysis was performed for a special case of indicator functions that can be easily integrated analytically.

Using probability theory, the standard deviation σ_U of the quantity in Eq. (3.10), when applied to an indicator function $I = h < \nu$ can be written as

$$\sigma_U^2 = \frac{1}{n}\left(\int I^2 p^2 \mathrm{d}\boldsymbol{\xi} - \left(\int I p \mathrm{d}\boldsymbol{\xi}\right)^2\right) \leq \frac{1}{n}\int p^2 \mathrm{d}\boldsymbol{\xi}, \tag{3.11}$$

since $0 \leq I \leq 1$ in all domain. This expression is easily estimated, since the probability distribution is over independent variables:

$$\sigma_U^2 \leq \frac{1}{n}\prod_{k=1}^{n_\xi}\int_0^1 \left(b_{j_k,q_k}(x_k)\right)^2 \mathrm{d}x_k. \tag{3.12}$$

where $x_k = \tau_k(\xi_k)$.

Lemma 3. *The highest value of* $\int_0^1 \left(b_{j,q}(x)\right)^2 \mathrm{d}x$ *occurs for both* $j = 0$ *and* $j = q$, q *being the degree of the Bernstein basis function* $b_{j,q}(x)$.

Proof. (With software aid.) Let $v_j = \int_0^1 \left(b_{j,q}(x)\right)^2 \mathrm{d}x$. Using Wolfram Mathematica ®, one obtains

$$v_{j+1} - v_j = \binom{q+1}{j+1}^2 \binom{2q+1}{2j}^{-1} \frac{(j+1)(1+2j-q)}{4(j-q)^2-1}\ , \quad 0 \leq j < q\ , \tag{3.13}$$

which shows that $\mathrm{sgn}(v_{j+1} - v_j) = \mathrm{sgn}(1 + 2j - q)$, where $\mathrm{sgn}(\cdot)$ is the sign function. Thus for $j < \frac{q-1}{2}$, $v_j > v_{j+1}$ and for $j > \frac{q-1}{2}$, $v_{j+1} > v_j$, showing that the maximum value of v_j must occur at either $j = 0$ or $j = q$, the two extremes of this variable. Since there is a symmetry in the Bernstein basis functions, it holds that $v_j = v_{q-j}$, and thus both v_0 and v_q maximise v_j. □

From Lemma 3, if q is the degree of the polynomials in each dimension, the upper bound on σ_U^2 becomes:

$$\frac{1}{n}\left(\frac{(q+1)^2}{2q+1}\right)^{n_\xi}. \tag{3.14}$$

This upper bound for the standard deviation grows exponentially with n_ξ. For example, if $n_\xi = 10$, $q = 4$, for the standard deviation to be below 0.01, $n > 2.7 \times 10^8$ samples would be necessary.

To reduce the number of samples required to obtain sufficient accuracy, importance sampling is used. By changing the sampling of the variables and correcting the values of the samples appropriately, it is possible to obtain an estimator that is also unbiased, but which has a different, ideally lower, variance. One possibility is to have $\boldsymbol{\xi} \sim B_{\mathbf{j}}$, henceforth referred to as *P-sampling*.

A sequence following a univariate distribution P with cumulative distribution function (CDF) $C(x)$, is obtained from a uniform distribution $x_u \sim U$, by having $x_P = C^{-1}(x_u) \sim P$. With a multivariate distribution of independent variables, such as $B_{\mathbf{j}}$, if $C_k(\mathbf{c})$ is the CDF along the k^{th} variable, $x_{k,P} = C_k^{-1}(x_{k,U})$ results in a sampling $\boldsymbol{x}_P \sim B_{\mathbf{j}}$. The inverse of C_k cannot, in general, be found analytically, so instead it is approximated using linear interpolation with 10000 intervals. Appendix B contains an explanation of this method as well as a discussion of its accuracy.

With P-sampling, the estimator becomes:

$$\hat{\mathrm{E}}_P(I;p) = \frac{1}{n}\sum_{i=1}^{n} I(\xi_i)\ , \quad \xi_i \sim p \tag{3.15}$$

The standard deviation now follows

$$\sigma_P^2 = \frac{1}{n}\left(\int I(\xi)^2 p(\xi)\mathrm{d}\xi - \mathrm{E}^2\right) \leq \frac{1}{n}(\mathrm{E} - \mathrm{E}^2) \leq \frac{1}{4n}\,, \tag{3.16}$$

again using the fact that $0 \leq I \leq 1$. This new expression does not increase with n_ξ, thus, in order to obtain a similar limit on the standard deviation as before, one only needs 2500 samples. In all the case studies in this work the number of samples is 5000.

To justify the choice of using P-sampling, a numerical experiment was run. For a number of randomly generated hyper-rectangular indicator functions, the exact integral was calculated using analytical formulae, and estimated with both strategies considered here. The degree q of the polynomial is 4, and n_ξ was between 1 and 10. Table 3.1 contains the maximum absolute errors with 10000 test cases obtained by comparing both estimations with the analytical result. This test is equivalent to estimating the expected value of quantities $h(\boldsymbol{\xi}) < \nu$ which are hyper-rectangular shaped. This work focuses on lower expectations of indicator functions. Regardless of their shape, for a realistic h, such indicator functions can be approximated as the union of disjoint hyper-rectangular subsets of Ξ, an approximation that could be obtained, for example, by dividing the space into a fine grid and creating a union of the cells which intersect the set defined by $h(\boldsymbol{\xi}) < \nu$. As such, this test is believed to be representative of more complex functions h. The results in the first column are for uniform sampling with $n = 1.5 \times 10^6$ samples, and in the second are for P-sampling with $n = 5000$ samples. Table 3.1 shows that even with several orders of magnitude less samples the expectation can still be estimated with good accuracy when using P-sampling, and that for high n_ξ P-sampling is better than uniform sampling.

Table 3.1: Maximum absolute errors for different sampling techniques

	Uniform sampling $n = 1.5 \times 10^6$	P-sampling $n = 5000$
n_ξ	max abs error	max abs error
1	8.3×10^{-6}	6.4×10^{-4}
2	5.5×10^{-5}	1.4×10^{-3}
3	3.1×10^{-4}	1.5×10^{-3}
4	1.0×10^{-3}	2.3×10^{-3}
5	3.1×10^{-3}	2.5×10^{-3}
6	1.1×10^{-2}	4.1×10^{-3}
7	2.9×10^{-2}	3.6×10^{-3}
8	7.9×10^{-2}	4.7×10^{-3}
9	1.7×10^{-1}	4.3×10^{-3}
10	5.2×10^{-1}	5.7×10^{-3}

The fact that $\boldsymbol{\xi}$ needs to be re-sampled for each evaluation of $\mathrm{E}(h; p)$ with a different

p, is a disadvantage, but, according to Eq. (3.14), for a given σ^2, the number of samples that one would need to generate with uniform sampling grows exponentially with n_ξ, as opposed to the number of samples generated with P-sampling that, according to Eq. (3.16), is constant with n_ξ. Given that, with P-sampling, a new set of samples is generated for each call to E, the theoretical maximum number of estimates of $\mathrm{E}(h, p)$ below which P-sampling leads to a lower number of evaluations of I than uniform sampling, assuming the same level of accuracy, is

$$N_E^* = 4\left(\frac{(q+1)^2}{2q+1}\right)^{n_\xi}. \tag{3.17}$$

3.3.2 Expectation Minimisation Algorithms

The value of the lower expectation is obtained by finding the multi-index $\mathbf{j}$ that solves Eq. (3.9). Applying an enumerative approach that goes through each possible combination of indices is not a viable solution because the time complexity would be exponential with the number of dimensions. Thus, in order to keep the algorithmic time complexity polynomial with n_ξ, in this section, we propose the use of a pattern search algorithm. With reference to Fig. 3.1, at each iteration k, the pattern search logic we propose takes the multi-index $\mathbf{j}^k = (j_1^k, \ldots, j_{n_\xi}^k)$, and a neighbourhood N_k, and selects the multi-index $\mathbf{j}^{k+1} \in N_k$ that minimises $\mathrm{E}(h; B_{\mathbf{j}^{k+1}})$. The neighbourhood N_k is composed of multi-indices where only one index is different from $\mathbf{j}^k$. We consider two different neighbourhoods, N_k^B and N_k^S (see Fig. 3.1).

The former contains every multi-index in $\mathcal{J}$ that is identical to $\mathbf{j}^k$ in all but one index. The latter contains only the multi-indices in the former for which the different index only differs by one unit from the corresponding index in $\mathbf{j}^k$.

$$N_k^B(\mathbf{j}^k) = \left\{\mathbf{j} \in \mathcal{J} \setminus \{\mathbf{j}^k\} : \exists m, \forall i \neq m, \mathbf{j}_i = \mathbf{j}_i^k\right\} \tag{3.18}$$

$$N_k^S(\mathbf{j}^k) = \left\{\mathbf{j} \in N_k^B : \exists m, \mathbf{j}_m = \mathbf{j}_m^k \pm 1\right\} \tag{3.19}$$

This pattern search scheme is not guaranteed to always find the global optimum.

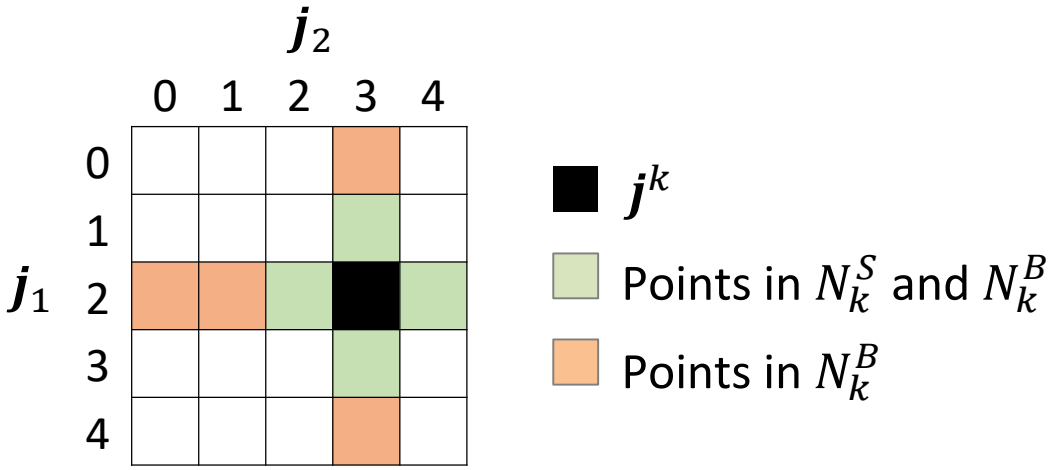


Figure 3.1: Diagram illustrating the neighbourhoods N_k^B and N_k^S around a 2D multi-index $\mathbf{j}^k$.

The solution returned at iteration k depends on the initial guess $\mathbf{j}^0$. Thus we considered different initialisations. One is a greedy initialisation, starting from:

$$\forall k,\ \mathbf{j}_k^0 = \underset{i}{\operatorname{argmin}} \int_{\Xi} h(\boldsymbol{\xi}) b_{i,q_k}(\xi_k) \mathrm{d}\boldsymbol{\xi}. \tag{3.20}$$

Let $\mathbf{j}_k^{f_1}$ be the outcome of the pattern search algorithm starting from the initial point in Eq. (3.20). A second initialisation re-starts the search from the point $\mathbf{j}_k^{f_1+1}$ defined as:

$$j_k^{f_1+1} = q_k - j_k^{f_1}\ \forall k. \tag{3.21}$$

This is also referred to as "symmetric restart" in the results, and the solution obtained from this new starting point is termed $\mathbf{j}^{f_2}$. When using this restart, the better solution of the two $j^{\omega_2} = \operatorname{argmin}_{\mathbf{j}\in\{\mathbf{j}^{f_1},\mathbf{j}^{f_2}\}} \mathrm{E}(h;\mathbf{j})$ is retained.

We also considered an initialisation from a random starting points and a multi-restart procedure. These alternative initialisations of the proposed pattern search algorithm are compared with each other in Section 5.3.1, and against a number of other heuristic search algorithms, namely simulated annealing, taboo search, and Matlab's® genetic algorithm, in Section 5.3.3.

3.4 Lower Expectation with Linear Constraints

In [27], Golodnikov et al. approach the problem where one only the mean and variance of a distribution are known, but not the exact distribution. Common practice for the particular problem under study, Golodnikov et al. state, is to assume a known distribution, and to find the parameters that make the distribution match these moments. Golodnikov et al. then show, with some examples, how different distributions lead to very different estimates of a parameter. That work never explicitly mentions epistemic uncertainty, but it is directly dealing with the problem of an unknown probability distribution - only mean and variance are known, but not the actual distribution - and therefore it is handling epistemic uncertainty.

Despite being motivated by a scenario where the first two moments are known, their formulation is more general, and applies to any linear constraints. In this section, this work is summarised, explained following our terminology, and extended to the case where the constraints are interval valued. Afterwards, the method used in this work to estimate the lower expectation with this type of uncertainty is described. Chapter 4, for feedback control under epistemic uncertainty, is mostly based on this formulation of epistemic uncertainty, even though it applies more generally.

This section considers epistemic uncertainty constraints which can be defined by writing the family of distributions as:

$$\mathcal{Q} = \left\{ p(\boldsymbol{\xi}) \in \mathcal{P}(\Xi) : c_i^L \leq \int_{\Xi} p(\boldsymbol{\xi}) g_i(\boldsymbol{\xi}) \, \mathrm{d}\boldsymbol{\xi} \leq c_i^U, \ \forall i \in \{1, \ldots, n_Q\} \right\} . \tag{3.22}$$

With an adequate choice of kernel functions $\zeta_i(\boldsymbol{\xi})$, any probability distribution can be approximated as a weighted sum of n of those kernel functions,

$$p(\boldsymbol{\xi}) = \sum_{i=1}^{n} w_i \zeta_i(\boldsymbol{\xi}) \tag{3.23}$$

If the kernels ζ_i are non-negative and scaled such that $\int_{\Xi} \zeta_i(\boldsymbol{\xi}) \, \mathrm{d}\boldsymbol{\xi} = 1$, this parameterisation of the space $\mathcal{P}(\Xi)$ is convenient as it allows writing the problem of estimating

the lower expectation as the linear program

$$\begin{aligned} \underline{\mathrm{E}}(h) \approx \hat{\underline{\mathrm{E}}}(\mathbf{h}_K(h)) = & \min_{\mathbf{w}} \mathbf{h}_K^T(h)\mathbf{w} \\ & s.t. \sum_i w_i = 1 \\ & \mathbf{w} \geq \mathbf{0} \\ & \mathbf{c}^L \leq A\mathbf{w} \leq \mathbf{c}^U \end{aligned} \tag{3.24}$$

where the objective vector and the constraint matrix are defined as $\mathbf{h}_K(h) = \int_\Xi \hat{\mathbf{h}}_K(h, \boldsymbol{\xi})\, \mathrm{d}\boldsymbol{\xi}$ and $A = \int_\Xi \hat{A}(\boldsymbol{\xi})\, \mathrm{d}\boldsymbol{\xi}$, where

$$\hat{\mathbf{h}}_K(h, \boldsymbol{\xi}) = \begin{bmatrix} h(\boldsymbol{\xi})\zeta_1(\boldsymbol{\xi}) \\ \vdots \\ h(\boldsymbol{\xi})\zeta_n(\boldsymbol{\xi}) \end{bmatrix} \tag{3.25}$$

$$\hat{A}(\boldsymbol{\xi}) = \begin{bmatrix} g_1(\boldsymbol{\xi})\zeta_1(\boldsymbol{\xi}) & \dots & g_1(\boldsymbol{\xi})\zeta_n(\boldsymbol{\xi}) \\ \vdots & \vdots & \vdots \\ g_{n_Q}(\boldsymbol{\xi})\zeta_1(\boldsymbol{\xi}) & \dots & g_{n_Q}(\boldsymbol{\xi})\zeta_n(\boldsymbol{\xi}) \end{bmatrix} \tag{3.26}$$

The equality constrain ensures that p integrates to 1, and $\mathbf{w} > \mathbf{0}$ enforces $p > 0$, which together enforce $p \in \mathcal{P}$. The inequality constrains implement the interval valued constrains.

For any linear program given as in Eq. (3.24), the set of solutions $\mathbf{w}$ that minimise the problem also minimise a version of this problem where the inequalities are replaced by an equality matching the value of the optimal solution. The solution to such a problem always includes a basic feasible solution, which is a vector $\mathbf{w}$ with at most $n_Q + 1$ non-zero weights, since there are $n_Q + 1$ linear constraints in the equality version of the problem.

This is the case regardless of what kernels ζ_i are being used, no matter how fine or rough their approximation of $\mathcal{P}$ is. If one considers using kernels that are uniform distributions over connected non-intersecting sets A_i such that $\cup_i A_i = \Xi$, as the approximation is improved by making the measures of those sets go to zero, these kernels

tend towards Dirac delta functions δ, but there is always an optimal $\mathbf{w}$ with at most $n_Q + 1$ non-zero elements. This provides intuition for the fact that the optimal probability distribution p^* for problem (3.22) is always a sum of at most $n_Q + 1$ Dirac deltas,

$$p^*(\boldsymbol{\xi}) = \sum_{j=1}^{n_Q+1} w_j \delta\left(\boldsymbol{\xi} - \boldsymbol{\xi}_c^{(j)}\right) \tag{3.27}$$

In [27], Golodnikov et al. put this in rigorous terms, but for a slightly different problem. They consider problems which are equivalent to finding a lower expectation, as in Eq. (3.2), and with a feasible set identical to Eq. (3.22), except with only upper bounds. It is also shown in [27] that for these problems, the set of optimal solutions always includes a distribution described by a step cumulative distribution function (CDF) with at most $n_Q + 1$ steps, which corresponds to a probability density function (PDF) that is a sum of $n_Q + 1$ Dirac deltas. In this work, the feasible set $\mathcal{Q}$ in Eq. (3.22) is defined with interval valued constraints. It is trivial to write the problem studied in this work into one written with only upper bounds as in [27], but that would result in $2n_Q$ inequalities, suggesting the solution has at most $2n_Q + 1$ Dirac deltas, whereas it still only has at most $n_Q + 1$. Suppose that the set $\mathcal{Q}$ in Eq. (3.22) was defined with only the upper bounds c_i^U, i.e without the c_i^L or equivalently with $c_i^L = -\infty$. In Section 4 of [27], it is shown that all possible values of $\int_\Xi g_i(\boldsymbol{\xi})p(\boldsymbol{\xi})\,\mathrm{d}\boldsymbol{\xi}$ and $\int_\Xi h(\boldsymbol{\xi})p(\boldsymbol{\xi})\,\mathrm{d}\boldsymbol{\xi}$ within such a $\mathcal{Q}$ set can be obtained by a CDF with at most $n_Q + 1$ steps[1]. Adding the lower bounds c_i^L results in a subset, and therefore this conclusion still applies, i.e., any combination of $\int_\Xi g_i(\boldsymbol{\xi})p(\boldsymbol{\xi})\,\mathrm{d}\boldsymbol{\xi}$ and $\int_\Xi h(\boldsymbol{\xi})p(\boldsymbol{\xi})\,\mathrm{d}\boldsymbol{\xi}$ obtainable for distributions p in $\mathcal{Q}$ of Eq. (3.22) can be obtained with a distribution whose PDF is a sum of at most $n_Q + 1$ Dirac deltas.

Because the optimal distribution PDF can be written as a sum of Dirac deltas, the kernels ζ_i are Dirac delta functions, $\zeta_i(\boldsymbol{\xi}) = \delta\left(\boldsymbol{\xi} - \boldsymbol{\xi}_c^{(j)}\right)$, with $N > n_Q + 1$ kernels, and the kernel centres $\boldsymbol{\xi}_c^{(j)}$ are formed as a random sample of Ξ. This sample is written as

[1] This is stated only for the boundary of the set they call "coZ", which is the set containing the values these integrals take for distributions in $\mathcal{Q}$, but the theorem they invoke, Carathéodory's theorem, applies to the entirety of such a set, not just its boundary, and therefore all possible values of those integrals (subject to upper bounds c_i^U) can be obtained by a CDF with at most $n_Q + 1$ steps, not just those in the boundary of coZ

$\mathcal{K}$. This choice has the additional benefit of greatly simplifying the calculation of $\mathbf{h}_K$ and A used in Eq. (3.24),

$$A = \begin{bmatrix} g_1(\mathcal{K}) \\ \vdots \\ g_{n_Q}(\mathcal{K}) \end{bmatrix} \tag{3.28}$$

where $\mathcal{K}$ represents the sample of $\boldsymbol{\xi}_c^{(j)}$ as a n_ξ by N matrix, n_ξ being the dimensionality of Ξ, such that $\Xi \subset \mathbb{R}^{n_\xi}$. The row vectors $g_i(\mathcal{K})$ are composed of the values of g_i applied to the elements of $\mathcal{K}$. The vector $\mathbf{h}_K$ consists simply of the values of $h(\boldsymbol{\xi}_c^{(j)})$.

In [27], the authors also show that it is possible to approximate the lower expectation by solving an equivalent linear program as in Eq. (3.24), for the problem with inequality constraints, as well as for the problem with equality constraints on moments for a 1D space of uncertain variables. These proofs cannot be easily extended to the case of interval valued moment constraints. However, the results in Section 6.1 show empirically that the estimation error decreases with n following a power law.

3.4.1 Moment constraints

One of the applications of this formulation, which is the one for which test cases were obtained, are moment constraints. This work focuses on interval constraints on the first and second order central moments of the distributions of each variable in $\boldsymbol{\xi}_k$, i.e., the means μ_i, and standard deviations σ_i^2 of ξ_i, the i-th scalar variable in vector of variables $\boldsymbol{\xi}_k$,

$$\mu_i = \int_{\Xi_i} \xi_i \, \mathrm{d}\xi_i \tag{3.29}$$

$$\sigma_i^2 = \int_{\Xi_i} (\xi_i - \mu_i)^2 \, \mathrm{d}\xi_i \tag{3.30}$$

and the constraints are written as $\min \mu_i \leq \mu_i \leq \max \mu_i$, and $\min \sigma_i^2 \leq \sigma_i^2 \leq \max \sigma_i^2$.

Following Eq. (3.22), the mean constraints can be written by setting

$$g_i(\boldsymbol{\xi}) = \xi_i \tag{3.31a}$$

$$c_i^L = \min \mu_i \tag{3.31b}$$

$$c_i^U = \max \mu_i \tag{3.31c}$$

$$\text{for } i = 1, \ldots, n_\xi \tag{3.31d}$$

where ξ_i is the i-th element of $\boldsymbol{\xi}$ and n_ξ is the dimensionality of Ξ.

The formulation in Eq. (3.22) cannot exactly convey interval bounds on central moments. For instance, given that the mean is interval valued, one cannot simply write $g_l(\boldsymbol{\xi}) = (\xi_l - \mu_l)^2$ to constrain the variance. The raw moments can easily be bounded, and the relation $\mu_i^{(2)} = \sigma_i^2 + \mu_i^2$ can be used to convert variance bounds into raw moment bounds. These can then be implemented by writing

$$g_i(\boldsymbol{\xi}) = \xi_{i-n_\xi}^2 \tag{3.32a}$$

$$c_i^L = \min \sigma_{i-n_\xi}^2 + \min \mu_{i-n_\xi}^2 \tag{3.32b}$$

$$c_i^U = \max \sigma_{i-n_\xi}^2 + \max \mu_{i-n_\xi}^2 \tag{3.32c}$$

$$\text{for } i = n_\xi, \ldots, 2n_\xi \tag{3.32d}$$

These are conservative bounds, i.e. some distributions that lie within those bounds may have variances outside the original bounds. To be exact, the bounds implemented by using Eqs. (3.32) may include distributions with variances in the interval,

$$\left[\min \sigma_i^2 + \min \mu_i^2 - \max \mu_i^2 \ , \ \max \sigma_i^2 + \max \mu_i^2 - \min \mu_i^2\right] \tag{3.33}$$

Instead of using raw moments, the moments used could be centred on an arbitrary

point $\tilde{\mu}_i$, leading to

$$g_i(\boldsymbol{\xi}) = \left(\xi_{i-n_\xi} - \tilde{\mu}_{i-n_\xi}\right)^2 \tag{3.34a}$$

$$c_i^L = \min \sigma_{i-n_\xi}^2 + \min\left(\mu_{i-n_\xi} - \tilde{\mu}_{i-n_\xi}\right)^2 \tag{3.34b}$$

$$c_i^U = \max \sigma_{i-n_\xi}^2 + \max\left(\mu_{i-n_\xi} - \tilde{\mu}_{i-n_\xi}\right)^2 \tag{3.34c}$$

$$\text{for } i = n_\xi, \ldots, 2n_\xi \tag{3.34d}$$

A similar relation holds, $\mathrm{E}\left[\left(\xi_i - \tilde{\mu}_i\right)^2\right] = \sigma_i^2 + \left(\mu_i - \tilde{\mu}_i\right)^2$, which was used to obtain Eqs. (3.34). The resulting $\mathcal{Q}$ still includes distributions with variances outside the original bounds, but the range of variances that may be present in $\mathcal{Q}$ depends on the value of $\tilde{\mu}_i$,

$$\left[\min \sigma_i^2 + \min(\mu_i - \tilde{\mu}_i)^2 - \max(\mu_i - \tilde{\mu}_i)^2 \ , \ \max \sigma_i^2 + \max(\mu_i - \tilde{\mu}_i)^2 - \min(\mu_i - \tilde{\mu}_i)^2\right] \tag{3.35}$$

The value of the difference $\max(\mu_i - \tilde{\mu}_i)^2 - \min(\mu_i - \tilde{\mu}_i)^2$ quantifies how loose the bounds are. It is minimised when

$$\tilde{\mu}_i = \frac{\max \mu_i + \min \mu_i}{2} \tag{3.36}$$

therefore that is the choice for this value used in this work.

Although the constraints on the moments just described are the only linear constraints of the type in Eq. (3.22) for which results were obtained, other epistemic uncertainty formulations are compatible. Two of these are discussed briefly in the next subsections.

3.4.2 Evidence Theory

Even though the work in this dissertation does not include numerical experiments with evidence theory, also known as Dempster-Shafer Theory [13, 17, 61], it is of interest to relate the previous sections to this theory, both to show that they apply to this theory, but also to set up some ground work for further theoretical results in Section 7.2. In Evidence Theory, for a proposition represented as a set $A \subset \Xi$, and to a set of focal elements $\theta_i \subset \Xi$, a basic probability assignment (BPA) $\mathrm{bpa}(\theta_i)$ is assigned.

Let $\mathcal{T}$ be the set of all focal elements θ_i with non-zero BPA. These are defined such that $\text{bpa}(\theta_i) \geq 0$ and $\sum_{\theta_i \in \mathcal{T}} \text{bpa}(\theta_i) = 1$. The set $\mathcal{T}$, together with the BPAs of the focal elements therein, form a Dempster-Shafer structure. The belief function $B(A)$ is defined as the sum of all $\text{bpa}(\theta_i)$ for the focal elements which are contained in the target set A, i.e.:

$$B(A) = \sum_{\theta_i \in \mathcal{T} \,:\, \theta_i \subseteq A} \text{bpa}(\theta_i) \ . \tag{3.37}$$

The plausibility $Pl(A)$ has a similar definition, being the sum of all $\text{bpa}(\theta_i)$ for the focal elements which intersect the target set, i.e.:

$$Pl(A) = \sum_{\theta_i \in \mathcal{T} \,:\, \theta_i \cap A \neq \varnothing} \text{bpa}(\theta_i) \ . \tag{3.38}$$

Since $Pl(A) = 1 - B(A^C)$, where A^C denotes the complement of set A, only the belief B is discussed henceforth, without loss of generality. Of note is that the entire set Ξ is a possible focal element, and that focal elements can have non-empty intersections.

The following defines the family of distributions $\mathcal{Q}^B$ that allows for a correspondence between $\underline{\text{E}}$ and B:

Definition 3.4.1. If all $\theta_i \in \mathcal{T}$ are such that $\theta_i \cap \theta_j = \varnothing \ \forall i \neq j$, i.e. if all focal elements are disjoint, the following set $\mathcal{Q}^B$ is defined such that $\underline{\text{E}}(\boldsymbol{\xi} \in A)$ is equal to $B(A)$:

$$\mathcal{Q}^B = \left\{ p \in \mathcal{P}(\Xi) : \int_{\theta_i} p(\boldsymbol{\xi}) \text{d}\boldsymbol{\xi} = \text{bpa}(\theta_i) \ \forall \theta_i \in \mathcal{T} \right\} \ . \tag{3.39}$$

Following the formulation in this section, this corresponds to Eq. (3.22) when assigning

$$h(\boldsymbol{\xi}) = \boldsymbol{\xi} \in A \tag{3.40a}$$

$$g_i(\boldsymbol{\xi}) = \boldsymbol{\xi} \in \theta_i \tag{3.40b}$$

$$c_i = c_i^L = c_i^U = \text{bpa}(\theta_i) \ , \tag{3.40c}$$

where h is the indicator function for set A, each constraint g_i is the indicator function for focal element θ_i, and $c_i = c_i^L = c_i^U$ contain the BPAs.

The set of optimal distributions for problem (3.2) with this set $\mathcal{Q}^B$ includes distributions whose PDF is a sum of Dirac delta functions with one Dirac delta centred in each focal element θ_i with weight $\text{bpa}(\theta_i)$, and located in $\theta_i \setminus A$, if this region is not empty, or anywhere in θ_i otherwise.

Finally, let us consider the inclusion in $\mathcal{T}$ of focal elements which intersect each other. One issue with using (3.39) arises when one considers distributions with probability assigned to those intersections. For instance, if θ_i and θ_j are intersecting focal elements each with a BPA of 0.5, a probability distribution such that $p(\theta_i \cap \theta_j) = 0.5$ and $p(\theta_i \setminus \theta_j) = p(\theta_j \setminus \theta_i) = 0$ would meet the constraints in Eq. (3.39) and so $\underline{\mathrm{E}}(\theta_i \cup \theta_j) = 0.5$, but since those two sets contain all BPAs, one must have that $B(\theta_i \cup \theta_j) = 1$.

One way of interpreting the formalism in Evidence theory is that each BPA must introduce additional probability mass "somewhere" in its focal element, while the definition in Eq. (3.39) would allow the same probability mass to contribute to the BPA of two different focal elements. To resolve this, one must ensure that all BPAs are assigned somewhere, and separately, by verifying also that the unions of focal elements contain the sums of those elements' BPAs. Let $\mathcal{U}$ be the set of all sets that result from the union of any number of focal elements in $\mathcal{T}$, in addition to the focal elements themselves. The set $\mathcal{Q}^{B'}$ that results in $\underline{\mathrm{E}}(\boldsymbol{\xi} \in A)$ being equal to $B(A)$ in this more general case is

$$\mathcal{Q}^{B'} = \left\{ p \in \mathcal{P}(\Xi) : \int_{\alpha_j} p(\boldsymbol{\xi}) \mathrm{d}\boldsymbol{\xi} \geq \sum_{\theta_i \in \mathcal{T} \,:\, \theta_i \subseteq \alpha_j} \text{bpa}(\theta_j) \;\forall \alpha_j \in \mathcal{U} \right\} , \tag{3.41}$$

which simplifies to Eq. (3.39) when all elements of $\mathcal{T}$ are disjoint. Notice that this set is now expressed with non-strict inequalities as opposed to equalities. This can also be written by choosing the correct g_i, c_i^L and c_i^U. This formulation ensures that all BPAs are actually assigned for the distributions in $\mathcal{Q}_B$. For instance, in the previous example, the distribution would not meet the constraint that $p(\theta_i \cup \theta_j) \geq 0.5 + 0.5 = 1$, which is part of Eq. (3.41).

In general, unless every focal element intersects every other focal element, some of the inequalities in Eq. (3.41) are redundant. For example, if a set $\alpha_i \in \mathcal{U}$ is

obtained from the union of two disjoint non-empty elements of $\mathcal{U}$, then the inequality corresponding to α_i is redundant.

This section was intended only as a very brief introduction to Evidence Theory, focused solely on showing how it relates to the formulation used in this work. For more information, the interested reader can consult [13, 17, 24, 50, 61]. The issue of defining the belief of a set $A \subset \Xi_1 \times \Xi_2$, for variables $\boldsymbol{\xi}_1 \in \Xi_1$ and $\boldsymbol{\xi}_2 \in \Xi_2$, each with their own Dempster-Shafer structures, is discussed in Section 7.2 of the next chapter.

3.4.3 P-Boxes

A different, commonly used method for representing epistemic uncertainty are P-Boxes [24]. P-Boxes can be described parametrically as shown in section 3.3, but they are often defined by placing lower and upper bounds on the CDF, $\underline{F}$ and $\overline{F}$ respectively. A 1D P-Box can be written as

$$\mathcal{Q}_{\text{PBox}} = \left\{ p \in \mathcal{P}(\Xi) : \underline{F}(x) \leq \int_{-\infty}^{x} p(\xi)\,\mathrm{d}\xi \leq \overline{F}(x)\ \forall x \in \Xi \right\} . \tag{3.42}$$

A 1D P-Box can be converted into a Dempster-Shafer structure by discretising it and creating a focal element for each successive pair of discretisation points, see [24] for details. The discretised P-Box can be written as

$$\mathcal{Q}_{\text{PBox}} = \left\{ p \in \mathcal{P}(\Xi) : \int_{-\infty}^{\underline{x}_i} p(\xi)\,\mathrm{d}\xi \geq \underline{F}(\underline{x}_i) \text{ and } \int_{-\infty}^{\overline{x}_j} p(\xi)\,\mathrm{d}\xi \leq \overline{F}(\overline{x}_j)\ \forall i,j \right\} . \tag{3.43}$$

Whichever discretisation is used, it is possible to obtain an equivalent one where the points $\underline{x}_i$ and $\overline{x}_i$ are such that $\underline{x}_i < \overline{x}_i$, $\underline{x}_1 \leq \ldots \leq \underline{x}_{n_\theta}$, $\overline{x}_1 \leq \ldots \leq \overline{x}_{n_\theta}$, and in addition $\overline{F}(\overline{x}_i) = \underline{F}(\underline{x}_{i+1})$, as described in [24].

The results in [24] provide a method for converting P-boxes into near-equivalent Dempster-Shafer structures, i.e., a set of focal elements and respective BPAs, obtained by discretising the P-Boxes. This shows that formulation in Eq. (3.22) describes many commonly used strategies for epistemic uncertainty quantification.

3.5 Threshold Mapping

The search space for the optimisation problems in Eqs. (2.28) and (2.37) are defined, for convenience, as hyper-rectangles. This includes the set of thresholds $\mathcal{N}$. This way, many function evaluations of the lower expectations in those problems performed while running MACS might be wasted because they correspond to two plateaus, one with lower expectation equal to 0 and one with lower expectation equal to 1, as illustrated in Fig. 3.2. Furthermore, the ridges of these two plateaus correspond to a sharp change in the value of the lower expectation (see solid and dashed lines in Fig. 3.2), which is undesirable if one uses a continuous function to build a surrogate model.

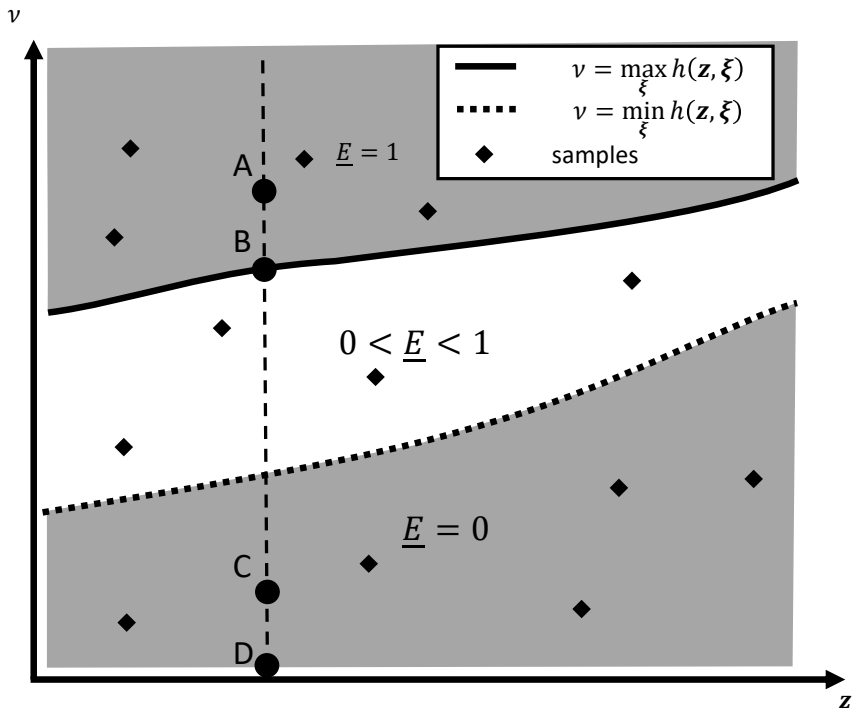


Figure 3.2: A diagram representing the search space - the vertical axis represents the threshold value ν, while the horizontal axis represents the proxy control variable $\mathbf{z}$. When using a hyper-rectangular sample region, many of the samples fall into the regions where the lower expectation is either 0 or 1, which are shaded in this diagram. These regions are not useful. All points in the upper shaded region are dominated by points on the solid black line, for example point A is dominated by B, while all points on the lower shaded region are dominated by making ν the smallest value possible, for example, point C is dominated by point D. Furthermore, in this lower shaded region the lower expectation equals zero, meaning that meeting the constraint is impossible.

For this purpose, a threshold mapping strategy, akin to the control maps in Section 2.7, is proposed to focus the search only in regions where the lower expectation is in the open interval $(0, 1)$. That is, a proxy threshold variable ν^* is introduced such that

$$0 < \underline{\mathrm{E}}(\mathbf{z}, \boldsymbol{\nu}(\boldsymbol{\nu}^*, \mathbf{z})) < 1 \ . \tag{3.44}$$

Let $h(\mathbf{z}, \boldsymbol{\xi})$ be the quantity of interest in the definition of the lower expectation, as explained in Section 3.2, with $\underline{h}(\mathbf{z}) = \min_{\boldsymbol{\xi}\in\Xi} h(\mathbf{z}, \boldsymbol{\xi})$ and $\overline{h}(\mathbf{z}) = \max_{\boldsymbol{\xi}\in\Xi} h(\mathbf{z}, \boldsymbol{\xi})$. Eq. (3.44) implies $\underline{h} < \nu < \overline{h}$, which can be trivially satisfied with the following mapping:

$$\nu(\nu^*, \mathbf{z}) = \underline{h}(\mathbf{z}) + \left(\overline{h}(\mathbf{z}) - \underline{h}(\mathbf{z})\right) \nu^* \ , \quad \nu^* \in [0, 1] \ . \tag{3.45}$$

This mapping can be used only to obtain the samples used to train the surrogate model of the lower expectation, or it can be called throughout the optimisation with MACS. In the following, the former use of the mapping will be called *thMap-dist* and the latter use of the mapping will be called *thMap-whole*.

With *thMap-dist*, a set of points $\mathbf{z} \in \mathbf{Z}$ and $\boldsymbol{\nu}^* \in [0, 1]^{n_s}$ are sampled using Latin hypercube sampling from a uniform distribution (LHSU) [47], where n_s is the number of quantities of interest h. Note that they are both sampled in a hyper-rectangular region. Both the control map and the threshold map are applied to each of these points, and the lower expectations $\underline{\mathrm{E}}$ are calculated. The surrogate model on $\underline{\mathrm{E}}$ is then trained in the same way as if there was no threshold map, on the values of $\mathbf{z}$ and the mapped values of the thresholds ν, so that $\tilde{\underline{\mathrm{E}}}(\mathbf{z}, \nu) \approx \underline{\mathrm{E}}(\mathbf{z}, \nu)$ as before.

With *thMap-whole*, the surrogate model is trained on $\mathbf{z}$ and ν^*, so that $\tilde{\underline{\mathrm{E}}}(\mathbf{z}, \nu^*) \approx \underline{\mathrm{E}}(\mathbf{z}, \nu(\nu^*, \mathbf{z}))$. Therefore the search space for MACS is $Z \times [0, 1]^{n_s}$. Since the actual quantity to be minimised is ν and not ν^*, the *thMap-whole* option requires the additional surrogate model $\tilde{\nu}(\nu^*, z) \approx \nu(\nu^*, z)$.

The method *thMap-dist* has the disadvantage of running MACS on a set that is not contained within the set used to train the surrogate model. On the other hand, the method *thMap-whole* has the disadvantage of doubling the amount of surrogate models needed, which can increase computational complexity as well as adding inaccuracy to

the expectation estimation. However, the value ν^*, which indicates where the threshold is between minimum and maximum of the quantity of interest, may be a better predictor of the lower expectation than the threshold itself.

In Section 5.4, these methods are compared with each other as well as with the algorithm without any threshold map.

3.6 Overall Algorithm

The method proposed in this work was implemented in Algorithm 1 corresponding to the flowchart in Fig. 4.4.

The algorithm starts by generating sets $\mathcal{Z}$ and $\mathcal{N}$ of samples for the proxy control variable $\mathbf{z}$ and threshold array $\boldsymbol{\nu}$, line 1 of Algorithm 1, with Latin hypercube sampling from a uniform distribution. For each sample points $\mathbf{z}$ and $\boldsymbol{\nu}$ in sets $\mathcal{Z}$ and $\mathcal{N}$, line 3, the following process is applied. The control law $\mathbf{y}$ is calculated from $\mathbf{z}$, as described in Section 2.7. From the control law $\mathbf{y}$ and thresholds $\boldsymbol{\nu}$, the lower expectations $\underline{\mathrm{E}}$ are calculated, at line 5, using Algorithm 2. The values of $\underline{\mathrm{E}}$ are then added to set $\mathcal{E}$, at line 6. The values in $\mathcal{Z}$, $\mathcal{N}$ and $\mathcal{E}$ are then used, at line 6, to train the surrogate models $\underline{\tilde{\mathrm{E}}}$ for each of the expectation functions.

If the threshold map *thMap-dist* is used, in Line 1 the LHSU generates the set $\mathcal{N}^*$ of sample vectors $\boldsymbol{\nu}^*$. These samples are then converted to $\boldsymbol{\nu}$ with Eq. (3.45) before being added to $\mathcal{N}$ to calculate the surrogate of the lower expectation in Line 5. If, instead, *thMap-whole* is used, the surrogate of the lower expectation is built directly on the set $\mathcal{N}^*$ and an additional surrogate model $\tilde{\boldsymbol{\nu}}(\boldsymbol{\nu}^*, \mathbf{z}) \approx \boldsymbol{\nu}(\boldsymbol{\nu}^*, \mathbf{z})$ is trained.

The loop starting at line 9 is the core of the algorithm: MACS is repeatedly run for $N_{\mathrm{re-train}}$ subsequent iterations. This part differs depending on the threshold map in use. If no threshold map is in use, or if *thMap-dist* is in use, only the lower expectation needs to be approximated by the surrogate model, and as such MACS is solving

$$\min_{\mathbf{z},\boldsymbol{\nu}} \left[-\underline{\tilde{\mathrm{E}}}(\mathbf{z}, \boldsymbol{\nu}), \boldsymbol{\nu} \right] , \tag{3.46}$$

where the surrogate $\underline{\tilde{\mathrm{E}}}(\mathbf{z}, \boldsymbol{\nu})$ approximates $[\underline{\tilde{\mathrm{E}}}(h_1(\mathbf{f}_{\mathrm{CMap}}(\mathbf{z})) < \nu_1), \ldots]$. Otherwise, if

thMap-whole is in use, the threshold map becomes a part of the objective function, and must thus also be approximated by a surrogate model,

$$\min_{\mathbf{z},\boldsymbol{\nu}^*} \left[-\tilde{\underline{\mathrm{E}}}(\mathbf{z}, \boldsymbol{\nu}^*), \tilde{\boldsymbol{\nu}}(\mathbf{z}, \boldsymbol{\nu}^*) \right] \ , \tag{3.47}$$

where one has the additional surrogate model $\tilde{\nu}(\nu^*, z) \approx \nu(\nu^*, z)$ of the threshold map.

At each iteration MACS evaluates the surrogate models n_f times to find a set of vectors $\mathcal{Z}_{\mathrm{MACS}}$ and $\mathcal{N}_{\mathrm{MACS}}$ that approximate the global Pareto front (line 10). Using the same archival strategy used in MACS [56], and briefly described in Section 2.6, at line 11, N_s solutions are selected and added to $\mathcal{Z}$, $\mathcal{N}$ respectively, at lines 12 and 13. At lines 14-18 the lower expectations $\underline{\mathrm{E}}$ corresponding to the solutions in $\mathcal{Z}_{\mathrm{MACS}}$ and $\mathcal{N}_{\mathrm{MACS}}$ are calculated with Algorithm 2 and added to $\mathcal{E}$. The surrogate model is then re-trained at line 20, by applying the same training process used to train the surrogate at the start, which is described in [49], but with the newly updated sets $\mathcal{Z}$, $\mathcal{N}$ and $\mathcal{E}$.

After $N_{\mathrm{re-train}}$ iterations, this algorithm will have created sets $\mathcal{Z}$, $\mathcal{N}$ and $\mathcal{E}$ which include the initial sample used to initialise the surrogate model, as well as the N_s solutions coming from each run of MACS. The last step at line 22 filters out all dominated solutions from $\mathcal{Z}$, $\mathcal{N}$ and $\mathcal{E}$ to obtain the final output sets $\mathcal{Z}_f$, $\mathcal{N}_f$ and $\mathcal{E}_f$. This is done, instead of just retaining the solutions found at the last iteration, because many of the solutions found at each iteration may not be dominated by the ones found at subsequent iterations.

Algorithm 2 implements the computation of the lower expectation. For each stochastic variable (line 1), line 2 greedily initialises the index vector $\mathbf{j}$ according to Eq. (3.21). At line 3, the pattern search is run starting from this initial point. Then, from the solution of the pattern search, a new index vector is generated at line 4, and at line 5 the pattern search is run again from this new starting point. At line 6, the multi-index corresponding to the smallest lower expectation is selected.

Since part of the computational cost comes from propagating a given control law corresponding to a given sample point, and the same sample points $\boldsymbol{\xi}$ are used for each quantity of interest, whenever the expected value is calculated for a specific quantity $h_i < \nu_i$, and for a specific distribution p, proxy control variables $\mathbf{z}$, and thresholds $\boldsymbol{\nu}$,

it is also calculated for all other quantities $h_j < \nu_j \ \forall j$, and these results are stored for possible future use, an example of memoisation.

The number of initial training points N_{init}, number of agents N_p, and archive size N_a for MACS, as well as the number of points trained from each iteration N_s can all be changed to accommodate the need for accuracy versus keeping the execution time low. It is possible to start this method with a small amount of points, and then add more, if a more accurate representation of the Pareto front is desired.

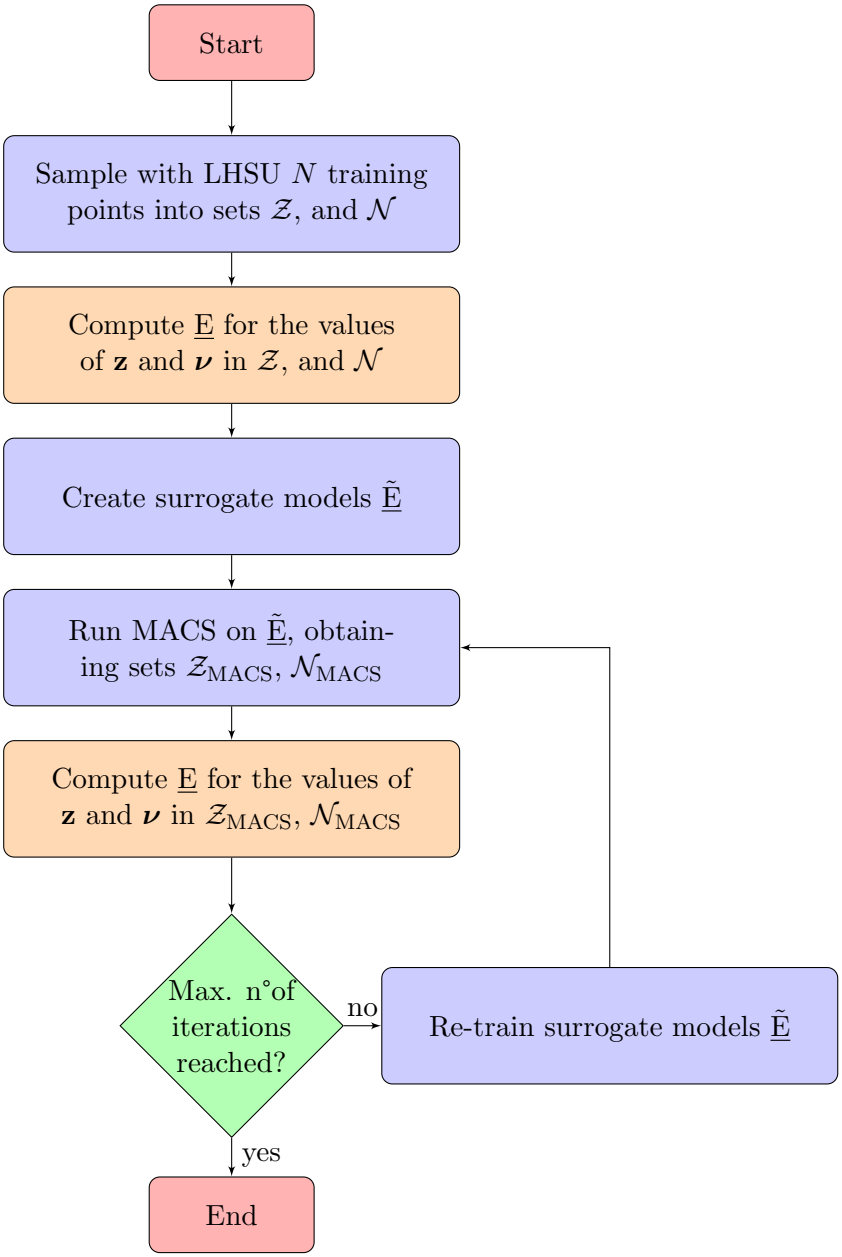


Figure 3.3: Flowchart describing the overall process used in this work. The orange blocks represent expensive computations of the control map and lower expectation. The same process is described with more detail in Algorithms 1 and 2.

Algorithm 1: Many-Objective Optimisation Under Epistemic Uncertainty

Output: Samples from the Pareto Front contained in $\mathcal{Z}_f$ and $\mathcal{E}_f$

$\mathcal{Z}, \mathcal{N} \leftarrow$ `getLhsuSamples()`
for $i = 1$ *to* N_{init} **do**
  $\mathbf{z}, \boldsymbol{\nu} \leftarrow \mathcal{Z}_i, \mathcal{N}_i$
  $\mathbf{y} \leftarrow \mathbf{f}_{\text{CMap}}(\mathbf{z})$
  $\underline{\mathbf{E}} \leftarrow$ `LowerExp(`$\mathbf{h}_K, \boldsymbol{\nu}$`)` `// See Algorithm 2`
  $\mathcal{E}$`.append(`$\underline{\mathbf{E}}$`)`
end
model $\leftarrow$ `DACEfit(`$\mathcal{Z}$,$\mathcal{N}$`)` `// From the DACE library [49]`
for $i = 1$ **to** $N_{\text{re-train}}$ **do**
  $\mathcal{Z}_{\text{MACS}}, \mathcal{N}_{\text{MACS}} \leftarrow$ `MACS(model.predictor)// See [56]`
  $\mathcal{Z}_{\text{MACS}}, \mathcal{N}_{\text{MACS}} \leftarrow$ `Select_`N_s`(`$\mathcal{Z}_{\text{MACS}}$, $\mathcal{N}_{\text{MACS}}$`)`
  $\mathcal{Z} \leftarrow$ `Join(`$\mathcal{Z}$, $\mathcal{Z}_{\text{MACS}}$`)`
  $\mathcal{N} \leftarrow$ `Join(`$\mathcal{N}$, $\mathcal{N}_{\text{MACS}}$`)`
  for $\mathbf{z}$, $\boldsymbol{\nu}$ *in* $\mathcal{Z}_{\text{MACS}}$, $\mathcal{N}_{\text{MACS}}$ **do**
    $[\mathbf{r}_i, \mathbf{v}_i, \boldsymbol{\xi}] \leftarrow$ `splitZ` $(\mathbf{z})$
    $\mathbf{u} \leftarrow \mathbf{f}_{\text{CMap}}([\mathbf{r}_i, \mathbf{v}_i, \boldsymbol{\xi}])$
    $\underline{\mathbf{E}} \leftarrow$ `LowerExp(`$\mathbf{u}, \boldsymbol{\nu}$`)` `// See Algorithm 2`
    $\mathcal{E}$`.append(`$\underline{\mathbf{E}}$`)`
  end
  model $\leftarrow$ `DACEfit` $(\mathcal{Z}, \mathcal{E})$ `// From the DACE library [49]`
end
$(\mathcal{Z}_f, \mathcal{E}_f) \leftarrow$ `RemoveDominated` $(\mathcal{Z}, \mathcal{E})$

3.7 Conclusion

This chapter introduced a formulation of epistemic uncertainty which differs from standard probability theory. Conceptually, this can be summed up as characterising the uncertainty in a variable not with a probability distribution, but with a family of distributions. The counterpart to the expected value in standard probability theory is the lower expectation. This quantity is used to quantify the uncertainty in the quantities of interest, and was used in the previous chapter to formulate the many objective optimisation problem under uncertainty. The lower expectation was also used to define the upper quantile, which is the counterpart to the quantile in standard probability theory, and will also be used to formulate variants of the many-objective optimisation problem under uncertainty.

Algorithm 2: Lower expectation

Input : A control law $\mathbf{y}$ and a vector of thresholds $\boldsymbol{\nu}$
Output: Vector of lower expectations $\underline{\mathbf{E}}$
// Each $\underline{\mathbf{E}}_i$ is the lower expectation of $I_i(\boldsymbol{\xi}) = h_i(\boldsymbol{\xi}; \mathbf{y}) < \nu_i$
for $i = 1$ **to** n_s **do**
  $\mathbf{j}^0 \leftarrow$ GreedyInitialisation($I_i(\boldsymbol{\xi})$) // See Eq. (3.20)
  $\mathbf{j}^{f_1} \leftarrow$ PatternSearch($\mathbf{j}^0, I_i(\boldsymbol{\xi})$) // See Section 3.3.2
  // Re-start
  forall k **do** $\mathbf{j}_k^{f_1+1} \leftarrow q_k - \mathbf{j}_k^{f_1}$ // See Eq. (3.21)
  $\mathbf{j}^{f_2} \leftarrow$ PatternSearch($\mathbf{j}^{f_1+1}, I_i(\boldsymbol{\xi})$) // See Section 3.3.2
  $\mathbf{j}^{\omega_2} \leftarrow \operatorname{argmin}_{\mathbf{j} \in \{\mathbf{j}^{f_1},\, \mathbf{j}^{f_2}\}} \mathrm{E}(\mathbf{j})$
  $\underline{\mathbf{E}}_i \leftarrow \mathrm{E}(I_i(\boldsymbol{\xi}), \mathbf{j}^{\omega_2})$
end

Different approaches for formulating the family of distributions were also presented, alongside methods for computing the lower expectation when these are used to define the epistemic uncertainty. This included two types of families, one based on linear combinations of basis polynomials, and another based on applying linear constraints on the distributions. For both cases, algorithms are proposed with efficiency in mind. It is worth also noting that the linear constraints based families of distributions can be related to many common formulations of epistemic uncertainty found in the literature, such as evidence theory. As such, some of the theoretical results obtained here and in the next chapter are applicable beyond the cases covered by the applications in Part II.

A separate but related development presented in this chapter were the threshold maps, which, when optimising the lower expectation of an indicator function of the form $h(\boldsymbol{\xi}) < \nu$ alongside the threshold ν, can be used to increase the efficiency of the search algorithm by focusing the search on more useful lower-dimensional subsets of the search space.

The chapter finalised by combining the topics herein introduced with those of Chapter 2 to create a many-objective optimisation algorithm that can produce solutions that are robust to epistemic uncertainty.

Chapter 4

Dynamic Programming Approach to Epistemic Uncertainty Quantification

This Chapter proposes a novel method for quantifying epistemic uncertainty that is amenable to trajectories subject to uncertainty in system parameters, which is formulated using uncertain variables that affect separate segments of the trajectory, such as when the evolution of the thrust and specific impulse of a spacecraft vary in an epistemically uncertain way. This approach is based on the technique of dynamic programming, and in addition to allowing this separation of variables, since it is a multi-segment approach, it is directly applicable to feedback control problems, where the control law is a function of sparse observations.

4.1 Related Works

In the literature on epistemic uncertainty, it is common for the distributions of each uncertain variable to be constrained separately, without consideration for the full joint distribution. In [27], the distributions of each variable are constrained by their moments, but no constraints are applied to the joint distribution. As another example, in [13], using evidence theory, focal elements from different variables are combined

to produce focal elements on the full vector of uncertain variables $\boldsymbol{\xi}$ by means of a Cartesian product, resulting not only in exponential computational cost, but also some hidden assumptions on the type of distributions. The latter example was not the focus of this work, but some discussion is present in Section 7.2, as suggestions of future work.

In addition to the novel dynamic programming based approach, this chapter also aims to examine different ways to extend families of distributions of individual variables to a combination of variables. In fact, this new approach can be seen as an alternative to these. This chapter also gives inequalities for the resulting lower expectations following different formulations, including the novel dynamic programming based method.

Dynamic programming is an approach, introduced by Richard Bellman in [5], for solving "segmented" optimisation problems. In these problems, at each segment there is a state variable that evolves according to decisions that are made at the start of the segment. The problem is formulated in such a way that only the state and the decision at the start of a segment affect the state at its end, which is also the state at the start of the next segment. In other words, the full history before the segment does not matter to the outcome of this decision. Additionally, the objective in such problems is to minimise or maximise some function of the state at the end of the last segment, which can also allow for cost functions that depend directly on the decisions taken at each segment. [5] For such problems, Bellman's Principle of Optimality [5] applies, which relates the solution to the original problem to the solutions to the sub-problems starting at each segment. This allows to solve the original problem by iteratively solving these sub-problems, which can significantly reduce the computational complexity of an optimisation problem. The exact procedure depends on the specifics, since dynamic programming is best seen as a paradigm than a specific algorithm.

When these principles are applied to continuous trajectory optimisation problems, the differential dynamic programming (DDP) algorithm, introduced by Meyer in [46] among other works, is used. A basic concept in DDP is the cost-to-go, which represents the cost for the sub-trajectory starting at a given segment assuming optimal decisions are taken at the next segments. Due to Bellman's Principle of Optimality, the decisions

that form the solution to the original problem will also optimise the cost-to-go for each segment for a particular state. Given the infinite nature of the state and decision spaces in trajectory optimisation problems, in DDP a local quadratic approximation of the cost-to-go function is made at the start of each segment. [46] A DDP algorithm is generally composed of a forward pass obtaining a reference trajectory and forming a quadratic approximation around this reference trajectory, followed by a backward pass that iteratively optimises each sub-problem. The solution to each sub-problem, and the corresponding cost-to-go, are also approximated as quadratic functions of the initial state for that sub-problem, and passed to the previous segment during the backward pass. The resulting algorithm is similar (but different) to the iterative linear-quadratic regulator. [64]

By breaking up the problem into sub-problems in this way, DDP tackles the so called "curse of dimensionality" (a term also introduced by Bellman in [5]). The optimisation problem has a number of decision variables proportional to the number of segments, and most non-linear programming (NLP) solvers have quadratic complexity with this number. DDP breaks up the high dimensional original problem into a sequence of lower dimensional ones, and it has been shown to require a computational effort that is linear, instead of quadratic, with the number of segments. [41, 42]

The theory section of [41], which is freely available online as of the writing of this thesis, provides a more complete and detailed history and literature review of DDP. The same reference [41] also proposes a variant of DDP, hybrid differential dynamic programming, which incorporates certain constrained optimisation techniques into the traditional DDP algorithm.

The DDP algorithm has also been applied to optimal control under (aleatory) uncertainty in the stochastic dynamic programming algorithm introduced by Ozaki et al. [53] that was mentioned in Chapter 1. In [53], the unscented transform [36] is used to write a cost function that estimates the expected value of the objective, while being amenable optimisation with DDP. In addition, the authors optimise a feedback control law with this method. To the best of our knowledge, all dynamic programming formulations for optimisation under uncertainty consider only aleatory uncertainty, with the work

done in this PhD being the first to apply dynamic programming to the quantification of epistemic uncertainty.

One key point to note is that in this work, dynamic programming is not being used to optimise the control law directly, but to compute the lower expectation, which is also defined as an optimisation problem where the decision variables are probability distributions. While DDP is often used to obtain feedback control laws, and we are using a dynamic programming formulation partly to more easily calculate the lower expectation when there is feedback control, the reader should not interpret that we are applying DDP, obtaining a feedback control law directly using dynamic programming, or that there is any relation between our approach and DDP other than the general dynamic programming structure, and a few other similarities such as the existence of forward and backward passes as will be described.

A variant of our dynamic programming formulation that also obtains a feedback control policy was considered but not fully pursued due to lack of time. It is, however, considered worthwhile to pursue as future work, and what theoretical developments were obtained are shown in Section 7.1.

4.2 Dynamic Programming Formulation

With dynamic programming, the propagation of a trajectory is split into M segments, each affected by a control law $\mathbf{u}_k$ and uncertain parameter $\boldsymbol{\xi}_k$ arrays, as exemplified in Fig. 4.1. The following state transition function is defined:

$$\mathbf{x}_{k+1} = \Phi_k(\mathbf{x}_k, \mathbf{u}_k, \boldsymbol{\xi}_k) \tag{4.1}$$

The control laws $\mathbf{u}_k$ can be seen, for maximum generality, as functions of time and state $\mathbf{u}_k(\mathbf{x}_k, t)$ for, $t \in [t_k, t_{k+1}]$, and each $\boldsymbol{\xi}_k$ is an array with n_k elements. The control law and uncertainty vector are also written for multiple segments of the trajectory as $\mathbf{u}_{k:j} = [\mathbf{u}_k, \ldots, \mathbf{u}_j]$ and $\boldsymbol{\xi}_{j:k} = [\boldsymbol{\xi}_j, \ldots, \boldsymbol{\xi}_k]$, respectively, and $\mathbf{u} = \mathbf{u}_{1:M}$, $\boldsymbol{\xi} = \boldsymbol{\xi}_{1:M}$. The probability distributions over $\boldsymbol{\xi}_{j:k}$ will be written $p_{j:k}$ but the subscript will be dropped whenever this does not create confusion.

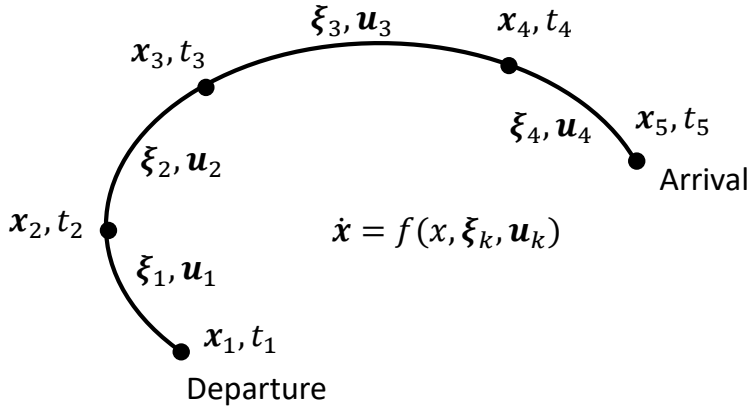


Figure 4.1: Example of a trajectory split into segments for the computation of V_k, with $M = 4$

The quantity of interest, whose lower expectation one wants to estimate, can also be written as

$$h(\mathbf{u}, \boldsymbol{\xi}) = L(\mathbf{x}_{M+1}) + \sum_{k=1}^{M} l_k(\mathbf{x}_k, \mathbf{u}_k, \boldsymbol{\xi}_k) \tag{4.2}$$

where $L(\mathbf{x}_{M+1})$ is a function of the final states, while $l_k(\mathbf{x}_k, \mathbf{u}_k, \boldsymbol{\xi}_k)$ represent path costs. In this section we pursue a method for separating the computation of the lower expectation into separate segments, analogously to what is done with the optimization of a control law in differential dynamic programming [53].

4.3 Segmented Epistemic Uncertainty Definitions

Let $\hat{\mathcal{Q}}_k$ be the sets of distributions that are known to include the real distributions of the uncertain variables over each segment, $\boldsymbol{\xi}_k$. These sets represent the expert information that is available but insufficient to specify a single distribution. The set $\mathcal{Q}$ is defined as the family of distributions over all uncertain variables $\boldsymbol{\xi}$ resulting from such information, without any additional information on joint probability distributions of variables in different segments. The corresponding lower expectation is $\underline{\mathrm{E}}_d$. The following definition formalises this:

Definition 4.3.1. If information is only provided about the distributions of the un-

certain variables over each segment $\boldsymbol{\xi}_k$, of the form $p(\boldsymbol{\xi}_k) \in \hat{\mathcal{Q}}_k$, and nothing is known about the joint distributions of variables corresponding to different segments, one has $p_{m:M}(\boldsymbol{\xi}_{m:M}) \in \mathcal{Q}_{m:M}$, with

$$\mathcal{Q}_{m:M} = \left\{ p \in \mathcal{P}(\Xi_{m:M}) : \mathcal{M}_k[p](\boldsymbol{\xi}_k) \in \hat{\mathcal{Q}}_k \ , k \in \{m, \dots, M\} \right\} \tag{4.3}$$

where $\mathcal{M}_k[p]$ is the marginal distribution of p over $\boldsymbol{\xi}_k$, i.e.,

$$\mathcal{M}_k[p](\boldsymbol{\xi}_k) = \int_{\Xi_1} \dots \int_{\Xi_{k-1}} \int_{\Xi_{k+1}} \dots \int_{\Xi_M} p(\boldsymbol{\xi}) \mathrm{d}\boldsymbol{\xi}_1 \dots \mathrm{d}\boldsymbol{\xi}_{k-1} \mathrm{d}\boldsymbol{\xi}_{k+1} \dots \mathrm{d}\boldsymbol{\xi}_M \tag{4.4}$$

The lower expectation defined with respect to the family of distributions $\mathcal{Q}_{k:M}$ in Eq. (4.3) is written as $\underline{\mathrm{E}}_k^d$,

$$\underline{\mathrm{E}}_k^d(h) = \min_{p \in \mathcal{Q}_{k:M}} \int_{\Xi_{k:M}} h(\boldsymbol{\xi}_{k:M}) p(\boldsymbol{\xi}_{k:M}) \, \mathrm{d}\boldsymbol{\xi}_{k:M} \tag{4.5}$$

Furthermore, we write $\mathcal{Q} = \mathcal{Q}_{1:M}$ and $\underline{\mathrm{E}}^d = \underline{\mathrm{E}}_1^d$.

The set $\mathcal{Q}_{k:M}$ can be seen as the most conservative way to obtain information on the distributions on the variables $\boldsymbol{\xi}_{k:M}$ based on the information given by $\hat{\mathcal{Q}}_i$ for $k \leq i \leq M$, as no additional information is assumed besides the fact that the marginals over segment $\boldsymbol{\xi}_k$ are in $\hat{\mathcal{Q}}_k$. Adding to $p \in \mathcal{Q}_{k:M}$ the constraint that the variables $\boldsymbol{\xi}_k$ are independent results in the family of distributions in the following definition:

Definition 4.3.2. If, in addition to knowing that $p(\boldsymbol{\xi}_k) \in \hat{\mathcal{Q}}_k$, one also knows that each pair of variables $\boldsymbol{\xi}_i$ and $\boldsymbol{\xi}_j$ for $i \neq j$ is independent[1], one can write that $p(\boldsymbol{\xi}_j, \boldsymbol{\xi}_k) = p(\boldsymbol{\xi}_j) p(\boldsymbol{\xi}_k) \forall j \neq k$, leading to the following family of distributions $p(\boldsymbol{\xi}_{k:M})$,

$$\hat{\mathcal{Q}}_{k:M} = \left\{ p_k(\boldsymbol{\xi}_k) \times \dots \times p_M(\boldsymbol{\xi}_M) : p_k(\boldsymbol{\xi}_k) \in \hat{\mathcal{Q}}_k, \dots, p_M(\boldsymbol{\xi}_M) \in \hat{\mathcal{Q}}_M \right\} \tag{4.6}$$

composed of all possible products of distributions in each set $\hat{\mathcal{Q}}_k$, which can also be

[1] Independence here refers to "physical independence", which is the definition of independence used in standard probability theory, as opposed to "epistemic independence" [71] which is a topic outside the scope of this work.

written as

$$\hat{\mathcal{Q}}_{k:M} = \hat{\mathcal{Q}}_k \times \hat{\mathcal{Q}}_{k+1} \times \cdots \times \hat{\mathcal{Q}}_M \tag{4.7}$$

As before, we write $\hat{\mathcal{Q}} = \hat{\mathcal{Q}}_{1:M}$. The lower expectation defined with this family of distributions is referred to as $\underline{\mathrm{E}}^i_k$, in Eq. (4.8), with $\underline{\mathrm{E}}^i = \underline{\mathrm{E}}^i_1$. Note that when it is stated that "variables $\boldsymbol{\xi}_k$ are independent from each other", it is meant that $p(\boldsymbol{\xi}_j, \boldsymbol{\xi}_k) = p(\boldsymbol{\xi}_j)p(\boldsymbol{\xi}_k) \forall j \neq k$, but remembering that $\boldsymbol{\xi}_k$ are vectors, the variables within those vectors can be dependent on each other. Let $\underline{\mathrm{E}}^i$ be the lower expectation with respect to $\hat{\mathcal{Q}}$,

$$\underline{\mathrm{E}}^i_k(h) = \min_{p \in \hat{\mathcal{Q}}_{k:M}} \int_{\Xi_{k:M}} h(\boldsymbol{\xi}_{k:M}) p(\boldsymbol{\xi}_{k:M}) \, \mathrm{d}\boldsymbol{\xi}_{k:M} \tag{4.8}$$

Before considering the full implications of the definitions 4.3.1 and 4.3.2, let us briefly look at the simpler case where the marginal distributions are as unconstrained as possible. In that case, the sets $\hat{\mathcal{Q}}_k$ would equal the sets of all possible distributions over Ξ_k, i.e. $\mathcal{P}(\Xi_k)$. The lower expectation of some function $h(\xi_k)$ over such a set is simply the minimum,

$$\min_{p \in \mathcal{P}(\Xi_k)} \int_{\Xi_k} p(\boldsymbol{\xi}_k) h(\boldsymbol{\xi}_k) \, \mathrm{d}\boldsymbol{\xi}_k = \min_{\boldsymbol{\xi}_k \in \Xi_k} h(\boldsymbol{\xi}_k) \tag{4.9}$$

This minimum occurs for a distribution p that is a single Dirac delta function centred at the $\boldsymbol{\xi}_k$ that minimises h. In such a case, the expressions for $\underline{\mathrm{E}}^d_k$ and $\underline{\mathrm{E}}^i_k$ become equivalent, as

$$\underline{\mathrm{E}}^d_k = \min_{p \in \mathcal{P}(\Xi_{k:M})} \int_{\Xi_{k:M}} h(\boldsymbol{\xi}_{k:M}) p(\boldsymbol{\xi}_{k:M}) \, \mathrm{d}\boldsymbol{\xi}_{k:M} = \min_{\boldsymbol{\xi}_{k:M} \in \Xi_{k:M}} h(\boldsymbol{\xi}, \mathbf{u}) \tag{4.10}$$

and

$$\begin{aligned}
\underline{\mathrm{E}}^i_k &= \min_{p \in \mathcal{Q}_{k:M}} \int_{\Xi_{k:M}} h(\boldsymbol{\xi}_{k:M}) p(\boldsymbol{\xi}_{k:M}) \, \mathrm{d}\boldsymbol{\xi}_{k:M} \\
&= \min_{p_1 \in \mathcal{P}(\Xi_1)} \min_{p_2 \in \mathcal{P}(\Xi_2)} \cdots \min_{p_M \in \mathcal{P}(\Xi_M)} \int_{\Xi_1} p_1(\boldsymbol{\xi}_1) \int_{\Xi_2} p_2(\boldsymbol{\xi}_2) \cdots \int_{\Xi_1} p_M(\boldsymbol{\xi}_M) h(\boldsymbol{\xi}, \mathbf{u}) \, \mathrm{d}\boldsymbol{\xi} \\
&= \min_{\boldsymbol{\xi}_1 \in \Xi_1} \min_{\boldsymbol{\xi}_2 \in \Xi_2} \cdots \min_{\boldsymbol{\xi}_M \in \Xi_M} h(\boldsymbol{\xi}, \mathbf{u}) = \min_{\boldsymbol{\xi}_{k:M} \in \Xi_{k:M}} h(\boldsymbol{\xi}, \mathbf{u})
\end{aligned} \tag{4.11}$$

This equality is only valid in this case of totally unconstrained distributions (apart from the implicit constraint that the variables $\boldsymbol{\xi}_k$ can only take values within Ξ_k). The aim of this study is to consider cases where some information is available to characterise the distributions, such that $\mathcal{Q}_k \subset \mathcal{P}(\Xi_k)$, but where this information is not sufficient to specify a single probability distribution.

In the general case, since $\hat{\mathcal{Q}}$ is the result of adding to $\mathcal{Q}$ the constraint that the variables $\boldsymbol{\xi}_k$ are independent from each other, it holds that $\hat{\mathcal{Q}}_{k:M} \subseteq \mathcal{Q}_{k:M}$, from which, in general, $\underline{\mathrm{E}}_k^d \leq \underline{\mathrm{E}}_k^i$. This is formalised in Theorem 4.

Theorem 4. *Suppose $\underline{\mathrm{E}}_k^d$ is defined by Eq. (4.5), based on the family of distributions $\mathcal{Q}_{k:M}$ in Definition 4.3.1, while $\underline{\mathrm{E}}_k^i$ is defined by Eq. (4.8) with respect to the family of distributions $\hat{\mathcal{Q}}_{k:M}$ in Definition 4.3.2. For any quantity of interest $h : \Xi \to \mathbb{R}$, and for any families of distributions $\hat{\mathcal{Q}}_k$ used in Definitions 4.3.1 and 4.3.2, the following inequality holds:*

$$\underline{\mathrm{E}}_k^d \leq \underline{\mathrm{E}}_k^i \tag{4.12}$$

Proof. From Eqs. (4.5) and (4.8), both $\underline{\mathrm{E}}_k^d$ and $\underline{\mathrm{E}}_k^i$ are defined as a minimum value of the same function (the expected value of some function of $\boldsymbol{\xi}$), over sets $\mathcal{Q}_{k:M}$ and $\hat{\mathcal{Q}}_{k:M}$. Since $\hat{\mathcal{Q}}_{k:M}$ is the subset of the distributions in $\mathcal{Q}_{k:M}$ which are of independent variables, i.e. $\hat{\mathcal{Q}}_{k:M} \subseteq \mathcal{Q}_{k:M}$, it must hold that $\underline{\mathrm{E}}_k^d \leq \underline{\mathrm{E}}_k^i$.

This can be proven by contradiction. Suppose $\underline{\mathrm{E}}_k^d > \underline{\mathrm{E}}_k^i$. Let the distribution $q \in \hat{\mathcal{Q}}_{k:M}$ be the minimiser of Eq. (4.8), such that $\underline{\mathrm{E}}_k^i = \int_{\Xi_{k:M}} h(\boldsymbol{\xi}_{k:M}) q(\boldsymbol{\xi}_{k:M}) \, \mathrm{d}\boldsymbol{\xi}_{k:M}$. Because $\hat{\mathcal{Q}}_{k:M} \subseteq \mathcal{Q}_{k:M}$, the distribution q is also in $\mathcal{Q}_{k:M}$, and therefore the fact that $\underline{\mathrm{E}}(q) < \underline{\mathrm{E}}_k^d$ would contradict, in Definition 4.40, that $\underline{\mathrm{E}}_k^d$ is the minimum value of $\underline{\mathrm{E}}$ for distributions in $\mathcal{Q}_{k:M}$ as in Eq. (4.5). Therefore, it is impossible that $\underline{\mathrm{E}}_k^d > \underline{\mathrm{E}}_k^i$, which proves this theorem. □

To summarise, $\mathcal{Q}$ in Definition 4.3.1 allows distributions of dependent variables, while $\hat{\mathcal{Q}}$ in Definition 4.3.2 does not. To illustrate the differences between these definitions, consider the following example. Suppose $M = 2$ and the marginal distributions are defined as $\hat{\mathcal{Q}}_k = \{\mathcal{N}(0, \Sigma) \; \forall \Sigma \in \Omega_\Sigma\}$, where $\mathcal{N}(0, \Sigma)$ is the normal distribution with mean zero and covariance matrix Σ, and so $\hat{\mathcal{Q}}_k$ is the set of normal distributions with

zero mean and standard deviations in a set Ω_Σ. With these definitions, $\mathcal{Q}$ will include (while not being limited to) distributions of the form

$$\mathcal{N}\left(0, \begin{bmatrix} \Sigma & \Pi \\ \Pi^T & \Sigma \end{bmatrix}\right) \ \forall \Sigma \in \Omega_\Sigma, \forall \Pi \in \mathbb{R}^{n_1 \times n_2} \tag{4.13}$$

where Π is constrained such that the whole covariance matrix is still positive semi-definite. This can be readily seen from the definition in Eq. (4.3). In other words, the set $\mathcal{Q}$ can allow for any type of dependency between variables $\boldsymbol{\xi}_k$, and as such is the most general way to turn imprecise probabilities on subsets of variables into a definition on the whole set. The set $\hat{\mathcal{Q}}$, on the other hand, would only include distributions of the form

$$\mathcal{N}\left(0, \begin{bmatrix} \Sigma & \mathbf{0} \\ \mathbf{0} & \Sigma \end{bmatrix}\right) \ \forall \Sigma \in \Omega_\Sigma \tag{4.14}$$

where $\mathbf{0}$ is the zero matrix.

Neither $\underline{\mathrm{E}}_k^d$ nor $\underline{\mathrm{E}}_k^i$ can be computed separately on each segment, but a dynamic programming formulation allows this. Towards that end, we propose the following "epistemic cost-to-go" function:

$$V_k(\mathbf{u}_{k:M}, \mathbf{x}_k) = \min_{p_k \in \hat{\mathcal{Q}}_k} \int_{\Xi_k} p_k(\boldsymbol{\xi}_k) \left(V_{k+1}(\mathbf{u}_{k+1:M}, \Phi_k(\mathbf{x}_k, \mathbf{u}_k, \boldsymbol{\xi}_k)) + l_k(\mathbf{x}_k, \mathbf{u}_k, \boldsymbol{\xi}_k)\right) \mathrm{d}\boldsymbol{\xi}_k \tag{4.15}$$

where $\mathbf{u}_{k:M}$ refers to the control laws from segments k to M, and $\hat{\mathcal{Q}}_k$ is the family of distributions p_k for variables $\boldsymbol{\xi}_k$. The definition is completed by adding

$$V_{M+1}(\mathbf{x}_{M+1}) = L(\mathbf{x}_{M+1}) \tag{4.16}$$

Furthermore, the upper quantile in Definition (3.2.3) is to be defined with V_1 instead of $\underline{\mathrm{E}}$.

The value of V_k is an approximation of the lower expectation of the quantity of interest $h(\mathbf{u}, \boldsymbol{\xi})$ when the propagation is started from segment k. If one again briefly considers the case where the distributions are completely unconstrained, i.e. $\hat{\mathcal{Q}}_k =$

$\mathcal{P}(\Xi_k)$, then

$$V_k(\mathbf{u}_{k:M}, \mathbf{x}_k) = \min_{\boldsymbol{\xi}_k \in \Xi_k} \left(V_{k+1}(\mathbf{u}_{k+1:M}, \Phi_k(\mathbf{x}_k, \mathbf{u}_k, \boldsymbol{\xi}_k)) + l_k(\mathbf{x}_k, \mathbf{u}_k, \boldsymbol{\xi}_k) \right) \tag{4.17}$$

This expression is now similar to the cost-to-go used in differential dynamic programming [53][2], albeit with the uncertain variable functioning like the control variable. In this case, if one obtains the non-recursive form of Eq. (4.17), it becomes evident that $\underline{\mathrm{E}}_k^d = V_k = \underline{\mathrm{E}}_k^i$. When $\hat{\mathcal{Q}}_k \subset \mathcal{P}(\Xi_k)$, in general these equalities do not hold.

A very desirable property of V_k, is that while it does not in general equal either $\underline{\mathrm{E}}_k^d$ or $\underline{\mathrm{E}}_k^i$, it can be shown that $\underline{\mathrm{E}}_k^d \leq V_k \leq \underline{\mathrm{E}}_k^i$. This statement means that the quantity V_k is always bounded by relevant values of the lower expectation, it can be seen as a conservative estimate of $\underline{\mathrm{E}}_k^i$, and the amount by which it is conservative is limited by $\underline{\mathrm{E}}_k^d$. Two theorems follow which prove each of the inequalities in that statement, but first, the definition of a convex family of distributions is introduced:

Definition 4.3.3. A family of distributions $\mathcal{Q}$ is convex if all linear mixtures of distributions in $\mathcal{Q}$ are themselves in $\mathcal{Q}$. More rigorously, if this family is parameterised using $\boldsymbol{\lambda}$, by writing $p(\boldsymbol{\xi}; \boldsymbol{\lambda})$ such that

$$p(\boldsymbol{\xi}; \boldsymbol{\lambda}) \in \mathcal{Q} \; \forall \boldsymbol{\lambda} \in \Lambda \subseteq \mathbb{R}^{N_{\boldsymbol{\lambda}}} \tag{4.18}$$

and there is a weight function $w(\boldsymbol{\lambda}) : \Lambda \to \mathbb{R}$ such that $\int_\Lambda w(\boldsymbol{\lambda}) \, \mathrm{d}\boldsymbol{\lambda} = 1$, then the set $\mathcal{Q}$ is convex iff the following holds for all such weight functions w and parameterisations $\boldsymbol{\lambda}$,

$$\int_\Lambda p(\boldsymbol{\xi}; \boldsymbol{\lambda}) w(\boldsymbol{\lambda}) \, \mathrm{d}\boldsymbol{\lambda} \in \mathcal{Q} \tag{4.19}$$

Families of distributions of the type in Section 3.4 are always convex in the sense of Definition 4.3.3. Assume a parameterisation of the set $\mathcal{Q}_k$ in Eq. (3.22), $p(\boldsymbol{\xi}_k; \boldsymbol{\lambda})$, with parameter $\boldsymbol{\lambda}$, such that for some set Λ, if $\lambda \in \Lambda$, then $p(\boldsymbol{\xi}_k; \boldsymbol{\lambda}) \in \mathcal{Q}_k$. The following shows that a linear mixture $\int_\Lambda p(\boldsymbol{\xi}_k; \boldsymbol{\lambda}) w(\boldsymbol{\lambda}) \, \mathrm{d}\boldsymbol{\lambda}$, with $\int_\Lambda w(\boldsymbol{\lambda}) \, \mathrm{d}\boldsymbol{\lambda} = 1$, is also in the

[2] See e.g. Eq. (5) of [53]

same set, thus meeting the requirements for convexity of set $\mathcal{Q}_k$:

$$\int_{\Xi_k}\int_{\Lambda} p(\boldsymbol{\xi}_k;\boldsymbol{\lambda})w(\boldsymbol{\lambda})\,\mathrm{d}\boldsymbol{\lambda}\; g_i(\boldsymbol{\xi}_k)\,\mathrm{d}\boldsymbol{\xi}_k = \int_{\Lambda} w(\boldsymbol{\lambda}) \underbrace{\int_{\Xi_k} p(\boldsymbol{\xi}_k;\boldsymbol{\lambda})g_i(\boldsymbol{\xi}_k)\,\mathrm{d}\boldsymbol{\xi}_k}_{\in\left[c_i^L,c_i^U\right] \text{ by definition}}\,\mathrm{d}\boldsymbol{\lambda} \tag{4.20}$$

The outer integral consists of an average of the value of the inner integral, whose value by definition is always in the interval $\left[c_i^L, c_i^U\right]$, and therefore the outer integral must also be in this interval.

Theorem 5. *Suppose $\underline{\mathrm{E}}_k^d$ is defined by Eq. (4.5) with family of distributions $\mathcal{Q}_{k:M}$ in Definition 4.3.1, and V_k is given by Eqs. (4.15) and (4.16). Suppose also that the sets $\hat{\mathcal{Q}}_k$ are convex according to Definition 4.3.3, and all distributions contained therein are Riemann integrable. The following inequality then holds independently of the quantity of interest h:*

$$\underline{\mathrm{E}}_k^d \leq V_k \tag{4.21}$$

In other words, the lower expectation obtained by allowing distributions of dependent variables gives a lower bound for the epistemic cost-of-go function V, when the sets $\hat{\mathcal{Q}}_k$ are convex. [3]

Proof. Without loss of generality, to simplify notation this is a proof for $\underline{\mathrm{E}}^d \leq V_1$, which is equivalent to the statement of the theorem. Let us define the following families of conditional distributions,

$$\tilde{\mathcal{Q}}_k(p_{1:k-1}) = \left\{p_k : p_k(\boldsymbol{\xi}_k|\boldsymbol{\xi}_{1:k-1})p_{1:k-1}(\boldsymbol{\xi}_{1:k-1}) \in \mathcal{Q}_{1:k}\right\} \tag{4.22}$$

where the argument must obey $p_{1:k-1} \in \mathcal{Q}_{1:k-1}$ (see Eq. 4.3), $\tilde{\mathcal{Q}}_1$ has no arguments, and $\tilde{\mathcal{Q}}_1 = \hat{\mathcal{Q}}_1$. Note that the distributions in $\tilde{\mathcal{Q}}_k(p_{1:k-1})$ for $k > 1$ depend on $\boldsymbol{\xi}_{1:k-1}$ as they are conditional distributions, as opposed to those in $\hat{\mathcal{Q}}_k$. The set $\mathcal{Q}$ can then be

[3] If the sets were not convex, a lower bound could instead be obtained by defining $\underline{\mathrm{E}}^d$ with the convex hulls of sets $\hat{\mathcal{Q}}_k$.

expressed as

$$\mathcal{Q} = \left\{ p \in \mathcal{P}(\Xi) : \begin{array}{c} p = p_1(\boldsymbol{\xi}_1)p_2(\boldsymbol{\xi}_2|\boldsymbol{\xi}_1)\cdots p_M(\boldsymbol{\xi}_M|\boldsymbol{\xi}_{1:M-1}), \\ \forall p_M \in \tilde{\mathcal{Q}}_M(p_{1:M-1}),\ \cdots,\ \forall p_2 \in \tilde{\mathcal{Q}}_2(p_1),\ \forall p_1 \in Q_1 \end{array} \right\} \tag{4.23}$$

which can be seen as analogous to writing a multivariate distribution as the product of conditional distributions, i.e. $p(\boldsymbol{\xi}) = p_1(\boldsymbol{\xi}_1)p_2(\boldsymbol{\xi}_2|\boldsymbol{\xi}_1)\dots p_M(\boldsymbol{\xi}_M|\boldsymbol{\xi}_{1:M-1})$. That the definitions in Eqs. (4.3) and (4.23) are equivalent is shown in Appendix 4.4. Using these definitions, one writes

$$\underline{\mathrm{E}}^d = \min_{p_1 \in \mathcal{Q}_1} \min_{p_2 \in \tilde{\mathcal{Q}}_2(p_1)} \cdots \min_{p_M \in \tilde{\mathcal{Q}}_M(p_{1:M-1})} \int_\Xi p_1(\boldsymbol{\xi}_1)p_2(\boldsymbol{\xi}_2|\boldsymbol{\xi}_1)\cdots p_M(\boldsymbol{\xi}_M|\boldsymbol{\xi}_{1:M-1})h(\boldsymbol{\xi},\mathbf{u})\,\mathrm{d}\boldsymbol{\xi} \tag{4.24}$$

Let us now take Eqs. (4.15) and (4.16), which define V_k, and recursively replace V_k by its expression as a function of V_{k+1} given by Eq. (4.15) (or Eq. (4.16) for V_{M+1}), and then using Eq. (4.2) to collect all the l_k terms and L into $h(\mathbf{u},\boldsymbol{\xi})$, resulting in the following expression for V_1.

$$V_1 = \min_{p_1 \in \hat{\mathcal{Q}}_1} \int_{\Xi_1} p_1(\boldsymbol{\xi}_1) \min_{p_2 \in \hat{\mathcal{Q}}_2} \int_{\Xi_2} p_2(\boldsymbol{\xi}_2) \dots \min_{p_M \in \hat{\mathcal{Q}}_M} \int_{\Xi_M} p_M(\boldsymbol{\xi}_M)h(\boldsymbol{\xi},\mathbf{u})\mathrm{d}\boldsymbol{\xi} \tag{4.25}$$

To compare V_1 with $\underline{\mathrm{E}}^d$ it is desirable to somehow move all minimisations to the outside of the integrals. The minimisations being inside the integrals means a different p_k may be found for each value of the integration variable $\boldsymbol{\xi}_{1:k-1}$. To write V_1 in an expression where all minimisations are outside the integrals, this must be made explicit, by writing the search space for the minimisation of p_k as being explicitly dependent on the integration variables $\boldsymbol{\xi}_{1:k-1}$.

Before doing that, let us gain some intuition by considering a case with discrete sums instead of integrals. Suppose that for two real valued functions f_1 and f_2 one has a minimisation of the form

$$\min_{\mathbf{x} \in X} f_1(\mathbf{x}) + \min_{\mathbf{x} \in X} f_2(\mathbf{x}) \tag{4.26}$$

It is well known that

$$\min_{\mathbf{x}\in X} f_1(\mathbf{x}) + \min_{\mathbf{x}\in X} f_2(\mathbf{x}) \leq \min_{\mathbf{x}\in X} (f_1(\mathbf{x}) + f_2(\mathbf{x})) \tag{4.27}$$

so in general the minimisation cannot be swapped with the sum in this manner. Since the two minimisations are conducted separately, nothing changes if the name of one of the variables is changed from $\mathbf{x}$ to $\mathbf{y}$,

$$\min_{\mathbf{x}\in X} f_1(\mathbf{x}) + \min_{\mathbf{y}\in X} f_2(\mathbf{y}) \tag{4.28}$$

and now the minimisation over $\mathbf{y}$ can be moved to before the sum without changes,

$$\min_{\substack{\mathbf{x}\in X \\ \mathbf{y}\in X}} (f_1(\mathbf{x}) + f_2(\mathbf{y})) \tag{4.29}$$

which is equivalent to Eq. (4.26). Now let us look at a more general case,

$$\min_{\mathbf{a}\in A\subseteq\mathbb{R}} \sum_{i=1}^{N_a} a_i \min_{\mathbf{b}\in B\subseteq\mathbb{R}} \sum_{j=1}^{N_b} b_j c_{ij} \tag{4.30}$$

The same idea applies, i.e. a different $\mathbf{b}$ will be found for each value of i. Following the same logic, the above problem can equivalently be written as

$$\min_{\mathbf{a}\in A} \min_{\substack{\mathbf{b}_i\in B \\ \forall i=1,\cdots,N_i}} \sum_{i=1}^{N_a} a_i \sum_{j=1}^{N_b} b_{i;j} c_{ij} \tag{4.31}$$

where the search space for the second minimisation is B^{N_i}.

A similar transformation can be done with the formula for V_1, although there one has integrals instead of sums, and real variables instead of indices. Consider for now its expression in Eq. (4.25), when $M = 2$. Relative to Eqs. (4.30) and (4.31), i and j are analogous to variables $\boldsymbol{\xi}_1$ and $\boldsymbol{\xi}_2$, a_i and b_j are analogous to $p_1(\boldsymbol{\xi}_1)$ and $p_2(\boldsymbol{\xi}_2)$, and c_{ij} to the value of $h(\boldsymbol{\xi}, \mathbf{u}) = h(\boldsymbol{\xi}_1, \boldsymbol{\xi}_2, \mathbf{u})$. After interchanging the integration and minimisation, $b_{i;j}$ is analogous to a probability distribution of $\boldsymbol{\xi}_2$ that is parameterised by $\boldsymbol{\xi}_1$. This is written in the form of a conditional distribution, i.e. $p_2(\boldsymbol{\xi}_2|\boldsymbol{\xi}_1)$.

An intuition for applying the transformation from Eq. (4.30) to Eq. (4.31) to an expression with integrals instead of sums such as Eq. (4.25) can be formed by writing the integral as the limit of a Riemann sum. [39] Consider again the case $M = 2$, and suppose $\boldsymbol{\xi}_1$ is a scalar and $\Xi_1 = \left[\xi_1^{\min}, \xi_1^{\max}\right]$. In this case,

$$V_1 = \min_{p_1 \in \hat{\mathcal{Q}}_1} \int_{\Xi_1} p_1(\xi_1) \min_{p_2 \in \hat{\mathcal{Q}}_2} \mathcal{F}(p_2 \; ; \; \xi_1) \, \mathrm{d}\xi_1 \tag{4.32}$$

where for convenience the following function was defined,

$$\mathcal{F}(p_2 \; ; \; \xi_1) = \int_{\Xi_2} p_2(\boldsymbol{\xi}_2) h(\boldsymbol{\xi} = [\xi_1, \boldsymbol{\xi}_2], \mathbf{u}) \mathrm{d}\boldsymbol{\xi}_2 \tag{4.33}$$

If the integrand in Eq. (4.32) is Riemann integrable, its integral can be written as the limit of a Riemann sum, for partitions $\xi_{1,0}, \xi_{1,1}, \ldots, \xi_{1,N_i}$, as the mesh tends to zero, i.e. $\max_i |\xi_{1,i} - \xi_{1,i-1}| \to 0$, and for any choice of $\xi_{1,i}^t \in [\xi_{1,i-1}, \xi_{1,i}]$, which can be written as [39]

$$\lim_{\max_i |\xi_{1,i} - \xi_{1,i-1}| \to 0} \sum_{i=1}^{N_i} \left(\xi_{1,i} - \xi_{1,i-1}\right) p_1\left(\xi_{1,i}^t\right) \min_{p_2 \in \hat{\mathcal{Q}}_2} \mathcal{F}\left(p_2 \; ; \; \xi_{1,i}^t\right) \tag{4.34}$$

This sum is in the form of Eq. (4.30), and it can be written in the form of Eq. (4.31) as follows

$$\lim_{\max_i |\xi_{1,i} - \xi_{1,i-1}| \to 0} \min_{p_{2,i} \in \hat{\mathcal{Q}}_2 \; \forall i \in \{1, \ldots, N_i\}} \sum_{i=1}^{N_i} \left(\xi_{1,i} - \xi_{1,i-1}\right) p_1\left(\xi_{1,i}^t\right) \mathcal{F}\left(p_2 \; ; \; \xi_{1,i}^t\right) \tag{4.35}$$

Each $p_{2,i}$ is a probability distribution of $\boldsymbol{\xi}_2$, and inspection of the formula above shows that the value of each optimal p_2 depends only on $\xi_{1,i}^t$. For this reason, instead of $p_2(\xi_2)$ one can write $p_2\left(\xi_2 \mid \xi_{1,i}^t\right)$. Replacing in Eq. (4.32) the integral with the limit of Eq. (4.35), one obtains an expression where all the minimisations are outside the integrals

$$V_1 = \min_{p_1 \in \hat{\mathcal{Q}}_1} \min_{p_2(\boldsymbol{\xi}_2 | \xi_1) \in \hat{\mathcal{Q}}_2 \; \forall \xi_1 \in \Xi_1} \int_{\Xi_1} p_1(\xi_1) \mathcal{F}(p_2 \; ; \; \xi_1) \, \mathrm{d}\xi_1 \tag{4.36}$$

Extension to non-scalar variable $\boldsymbol{\xi}_1$ proceeds the same way as does the definition of the Riemann integral, i.e. by nesting the Riemman sums, while extension to higher M by applying the preceding argument iteratively. In the extension to $M \geq 2$, the following

sets are defined, which are analogous to B^{N_i} in Eq. (4.31):

$$\hat{\mathcal{Q}}_k^{\Xi_{1:k-1}} = \left\{ p_k : p_k(\boldsymbol{\xi}_k|\boldsymbol{\xi}_{1:k-1}) \in \hat{\mathcal{Q}}_k \ \forall \boldsymbol{\xi}_{1:k-1} \in \Xi_{1:k-1} \right\} \tag{4.37}$$

These are sets that contain all functions in $\hat{\mathcal{Q}}_k$ for each value of $\boldsymbol{\xi}_{1:k-1} \in \Xi_{1:k-1}$, similarly to how B^{N_i} contained all real vectors $\mathbf{b} \in B$ for each value of $i \in \{1, \cdots, N_i\}$. Previously, the variables $\mathbf{b} \in B$ were indexed using only index j, whereas after swapping the minimisation and the sum they become indexed with j and i, see Eqs. (4.30) and (4.31). Analogously, the distributions p_k go from being functions of $\boldsymbol{\xi}_k$ only, to depending on both $\boldsymbol{\xi}_k$ and $\boldsymbol{\xi}_{1:k-1}$. Note that they are only distributions of $\boldsymbol{\xi}_k$, as the p_k are only constrained to integrate to 1 over that variable. For this reason, they are written as conditional distributions $p_k(\boldsymbol{\xi}_k|\boldsymbol{\xi}_{1:k-1})$. Finally, V_1 can be written with all minimisations outside all integrals:

$$V_1 = \min_{p_1 \in \mathcal{Q}_1} \min_{p_2 \in \hat{\mathcal{Q}}_2^{\Xi_1}} \cdots \min_{p_M \in \hat{\mathcal{Q}}_M^{\Xi_{1:M-1}}} \int_{\Xi} p_1(\boldsymbol{\xi}_1) p_2(\boldsymbol{\xi}_2|\boldsymbol{\xi}_1) \cdots p_M(\boldsymbol{\xi}_M|\boldsymbol{\xi}_{1:M-1}) h(\boldsymbol{\xi}, \mathbf{u}) \, \mathrm{d}\boldsymbol{\xi} \tag{4.38}$$

Now Eqs. (4.24) and (4.38) are very similar. The only difference between them is that the minimisation sets in Eq. (4.24) are $\tilde{\mathcal{Q}}_k(p_{1:k-1})$ whereas in Eq. (4.38) they are $\hat{\mathcal{Q}}_k^{\Xi_{1:k-1}}$. Using again the fact that a minimisation over a subset must result in a less or equal value, the proof that $\underline{\mathrm{E}}^d \leq V_1$ can now be achieved by proving $\hat{\mathcal{Q}}_k^{\Xi_{1:k-1}} \subseteq \tilde{\mathcal{Q}}_k(p_{1:k-1}), \ \forall p_{1:k-1} \in \mathcal{Q}_{1:k-1}$.

Suppose $p_k \in \hat{\mathcal{Q}}_k^{\Xi_{1:k-1}}$, and recall the assumption that the set $\hat{\mathcal{Q}}_k$ is convex. The marginal distribution $\mathcal{M}_k \left[p_k(\boldsymbol{\xi}_k|\boldsymbol{\xi}_{1:k-1}) p_{1:k-1}(\boldsymbol{\xi}_{1:k-1}) \right]$ is a linear mixture in the sense used in Definition 4.3.3, where the parameter $\boldsymbol{\lambda} \in \Lambda$ is $\boldsymbol{\xi}_{1:k-1} \in \Xi_{1:k-1}$, and the weight function w is given by $p_{1:k-1}$. Therefore, since from the assumption $p_k \in \hat{\mathcal{Q}}_k^{\Xi_{1:k-1}}$ it holds that $p_k(\boldsymbol{\xi}_k|\boldsymbol{\xi}_{1:k-1}) \in \hat{\mathcal{Q}}_k \ \forall \boldsymbol{\xi}_{1:k-1} \in \Xi_{1:k-1}$, the necessary conditions are present to state that, when $\hat{\mathcal{Q}}_k$ is convex, $p_k \in \hat{\mathcal{Q}}_k^{\Xi_{1:k-1}}$ implies $\mathcal{M}_k \left[p_k(\boldsymbol{\xi}_k|\boldsymbol{\xi}_{1:k-1}) p_{1:k-1}(\boldsymbol{\xi}_{1:k-1}) \right] \in \hat{\mathcal{Q}}_k \ \forall p_{1:k-1} \in \mathcal{Q}_{1:k-1}$, which in turn implies $p_k \in \tilde{\mathcal{Q}}_k(p_{1:k-1}), \ \forall p_{1:k-1} \in \mathcal{Q}_{1:k-1}$. This proves that $\hat{\mathcal{Q}}_k^{\Xi_{1:k-1}} \subseteq \tilde{\mathcal{Q}}_k(p_{1:k-1}), \ \forall p_{1:k-1} \in \mathcal{Q}_{1:k-1}$, and in turn, that $\underline{\mathrm{E}}^d \leq V_1$, when all $\hat{\mathcal{Q}}_k$ are convex.

□

Theorem 6. *Now consider* $\underline{\mathrm{E}}_k^i$ *defined by Eq. (4.8) with respect to the family of distributions* $\hat{\mathcal{Q}}_{k:M}$ *in Definition 4.3.2, and* V_k *given by Eqs. (4.15) and (4.16). The following inequality then holds:*

$$V_k \leq \underline{\mathrm{E}}_k^i \tag{4.39}$$

and just like in the previous theorem, this holds independently of h*. Unlike in the previous theorem, here it does not matter whether* $\hat{\mathcal{Q}}_k$ *are convex. This inequality shows that* V_k *can be seen as a conservative estimate of the lower expectation that is obtained by restricting the families of distributions to be those of independent variables.*

Proof. Again, to simplify the notation, we prove $V_1 \leq \underline{\mathrm{E}}^i$, which is equivalent to $V_k \leq \underline{\mathrm{E}}_k^i$. Let us start by writing

$$\underline{\mathrm{E}}^i = \min_{p_1 \in \mathcal{Q}_1} \min_{p_2 \in \hat{\mathcal{Q}}_2} \dots \min_{p_M \in \hat{\mathcal{Q}}_M} \int_{\Xi_1} p_1(\boldsymbol{\xi}_1) \int_{\Xi_2} p_2(\boldsymbol{\xi}_2) \dots \int_{\Xi_1} p_M(\boldsymbol{\xi}_M) h(\boldsymbol{\xi}, \mathbf{u}) \mathrm{d}\boldsymbol{\xi} \tag{4.40}$$

The obstacle to separating the variables in this formulation in the dynamic programming sense, is that the optimal p_k depends on p_j for $j < k$. Solving this problem requires moving the minimisations to the inside of each integral. By moving a minimization to the inside of an integral, the resulting value must be lower or equal to the original value, i.e., if $\mathcal{F}$ is a set of real functions of $\mathbf{x} \in X$:

$$\int_X \min_{h(\mathbf{x}) \in \mathcal{F}} h(\mathbf{x}) \mathrm{d}\mathbf{x} \leq \min_{h(\mathbf{x}) \in \mathcal{F}} \int_X h(\mathbf{x}) \mathrm{d}\mathbf{x} \tag{4.41}$$

Moving each minimization over p_k to just behind the integral on Ξ_k, results in the same expression for V_1 seen in the proof of the previous theorem,

$$V_1 = \min_{p_1 \in \mathcal{Q}_1} \int_{\Xi_1} p_1(\boldsymbol{\xi}_1) \min_{p_2 \in \hat{\mathcal{Q}}_2} \int_{\Xi_2} p_2(\boldsymbol{\xi}_2) \dots \min_{p_M \in \hat{\mathcal{Q}}_M} \int_{\Xi_M} p_M(\boldsymbol{\xi}_M) h(\boldsymbol{\xi}, \mathbf{u}) \mathrm{d}\boldsymbol{\xi}_M \dots \mathrm{d}\boldsymbol{\xi}_1 \tag{4.42}$$

Given that this expression was obtained from Eq. (4.40) by moving each minimization to the inside of each integral, one concludes that $V_1 \leq \underline{\mathrm{E}}^i$, from which $V_k \leq \underline{\mathrm{E}}_k^i$, which concludes this proof.

□

The action of moving each minimisation to the inside of each integral in the proof of Theorem 6, can be seen as a case where the distributions p_k, instead of being independent, are being chosen by an adversarial agent who wants the expected value of the quantity of interest to be as low as possible, and is choosing the distributions based on the current state to achieve this. This provides an additional intuition for the statement proven in that theorem. The fact that this adversarial agent is only able to act based on the current state, as opposed to being able to optimise for the whole trajectory which would correspond to $\underline{\mathrm{E}}^d$, provides the intuition for Theorem 5.

The formulation with independent variables, $\underline{\mathrm{E}}^i$, is suitable to describe some commonly used stochastic processes. For example, a Gaussian white noise process can be discretised as a sequence of independent random variables corresponding to the value of the process at various time instants. Other processes, such as the Wiener process, being an integral of the white noise process, can be described by the addition of a state variable, containing the value of this process at the start of the segment, while still modelling the process using independent variables. In a scenario modelled with stochastic processes in this way, the fact that $V_k \leq \underline{\mathrm{E}}^i_k$, means that V_k will be a conservative estimate of the real lower expectation.

In general, the exact values of $\underline{\mathrm{E}}^d_k$, V_k and $\underline{\mathrm{E}}^i_k$ cannot be obtained analytically. Section 4.6 covers the practical computation of these quantities. However, in Section 4.5, a case is presented where these values can all be obtained analytically. Their computation is shown step-by-step. When looking at the theorems in this section, the reader may suspect that it is possible that some of them may actually be equalities in the general case. Therefore, this example was constructed to be an as-simple-as-possible case where all the inequalities of the theorems 4, 5 and 6 are verified in the strict sense, i.e. $\underline{\mathrm{E}}^d_k < V_k < \underline{\mathrm{E}}^i_k$.

4.4 Representation with Families of Conditional Distributions

When proving Theorem 5, the set $\mathcal{Q}$ was represented in a form of a decomposition into sets of conditional distributions $\tilde{\mathcal{Q}}_k(p_{1:k-1})$, see Eq. (4.23). Here it is shown that this is equivalent to the original definition in Eq. (4.3). To avoid confusion, let us re-write Eq. (4.23), but represent the left-hand side by a different symbol,

$$\tilde{\mathcal{Q}}_{1:M} = \left\{ p_{1:M} \in \mathcal{P}(\Xi_{1:M}) : \begin{array}{c} p = p_1(\boldsymbol{\xi}_1)p_2(\boldsymbol{\xi}_2|\boldsymbol{\xi}_1)\cdots p_M(\boldsymbol{\xi}_M|\boldsymbol{\xi}_{1:M-1}), \\ \forall p_M \in \tilde{\mathcal{Q}}_M(p_{1:M-1}),\ \cdots,\ \forall p_2 \in \tilde{\mathcal{Q}}_2(p_1),\ \forall p_1 \in Q_1 \end{array} \right\} \tag{4.43}$$

The proof is by induction. First it is shown that $\tilde{\mathcal{Q}}_1 = \mathcal{Q}_1$ (base case), and then that $\tilde{\mathcal{Q}}_{1:M-1} = \mathcal{Q}_{1:M-1}$ implies $\tilde{\mathcal{Q}}_{1:M} = \mathcal{Q}_{1:M}$ (induction step). The base case is simple. Replacing M by 1 in Eq. (4.43) directly results in $\tilde{\mathcal{Q}}_1 = \mathcal{Q}_1$.

For the induction step, first write Eq. (4.43) in a recursive form,

$$\tilde{\mathcal{Q}}_{1:M} = \left\{ p_{1:M} \in \mathcal{P}(\Xi_{1:M}) : \begin{array}{c} p = p_{1:M-1}(\boldsymbol{\xi}_{1:M-1})p_M(\boldsymbol{\xi}_M|\boldsymbol{\xi}_{1:M-1}), \\ \forall p_M \in \tilde{\mathcal{Q}}_M(p_{1:M-1}),\ \forall p_{1:M-1} \in \tilde{\mathcal{Q}}_{1:M-1} \end{array} \right\} \tag{4.44}$$

To prove that $\tilde{\mathcal{Q}}_{1:M} = \mathcal{Q}_{1:M}$ under the assumption that $\tilde{\mathcal{Q}}_{1:M-1} = \mathcal{Q}_{1:M-1}$, we prove both that $p_{1:M} \in \tilde{\mathcal{Q}}_{1:M} \implies p_{1:M} \in \mathcal{Q}_{1:M}$ (equivalent to $\tilde{\mathcal{Q}}_{1:M} \subseteq \mathcal{Q}_{1:M}$) and vice versa, i.e. that $p_{1:M} \in \mathcal{Q}_{1:M} \implies p_{1:M} \in \tilde{\mathcal{Q}}_{1:M}$ (equivalent to $\mathcal{Q}_{1:M} \subseteq \tilde{\mathcal{Q}}_{1:M}$).

The marginal distributions $\mathcal{M}_j[p_{1:M}]$ of any distribution $p_{1:M-1}$ can be written, by decomposing them into conditional distributions, as

$$\mathcal{M}_j[p_{1:M}] = \begin{cases} \mathcal{M}_j\left[p_{1:M-1}(\boldsymbol{\xi}_{1:M-1})\right] & j < M \\ \mathcal{M}_M\left[p_M(\boldsymbol{\xi}_M \mid \boldsymbol{\xi}_{1:M-1}) \cdot p_{1:M-1}(\boldsymbol{\xi}_{1:M-1})\right] & j = M \end{cases} \tag{4.45}$$

If $p_{1:M} \in \tilde{\mathcal{Q}}_{1:M}$, for $j < M$, Eq. (4.44) implies that $p_{1:M-1} \in \tilde{\mathcal{Q}}_{1:M-1}$, which combined with the induction step assumption gives $p_{1:M-1} \in \mathcal{Q}_{1:M-1}$. Using the definition of $\mathcal{Q}_{1:M-1}$ in Eq. (4.3), it follows that $\mathcal{M}_j[p_{1:M-1}] \in \hat{\mathcal{Q}}_j\ \forall j < M$. For $j = M$, the assumption $p_{1:M} \in \tilde{\mathcal{Q}}_{1:M}$ and Eq. (4.44) also imply that $p_M \in \tilde{Q}_M(p_{1:M-1})$. Equation

(4.22) then gives $p_M(\boldsymbol{\xi}_M \mid \boldsymbol{\xi}_{1:M-1}) \cdot p_{1:M-1}(\boldsymbol{\xi}_{1:M-1}) \in \mathcal{Q}_{1:M}$, and again from the definition of $\mathcal{Q}_{1:M}$ given by Eq. (4.3) this gives $\mathcal{M}_M \left[p_M(\boldsymbol{\xi}_M \mid \boldsymbol{\xi}_{1:M-1}) \cdot p_{1:M-1}(\boldsymbol{\xi}_{1:M-1})\right] \in \hat{\mathcal{Q}}_M$. Together, these conclusions imply that $\mathcal{M}_j[p_{1:M}] \in \mathcal{Q}_{1:M}$, which means, from Eq. (4.3), that $p_{1:M} \in \tilde{\mathcal{Q}}_{1:M} \implies p_{1:M} \in \mathcal{Q}_{1:M}$.

For the other direction, i.e. $p_{1:M} \in \mathcal{Q}_{1:M} \implies p_{1:M} \in \tilde{\mathcal{Q}}_{1:M}$, again write $p_{1:M}(\boldsymbol{\xi}_{1:M}) = p_M(\boldsymbol{\xi}_M \mid \boldsymbol{\xi}_{1:M-1}) \cdot p_{1:M-1}(\boldsymbol{\xi}_{1:M-1})$. Using Eq. (4.3), and the induction step assumption $\tilde{\mathcal{Q}}_{1:M-1} = \mathcal{Q}_{1:M-1}$, results in $p_{1:M-1} \in \tilde{\mathcal{Q}}_{1:M-1}$. Now, from Eq. (4.44), to prove $p_{1:M} \in \tilde{\mathcal{Q}}_{1:M}$ one only needs $p_M \in \tilde{\mathcal{Q}}_M(p_{1:M-1})$. Using Eq. (4.22), one sees that this is a consequence of the initial assumption $p_{1:M}(\boldsymbol{\xi}_{1:M}) = p_M(\boldsymbol{\xi}_M \mid \boldsymbol{\xi}_{1:M-1}) \cdot p_{1:M-1}(\boldsymbol{\xi}_{1:M-1}) \in \mathcal{Q}_{1:M}$, concluding that $p_{1:M} \in \mathcal{Q}_{1:M} \implies p_{1:M} \in \tilde{\mathcal{Q}}_{1:M}$.

This completes the proof that $\tilde{\mathcal{Q}}_{1:M} = \mathcal{Q}_{1:M}\ \forall M \geq 1$.

4.5 Examples

Several non-strict inequalities were presented and proven. However, it is possible for a doubt to remain, that some of those non-strict inequalities may actually be equalities, which would not contradict the theorem in question, but would make it somewhat misleading. In this section, an example is presented where the inequalities in theorems 4, 5, and 6 are all strict, i.e. where it holds that $\underline{\mathrm{E}}^d < V_1 < \underline{\mathrm{E}}^i$. This serves to prove that neither of the non-strict inequaltities in those theorems could have been written as equalities, as this example would be a counterexample.

Consider two scalar variables, $\xi_1 \in \Xi_1 = [-1, 1]$ and $\xi_2 \in \Xi_2 = [-1, 1]$. The families of distributions of marginals $\hat{\mathcal{Q}}_k$ are defined simply by constraining the means to be zero,

$$\hat{\mathcal{Q}}_k = \left\{ p(\xi_k) \in \mathcal{P}(\Xi_k) : \int_{\Xi_k} p(\xi_k)\xi_k \,\mathrm{d}\xi_k = 0 \right\} . \tag{4.46}$$

The quantity of interest function $h(\xi_1, \xi_2)$ is a piecewise constant function:

$$h(\xi_1, \xi_2) = \begin{cases} 1, & \text{if } \xi_1 \leq \epsilon \text{ and } \xi_2 \leq \epsilon \\ 2, & \text{otherwise and if } \xi_1 \geq 0 \text{ and } \xi_2 \geq 0 \\ 0, & \text{otherwise} \end{cases} \tag{4.47}$$

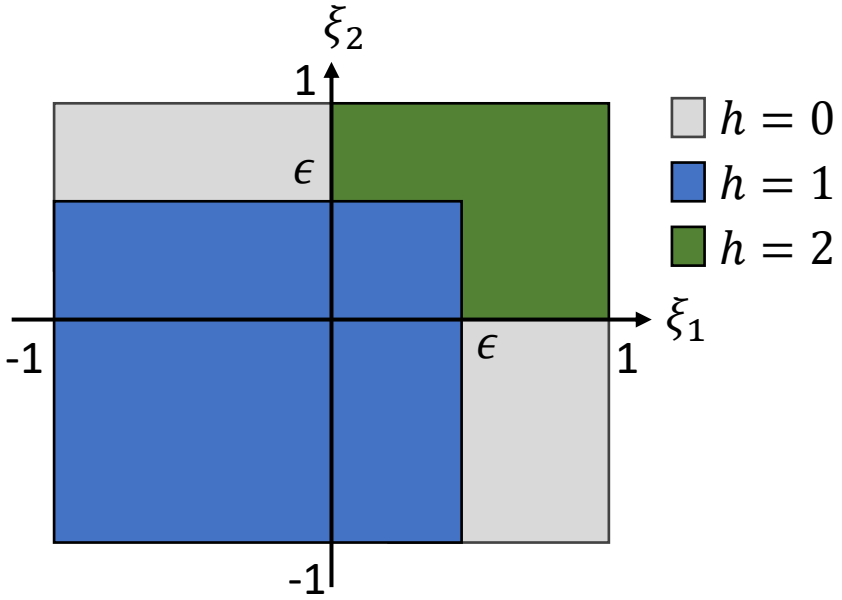


Figure 4.2: Illustration of quantity of interest function $h(\xi_1, \xi_2)$

where the parameter ϵ, for the purpose of this example, should be chosen such that $0 < \epsilon < 1$. In subsequent computations, it will be taken to be $\epsilon = 1/2$. Figure 4.2 illustrates this function.

The work of Golodnikov [27] shows that the marginal distributions in the sets of Eq. (4.46) will be a weighted sum of 2 Dirac delta kernels,

$$p_k\left(\xi_k\right) = w_{k,1}\delta\left(\xi_k - \xi_{k,1}^c\right) + w_{k,2}\delta\left(\xi_k - \xi_{k,2}^c\right) \ , \tag{4.48}$$

where δ is the Dirac delta function. Confirming that this result also applies to this example is outside the scope of this work, so to remove all doubt, the results in this section can also be obtained by adding this constraint to the definition of $\hat{\mathcal{Q}}_k$,

$$\hat{\mathcal{Q}}_k = \left\{ p \in \mathcal{P}(\Xi_k) : p(\xi_k) = w_{k,1}\delta\left(\xi_k - \xi_{k,1}^c\right) + w_{k,2}\delta\left(\xi_k - \xi_{k,2}^c\right) \ , \ \int_{\Xi_k} p(\xi_k)\xi_k \, \mathrm{d}\xi_k = 0 \right\} . \tag{4.49}$$

The implicit constraint that $\int p_k \, \mathrm{d}\xi_k = 1$, in addition to the mean constraint, allow writing

$$w_{k,1} = \frac{\xi_{k,2}^c}{\xi_{k,2}^c - \xi_{k,1}^c} \ , \ w_{k,2} = \frac{-\xi_{k,1}^c}{\xi_{k,2}^c - \xi_{k,1}^c} \ . \tag{4.50}$$

4.5.1 Lower Expectation with Marginal Constraints Only, $\underline{\mathrm{E}}^d$

The following distribution places all probability mass in the region where $h = 0$, resulting in $E[h] = 0$,

$$p^d(\xi_k) = \frac{1}{2}\delta\left(\xi_1 - 1\right)\delta\left(\xi_2 - 1\right) + \frac{1}{2}\delta\left(\xi_1 + 1\right)\delta\left(\xi_2 + 1\right) \ . \tag{4.51}$$

To see that $p^d \in \mathcal{Q}$, it is enough to show that the marginals of p^d have marginals that are the sum of two Dirac delta functions with zero mean for both variables, in particular,

$$\mathcal{M}_k\left[p^d\right] = \frac{1}{2}\delta\left(\xi_k - 1\right) + \frac{1}{2}\delta\left(\xi_k + 1\right) \ , \tag{4.52}$$

where it is evident that the means of both variables are zero, and so $\mathcal{M}_k\left[p^d\right] \in \hat{\mathcal{Q}}_k$, i.e. $p^d \in \mathcal{Q}$.

The distribution in Eq. (4.51) is not one of independent variables, as it cannot be written as a product of a distribution over ξ_1 and a distribution over ξ_2, and so it is not in $\hat{\mathcal{Q}}$. Therefore, the value of $\underline{\mathrm{E}}^i$ may not be zero, and in fact, in the following subsection it is shown that it is greater than zero.

4.5.2 Lower Expectation Assuming Independent Variables, $\underline{\mathrm{E}}^i$

Because the distributions are over independent variables, $p^i \in \hat{\mathcal{Q}}$ is written as

$$\begin{aligned} p^i(\xi_1, \xi_2) &= p_1(\xi_1)p_2(\xi_2) \\ &= w_{1,1}w_{2,1}\delta\left(\xi_1 - \xi^c_{1,1}\right)\delta\left(\xi_2 - \xi^c_{2,1}\right) + w_{1,1}w_{2,2}\delta\left(\xi_1 - \xi^c_{1,1}\right)\delta\left(\xi_2 - \xi^c_{1,2}\right) \\ &+ w_{1,2}w_{2,1}\delta\left(\xi_1 - \xi^c_{1,2}\right)\delta\left(\xi_2 - \xi^c_{1,1}\right) + w_{1,2}w_{2,2}\delta\left(\xi_1 - \xi^c_{1,2}\right)\delta\left(\xi_2 - \xi^c_{1,2}\right) \ , \end{aligned} \tag{4.53}$$

with $w_{k,l}$ as defined in Eqs. (4.50).

The constraint on the mean being zero implies that if, for the k-th variable, one of the kernel centres $\xi^c_{k,l}$ is non-negative, the other must be non-positive and vice-versa. Without loss of generality, consider $\xi^c_{k,1} \geq 0$ and $\xi^c_{k,2} \leq 0$. It helps analysing the

expressions if they are written in terms of non-negative values, so define $x_{k,l} \geq 0$ and

$$\begin{aligned} \xi^c_{k,1} &= x_{k,1} \\ \xi^c_{k,2} &= -x_{k,2} \end{aligned} , \tag{4.54}$$

and now the weights can be written as

$$w_{k,1} = \frac{x_{k,2}}{x_{k,2} + x_{k,1}} \ , \ w_{k,2} = \frac{x_{k,1}}{x_{k,2} + x_{k,1}} \ . \tag{4.55}$$

The problem of finding $\underline{\mathrm{E}}^i$ can be written as

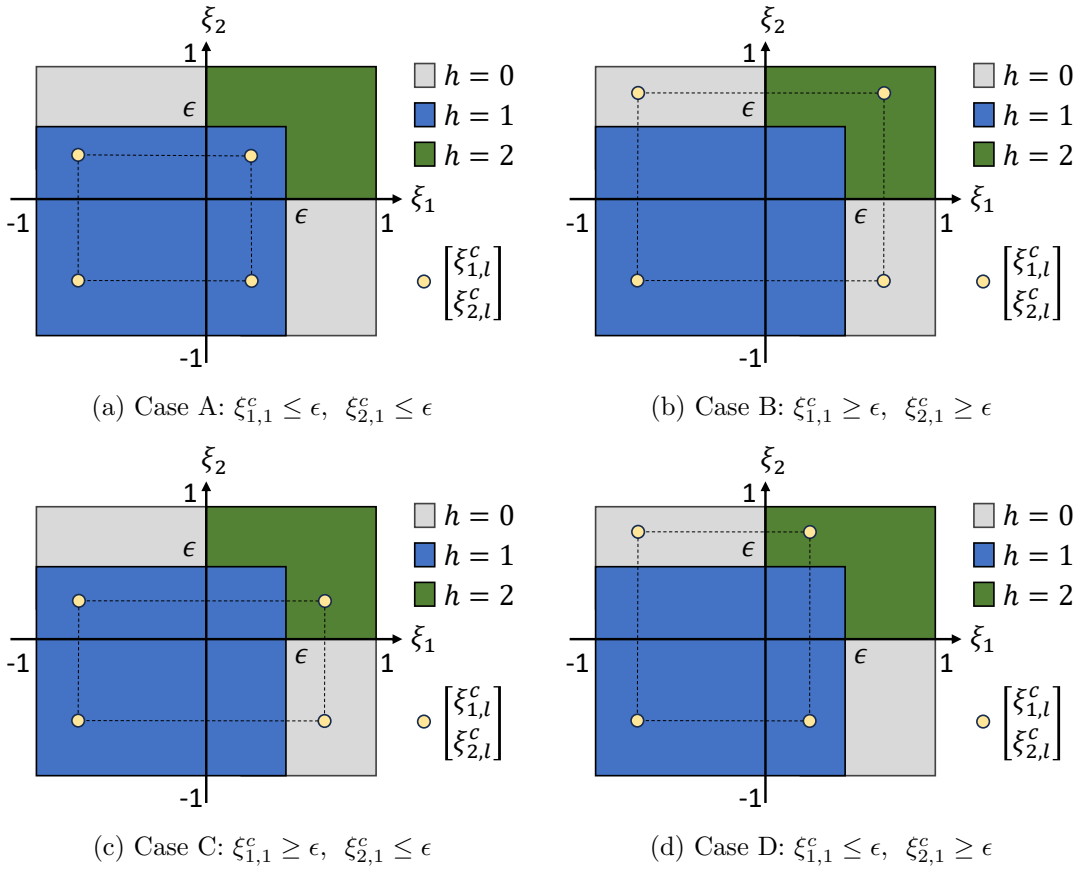
$$\underline{\mathrm{E}}^i = \min_{0 \leq x_{k,l} \leq 1} \frac{\sum_{i=1}^{2} \sum_{j=1}^{2} x_{1,3-i} x_{2,3-j} h(x_{1,i}, x_{2,j})}{(x_{1,2} + x_{1,1})(x_{2,2} + x_{2,1})} = \frac{1}{3} \tag{4.56}$$

This is easily solved by the symbolic computation software of Wolfram Mathematica [34], but a sketch of a proof that this is indeed the global minimum follows. There are four cases for the values of $h(x_{1,i}, x_{2,j})$, illustrated in Fig. 4.3. The proof proceeds by finding the minimum for each case individually. The global minimum is the smallest of these values.

For case A, all kernel centres are located in the region where $h = 1$, so for that case the lowest expected value is $\underline{\mathrm{E}}^{(i,A)} = 1$. For case B,

$$\underline{\mathrm{E}}^{(i,B)} = \min_{\substack{\epsilon \leq x_{1,1} \leq 1 \\ \epsilon \leq x_{2,1} \leq 1 \\ 0 \leq x_{1,2} \leq 1 \\ 0 \leq x_{2,2} \leq 1}} \frac{x_{1,1} x_{2,1} + 2 x_{1,2} x_{2,2}}{(x_{1,1} + x_{1,2})(x_{2,1} + x_{2,2})} = \frac{1}{3} \ . \tag{4.57}$$

This global minimum corresponds to $x_{1,1} = \frac{1}{2}$, $x_{1,2} = 0$, $x_{2,1} = \frac{1}{2}$, $x_{2,2} = 1$, where one could also swap $x_{1,l}$ with $x_{2,l}$. Showing that this is a global optimum requires a rather lengthy proof, of which we only present an outline. The sign of the derivative $\frac{\partial}{\partial x_{1,2}}$ of the expression being minimised is the sign of $x_{2,1} - 2x_{2,2}$. The search space is split in three regions, $x_{2,1} - 2x_{2,2} = 0$, $x_{2,1} - 2x_{2,2} < 0$, and $x_{2,1} - 2x_{2,2} > 0$, named cases B1, B2, and B3 respectively. For case B1, the expression simplifies to 2/3, a constant. For cases B2 and B3, the inequality on the derivative implies that, for the



(a) Case A: $\xi^c_{1,1} \leq \epsilon, \;\; \xi^c_{2,1} \leq \epsilon$

(b) Case B: $\xi^c_{1,1} \geq \epsilon, \;\; \xi^c_{2,1} \geq \epsilon$

(c) Case C: $\xi^c_{1,1} \geq \epsilon, \;\; \xi^c_{2,1} \leq \epsilon$

(d) Case D: $\xi^c_{1,1} \leq \epsilon, \;\; \xi^c_{2,1} \geq \epsilon$

Figure 4.3: The four different cases for the positions of the kernel centres, with regards to the values of $h(x_{1,i}, x_{2,j})$

global optimum, $x_{1,2} = 0$ and $x_{1,2} = 1$, respectively. Inserting that into the expression to be minimised considerably simplifies the problem, resulting in an optimum of 1/3 for both cases. For instance, for case B2,

$$\underline{\mathrm{E}}^{(i,B2)} = \min_{\substack{\epsilon \leq x_{2,1} \leq 1 \\ 0 \leq x_{2,2} \leq 1}} \frac{x_{2,1}}{x_{2,1} + x_{2.2}} = \frac{\epsilon}{\epsilon + 1} = \frac{1}{3} \ . \tag{4.58}$$

Cases C and D are equivalent by just swapping the variables, therefore it is enough to analyse one of them.

$$\underline{\mathrm{E}}^{(i,C)} = \underline{\mathrm{E}}^{(i,D)} = \min_{\substack{0 \leq x_{1,1} \leq \epsilon \\ \epsilon \leq x_{2,1} \leq 1 \\ 0 \leq x_{1,2} \leq 1 \\ 0 \leq x_{2,2} \leq 1}} \frac{(x_{1,1} + x_{1,2})x_{2,1} + 2x_{1,2}x_{2,2}}{(x_{1,1} + x_{1,2})(x_{2,1} + x_{2,2})} = \frac{1}{3} \ . \tag{4.59}$$

The same derivative used previously, $\frac{\partial}{\partial x_{1,2}}$, is now non-negative, which means that for the global optimum $x_{1,2} = 0$. Again substituting that into the expression being minimised results in a simpler minimisation problem, of the same form as Eq. (4.58), with minimum 1/3.

The global optimum is $\underline{\mathrm{E}}^{i} = 1/3$. Therefore, this is an example where the inequality in Theorem 4 is strict, as $\underline{\mathrm{E}}^{d} < \underline{\mathrm{E}}^{i}$.

4.5.3 Epistemic Cost-to-Go, V_k

For the value of V_1, one has to first find V_2. It is written as

$$V_2(\xi_1) = \min_{0 \leq x_{1,l} \leq 1} \frac{x_{1,2}h(\xi_1, x_{1,1}) + x_{1,1}h(\xi_1, -x_{1,2})}{x_{1,1} + x_{1,2}} \ . \tag{4.60}$$

There are three possible curves $h(\xi_1, \xi_2)$ for fixed ξ_1, which are found in $\xi_1 \leq 0$, $0 < \xi_1 \leq \frac{1}{2}$, and $\xi_1 > \frac{1}{2}$. For each of these curves the problem of finding V_2 can be reduced into a problem similar to Eq. (4.58). Each of these curves results in a different

value of V_2, which results in V_2 being the following piecewise constant function,

$$V_2(\xi_1) = \begin{cases} \frac{1}{3}, & \xi_1 \leq 0 \\ 1, & 0 < \xi_1 \leq \frac{1}{2} \\ 0, & \xi_1 > \frac{1}{2} \end{cases} \tag{4.61}$$

Then V_1 is

$$V_1 = \min_{0 \leq x_{2,l} \leq 1} \frac{x_{1,2} V_2(x_{1,1}) + x_{1,1} V_2(-x_{1,2})}{x_{1,1} + x_{1,2}} = \frac{1}{9} \, . \tag{4.62}$$

In summary,

$$\begin{aligned} \underline{\mathrm{E}}^d &= 0 \\ V_1 &= \frac{1}{9} \\ \underline{\mathrm{E}}^i &= \frac{1}{3} \end{aligned} \tag{4.63}$$

and thus for the example given one has strict inequalities for all theorems, i.e. $\underline{\mathrm{E}}^d < V_1 < \underline{\mathrm{E}}^i$.

4.6 Algorithmics

The formulation for the many-objective optimization problem in Chapter 2 with its transcription in Section 2.5, and the uncertainty quantification method in Section 3.4 are combined into an algorithm that is described in this section. It can be split into the computation of the epistemic cost-to-go V_k and the upper quantile $\overline{\nu}$ to obtain the training data for the surrogate models, followed by the use of these surrogate models to run the optimization algorithm.

4.6.1 Estimating the Epistemic Cost-to-Go

The formula that relates V_k to V_{k+1}, Eq. (4.15), resembles the formula for the lower expectation Eq. (3.2). The former can be turned into the latter with the substitutions

$$h(\xi) \leftarrow V_{k+1}(\mathbf{u}_{k+1:M}, \Phi_k(\mathbf{x}_k, \mathbf{u}_k, \boldsymbol{\xi}_k)) + l_k(\mathbf{x}_k, \mathbf{u}_k, \boldsymbol{\xi}_k) \tag{4.64}$$

$$\Xi \leftarrow \Xi_k \tag{4.65}$$

$$\boldsymbol{\xi} \leftarrow \boldsymbol{\xi}_k \tag{4.66}$$

$$\mathcal{Q} \leftarrow \hat{\mathcal{Q}}_k \tag{4.67}$$

For the specific case of families of distributions $\hat{\mathcal{Q}}_k$ defined based on interval-valued moments as in Section 3.4, this allows using the method in Section 3.4, namely Eq. (3.24), developed to compute $\underline{\mathrm{E}}$ for families $\mathcal{Q}$ defined by linear constraints, to instead compute V_k for $\hat{\mathcal{Q}}_k$ also defined with linear constraints. The estimate of $\underline{\mathrm{E}}$ made in Section 3.4, was referred to as $\hat{\underline{\mathrm{E}}}$. Here, when the substitutions in Eqs. (4.64) to (4.67) are made, the resulting estimate of V_k is referred to as $\hat{\underline{\mathrm{E}}}_k$.

This means that to estimate V_k for a particular set of inputs $\mathbf{u}_{k:M}$ and $\mathbf{x}_k$, one needs a sample of values of V_{k+1} with which to form the vector $\mathbf{h}_K$, and which are obtained by evaluating the expression in Eq. (4.64) for values of $\boldsymbol{\xi}_k$ in set the new set $\mathcal{K}_k$ of kernels centres in Ξ_k. Applied recursively, this would lead to an exponentially increasing number of function evaluations. To avoid this issue, instead of evaluating V_{k+1}, a surrogate model is built $\tilde{V}_{k+1}$, for $2 \leq k \leq M$. When computing V_M, since V_{M+1} is deterministic, V_{M+1} is sampled directly without there being a surrogate model for it. No surrogate model is built for V_1, for $k = 1$ one directly computes the upper-quantile $\overline{\nu}$ for a sample of values and uses the results to train the surrogate model $\tilde{\overline{\nu}}$ directly from $\tilde{V}_2$, more details in Section 4.6.2.

The process to obtain the training samples to train the surrogate model, and to then train and use them to solve the many-objective optimal control problem, is the subject of Section 4.6.3.

The computation of the V_k given the surrogate $\tilde{V}_{k+1}$, is shown in pseudo-code in Algorithm 3. It is a lower expectation of the values of V_{k+1}, as in Eq. (4.15), except

here it is formulated as a function of the proxy control variables $\mathbf{z}_k$, which means the control map must be called, in Line 1. For each objective ϕ_i in Eq. (2.28), there is a cost-to-go function $V_{k,i}$.

For $V_{M+1,i}$, used in line 5, the definition in Eq. (4.16) becomes

$$V_{M+1,i}\left(\mathbf{x}_{M+1};\nu_i\right) = L\left(\mathbf{x}_{M+1};\nu_i\right) \ . \tag{4.68}$$

The reason for the inclusion of parameter ν_i is to support the estimation of lower expectation of quantities of the form

$$L\left(\mathbf{x}_{M+1};\nu_i\right) = h_i\left(\mathbf{x}_{M+1}\right) < \nu_i \tag{4.69}$$

which is used to compute the upper quantile in the next section. Thus, when computing lower expectations with this definition for L, the value of the threshold ν_i for the particular objective is also an argument to $V_{k,i}$. Therefore, we have that $V_{k,i} = V_{k,i}\left(\mathbf{z}_k, \mathbf{x}_k; \nu_i\right)$. In addition, when estimating lower expectations of this form, one must make $l_k(\mathbf{x}_k, \mathbf{u}_k; \boldsymbol{\xi}_k) = 0$.

The lower expectation is estimated by solving the linear program in Eq. (3.24) with the above substitutions.

Line 2 obtains $\mathcal{K}_k$ by uniform random sampling. The objective vector $\mathbf{h}_K$ is then obtained by propagating the trajectory for all samples in $\mathcal{K}_k$, producing an array $\mathcal{X}_{k+1}$ of the states propagated to the next observation. In Lines 4-7, the vector $\mathbf{h}_K$ is computed. If $k = M$, the definition in Eq. (4.68) is used, otherwise the surrogate of the epistemic cost-to-go at the next step, $\tilde{V}_{k+1,i}$, is used. To reduce the computational cost, V_{k+1} is replaced by a Kriging surrogate model [43], $\tilde{V}_{k+1}$. Finally, Eq. (3.24) is solved using Matlab's® `linprog` function, with the *dual-simplex* solver, in Line 8.

4.6.2 Estimating the Upper Quantile

The upper quantile $\overline{\nu}$, in Definition 3.2.3, is estimated using Matlab's® root-finding algorithm `fzero` in Algorithm 4. Lines 1-3 of this algorithm do the same as Lines 1-3 of Algorithm 3, i.e., evaluate the control map, sample the uncertain space, and finally

Algorithm 3: EpistemicCostToGo

Input: $\mathbf{x}_k$, $\mathbf{z}_{k:M}$, ν_i, $\tilde{V}_{k+1,i}$
Output: $V_{k,i}$

1 $\mathbf{u}_{k:M} \leftarrow \mathbf{f}_{\text{CMap}}(\,\mathbf{x}_k,\,\mathbf{z}_{k:M})$
2 $\mathcal{K}_k \leftarrow$ `rand_sample(` Ξ_k`)`
3 $\mathcal{X}_{k+1} \leftarrow \Phi\left(\mathbf{x}_k, \mathbf{u}_k, \mathcal{K}_k\right)$
4 **if** $k = M$ **then**
5 $\quad \mathbf{h}_K \leftarrow L\left(\mathcal{X}_{M+1}; \nu_i\right)$
6 **else**
7 $\quad \mathbf{h}_K \leftarrow \tilde{V}_{k+1,i}\left(\mathbf{u}_{k+1:M}, \mathcal{X}_{k+1}, \nu_i\right) + l_k(\mathbf{x}_k, \mathbf{u}_k; \boldsymbol{\xi}_k)$
8 $V_{k,i}(\mathbf{u}_{k:M}, \mathbf{x}_k; \nu_i) \leftarrow \underline{\hat{\mathrm{E}}}_k\left(\mathbf{h}_K\right)$ `// Eq. (3.24)`

propagate to obtain a sample of the state. The function $f_{\underline{\mathrm{E}}}(\nu)$, obtained in Lines 4 and 5, computes the value of V_1 for each sample point in $\mathcal{X}_1$, similarly to Line 8 of Algorithm 3, except here it is treated as a function of ν, and from its value is subtracted $\hat{\mathrm{E}}$, so that the upper quantile is this function's root, which is found using Matlab's ® function `fzero`.

Algorithm 4: UpperQuantile

Input: Proxy control vector $\mathbf{z_2}$ and surrogate $\tilde{V}_{2,i}$
Output: Upper quantile $\overline{\boldsymbol{\nu}}_i$

1 $\mathbf{u} \leftarrow \mathbf{f}_{\text{CMap}}(\mathbf{z}, \mathbf{x}_1)$
2 $\mathcal{K}_1 \leftarrow$ `rand_sample(` Ξ_1`)`
3 $\mathcal{X}_2 \leftarrow \Phi\left(\mathbf{x}_1, \mathbf{u}_1, \mathcal{K}_1\right)$
4 $f_\nu(\nu) \leftarrow \tilde{V}_{2,i}(\mathcal{X}_2, \mathbf{z}_{2:M}; \nu)$
5 $f_{\underline{\mathrm{E}}}(\nu) \leftarrow \underline{\hat{\mathrm{E}}}_1(f_\nu)$ - $\hat{\mathrm{E}}$
6 $\overline{\nu}_i \leftarrow$ `fzero(`$f_{\underline{\mathrm{E}}}(\nu)$`) // Root of` $f_{\underline{\mathrm{E}}}(\nu)$

When estimating the upper quantile using a surrogate model in the call to `fzero`, this model's predictor function is repeatedly called with the same values of $\mathbf{x}_2$ and $\mathbf{z}_2$, but with varying ν. As described in Section 2.8, in this situation it is possible to save computational time by storing some intermediate results, in particular the components of the correlation coefficients that depend on $\mathbf{x}_2$ and $\mathbf{z}_2$.

4.6.3 Solving the Many-Objective Optimisation Problem

The entire process to obtain solutions to the many-objective (MO) optimisation problem is summarised in Algorithm 5, and partly in Fig. 4.4. Again to simplify the pseudocode, when a function is called with an array of values in place of a single value (represented by calligraphic symbols like $\mathcal{K}$, $\mathcal{Z}$, etc.), the function is being called on each element and the output is also an array.

A sample of proxy control variables $\mathcal{Z}$ is produced using Latin Hypercube Sampling from a Uniform distribution (LHSU) [47], in Line 1, and a sample of random variables $\mathcal{K}$ is obtained from uniform random sampling in Ξ in Line 2. These can be split into $\mathcal{Z}_k$ and $\mathcal{K}_k$, which are sets, respectively, of the component of the elements of $\mathcal{Z}$ and $\mathcal{K}$ that describe the k-th segment of the trajectory. Samples of the states at each observation $\mathcal{X}_k$ are produced in Lines 3-7, by, for all combinations of samples in $\mathcal{Z}_k$ and $\mathcal{K}_k$, calculating the control map, and propagating the states. This would lead an exponential increase with k in the number of sample points, but MACS's archival technique [56], briefly described in Section 2.6, is used to downsample N_V training points, keeping the number of sample points in $\mathcal{X}_k$ constant with k.

Surrogate models $\tilde{\mathbf{V}}_k$ are produced in lines 9-11 by calculating V_k using algorithm 3 and using the cost-to-go at the next step $\tilde{\mathbf{V}}_{k+1}$ on the training data, and using the results to train a Kriging model. This iterative process of starting at the end of the trajectory and going backwards to obtain the cost-to-go functions is analogous to the similar process used to obtain the cost-to-go in differential dynamical programming [53]. Besides $\mathcal{Z}$ and $\mathcal{K}$, the arguments used to compute V include the thresholds $\boldsymbol{\nu}$, which are in $\mathcal{N}$, an array of evenly spaced values between the minimum and maximum value of each Quantity of Interest (QoI). Because there are multiple quantities of interest (QoI), the surrogate model $\tilde{\mathbf{V}}_k$ has as output a vector, and is an approximation of $\left[V_{k,1}, \ldots, V_{k,N_{\text{QoI}}}\right]$, where N_{QoI} is the number of QoI's. The same is done in Lines 12 and 13 to obtain a model of the upper quantile $\tilde{\bar{\boldsymbol{\nu}}}$, using training data obtained with Algorithm 4. A flowchart of the part of Algorithm 5 that computes $\tilde{\bar{\boldsymbol{\nu}}}$ is shown in Fig. 4.4. The block labelled "to MACS loop" refers to Lines 14-20, explained hereafter.

This surrogate model is then passed to the MO optimiser MACS [56] in Line 15,

which returns an array of N_a solutions $\hat{\mathcal{Z}}$. These are then evaluated by computing the upper quantile again following Algorithm 3, and then added to the training set of the surrogate model (Lines 17 and 18). Similarly to previous work [14, 15], this process is repeated $N_{\mathrm{re-train}}$ times (Line 14).

The number of solution points returned by MACS is at most N_a, the size of the archive, which in this work also matches the number of agents.

Algorithm 5: Many-Objective Optimisation Under Epistemic Uncertainty

1 $\mathcal{Z} \leftarrow$ `LHSU(`Z`)`
2 $\mathcal{K} \leftarrow$ `rand(`Ξ`) // Uniform random sampling`
3 $\mathcal{X}_1 \leftarrow \mathbf{x}_1$
4 **for** $k \leftarrow 1$ ***to*** $M-1$ **do**
5 $\quad \mathcal{U}_k \leftarrow f_u\left(\mathcal{Z}_k, \mathcal{X}_k\right)$
6 $\quad \mathcal{X}_{k+1} \leftarrow \Phi\left(\mathcal{X}_k, \mathcal{U}_k, \mathcal{K}_k\right)$
7 $\quad \mathcal{X}_{k+1} \leftarrow$ `MacsArchiveShrink(`$\mathcal{X}_{k+1}, N_V$`)`
8 $\tilde{\mathbf{V}}_{M+1} \leftarrow \boldsymbol{h}$
9 **for** $k \leftarrow M$ ***to*** 2 **do**
10 $\quad \mathcal{V}_k \leftarrow$ `EpistemicCostToGo` $\left(\mathcal{X}_k, \mathcal{Z}_k, \mathcal{N}, \tilde{\mathbf{V}}_{k+1}\right)$ `// Algorithm 3`
11 $\quad \tilde{\mathbf{V}}_k \leftarrow$ `TrainKriging` $\left(\left[\mathcal{X}_k, \mathcal{Z}_k, \mathcal{N}\right], \mathcal{V}_k\right)$
12 $\overline{\mathcal{N}} \leftarrow$ `UpperQuantile` $\left(\mathcal{Z}, \tilde{\mathbf{V}}_2\right)$ `// Algorithm 4`
13 $\tilde{\bar{\nu}} \leftarrow$ `TrainKriging` $\left(\mathcal{Z}, \overline{\mathcal{N}}\right)$
14 **for** $i \leftarrow 1$ ***to*** $N_{\mathrm{re-train}}$ **do**
15 $\quad \tilde{\hat{\mathcal{Z}}} \leftarrow$ `MACS` $\left(\tilde{\bar{\nu}}\right)$ `// See [56]`
16 $\quad \tilde{\mathcal{N}} \leftarrow$ `UpperQuantile` $\left(\tilde{\hat{\mathcal{Z}}}, \tilde{\mathbf{V}}_2\right)$ `// Algorithm 4`
17 $\quad \mathcal{Z} \leftarrow$ `Join` $\left(\mathcal{Z}, \hat{\mathcal{Z}}\right)$
18 $\quad \overline{\mathcal{N}} \leftarrow$ `Join` $\left(\overline{\mathcal{N}}, \hat{\mathcal{N}}\right)$
19 $\quad \tilde{\bar{\nu}} \leftarrow$ `TrainKriging` $\left(\mathcal{Z}, \overline{\mathcal{N}}\right)$
20 $\left[\mathcal{Z}, \overline{\mathcal{N}}\right] \leftarrow$ `RemoveDominated` $\left(\left[\mathcal{Z}, \overline{\mathcal{N}}\right]\right)$

4.6.4 Computational Complexity and Scalability

This section discusses how the computational resources required by this method scale with the size of the input, i.e., its algorithmic complexity.

The memory requirements for training the Kriging surrogate model scale quadratically with the number of training points due to the calculation of the pairwise corre-

lation coefficients. The computational time scales cubically, due to the steps taken to compute the optimal regression coefficients, in particular, the cholesky factorization of the matrix of correlation coefficients. The number of training points is always greater than the number of inputs to the model, and in general should grow with it, so as to ensure good quality of fit. Since in this case, the number of inputs to the model is the number of uncertain variables, it follows that the computational time of this method scales with the cube of the number of uncertain variables.

However, this is a characteristic of the surrogate model chosen, whereas this method would work with any other surrogate model. The quantification of the epistemic uncertainty step when obtaining the training points is by far the most time consuming step. It too also scales with the number of uncertain variables, but only with those that fall within each segment. Therefore, by splitting the trajectory into small enough segments, this method would be scalable even if the solution to Eq. (4.15) were to scale badly with the number of uncertain variables, as it happens with some formulations using evidence theory, e.g. [13], where the computational cost of UQ scales exponentially with the number of uncertain variables.

In terms of computational resources required, the proposed method scales linearly with the number of segments the trajectory is split into, if this split is made without increasing the number of uncertain and control variables. This is because the trajectory splitting method allows independent handling of each segment. Since this split is made for the time instants where the observations take place, this method scales linearly with the number of observations.

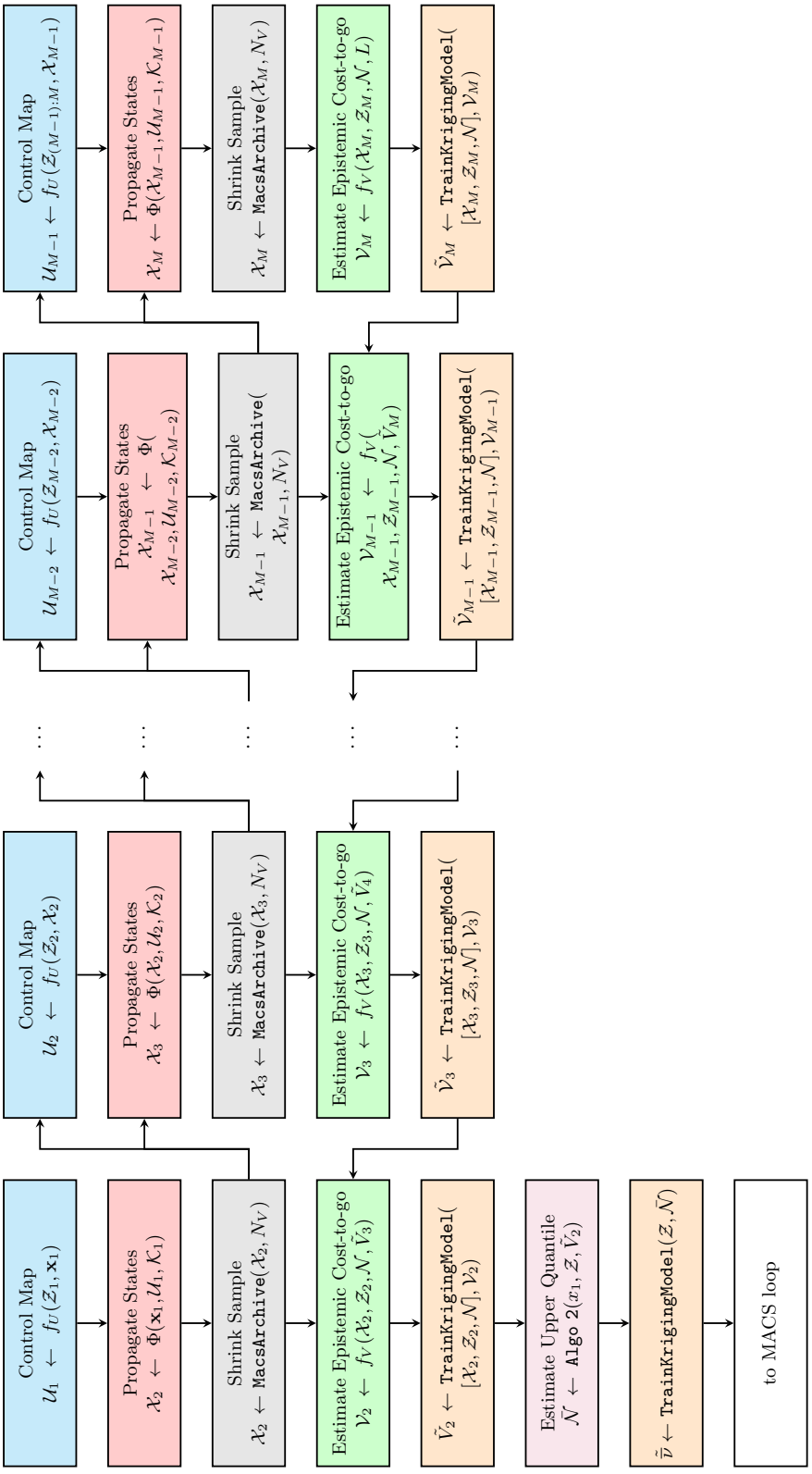


Figure 4.4: A flowchart of Lines 3-13 of Algorithm 5, where f_V refers to Algorithm 3

4.7 Conclusion

This chapter has proposed a novel method for epistemic uncertainty quantification that was specifically designed with trajectory optimisation problems in mind. This method turns the computation of the lower expectation from a search over the entire space of uncertain variables Ξ, into an iterative algorithm composed by smaller searches over spaces Ξ_k. This is advantageous when the complexity of the lower expectation calculation grows more than linearly with the number of variables.

Given its novelty, and the fact that the relationship between the introduced quantity V_k and the usual lower expectation is not immediately obvious, this chapter spent much time on theoretical developments. These show that, while it is not possible to say that V_k corresponds to the lower expectation under some definition, it is possible to bound it from below and from above with lower expectation values $\underline{\mathrm{E}}_d$ and $\underline{\mathrm{E}}_i$, where the first does not restrict the joint probability between variables $\boldsymbol{\xi}_i$ and $\boldsymbol{\xi}_j$ for $i \neq j$ while the latter assumes that each pair of such variables are independent.

The practical algorithmic aspects of its computation are also detailed, which included its main weakness, the use of surrogate models, that should be the subject of further investigation for the next developments in this approach.

Ultimately, one potential development that would follow naturally from this could not be carried out during this PhD work due to lack of time. This is the integration of the control law optimisation and the computation of the lower expectation into the same dynamic programming algorithm. Chapter 7, discussing future work, will cover this in more detail.

Part II

Applications

Chapter 5

Numerical Results for Epistemic Uncertainty Optimisation and Quantification

Some test cases for a spacecraft with a low thrust ion engine are used to apply the methodology described in Part I of this thesis. The trajectory is transcribed as described in Section 2.5. The uncertain parameters refer to the performance of the launcher that places the spacecraft in heliocentric orbit, and the performance of the spacecraft engine. The quantification of the epistemic uncertainty is as defined in Chapter 3, and results using the dynamic programming based uncertainty quantification of Chapter 4 are left for the next chapter. The specifics of the modelling of the uncertainty vary for each test case, as detailed in the following sections.

In all of these test cases, the model for the epistemic uncertainty is as defined in Section 3.3, so the distributions of the uncertain variables $\boldsymbol{\xi}$ belong to family $\mathcal{Q}_u$, given by Eq. (3.7). No additional bounds for the coefficients $c_{\mathbf{j}}$ were specified in the definition of the families of distributions. As shown also in Section 3.3, this is equivalent to considering only distribution functions in family $\mathcal{Q}_j$. Thus, all minimisations of all expectation functions were run over family $\mathcal{Q}_j$. The degree of the polynomials was $q_k = 4 \; \forall k \in \{1, \ldots, n_\xi\}$.

The objective functions considered in all three test cases were the required propel-

lant mass m_p, the distance to target Δr and, for test case A (Section 5.1), the relative velocity Δv between spacecraft and target at rendezvous is also considered. Test case B (Section 5.2) is based on an asteroid tour [21], designed to visit multiple asteroids in the solar system with a single mission, which was achieved by not placing any constraints on the relative velocity Δv at fly-by. For test case B, only m_p and Δr are considered. The thresholds $\nu_{(\cdot)}$ for these quantities were also optimised simultaneously with the other quantities, as in Eq. (2.37). The different control maps described in Section 2.7 and the threshold maps in Section 3.5 are tested and compared in Sections 5.4 and 5.5.

Real life asteroid missions use spacecraft with initial mass in the range between 500kg and 2000kg, see for example DART [2] (610 kg), Hayabusa2 [3] (609 kg), and OSIRIS-REx [1] (2110 kg). Therefore, for both test cases, the initial mass for the spacecraft was picked as some value in the middle, 1000kg. The engine parameters for both test cases differ and are described in the next sections.

The settings used for MACS and for the many-objective optimisation algorithm are as follows, except when testing other choices of parameters (for example in Section 5.5), wherein it is indicated what values these parameters take instead. The number of function evaluations for each run of MACS was n_f=10000 and the number of iterations $N_{re-train}$ was set to 10. The size of the archive N_a and the number of agents in the population N_p for MACS [56] were 10, and the number of solutions to be validated N_s before adding to the training data of the surrogate model was also 10.

5.1 Test Case A: Apophis Rendezvous

Apophis is an asteroid that in 2004 was classified as level 4 in the Torino scale, the highest ever score, due to a probability of impact that reached 1 in 60 at its highest estimate [72]. Later observations reduced this probability to the point that the asteroid was reclassified as level 0 in the Torino scale. Nonetheless, its orbit regularly intersects the Earth's, and on average it passes within two lunar orbits every five years [72].

This makes it an excellent test case for robust design of asteroid rendezvous missions, with the purpose of improving predictions about future approaches or even redirect it

if ever necessary. For such missions it is important to maximise the attainment of the objectives even in the presence of significant epistemic uncertainty.

The orbital elements for Apophis, for the epoch 28 September 2008, were taken from the JPL small object database [1]. The initial wet mass of the spacecraft m_0 was set to 1000 kg.

In this test case, the mission objective is a rendezvous with Apophis, so the relative velocity to target is also a quantity of interest, and therefore Eq. (2.37) becomes, for this problem,

$$\min_{\mathbf{z},\boldsymbol{\nu}} \begin{bmatrix} -\underline{\mathrm{E}}\left(m_p\left(\mathbf{f}_{\mathrm{CMap}}\left(\mathbf{z}\right)\right) < \nu_{m_p}\right) \\ -\underline{\mathrm{E}}\left(\Delta r\left(\mathbf{f}_{\mathrm{CMap}}\left(\mathbf{z}\right)\right) < \nu_{\Delta r}\right) \\ -\underline{\mathrm{E}}\left(\Delta v\left(\mathbf{f}_{\mathrm{CMap}}\left(\mathbf{z}\right)\right) < \nu_{\Delta v}\right) \\ \nu_{m_p} \\ \nu_{\Delta r} \\ \nu_{\Delta v} \end{bmatrix} \tag{5.1}$$

where $\underline{\mathrm{E}}\left(h(\mathbf{z}) < \nu_h\right)$ represents the lower expectation of $h < \nu_h$, an indicator function, where h is a quantity of interest (which can be m_p, Δr or Δv), and ν_h is the respective threshold.

This test case is subdivided into three sub-cases, which only differ in the specific definition of the epistemic uncertainty, as defined in the following.

5.1.1 Test Case A1: Uncertain Engine Parameters

In this test case the engine thrust T and I_{sp} were considered to vary with true longitude in a way that is uncertain. This is modelled with uncertain variables $\xi_{T,i}$ and $\xi_{I,i}$ representing the value of these parameters at equispaced longitudes. The values at intermediate longitudes were obtained by linear interpolation. The hyperbolic excess velocity v_∞ of the spacecraft as it leaves the Earth's gravity well was also considered

[1] https://ssd.jpl.nasa.gov/sbdb.cgi

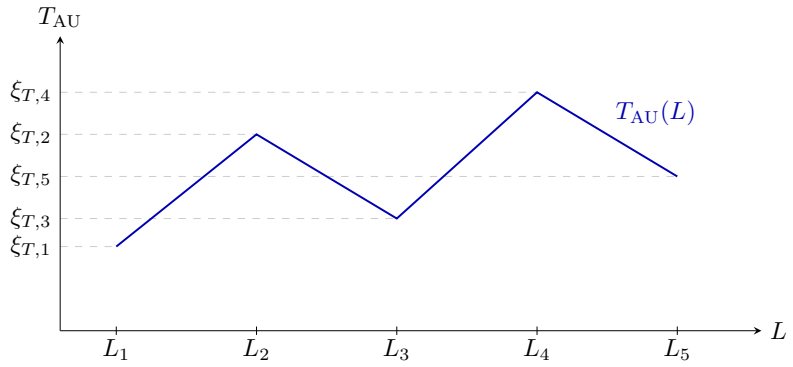


Figure 5.1: Example curve for T_{AU} for test case A1

uncertain. Therefore, variable $\boldsymbol{\xi}$ is given as

$$\boldsymbol{\xi} = [v_\infty \ \xi_{T,1} \ \cdots \ \xi_{T,n_T} \ \xi_{I,1} \ \cdots \ \xi_{I,n_T}], \tag{5.2}$$

and, referring back to Eqs. (2.20) and (2.21), η_i is given as:

$$\eta_i(L) = \begin{Bmatrix} \dfrac{L - L_{i-1}}{L_i - L_{i-1}} & \text{if } L_{i-1} < L \leq L_i \\ \dfrac{L_{i+1} - L}{L_{i+1} - L_i} & \text{if } L_i < L \leq L_{i+1} \\ 0 & \text{otherwise} \end{Bmatrix}, \tag{5.3}$$

where L_i are a sequence of n_T or n_I true longitudes equispaced and covering the whole trajectory. The result of defining η_i as in Eq. (5.3) is that the value of T_{AU} and I_{sp} at some true longitude L corresponds to the linear interpolation of the values of $\xi_{T,i}$ or $\xi_{I,i}$, respectively. An example of what this might look like for T_{AU} is in Fig. 5.1, which would be similar for I_{sp}. However, remember the relationship between the actual thrust T and T_{AU}, which is that $T = T_{\mathrm{AU}} r_{\mathrm{AU}}^2 / r^2$ where r is the heliocentric distance and r_{AU} is one astronomical unit, as conveyed by Eq. (2.4).

We considered $n_T = 5$ and $n_I = 4$ so that $n_\xi = 10$. This number was chosen to demonstrate the ability of this method to deal with high numbers of epistemic uncertain variables. In some tests, a variation with $n_\xi = 5$ is also considered. In that case, $n_T = 2$ and $n_I = 2$. The uncertain variables are bounded according to Table 5.2, where $\boldsymbol{\xi}^L$

and $\boldsymbol{\xi}^U$ represent the lower and upper bounds, such that $\Xi = \left[\boldsymbol{\xi}^L \; \boldsymbol{\xi}^U\right]$. The thrust T_{AU} and I_{sp} levels are around 0.05 N and 3000 s, respectively, which are representative of existing hardware. For example, the NEXT Ion Thruster is capable of 0.066 N of thrust with an I_{sp} of 2920 s when operated at 1.69 kW [63], the shaded row in Table 5.1. In Section 2.3, a modelling assumption was made that the thrust T would vary while I_{sp} would stay constant, as the power varies. While this was a choice driven by a desire to obtain simpler expressions for the integration of the dynamics, this was also argued with recourse on literature on the theory behind ion engines [26]. One also sees evidence that the NEXT engine can be operated in this way by consulting [63]. Table 5.1 contains a subset of Table 3 in [63], and by selecting the beam current and beam voltage as in the unshaded rows, one can keep the I_{sp} approximately constant while the power input changes.[2]

The ratio of the resulting acceleration to the Sun's gravitational acceleration at $r = 1$ AU is around 0.008, corresponding to low-thrust, the scenario for which the formulae used to propagate the trajectory in this paper were designed for. The values for v_∞ are based on those reported in [22].

Beam Current (A)	**Beam Voltage** (V)	**Power** (kW)	**Thrust** (mN)	**Isp** (s)
1.206	1010	1.51	61.4	2720
1.203	1790	2.43	81.4	3610
2.009	1560	3.52	127	3690
3.106	1390	4.81	185	3600

Table 5.1: A subset of Table 3 in [63], with NEXT ion engine performance data for different beam voltages and currents.

[2]The unshaded values correspond to a higher value of thrust - the data sheet shows mostly results with thrust values higher than the one we consider.

Table 5.2: Lower and upper bounds that define Ξ, the hyper-rectangular space of uncertain variables, for the test case A1.

	v_∞ [km/s]	$\xi_{I,i}$ [s]	$\xi_{T,i}$ [N]
$\boldsymbol{\xi}^L$	3.5	2850	0.052
$\boldsymbol{\xi}^U$	3.7	3150	0.058

5.1.2 Test Case A2: Engine Outage

A different type of unexpected engine behaviour is an engine outage, formulated as in Eq. (2.27). A single outage event is considered in this test case, thus the subscript i is dropped from the uncertain variables, which are the location of the beginning of the outage ξ_{L0}, its duration $\xi_{\Delta L}$, and intensity ξ_δ. For longitudes between ξ_{L0} and $\xi_{L0} + \xi_{\Delta L}$, the available engine thrust, which would be T without the outage, becomes $\xi_\delta T$. Variable v_∞ is still considered uncertain. The uncertain vector $\boldsymbol{\xi}$ for this test case is given as

$$\boldsymbol{\xi} = [v_\infty \ \xi_{L0} \ \xi_{\Delta L} \ \xi_\delta], \tag{5.4}$$

so the number of uncertain variables is $n_\xi = 4$.

As an example of engine outage being applied to a control vector, consider the case illustrated in Fig. 5.2. The curves represent the engine thrust as a fraction of its maximum value, with and without the effect of outage.

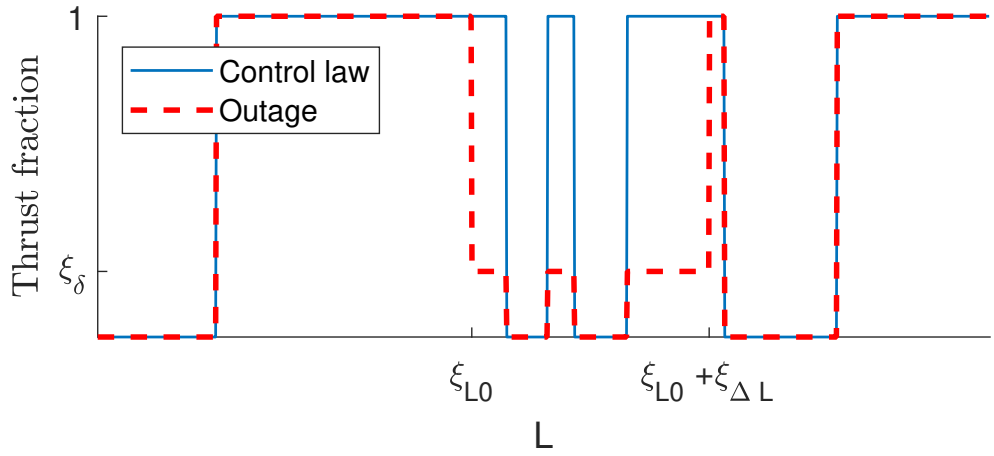


Figure 5.2: Example of the effect of engine outage as modelled in this work. Both lines represent the fraction of maximum available thrust as a function of the true longitude L. The blue solid line represents this fraction as specified by the control vector, and the red dashed line is the thrust that is actually applied due to the outage.

The bounds for $\boldsymbol{\xi}$ for this test case are represented in Table 5.3.

Table 5.3: Lower and upper bounds that define Ξ, the hyper-rectangular space of uncertain variables, for the engine outage test case, A2.

	v_∞ [km/s]	ξ_{L0} [rad]	$\xi_{\Delta L}$ [rad]	ξ_δ
$\boldsymbol{\xi}^L$	3.5	0	0	0
$\boldsymbol{\xi}^U$	3.7	4π	$\pi/8$	1

Table 5.4: Values of d_j used in the definition of the reduced uncertainty space Ξ_r in Eq. (5.5).

	v_∞	$\xi_{I,i}$	$\xi_{T,i}$
d_j	100 m/s	15 s	0.3 mN

5.1.3 Test Case A3: Uncertain Engine Parameters with Reduced Uncertainty Bounds

The formulation in Section 5.1.1 assumes that the uncertainty in system parameters is the same during the operational phase as it was during the design phase. However, this assumption can be considered to be over-pessimistic, as the knowledge of the system improves through the design, construction and testing phases. Hence, in this test case, the search space for the proxy control variable is kept the same as $Z = \Xi \times D_r \times D_v$, but the lower expectation is calculated with a reduced uncertainty space Ξ_r. The reduced uncertainty space is defined as the hyper-rectangle centred on the value of $\boldsymbol{\xi}$, used in the control map, and with a range that is 10 times smaller than the range of Ξ:

$$\Xi_r = \left\{ \hat{\boldsymbol{\xi}} \in \mathbb{R}^{n_\xi} : \hat{\boldsymbol{\xi}}_j = \boldsymbol{\xi}_j + b_j d_j \ \forall b_j \in [-1, 1] \ \forall j \in \{1, \ldots, n_\xi\} \right\}, \tag{5.5}$$

where the values of d_j are given in Table 5.4. This formulation simulates a scenario where the design variables can be chosen a priori, but during the trajectory their actual values can vary slightly about their design values.

5.2 Test Case B: Asteroid Tour

The developed methodology is applied to applied to a multi fly-by mission to a sequence of four near Earth asteroids (NEA), based on the tour proposed in [21][3], visiting 2006 UJ47, 2007 UV, 2005 YN176, and Ockeghem (Ock).

A case with 10 uncertain variables is considered ($n_\xi = 10$). These are the hyperbolic exhaust velocity at departure from Earth v_∞, the Thrust at 5 points in the trajectory T_i for $i = 1, \ldots, 5$ and the specific impulse at 4 points in the trajectory $I_{sp,i}$ for $i = 1, \ldots, 4$. Their nominal values, $\boldsymbol{\xi}_0$, are presented in the Table 5.5. The value of thrust or specific impulse at each point in the trajectory is obtained by linear interpolation, with the same η_i functions of Eq. (5.3).

Table 5.5: Nominal values for the uncertain variables.

	v_∞ [Km/s]	$I_{sp,i}$ [s]	T_i [N]
$\boldsymbol{\xi}_0$	5.846	3000	0.15

The uncertainty space is the hyper-rectangle $\Xi = [\boldsymbol{\xi}^L, \boldsymbol{\xi}^U]$, where $\boldsymbol{\xi}^L = 0.9\boldsymbol{\xi}_0$ and $\boldsymbol{\xi}^U = 1.1\boldsymbol{\xi}_0$, and $\boldsymbol{\xi}_0$ is the nominal value of the uncertain parameters, as defined in Table 5.5. These nominal values correspond to the same specific impulse as for test case A, and a thrust that is three times higher. The ratio of the resulting acceleration to the Sun's gravitational acceleration is around 0.025. The NEXT Ion Thruster can also produce thrust at this level when run at a higher power, outputting 0.149 N of thrust with an I_{sp} of 3280 s using 3.64 kW [63]. The nominal excess velocity v_∞ was taken directly from [21]. For the control law there are $n_{LT} = 20$ thrust and coast arcs.

The optimal trajectory for this nominal case is displayed in Figure 5.3.

The many-objective problem for this test case consists in maximising the lower expectation that the mass and the distance to each target are below the respective threshold, and the minimisation of these thresholds:

$$\min_{\mathbf{z},\boldsymbol{\nu}} \begin{bmatrix} -\underline{\mathrm{E}}(m_p(\mathbf{f}_{\mathrm{CMap}}(\mathbf{z}), \boldsymbol{\xi}) < \nu_m) & , & \nu_m \\ -\underline{\mathrm{E}}(\Delta r_i(\mathbf{f}_{\mathrm{CMap}}(\mathbf{z}), \boldsymbol{\xi}) < \nu_i) & , & \nu_i \text{ for each target } i \end{bmatrix}, \tag{5.6}$$

[3]This trajectory corresponds to "T2" in Table 7 of the cited paper

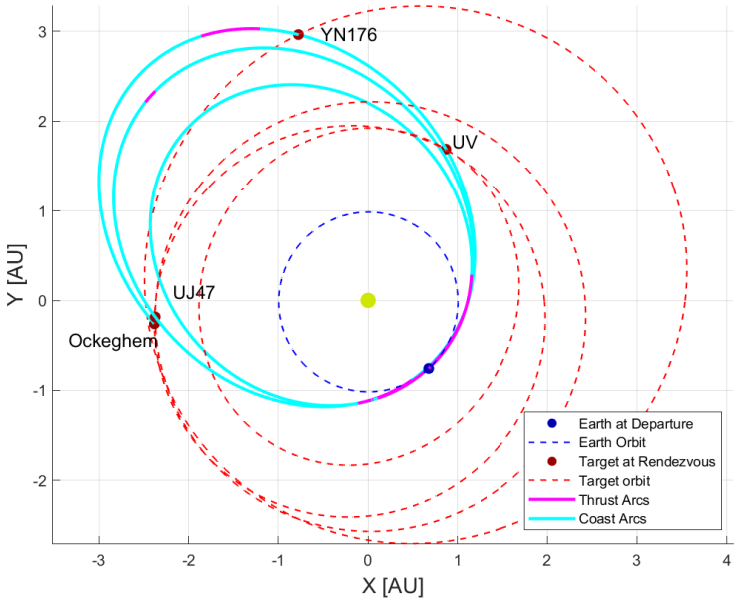


Figure 5.3: Plot of a trajectory that visits flies by the asteroids in the asteroid tour. Obtained for the nominal case.

where Δr_i is the distance to target i, and ν_i is the threshold on this quantity.

Because these targets are all fly-by only, in the control maps of Section 2.7, the constraints on velocity are not applied, and the displacements in velocity are not part of the proxy control variables $\mathbf{z}$.

In order to use with the dynamic programming based method of Chapter 4, it is convenient that each value of $T(L)$ and $I_{sp}(L)$ only depends on one value of $\xi_{T,i}$ and $\xi_{I,i}$, respectively. As such, for this test case, one has a piecewise-constant T_{AU} and I_{sp}, corresponding to

$$\eta_i(L) = \begin{Bmatrix} 1 & \text{if } L_i < L \leq L_{i+1} \\ 0 & \text{otherwise} \end{Bmatrix}, \tag{5.7}$$

with an example in Fig. 5.4.

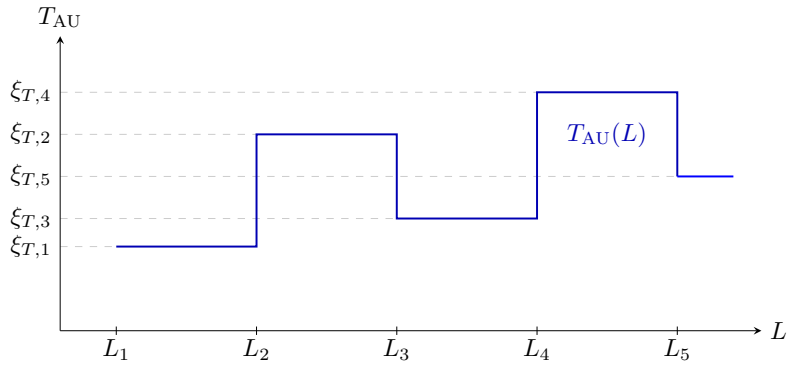


Figure 5.4: Example curve for T_{AU} for test case B

5.3 Lower Expectation Algorithm Comparison

In this section, the algorithms for the estimation of the lower expectation that were presented in Section 3.3.2 are compared in terms of computational cost and accuracy.

For test cases A1, A2, and B, $N_{\mathrm{init}} = 100$ initial guess solutions were generated to build the initial surrogate model, followed by $N_{\mathrm{re-train}} = 10$ iterations, or calls to MACS, with $N_p = N_a = N_s = 10$, giving a total of 200 evaluations of the lower expectation for each test case. If dominated solutions are not removed this gives, for each test case, a sample of 200 pairs of $\mathbf{z}$, $\boldsymbol{\nu}$ which is representative of a typical run of Algorithm 1 when applied to the case studies in this work. Given the similarity of test case A3 with test case A1, the analyses in this Section do not include case A3. However, the next sections will present specific results also on this test case. For each pair of $\mathbf{z}$, $\boldsymbol{\nu}$, the values of $\underline{\mathrm{E}}(h_i < \nu_i)$ were estimated using each of the proposed methods, for each quantity of interest (QoI) h_i, using the methods in Section 3.3.2. Since for test cases A1 and A2 there are 3 QoIs, three values of the expectation were estimated per pair $\mathbf{z}$, $\boldsymbol{\nu}$, corresponding to a total of 600 values of the lower expectation for each test case. For test case B, where there are 5 QoIs, there are a total of 1000 values of the lower expectation.

5.3.1 Pattern Search Variants

In Tables 5.6, 5.7, and 5.8, the variants of the pattern search algorithm are compared for test cases A1, A2, and B, respectively. In these tables, the error rate is the percentage of pairs $\mathbf{z}$, $\boldsymbol{\nu}$ for which each method does not obtain a value of E higher than the lowest known minimum by less than 10^{-6}, for each QoI. This lowest known minimum value of E is the putative global minimum and is estimated by taking the best estimate of $\underline{\mathrm{E}}$ from all algorithms being tested, including the ones in Section 5.3.3. The average number of evaluations of $\mathrm{E}(I;\mathbf{j})$, N_{E}, is used to measure the efficiency of each algorithm. Due to memoisation (see Section 3.6), estimates of E used when computing the lower expectation for one of the QoIs, are used also for the other QoIs, so N_{E} is the average number of expectation evaluations to estimate all three lower expectations, $\underline{\mathrm{E}}(m_p)$, $\underline{\mathrm{E}}(\Delta r)$ and $\underline{\mathrm{E}}(\Delta v)$, for each pair $\mathbf{y}$, $\boldsymbol{\nu}$. The greedy initialisation in Eq. (3.20) is computed by estimating the expectation for each possible marginal distribution $b_{i,q}$ on all uncertain variables. The starting multi-index $\mathbf{j}^0$ is then formed by choosing for each dimension the marginal that minimises the expectation. This results in an extra $(q+1)n_\xi$ evaluations of the expectation, which, for the methods that use this initialisation strategy (G and G+S), is included in the value of N_{E} reported in Tables 5.6 and 5.7.

The letter G in Tables 5.6, 5.7, and 5.8 refers to the greedy initialisation in Eq. (3.20), and the letters G+S refer to the re-initialising in Eq. (3.21), the symmetric restart. R5 and R20 refer to initialising with 5 and 20 random starting points, respectively. For the initialisation G and G+S we compared both neighbourhoods N_k^S and N_k^B. However, since N_k^B was consistently producing better results, both tests with random initialisations R5 and R20 used only neighbourhood N_k^B.

The results on Tables 5.6, 5.7, and 5.8 show that the G+S initialisation is the one that consistently produced the best results. Furthermore, it used significantly fewer function evaluations than R5 and R20. Note that the G and G+S strategies are not necessarily globally convergent. Thus one can quantify the computational effort to converge locally but cannot guarantee global convergence, even asymptotically. In Section 5.3.2 it is shown that a random restart can provide global convergence in

probability. Table 5.7 shows the results for case A2. In this case, there are only 4 uncertain variables, thus fewer evaluations of E are required to converge to a minimum. However, this problem appears to have a more complex landscape and thus the error rates are higher. Table 5.8, with results for test case B, allows similar conclusions, with the G+S initialisation with neighbourhood N_k^B producing more consistent results in terms of accuracy. The value of N_E is now significantly higher, a consequence both of the higher number of QoI, but also of the higher complexity of this problem. The results in the last column, using the genetic algorithm, are discussed in Section 5.3.3.

Table 5.6: Comparison of pattern search variants proposed in Section 3.3.2 for test case 1, with 10 uncertain variables. The error rate is the percentage of pairs $\mathbf{z}$, $\boldsymbol{\nu}$ for which each method does not obtain the minimum value of E within $\pm 10^{-6}$, this minimum being computed across all algorithms being tested.

Starting point		G		G + S		R5	R20
Neighbourhood		N_k^S	N_k^B	N_k^S	N_k^B	N_k^B	N_k^B
Error rate	$\underline{\mathrm{E}}(m_p)$	5.0%	4.0%	1.5%	**1.0%**	2.5%	2.0%
	$\underline{\mathrm{E}}(\Delta r)$	3.0%	2.0%	3.0%	**0.5%**	1.0%	**0.5%**
	$\underline{\mathrm{E}}(\Delta v)$	5.5%	2.0%	5.5%	**0.0%**	1.5%	**0.0%**
	average	4.5%	2.7%	3.3%	**0.5%**	1.7%	0.8%
N_{E}		113.3	177.1	203.4	390.3	1263.0	7469.6

Table 5.7: Comparison of pattern search variants proposed in Section 3.3.2, as in Table 5.6, for test case A2

Starting point		G		G + S		R5	R20
Neighbourhood		N_k^S	N_k^B	N_k^S	N_k^B	N_k^B	N_k^B
Error rate	$\underline{\mathrm{E}}(m_p)$	16.5%	3.5%	10.0%	**1.0%**	2.0%	1.5%
	$\underline{\mathrm{E}}(\Delta r)$	20.5%	4.5%	14.0%	**2.5%**	5.5%	**2.5%**
	$\underline{\mathrm{E}}(\Delta v)$	24.5%	8.0%	14.5%	**2.5%**	8.5%	5.0%
	average	20.5%	5.3%	12.8%	**2.0%**	5.3%	3.0%
N_{E}		54.4	77.7	110.1	139.5	212.3	472.3

Table 5.8: Comparison of pattern search variants proposed in Section 3.3.2 for test case B

Starting point		G		G + S		R5	Genetic
Neighbourhood		N_k^S	N_k^B	N_k^S	N_k^B	N_k^B	-
Accuracy	$\underline{\mathrm{E}}^{(m_p)}$	10.0%	4.5%	10.0%	**1.0%**	**1.0%**	5.5%
	$\underline{\mathrm{E}}^{\mathrm{UJ47}}$	2.0%	0.5%	1.5%	**0.0%**	0.5%	**0.0%**
	$\underline{\mathrm{E}}^{\mathrm{UV}}$	3.0%	1.5%	1.5%	**0.0%**	1.0%	**0.0%**
	$\underline{\mathrm{E}}^{\mathrm{YN176}}$	3.0%	1.5%	1.5%	0.5%	0.5%	**0.0%**
	$\underline{\mathrm{E}}^{\mathrm{Ock}}$	6.0%	2.5%	3.5%	**1.0%**	**1.0%**	**1.0%**
	average	4.8%	2.1%	3.6%	**0.5%**	0.8%	1.3%
N_{E}		320.7	486.8	590.1	1049.5	15054	2553

5.3.2 Experimental Convergence Analysis of the Pattern Search Algorithm

In this section we present an experimental analysis of the convergence of the pattern search algorithm with N_k^B and a random initial point $\mathbf{j}_0$. We call this method R1. We performed 200 independent runs of method R1, each with a different randomly chosen starting $\mathbf{j}_0$ and recorded the value of the solution to which each run converged, on data from test case A1.

Firstly, the problem appears to have a small number of basins of attraction. As shown in Fig. 5.5, for most of the expectation values, there were 3 or fewer different outcomes. Therefore, for each expectation value, the density function $p(e)$ most appropriate to represent the outcomes e of each search, started from a random initial solution, is a multinomial distribution. The probabilities of each outcome e are estimated for each pair $\mathbf{z}$, $\boldsymbol{\nu}$ by their relative frequency in the 200 random restart test.

For each pair $\mathbf{z}, \boldsymbol{\nu}$, we computed the probability $P(e < \underline{\mathrm{E}} + \delta)$ for an array of values of δ. Figure 5.6 presents the resulting 90% confidence lower bounds for this probability, i.e. the values r such that $P(e < \underline{\mathrm{E}} + \delta) \in [r, 1]$ with 90% probability. Also note that if N_R random restarts are performed, where we take the best estimate $\hat{e}$, we have $P(\hat{e} < \underline{\mathrm{E}} + \delta) = 1 - (1 - P(e < \underline{\mathrm{E}} + \delta))^{N_R}$.

Thus, for test case A1, the pattern search algorithm can estimate the global lower expectation with a maximum error of 20% with probability 0.95. Note, however, that the error quickly decreases to 5% with probability 0.87. This is true in the case of a

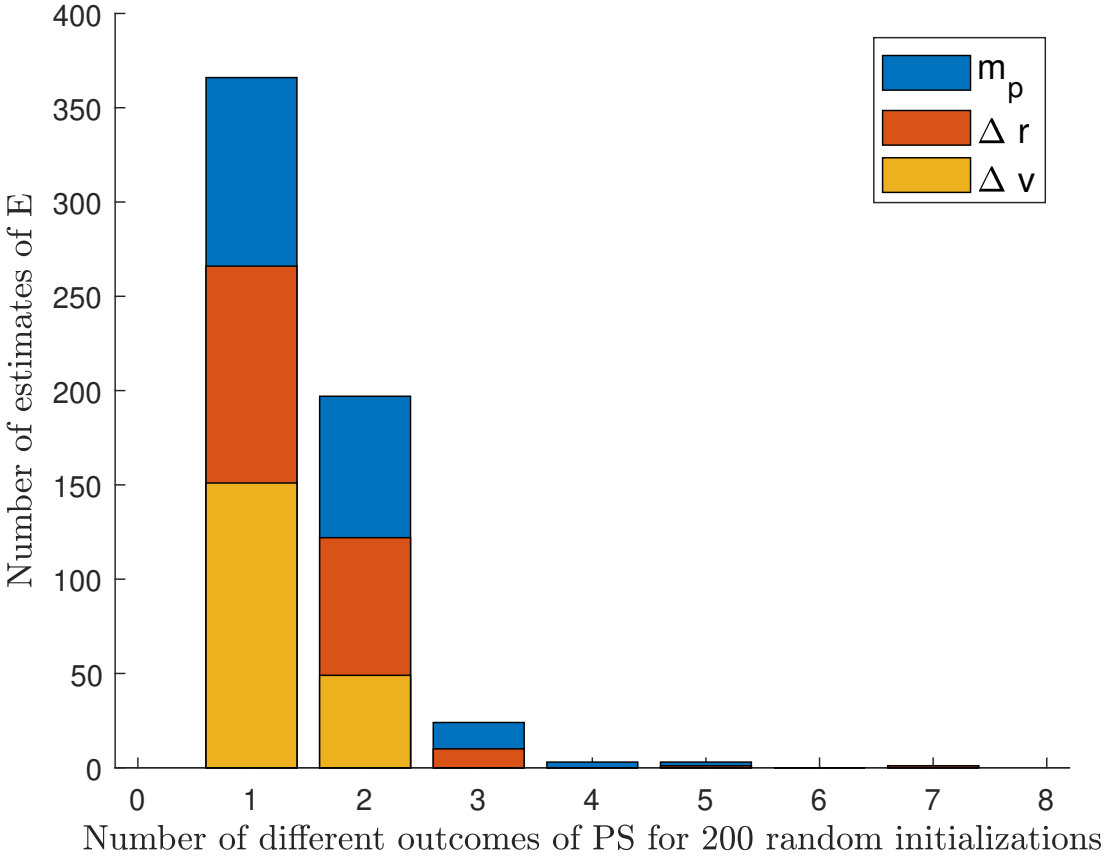


Figure 5.5: Histogram of the number of different outcomes obtained by the 200 random restarts with pattern search (PS). These results were obtained for 200 pairs $\mathbf{z}, \boldsymbol{\nu}$, with 3 QoIs, producing 600 estimates of the expectation, for test case A1. The results are segregated for each QoI.

random restart of the pattern search but we have empirically shown that the greedy + symmetric initialisation is more robust than a random start (Tables 5.6 and 5.7) thus Fig. 5.6 is providing an empirical lower bound on the probability of identifying the global lower expectation.

5.3.3 Other Heuristic Search Algorithms

Finally, we compared the performance of other heuristic search algorithms, namely simulated annealing (SA), taboo search (TS) and Matlab's genetic algorithm (GA).

The implementation of simulated annealing follows [52] by using the parameters suggested in [66] and having $5 \times |N_k|$ moves per iteration, where $|N_k|$ is the size of

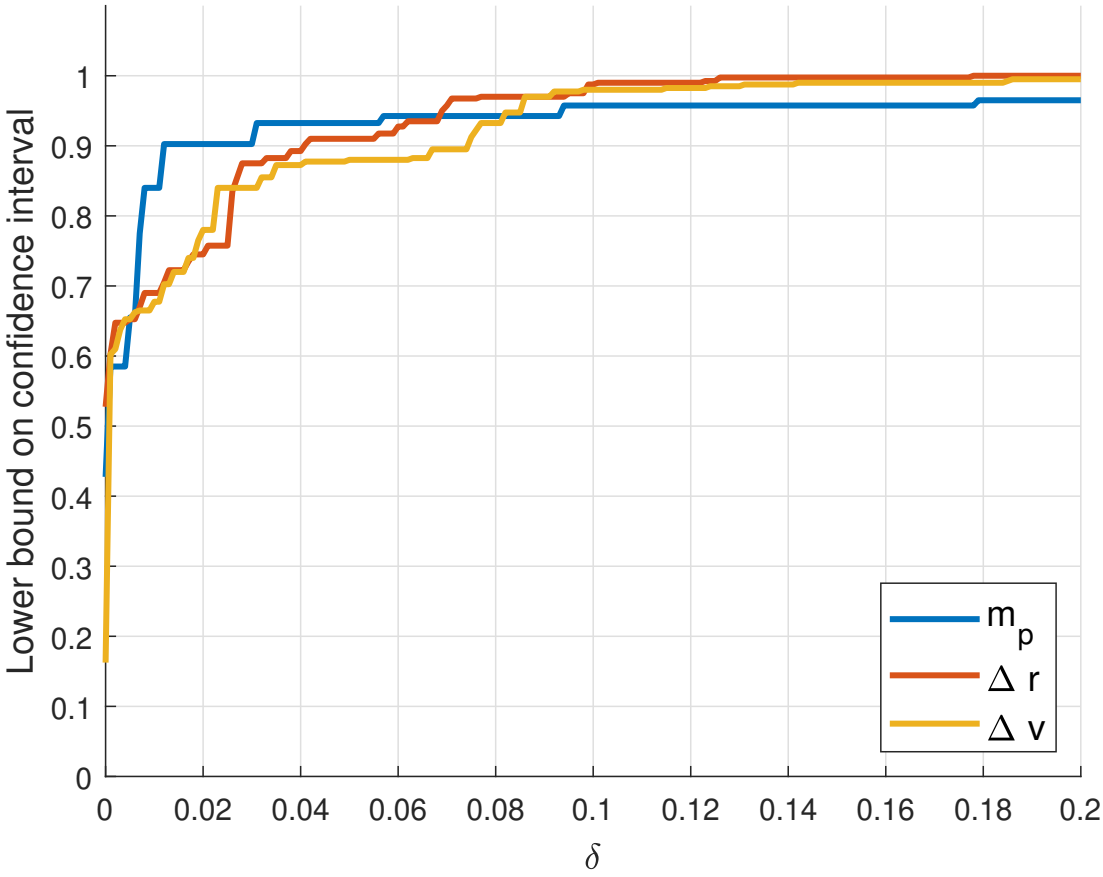


Figure 5.6: Curves of 90% confidence lower bounds for lower expectations on each of the QoIs. The plots represent r such that $P(e < \underline{\mathrm{E}} + \delta) \in [r, 1]$ with 90% probability (i.e. for 90% of pairs $\mathbf{z}$, $\boldsymbol{\nu}$) where e is the outcome of pattern search from a random initial point, and $\underline{\mathrm{E}}$ is the best estimate of the lower expectation available. This data was obtained for test case A1.

the neighbourhood. The taboo search implementation is completely deterministic, the point in the neighbourhood that minimises E and is not on the taboo list is always chosen. For this reason, the taboo list has no size limit. Both of these algorithms are run for both N_k^B and N_k^S. Because the lower expectation is always greater than or equal to zero, whenever any of these algorithms finds a zero, the search is terminated. Matlab's genetic algorithm is run with the default parameters, i.e., a uniformly random initial population with 50 elements, the scattered crossover function with 0.8 crossover fraction, and a Gaussian mutation function. The results are in Tables 5.9 and 5.10.

These results demonstrate the efficiency, robustness and accuracy of the pattern

search algorithm with symmetric re-start. For test case A1, in Table 5.9, the pattern search algorithm with symmetric restart (G+S) and neighbourhood N_k^B performs better in accuracy and efficiency than all non-pattern search algorithms. For the outage scenario, Table 5.10, the taboo search has the smallest error rate, but it is evaluating almost $5^4+4\times5$ values of E, which corresponds to the cardinality of $\mathcal{P}_j$ plus the number of integrations required for the initial guess, in Eq. (3.20), so this algorithm is close to a brute-force search.

Table 5.8 shows results for test case B, and only include the genetic algorithm in addition to the pattern search results. For this test case, the genetic algorithm is more competitive, achieving an average error rate of 1.3% but still performing worse both in terms of accuracy and in terms of N_{E} than G+S with neighbourhood N_k^B.

Table 5.9: Comparing the best pattern search algorithm variants (PS G+S) with other algorithms, namely simulated annealing (SA), taboo search (TS) and genetic algorithm (GA), for test case A1.

Method		PS G+S		SA		TS		GA
Neighbourhood		N_k^S	N_k^B	N_k^S	N_k^B	N_k^S	N_k^B	N_k^B
Error rate	$\underline{\mathrm{E}}(m_p)$	1.5%	**1.0%**	4.0%	4.5%	4.5%	4.0%	2.0%
	$\underline{\mathrm{E}}(\Delta r)$	3.0%	**0.5%**	4.0%	2.5%	1.5%	1.0%	1.0%
	$\underline{\mathrm{E}}(\Delta v)$	5.5%	**0.0%**	4.5%	3.5%	2.0%	1.0%	2.0%
	average	3.3%	**0.5%**	4.2%	3.5%	2.7%	2.0%	1.7%
N_{E}		203.4	390.3	4884.9	6623.7	4015.9	5953.3	10165.9

Table 5.10: Comparing the best pattern search algorithm variants (PS G+S) with other algorithms, namely simulated annealing (SA), taboo search (TS) and genetic algorithm (GA), for test case A2.

Method		PS G+S		SA		TS		GA
Neighbourhood		N_k^S	N_k^B	N_k^S	N_k^B	N_k^S	N_k^B	N_k^B
Accuracy	$\underline{\mathrm{E}}(m_p)$	10.0%	1.0%	6.0%	0.0%	1.0%	**0.0%**	2.0%
	$\underline{\mathrm{E}}(\Delta r)$	14.0%	2.5%	10.5%	2.5%	**0.0%**	**0.0%**	3.0%
	$\underline{\mathrm{E}}(\Delta v)$	14.5%	2.5%	12.5%	7.0%	**0.5%**	**0.5%**	6.0%
	average	12.8%	2.0%	9.7%	3.2%	0.5%	**0.2%**	3.7%
N_{E}		110.1	139.5	233.8	401.0	616.8	629.5	431.1

5.3.4 Scalability with the Number of Uncertain Variables

This section is concerned with the experimental assessment of the number of evaluations of expected values E that are required to find the putative global minimum $\underline{\mathrm{E}}^*$ with error less or equal than 10^{-6}, as the number of uncertain variables increases. Figure 5.7 shows the average number of evaluations of E (over all 200 solutions evaluated in test case 1) as a function of n_ξ, for the two initialisations G and G+S in Table 5.6. Note that while initialisation strategy G scales better than G+S, because there is no restart, the percentage of times it converges to the global minimum is lower, as Tables 5.6, 5.7 and 5.8 show.

Figure 5.7 also represents the limit case given by Eq. (3.17), which is the number of evaluations of the expectation E that one would need to perform to have a total number of samples with P-sampling equal to the number of samples that would be required with uniform sampling. Note that for the sake of clarity of representation, the black line is limited to $n_\xi \leq 5$.

5.4 Threshold Map Comparison

This section contains tests of the threshold mapping methods discussed in Section 3.5, on test cases 1A, 1B, and 1C. The values of $\underline{h}$ and $\overline{h}$ were calculated with Matlab's

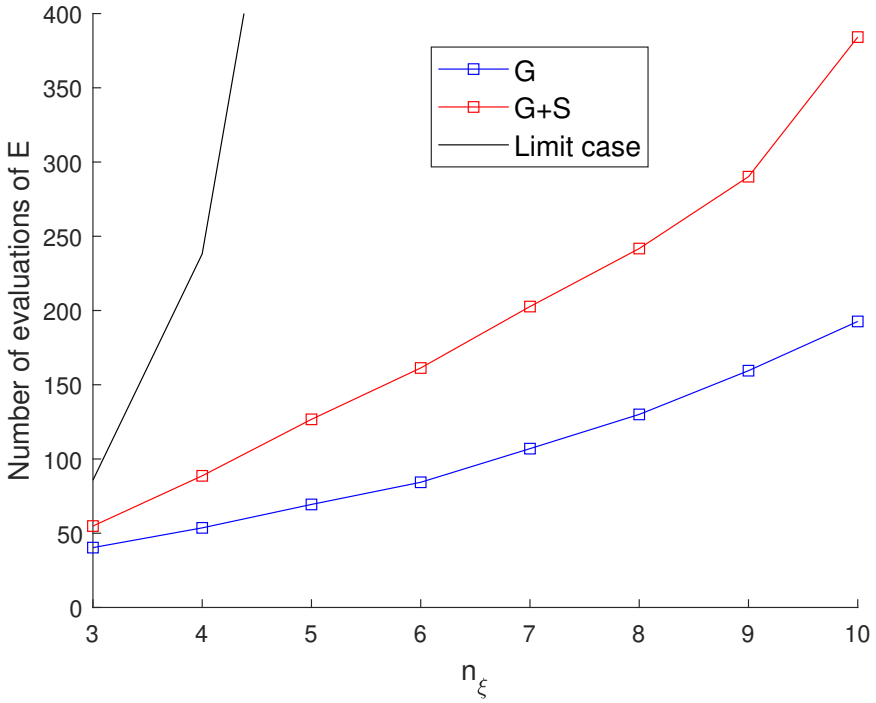


Figure 5.7: Average number of evaluations of $\mathrm{E}(I, \mathbf{j})$, for test case 1, for the pattern search algorithms with N_k^B with only the greedy start (G, in blue line) and with the greedy start followed by the symmetric restart (G+S, in red line). The black line is the limit number of calls to the expectation E given by Eq. (3.17).

fmincon-sqp, starting from the minimum and maximum of 5000 samples drawn from a Halton sequence. The evolution with the number of iterations of four MO optimisation metrics and of the accuracy of the surrogate model for different threshold mapping methods was compared. Figures 5.8 and 5.9 show test case 1A with 10 uncertain variables, Figs 5.10 and 5.11 are the same case with 5 uncertain variables, Figs 5.12 and 5.13 show test case 1B from Section 5.1.2 which has 4 uncertain variables, and finally, Figs 5.14 and 5.15 show the results for test case A3 from Section 5.1.3 with 10 uncertain variables.

The hypervolume was estimated with quasi-Monte Carlo sampling with 10^6 samples and a reference point obtained as the maximum value of each minimisation of the objectives for the points obtained in all the methods being compared. We also measured the generational distance (GD), the Hausdorff distance and the modified in-

verse generational distance (IGD+) [8]. The reference Pareto samples used to calculate these quantities were the combination of all non-dominated samples obtained from all methods being compared.

As can be seen from Figs. 5.8, 5.10, 5.12 and 5.14, the threshold mapping methods offer a statistically significant improvement of the convergence to the global Pareto front and a faster convergence. Without threshold mapping, the many-objective metrics barely improved with each iteration. Overall, *thMap-whole* appears to be the more efficient threshold mapping algorithm.

The model accuracy was measured as the root mean squared error (RMSE) between the model prediction and the exact value of the lower expectations for the values obtained in the last iteration for each method. This is plotted in Figs. 5.9, 5.11, 5.13 and 5.15. The reason for this choice is that the surrogate model will only converge for regions near the estimated Pareto front, since those are the training points. Because this is a Kriging model, the error for any of the training points is zero, which is why only points from the last iteration are included - otherwise the RMSE would trivially decrease as the points from previous iterations get added to the training set. This is also why the last iteration always has error equal to zero.

These tests show that, for the *thMap-whole* method, the error in the estimation of the lower expectation after 10 iterations of the surrogate was below 10% on average and in some cases even lower than 5%. This level of approximation was deemed a sufficiently accurate result considering the computational cost. An exception was the outage test case, where the surrogate model errors were much larger, Fig. 5.13, perhaps due to the discontinuous nature of the uncertainty. Here, the *thMap-whole* method produced much larger errors, while the others approach the 10% value (except the no threshold map version for $\underline{\mathrm{E}}(\Delta v)$). However, even here it is *thMap-whole* that produced the best results in terms of the multi-objective metrics, which were calculated using the real values, not values obtained using surrogates.

The no threshold map method tends to produce the worst results in terms of MO metrics, see Fig. 5.8. In most of the cases it also produced the lowest RMSE on $\underline{\mathrm{E}}(m_p < \nu_m)$, but the highest RMSE on the other QoIs, see Figs 5.9 to 5.15. The *thMap-*

dist method tended to have only slightly worse results than *thMap-whole*, and thus can be seen as an acceptable trade-off between MO metric convergence and computational complexity. In test case 2, *thMap-dist* produced better surrogate models than *thMap-whole* (Fig. 5.13), but worse MO metrics, as shown in Fig. 5.12.

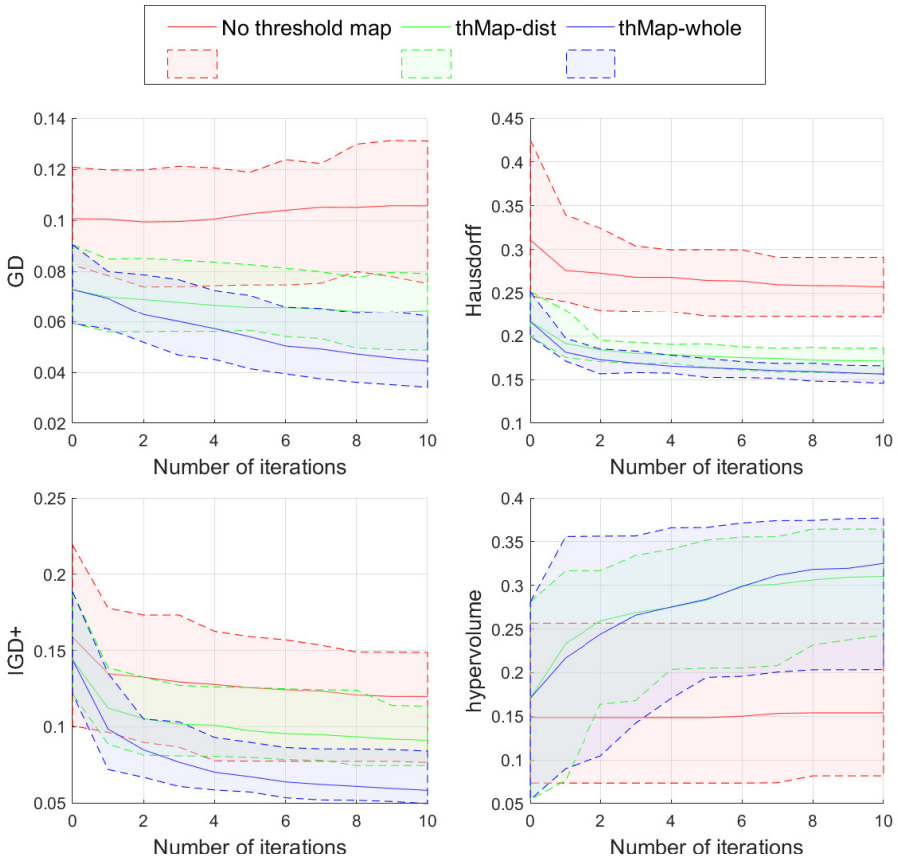


Figure 5.8: Pareto sample metrics as a function of the number of iterations. We include the generational distance (GD), the modified inverted generational distance (IGD+), the Hausdorff distance, and the hypervolume. Apart from the hypervolume, smaller value is better. The value for 0 iterations corresponds to the metrics for the initial training points. The shaded areas enclose the observed values of these quantities for 90% of 50 trials and the solid lines show the median.

It is possible for a method to produce better solutions in terms of many-objective metrics, while producing worse results in the RMSE for the surrogate. A method that obtains a wider variety of points will also make it more difficult for the surrogate model to fit the data. For instance, a method that only produced points in a very small set, would result in a surrogate model with good accuracy but the quality of its Pareto

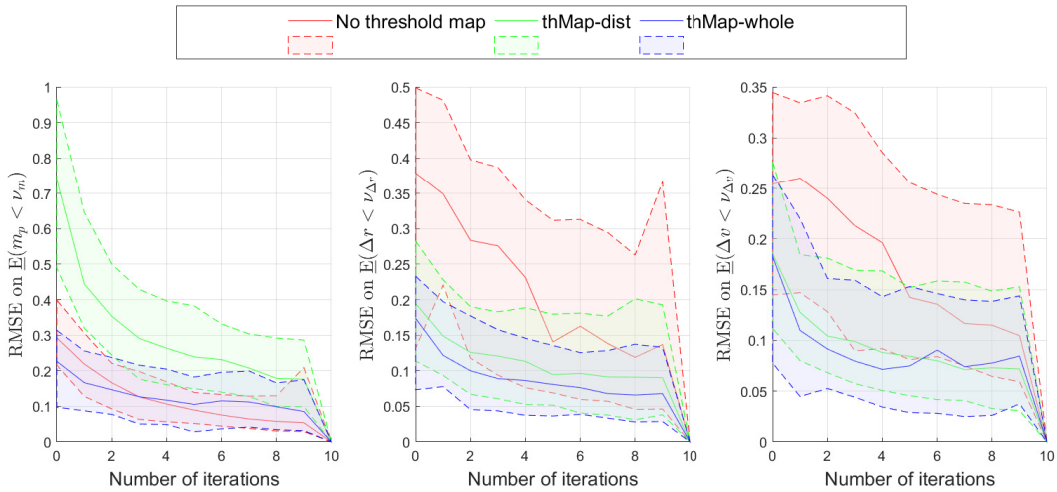


Figure 5.9: Plots of the RMSE of the surrogate model, at each iteration, for the values obtained at the last iteration. Four runs are presented here, each a line on each subplot. The sub-figures, from left to right, contain the RMSE on the surrogates of the propellant mass m_p, the distance to target Δr and the relative speed to target Δv. Again the shaded areas enclose the observed values of these quantities for 90% of 50 trials and the solid lines show the median.

samples would be very low. Thus, when comparing these methods, the factors that decide between better and worse methods are the MO metrics, not the surrogate model accuracy, since it is the former which measures the quality of the resulting Pareto set.

Comparing the results obtained in test case A1 with $n_\xi = 5$ and with $n_\xi = 10$ gives insight into the effect of n_ξ on the quality of the results. The difference does not appear significant, either in terms of the evolution of the MO metrics, Figs 5.8 and 5.10, or the RMSE of the surrogate models, Figs 5.9 and 5.11. The only substantive difference is that the range of values of the RMSE for the *thMap-whole* method is higher with $n_\xi = 10$ than with $n_\xi = 5$.

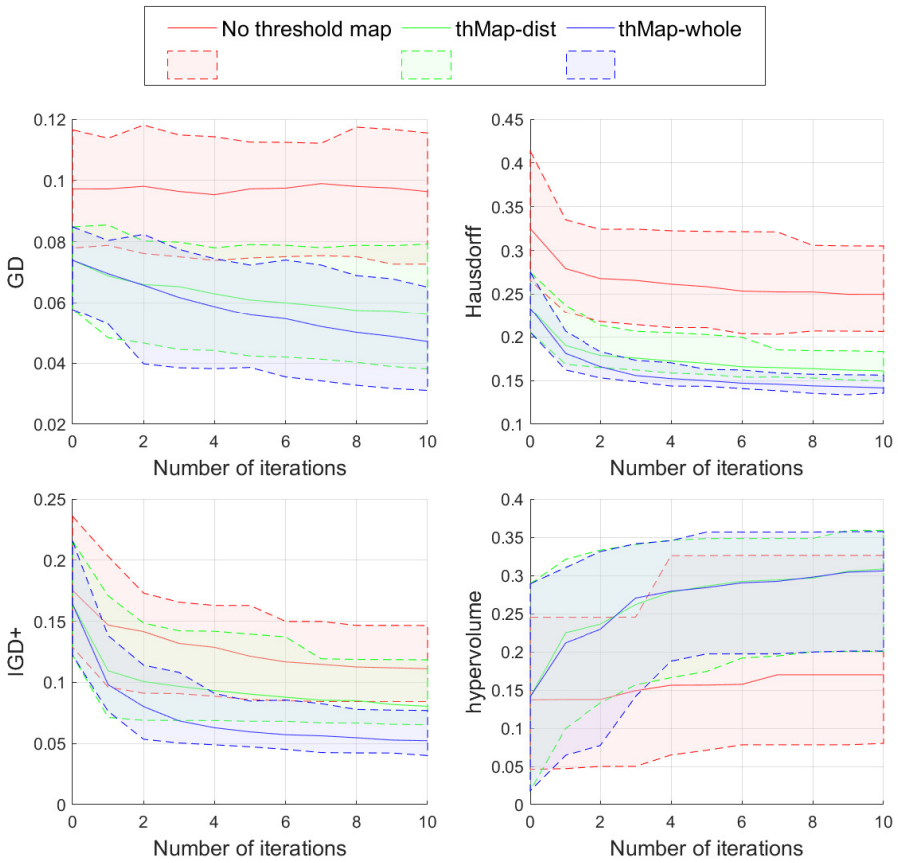


Figure 5.10: Many-objective metric plots, as in Fig. 5.8, for test case 1 with 5 uncertain variables.

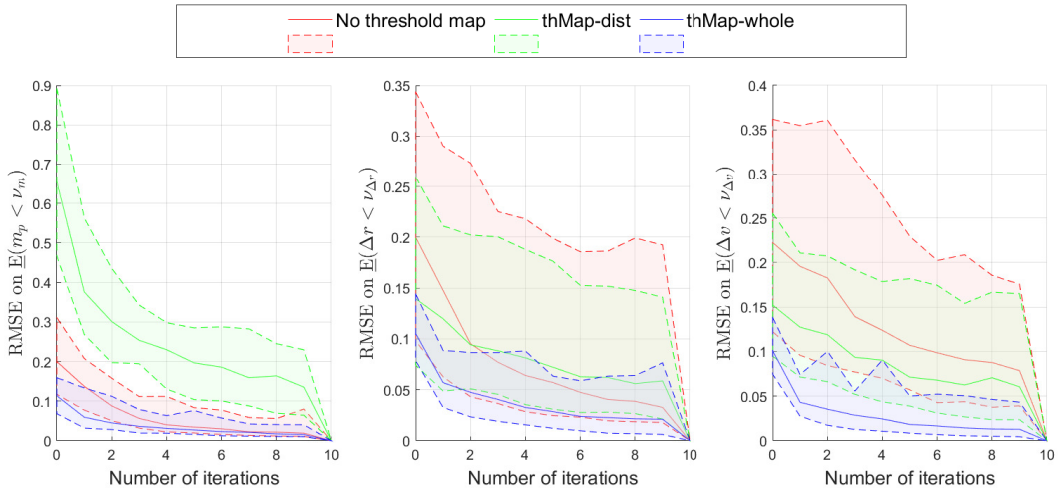


Figure 5.11: RMSE plots, as in Fig. 5.9, for test case 1 with 5 uncertain variables.

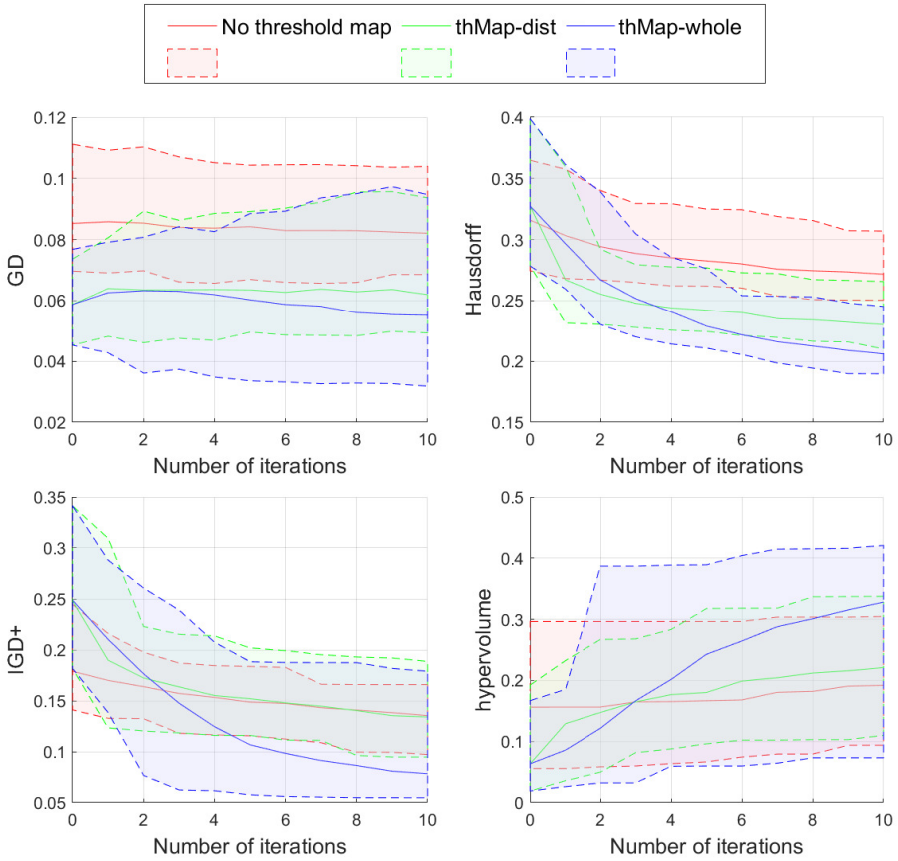


Figure 5.12: Many-objective metric plots, as in Fig. 5.8, for the outage test case.

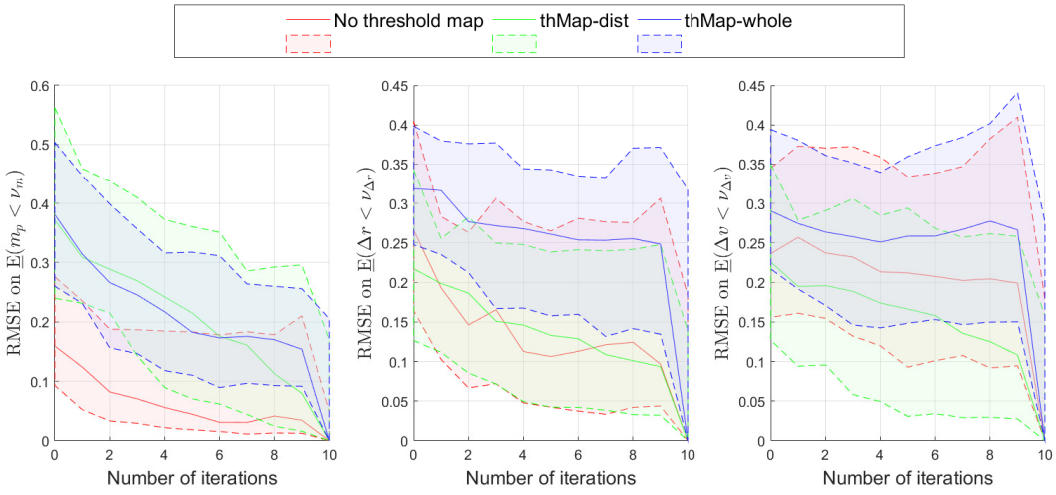


Figure 5.13: RMSE plots, as in Fig. 5.9, for the outage test case.

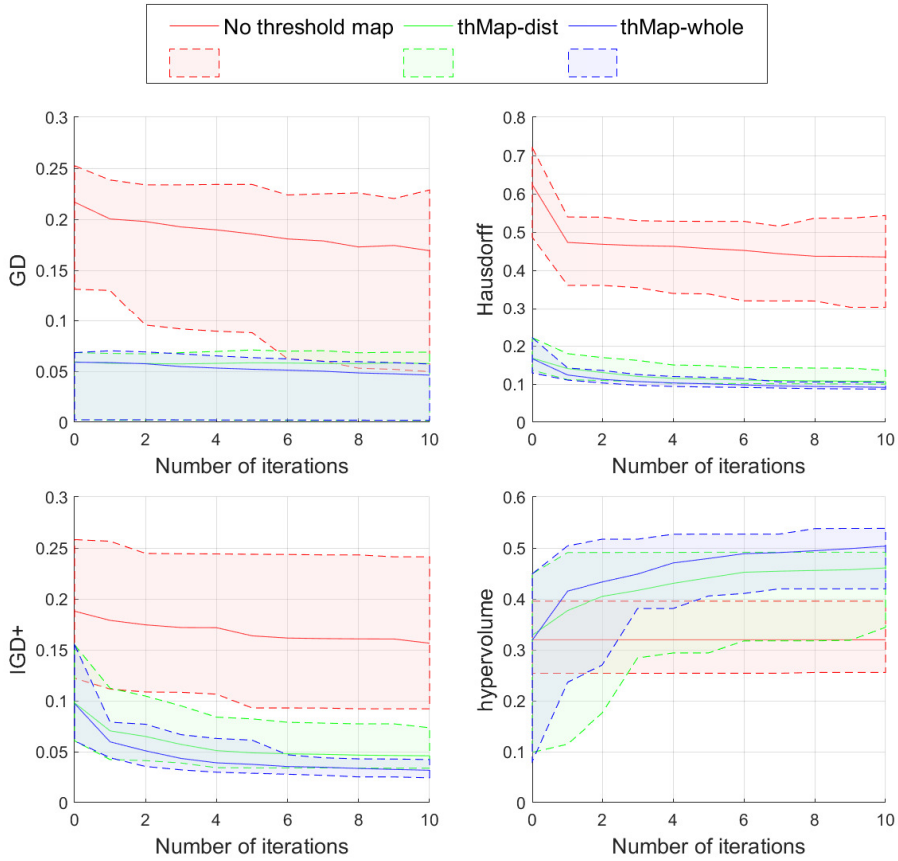


Figure 5.14: Many-objective metric plots, as in Fig. 5.8, for test case 3.

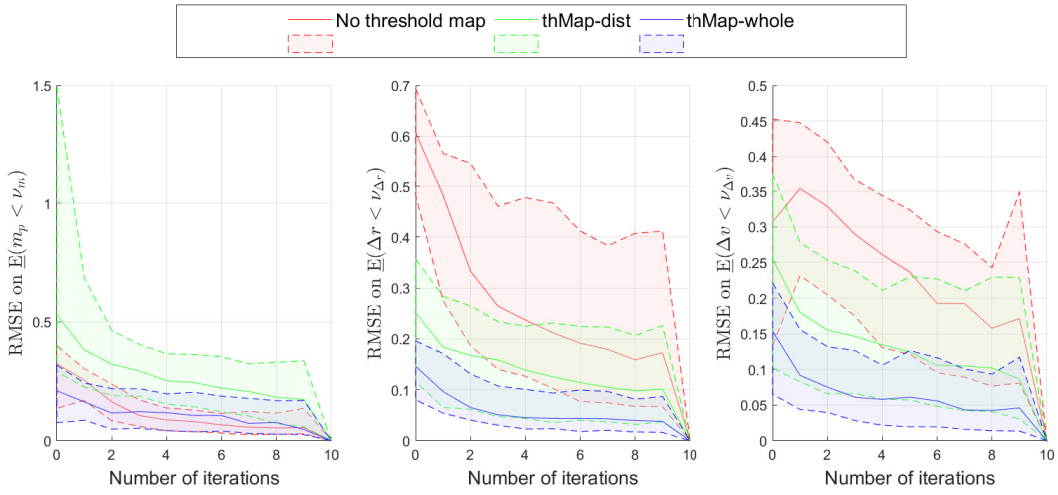


Figure 5.15: RMSE plots, as in Fig. 5.9, for test case 3.

5.5 Control Map Comparison

This Section is mainly concerned with comparing the different control mapping strategies proposed in Section 2.7, although these comparisons are repeated both without any threshold map, and using *thMap-whole*.

In Fig 5.16 some many-objective metrics are shown as a function of the iteration number, for test case B. Methods are compared with and without the threshold map in Section 3.5, and with the four different control maps explained in Section 2.7. Throughout these results, the control mapping strategies are referred to by the reduced dimensionality subset of the transcribed control variables Y that they result in, which are Y^*, Y^-, Y^B, and Y^M, and whose definition is in Section 2.7.

The metrics used are the generational distance (GD), the Hausdorff distance (Hausdorff), the modified inverse generational distance (IGD+) [8], and the number of points that are not dominated by the reference front (N_P). The reference front, used to calculate all of these metrics, is the combination of all samples obtained from all methods being compared, after removing all dominated values. These results suggest that Y^M is an improvement on the simpler control maps Y^* and Y^-. The deterministic control map Y^*, struggles to improve the metrics beyond their initial values, shown in Fig 5.16 as iteration 0, and which are determined only by the initial training points for the surrogate model. Furthermore, the threshold map consistently improves the results obtained using the Y^B and Y^M control maps. Furthermore, this improvement is not simply due to a reduction in the number of dimensions, since the dimensionality of Y^- ($N_\xi = 10$) is lower than Y^M (12), yet Y^M performs better.

Finally, the errors on the surrogate models are presented in Fig. 5.17.

The same tests were performed for the Apophis rendezvous test case A1. Figure 5.18 shows the MO metric comparisons, and Figure 5.19 plots the errors on the surrogate models. For this test case, the threshold map is still consistently advantageous, but the newly proposed control maps are not. Possibly the higher number of targets, or the fact that they are fly-by only, is what makes the asteroid tour test case benefit from the control map Y^M, which takes displacements on target positions as input.

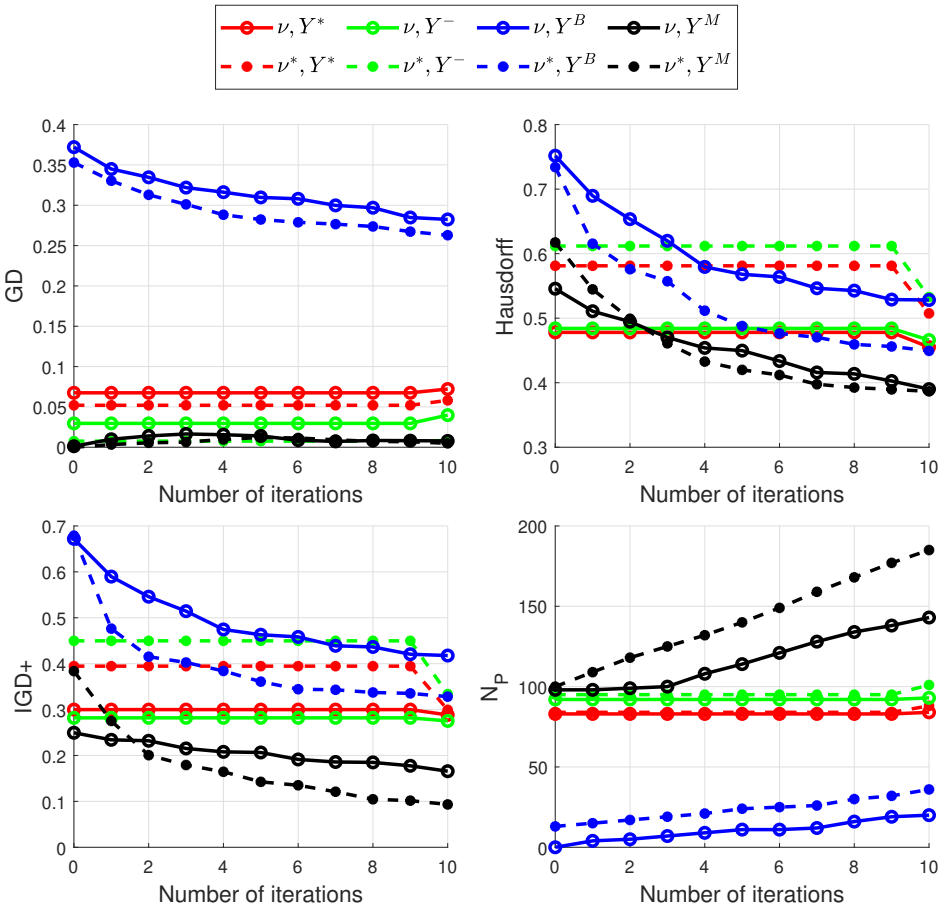


Figure 5.16: MO metrics for test case B. The colour of the lines indicates which control map was used, red for Y^*, green for Y^-, blue for Y^B and black for Y^M, which are described in Section 2.7. The line and marker styles indicate whether threshold mapping is used, solid lines with circles when threshold mapping is not used (ν), and dashed lines with asterisks when *thMap-whole* is used (ν^*).

5.6 Influence of MACS Settings

This section considers the effect of three parameters that define the settings of MACS: the number of agents N_p, the size N_a of the archive where Pareto solutions are stored, and the number of points N_s, in the archive, which are validated every time the surrogate is re-trained. By definition the number N_s has to be less or equal to N_a.

Note that no historical point is used, but only the most recent set of points returned by MACS. For all tests the number of function evaluations of the surrogate by MACS, n_f, was kept constant at 10000.

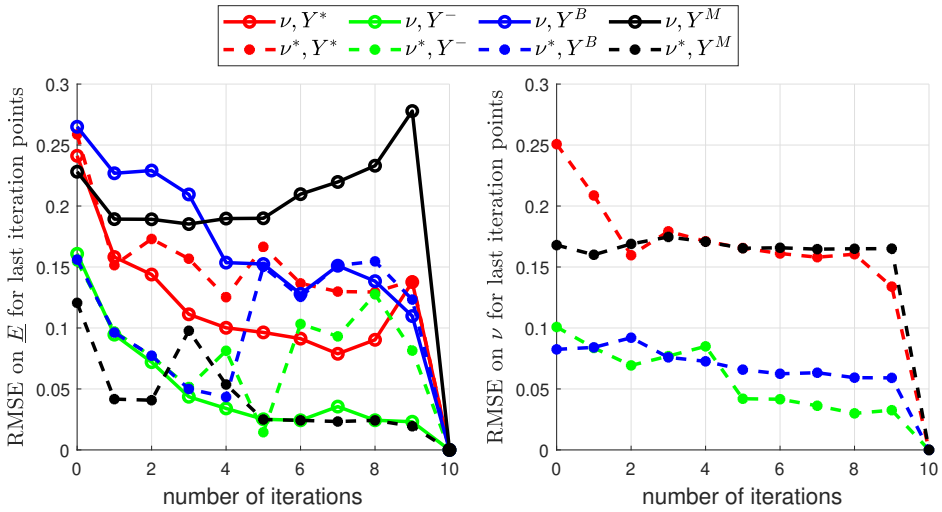


Figure 5.17: Root mean square errors (RMSE) for the surrogates of the lower expectation (left) and threshold map (right), for the asteroid tour test case. The RMSE is evaluated for the points that are obtained at the last iteration, and the mean is taken across all metrics. For the threshold map, the error is normalised by dividing by the difference between the maximum and minimum values. The line and marker styles follow the same pattern as in Fig. 5.16.

Since the validation of the N_s points requires the exact calculation of the lower expectation and the control mapping, it is reasonable to try to keep this number as low as possible. On the other hand, an increase in these three parameters is expected to lead to an improvement of the results, thus there is a trade-off between quality of the Pareto front and computational cost. In these tests, two different values for each parameter are used, 10 and 20 and we tested many combinations of these values. Figure 5.20 shows that the combination $N_p = N_a = N_s = 20$ is statistically better. However, it comes with the cost of validating twice as many points per re-training of the surrogate. It is also apparent that an increase in N_s is what leads to an improvement of all the metrics. Because these improvements come at the expense of a higher computational cost, tests in following sections will continue to use $N_p = N_a = N_s = 10$. Figure 5.21, on the other hand, does not show a significant difference in the RMSE for the different parameter choices, even though when $N_s = 20$, the surrogate model has more training points.

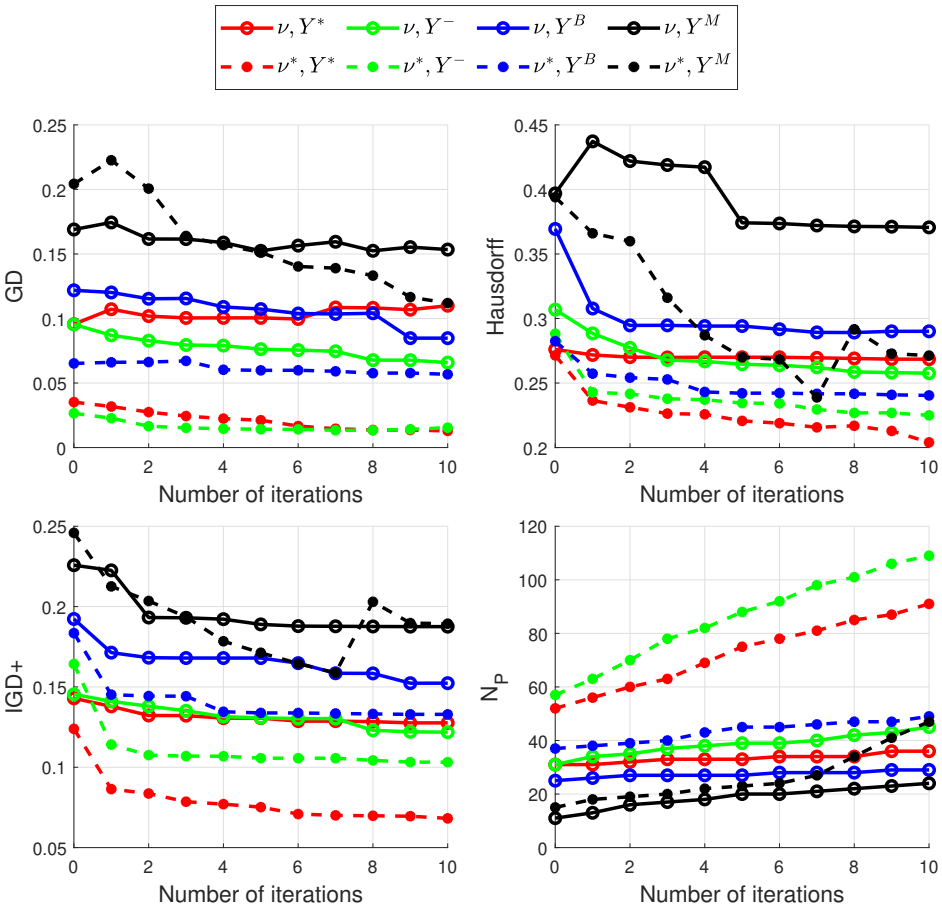


Figure 5.18: MO metrics for test case A1, as in Fig. 5.16.

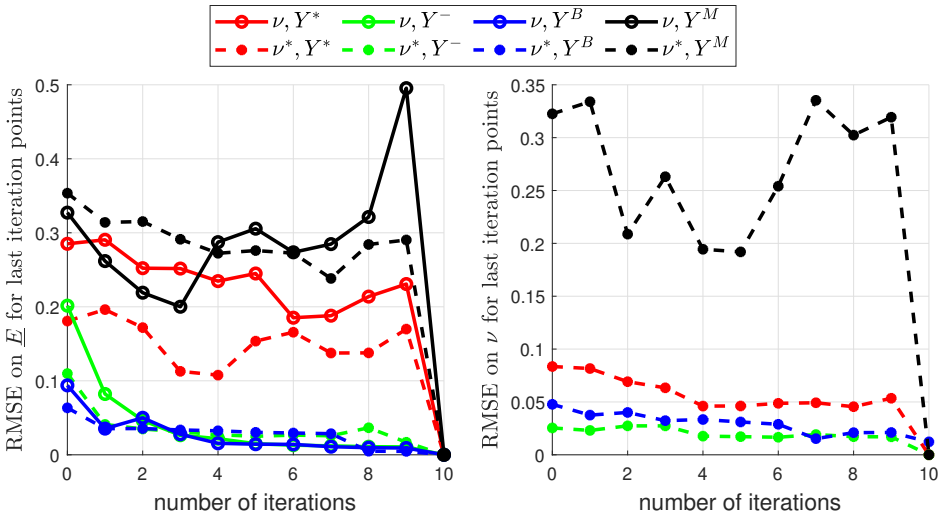


Figure 5.19: Root mean square errors (RMSE) for the surrogates of the lower expectation (left) and threshold map (right), for the Apophis Rendezvous test case. Same format as Figure 5.17.

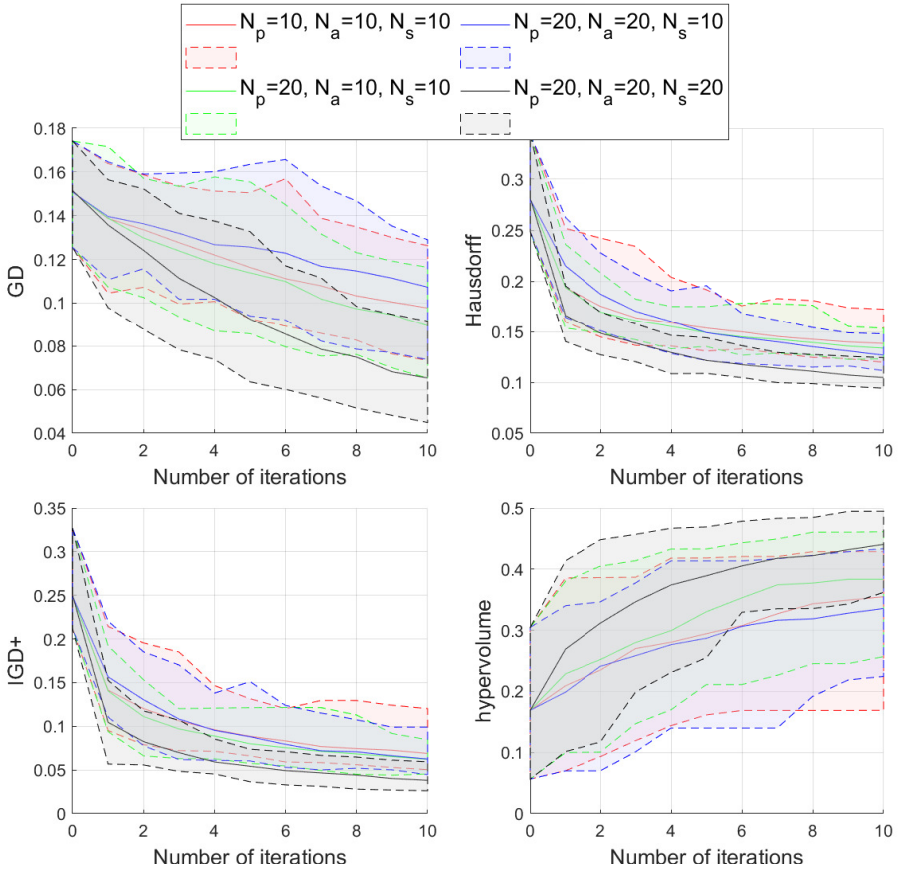


Figure 5.20: Many-objective metrics, as in Fig. 5.8, but varying the numbers N_p, N_a and N_s for test case 1 with $n_\xi = 10$.

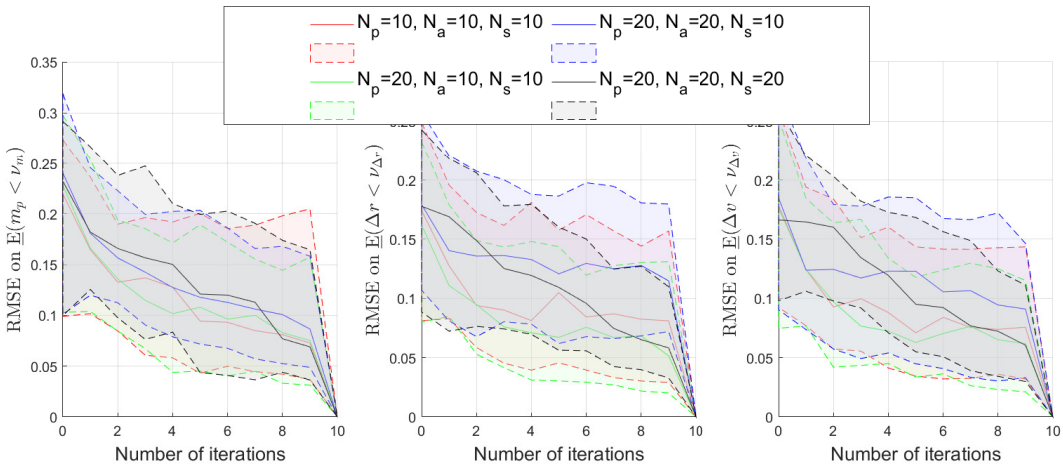


Figure 5.21: RMSE of the surrogate model, as in Fig. 5.9, varying the numbers N_p, N_a and N_s for test case 1 with $n_\xi = 10$.

5.7 Resulting Trajectories

In this Section, some solutions obtained with the methodology proposed in Chapter 3 are presented, with some values of the lower expectation of quantities of interest and associated thresholds. Chapter D then shows additional details such as selected plots of the trajectories and control laws for some of these solutions. This includes some discussion on the nature of these trajectories.

5.7.1 Case A1

Table 5.11 shows a selection of validated solutions found by MACS for the 6 objective problem in Eq. (5.1), using the threshold mapping technique *thMap-whole* and $N_p = N_a = N_s = 10$. The thresholds on the distance $\nu_{\Delta r}$ and relative speed $\nu_{\Delta v}$ to target that allow a non-negligible lower expectations ($\underline{\mathrm{E}} > 0.01$) are quite large for practical mission planning purposes. On the distance, $\nu_{\Delta r}$ ranges from $0.042AU$ to $0.094AU$ or, in kilometres, from $6.28 \times 10^6\ km$ to $1.41 \times 10^7\ km$. On the velocity, $\nu_{\Delta v}$ ranges from $0.856km/s$ to $2.711km/s$.

If the problem had been solved with fixed thresholds, all one would find is that the problem is unfeasible with those thresholds. With this approach, one has information on how large the thresholds would have to be to allow feasibility, which can be useful information for the mission designer to re-adjust the parameters of the mission accordingly. Besides the size of the bounds on the uncertain variables, which are reduced for Test Case A3, another reason for the high degree of uncertainty in these results is the inability of the control law to correct errors based on information on the spacecraft's orbital state. The dynamic programming approach introduced in Chapter 4, with results in Chapter 6, was motivated partly by this shortcoming.

Two reduced optimisation problems were also addressed within test case A1, to get results that are easier to visualise. One with only the three expectations as objectives, and fixed thresholds, and one with the three expectations plus the threshold on the terminal velocity as objectives and the rest of the thresholds kept fixed.

Figure 5.22 shows the Pareto front for the 3 objective case. Given the lower number

Table 5.11: A selection of solutions for the 6 objective problem in test case 1. After solutions with any $\underline{\mathrm{E}} \leq 0.01$ were removed, 16 solutions were left, of which 10 are presented which were selected using the archival strategy used by MACS [56]. $\underline{\mathrm{E}}(x)$ is used as short-hand for $\underline{\mathrm{E}}(x < \nu_x)$.

$\underline{\mathrm{E}}(m_p)$	$\underline{\mathrm{E}}(\Delta r)$	$\underline{\mathrm{E}}(\Delta v)$	ν_m [kg]	$\nu_{\Delta r}$ [AU]	$\nu_{\Delta v}$ [km/s]
0.989	0.024	0.277	47.922	0.042	2.348
0.442	0.416	0.123	53.881	0.061	1.692
1.000	1.000	1.000	33.643	0.089	2.711
0.188	0.061	0.925	40.955	0.045	2.645
0.013	0.010	0.013	50.275	0.023	0.856
0.342	0.999	0.278	50.330	0.097	2.516
0.522	0.975	0.032	42.922	0.079	0.965
0.969	0.519	0.029	46.256	0.094	1.774
0.035	0.133	0.618	51.188	0.049	2.622
0.996	0.756	0.230	51.204	0.083	2.303

of objectives we increased the values of N_p, N_a, and N_s to generate a clearer figure. Figure 5.22 was generated with $N_p = N_a = N_s = 10$, 100 initial training points, and thresholds ν_{m_p}, $\nu_{\Delta r}$, and $\nu_{\Delta v}$ fixed to 38 kg, 0.05 AU and 1.5 km/s respectively. Figure 5.23 corresponds to the same problem but was generated with $N_p = N_a = N_s = 105, 200$ initial training points. It is interesting to note that Fig. 5.22 is a good approximation of a portion of Fig. 5.23 which implies that Algorithm 1 can be effectively run with a reduced computational cost and still produce a good first approximation of the Pareto set.

Figure 5.24 shows an approximation of the Pareto front for the 4 objective problem with $N_p = N_a = N_s = 220$ and 200 initial training points. Here the thresholds ν_{m_p} and $\nu_{\Delta r}$ were fixed to 38 kg and 0.06 AU, respectively. In order to better visualise this result, Fig. 5.25 shows three slices of the 4-dimensional Pareto front obtained by only showing the points for which each of the lower expectations is higher than 95%. In this figure one can better appreciate the relation between the threshold $\nu_{\Delta v}$ and $\underline{\mathrm{E}}(\Delta v < \nu_{\Delta v})$.

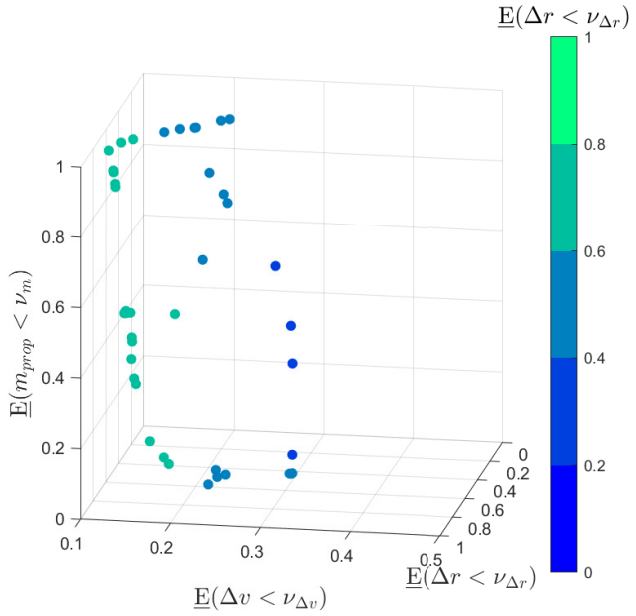


Figure 5.22: Scatter plot of the 41 non-dominated solutions obtained for the 3 objective problem, test case 1, of optimising all lower expectations, having the thresholds ν_{m_p}, $\nu_{\Delta r}$, and $\nu_{\Delta v}$, fixed to 38 kg, 0.05 AU and 1.5 km/s respectively. The lower expectations are represented on the figure axis. The value of $\underline{\mathrm{E}}\left(\Delta r < \nu_{\Delta r}\right)$ is also represented with the colour map. This graph was obtained with $N_p = N_a = N_s = 10$ and 100 initial training points.

5.7.2 Case A2

Again in this test case, the solution of the six objective problem is obtained using threshold mapping *thMap-whole*, with $N_{\text{init}} = 100$ and $N_p = N_a = N_s = 10$. Table 5.12 shows a selection of solutions obtained as for case A1. The thresholds also have large values, albeit slightly smaller, with $\nu_{\Delta r}$ still in the order of $0.01AU$ or millions of km, and delta-v in the order of $1km/s$. To help illustrate the outage scenario, Fig. 5.26 shows, for the same control law, the thrust fraction that results from three different values of the uncertain parameters.

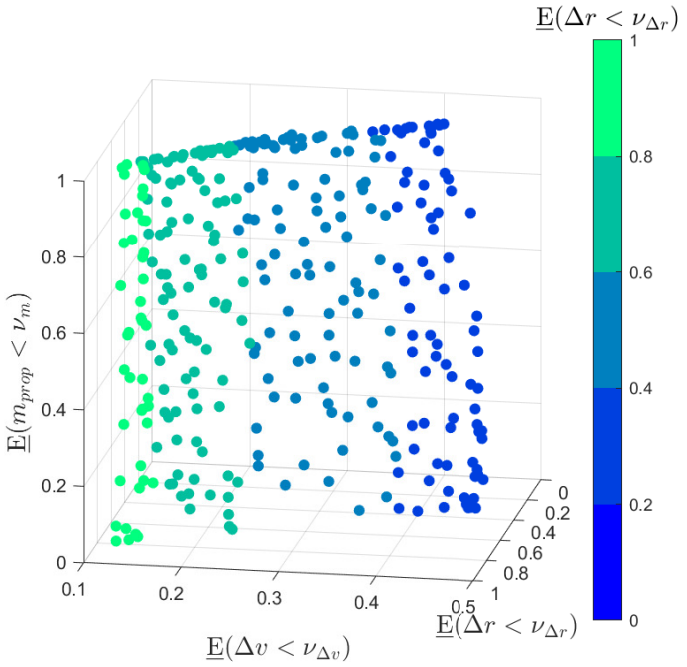


Figure 5.23: Scatter plot of the 324 non-dominated solutions obtained for same conditions as Fig. 5.22, but with $N_p = N_a = N_s = 105$ and 200 initial training points.

Table 5.12: A selection of solutions of the 6 objective problem in test case A2 returned by our method. These results were chosen in the same was as those in Table 5.11

$\underline{\mathrm{E}}(m_p)$	$\underline{\mathrm{E}}(\Delta r)$	$\underline{\mathrm{E}}(\Delta v)$	ν_m [kg]	$\nu_{\Delta r}$ [AU]	$\nu_{\Delta v}$ [km/s]
0.015	0.024	0.085	36.629	0.054	2.623
0.742	0.324	0.036	45.798	0.070	1.726
0.902	1.000	1.000	44.033	0.094	3.301
0.898	0.151	0.105	40.790	0.178	5.240
0.152	0.979	0.560	32.897	0.070	1.945
0.041	0.073	0.751	39.594	0.041	2.746
0.993	0.879	0.018	38.407	0.081	1.084
0.904	0.011	0.019	48.366	0.012	0.534
0.797	0.365	0.769	43.871	0.045	2.193
0.032	0.044	0.015	44.112	0.023	0.461

Figure 5.27 shows histograms for the values of the QoIs h. It is noticeable that

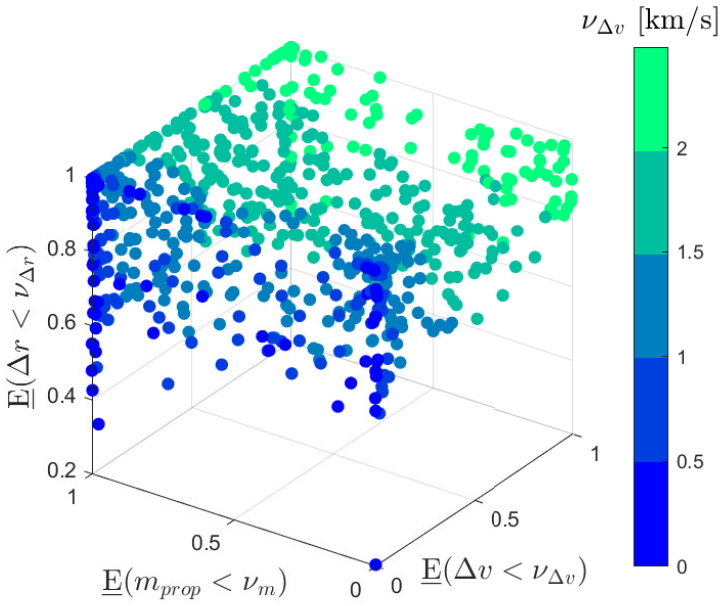


Figure 5.24: A plot of the 711 samples of the Pareto front obtained for the maximisation of the lower expectations and the minimisation of the threshold $\nu_{\Delta v}$ with the other thresholds fixed. Each axis represents one of the lower expectations, and the colour scale represents the threshold $\nu_{\Delta v}$. The thresholds ν_{m_p} and $\nu_{\Delta r}$ were fixed to 38 kg and 0.06 AU respectively. This data was obtained using $N_p = N_a = N_s = 220$ and 200 initial training points.

these histograms are not uni-modal, a characteristic of the outage test case that is possibly related to the higher difficulty experienced in optimising this problem, most clearly seen in the higher errors for the surrogate model (Fig. 5.13), and in the lower expectation estimation (Table 5.7), as well as the somewhat lower quality Pareto front plots in Fig. 5.29. Nonetheless, this method produced a variety of robust solutions for this problem.

After studying the 6 objective case, we fixed the thresholds on terminal position and velocity and solved a reduced 4 objective case with only the three lower expectations and the threshold on mass. The remaining thresholds $\nu_{\Delta r}$ and $\nu_{\Delta v}$ were fixed to 0.05 AU and 2 km/s. Figure 5.28 shows an approximation to the Pareto front with 415 solutions. Figure 5.29 shows three slices of the four dimensional Pareto front corresponding to expectations $\underline{\mathrm{E}} > 80\%$.

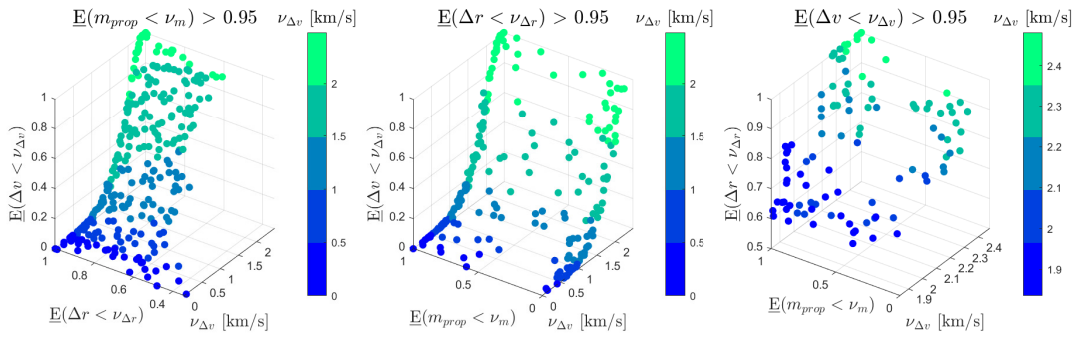


Figure 5.25: This figure contains approximate slices of the Pareto front in Fig. 5.24, obtained by only showing the points for which each of the lower expectations is higher than 95%. The other two lower expectations that are not used to filter the data are represented in the vertical and horizontal axis, as labelled. The threshold $\nu_{\Delta v}$ is represented in the depth axis, as well as the colour scale.

5.7.3 Case A3

The experiments carried out so far resulted in large values of the thresholds, due to the severity of the assumed epistemic uncertainty during operations. Therefore, test case A1 was run again but with a smaller uncertainty space Ξ_r for the calculation of the lower expectation, while the uncertainty space used in the control map Ξ remained the same.

A sample of the solutions produced by the many-objective optimisation is presented in Table 5.13. Compare with Table 5.11, while the values of $\nu_{\Delta r}$ and $\nu_{\Delta v}$ are still large, they are now around one order of magnitude smaller, with $\nu_{\Delta r}$ in particular now in the order of the hundreds of thousands of km.

Similarly to Fig. 5.25, Fig. 5.30 shows slices resulting from optimising the 4 objective problem, where the objectives are all the lower expectations, and the threshold on the propellant mass ν_m. The remaining thresholds $\nu_{\Delta r}$ and $\nu_{\Delta v}$ are fixed to 0.007 AU and 250 m/s, respectively. A total of 778 non-dominated points were found. It is clear from both Table 5.13 and Fig. 5.30 that this uncertainty formulation produced solutions with much better thresholds $\nu_{\Delta r}$ and $\nu_{\Delta v}$. Figure 5.30 illustrates the dependency of the lower expectations on the thresholds. Note in particular how the expectation on the terminal velocity decreases as the available propellant decreases.

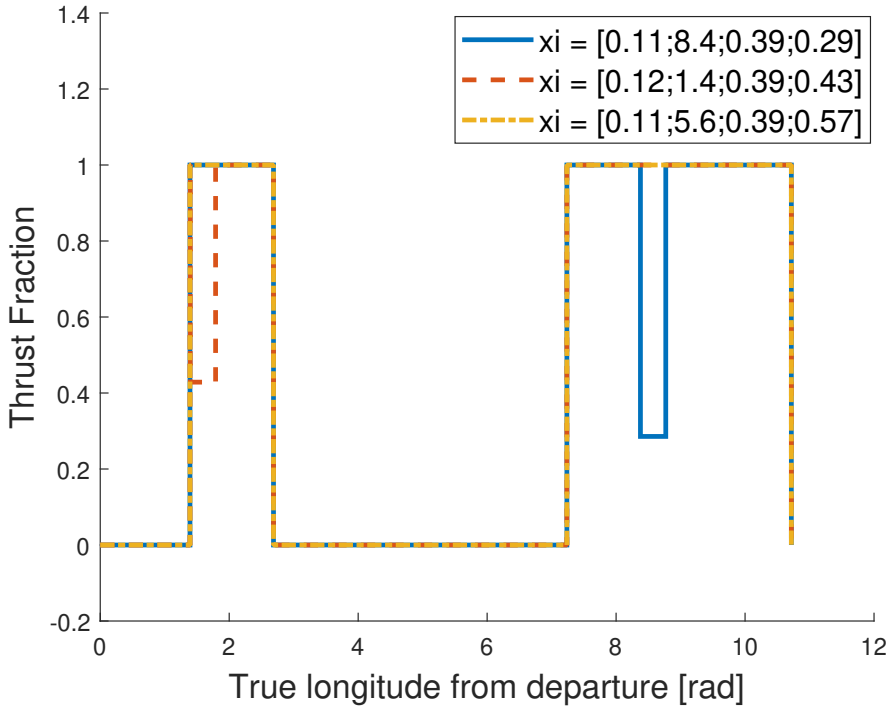


Figure 5.26: Plot of the thrust modulus for three different values of the uncertain parameters $\boldsymbol{\xi}$. The legend shows the numerical values of $\boldsymbol{\xi}$, defined as in Eq. (5.4). For the yellow line, the outage happened during coasting arcs, producing no effect.

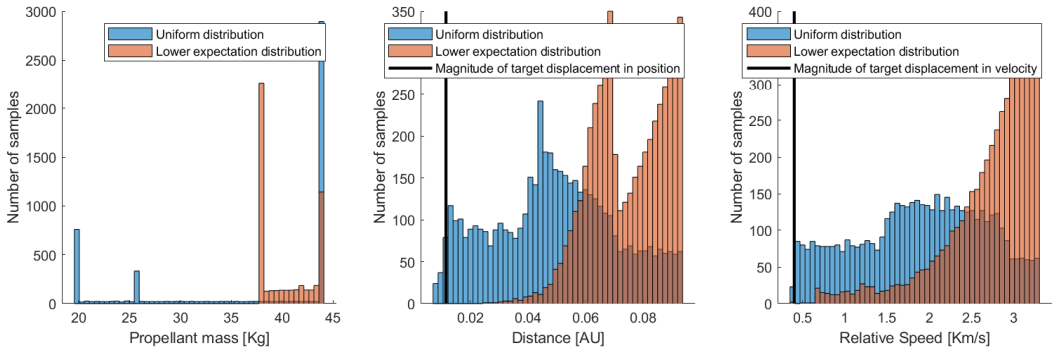


Figure 5.27: Histograms showing the samples of the QoIs that are obtained for the third solution in Table 5.12 for test case 2. The blue histogram is the result of assuming a uniform distribution in Ξ, while in red is the distribution that corresponds to the lower expectation for each QoI.

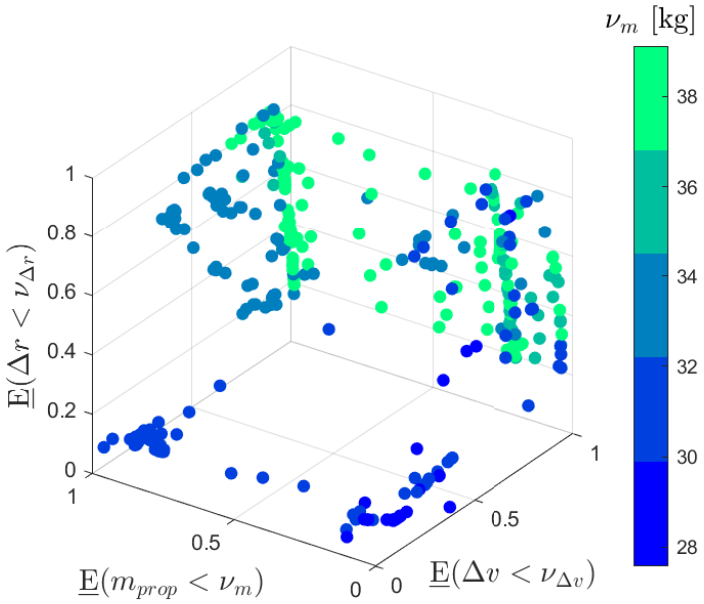


Figure 5.28: Scatter plot representing the 415 non-dominated solutions obtained for the maximisation of $\underline{\mathrm{E}}\left(m_p < \nu_{m_p}\right)$, $\underline{\mathrm{E}}\left(\Delta r < \nu_{\Delta r}\right)$, $\underline{\mathrm{E}}\left(\Delta v < \nu_{\Delta v}\right)$ and the minimisation of the threshold on the propellant mass ν_{m_p}, for test case 2. The lower expectations are represented on the figure axis, while the threshold on propellant mass is in the colour scale. The thresholds $\nu_{\Delta r}$ and $\nu_{\Delta v}$ were fixed to 0.05 AU and 2 km/s respectively. Obtained using $N_p = N_a = N_s = 220$, and 200 initial training points.

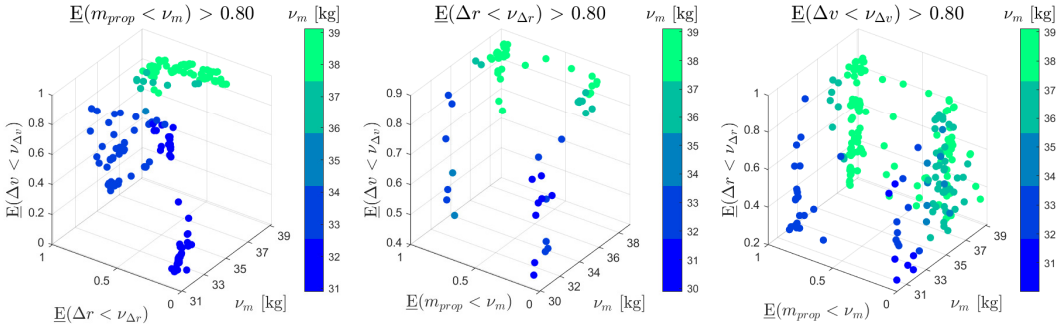


Figure 5.29: This figure contains approximate slices of the Pareto front in Fig. 5.28, obtained by only showing the points for which each of the lower expectations is higher than 80%.

Table 5.13: A selection of solutions of the 6 objective problem in test case A3 returned by our method. These results were chosen in the same was as those in Table 5.11

$\underline{\mathrm{E}}(m_p)$	$\underline{\mathrm{E}}(\Delta r)$	$\underline{\mathrm{E}}(\Delta v)$	ν_m [kg]	$\nu_{\Delta r}$ [AU]	$\nu_{\Delta v}$ [km/s]
0.875	0.036	0.557	41.575	0.006	0.361
0.987	0.010	0.201	37.724	0.009	0.245
1.000	1.000	1.000	57.415	0.019	0.640
0.025	0.995	0.018	50.855	0.010	0.299
0.014	0.105	1.000	47.556	0.009	0.351
0.990	1.000	0.081	31.558	0.016	0.477
0.061	0.073	0.013	45.436	0.008	0.278
0.999	0.970	0.740	49.545	0.011	0.402
0.704	0.999	0.066	45.769	0.011	0.264
0.991	1.000	1.000	34.165	0.020	0.681

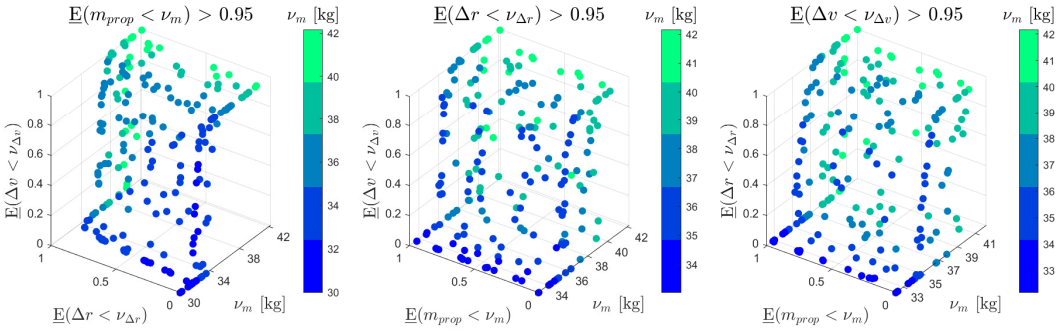


Figure 5.30: This figure contains approximate slices for the maximisation of the lower expectations, and the minimisation of the threshold on the propellant mass ν_m. The remaining thresholds $\nu_{\Delta r}$ and $\nu_{\Delta v}$ are fixed to 0.007 AU and 250 m/s, respectively. These figures are obtained by only showing the points for which each of the lower expectations is higher than 95%, as in Fig. 5.25. Obtained using $N_p = N_a = N_s = 220$, and 200 initial training points.

5.7.4 Case B

In this section some of the solutions to the asteroid tour test case, obtained using the Min-Max Y^M control map, are presented. Table 5.14 shows ten solutions, selected form the 188 non-dominated solutions found with this method, by using the same archival strategy employed by MACS [56]. The values of the thresholds here are extremely large, representing distances on the order of tens of millions of kilometres. The dynamic programming approach, with results in Chapter 6, will be applied to this same test case, achieving much better results.

Table 5.14: A selection of solutions returned by our method. Solutions were selected using the same archival algorithm employed by MACS [56]. If x is one of our targets, $\underline{\mathrm{E}}^{(x)}$ is used as short-hand for $\underline{\mathrm{E}}(\Delta_{r_x} < \nu_{\Delta_{r_x}})$ and ν_x is short-hand for $\nu_{\Delta r_x}$ and is in AU. For the propellant mass m_p, $\underline{\mathrm{E}}^{(m_p)} = \underline{\mathrm{E}}(m_p < \nu_m)$, and its threshold ν_m is in Kg.

$\underline{\mathrm{E}}^{(m_p)}$	$\underline{\mathrm{E}}^{\mathrm{UJ47}}$	$\underline{\mathrm{E}}^{\mathrm{UV}}$	$\underline{\mathrm{E}}^{\mathrm{YN176}}$	$\underline{\mathrm{E}}^{\mathrm{Ock}}$	ν_m [Kg]	ν_{UJ47}	ν_{UV}	ν_{YN176}	ν_{Ock}
0.044	0.239	0.232	0.548	0.032	129.742	0.212	0.306	0.327	0.104
0.000	0.333	0.727	0.162	0.657	74.023	0.265	0.396	0.255	0.522
1.000	1.000	0.000	0.000	1.000	89.345	0.463	0.012	0.040	0.700
1.000	0.000	1.000	1.000	1.000	190.913	0.016	0.519	0.491	0.810
1.000	1.000	1.000	1.000	0.000	79.550	0.484	0.539	0.536	0.024
0.000	1.000	0.000	1.000	0.993	259.146	0.488	0.004	0.534	0.648
1.000	0.000	0.000	0.000	0.000	87.848	0.005	0.041	0.062	0.013
0.000	1.000	1.000	0.000	0.000	189.049	0.478	0.507	0.009	0.063
1.000	0.000	0.000	1.000	1.000	92.752	0.039	0.018	0.555	0.694
1.000	1.000	1.000	0.000	1.000	62.334	0.437	0.577	0.052	0.746

5.8 Execution Times

The execution time was split into four parts: the computation of $\underline{\mathrm{E}}$, the control map, the threshold map, and running MACS on the surrogate model. The following times were obtained in Matlab® 2018b on a Windows 10 computer with Intel(R) Core(TM) i7-8700 @3.2GHz and 8GB of RAM. Matlab is used, a language that is not amongst the most computationally efficient, but since it, and the other software and hardware specifications, remain a constant in all experiments, any comparisons that are made between different approaches remain valid.

- Running MACS on the surrogate model took 27 seconds per iteration with 10000 function evaluations for the test case A1. For the test case A2, it took 15 seconds per iteration with the same settings, due to the lower dimensionality of the search space.
- The control map Y^* took on average 4.4 seconds, to calculate $\mathbf{y}$ from $\mathbf{z}$. For the computational times with other control maps, see text later in this section.
- Solving Problem (3.9) to obtain $\underline{\mathrm{E}}$ from $\mathbf{v}$ took on average 390.3 estimates of E and 11.7 seconds for the test case A1 (variable thrust and Isp), for an average of 0.030 seconds per estimate of E. For the engine outage test case, A2, there was an average of 139.5 evaluations of E (lower due to the smaller number of uncertain variables) and 15.1 seconds, for an average of 0.108 seconds per estimate of E (higher because the calculation of the QoIs for the quasi-Monte Carlo estimation of E could not be fully vectorised).
- The computation of the threshold map, requiring the estimate of $\underline{h}$ and $\overline{h}$, took on average 0.45 seconds for test case A1 and 0.50 seconds for test case A2.

The total time when using the Y^- on test cases A1, A2, and A3, evaluating 100 initial points plus 10 per each of 10 iterations, is approximately 1 hour. Despite the outage scenario (A2) having fewer dimensions in the uncertainty space, this is counter-acted by the difficulty in vectorising the code that applies the outage conditions, which in Matlab can make a significant difference in the execution time.

The control maps other than Y^*, defined in Section 2.7, were only used to obtain the results in Section 5.5, which were also the only results in this chapter using test case B, the asteroid tour. Because, for those results only, the different test cases were run in parallel and on different machines, the execution times obtained for that case were different from the ones shown above, and are merely indicative of the relative amount of time it took to compute each control map.

- Y^* and Y^- took approximately 35 seconds for test case B and 8 seconds for case A1,
- Y^B took approximately 205 seconds for test case B and 55 seconds for case A1,
- Y^M took approximately 128 seconds for test case B and 75 seconds for case A1.

Test case B with control map Y^M took approximately 8 hours.

The newly proposed control maps have considerably higher computational times, however, they use algorithms with many hyper-parameters, that could be further fine-tuned. Furthermore, in the asteroid tour test case, control maps Y^* and Y^- were not allowing any improvement in the many-objective metrics at all, in light of which the newly proposed control maps are certainly advantageous.

The threshold map, with its relatively low computational cost and its significant advantage it brings to the MO metrics, is a promising approach.

5.9 Remarks on Accuracy of the Results

Although a formal convergence theory was not derived, it is important to remark some features of our results that come, by construction, from the way the method was devised.

- Since each uncertain variable is modelled with a p-box and the upper and lower probability of each p-box are modelled with Bernstein polynomials, the range of probability distributions within each p-box depends on the degree of the Bernstein polynomials.
- The lower expectation is minimal with respect to the set of distributions that can be represented with Bernstein polynomials of a given degree defined on a bounded

support. Thus our results are valid for every distribution in that set. This implies that a Monte Carlo simulation drawing samples from any distribution, within that set, that is not minimal would return a higher expectation.

- From the error analysis on the minimisation of the expectation and on the convergence of the surrogate one can expect an average error of $< 10\%$ on the value of the lower expectation. This error does not necessarily correspond to an outer approximation of the lower expectation but is deemed to be sufficient to compute an optimal solution. Once a solution is available one can invest more computational resources to further refine the computation of the lower expectation.

- The control law is optimal within the reduced control space defined by the dimensionality reduction technique. Thus no consideration can be made on control laws that are outside this reduced space.

5.10 Conclusion

This chapter applies the methods of Chapters 2 and 3 to two test cases, both based on low thrust trajectory optimisation under epistemic uncertainty. The main purpose was to exemplify the application of the algorithms developed in both of those chapters, and also to test different variations for subcomponents of the overall method, such as the threshold and control maps, and the settings for MACS.

The results conclusively showed that applying either of the two threshold maps introduced in this work improves the quality of the results, with *thMap-whole* appearing to be the more efficient version.

The comparison on the control maps was not as conclusive. On test case B, the control maps Y^M appears to significantly improve the quality of the results relative to the simpler control maps which were developed earliest, Y^* and Y^-, while Y^B appears to be worse. However, on test case A1, the exact opposite conclusion is reached, here Y^M performs the worst, while Y^* and Y^- perform the best. This test was carried out for all control maps both with the threshold map *thMap-whole*, and without any threshold map, to test whether its usage had an effect on which control map is most

advantageous. While no such effect was found, this had the incidental outcome of confirming the advantage of using this threshold map since the tests with *thMap-whole* always led to better results, for both test cases, and for all control maps.

Finally, there was a test looking at some parameters used by the numerical many-objective solver. The only parameter that was found to have a significant influence on the quality of the results was the size of the archive, N_a, which unfortunately is also the one parameter that has by far the most effect on the computational cost of the algorithm. There is therefore a trade-off between quality and computational cost, that is mainly controlled by N_a.

In terms of the actual resulting trajectories, and the values of the objective functions, which were of the form $-\underline{\mathrm{E}}(h < \nu_h)$ and ν_h for each quantity of interest h, the results shows that $\underline{\mathrm{E}}$ only has a high value, indicating a high degree of belief in the condition $h < \nu_h$ being met, when ν_h has an impractically large value. This is the case for the quantities of interest related to the distance and relative speed to each target. One factor contributing to this is the unrealistic implicit assumption that the trajectory, once started, cannot be corrected based on observed deviations from the nominal trajectory. One of the aims of Chapter 4 was the development of a method that can handle this, and thus produce a more realistic scenario. The next chapter applies this method to test case B.

Chapter 6

Numeric Results with Dynamic Programming Approach

As in the work in [16], the proposed approach is applied to a multi fly-by mission to a sequence of four near earth asteroids (NEA), based on the asteroid tour designed in [21], visiting 2006 UJ47, 2007 UV, 2005 YN176, and Ockeghem (Ock). The optimisation problem for this test case is written as

$$\min_{\mathbf{u}} \left[\overline{\nu}_{\hat{\mathrm{E}}}(m_p), \overline{\nu}_{\hat{\mathrm{E}}}(\Delta r_1), \overline{\nu}_{\hat{\mathrm{E}}}(\Delta r_2), \overline{\nu}_{\hat{\mathrm{E}}}(\Delta r_3), \overline{\nu}_{\hat{\mathrm{E}}}(\Delta r_4)\right]^T \tag{6.1}$$

where m_p is the propellant mass, and Δr_i is the distance to the i-th target. The control laws $\mathbf{u}$ are functions of time t, and of a single observation of the state $\mathbf{x}_2$, along with the proxy control variable $\mathbf{z}$ through the control map Y^-, described in Section 2.7.

A total of 20 thrust arcs are considered, each with the same length. The trajectory is divided into two transfer legs, the first comprising the first 5 arcs and the second the remaining 15 arcs. In this test case, the trajectory is split into two segments, i.e., $M = 2$. At the instant when going from the first leg to the second leg, t_2, the state $\mathbf{x}_2$ is observed with exactitude and used to obtain a new control law for the second transfer leg. MACS is ran with $N_a = 11$ agents, which is also the size of the archive. For the upper quantile, we use $\hat{\mathrm{E}} = 0.9$ as the fixed value of the lower expectation $\underline{\mathrm{E}}$. In the results, the upper quantile with this value of $\hat{\mathrm{E}}$ is written as $\overline{\nu}_{90}$.

The uncertain variables are the thrust T, specific impulse I_{sp}, and the magnitude of the hyperbolic excess velocity at departure v_∞. The thrust and specific impulse are considered to vary randomly throughout the trajectory. In this test-case this is represented by having variables T_i and $I_{sp,i}$ which describe the values of these quantities in different thrust arcs of the trajectory. Table 6.1 shows the nominal values of these quantities. The space Ξ is defined as the hyper-rectangle $[0.99\boldsymbol{\xi}_0\ ,\ 1.01\boldsymbol{\xi}_0]$. The epistemic uncertainty in these variables is defined following the framework in Section 3.4.1, by specifying the means and variances of the distribution to lie in intervals. The means are taken within the interval $\boldsymbol{\mu} \in [(1-10^{-4}))\boldsymbol{\xi}_0\ ,\ (1+10^{-4})\boldsymbol{\xi}_0]$, and the variances are in $\boldsymbol{\sigma}^2 \in [0.025\ ,\ 0.045] \times 10^{-4}\boldsymbol{\xi}_0^2$.

Table 6.1: Nominal values for the uncertain variables.

	v_∞ [Km/s]	$I_{sp,i}$ [s]	T_i [N]
$\boldsymbol{\xi}_0$	5.846	3000	0.15

The observation which is used to compute the feedback law is obtained after 5 thrust arcs. Thus the trajectory is split into two segments, the first comprised of the first 5 thrust arcs and the second of the remaining 15. Variables T_1 and $I_{sp,1}$ describe the first segment, while T_j and $I_{sp,j}$ for $j \in \{2,3,4\}$ describe the following segments of 5 arcs. Therefore, $\boldsymbol{\xi}_1 = [v_\infty\ T_1\ I_{sp,1}]^T$ and $\boldsymbol{\xi}_2 = [T_2\ T_3\ T_4\ I_{sp,2}\ I_{sp,3}\ I_{sp,4}]^T$.

6.1 Estimation of the Lower Expectation

For the estimation of the lower expectation, the user chosen parameter that controls the trade-off between accuracy and computational time is the number of kernels used n. To determine the influence of the number of kernels and variables on the mean absolute error ϵ and execution time t to calculate the lower expectation, the following experiment was carried out. For various numbers of kernels n and numbers of uncertain variables n_ξ, spherical indicator functions were used, defined as $h(\boldsymbol{\xi}) < \nu$, with $h(\boldsymbol{\xi}) = \|\boldsymbol{\xi}\|$, and ν chosen such that 80% of the volume was inside the sphere. The accuracy in Fig. 6.1 is the mean absolute accuracy for 100 different random samples for each pair of number

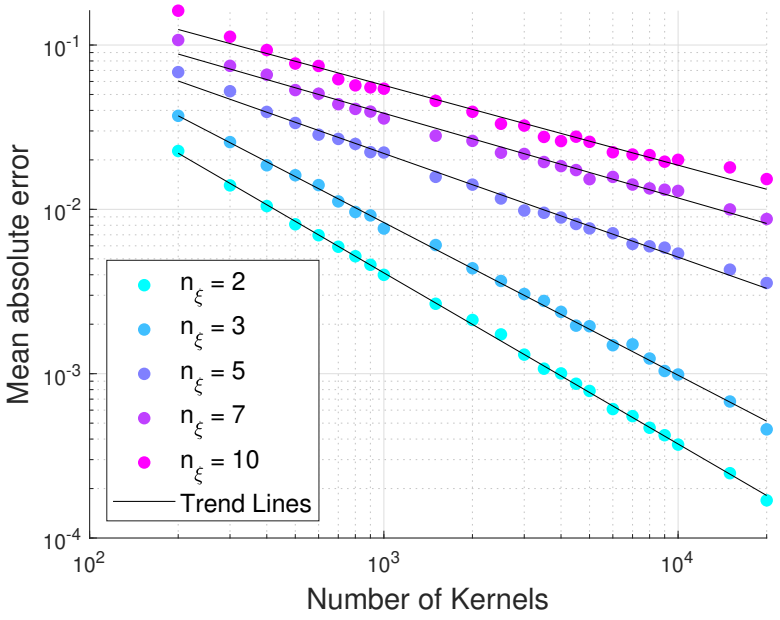


Figure 6.1: Log-log plots of mean absolute errors of lower expectation estimations for spherical indicator functions. Least squares trend lines are also shown.

of kernels and number of variables. The real value of the lower expectation used to calculate the accuracy is estimated as the smallest value obtained in all experiments for a particular number of uncertain variables. The results in Fig. 6.1 show that the mean absolute error ϵ follows a power law, with the respective trend line shown on the log-log plot. These trend lines are obtained using the method of least squares, and can be written as

$$\epsilon = \epsilon_0 n^{\theta_\epsilon} \tag{6.2}$$

In Fig. 6.2, the accuracy estimations for the asteroid tour test case previously presented are shown overlaid on the data in Fig. 6.1. The data is separated for each QoI, showing some dependency of the accuracy on the QoI. For both Figs 6.1 and 6.2, the indicator functions were obtained by choosing a threshold such that for 80% of the measure of Ξ, the value of the QoI is below that threshold, to make the results more comparable. Even though the asteroid tour data is obtained with $n_\xi = 9$, the mean absolute errors obtained with this data are higher than those obtained with the ball shaped indicators for $n_\xi = 10$ in some cases, and lower than those with $n_\xi = 7$ in other

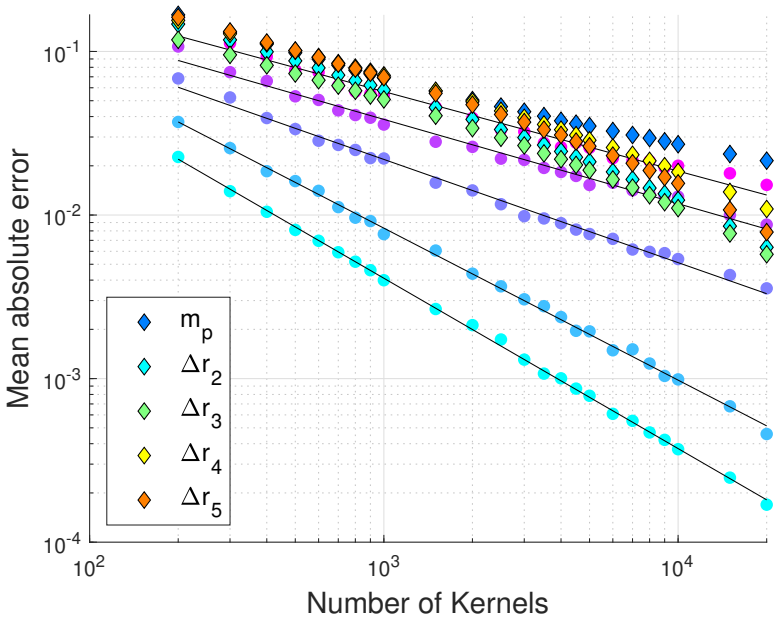


Figure 6.2: Log-log plots of mean absolute errors of lower expectation for the QoIs obtained for the asteroid tour test case overlaid on Fig. 6.1.

cases, due to the influence of the shape of the indicator function on the accuracy of the estimate of $\underline{\mathrm{E}}$.

The execution times also follow a power law relationship, as shown Fig. 6.3, as

$$t = t_0 n^{\theta_t} \tag{6.3}$$

The evolution of the exponents θ_ϵ and θ_t in Eqs. (6.2) and (6.3) with n_ξ is shown in Fig. 6.4. Since there are mean and variance constraints being applied on each variable, n_ξ affects the number of constraints, which are $n_Q = 2n_\xi$. The execution time, regardless of n_ξ, is approximately proportional to n^2, while the error ϵ is inversely proportional to n for low n_ξ, but for higher n_ξ becomes inversely proportional to $\sqrt{n}$.

6.2 Accuracy of Surrogate Model $\tilde{V}_2$

For the asteroid tour test case, the surrogate model $\tilde{V}_2$ was trained with different numbers of uncertain variables and control variables in the second segment, to study

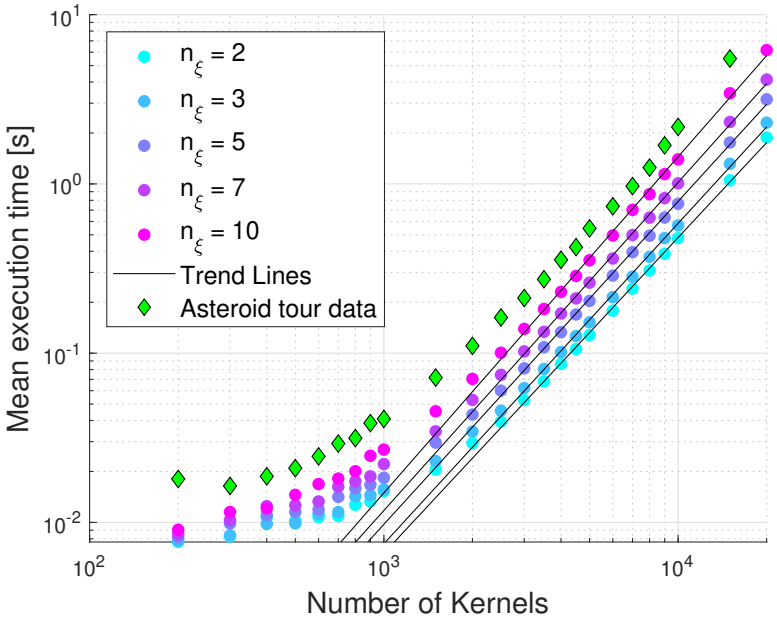


Figure 6.3: Log-log plots of mean execution time of lower expectation estimations for spherical indicator functions. Least squares trend lines are also shown. Only points with $n \geq 2500$ were included in the least squares, as only then does the power law appear to hold.

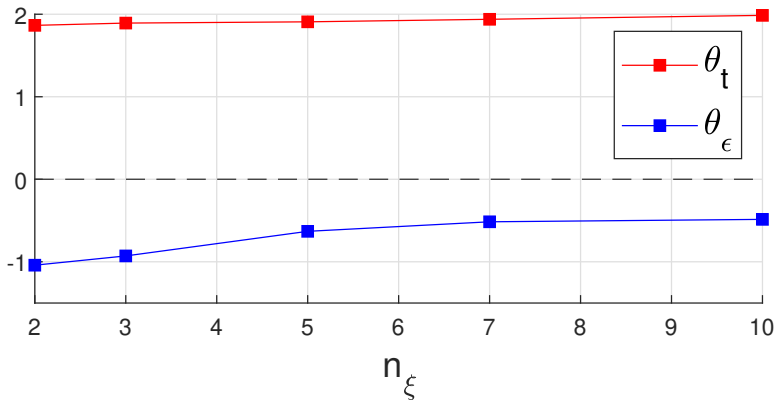


Figure 6.4: Plot of exponents θ_ϵ and θ_t in Eqs. (6.2) and (6.3) versus the number of uncertain variables n_ξ.

Table 6.2: Mean absolute errors for the surrogate models of $\tilde{V}_2$, for various values of $n_{\xi,2}$, the dimensionality of $\boldsymbol{\xi}_2$.

$n_{\xi,2}$	m_p	Δr_1	Δr_2	Δr_3	Δr_4
2	0.016	0.014	0.014	0.014	0.014
4	0.017	0.016	0.016	0.016	0.016
6	0.021	0.017	0.018	0.023	0.019

the resulting model accuracy. The results are in Table 6.2. The values reported in Table 6.2 are obtained by training the model with 800 training points and evaluating it on 200 testing points. These points are randomly selected from the set of 1000 points that is obtained using MACS's archival technique [56], described in Section 2.6.

These errors grow with $n_{\xi,2}$, the dimensionality of $\boldsymbol{\xi}_2$, and the error for m_p appears higher than the other cases, being between 1.4% and 2.3%. In addition, these errors are of similar magnitude to the results shown in Fig. 6.2.

6.3 Accuracy of Surrogate Model of Upper Quantiles $\tilde{\tilde{\nu}}$

The accuracy of the outer surrogate $\tilde{\tilde{\nu}}$ is less important than that of $\tilde{V}_2$ since the solution points returned by MACS, which is run with evaluations of $\tilde{\tilde{\nu}}$, are then evaluated as in Algorithm 4, obtaining more accurate values, whose accuracy depends nonetheless on the accuracy of $\tilde{V}_2$. The accuracy of $\tilde{\tilde{\nu}}$ affects, however, the efficacy of the optimiser.

The accuracy of the $\tilde{\tilde{\nu}}$ models, for each quantity of interest, is estimated by comparing the values obtained from $\tilde{\tilde{\nu}}$ with those obtained using Algorithm 4, for the values obtained in the last iteration. As more points are added on each iteration, the model accuracy should converge, at least in the vicinity of the Pareto front, since the points that are being obtained by MACS should be in that region. This is why the model is tested with the points obtained in the last iteration. Figure 6.5 shows the evolution of the root mean square of this error, normalised by dividing by the standard deviation of the data, which is equivalent to $\sqrt{1-R^2}$, where R is the coefficient of determination. For each iteration, the model is trained with the initial training points plus all the points returned by MACS up to and including that iteration, and is then tested. At

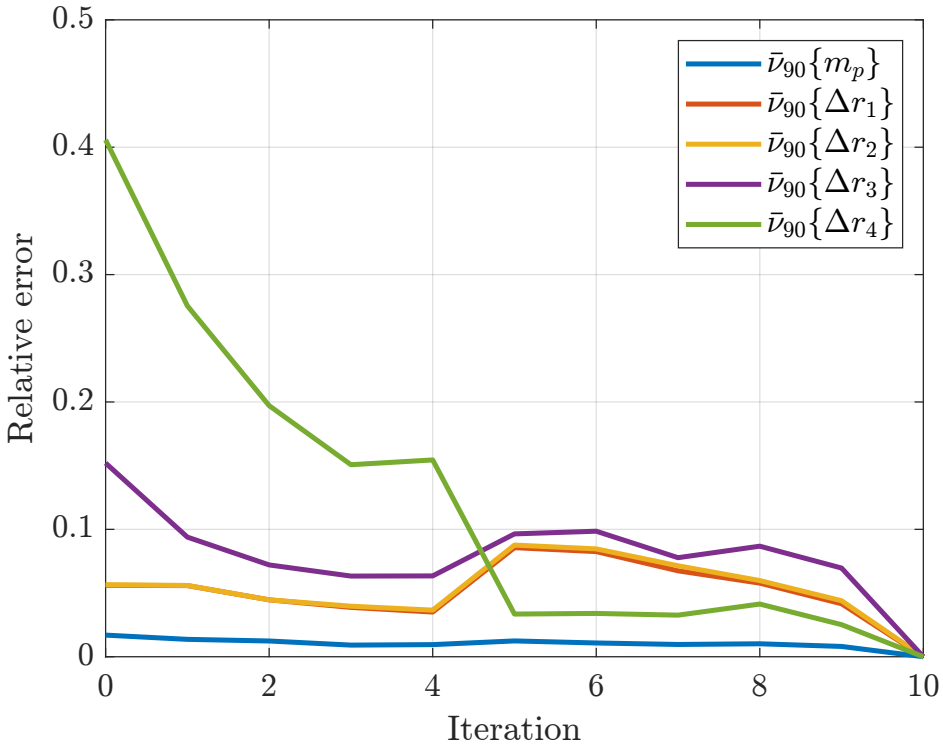


Figure 6.5: RMSE of model error normalised by dividing by the standard deviation of the data.

the last iteration, the model contains in its training data the points it is being tested with, which for the case of Kriging models results in zero error, as the Fig. 6.5 also shows. The results in Fig. 6.5 show that we do not have a monotonic convergence, but the model error tends to decrease when it starts large, or remain small otherwise.

6.4 Many-Objective Optimisation Results

Running the MO search algorithm in Algorithm 1, a number of robust solutions to the asteroid tour problem are obtained. Some of these are presented in Table 6.3. There is an interesting pattern in these results, which becomes clear when they are plotted in a spider plot, in Fig. 6.6[1]. Note that the axis on $\overline{\nu}(m_p)$ is broken for better visualisation. The shaded area is drawn between the minima and maxima of each objective, and each line corresponds to a different solution. The Δr upper quantiles $\overline{\nu}(\Delta r.)$ do not usually

[1]Plot based on Stack Overflow answer: `https://stackoverflow.com/a/78122416`

exhibit trade-offs among themselves, but there is a trade off between them and $\overline{\nu}(m_p)$. In other words, generally, when either $\overline{\nu}(\Delta r_\cdot)$ decreases, so do the other values of $\overline{\nu}(\Delta r_\cdot)$ decrease while $\overline{\nu}(m_p)$ increases.

Some of the solutions in Table 6.3 are represented in Figs. 6.7, 6.8, and 6.9, which show the engine throttle, azimuth, and declination, respectively, throughout the trajectory for these robust solutions.

Table 6.3: A selection of solutions of the 5 objective problem for the asteroid tour test case when using feedback. Of the 25 resulting solutions, 10 were chosen using MACS's archival strategy [56].

$\bar{\nu}_{90}(m_p)$ $[kg]$	$\bar{\nu}_{90}(\Delta r_1)$ $[AU]$	$\bar{\nu}_{90}(\Delta r_2)$ $[AU]$	$\bar{\nu}_{90}(\Delta r_3)$ $[AU]$	$\bar{\nu}_{90}(\Delta r_4)$ $[AU]$
52.6	0.0254	0.0300	0.0329	0.0673
52.2	0.0480	0.0566	0.0595	0.1167
52.4	0.0366	0.0435	0.0464	0.0962
52.5	0.0306	0.0361	0.0394	0.0835
52.3	0.0427	0.0503	0.0537	0.1082
52.5	0.0277	0.0329	0.0355	0.0770
52.3	0.0400	0.0473	0.0496	0.1024
52.4	0.0338	0.0399	0.0430	0.0903
52.2	0.0450	0.0533	0.0565	0.1124
52.7	0.0272	0.0322	0.0333	0.0638

The last trajectory in Table 6.3 is also plotted in Fig. 6.10. The observation is shown as a green diamond. After the observation, the real trajectory would change according to the state observed at that location. In the plot, however, only the nominal case ($\boldsymbol{\xi} = \boldsymbol{\xi}_0$) is shown.

The whole Pareto front lies in a 5D space, so given the difficulty of representing such a shape, instead plots of two of the objectives are shown in Fig. 6.11, for the asteroid tour test case with and without feedback. These plots make it evident that, as expected, the solutions with feedback are better than those without. In fact every

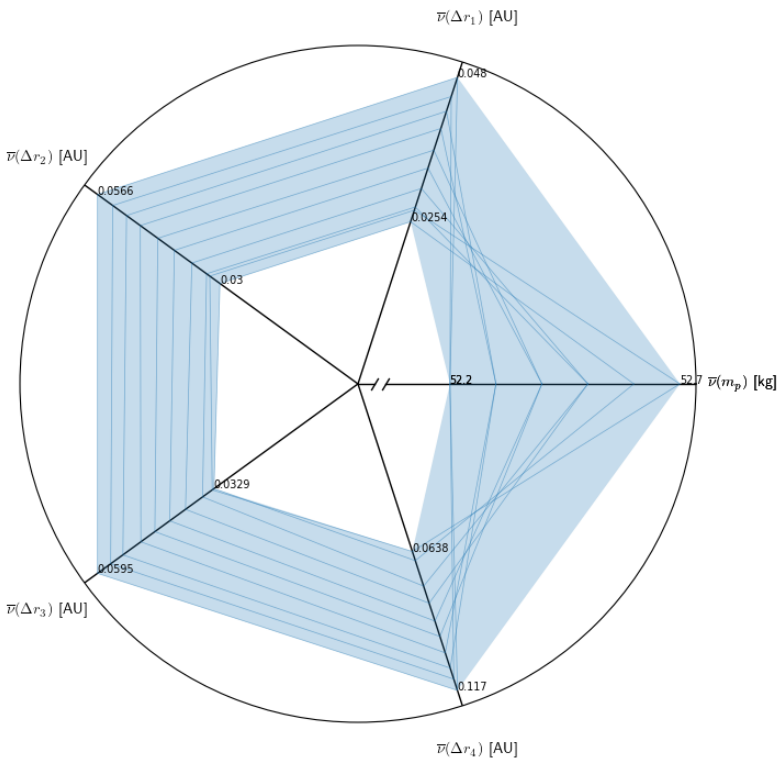
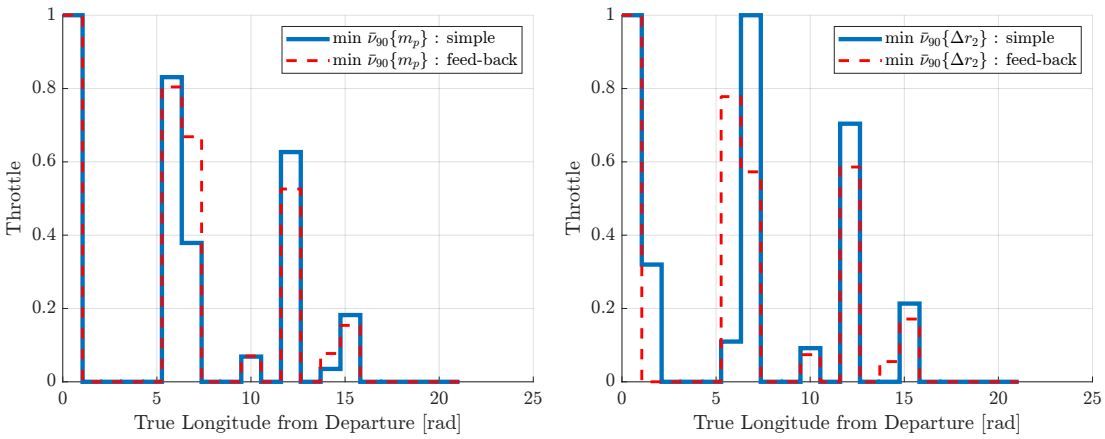


Figure 6.6: Spider plot representation of Table 6.3

single point in the Pareto front of the no feedback problem is dominated by those of the feedback problem.

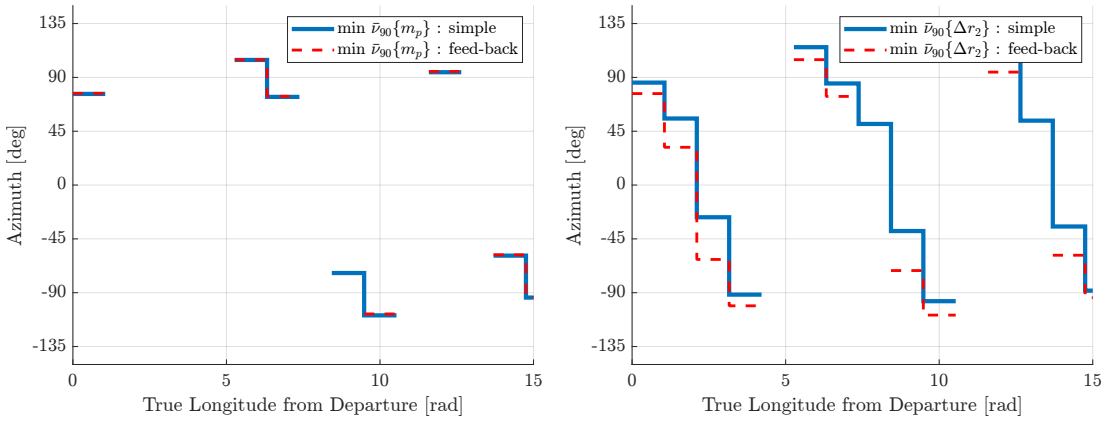
6.5 Execution Times

The usage of the MACS archival strategy to select 1000 points from a sample of 500000 took 84 seconds, and does not constitute a significant part of the total execution time. The feedback law $\mathbf{z}_2 \to \mathbf{u}_2$ took 11 seconds per sample point, and the calculation of the lower expectation V_2 for each QoI, and each sample of $\mathbf{x}_1$, took on average 2 seconds for the sample of $\boldsymbol{\nu}$. Thus, obtaining the training data for the surrogate model of V_2 took around 6 hours in total. Training the $\tilde{V}_2$ surrogate models for all QoI took 310 seconds, and each evaluation of it takes 1.3 milliseconds on average, which highlights



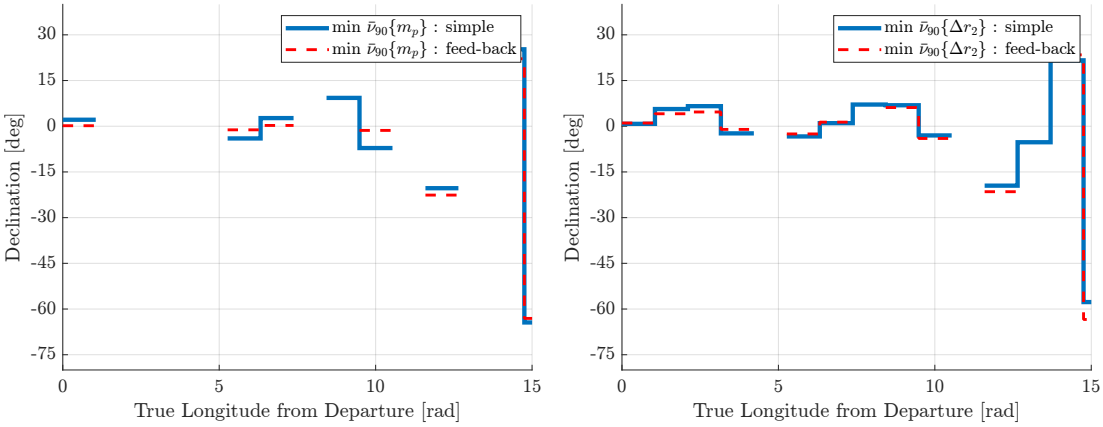
(a) Solution that minimises $\overline{\nu}\{m_p\}$ (b) Solution that minimises $\overline{\nu}\{\Delta r_2\}$

Figure 6.7: Engine throttle τ plotted against true longitude L for solutions that minimise objectives $\overline{\nu}\{m_p\}$ and $\overline{\nu}\{\Delta r_2\}$, obtained with and without feedback. For the case with feedback, the nominal case ($\boldsymbol{\xi} = \boldsymbol{\xi}_0$) is shown. The engine throttle is the control variable corresponding to the ratio between the actual thrust and the maximum thrust, the latter being an uncertain variable.



(a) Solution that minimises $\overline{\nu}\{m_p\}$ (b) Solution that minimises $\overline{\nu}\{\Delta r_2\}$

Figure 6.8: Thrust azimuth α plotted against the true longitude L for the solutions that minimise each objectives $\overline{\nu}\{m_p\}$ and $\overline{\nu}\{\Delta r_2\}$, obtained with and without feedback. An azimuth of 0^{o} corresponds to having the thrust vector pointing in the radial direction, and 90^{o} to the transverse direction. For the case with feedback, the nominal case ($\boldsymbol{\xi} = \boldsymbol{\xi}_0$) is shown.



(a) Solution that minimises $\overline{\nu}\{m_p\}$ (b) Solution that minimises $\overline{\nu}\{\Delta r_2\}$

Figure 6.9: Thrust declination β plotted against the true longitude L for the solutions that minimise each objectives $\overline{\nu}\{m_p\}$ and $\overline{\nu}\{\Delta r_2\}$, obtained with and without feedback. A declination of 0^{o} corresponds to in-plane thrust. For the case with feedback, the nominal case ($\boldsymbol{\xi} = \boldsymbol{\xi}_0$) is shown.

the time savings afforded by the surrogate model. The calculation of the control map for the whole trajectory took on average 15 seconds, while the calculation of $\overline{\nu}$ using that surrogate took 2.2s on average per QoI, leading to an average of $15 + 2.2 \times 5$ seconds for the evaluation of all objectives starting from the proxy control variable $\mathbf{z}$, i.e. Algorithm 4. There were a total of 210 evaluations of the objective vector. In addition, each run of MACS took 10 seconds, leading to a total of 1.7 minutes spent in MACS, a minority of the total run time. In total the whole algorithm took approximately 7.5 hours to complete. In all of these steps, there is the option to trade-off execution time for accuracy. More or fewer training points for $\tilde{V}_2$ and $\tilde{\overline{\nu}}$, different number of agents in MACS, etc.

6.6 Conclusion

Ultimately, the main results obtained during the course of this PhD for the Dynamic Programming based epistemic uncertainty quantification were theoretical, i.e. the theorems that show the bounds of the estimate of the epistemic cost-to-go. The numerical results, though mainly aimed at exemplifying the application of the algorithm, still showed important practical results on the quality of the surrogate models used in this

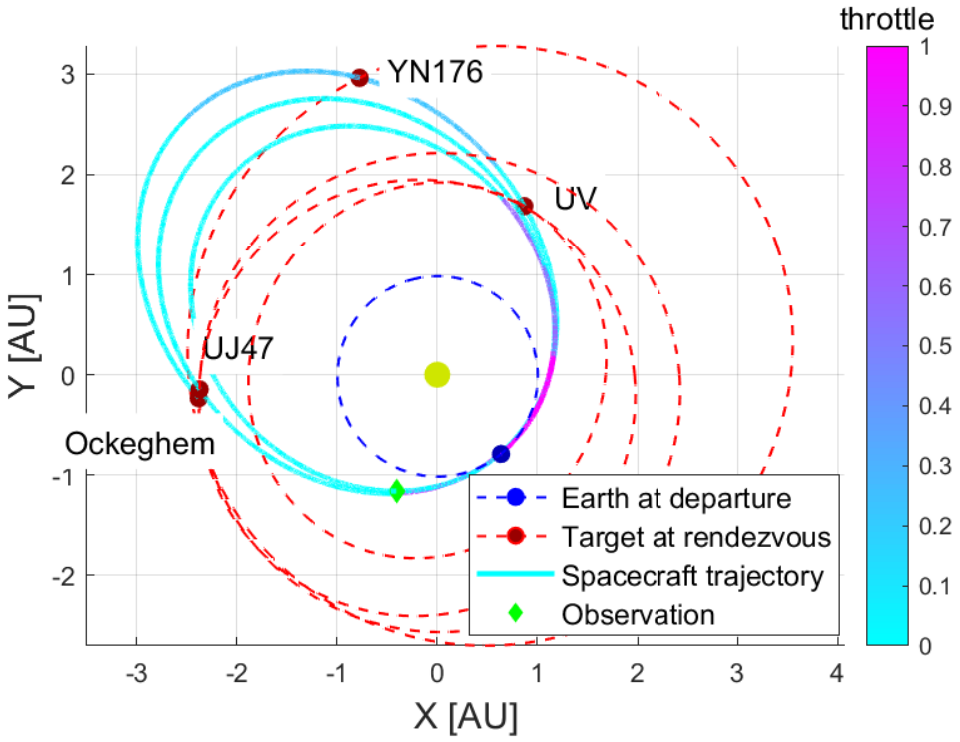
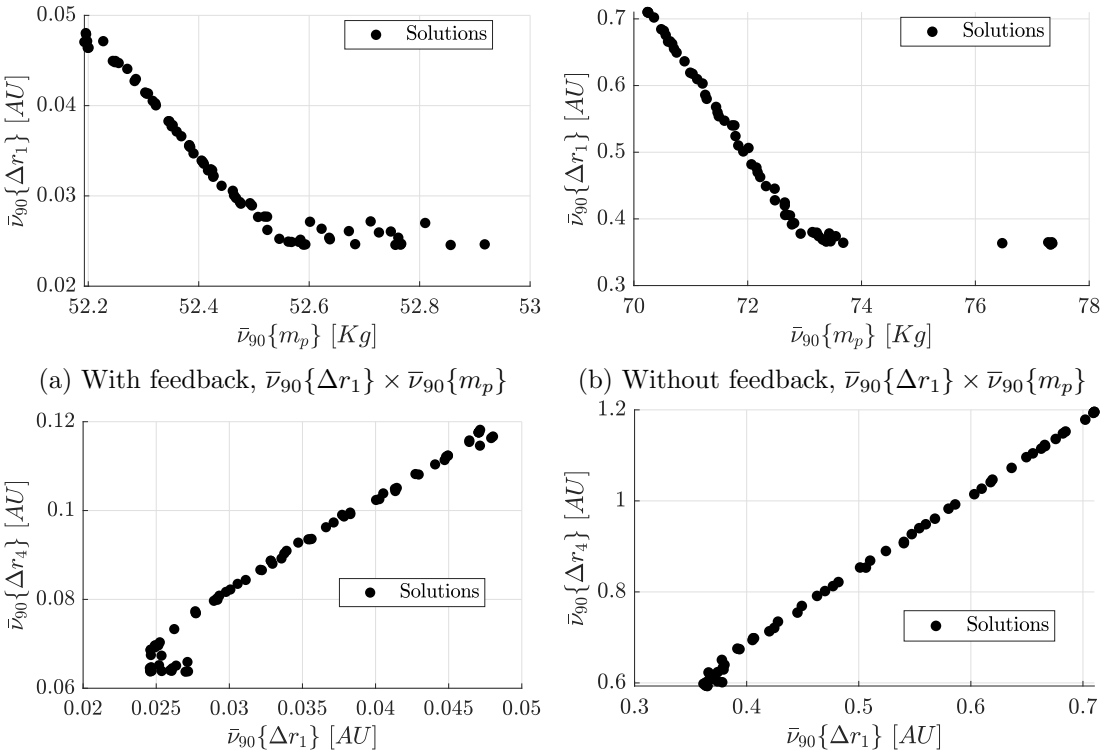


Figure 6.10: Plot of the last trajectory in Table 6.3. The spacecraft's trajectory is shown as a cyan to magenta line, which, as indicated by the colour bar on the right side of the figure, indicates the engine throttle.



(a) With feedback, $\overline{\nu}_{90}\{\Delta r_1\} \times \overline{\nu}_{90}\{m_p\}$

(b) Without feedback, $\overline{\nu}_{90}\{\Delta r_1\} \times \overline{\nu}_{90}\{m_p\}$

(c) With feedback, $\overline{\nu}_{90}\{\Delta r_4\} \times \overline{\nu}_{90}\{\Delta r_1\}$

(d) Without feedback, $\overline{\nu}_{90}\{\Delta r_4\} \times \overline{\nu}_{90}\{\Delta r_1\}$

Figure 6.11: Plots of the objectives for the estimates of the Pareto front produced by our method, for the asteroid tour test case with and without feedback.

algorithm, and the estimation of the lower expectation for the family of distributions chosen. The quality of the surrogate model is the main challenge to increasing the number of segments, and thus, observations along the trajectory. If the surrogate models for V_k are of bad quality, errors will accumulate and the value of V_1 will be inaccurate. Besides considering alternatives to the surrogate model, one option is to develop the concept in Section 7.1, as it could reduce the set of states that each surrogate model has to be valid over. This and other possible future developments to this method are some of the topics of Chapter 7.

Chapter 7

Theoretical Considerations and Future Research Perspectives

The research direction taken during this PhD for which it would be the most beneficial to dedicate further work to, is, in my view, the work of Chapter 4. It is a method for epistemic uncertainty quantification specifically designed for application to trajectory optimisation problems, as it takes advantage, in the same way as some dynamic programming methods, of the sequential nature of a trajectory. This method was at one point conceived as a joint method for both epistemic uncertainty quantification and also for trajectory optimisation, instead of the approach that ended up being taken where a separate algorithm is used for optimising the trajectory. The idea is outlined in Section 7.1, and some work was started in this direction, but unfortunately there was not enough time to conclude it.

An additional worthwhile avenue of future work would be the improvement of the surrogate model used as part of this algorithm. This approach was initially inspired by differential dynamic programming (DDP) algorithms, where the surrogate model of the cost-to-go is a simple quadratic function. However, this would not work with epistemic uncertainty quantification, because for this application, this surrogate model has to cover the entire space of possible states, while in DDP the surrogate model only needs to cover on the neighbourhood of a single state. The surrogate model thus has to cover a much greater subset of the state space, which introduces larger errors that

diminish the accuracy of the quantification of the epistemic uncertainty, as shown in Chapter 6.

There are also some interesting theoretical aspects of applying the dynamic programming approach of Chapter 4 to evidence theory [17, 60] (see also Section 3.4.2), which are discussed at length in Section 7.2. Since evidence theory is one of the most common formulations of epistemic uncertainty, another potential avenue of future work would be the practical application of the as of yet theoretical concepts in Section 7.2.

Finally, in combination with the above, it would be interesting to apply this method to more complex problem settings, such as having a larger number of intermediate observations, modelling uncertainty in the observations themselves, among other possible extensions of the work.

7.1 Dynamic Programming Trajectory Optimisation under Epistemic Uncertainty

In Chapter 4, the quantity V_k only served to estimate the lower expectation. A natural extension to this formulation would be to have it also provide a feed-back control law that maximises this lower expectation, similarly to differential dynamic programming (DDP), analogously to what is done in [53] for the aleatory case. While results were not obtained with this formulation in this work, it is presented here as a suggestion of future work.

The epistemic cost-to-go function in this case is defined as

$$V_k^+(\mathbf{u}_k, \mathbf{x}_k) = \min_{p_k \in \hat{\mathcal{Q}}_k} \int_{\Xi_k} p_k(\boldsymbol{\xi}_k) \left(V_{k+1}^*(\Phi_k(\mathbf{x}_k, \mathbf{u}_k, \boldsymbol{\xi}_k)) + l_k(\mathbf{x}_k, \mathbf{u}_k, \boldsymbol{\xi}_k) \right) \mathrm{d}\boldsymbol{\xi}_k \ , \tag{7.1}$$

$$V_k^*(\mathbf{x}_k) = \max_{\mathbf{u}_k \in U_k} V_k^+(\mathbf{x}_k, \mathbf{u}_k) \ , \tag{7.2}$$

with the same final cost as in Eq. (4.16), i.e.,

$$V_{M+1}^+(\mathbf{x}_{M+1}) = V_{M+1}^*(\mathbf{x}_{M+1}) = L(\mathbf{x}_{M+1}) \ . \tag{7.3}$$

One advantage of this formulation over the previous, evident in Eqs. (7.1) and (7.2), is that the epistemic cost-to-go is no longer a function of the whole control law "to go", i.e. $\mathbf{u}_{k:M}$, but only of the control law during each leg of the trajectory $\mathbf{u}_k$. This reduces the domain of U that the surrogate models of V_k need to cover, allowing simpler models to be used. Other advantages are that the both the optimisation of the control law and the optimisation used to calculate $\underline{\mathrm{E}}$ are applied to each leg only, making it scale well with the number of segments/observations. In addition we also get a fast to compute feedback control law.

One challenge in implementing this approach is the fact that while in classical DDP, the surrogate model of the cost-to-go only needs to be valid over neighbourhood of the nominal value of the state $\mathbf{x}_k$, with a stochastic formulation, the values of V_k^+ corresponding to the states $\mathbf{x}_k$, which may be realised for the different values of the epistemic variables $\boldsymbol{\xi}_{1:k}$, must also be adequately approximated.

Another challenge is the fact that this approach is geared towards single objective optimisation, and extension to many objectives is not trivial, apart from the use of scalarisation techniques.

7.2 Dynamic Programming Approach to Evidence Theory

A few theoretical observations are pertinent when considering an Evidence Theory formalisation of the epistemic uncertainty, as defined in Section 3.4, in particular relating to extending epistemic uncertainty formulations in each variable $\boldsymbol{\xi}_k$ to the vector containing all such variables $\boldsymbol{\xi}$. Suppose the uncertainty in $\boldsymbol{\xi}_k$ is described using evidence theory, where the proposition is represented by set $A \subset \Xi_k$, and one has focal elements $\theta_{k,i} \subset \Xi_k$, with basic probability assignment (BPA) $\mathrm{bpa}(\theta_{k,i})$, and $\theta_{k,i} \in \mathcal{T}_k$. For simplicity, let us also restrict ourselves to the case where the focal elements with non-zero BPA do not intersect. In this case, the belief can be obtained by computing the lower expectation with the following family of distributions following definition 3.4.1, from

which Eq. (3.39) is repeated here for convenience:

$$\hat{\mathcal{Q}}_k = \left\{ p \in \mathcal{P}(\Xi_k) : \int_{\theta_{k,i}} p(\boldsymbol{\xi}_k) \mathrm{d}\boldsymbol{\xi}_k = \mathrm{bpa}(\theta_{k,i}) \right\} \tag{7.4}$$

In the literature [13, 22], univariate focal elements are extended to the multivariate case by means of a Cartesian product, resulting in an exponential growth of the number of multivariate focal elements with the number of epistemic variables. If one is considering the epistemic variables from segment k to M, the multivariate focal elements are then given as

$$\Theta_{i_k,i_2,\ldots,i_M} = \theta_{k,i_k} \times \theta_{k+1,i_{k+1}} \times \cdots \times \theta_{M,i_M} \ , \tag{7.5}$$

and the respective BPAs are defined as the products:

$$\mathrm{bpa}(\Theta_{i_k,i_{k+1},\ldots,i_M}) = \mathrm{bpa}(\theta_{1,i_k}) \times \mathrm{bpa}(\theta_{2,i_{k+1}}) \times \cdots \times \mathrm{bpa}(\theta_{M,i_M}) = b_{i_k,i_{k+1},\ldots,i_M} \ . \tag{7.6}$$

Definition 7.2.1. The value of the belief that is obtained with the focal elements and BPAs of Eqs. (7.5, 7.6) is written as $B_k^{\times}$, where as before $B^{\times} = B_1^{\times}$. The family of distributions $\mathcal{Q}_{k:M}^{B\times}$ that allows the correspondence between $\underline{\mathrm{E}}$ and $B_k^{\times}$ is defined the same way as Definition 3.4.1 but using the aforementioned focal elements and BPAs.

The value of $B_k^{\times}$ does not, in general, match either $\underline{\mathrm{E}}_k^i$, $\underline{\mathrm{E}}_k^d$, or V_k. It can, however, be shown that $\underline{\mathrm{E}}_k^d \leq B_k^{\times} \leq V_k \leq \underline{\mathrm{E}}_k^i$. Both of the inequalities involving $B_k^{\times}$ are now presented in separate theorems, alongside the respective proofs.

Theorem 7. *The belief $B_k^{\times}$, in Definition 7.2.1, is greater or equal to $\underline{\mathrm{E}}_k^d$, when the latter is defined with the families of distributions in Definition 4.3.1, with $\hat{\mathcal{Q}}_k$ as in Eq. (7.4), i.e.*

$$B^{\times} \geq \underline{\mathrm{E}}^d \ . \tag{7.7}$$

For simplicity, this statement is restricted to the case where the focal elements do not intersect. As for the previous theorems, without loss of generality this is proven for $k = 1$.

Proof. The set $\mathcal{Q}^{B\times}$ of Definition 7.2.1 can also be written as the set of distributions

p_B that satisfy, for all sequences $i_1, \ldots, i_M$,

$$\int_{\Theta_{i_1,\ldots,i_M}} p_B(\boldsymbol{\xi}_{1:M}) \mathrm{d}\boldsymbol{\xi} = b_{i_1,\ldots,i_M} \tag{7.8}$$

while the definition of $\mathcal{Q}$ that results in $\underline{\mathrm{E}}^d$, in Eq. (4.3), when combined with Eq. (7.4), corresponds to the set of distributions p_d such that, for all $1 \leq j \leq M$ and i_j,

$$\int_{\theta_{j,i_j}} \mathcal{M}_j[p_d](\boldsymbol{\xi}_j) \mathrm{d}\boldsymbol{\xi}_j = b_{j,i_j} \ . \tag{7.9}$$

This theorem is proven by showing that Eq. (7.8) implies Eq. (7.9), as that means the search space in the optimisation that defines $B^{\times}$, $\mathcal{Q}^{B\times}$, is a subset or equal to the one which defines $\underline{\mathrm{E}}^d$, $\mathcal{Q}$.

Equation (7.9) can be re-written, using the definition of the operator $\mathcal{M}$, as

$$\int_{\Xi_1} \cdots \int_{\Xi_{j-1}} \int_{\theta_{j,i_j}} \int_{\Xi_{j+1}} \cdots \int_{\Xi_M} p_d(\boldsymbol{\xi}) \mathrm{d}\boldsymbol{\xi} = b_{j,i_j} \ . \tag{7.10}$$

Without loss of generality, assume that $\bigcup_{i_j} \theta_{j,i_j} = \Xi_j$. If this is not the case, one can redefine the set Ξ_j such that this equality holds. This allows writing Eq. (7.10) as

$$\sum_{i_1} \cdots \sum_{i_{j-1}} \sum_{i_{j+1}} \cdots \sum_{i_M} \int_{\Theta_{i_1,\ldots,i_{j-1},i_j,i_{j+1},\ldots,i_M}} p_d(\boldsymbol{\xi}) \mathrm{d}\boldsymbol{\xi} = b_{k,i_k} \ . \tag{7.11}$$

If p_d is replaced by p_B in Eq. (7.11), using Eq. (7.8) for the definition of set $\mathcal{Q}$, and the definition in Eq. (7.6) is used, the left-hand-side of Eq. (7.11) becomes

$$\sum_{i_1} b_{1,i_1} \cdots \sum_{i_{k-1}} b_{k-1,i_{k-1}} b_{k,i_k} \sum_{i_{k+1}} b_{k+1,i_{k+1}} \cdots \sum_{i_M} b_{M,i_M} = b_{k,i_k} \ , \tag{7.12}$$

which shows that if $p_B \in Q^{B\times}$, then $p_B \in Q$, in other words, $Q^{B\times} \subseteq Q$, which proves this theorem. □

Theorem 8. *The belief $B^{\times}$, defined by taking the Cartesian product of the focal elements and the product of the BPAs as defined in Eqs. (7.5, 7.6), is less or equal to V_1*

defined with the $\hat{\mathcal{Q}}_k$ in Eq. (7.4), i.e.

$$B^{\times} \leq V_1 \ . \tag{7.13}$$

Combined with theorems 5, 6, and 7, this means $\underline{\mathrm{E}}^d \leq B^{\times} \leq V_1 \leq \underline{\mathrm{E}}^i$.

Proof. When applied to families of distributions in Eq. (7.4), the values of $B^{\times}$ and V_1 are given by Eqs. (7.14) and (7.15), respectively:

$$B^{\times} = \sum_{i_1} \ldots \sum_{i_M} \mathrm{bpa}(\theta_{1,i_1}) \times \ldots \times \mathrm{bpa}(\theta_{M,i_M}) \min_{\boldsymbol{\xi}_k \in \theta_{j,i_j} \forall j} h(\boldsymbol{\xi}) \tag{7.14}$$

$$V_1 = \sum_{i_1} \mathrm{bpa}(\theta_{1,i_1}) \min_{\boldsymbol{\xi}_1 \in \theta_{1,i_1}} \ldots \sum_{i_M} \mathrm{bpa}(\theta_{M,i_M}) \min_{\boldsymbol{\xi}_M \in \theta_{M,i_M}} h(\boldsymbol{\xi}) \tag{7.15}$$

Note that writing $\boldsymbol{\xi} \in \Theta_{i_1,\ldots,i_N}$ is equivalent to writing $\boldsymbol{\xi}_k \in \theta_{k,i_k} \forall k$. Equation (7.14) can therefore be obtained from (7.15) by moving all minimisations over θ_{j,i_j} to within the sums, naturally resulting in $B^{\times} \leq V_1$. □

One interpretation of Theorem 7 is that the extension of Dempster-Shafer structures from single variables to their joint distributions proposed in [13, 22] is not the most conservative possible, as it is not as conservative as $\underline{\mathrm{E}}^d$, while still not being as optimistic as the assumption of independence inherent to the definition of $\underline{\mathrm{E}}^i$. The examples in Section 7.2.1, besides showing that in general $B^{\times} \neq \underline{\mathrm{E}}^d$ and $B^{\times} \neq \underline{\mathrm{E}}^i$, also intend to provide intuition to the previous statement.

7.2.1 Demonstrative Examples

This section contains examples that, like the ones of Section 4.5, are intended to show that the inequalities in Theorems 7 and 8 can be observed in the strict sense, i.e., that they are not equalities in general. In addition, they are intended to provide some intuition to those theorems.

Firstly, let us look at a case where $\underline{\mathrm{E}}_d \leq B^{\times}, \underline{\mathrm{E}}_i$. Consider two scalar uncertain variables ξ_1 and ξ_2, on each of which are defined two focal elements, resulting in four focal element products Θ, as defined in Eq. (7.5). A set A is such that $\Theta_{2,2} \subset$

A, but none of the other focal elements are fully contained in A. This scenario is illustrated in Fig. 7.1. The distribution functions p corresponding to distributions that are optimal in terms of minimizing $\underline{\mathrm{E}}^i$ can be written, without loss of generality, as discrete distributions over the numbered points in the Figure, following the same argument as in [27], and discussed in Section 3.4, in other words, those points are the centers of the Dirac delta kernels that form the distribution. A geometric consequence of $p \in \hat{\mathcal{Q}}$ is that, if p is written as a sum of identical kernels, these must form a grid, as that is required to have independence between ξ_1 and ξ_2. Therefore, the numbered points in Fig. 7.1 are the corners of a rectangle aligned with the axes.

Because in the definition of the belief, there must be probability mass in each focal element, one of which is completely within set A, the belief of set A must be $b_{1,2}b2, 2$, the bpa of set $\Theta_{2,2}$, as that is the only one fully contained in A. But for $\mathcal{Q}$ it is sufficient that the marginal distributions have the correct amount of probability mass on the focal elements. This translates into the following equations,

$$\begin{aligned} p_1 + p_2 &= b_{1,1} \\ p_3 + p_4 &= b_{1,2} \\ p_2 + p_3 &= b_{2,1} \\ p_1 + p_4 &= b_{2,2} \ , \end{aligned} \tag{7.16}$$

where we refer to the probability of point i in Fig. 7.1 as p_i.

Because the distributions in $\mathcal{Q}$ only have their marginal distributions constrained to have the correct mass in each focal element, it is possible to assign all mass to points 1,2,3 with none in 4. The assignment of probability mass in the second row of Table 7.1 shows how this can be achieved. The same table also shows assignments of probability mass that are optimal for $\mathcal{Q}^{B\times}$ and $\hat{\mathcal{Q}}$. Because both of the other sets, $\hat{\mathcal{Q}}$ and $\mathcal{Q}^{B\times}$, have definitions that force there to be probability mass in each focal element product Θ (provided the BPAs are not zero), they both have non-zero $\underline{\mathrm{E}}$, which in our case corresponds to the mass assigned to p_4 in Table 7.1. Therefore, in this example, one has the strict inequalities $\underline{\mathrm{E}}^d < B^\times, \underline{\mathrm{E}}^i$.

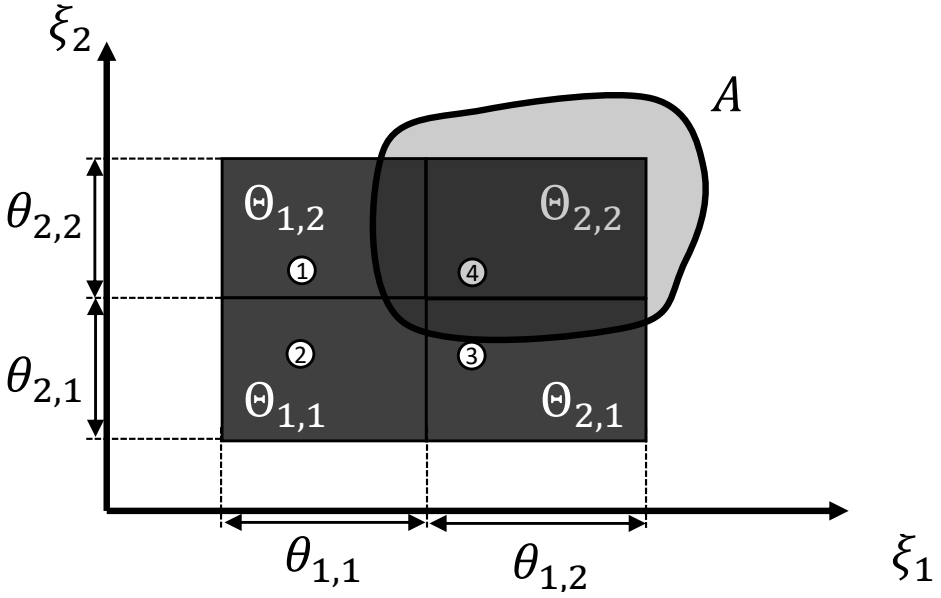


Figure 7.1: An example where $\underline{\mathrm{E}}^d < B^{\times}, \underline{\mathrm{E}}^i$.

Table 7.1: Probabilities of the labelled points in Fig. 7.1 so that distributions are optimal and within each family of distributions

	p_1	p_2	p_3	$p_4 = p(A)$
$p \in Q^{B\times}$	$b_{1,1}b_{2,2}$	$b_{1,1}b_{2,1}$	$b_{1,2}b_{2,1}$	$b_{1,2}b_{2,2}$
$p \in Q$	$b_{2,2}$	$b_{1,1} - b_{2,2}$	$b_{1,2}$	0
$p \in \hat{Q}$	$b_{1,1}b_{2,2}$	$b_{1,1}b_{2,1}$	$b_{1,2}b_{2,1}$	$b_{1,2}b_{2,2}$

In the next example, in Fig. 7.2, it holds that $\underline{\mathrm{E}}^d, B^{B\times} < \underline{\mathrm{E}}^i, V_1$. As before, the probability is constrained, without loss of generality, to the labeled points in Fig. 7.2. There are now 5 labeled points, of which points 1-4 are again placed such that they are the corners of a rectangle aligned with the axes, while the fifth point is not. The set of inequalities that must hold for $p \in \mathcal{Q}$ is now

$$\begin{aligned} p_1 + p_2 + p_5 &= b_{1,1} \\ p_3 + p_4 &= b_{1,2} \\ p_2 + p_3 + p_5 &= b_{2,1} \\ p_1 + p_4 &= b_{2,2} \ . \end{aligned} \tag{7.17}$$

In this case there is a variety of solutions with $p \in \mathcal{Q}$ and $p(A) = 0$ that meet these conditions. An example is shown in Table 7.2.

The set A is such that for all ξ_1, if there is a ξ_2 such that $[\xi_1, \xi_2] \in \Theta_{1,1} \setminus A$, then there is no ξ_2 such that $[\xi_1, \xi_2] \in \Theta_{1,2} \setminus A$, and vice versa. The definition of set $\hat{\mathcal{Q}}$ is such that if any probability is assigned to a point $[\xi_1, \xi_2]$ in $\Theta_{1,1}$, then some mass must be assigned to a point with the same value of ξ_1 in $\Theta_{1,2}$, provided all BPAs are non-zero. What this means is that when $p \in \hat{\mathcal{Q}}$, either point 1 or 2 will be inside A. However, there are points both in $\Theta_{1,2}$ and $\Theta_{1,1}$ that lie outside A. This is the reason why in this case $B \leq E^i$, since $B = 0$ by assigning the BPA associated with $\Theta_{1,1}$ to point 5, but for a distribution in $\hat{\mathcal{Q}}$ some mass must be assigned to point 2, or more precisely, to some point within $A \cap (\Theta_{1,1} \cup \Theta_{1,2})$.

Table 7.2 shows probability distributions that are optimal within sets $\mathcal{Q}$, $\mathcal{Q}^{B\times}$, and $\hat{\mathcal{Q}}$, assuming $b_{2,2} \leq b_{2,1}$. Because point 2 is the only one within A, the mass assignments to p_2 in that table correspond to the values of $B^\times, \underline{\mathrm{E}}^d, \underline{\mathrm{E}}^i$.

Let us now take the same example and see what values V_2 can take. Figure 7.3 extends the Fig. 7.2 to show $V_2(\xi_1)$ as a plot. The definition in Eq. (4.15) in this case becomes

$$V_2(\xi_1) = \min_{p_2 \in \hat{\mathcal{Q}}_2} \int_{\Xi_2} h(\boldsymbol{\xi}) \, \mathrm{d}\xi_2 = \sum_{i_2} b_{2,i_2} \min_{\xi_2 \in \theta_{2,i_2}} (\boldsymbol{\xi} \in A) = \sum_{\substack{i_2 \\ \exists \xi_2 \in \theta_{2,i_2}:\ \boldsymbol{\xi} \in A}} b_{2,i_2} \ , \tag{7.18}$$

Table 7.2: Probabilities of the labelled points in Fig. 7.2 so that distributions are optimal and within each family of distributions

	p_1	$p_2 = p(A)$	p_3	p_4	p_5
$p \in \mathcal{Q}^{B\times}$	$b_{1,1}b_{2,2}$	0	$b_{1,2}b_{2,1}$	$b_{1,2}b_{2,2}$	$b_{1,1}b_{2,1}$
$p \in \mathcal{Q}$	$b_{2,2}$	0	$b_{1,2}$	0	$b_{1,1} - b_{2,2}$
$p \in \hat{\mathcal{Q}}$	$b_{1,1}b_{2,2}$	$b_{1,1}b_{2,1}$	$b_{1,2}b_{2,1}$	$b_{1,2}b_{2,2}$	0

and

$$V_1 = \min_{p_1 \in \hat{\mathcal{Q}}_1} \int_{\Xi_1} V_2(\xi_1)\, \mathrm{d}\xi_1 = \sum_{i_1} b_{1,i_1} \min_{\xi_1 \in \theta_{1,i_1}} V_2(\xi_1)\ . \tag{7.19}$$

The function $V_2(\xi_1)$ takes the form of a step function, depending on whether, for a particular ξ_1, it is possible to assign the BPAs to the focal elements θ_{2,i_2} outside of A. For example, for the ξ_1 in the first step, where $V_2(\xi_1) = b_{2,1}$, it is possible to find $\xi_2 \in \theta_{2,2}$ such that $[\xi_1, \xi_2] \notin A$, but it is not possible to find the same when $\xi_2 \in \theta_{2,1}$. Therefore, the value of $V_2(\xi_2)$ in this first step is the BPA of $\theta_{2,1}$, $b_{2,1}$. The calculation of V_1 then results in $V_1 = b_{1,1}b_{2,1}$. Since $B^\times$ is zero, this is a case where it holds that $B^\times < V_1$, representing an example where the inequality in Theorem 8 is strict.

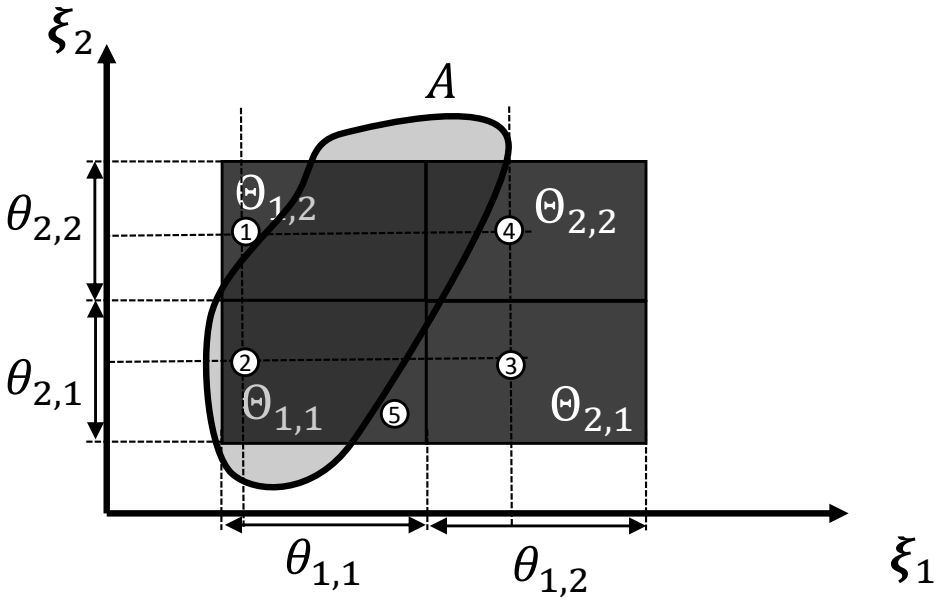


Figure 7.2: An example where $B, \underline{\mathrm{E}}^d < \underline{\mathrm{E}}^i, V_1$.

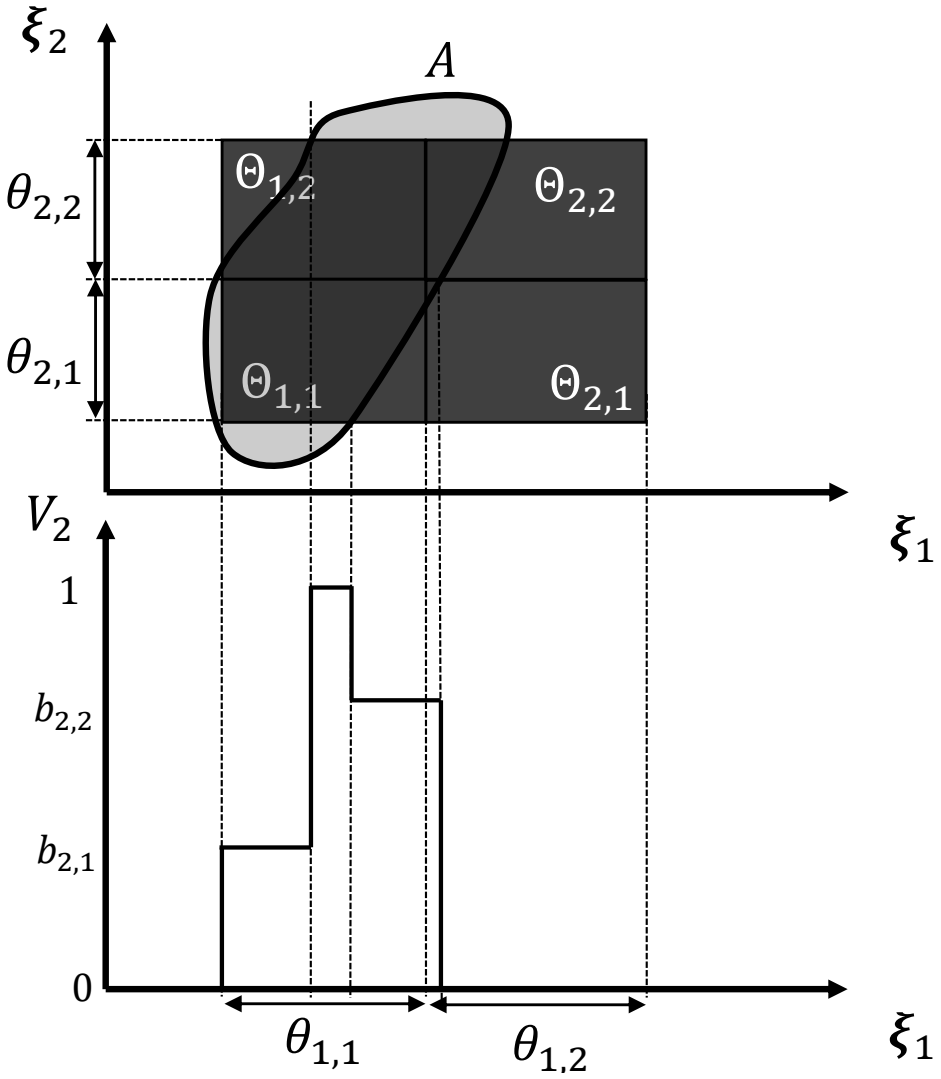


Figure 7.3: Detailed diagram on the calculation of V.

Chapter 8

Conclusions

The main developments presented in this thesis, in line with the research objectives summarised in Section 1.1, are all related to reducing the computational load of optimising trajectories subject to epistemic uncertainty. In this work, progress was made along several fronts, including efficient algorithms supported by novel theoretical developments.

This PhD was aimed at developing many-objective trajectory optimisation algorithms under epistemic uncertainty. While the trajectory optimisation part of the work was initially foreseen to be the main focus of the PhD, the challenges in efficiently quantifying epistemic uncertainty gradually led to this component of the work becoming a bigger component of it. This evolution is roughly reflected in the order of the chapters of part I. Even now, this seems to be the main challenge to be overcome in this area of research.

Ultimately, the main contributions of this work ended up focusing on either novel methods for efficiently quantifying epistemic uncertainty, such as the dynamic programming based approach, or on methods for making the search for optimal trajectories more efficient in terms of reducing the number of evaluations of the objective function, such as the control and threshold maps. The following is a more detailed break-down by chapter of the contributions of this PhD.

In Chapter 2, the focus was on methods for reducing the number of times the computation of the quantification of the epistemic uncertainty. These included control

maps, which allow a reduction of the dimensionality of the problem, and focusing the search effort in regions of the design space which are likely to contain good solutions. These approaches are tested and compared in Chapter 5, for a test case based on a rendezvous with an asteroid, and for a flyby tour of multiple asteroids.

The quantification of the epistemic uncertainty was formulated in terms of the lower expectation, which is shown to be a broadly applicable formulation [71], with some examples in Chapter 3. These included a formulation based on Bernstein polynomials, for which a combinatorial optimisation approach is proposed to efficiently estimate the lower expectation in this case. This approach was compared in Chapter 5 with other traditional combinatorial optimisation algorithms to assess its accuracy and efficiency relative to them and found it to compare favourably. Another approach in Chapter 3 was based on the interval valued momentum constraints, using a kernel based approach. In addition to some theoretical development, the relationship of accuracy and computational cost with the number of these kernels is found experimentally in Chapter 6.

Chapter 3 included a brief theoretical discussion on how the formulation of lower expectation can also be applied to evidence theory, which was particularly relevant to highlight the fact that the developments of Chapter 4 can also be applied to it.

Chapter 3 also introduces threshold maps, which are motivated by the desire to avoid wasting function evaluations in areas of the search space where the threshold are either too large that achievement of mission objectives becomes certain, or too low that it becomes impossible. These are shown to improve the efficiency of the search over the thresholds of the quantities of interest in results of Chapter 5.

Finally, Chapter 4 proposes a novel dynamic programming approach to compute the lower expectation of a quantity of interest in multi-segment optimal control problems affected by epistemic uncertainty. It was theoretically proven that under suitable constraints on the distributions, the approach returns guaranteed bounds on the value of the lower expectation. The lower bound is given by the lower expectation resulting from assuming only information on the distributions of the uncertain variables covering each segment, while the upper bound is given by the lower expectation obtained when

in addition to that information, one also knows that these variables on each segment are independent. Despite the simplicity of their definitions, neither of these lower expectations can be computed in a segmented approach, because they are defined as an optimisation of an expected value with respect to the uncertain variables covering the whole trajectory. The proposed dynamic programming approach, on the other hand, is defined to allow a segmented approach by following the pattern of the cost-to-go function of dynamic programming. The estimate returned by the proposed dynamic programming approach can, therefore, be seen as an easier-to-compute conservative approximation of $\underline{\mathrm{E}}_i$, with an error that is limited by the difference between $\underline{\mathrm{E}}_i$ and $\underline{\mathrm{E}}_d$. For the upper bound, no assumptions are needed on the type of information available to constrain the distributions of the uncertain variables, making it very general. For the lower bound, there is the assumption of convexity of the sets of distributions for each segment, but this nonetheless applies to some cases, such as the example presented in this work where the statistical moments are constrained. Even if the sets of distributions are not convex, a lower bound can still be obtained by defining $\underline{\mathrm{E}}_d$ on the convex hulls of these sets.

It was also demonstrated that the method achieves a complexity that is polynomial with the number of segments. This implies that problems with long sequences of segment dependent uncertainty, coming from stochastic processes or measurements, become computationally tractable.

As an illustrative example, the overall methodology was applied to the design of a low-thrust asteroid fly-by tour in Chapter 6. It was found that introducing observations drastically reduce the quantiles on propellant consumption and constraint satisfaction. This is to be expected given that the trajectory post observation can be optimised to minimise the effect of uncertainty. The main limitation of the experimental results comes from the assumption of negligible uncertainty on the observables. However, the uncertainty in the observables at each segment can be simply included in the list of uncertain quantities. Thus the overall methodology and the theoretical results hold true even in the case of uncertain observations.

Appendix A

Proofs of Lemmas

This Appendix contains proofs for lemmas 1 and 2. The lemmas themselves are also repeated here for clarity.

Lemma 1. If $\underline{\mathrm{E}}$ is defined using the family $\mathcal{P}_m$, there is always at least one distribution $\hat{p} \in \mathcal{P}_j$ such that $\underline{\mathrm{E}}(I) = \mathrm{E}(I; \hat{p})$.

Proof. The problem of finding $\underline{\mathrm{E}}$ with $\mathcal{P} = \mathcal{P}_m$ can be written as

$$\min_{\mathbf{c}} \sum_{\mathbf{j} \in \mathcal{J}} \lambda_{\mathbf{j}} c_{\mathbf{j}}, \tag{A.1}$$

where

$$\lambda_{\mathbf{j}} = \int_{\Xi} I(\boldsymbol{\xi}) B_{\mathbf{j}}(\tau(\boldsymbol{\xi})) \mathrm{d}\boldsymbol{\xi} = \mathrm{E}(I; B_{\mathbf{j}}). \tag{A.2}$$

This is a linear problem. Let $\tilde{\mathbf{j}}$ be such that $\lambda_{\tilde{\mathbf{j}}} = \min_{\mathbf{j}} \lambda_{\mathbf{j}}$. The original problem can be written as:

$$\min_{\mathbf{c}} \lambda_{\tilde{\mathbf{j}}} + \sum_{\mathbf{j} \in \mathcal{J}} (\lambda_j - \lambda_{\tilde{\mathbf{j}}}) c_{\mathbf{j}} \geq \lambda_{\tilde{\mathbf{j}}}. \tag{A.3}$$

It is now evident that $\min_{\mathbf{c}} \sum_{\mathbf{j} \in \mathcal{J}} \lambda_{\mathbf{j}} c_{\mathbf{j}} = \lambda_{\tilde{\mathbf{j}}} = \mathrm{E}(I; B_{\tilde{\mathbf{j}}})$. Since $B_{\tilde{\mathbf{j}}} \in \mathcal{P}_j$, the lemma is proven. □

Lemma 2. If $\underline{\mathrm{E}}$ is defined using the family $\mathcal{P}_u$, there is always at least one distribution

$\hat{p} \in \mathcal{P}_j$ such that $\underline{\mathrm{E}}(I) = \mathrm{E}(I; \hat{p})$.

Proof. Any distribution $\hat{p} \in \mathcal{P}_j$ can be defined in terms of $\mathcal{P}_u$ by choosing a coefficient vector $\hat{\mathbf{c}}$ such that each marginal distribution is a basis function,

$$\forall k \exists \tilde{j} : \hat{\mathbf{c}}_{\tilde{j}}^{(k)} = 1 \text{ and } \hat{\mathbf{c}}_{j}^{(k)} = 0 \ \forall j \neq \tilde{j}. \tag{A.4}$$

Let $\tilde{\mathbf{c}}$ be a vector such that $\underline{\mathrm{E}}(I) = \mathrm{E}(I, p_u(\boldsymbol{\xi}; \tilde{\mathbf{c}}))$, but which does not satisfy Eq. (A.4). We will show that it can be used to obtain a vector $\hat{\mathbf{c}}$ that satisfies both $\underline{\mathrm{E}}(I) = \mathrm{E}(I, p_u(\boldsymbol{\xi}; \hat{\mathbf{c}}))$ and Eq. (A.4).

If $\tilde{\mathbf{c}}$ does not satisfy that constraint, then for at least one s, $\tilde{\mathbf{c}}^{(s)}$ must have more than one non-zero element. Consider the problem of optimising $\min_{\mathbf{c}^{(s)}} \mathrm{E}(I; p_u(\boldsymbol{\xi}; \mathbf{c}))$, where all elements of $\mathbf{c}$ are fixed except for those in $\mathbf{c}^{(s)}$. This can be written as

$$\min_{\mathbf{c}^{(s)}} \sum_{j=0}^{q_s} \lambda_j c_j^{(s)}, \tag{A.5}$$

where

$$\lambda_\eta = \int_{\Xi} I(\boldsymbol{\xi}) \prod_{k \neq s} \sum_{j=0}^{q_k} \left[b_{j;q_k}(\tau_k(\xi_k)) \tilde{c}_j^{(k)} \right] b_{\eta;q_s}(\tau_s(\xi_s)) \mathrm{d}\boldsymbol{\xi}. \tag{A.6}$$

This is a linear problem. This part of the proof proceeds similarly to Lemma 1. Let $\tilde{j}$ be such that $\lambda_{\tilde{j}} = \min_j \lambda_j$. The original problem can be written as:

$$\min_{\mathbf{c}^{(s)}} \lambda_{\tilde{j}} + \sum_{j=0}^{q_s} (\lambda_j - \lambda_{\tilde{j}}) c_j^{(s)} \geq \lambda_{\tilde{j}}. \tag{A.7}$$

It is now evident that $\min_{\mathbf{c}^{(s)}} \sum_{j=0}^{q_s} \lambda_j c_j^{(s)} = \lambda_{\tilde{j}}$, which can be obtained by making $c_{\tilde{j}}^{(s)} = 1$ and $c_j^{(s)} = 0 \ \forall j \neq \tilde{j}$. Altering $\tilde{\mathbf{c}}$ in this way does not change the value of E, since $\tilde{\mathbf{c}}$ minimised $\mathrm{E}(I, p_u(\boldsymbol{\xi}; \mathbf{c}))$, it must have also minimised Problem (A.5), and so $\sum_{j=0}^{q_s} \lambda_j \tilde{c}_j^{(s)} = \lambda_{\tilde{j}}$.

Repeating this process for all s for which $c^{(s)}$ has more than one non-zero element, results in a new vector $\hat{\mathbf{c}}$ which satisfies Eq. (A.4), and for which $\mathrm{E}(I, p_u(\boldsymbol{\xi}; \hat{\mathbf{c}})) = \mathrm{E}(I, p_u(\boldsymbol{\xi}; \tilde{\mathbf{c}})) = \underline{\mathrm{E}}(I)$. □

Appendix B

Accuracy of Linear Interpolation Estimate of C_k^{-1}

Let f be a monotonically increasing function that is estimated using linear interpolation. The nodes for interpolation are $x_0 < x_1 < \cdots < x_N$. If a query point x is such that $x_i \leq x \leq x_{i+1}$, the value of $f(x)$ is estimated as

$$\tilde{f}(x) = \frac{x - x_i}{x_{i+1} - x_i} f(x_i) + \frac{x_{i+1} - x}{x_{i+1} - x_i} f(x_{i+1}) \ . \tag{B.1}$$

Evidently, this estimate obeys $\tilde{f}(x) \in [f(x_i), f(x_{i+1})]$. Due to the monotonicity of f, the exact function value $f(x)$ also lies in this interval. Therefore, the error of the linear interpolation estimate for this query point cannot be greater than $f(x_{i+1}) - f(x_i)$, so an upper bound of the error for any query point x is

$$\left| \tilde{f}(x) - f(x) \right| \leq \max_i f(x_{i+1}) - f(x_i) \ . \tag{B.2}$$

Now let $f = C_k^{-1}$, which is a monotonically increasing function. This function is being approximated with linear interpolation by calculating values of its inverse $f^{-1} = C_k$. Let values $y_0 < \cdots < y_N$ be such that $f(x_i) = y_i$. Calculating $x_i = f^{-1}(y_i) = C_k(y_i)$, linear interpolation is applied as previously described to obtain estimates of $C_k^{-1}(x)$. Following the same reasoning, the error upper bound is $\max_i(y_{i+1} - y_i)$. The values of y can be freely chosen, and so, to minimise this error upper bound they are

defined to be equidistant, simplifying the error upper bound to $(y_N - y_0)/N$. Since the domain of function C_k is the interval $[0, 1]$, $y_0 = 0$ and $y_N = 1$, which simplifies the error upper bound further to $1/N$. In this work, $N = 10000$, which produces a very small error upper bound.

Appendix C

Analytical Propagation Formulas for Equinoctial Elements

These analytical formulas were published in the Journal of Celestial Mechanics and Dynamical Astronomy [19]. I include here only that which I developed, i.e. the third-body perturbations, as well as some of the explanation of how these formulas are derived.

As explained in Section 2.3, the motion of the spacecraft is described by non-singular equinoctial elements, in order to avoid singularities when the eccentricity or inclination of the orbit are zero. The set of equinoctial elements is [9]

$$\begin{aligned} &a, \\ &P_1 = e\sin\left(\Omega+\omega\right), \\ &P_2 = e\cos\left(\Omega+\omega\right), \\ &Q_1 = \tan\frac{i}{2}\sin\Omega, \\ &Q_2 = \tan\frac{i}{2}\cos\Omega, \\ &L, \end{aligned} \tag{C.1}$$

where $L = \Omega + \omega + \theta$ is the true longitude. The perturbing acceleration is expressed in a radial-transverse-normal reference frame (RTN).

Any perturbing acceleration to the Keplerian orbital motion (including low-thrust

propulsion actions) is, therefore, expressed in the RTN frame as:

$$\mathbf{f} = \begin{bmatrix} f_R \\ f_T \\ f_N \end{bmatrix} . \tag{C.2}$$

The Gauss' planetary equations, expressed in terms of equinoctial elements, are [4]:

$$\begin{aligned}
\frac{da}{dt} &= \frac{2}{B}\sqrt{\frac{a^3}{\mu}}\left[\left(P_2 \sin L - P_1 \cos L\right) f_R + \Phi(L) f_T\right], \\
\frac{dP_1}{dt} &= B\sqrt{\frac{a}{\mu}}\left[-f_R \cos L + \left(\frac{P_1 + \sin L}{\Phi(L)} + \sin L\right) f_T - P_2 \frac{Q_1 \cos L - Q_2 \sin L}{\Phi(L)} f_N\right], \\
\frac{dP_2}{dt} &= B\sqrt{\frac{a}{\mu}}\left[f_R \sin L + \left(\frac{P_2 + \cos L}{\Phi(L)} + \cos L\right) f_T + P_1 \frac{Q_1 \cos L - Q_2 \sin L}{\Phi(L)} f_N\right], \\
\frac{dQ_1}{dt} &= \frac{B}{2}\sqrt{\frac{a}{\mu}}\left(1 + Q_1^2 + Q_2^2\right)\frac{\sin L}{\Phi(L)} f_N, \\
\frac{dQ_2}{dt} &= \frac{B}{2}\sqrt{\frac{a}{\mu}}\left(1 + Q_1^2 + Q_2^2\right)\frac{\cos L}{\Phi(L)} f_N \ .
\end{aligned} \tag{C.3}$$

In Eq. (C.3), B and $\Phi(L)$ are defined as:

$$B = \sqrt{1 - e^2} = \sqrt{1 - P_1^2 - P_2^2}, \tag{C.4}$$

$$\Phi(L) = 1 + e\cos\theta = 1 + P_1 \sin L + P_2 \cos L \ . \tag{C.5}$$

The sixth Gauss' equation for dL/dt is [38]:

$$\frac{dL}{dt} = \sqrt{\frac{\mu}{a}}\frac{\Phi^2}{B^3} - \sqrt{\frac{a^3}{\mu}}\frac{B}{\Phi} f_N \left(Q_1 \cos L - Q_2 \sin L\right) \ . \tag{C.6}$$

Under the assumption that the perturbing acceleration is small compared to the local central body's Keplerian gravitational acceleration, the following approximation of Eq. (C.6) is obtained [74]:

$$\frac{dt}{dL} \approx \sqrt{\frac{a^3}{\mu}}\frac{B^3}{\Phi^2(L)} \ . \tag{C.7}$$

The validity of the approximation in Eq. (C.7) is justified in [19].

Combining Eqs. (**??**) and (C.7), the variations of the equinoctial elements with the true longitude can be expressed as:

$$\begin{aligned}
\frac{da}{dL} =& \frac{2a^3B^2}{\mu}\left[\frac{(P_2\sin L - P_1\cos L)}{\Phi^2(L)}f_R + \frac{1}{\Phi(L)}f_T\right], \\
\frac{dP_1}{dL} =& \frac{a^2B^4}{\mu}\left\{\left[-\frac{\cos L}{\Phi^2(L)}f_R + \left(\frac{P_1+\sin L}{\Phi^3(L)} + \frac{\sin L}{\Phi^2(L)}\right)f_T\right] - P_2\frac{Q_1\cos L - Q_2\sin L}{\Phi^3(L)}f_N\right\}, \\
\frac{dP_2}{dL} =& \frac{a^2B^4}{\mu}\left\{\left[\frac{\sin L}{\Phi^2(L)}f_R + \left(\frac{P_2+\cos L}{\Phi^3(L)} + \frac{\cos L}{\Phi^2(L)}\right)f_T\right] + P_1\frac{Q_1\cos L - Q_2\sin L}{\Phi^3(L)}f_N\right\}, \\
\frac{dQ_1}{dL} =& \frac{a^2B^4}{2\mu}\left(1+Q_1^2+Q_2^2\right)\frac{\sin L}{\Phi^3(L)}f_N, \\
\frac{dQ_2}{dL} =& \frac{a^2B^4}{2\mu}\left(1+Q_1^2+Q_2^2\right)\frac{\cos L}{\Phi^3(L)}f_N \ .
\end{aligned} \tag{C.8}$$

A first order analytical solution to Eq. (C.8), can be generated with the method of perturbations [65]. The idea at the basis of the method of perturbations is that small disturbing forces cause small deviations from the known solution to the unperturbed problem. The small perturbing forces can be associated with small parameters which characterise the magnitude of the disturbing forces. If one calls $\mathbf{X}$, the state of the spacecraft expressed in terms of equinoctial elements, $\mathbf{X} = [a, P_1, P_2, Q_1, Q_2, L]^T$, and the reduced vector with no true longitude $\widetilde{\mathbf{X}} = [a, P_1, P_2, Q_1, Q_2]^T$, Eqs. (C.8) can be rewritten in compact form as

$$\frac{d\widetilde{\mathbf{X}}}{dL} = \epsilon F\left(\widetilde{\mathbf{X}}, L\right) \ , \tag{C.9}$$

where $\epsilon = \|\mathbf{f}\|$. One can then look for the first-order approximation of the solution to Eq. (C.8) in the form

$$\widetilde{\mathbf{X}} \approx \widetilde{\mathbf{X}}_0 + \epsilon\widetilde{\mathbf{X}}_1\left(\widetilde{\mathbf{X}}_0, L_0, L, F\right) \ , \tag{C.10}$$

where

$$\widetilde{\mathbf{X}}_1 = \int_{L_0}^{L} F\left(\widetilde{\mathbf{X}}_0, \mathcal{L}\right) d\mathcal{L} \ , \tag{C.11}$$

and $\widetilde{\mathbf{X}}_0$ represents the vector of initial conditions $\widetilde{\mathbf{X}}_0 = [a_0, P_{10}, P_{20}, Q_{10}, Q_{20}]^T$. An

analytical solutions of Eq. (C.11) was previously presented in [73] and in [18] for some orbital perturbations and low-thrust profiles. In particular, in [73] it was shown that the analytical formulas in osculating elements can be used to efficiently propagate moderately long spirals composed of several tens of orbital revolutions. In this case, an analytical formula can be derived also for the variation of time with the true longitude L (see Eq. (C.7)). However, in this work the time equation is integrated with a quadrature method, as proposed in [73] because it does not increase the computational time and provide accurate results.

For longer spirals, an averaged propagation of the orbital elements is implemented; the variation of the equinoctial elements is, in this case, computed as

$$\bar{\mathbf{X}}(t) = \bar{\mathbf{X}}_0 + \int_{t_0}^{t} \dot{\bar{\mathbf{X}}}\left(\tau, \bar{\mathbf{X}}\left(\tau\right)\right) d\tau, \tag{C.12}$$

where

$$\dot{\bar{\mathbf{X}}} = \epsilon \bar{F}(\bar{\mathbf{X}}_0), \tag{C.13}$$

and

$$\bar{F}(\bar{\mathbf{X}}_0) = \frac{1}{2\pi} \int_0^{2\pi} F(\bar{\mathbf{X}}_0, \mathcal{L}) d\mathcal{L} \ . \tag{C.14}$$

We note that if one takes $\widetilde{\mathbf{X}}_0 = \bar{\mathbf{X}}_0$ then integral (C.11) can be used to compute integral (C.14) analytically. According to Verhulst [69] (see page 161 and following) the approximation is $O(\epsilon^2)$ and remains small on a time scale that is proportional to $1/\epsilon$.

While the integrals in Eq. (C.11) are computed analytically, the time integral in Eq. (C.12) is computed numerically; the resulting averaged propagator is, therefore, a semi-analytical method. In the remainder of this work, the focus is on the averaged formulation only. By using this approach the authors have derived analytical formulae for the following accelerations and orbital perturbations:

1. second zonal harmonic of the Earth's gravitational perturbation, J_2 (as in [73]);

2. third, fourth and fifth zonal harmonic of the Earth's gravitational perturbation,

J_3, J_4, J_5 (as in [19]);

3. atmospheric drag (as in [18]);
4. solar radiation pressure, including eclipses (as in [73]);
5. third-body gravitational perturbation (Section C.1);
6. constant tangential acceleration (as in [73]);
7. constant acceleration in a radial-transverse-normal reference frame (as in [73]);
8. acceleration with constant direction in a radial-transverse-normal reference frame, and with magnitude of the acceleration proportional to $1/r_{\odot}^2$, where $r_{\odot}$ is the distance from the central body (Sun) (as in [19]);
9. constant acceleration in an inertial reference frame (as in [73]).

My contribution, which I include in this thesis, were the formulae for third body perturbations.

C.1 Third-Body Perturbations

Similarly to the aspherical gravitational potential, the third-body potential is commonly written as a sum of Legendre polynomials written using the Legendre polynomials $\mathcal{P}_n$ (Equation C.16) [65]. Our work here is similar to that of Cefola [12], except instead of using the Lagrange VoP equations, we obtain the corresponding accelerations and introduce them in the Gauss VOP equations. Furthermore, instead of just obtaining averaged solutions, we also obtain osculating solutions.

In this section n refers to the order of the polynomial. The quantities r and $\hat{\mathbf{r}}$ refer to the norm and normalized vector of the position of the satellite relative to the Earth. The third-body disturbing potential can be written as

$$R_{3^{\text{rd}}} = \sum_{n=2}^{\infty} R_{\mathcal{P}_n} \ , \tag{C.15}$$

where $R_{\mathcal{P}_n}$ refers to the disturbing third-body potential of Legendre polynomial order n, and is given by:

$$R_{\mathcal{P}_n} = -\frac{\mu_3}{R_3} \sum_n \left(\frac{r}{R_3}\right)^n \mathcal{P}_n(\hat{\mathbf{r}} \cdot \hat{\mathbf{R}}_3) \ . \tag{C.16}$$

In Eq. (C.16), R_3 and $\hat{\mathbf{R}}_3$ are the norm and normalized vector of the position of the third-body relative to Earth. We can now obtain $f_{\mathcal{P}_n}$, the acceleration vector caused by this perturbation, by taking its gradient:

$$\mathbf{f}_{\mathcal{P}_n} = \frac{\mu_3}{R_3} \left(\frac{r}{R_3}\right)^n \frac{1}{r} \left(n\hat{\mathbf{r}}\mathcal{P}_n(\hat{\mathbf{r}} \cdot \hat{\mathbf{R}}_3) + (\hat{\mathbf{R}}_3 - \hat{\mathbf{r}}(\hat{\mathbf{R}}_3 \cdot \hat{\mathbf{r}}))\mathcal{P}'_n(\hat{\mathbf{r}} \cdot \hat{\mathbf{R}}_3) \right) \ , \tag{C.17}$$

where $\mathcal{P}'_n$ is the derivative of the Legendre polynomial. The exact third-body acceleration is $\mathbf{f}_{3^{\text{rd}}} = \sum_{n=2}^{\infty} \mathbf{f}_{\mathcal{P}_n}$. We approximate to order $n = 5$.

To obtain these accelerations in the RTN frame, they are first calculated in the equinoctial frame and then rotated. Therefore, in the following equations, the vector $\hat{\mathbf{r}}$ will be expressed in the equinoctial reference frame. The equinoctial reference frame is defined as having the x axis pointing towards the satellite when $L = 0$, the z direction along the direction of the angular momentum, and y, in order to complete the right-handed system, pointing towards the satellite when $L = \pi/2$. In this frame, the quantity $\hat{\mathbf{r}}$ in Eq. (C.17) is

$$\hat{\mathbf{r}} = [\cos L, \sin L, 0]^T \ , \tag{C.18}$$

and we use the same direction cosines as in Cefola [12]:

$$\begin{aligned} \hat{\mathbf{R}}_3 &= [\alpha_{3rd}, \beta_{3rd}, \gamma_{3rd}]^T \ , \\ \hat{\mathbf{r}} \cdot \hat{\mathbf{R}}_3 &= \alpha_{3rd} \cos L + \beta_{3rd} \sin L \ , \\ (\hat{\mathbf{R}}_3 - \hat{\mathbf{r}}(\hat{\mathbf{R}}_3 \cdot \hat{\mathbf{r}})) &= \begin{bmatrix} -\beta_{3rd} \cos L \sin L + \alpha_{3rd} \sin^2 L \\ -\alpha_{3rd} \cos L \sin L + \beta_{3rd} \cos^2 L \\ \gamma_{3rd} \end{bmatrix} \ . \end{aligned} \tag{C.19}$$

Introducing the equations above into the formula for the acceleration will give us the

acceleration in the equinoctial reference frame, which can be converted to RTN using

$$\mathbf{f}_{3^{\text{rd}}}^{RTN} = R_z[-L]\mathbf{f}_{3^{\text{rd}}}^{\text{equin}} \ , \tag{C.20}$$

where $R_z[-L]$ is the rotation matrix that rotates about the z axis of an angle L. Defining the function F as

$$F_l^{(n)}(L; c_{mn}, s_{mn}) = \frac{1}{\Phi^{n+l}(L)} \sum_m c_{mn} \cos mL + s_{mn} \sin mL \ , \tag{C.21}$$

the acceleration can be written as:

$$\begin{aligned} f_{3^{rd}}^R &= \sum_{n=2}^{5} \frac{\mu_3 (aB^2)^{n-1}}{R_3^{n+1}} F_{-1}^{(n)}(L; c_{mn}^R, s_{mn}^R) \ , \\ f_{3^{rd}}^T &= \sum_{n=2}^{5} \frac{\mu_3 (aB^2)^{n-1}}{R_3^{n+1}} F_{-1}^{(n)}(L; c_{mn}^T, s_{mn}^T) \ , \\ f_{3^{rd}}^N &= \sum_{n=2}^{5} \frac{\mu_3 (aB^2)^{n-1}}{R_3^{n+1}} F_{-1}^{(n)}(L; c_{mn}^N, s_{mn}^N) \ . \end{aligned} \tag{C.22}$$

The values of the coefficients c and s are presented in the Section C.2. These formulas were obtained using Wolfram Mathematica. Using the following properties of the function F,

$$\begin{aligned} \Phi^k F_l(L; c_{mn}, s_{mn}) &= F_{l-k}(L; c_{mn}, s_{mn}) \ , \\ \cos L \ F_l(L; c_{mn}, s_{mn}) &= \frac{1}{2} F_l(L; c_{m-1;n} + c_{m+1;n}, s_{m-1;n} + s_{m+1;n}) \ , \\ \sin L \ F_l(L; c_{mn}, s_{mn}) &= \frac{1}{2} F_l(L; s_{m+1;n} - s_{m-1;n}, c_{m-1;n} - c_{m+1;n}) \quad , \end{aligned} \tag{C.23}$$

we can insert the formula for the acceleration in the Gauss equations, multiply by dt/dL

(Eq. (C.7)), and obtain the following equations:

$$
\begin{aligned}
\frac{da}{dL} &= 2\epsilon_{3^{rd}} a \sum_{n=2}^{5} B^{2n} \frac{a^{n+1}}{a_0^2 R_3^{n-1}} F_1^{(n)}(L; c_{mn}^{(1)}, s_{mn}^{(1)}) \ , \\
\frac{dP_1}{dL} &= \epsilon_{3^{rd}} \sum_{n=2}^{5} B^{2n+4} \frac{a^{n+1}}{a_0^2 R_3^{n-1}} \left(F_2^{(n)}(L; c_{mn}^{(2)}, s_{mn}^{(2)}) + F_1^{(n)}(L; c_{mn}^{(3)}, s_{mn}^{(3)}) \right) \ , \\
\frac{dP_2}{dL} &= \epsilon_{3^{rd}} \sum_{n=2}^{5} B^{2n+4} \frac{a^{n+1}}{a_0^2 R_3^{n-1}} \left(F_2^{(n)}(L; c_{mn}^{(4)}, s_{mn}^{(4)}) + F_1^{(n)}(L; c_{mn}^{(5)}, s_{mn}^{(5)}) \right) \ , \\
\frac{dQ_1}{dL} &= \frac{1}{4} \epsilon_{3^{rd}} S \sum_{n=2}^{5} B^{2n+4} \frac{a^{n+1}}{a_0^2 R_3^{n-1}} F_2^{(n)}(L; s_{m+1,n}^N - s_{m-1,n}^N, c_{m+1,n}^N + c_{m-1,n}^N) \ , \\
\frac{dQ_2}{dL} &= \frac{1}{4} \epsilon_{3^{rd}} S \sum_{n=2}^{5} B^{2n+4} \frac{a^{n+1}}{a_0^2 R_3^{n-1}} F_2^{(n)}(L; c_{m+1,n}^N + c_{m-1,n}^N, -s_{m+1,n}^N + s_{m-1,n}^N) \ ,
\end{aligned}
\tag{C.24}
$$

where $\epsilon_{3^{rd}}$ is given by

$$
\epsilon_{3^{rd}} = \frac{\mu_3}{\mu} \frac{a_0^2}{R_3^2} \ , \tag{C.25}
$$

and the coefficients are defined as, using c_m as short-hand for c_{mn}:

$$
\begin{aligned}
c_m^{(1)} &= \frac{1}{2}\left(P_1(-c_{m+1}^R - c_{m-1}^R - s_{m-1}^T + s_{m+1}^T)\right. \\
&\quad \left. +P_2(s_{m+1}^R - s_{m-1}^R + c_{m-1}^T + c_{m+1}^T)\right) + c_m^T \ , \\
s_m^{(1)} &= \frac{1}{2}\left(P_1(s_{m+1}^R - s_{m-1}^R + c_{m-1}^T - c_{m+1}^T)\right. \\
&\quad \left. P_2(-c_{m+1}^R + c_{m-1}^R + s_{m-1}^T + s_{m+1}^T)\right) + s_m^T \ , \\
c_m^{(2)} &= P_1 c_m^T + \frac{1}{2}\left(s_{m+1}^T - s_{m-1}^T - P_2(+(c_{m+1}^N + c_{m-1}^N)Q_1 - (s_{m+1}^N - s_{m-1}^N)Q_2)\right) \ , \\
s_m^{(2)} &= P_1 s_m^T + \frac{1}{2}\left(c_{m+1}^T + c_{m-1}^T - P_2(-(c_{m+1}^N + c_{m-1}^N)Q_2 - (s_{m+1}^N - s_{m-1}^N)Q_1)\right) \ , \\
c_m^{(3)} &= \frac{1}{2}\left(-c_{m+1}^R - c_{m-1}^R + s_{m+1}^T - s_{m-1}^T\right) \ , \\
s_m^{(3)} &= \frac{1}{2}\left(+s_{m+1}^R - s_{m-1}^R + c_{m+1}^T + c_{m-1}^T\right) \ , \\
c_m^{(4)} &= P_2 c_m^T + \frac{1}{2}\left(c_{m+1}^T + c_{m-1}^T + P_1(+(c_{m+1}^N + c_{m-1}^N)Q_1 - (s_{m+1}^N - s_{m-1}^N)Q_2)\right) \ , \\
s_m^{(4)} &= P_2 s_m^T + \frac{1}{2}\left(-s_{m+1}^T + s_{m-1}^T + P_1(-(c_{m+1}^N + c_{m-1}^N)Q_2 - (s_{m+1}^N - s_{m-1}^N)Q_1)\right) \ , \\
c_m^{(5)} &= \frac{1}{2}\left(s_{m+1}^R - s_{m-1}^R + c_{m+1}^T + c_{m-1}^T\right) \ , \\
s_m^{(5)} &= \frac{1}{2}\left(c_{m+1}^R + c_{m-1}^R - s_{m+1}^T + s_{m-1}^T\right) \ .
\end{aligned}
\tag{C.26}
$$

Because the third-body moves along its orbit, the coefficients c and s, which depend solely on the direction cosines, vary as well as the value of R_3. In order to capture this effect one would have to express the motion of the third-body as a function of L. Alternatively, one can split the calculation of integral (C.11) into segments and change the position of the third-body on each segment. In this work, however, we use the simplifying assumption that the third-body moves slowly with respect to the orbital motion of the spacecraft. Thus the averaging integral is computed keeping the position of the third-body constant. By applying integral (C.11) to the right hand side of Eqs.

(C.24), one obtains the system

$$\widetilde{\mathbf{X}}_1^{3rd} = \begin{bmatrix} 2a_0 \sum_{n=2}^{5} B_0^{2n} \left(\frac{a_0}{R_3}\right)^{n-1} G_1^{(n)}(L; c_{mn}^{(1)}, s_{mn}^{(1)}) \\ \sum_{n=2}^{5} B_0^{2n+4} \left(\frac{a_0}{R_3}\right)^{n-1} \left(G_2^{(n)}(L; c_{mn}^{(2)}, s_{mn}^{(2)}) + G_1^{(n)}(L; c_{mn}^{(3)}, s_{mn}^{(3)})\right) \\ \sum_{n=2}^{5} B_0^{2n+4} \left(\frac{a_0}{R_3}\right)^{n-1} \left(G_2^{(n)}(L; c_{mn}^{(4)}, s_{mn}^{(4)}) + G_1^{(n)}(L; c_{mn}^{(5)}, s_{mn}^{(5)})\right) \\ \frac{1}{4} S_0 \sum_{n=2}^{5} B_0^{2n+4} \left(\frac{a_0}{R_3}\right)^{n-1} G_2^{(n)}(L; s_{m+1}^N - s_{m-1}^N, c_{m+1}^N + c_{m-1}^N) \\ \frac{1}{4} S_0 \sum_{n=2}^{5} B_0^{2n+4} \left(\frac{a_0}{R_3}\right)^{n-1} G_2^{(n)}(L; c_{m+1}^N + c_{m-1}^N, -s_{m+1}^N + s_{m-1}^N) \end{bmatrix} , \quad \text{(C.27)}$$

where

$$G_l^{(n)}(L; c_{mn}, s_{mn}) = \sum_m c_{mn} I_{m,n+l}^c + s_{mn} I_{m,n+l}^s \ . \quad \text{(C.28)}$$

The integrals $I_{m,n+l}^c$ and $I_{m,n+l}^s$ are defined as

$$\begin{aligned} I_{m,n+l}^c &= \int_{L_0}^{L} \frac{\cos m\mathcal{L}}{\Phi^{n+l}(\mathcal{L})} d\mathcal{L} \ , \\ I_{m,n+l}^s &= \int_{L_0}^{L} \frac{\sin m\mathcal{L}}{\Phi^{n+l}(\mathcal{L})} d\mathcal{L} \ , \end{aligned} \quad \text{(C.29)}$$

and their formulas are presented in Section C.2.1. The third-body perturbation is obtained by using $\widetilde{\mathbf{X}}_1^{3rd}$ and ϵ_{3rd} in Eq. (C.10).

C.2 Coefficients for acceleration

Equations (C.30-C.38) contain the values of c_{nm}^{RTN} and s_{nm}^{RTN}. These coefficients are presented as matrices, where each matrix presents all the non-zero (and some zero) values of c_{nm}^{RTN} or s_{nm}^{RTN} for a fixed n. Each row corresponds to a fixed value of m, starting from $m = 0$ for the top row. The columns are the radial (R), Transverse (T) and Normal (N) components, in this order. Each expression, and its LaTeX representation, was obtained using Mathematica. Missing coefficients are equal to zero. The direction cosines $\alpha_{3rd}, \beta_{3rd}, \gamma_{3rd}$ will be abbreviated α, β, γ.

Notice that $c_{mn}^R = c_{mn}^T = 0$ for any values of m, n where $m > n$ or their sum $m + n$ is odd. Similarly, $c_{mn}^N = 0$ for any values of m, n where $m > n - 1$ or their sum $m + n$

is even:

$$\begin{aligned} c_{mn}^{R} = c_{mn}^{T} = 0 \ \forall m,n \in \mathbb{N} : (m > n) \vee [(m+n) \text{ odd}] \\ c_{mn}^{N} = 0 \ \forall m,n \in \mathbb{N} : [m > (n-1)] \vee [(m+n) \text{ even}] \end{aligned} \tag{C.30}$$

$$c_2^{RTN} = \begin{bmatrix} \frac{1}{2}\left(3\alpha^2 + 3\beta^2 - 2\right) & 0 & 0 \\ 0 & 0 & 3\alpha\gamma \\ \frac{1}{2}\left(3\alpha^2 - 3\beta^2\right) & 3\alpha\beta & 0 \end{bmatrix} \tag{C.31}$$

$$s_2^{RTN} = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 3\beta\gamma \\ 3\alpha\beta & -\frac{3}{2}\left(\alpha^2 - \beta^2\right) & 0 \end{bmatrix} \tag{C.32}$$

$$c_3^{RTN} = \begin{bmatrix} 0 & 0 & \frac{3}{4}\gamma\left(5\alpha^2 + 5\beta^2 - 2\right) \\ \frac{9}{8}\alpha\left(5\alpha^2 + 5\beta^2 - 4\right) & \frac{3}{8}\beta\left(5\alpha^2 + 5\beta^2 - 4\right) & 0 \\ 0 & 0 & \frac{15}{4}\gamma(\alpha-\beta)(\alpha+\beta) \\ \frac{15}{8}\left(\alpha^3 - 3\alpha\beta^2\right) & \frac{15}{8}\left(3\alpha^2\beta - \beta^3\right) & 0 \end{bmatrix} \tag{C.33}$$

$$s_3^{RTN} = \begin{bmatrix} 0 & 0 & 0 \\ \frac{9}{8}\beta\left(5\alpha^2 + 5\beta^2 - 4\right) & -\frac{3}{8}\alpha\left(5\alpha^2 + 5\beta^2 - 4\right) & 0 \\ 0 & 0 & \frac{15\alpha\beta\gamma}{2} \\ \frac{15}{8}\left(3\alpha^2\beta - \beta^3\right) & -\frac{15}{8}\left(\alpha^3 - 3\alpha\beta^2\right) & 0 \end{bmatrix} \tag{C.34}$$

$$c_4^{RTN} = \begin{bmatrix} \frac{3}{16}\left(35\alpha^4 + 10\alpha^2\left(7\beta^2-4\right) + 35\beta^4 - 40\beta^2 + 8\right) & 0 & 0 \\ 0 & 0 & \frac{15}{8}\alpha\gamma\left(7\alpha^2 + 7\beta^2 - 4\right) \\ \frac{5}{4}(\alpha-\beta)(\alpha+\beta)\left(7\alpha^2+7\beta^2-6\right) & \frac{5}{4}\alpha\beta\left(7\alpha^2+7\beta^2-6\right) & 0 \\ 0 & 0 & \frac{35}{8}\alpha\gamma\left(\alpha^2-3\beta^2\right) \\ \frac{35}{16}\left(\alpha^4 - 6\alpha^2\beta^2 + \beta^4\right) & \frac{35}{4}\alpha\beta(\alpha-\beta)(\alpha+\beta) & 0 \end{bmatrix} \tag{C.35}$$

$$s_4^{RTN} = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & \frac{15}{8}\beta\gamma\left(7\alpha^2+7\beta^2-4\right) \\ \frac{5}{2}\alpha\beta\left(7\alpha^2+7\beta^2-6\right) & -\frac{5}{8}(\alpha-\beta)(\alpha+\beta)\left(7\alpha^2+7\beta^2-6\right) & 0 \\ 0 & 0 & -\frac{35}{8}\beta\gamma\left(\beta^2-3\alpha^2\right) \\ \frac{35}{4}\alpha\beta(\alpha-\beta)(\alpha+\beta) & -\frac{35}{16}\left(\alpha^4 - 6\alpha^2\beta^2 + \beta^4\right) & 0 \end{bmatrix} \tag{C.36}$$

$$c_5^{RTN} = \frac{15}{128}\begin{bmatrix} 0 & 0 & 2\gamma A_2 \\ 10\alpha A_1 & 2\beta A_1 & 0 \\ 0 & 0 & 56\gamma(\alpha^2-\beta^2)\left(3\alpha^2+3\beta^2-2\right) \\ \frac{35}{3}\alpha\left(\alpha^2-3\beta^2\right)\left(9\alpha^2+9\beta^2-8\right) & -7\beta\left(\beta^2-3\alpha^2\right)\left(9\alpha^2+9\beta^2-8\right) & 0 \\ 0 & 0 & 42\gamma\left(\alpha^4-6\alpha^2\beta^2+\beta^4\right) \\ 21\left(\alpha^5-10\alpha^3\beta^2+5\alpha\beta^4\right) & 21\left(5\alpha^4\beta-10\alpha^2\beta^3+\beta^5\right) & 0 \end{bmatrix} \tag{C.37}$$

$$s_5^{RTN} = \frac{15}{128}\begin{bmatrix} 0 & 0 & 0 \\ 10\beta A_1 & -2\alpha A_1 & 0 \\ 0 & 0 & 112\alpha\beta\gamma\left(3\alpha^2+3\beta^2-2\right) \\ -\frac{35}{3}\beta\left(\beta^2-3\alpha^2\right)\left(9\alpha^2+9\beta^2-8\right) & -7\alpha\left(\alpha^2-3\beta^2\right)\left(9\alpha^2+9\beta^2-8\right) & 0 \\ 0 & 0 & 168\alpha\beta\gamma(\alpha-\beta)(\alpha+\beta) \\ 21\left(5\alpha^4\beta-10\alpha^2\beta^3+\beta^5\right) & -21\left(\alpha^5-10\alpha^3\beta^2+5\alpha\beta^4\right) & 0 \end{bmatrix} \tag{C.38}$$

The following replacements were used to shorten the expressions:

$$A_1 = \left(7\left(3\alpha^4 + \alpha^2\left(6\beta^2 - 4\right) + 3\beta^4\right) - 28\beta^2 + 8\right) \tag{C.39}$$

$$A_2 = \left(63\alpha^4 + 14\alpha^2\left(9\beta^2 - 4\right) + 63\beta^4 - 56\beta^2 + 8\right) \tag{C.40}$$

C.2.1 Integrals

The expressions for the average integrals, I^c_{mn} and I^s_{mn}, which were obtained using Mathematica, have a secular term and a short-periodic term. The short-periodic term is too large to be presented here, but the secular term is much more manageable. In Tables C.1 and C.2 we have the averaged integrals $\bar{I}^c_{mn}$ and $\bar{I}^s_{mn}$ defined in Eq. (C.41). The secular components, $\mathcal{S}^c_{m,n}$ and $\mathcal{S}^s_{m,n}$, can be expressed using the averaged integrals according to Eq. (C.42). The complete results are collected in a Mathematica notebook that can be found at: `https://pureportal.strath.ac.uk/en/datasets/analytical-solutions-for-j3-j4-j5-and-third-body-perturbations-fo`

In this section, and in the notebook, P_1, P_2, e and B are used as short-hand for P_{10}, P_{20}, e_0 and B_0.

$$\begin{aligned} \bar{I}^c_{m,n} &= \frac{1}{2\pi}\int_{L_0}^{L_0+2\pi} \frac{\cos m\mathcal{L}}{\Phi^n(\mathcal{L})} d\mathcal{L} \\ \bar{I}^s_{m,n} &= \frac{1}{2\pi}\int_{L_0}^{L_0+2\pi} \frac{\sin m\mathcal{L}}{\Phi^n(\mathcal{L})} d\mathcal{L} \end{aligned} \tag{C.41}$$

$$\begin{aligned} \mathcal{S}^c_{m,n} &= 2\bar{I}^c_{m,n}\left[\Lambda(\mathcal{L})\right]^L_{L_0} \\ \mathcal{S}^s_{m,n} &= 2\bar{I}^s_{m,n}\left[\Lambda(\mathcal{L})\right]^L_{L_0} \end{aligned} \tag{C.42}$$

where $\Lambda(\mathcal{L})$ is defined as

$$\Lambda(\mathcal{L}) = \arctan\left(\frac{P_1 + (1 - P_2)\tan\left(\frac{\mathcal{L}}{2}\right)}{B}\right) \tag{C.43}$$

Table C.1: Values of $\bar{I}^c_{m,n}$. An auxiliary variable is used, $V = P_1^2 - P_2^2$. These expressions and their LaTeX representation were obtained using Mathematica.

m \ n	1	2	3	4	5
0	$\frac{e^2+2}{2B^5}$	$\frac{3e^2+2}{2B^7}$	$\frac{3e^4+24e^2+8}{8B^9}$	$\frac{15e^4+40e^2+8}{8B^{11}}$	$\frac{5e^6+90e^4+120e^2+16}{16B^{13}}$
1	$-\frac{3P_2}{2B^5}$	$-\frac{\left(e^2+4\right)P_2}{2B^7}$	$-\frac{5\left(3e^2+4\right)P_2}{8B^9}$	$-\frac{3\left(e^4+12e^2+8\right)P_2}{8B^{11}}$	$-\frac{7\left(5e^4+20e^2+8\right)P_2}{16B^{13}}$
2	$-\frac{3V}{2B^5}$	$-\frac{5V}{2B^7}$	$-\frac{5\left(e^2+6\right)V}{8B^9}$	$-\frac{21\left(e^2+2\right)V}{8B^{11}}$	$-\frac{7\left(e^4+16e^2+16\right)V}{16B^{13}}$
3		$\frac{5P_2\left(3P_1^2-P_2^2\right)}{2B^7}$	$\frac{35P_2\left(3P_1^2-P_2^2\right)}{8B^9}$	$\frac{7\left(e^2+8\right)P_2\left(3P_1^2-P_2^2\right)}{8B^{11}}$	$\frac{21\left(3e^2+8\right)P_2\left(3P_1^2-P_2^2\right)}{16B^{13}}$
4			$\frac{35\left(e^4-8P_1^2P_2^2\right)}{8B^9}$	$\frac{63\left(e^4-8P_1^2P_2^2\right)}{8B^{11}}$	$\frac{21\left(e^2+10\right)\left(e^4-8P_1^2P_2^2\right)}{16B^{13}}$
5				$\frac{63P_2\left(4P_2^4-5V^2\right)}{8B^{11}}$	$\frac{231P_2\left(4P_2^4-5V^2\right)}{16B^{13}}$
6					$-\frac{231V\left(e^4-16P_1^2P_2^2\right)}{16B^{13}}$

Finally, note that the average variation for the semi-major axis a is zero for each Legendre polynomial, that is,

$$G_1^{(n)}(L_0 + 2\pi; c_{mn}^{(1)}, s_{mn}^{(1)}) - G_1^{(n)}(L_0; c_{mn}^{(1)}, s_{mn}^{(1)}) = 0 \ \forall n \in \mathbb{N} \tag{C.44}$$

Table C.2: Values of $\bar{I}^s_{m,n}$. An auxiliary variable is used, $V = P_1^2 - P_2^2$. These expressions and their LaTeX representation were obtained using Mathematica.

m \ n	1	2	3	4	5
1	$-\frac{3P_1}{2B^5}$	$-\frac{\left(e^2+4\right)P_1}{2B^7}$	$-\frac{5\left(3e^2+4\right)P_1}{8B^9}$	$-\frac{3\left(e^4+12e^2+8\right)P_1}{8B^{11}}$	$-\frac{7\left(5e^4+20e^2+8\right)P_1}{16B^{13}}$
2	$\frac{3P_1P_2}{B^5}$	$\frac{5P_1P_2}{B^7}$	$\frac{5\left(e^2+6\right)P_1P_2}{4B^9}$	$\frac{21\left(e^2+2\right)P_1P_2}{4B^{11}}$	$\frac{7\left(e^4+16e^2+16\right)P_1P_2}{8B^{13}}$
3		$\frac{5P_1\left(4P_1^2-3e^2\right)}{2B^7}$	$\frac{35P_1\left(4P_1^2-3e^2\right)}{8B^9}$	$-\frac{7\left(e^2+8\right)P_1\left(3e^2-4P_1^2\right)}{8B^{11}}$	$-\frac{21\left(3e^2+8\right)P_1\left(3e^2-4P_1^2\right)}{16B^{13}}$
4			$-\frac{35P_1P_2V}{2B^9}$	$-\frac{63P_1P_2V}{2B^{11}}$	$-\frac{21\left(e^2+10\right)P_1P_2V}{4B^{13}}$
5				$\frac{63P_1\left(4P_1^4-5V^2\right)}{8B^{11}}$	$\frac{231P_1\left(4P_1^4-5V^2\right)}{16B^{13}}$
6					$\frac{231P_1P_2\left(3e^4-16P_1^2P_2^2\right)}{8B^{13}}$

Appendix D

Details on the Resulting Trajectories

This appendix contains details on the trajectories obtained as solutions to the robust many-objective optimal control problem that are shown in Section 5.7. This consists mainly of plots of the trajectories and of the corresponding control laws.

D.1 Case A1

As an illustration, the solution in the third row of Table 5.11 was plotted in Fig. D.1, since it corresponds to a case where the lower expectation was 1 for all QoIs.. The control laws corresponding to the elements in Table 5.11 are represented in Figs. D.2 and D.3. Figure D.2 shows the thrust profile, with blue segments representing thrust arcs, and white segments representing coast arcs. A significant part of the trajectory is spent with the thrusters active, mostly away from the pericentre. While the launch vehicle places the spacecraft in a trajectory that intersects the target orbit, this orbit has a similar apoapsis but a higher periapsis. To reduce the relative velocity at intersection, requires, in part, that the shapes of the orbits match, which in this case requires lowering the periapsis to match that of the target's. This is done most efficiently around the apoapsis. Since the engine is low thrust, and there are other necessary adjustments, the thrust is spread over a significant part of the orbit around the apoapsis. The angles

of the thrust vector are represented in Fig. D.3 for solutions 3, 6 and 9. The difference in thrust length for these three solutions is less than one radian, and the difference in thrust azimuth is below 10 degrees. To better understand the meaning of the angles α and β, refer back to Fig. 2.2. The azimuth α is near a value of -90 degrees, while the declination β is between -5 and 15 degrees, which corresponds to a nearly in-plane manoeuvre with a thrust direction close to the opposite of the velocity vector, which is expected since the target orbit has a lower semi-major axis and is close to being coplanar with the departure orbit, and such a manoeuvre will lower the semi-major axis of the space-craft's orbit without significantly altering its orbital plane. Figure D.4 shows the fourth solution in Table 5.11, with the magnitude of the associated displacements $\mathbf{D}_r$ and $\mathbf{D}_v$ in the vertical black lines, and the associated dispersion of the values of the quantities of interest when the uncertain variables follow a uniform distribution or the distribution that minimises the lower expectation. The lower expectation distribution is an extreme case that accounts for the lack of knowledge on the uncertain quantities. As such it gives an upper limit on the expected propellant mass and violation of the constraints during the operational phase even when the solution provided by MACS is close to satisfying the terminal constraints.

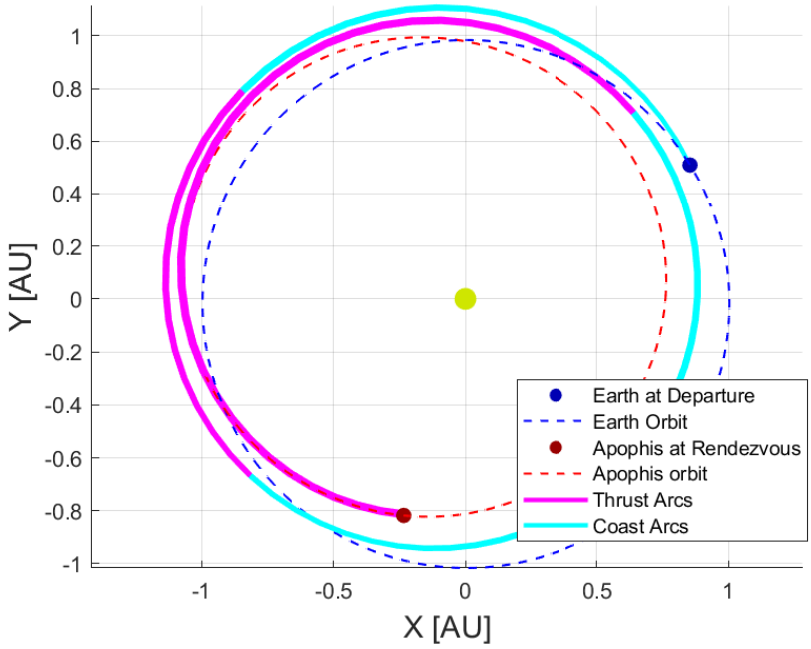


Figure D.1: Plot of trajectories for the control law corresponding to the third solution in Table 5.11 and for 200 different samples of $\boldsymbol{\xi}$. The plot is seen perpendicularly to the ecliptic plane and the axis are in AU.

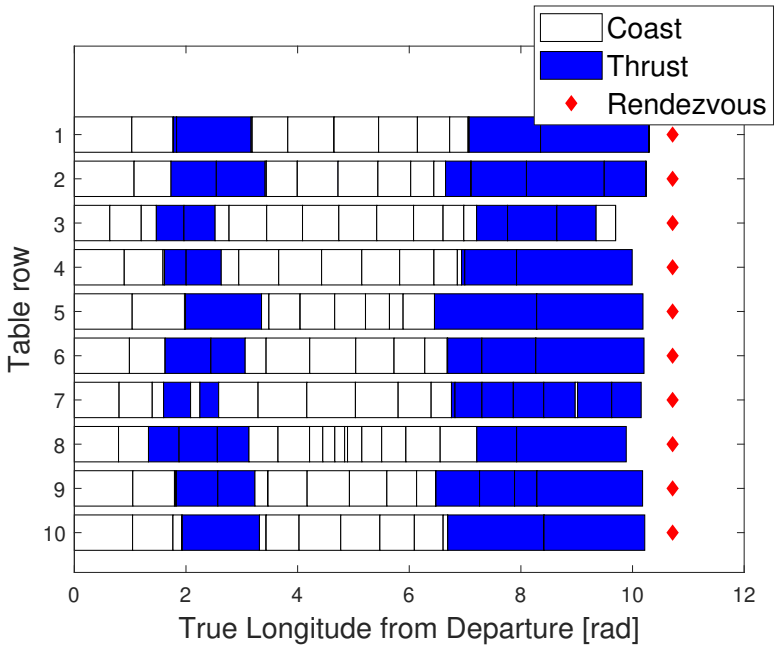


Figure D.2: Plot of the thrust profile as a function of the true longitude relative to departure for the solutions in Table 5.11. The red diamond is the true longitude where Apophis is encountered (in the nominal case).

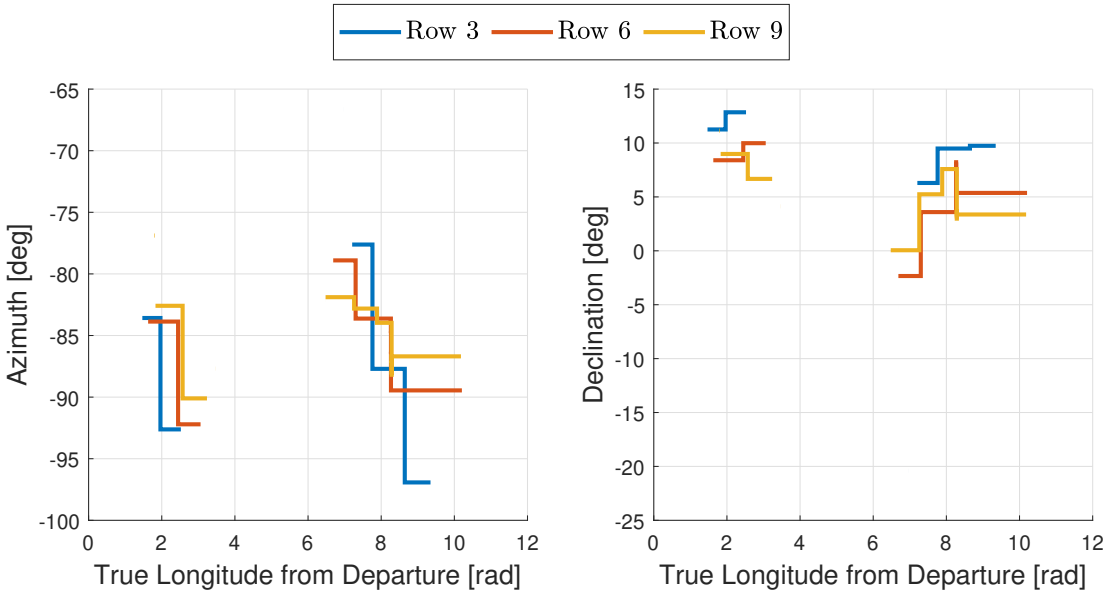


Figure D.3: Plots of the thrust vector azimuth and declination as a function of the true longitude relative to departure for 3 of the solutions solutions in Table 5.11.

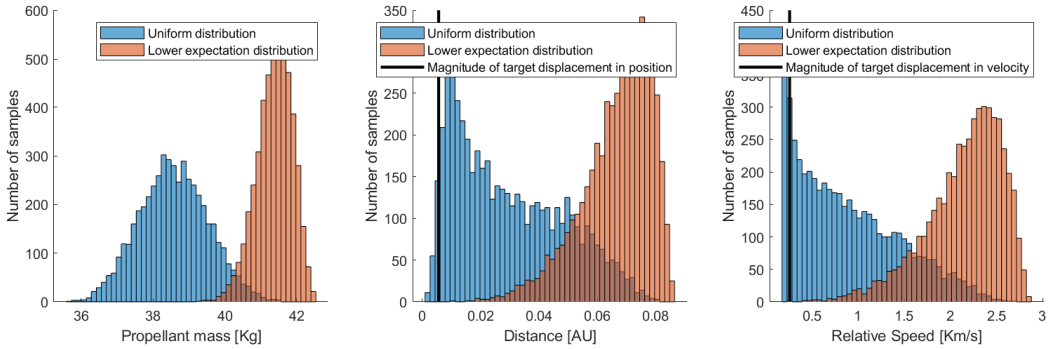


Figure D.4: Histograms of the QoIs that are obtained for the fourth solution in Table 5.11 for test case 1. The blue and red histograms are the result of assuming a uniform distribution, and the lower expectation distribution, respectively, in Ξ.

D.2 Case A2

The control laws for the selected results shown in Table 5.12 are represented in Figs D.6 and D.7 for completeness. The third solution in Table 5.12 was plotted in Fig. D.5, as it was the solution closest to having all lower expectations equal to one. The angles α and β show similar behaviour as for test case A1, with thrust mostly in-plane and pointing opposite to the direction of the velocity.

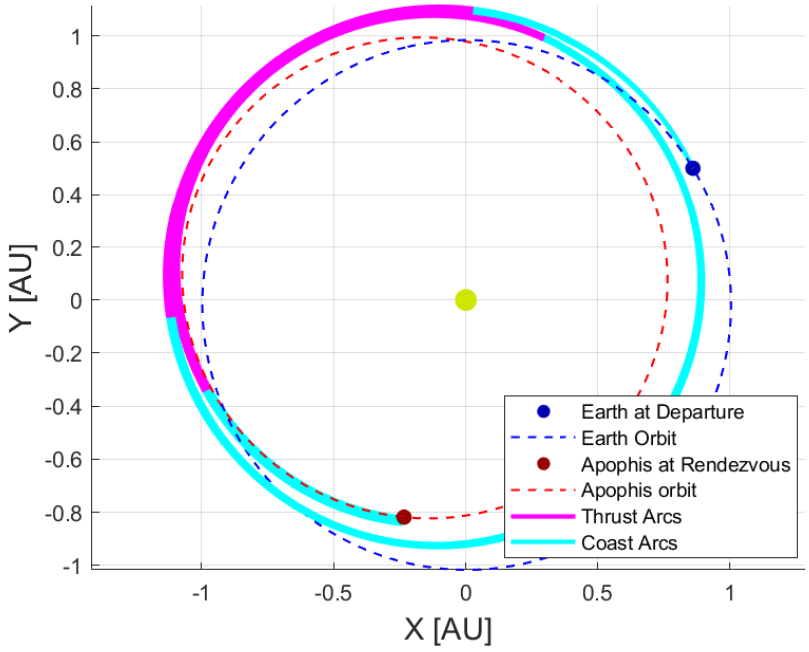


Figure D.5: Plot of trajectories for the control law corresponding to the third solution in Table 5.12 and for 200 different samples of $\boldsymbol{\xi}$. The plot is seen perpendicularly to the ecliptic plane and the axis are in AU.

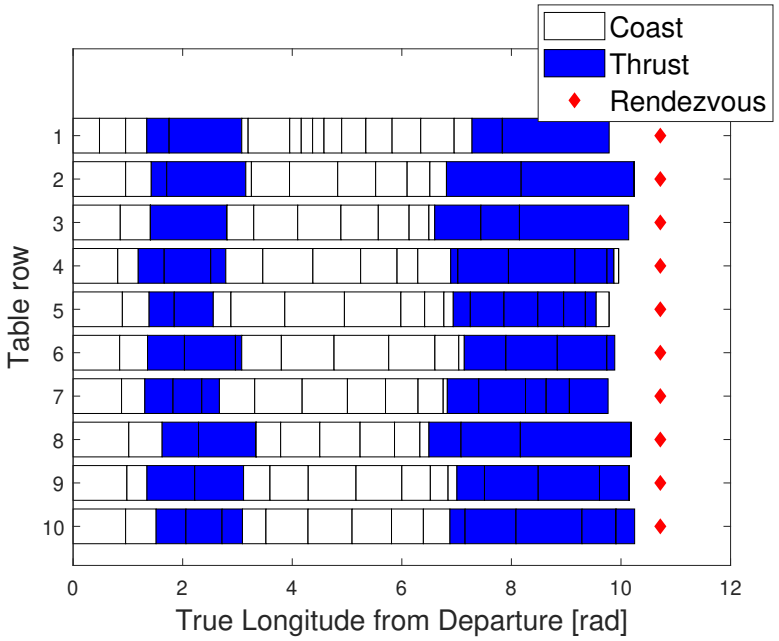


Figure D.6: Plot of the thrust profile as a function of the true longitude relative to departure for the solutions in Table 5.12. Blue segments represent thrust arcs, and white segments represent coast arcs. Note the presence of many zero-length arcs. The red diamond is the true longitude where Apophis is encountered (in the nominal case).

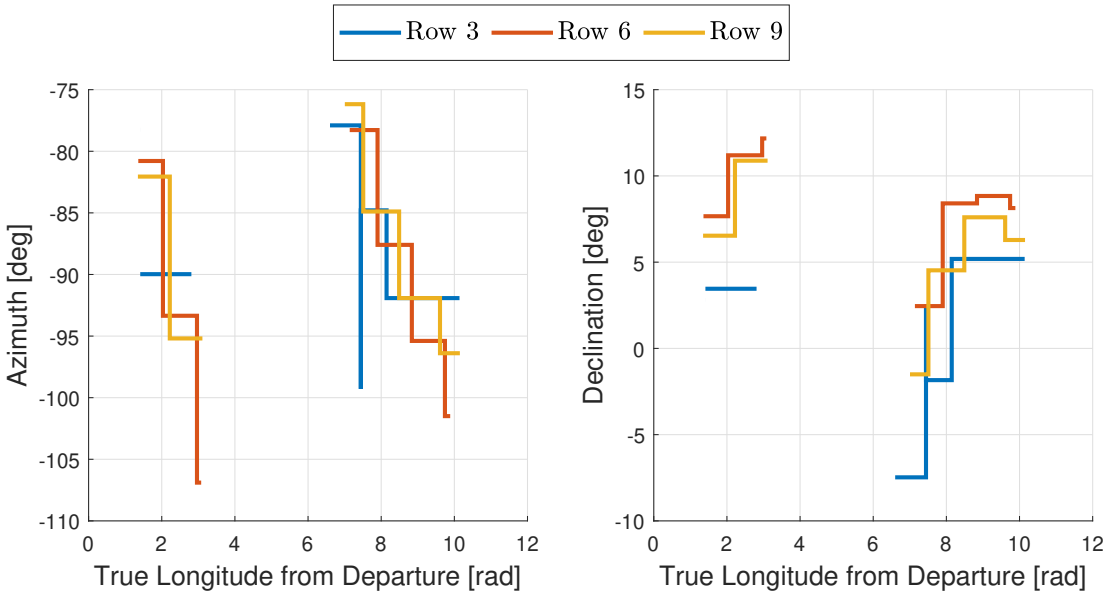


Figure D.7: Plot of the thrust vector azimuth as a function of the true longitude relative to departure for 3 of the solutions solutions in Table 5.12.

D.3 Case A3

Figure D.8 shows the histograms of the distribution of the QoIs for one of the solutions in the final approximation to the Pareto set provided by MACS, as in Fig. D.4. The black vertical line represents the selected solution from the final Pareto set. The blue histograms are the dispersion of the QoIs assuming a uniform distribution of the uncertain quantities.

The third solution is once again picked for having all lower expectations equal to one, and shown in Fig. D.9. Figures D.10, and D.11 show the trajectories in Table 5.13, which are similar to the ones shown in Section 5.7.1 since they are elements of the same set of control variables.

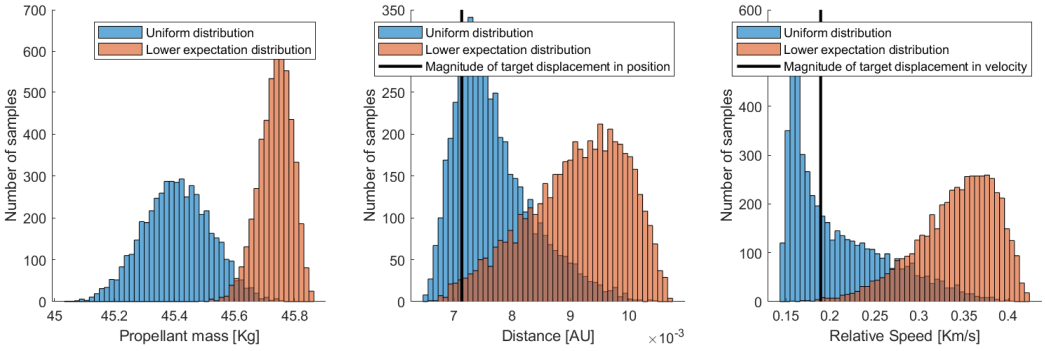


Figure D.8: Histograms showing the samples of the QoIs that are obtained for the 9th solution in Table 5.13 for test case 3. The blue histogram is the result of assuming a uniform distribution in Ξ, while in red is the distribution that corresponds to the lower expectation for each QoI. The vertical lines in the distance and relative speed histograms show the magnitude of the target displacement in position and velocity, respectively.

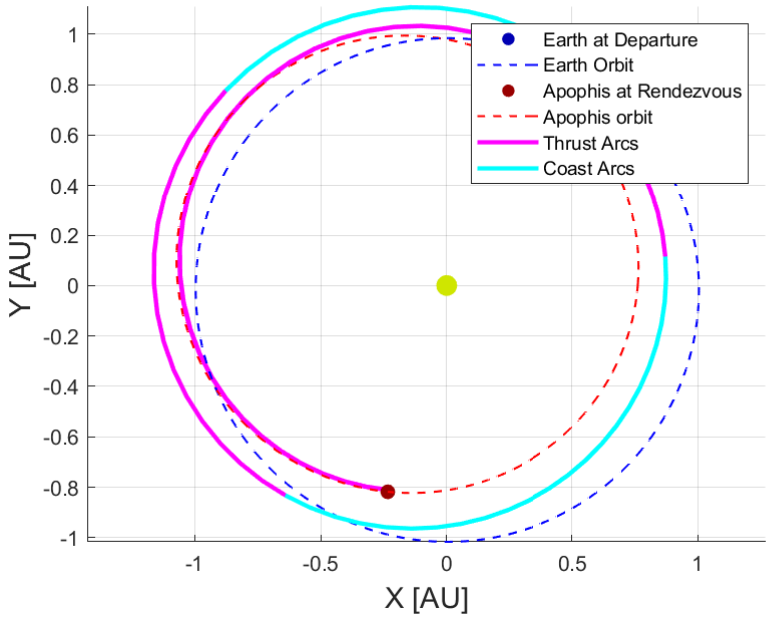


Figure D.9: Plot of the trajectory for the control law corresponding to the third solution in Table 5.13. The plot is seen perpendicularly to the ecliptic plane and the axis are in AU.

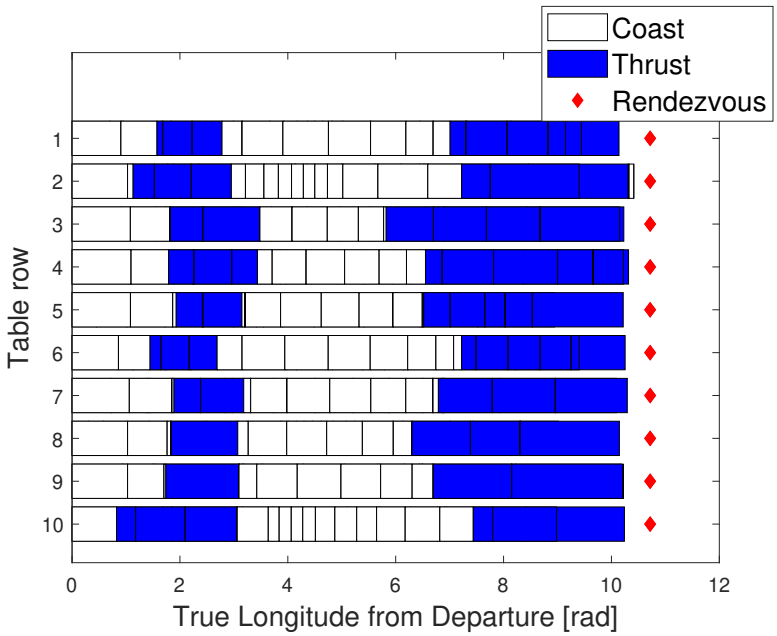


Figure D.10: Plot of the thrust profile as a function of the true longitude relative to departure for the solutions in Table 5.13. Blue segments represent thrust arcs, and white segments represent coast arcs. Note the presence of many zero-length arcs. The red diamond is the true longitude where Apophis is encountered (in the nominal case).

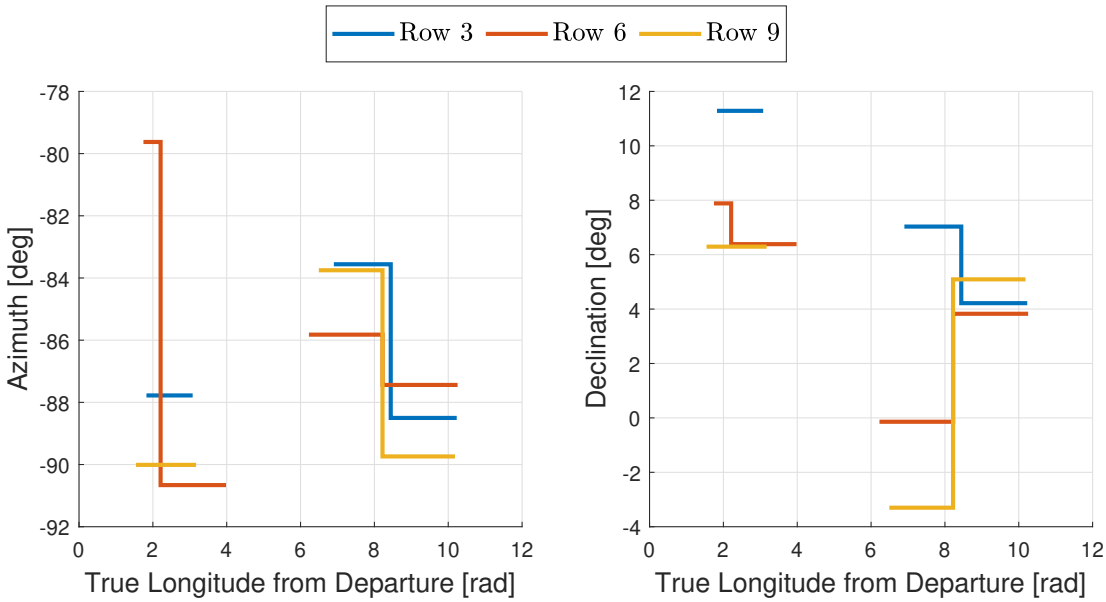


Figure D.11: Plot of the thrust vector azimuth as a function of the true longitude relative to departure for 3 of the solutions solutions in Table 5.13.

D.4 Case B

Figure D.12 shows the trajectory corresponding to the fifth solution in Table 5.14. Since none of the solutions had all 5 lower expectations equal to one, one of those with 4 $\underline{\mathrm{E}}$ equal to one was picked arbitrarily. Figures D.13 and D.14 represent the thrust/coast arcs, and the thrust azimuth and declination, respectively, for the solutions in the same table. This test case resulted in a much larger variety of solutions, with the thrust taking on multiple directions, in some cases tangential along the velocity direction, some cases in the opposite direction, and sometimes also pointing in the radial and normal directions. In most cases the thrust arcs occurred around the periapsis, where manoeuvres aimed at raising the apoapsis are most efficiently applied, which is necessary because in this case the launch vehicle did not impart sufficient velocity to reach all targets. The fact that the manoeuvres were not all applied tangentially, however, shows that some of the thrust was necessary for other reasons, such as to correct the orbital plane, or to ensure the spacecraft and the target reached the rendezvous location at the same time.

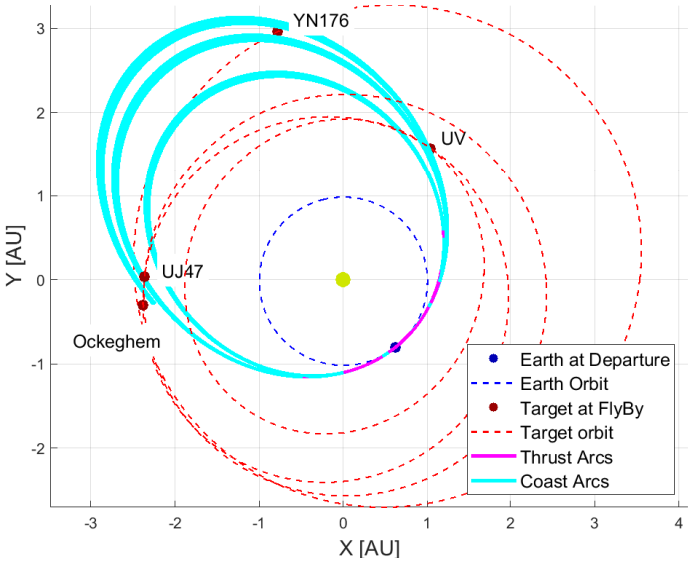


Figure D.12: Trajectories for the control law corresponding to the fifth solution in Table 5.14 and for 200 different samples of $\boldsymbol{\xi}$, so that the thickness represents the uncertainty in position. The plot is seen perpendicularly to the ecliptic plane.

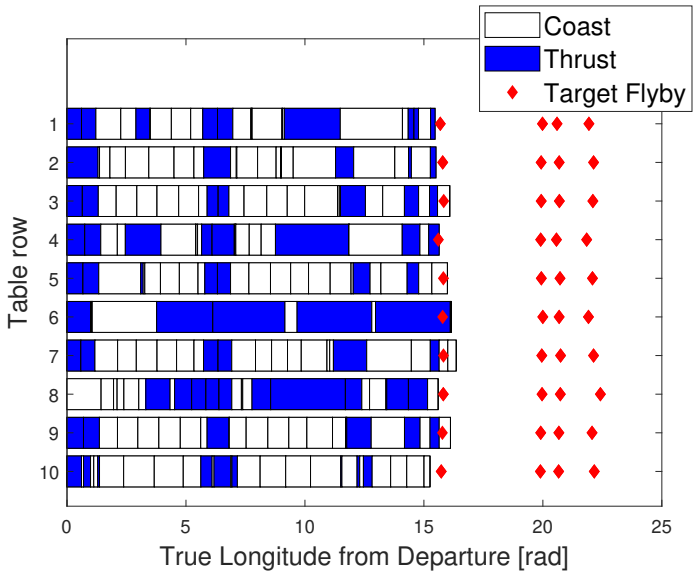


Figure D.13: Plot of the thrust profile as a function of the true longitude relative to departure for the solutions in Table 5.14. Note the many zero-length arcs.

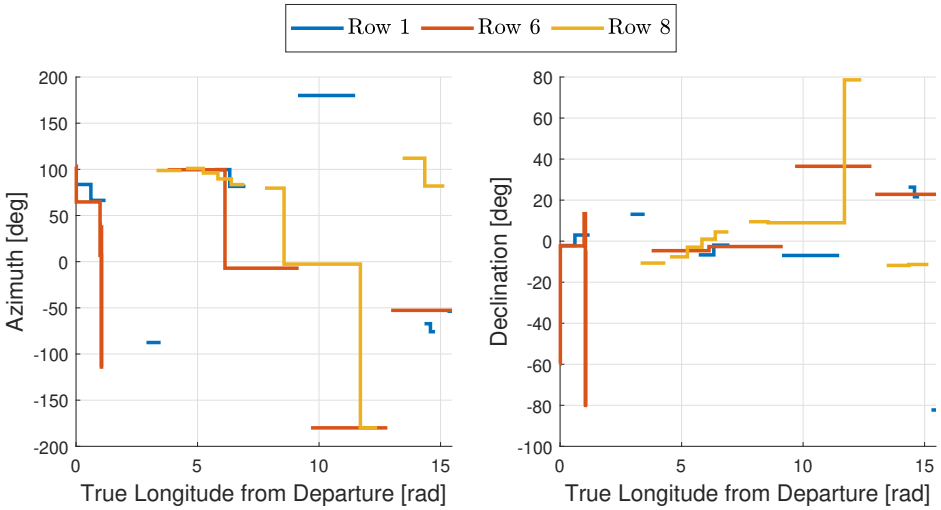


Figure D.14: Thrust vector azimuth (left) and declination (right) as a function of the true longitude relative to departure for some of the solutions in Table 5.14.

Bibliography

[1] National Aeronautics and Space Administration. OSIRIS-REx asteroid sample return mission fact sheet. `https://www.nasa.gov/wp-content/uploads/2016/06/osiris_rex_factsheet5-25.pdf`, 2016. Accessed: 2025-10-19.

[2] National Aeronautics and Space Administration. Double Asteroid Redirection Test (DART) Press Kit. `https://destination-orbite.net/documentations/DART-press-kit-web-FINAL.pdf`, 2022. Accessed: 2025-10-19.

[3] Japan Aerospace Exploration Agency. Hayabusa2 information fact sheet. `https://global.jaxa.jp/projects/sas/hayabusa2/pdf/sat33_fs_23_en.pdf`, 2018. Accessed: 2025-10-19.

[4] Richard H. Battin. *An Introduction to the Mathematics and Methods of Astrodynamics, Revised Edition.* American Institute of Aeronautics and Astronautics, 1999.

[5] Richard E. Bellman. *Dynamic Programming.* Princeton University Press, Princeton, NJ, 1957.

[6] F Bernelli-Zazzera, M Lavagna, R Armellin, P Di Lizia, A Morselli, J Olympio, and D Izzo. Trajectory optimisation under uncertainties. Ariadna Final Report, ESA/ACT, 2012.

[7] John T Betts. *Practical methods for optimal control and estimation using nonlinear programming.* SIAM, 2010.

[8] Leonardo C. T. Bezerra, Manuel López-Ibáñez, and Thomas Stützle. An empirical assessment of the properties of inverted generational distance on multi- and many-objective optimization. In Heike Trautmann, Günter Rudolph, Kathrin Klamroth, Oliver Schütze, Margaret Wiecek, Yaochu Jin, and Christian Grimme, editors, *Evolutionary Multi-Criterion Optimization*, pages 31–45, Cham, 2017. Springer International Publishing.

[9] R.A. Broucke and P. J. Cefola. On the equinoctial orbit elements. *Celestial Mechanics*, 5(3):303–310, 1972.

[10] Eduardo F Camacho and Carlos Bordons Alba. *Model predictive control*. Springer science & business media, 2013.

[11] M. Di Carlo, M. Vasile, and E. Minisci. Multi-population inflationary differential evolution algorithm with adaptive local restart. In *IEEE Congress on Evolutionary Computation (CEC)*, 2015.

[12] P. J. Cefola, A. C. Long, and G. Jr. Holloway. The long-term prediction of artificial satellite orbits. *AIAA 12th Aerospace Sciences Meeting*, 1974. , January 30-February 1, 1974, Washington.

[13] Nicolas Croisard, Massimiliano Vasile, Stephen Kemble, and Gianmarco Radice. Preliminary space mission design under uncertainty. *Acta Astronautica*, 66(5):654–664, 2010.

[14] Simão da Graça Marto and Massimiliano Vasile. Many-objective robust trajectory optimisation under epistemic uncertainty and imprecision. *Acta Astronautica*, 191:99–124, 2022.

[15] Simão da Graça Marto, Massimiliano Vasile, and Richard Epenoy. Multi-objective robust trajectory optimisation under epistemic uncertainty and imprecision. In *70th International Astronautical Congress*, 2019.

[16] Simão da Graça Marto and Massimiliano Vasile. Multi-objective robust trajectory optimization of multi asteroid fly-by under epistemic uncertainty. In Massimiliano

Vasile and Domenico Quagliarella, editors, *Advances in Uncertainty Quantification and Optimization Under Uncertainty with Aerospace Applications, Proceedings of the 2020 UQOP International Conference*. Springer International Publishing, November 2020.

[17] A. P. Dempster. Upper and lower probabilities induced by a multivalued mapping. *The Annals of Mathematical Statistics*, 38(2):325–339, 1967.

[18] M. Di Carlo, J. M. Romero Martin, and M. Vasile. Automatic trajectory planning for low-thrust active removal mission in Low-Earth Orbit. *Advances in Space Research*, 59(1):1234–1258, 2017.

[19] Marilena Di Carlo, Simão da Graça Marto, and Massimiliano Vasile. Extended analytical formulae for the perturbed keplerian motion under low-thrust acceleration and orbital perturbations. *Celestial Mechanics and Dynamical Astronomy*, 133, March 2021.

[20] Marilena Di Carlo, Juan Manuel Romero Martin, and Massimiliano Vasile. CAMELOT: Computational-analytical multi-fidElity low-thrust optimisation toolbox. *CEAS Space Journal*, 10(1):25–36, 2018.

[21] Marilena Di Carlo, Massimiliano Vasile, and Jamie Dunlop. Low-thrust tour of the main belt asteroids. *Advances in Space Research*, 62(8):2026 – 2045, 2018. Past, Present and Future of Small Body Science and Exploration.

[22] Marilena Di Carlo, Massimiliano Vasile, Cristian Greco, and Richard Epenoy. Robust optimisation of low-thrust interplanetary transfers using evidence theory. In Francesco Topputo, Andrew J. Sinclair, Matthew P. Wilkins, and Renato Zanetti, editors, *29th AAS/AIAA Space Flight Mechanics Meeting*, volume 168, pages 339–358, 2 2019.

[23] ESA ESTEC. Margin philosophy for science assessment studies. *ESA UNCLASSIFIED*, 2012.

[24] Scott Ferson, VLADIK KREINOVICK, Lev Ginzburg, and FARI SENTZ. Constructing probability boxes and dempster-shafer structures. Technical report, Sandia National Lab.(SNL-NM), Albuquerque, NM (United States), 2003.

[25] Gianluca Filippi and Massimiliano Vasile. A Memetic Approach to the Solution of Constrained Min-Max Problems. In *2019 IEEE Congress on Evolutionary Computation (CEC)*, pages 506–513, Wellington, New Zealand, June 2019. IEEE.

[26] Dan M. Goebel and Ira Katz. *Fundamentals of Electric Propulsion: Ion and Hall Thrusters*. Wiley, 1 edition, October 2008.

[27] Alexandr N. Golodnikov, Pavel S. Knopov, Panos M. Pardalos, and Stanislav P. Uryasev. Optimization in the space of distribution functions and applications in the bayes analysis. In Stanislav P. Uryasev, editor, *Probabilistic Constrained Optimization: Methodology and Applications*, Nonconvex Optimization and Its Applications, pages 102–131. Springer US, 2000.

[28] Cristian Greco, Stefano Campagnola, and Massimiliano L Vasile. Robust space trajectory design using belief stochastic optimal control. In *AIAA Scitech 2020 Forum*, page 1471, 2020.

[29] Cristian Greco, Marilena Di Carlo, Massimiliano Vasile, and Richard Epenoy. An intrusive polynomial algebra multiple shooting approach to the solution of optimal control problems. In *69th International Astronautical Congress*, 2018.

[30] Cristian Greco and Massimiliano Vasile. Closing the loop between mission design and navigation analysis. In *International Astronautical Congress*, 2020.

[31] John H. Halton. Quasi-probability: Why quasi-monte-carlo methods are statistically valid and how their errors can be estimated statistically. *Monte Carlo Methods and Applications*, 11(3), 2005.

[32] Clemens Heitzinger. *Simulation and Inverse Modeling of Semiconductor Manufacturing Processes*. dissertation, Technischen Universität Wien, 1997.

[33] Eyke Hüllermeier and Willem Waegeman. Aleatoric and epistemic uncertainty in machine learning: an introduction to concepts and methods. *Machine Learning*, 110(3):457–506, 2021.

[34] Wolfram Research, Inc. Mathematica, Version 13.3, 2023. Champaign, IL.

[35] Siwei Jiang, Yew-Soon Ong, Jie Zhang, and Liang Feng. Consistencies and contradictions of performance metrics in multiobjective optimization. *IEEE transactions on cybernetics*, 44(12):2391–2404, 2014.

[36] Simon J Julier and Jeffrey K Uhlmann. New extension of the kalman filter to nonlinear systems. In *Signal processing, sensor fusion, and target recognition VI*, volume 3068, pages 182–193. Spie, 1997.

[37] Refail Kasimbeyli, Zehra Kamisli Ozturk, Nergiz Kasimbeyli, Gulcin Dinc Yalcin, and Banu İçmen Erdem. Comparison of Some Scalarization Methods in Multiobjective Optimization. *The Bulletin of the Malaysian Mathematical Society Series 2*, 42:1875–1905, September 2019.

[38] Jean Albert Kechichian. Trajectory optimization using nonsingular orbital elements and true longitude. *Journal of Guidance, Control, and Dynamics*, 20(5):1003–1009, 1997.

[39] Steven G. Krantz. *Real Analysis and Foundations*. Studies in Advanced Mathematics. CRC Press, 2 edition, 2005.

[40] Frank E. Laipert and James M. Longuski. Automated missed-thrust propellant margin analysis for low-thrust trajectories. *Journal of Spacecraft and Rockets*, 52(4):1135–1143, 2015.

[41] Gregory Lantoine and Ryan P Russell. A hybrid differential dynamic programming algorithm for constrained optimal control problems. part 1: Theory. *Journal of Optimization Theory and Applications*, 154(2):382–417, 2012.

[42] Li-zhi Liao and Christine A Shoemaker. Advantages of differential dynamic programming over newton's method for discrete-time optimal control problems. Technical report, Cornell University, 1992.

[43] Søren N Lophaven, Hans Bruun Nielsen, and Jacob Søndergaard. A matlab kriging toolbox. *Technical University of Denmark, Kongens Lyngby, Technical Report No. IMM-TR-2002-12*, 2002.

[44] Wuan Luo. *Wiener chaos expansion and numerical solutions of stochastic partial differential equations*. California Institute of Technology, 2006.

[45] Simao Graca Marto, Massimiliano Vasile, and Richard Epenoy. Multi-Objective Robust Trajectory Optimisation Under Epistemic Uncertainty and Imprecision. *70th International Astronautical Congress*, page 13, 2019.

[46] David Q Mayne. Differential dynamic programming–a unified approach to the optimization of dynamic systems. In *Control and dynamic systems*, volume 10, pages 179–254. Elsevier, 1973.

[47] Budiman Minasny. Latin hypercube sampling. `https://www.mathworks.com/matlabcentral/fileexchange/4352-latin-hypercube-sampling`, MATLAB Central File Exchange., 2004. Retrieved May 23, 2021. File lhsu.m specifically was used.

[48] Harald Niederreiter. *Random number generation and quasi-Monte Carlo methods*. Number 63 in CBMS-NSF regional conference series in applied mathematics. Society for Industrial and Applied Mathematics, Philadelphia, Pa, 1992.

[49] H. B. Nielsen, S. N. Lophaven, and J. Søndergaard. DACE - a matlab kriging toolbox, 2002.

[50] W.L. Oberkampf and J.C. Helton. Investigation of evidence theory for engineering applications. In *AIAA 2002-1569, Denver Colorado*, 4th Non-Deterministic Approaches Forum, 2002.

[51] Joris T Olympio and Chit Hong Yam. Deterministic method for space trajectory design with mission margin constraints. In *61st International Astronautical Congress*, pages 1–12. International Astronautical Federation Prague, 2010.

[52] Irfan M. Ovacik, Srikanth Rajagopalan, and Reha Uzsoy. Integrating interval estimates of global optima and local search methods for combinatorial optimization problems. *Journal of Heuristics*, 6(4):481–500, 2000.

[53] Naoya Ozaki, Stefano Campagnola, Chit Hong Yam, and Ryu Funase. Differential dynamic programming approach for robust-optimal low-thrust trajectory design considering uncertainty. In *25th International Symposium on Space Flight Dynamics, Munich, Germany*, pages 19–23, 2015.

[54] Frédéric Pierret. Stochastic gauss equations. *Celestial Mechanics and Dynamical Astronomy*, 124(2):109–126, Oct 2015.

[55] Marc D. Rayman, Thomas C. Fraschetti, Carol A. Raymond, and Christopher T. Russell. Coupling of system resource margins through the use of electric propulsion: Implications in preparing for the dawn mission to ceres and vesta. *Acta Astronautica*, 60(10):930–938, 2007.

[56] Lorenzo A. Ricciardi and Massimiliano Vasile. Improved archiving and search strategies for multi agent collaborative search. In Edmondo Minisci, Massimiliano Vasile, Jacques Periaux, Nicolas R. Gauger, Kyriakos C. Giannakoglou, and Domenico Quagliarella, editors, *Advances in Evolutionary and Deterministic Methods for Design, Optimization and Control in Engineering and Sciences*, Computational Methods in Applied Sciences, pages 435–455. Springer International Publishing, 2019.

[57] Lorenzo Angelo Ricciardi, Massimiliano Vasile, Federico Toso, and Christie A. Maddock. Multi-objective optimal control of the ascent trajectories of launch vehicles. In *AIAA/AAS Astrodynamics Specialist Conference*. American Institute of Aeronautics and Astronautics, 2016. _eprint: https://arc.aiaa.org/doi/pdf/10.2514/6.2016-5669.

[58] Arthur Richards and Jonathan How. Robust stable model predictive control with constraint tightening. In *2006 American Control Conference*, pages 6–pp. IEEE, 2006.

[59] Simo Särkkä and Arno Solin. *Applied stochastic differential equations*, volume 10. Cambridge University Press, 2019.

[60] G. Shafer. *A mathematical theory of evidence*. Princeton University Press, 1976.

[61] Glenn Shafer. *A mathematical theory of evidence*. Princeton Univ. Press, 1976.

[62] Ankur Sinha, Zhichao Lu, Kalyanmoy Deb, and Pekka Malo. Bilevel optimization based on iterative approximation of multiple mappings. *Journal of Heuristics*, 26(2):151–185, 2020.

[63] G Soulas, M Domonkos, and M Patterson. Performance evaluation of the next ion engine. In *39th AIAA/ASME/SAE/ASEE Joint Propulsion Conference and Exhibit*, page 5278, 2003.

[64] Yuval Tassa, Nicolas Mansard, and Emo Todorov. Control-limited differential dynamic programming. In *2014 IEEE International Conference on Robotics and Automation (ICRA)*, pages 1168–1175. IEEE, 2014.

[65] D. A. Vallado. *Fundamentals of Astrodynamics and Applications*. Springer, 2007.

[66] Peter J. M. van Laarhoven, Emile H. L. Aarts, and Jan Karel Lenstra. Job shop scheduling by simulated annealing. *Operations Research*, 40(1):113–125, 1992.

[67] Massimiliano Vasile. Robust mission design through evidence theory and multiagent collaborative search. *Annals of the New York Academy of Sciences*, 1065(1):152–173, 2005.

[68] Massimiliano Vasile and Chiara Tardioli. On the use of positive polynomials for the estimation of upper and lower expectations in orbital dynamics. In *Stardust Final Conference*, pages 99–107, 2018.

[69] Ferdinand Verhulst. *Nonlinear differential equations and dynamical systems*. Universitext. Springer, Berlin, 1990. OCLC: 246747652.

[70] Akifumi Wachi, Naoya Ozakiz, and Shinichi Nakasuka. Fault-tolerant low-thrust trajectory design with backups for multiple targets. *IFAC-PapersOnLine*, 49(17):391–396, 2016.

[71] Peter Walley. *Statistical reasoning with imprecise probabilities*, volume 42. Springer, 1991.

[72] Don Yeomans, Steve Chesley, and Paul Chodas. Near-earth asteroid 2004 mn4 reaches highest score to date on hazard scale. `https://cneos.jpl.nasa.gov/news/news146.html`, accessed: 2019-09-17, 2004.

[73] Federico Zuiani and Massimiliano Vasile. Extended analytical formulas for the perturbed keplerian motion under a constant control acceleration. *Celestial Mechanics and Dynamical Astronomy*, 121(3):275–300, 2015.

[74] Federico Zuiani, Massimiliano Vasile, Alessandro Palmas, and Giulio Avanzini. Direct transcription of low-thrust trajectories with finite trajectory elements. *Acta Astronautica*, 72:108–120, March 2012.