Prospects on Solving an Optimal Control Problem with Bounded Uncertainties on Parameters using Interval Arithmetics

Etienne Bertin, Elliot Brendel, Bruno Hérissé, Julien Alexandre Dit Sandretto, Alexandre Chapoutot

HAL Id: hal-03142715
https://hal.science/hal-03142715

Submitted on 16 Feb 2021

Prospects on solving an optimal control problem with bounded uncertainties on parameters using interval arithmetic

Etienne Bertin*^{a}, Elliot Brendel*, Bruno Hérissé*,
Julien Alexandre dit Sandretto^a, Alexandre Chapoutot^a

Abstract

An interval method based on Pontryagin's Minimum Principle is proposed to enclose the solutions of an optimal control problem with embedded bounded uncertainties. This method is used to compute an enclosure of all optimal trajectories of the problem, as well as open loop and closed loop enclosures meant to validate an optimal guidance algorithm on a concrete system with inaccurate knowledge of the parameters. The differences in geometry of these enclosures are exposed, and showcased on a simple system. These enclosures can guarantee that a given optimal control problem will yield a satisfactory trajectory for any realization of the uncertainties. Contrarily, the probability of failure may not be eliminated and the problem might need to be adjusted.

Keywords: Optimal Control, Pontryagin's Principle, Interval Arithmetic, Bounded Uncertainties, Penalization

1 Introduction

Optimal control of aerospace systems is performed by modeling the considered system by dynamics depending on multiple uncertain parameters (for example, aerodynamic coefficients, mass,...). Usually, optimal control problems are solved for nominal values of these parameters and the robustness of the solution is demonstrated by dispersing the parameters around nominal values with Monte Carlo simulations [5]. Another approach is to solve a chance constrained optimal control problem in order to guarantee a probability of success [7]. These methods do not exclude the eventuality of failure, which can be problematic on critical systems. In

---

*DTIS, ONERA, Université Paris Saclay, F-91123 Palaiseau, France. E-mail: firstname.lastname@onera.fr

^{a}U2IS, ENSTA Paris, Institut Polytechnique de Paris, Palaiseau, France. E-mail: firstname.lastname@ensta-paris.fr
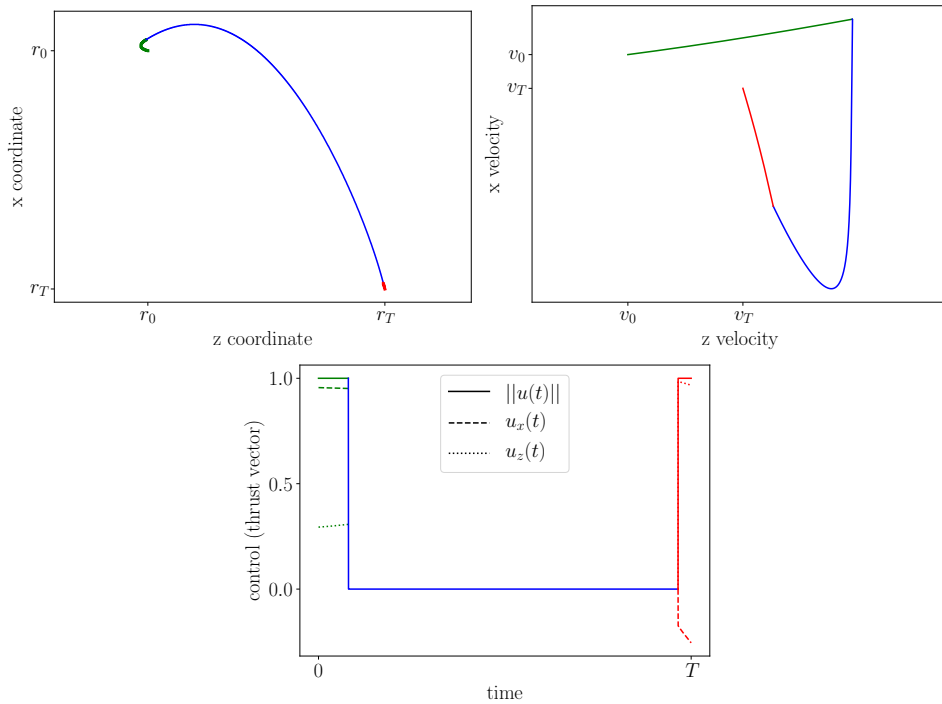
Figure 1: Optimal trajectory, velocity space trajectory and control for Goddard's re-entry problem in Cartesian Geo-centered coordinates[0]. In green the boostback burn, in thin blue the ballistic phase and in red the landing burn. Burn phases are characterized by a saturated control ($||u|| = 1$) and a dash in velocity space, while ballistic phases are characterized by no control ($||u|| = 0$). The model, the parameters and the initial and final positions are taken from [4]. More details on the model and boundary conditions are given in Section 5.

addition these problem-solving methods often introduce numerical approximations, which question the validity of the results.

Interval arithmetic has shown their ability to address several control problems [9], providing validated solutions while dealing with method uncertainties (numerical approximations) as well as with model uncertainties (unknown but bounded parameters). For instance the reachable set of a dynamic system can be over-approximated. This over-approximation can demonstrate that a system will remain out of a critical region, thus guarantee its safety [2]. These methods can also be used to design robust optimal control. For instance, [11, 12] propose algorithms to compute a control that stirs a system to a desired state for any realization of a bounded noise while achieving the lowest upper bound on the cost.

---

[0]Standard coordinates for rocket trajectory are altitude and latitude; Cartesian coordinates were chosen instead to allow some consistency with figures in Section 5

Pontryagin's Maximum Principle [13] (PMP) provides necessary optimality conditions for the resolution of optimal control problems by transforming an optimal control problem into a root-finding problem. Derived methods have proven their efficiency and their precision compared to direct methods [13, 14], but their convergence strongly depends on their initialization and a prior knowledge of the solution structure is needed.

Our goal is to address an aerospace challenge highlighted by [3], which is to enclose the trajectory of a reusable launch vehicle subject to external perturbations. The trajectory during the landing burn is computed online so the control is not known beforehand. Computing an enclosure could guarantee that the vehicle lands safely on the landing platform.

To that end, the return version of Goddard's problem [5] is considered. It consists in performing the landing of the first stage of a rocket while minimizing its fuel consumption. The optimal trajectory of this system for nominal values of the parameters and the associated control are presented on Figure 1. Uncertainties on the parameters will be added into the model in the form of intervals and deterministic enclosures will be computed by combining interval arithmetic and the necessary optimality conditions given by the application of PMP. Although this method cannot be applied to Goddard's problem yet, this paper exposes our reasoning on simplified problems.

Firstly, Section 2 introduces the problem considered. Section 3 presents the interval and optimal control methods that will be later used. Section 4 proposes three enclosures that can be utilized to discuss the reliability of an optimal control problem with uncertainties. Section 5 showcases these enclosures on a simplified problem. Lastly, Section 6 proposes some uses of these enclosures and the remaining work to apply these methods on Goddard's problem.

Notations

Interval variables and interval vectors are always enclosed in brackets: $[x]$.
Any variable that is not enclosed in bracket is a real or vector variable.
Interval valued functions are enclosed in brackets: $[f] : [x] \mapsto [f]([x])$.
The bounds of an interval $[a]$ are noted $\underline{a}, \overline{a}$. Its middle point is noted $\mathrm{mid}\,([a])$.
$(.)$ such as in $y(.)$ denotes a time function: $y(.) : t \in [0, T] \mapsto y(t) \in \mathbb{R}^n$.
$y(\tau)$ and y_τ are different, the later being a vector unrelated to function $y(.)$.
A hat as in $\hat{\xi}$ denotes a variable that is related to an estimation.

2   Control problem with uncertainties

In this section, the optimal control problem and the incorporation of uncertain parameters are presented. Then three enclosures meant to discuss the relevance of an optimal control problem will be introduced.

2.1   Optimal Control Problem (OCP)

The following non-interval optimal control problem is considered

$$\min_{u(.)\in\mathcal{U}} \int_0^T \ell(y(t), u(t))dt + \Psi(y(T)) \text{ s.t. } \begin{cases} \dot{y}(t) = f(y(t), u(t), \xi(t), t), \\ y(0) = y_0 \\ T \text{ is fixed.} \end{cases} \qquad (1)$$

Interpretation : the system, a launcher for instance, is characterized by the following data:

- A state $y(t)$ e.g. position, velocity and mass of the launcher. The time function $y(.)$ is called a trajectory.

- An initial state y_0.

- Dynamics $\dot{y}(t) = f(y(t), u(t), \xi(t), t)$ e.g. the time derivative of position is velocity, the derivative of velocity is the forces divided by mass etc. . . .

- A control input $u(.)$ e.g. the reactor thrust vector.

- Parameters $\xi(.)$ e.g. aerodynamic coefficients, maximum thrust. . . .

- A continuous cost $\ell(y(t), u(t))$ e.g. fuel consumption.

- A final cost $\Psi(y(T))$ e.g. the distance between the final state and the target.

System (1) is considered controllable. Hence for any initial state y_0 and specific parameter function $\xi(.)$, an optimal control $u(.)$ can be found. As a consequence, a control can be defined implicitly as the solution of an OCP, as opposed to explicit methods such as time series or gain tables. For instance, the Model Predictive Control (MPC) approach defines the control as the solution of an optimal control problem; the controller solves this optimal control problem at each time step to determine the control to be applied. Some parts of a launch vehicle trajectory are also defined by an OCP: [3] states that the control during the landing burn is the optimal solution of a problem that is solved autonomously during the flight.

Bounded uncertainties on the initial state and the parameters will be added into this model. The initial state y_0 and parameter function $\xi(.)$ take their values in interval enclosures: $y_0 \in [y_0]$ and $\xi(t) \in [\xi], \forall t$, where $[\xi]$ and $[y_0]$ are intervals of the form $\left[\underline{\xi}, \overline{\xi}\right]$ or boxes (see definition in Section 3.2.1).

Since they are challenging to simulate with validated simulation, optimal control problems with varying time horizons and state dependant transitions are not considered in this paper. However, it is worth noting that time dependent transitions, or switches, are possible (see Section 3.2.3). This assumption excludes Goddard's problem. The bang-bang transition from a zero control to a saturated control seen in Figure 1 is a state dependent transition and cannot be simulated with our method, unless a precise switch time is forced. Section 6.2 discusses the steps to be taken to apply this method to Goddard's problem.

Note also that the presented method requires additional regularity conditions which will be presented in Section 4.

2.2 Representation of the uncertainties in the OCP

Parameters are bounded functions of time: $\forall t \in [0, T], \xi(t) \in [\xi]$. It is also assumed that each function $\xi(.)$ is infinitely differentiable. This ensures that dynamics have all the needed mathematical properties to be simulated.

The time dependency establishes a general framework without invalidating the important assumptions of the optimality criterion used (see Section 3.1.2). It covers indeed more scenarios than the approach considering that the parameters are unknown constant values, which is an implicit assumption in [5] in which parameters are picked randomly at the start and stay constant during the simulations.

The most encompassing framework would consider that parameters are also functions of the state as a whole. However that would arise the need for more assumptions (bounded spatial derivatives...).

2.3 Contributions

The motivation is to assess whether a given OCP outputs an adequate trajectory for any realization of the initial state and parameter functions. A trajectory is characterized by two realizations of the parameter function: the actual parameter function $\xi(.)$ and the parameter function estimated by the system $\hat{\xi}(.)$. Both are taken amidst the same bound $[\xi]$ but they may differ as the actual parameters are often unknown in practice.

Hence, the following three enclosures of the solution of (1) are proposed:

- An *anticipative enclosure* containing trajectories whose control is computed with the perfect knowledge of the parameter function: $\hat{\xi}(.) = \xi(.)$.

- An *open loop*[1] *enclosure* containing trajectories whose control is computed once at the start with an inaccurate parameter function $\hat{\xi}(.) \neq \xi(.)$ and then followed blindly.

- A *closed loop*[2] *enclosure* containing trajectories whose control is computed several times online with accurate measurements of the state but an inaccurate parameter function $\hat{\xi}(.) \neq \xi(.)$.

Measures of the state are considered accurate, contrarily to parameters that are unknown. The reason is that knowing parameter in advance is fundamentally impossible because it requires seeing into the future, while having accurate measurement is only a matter of hardware. Bounded errors on measurements will be considered in future works.

These three enclosure can be used to validate all solutions of an OCP and assess risks. Contrarily to the approach in [11], which optimizes a single control that works for all realizations of the bounded uncertainties, the proposed method does not output a control directly. Instead, it can guarantee that a given optimal control

---

[1]also known as feed forward
[2]also known as feedback

problem will produce a satisfactory control for all realizations of the uncertainties on the initial state and parameter function.

To take the example of launch vehicles [3], an OCP is solved right before the landing burn to compute a guidance trajectory. This OCP is designed to provide a trajectory to the landing platform with any possible dynamics and from any possible state the launcher may be in right before the landing burn. The proposed approach is to use a conservative enclosure of all possible dynamics and of all possible states before the landing burn to derive enclosures of all trajectories that are solutions of a given OCP with these initials states and dynamics. If every element in the enclosures satisfy the final constraint of finishing on the landing platform, then the OCP is guaranteed to output a satisfactory trajectory in practice and it may be used on an actual launcher.

This is only a necessary condition. Since the enclosures are conservative, the fact that the enclosure does not satisfy the constraints does not prove the existence of a realization of the uncertainties that will cause the launcher to violate the constraints. Nevertheless, on such critical systems, the possibility of failure is not tolerated and may justify changing the OCP.

3   Mathematical preliminaries

In this section, a numerical method for optimal control is presented as well as interval-based validation methods. Those methods will be combined in Section 4.

3.1   Numerical resolution of a control problem

As a first step, properties of dynamical systems will be recalled and the flow and resolvent notations will be introduced. Then these notations will be used to formalize the indirect method known as Pontryagin's Minimum Principle.

3.1.1   Flow and resolvent of an ordinary differential equation (ODE)

Consider an ODE in the form:

$$\begin{cases} \dot{x}(t) & = g(x, \xi(t), t) \\ x(t_0) & = x_0 \end{cases}$$

To simplify notations and since parameters only depend on time, $g(x, \xi(t), t)$ is denoted by $g(x, t)$ whenever doing so does not impede comprehension.

A solution can be approximated by numerical methods, such as Euler's method, or more efficient Runge-Kutta methods [6]. Integrating the dynamics between two boundary times τ and T can be seen as a flow function $\Phi_{\tau,T}$. The function $\Phi_{\tau,T}$ takes an initial state, simulates it from τ to T and returns the final state. If g is twice differentiable, the flow $\Phi_{\tau,T}$ has a spatial derivative in x_τ which is the resolvent $R_{\tau,T}(x_\tau)$ of the linearized system [8].

To illustrate these notations, consider two boundary times τ and T. Let $x(.)$ be the solution of the ODE:

$$\begin{cases} \dot{x}(t) & = g(x,t) \\ x(\tau) & = x_\tau \end{cases}$$

$\Phi_{\tau,T}(x_\tau)$ is the state of the solution at time T, $\Phi_{\tau,T}(x_\tau) = x(T)$.

If $z(.)$ is the solution of the same ODE with an initial perturbation δx:

$$\begin{cases} \dot{z}(t) & = g(z,t) \\ z(\tau) & = x_\tau + \delta x \end{cases}$$

then its final state will be $z(T) = x(T) + R_{\tau,T}(x_\tau) \cdot \delta x + o(\|\delta x\|)$. The resolvent of the linearized system enables a first order approximation of the final error caused by an initial perturbation, and as such, it is the spatial first derivative of the flow.

The resolvent can sometimes be computed analytically, notably for linear time invariant systems: if $\dot{x} = A \cdot x$ then $R_{\tau,T}(x_\tau) = \exp((T-\tau)A)$. If no analytic formula is available, $R_{\tau,T}(x_\tau)$ can be computed by integrating the following ODE:

$$\begin{cases} \dot{x}(t) & = g(x) \\ \dot{R}_{\tau,t}(x_\tau) & = \dfrac{\partial g}{\partial x}(x(t)) \cdot R_{\tau,t}(x_\tau) \\ x(\tau) & = x_\tau \\ R_{\tau,\tau}(x_\tau) & = I_n. \end{cases} \tag{2}$$

In the following sections, the spatial arguments will often be removed to simplify notations. Hence $R_{\tau,T}$ means $R_{\tau,T}(x_\tau)$.

3.1.2 Optimal control

An OCP is considered as in Section 2

$$\min_{u(.)\in\mathcal{U}} \int_\tau^T \ell(y(t),u(t))dt + \Psi(y(T)) \text{ s.t. } \begin{cases} \dot{y}(t) = f(y(t),u(t),\xi(t)), \forall t \in [\tau,T], \\ y(\tau) = y_\tau \\ T \text{ is fixed.} \end{cases}$$

This control can be computed by multiple methods that mainly fall in two categories:

- *direct methods*: the control is discretized, thus turning the infinite dimensional problem into a high dimensional Euclidean optimization problem.

- *indirect methods*: using a characterization of the optimal trajectory and control, the infinite dimensional optimization problem is transformed into a simpler problem.

The second approach is considered in this article.

To characterize the optimal solution, a new variable is added, the co-state $p(.)$ which is analog to the dual multiplier in constrained Euclidean optimization. $p(.)$

is a vector valued time function on the time range $[0, T]$ and the vectors $p(t)$ are the same dimension as the vectors $y(t)$.

The main result is the following theorem. This is a simplified version that is sufficient for the problem tackled. It does not explicit the domain of each function and does not address abnormal cases. A complete theorem can be found in [4, 13].

Pontryagin's Minimum Principle[2] (PMP):   If $(y(.), u(.))$ are a normal optimum, then there exists a non trivial co-state $p(.)$ such that $(y(.), p(.), u(.))$ satisfy the following equations:

$$H(y, p, u, t) = \ell(y, u) + p.f(y, u, \xi(t))$$
$$\begin{cases} \dot{p}(t) = -\dfrac{\partial H}{\partial y}(y(t), p(t), u(t), t) \\ p(T) = \dfrac{\partial \Psi}{\partial y}(y_T) \end{cases}$$
$$u(t) \in \arg\min_u H(y(t), p(t), u, t) \forall t \in [\tau, T],$$

where H is called the pre-Hamiltonian. The optimal control $u(.)$ is defined implicitly as minimizing the pre-Hamiltonian at every time.

In many control problems this implicit definition yields an explicit expression. That is, there is a function $\mu : y, p, t \to \mu(y, p, t)$ such that:

$$u(t) \in \arg\min_u H(y, p, u, t) \iff u(t) = \mu(y, p, t). \tag{3}$$

For instance, [4] shows that in Goddard's problem, the minimization condition implies that the control (i.e. the thrust vector) has the same orientation as a part of the co-state and that the magnitude of the control tends to be either saturated or zero based on a switching function that depends solely on the state and co-state (which leads to a bang-off-bang control as seen on Figure 1). Thus, the control in Goddard's problem is determined by the state and the co-state. In the double integrator problem defined in Section 5, the control minimizes the pre-Hamiltonian $H(y, p, u) = ||u||^2/2 + p_v \cdot u$, which yields the explicit expression $u(t) = \mu(y, p, t) = -p_v$.

Noting $g^y(y, p, t) = f(y, \mu(y, p, t), t)$ and $g^p(y, p, t) = -\partial H/\partial y(y, \mu(y, p, t), p, t)$. The equations of the PMP can be combined with those of Problem (1) to form the following two point boundary value problem:

$$\begin{cases} \dot{y}(t) & = g^y(y(t), p(t), t) \\ \dot{p}(t) & = g^p(y(t), p(t), t) \\ y(\tau) & = y_\tau, \quad p(\tau) \quad \text{is free} \\ y(T) \quad \text{is free,} \quad p(T) & = \dfrac{\partial \Psi}{\partial y}(y_T) \end{cases} \tag{4}$$

This is a problem of finding the value of $p(\tau)$ and $y(T)$ such that integrating from the state at time τ leads to the state at time T and vice versa. In many applications, it

---

[2]Also known as Pontryagin's Maximum Principle. There are multiple formalization of this principle in the literature.

is enough to solve the Initial Value Problem (IVP), that is finding $p(\tau)$. By splitting the flow of Section 3.1.1, note $\phi_{\tau,T}^y(y_\tau, p_\tau)$ the flow function that returns the final state $y(T)$ and $\phi_{\tau,T}^p(y_\tau, p_\tau)$ the flow function that returns the final co-state $p(T)$. For a given y_τ, the IVP can be written as follow:

$$\text{find } p_\tau \text{ such that } C(y_\tau, p_\tau) = \phi_{\tau,T}^p(y_\tau, p_\tau) - \frac{\partial \Psi}{\partial y}\left(\phi_{\tau,T}^y(y_\tau, p_\tau)\right) = 0. \qquad (5)$$

The IVP (5) is solved with a shooting method. First a guess of the initial co-state is taken. Then the system is *shot*, i.e. the ODE is integrated until the final time. Then the initial guess is corrected depending on how far the system landed from the target. This is repeated until a satisfactory co-state is found.

A shooting method can be complemented with a continuation method[3] when the problem is hard to initialize. See [4, 5] for the application of this method to a launcher system.

It is worth noting that the more general PMP states that if the dynamics do not depend on time, then the pre-Hamiltonian is stationary. This is not the case under our assumption since the dynamics depend on parameters that depend on time. It follows that the method used in [4, 5] to compute singular arc of the control, which is based on the stationarity of the pre-Hamiltonian, cannot be used under our assumption. Hence we currently have no method to enclose singular arcs, apart from a very crude over-approximation.

3.1.3 First derivative of the IVP

One way to solve the OCP is to apply a zero finding method to Equation (5). It can be useful to compute the derivatives of $\phi_{\tau,T}^y(y_\tau, p_\tau)$ and $\phi_{\tau,T}^p(y_\tau, p_\tau)$.

Let the variable $x(.) = (y(.), p(.))$. This variable is subject to an ODE $\dot{x} = g(x, t)$ as seen in System (4). It follows that if g is twice differentiable, a resolvent $R_{\tau,T}^x$ can be computed for this system using Equation (2). The resolvent $R_{\tau,T}^x$ is split in four square Matrices:

$$R_{\tau,T}^x = \begin{pmatrix} * & R_{\tau,T}^y \\ * & R_{\tau,T}^p \end{pmatrix}.$$

The two square matrices marked by $*$ are related to the final deviation caused by a variation of the initial state and are not useful here. $R_{\tau,T}^y$ (resp. $R_{\tau,T}^p$) is the final state (resp. co-state) deviation caused by a variation of the initial co-state and is the spatial derivative of $\phi_{\tau,T}^y(y_\tau, p_\tau)$ (resp. $\phi_{\tau,T}^p(y_\tau, p_\tau)$).

By splitting the boundary condition $R_{\tau,\tau}^x = I_d$ and taking the upper right and bottom right corner, boundary condition $R_{\tau,\tau}^y = 0$ and $R_{\tau,\tau}^p = I_d$ are obtained. Similarly, splitting the dynamic $\dot{R}_{\tau,t}(x_\tau) = \partial f/\partial x(x(t)) \cdot R_{\tau,t}(x_\tau)$ yields the dynamics of $R_{\tau,T}^y$ and $R_{\tau,T}^p$. Hence ODE (6) is deduced.

---

[3]also known as homotopic method.

$$
\begin{cases}
\dot{y}(t) & = g^y(y(t), p(t), t) \\
\dot{p}(t) & = g^p(y(t), p(t), t) \\
\dot{R}^y_{\tau,t} & = \dfrac{\partial g^y}{\partial y}(y, p, t) \cdot R^y_{\tau,T} + \dfrac{\partial g^y}{\partial p}(y, p, t) \cdot R^p_{\tau,T} \\
\dot{R}^p_{\tau,t} & = \dfrac{\partial g^p}{\partial y}(y, p, t) \cdot R^y_{\tau,T} + \dfrac{\partial g^p}{\partial p}(y, p, t) \cdot R^p_{\tau,T} \\
y(\tau) & = y_\tau \\
p(\tau) & = p_\tau \\
R^y_{\tau,\tau} & = 0 \\
R^p_{\tau,\tau} & = I_d,
\end{cases}
\tag{6}
$$

By simulating ODE (6), the value $C(y_\tau, p_\tau)$ and first derivative $\partial C/\partial p(y_\tau, p_\tau)$ of the IVP are obtained.

3.2   Validation methods

First, the general idea of interval arithmetic is recalled. Then two methods based on this arithmetic are presented: Krawczyk method that encloses the solution of a vector-valued equation and validated simulation that encloses the evolution of a dynamical system.

3.2.1   Set representation using interval arithmetic

An interval $[a]$ is a convex subset of $\mathbb{R}$ that contains all reals between a lower bound $\underline{a}$ and an upper bound $\overline{a}$. With $\underline{a}, \overline{a} \in \mathbb{R} \cup \{-\infty, +\infty\}$. Interval arithmetic can be used as an alternative to floating-point arithmetic to obtain an enclosure of the solution of a problem, rather than an approximation [10].

For each real valued function $f : a \mapsto f(a)$, one can create an inclusion function $[f] : [a] \mapsto [f]([a])$ following a set-membership principle: the result of the interval function $[f]$ is an interval that contains each possible value of f on $[a]$

$$
[f]([a]) \supset \{f(a) | \forall a \in [a]\}.
$$

This definition has an inclusion rather than an equality because the set on the left might not be an interval. Moreover, finding the minimal enclosure of the actual set is difficult. The tightness of the results depends on the effort put into their computations. For this reason, there may be multiple inclusion functions for a single real valued function.

Interval vectors, or boxes, are an axis-aligned rectangular set in a finite dimensional space. They are an inexpensive representation of a high dimensional set (compared to polytops) but may induce a wrapping effect during computations [9].

In lower dimension, an accurate enclosure of a set can be achieved with a paving of boxes, a set of mutually disjoint boxes which together cover the whole set. A simple paving is illustrated on Figure 2. A paving in which boxes are all the same shape and aligned can be called a tiling. Such a representation is potentially very precise, but the computational cost grows very fast in high dimension. Indeed, to
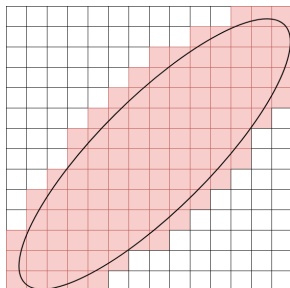
Figure 2: Outer tiling of a set. The ellipsoidal set is enclosed by the union of red boxes.

double the precision, the number of tiles has to be doubled in each direction, which means 2^n as many tiles, where n is the dimension of the vector space. Hence the complexity of this representation is exponential in the dimension of the state, which makes it ill-suited for many practical cases.

3.2.2 Krawczyk contractor

Krawczyk's method is an interval based quasi-Newton algorithm. The formulation used is inspired by [9]. This method encloses the solutions of an equation $C(a) = 0$. It uses a **contractor** $[K]$, an operator which takes an input box and outputs a smaller box when possible. This contractor is built using $[\partial C/\partial a]([a])$, an enclosure of the first derivative of C on $[a]$ in the form of an interval matrix.

Let mid $([a])$ the middle point of $[a]$, $[K]$ is defined as follows:

$$[K]([a]) = \text{mid}\,([a]) - M \cdot C(\text{mid}\,([a])) + \left(I_d - M \cdot \left[\frac{\partial C}{\partial a} \right]([a]) \right) \cdot ([a] - \text{mid}\,([a])), \quad (7)$$

where M is an invertible real matrix, typically $M = \text{mid}\,([\partial C/\partial a]([a]))^{-1}$.

If $a \in [a]$ is such that $C(a) = 0$, then $a \in [K]([a])$. If $[K]([a])$ is contained in the interior of $[a]$ then there is a unique solution in $[a]$ or none.

To sum up, Krawczyk contractor is the function: $[a] \mapsto [a] \cap [K]([a])$, Krawczyk's method is made of the following steps: i) Initialize $[a]$ with the search area. ii) Repeat $[a] \leftarrow [a] \cap [K]([a])$ until convergence. Its output is a box containing all solution of $F(a) = 0$ in $[a]$.

3.2.3 Validated simulation

A set membership ODE is considered:

$$\begin{cases} \dot{x}(t) & \in [f](x,t) \\ x(t_0) & \in [x_0]. \end{cases}$$

Validated simulation encloses every solution of this system in a sequence of boxes (it encloses the solution for any realization of the initial state $x_0 \in [x_0]$ with any realization of the dynamics $f(x) \in [f(x)]$).

To that end, the time range $[0, T]$ is discretized in $(t_i)_{i \in 0..N}$, $t_0 = 0, t_N = T$. Starting with an enclosure $[y_i]$ of the systems at time t_i, an enclosure of the system on the whole time range $[t_i, t_{i+1}]$ (called a Picard box) is built. Then an enclosure of $y(t_{i+1})$ is computed. In [1], interval Runge-Kutta method are used coupled with inflating terms that enclose the truncation error of the method. Indeed, if the dynamics are sufficiently differentiable, the truncation error can be bounded by evaluating the Lagrange remainder of the difference between the Taylor series of the actual solution and the Taylor series of the Runge-Kutta approximation. For instance, if dynamics are four time differentiable, the truncation error of Runge-Kutta 4 may be enclosed.

The output of validated simulation resembles Figure 10, with the plain boxes being state boxes and the dashed boxes being Picard boxes.

Validated simulation is akin to simulating multiple systems between two common time stamps. A time switch is a time horizon shared by all systems, hence all systems can be simulated by doing a first simulation up until the switch and another simulation starting right after the switch. Contrarily, a variable time horizons or a state dependent transitions differs from one system to another. As a consequence, there is no shared time stamp to use as a duration for the simulation, which makes validated simulation challenging.

4   Computation of three informative enclosures

In this section, the main contribution is presented. An interval valued OCP solver is proposed and is used to build enclosures of the trajectory of a concrete system using the OCP as a controller.

4.1   Interval OCP solver based on the PMP

As per our assumption, there are multiple possible initial states and multiple possible dynamics which means infinitely many possible solutions. Hence the following set membership method to solve the OCP is proposed.

Assumption   The presented method requires the dynamics g of the IVP (4) to be at least twice differentiable and to be $k + 1$ times differentiable if a validated simulation method of order k is used. Indeed, Section 3.1.3 states that if g is twice differentiable, then a first derivative of the IVP (5) may be computed with ODE (6). As ODE (6) involves $\partial g / \partial x$, if g is $k + 1$ times differentiable, then ODE (6) is k times differentiable and a method of order k may be used. This assumptions hold if f, μ and ℓ are infinitely differentiable, as will be the case in Section 5.

Method Based on the statements of Section 3.1.2, at a given time τ any couple (y_τ, p_τ) that satisfies the optimality condition $C(y_\tau, p_\tau) = 0$ is considered optimal. They will be referred as optimal (state, co-state) couples.

For a given enclosure of the state $[y_\tau]$, a Krawczyk operator of the function $p_\tau \to C(y_\tau, p_\tau)$ is built using Formula (7):

$$
\begin{aligned}
K([p_\tau]) \ &= \operatorname{mid}([p_\tau]) + M \cdot [C]([y_\tau], \operatorname{mid}([p_\tau])) \\
&+ \left(I_d - M \cdot \left[\frac{\partial C}{\partial p} \right]([y_\tau], [p_\tau]) \right) \cdot ([p_\tau] - \operatorname{mid}([p_\tau])),
\end{aligned}
\tag{8}
$$

where:

- $M = \operatorname{mid}\left(\left[\dfrac{\partial C}{\partial p} \right]([y_\tau], [p_\tau]) \right)^{-1}$,

- $[C]([y_\tau], [p_\tau]_m) = [\phi]^p_{\tau,T}([y_\tau], [p_\tau]_m) - \left[\dfrac{\partial \Psi}{\partial y} \right] \left([\phi]^y_{\tau,T}([y_\tau], [p_\tau]_m) \right)$,

- $\left[\dfrac{\partial C}{\partial p} \right]([y_\tau], [p_\tau]) = \left[R^p_{\tau,T} \right]([y_\tau], [p_\tau]) - \left[\dfrac{\partial^2 \Psi}{\partial^2 y} \right]([y_\tau], [p_\tau]) \cdot \left[R^y_{\tau,T} \right]([y_\tau], [p_\tau])$.

The enclosure $\left[R^y_{\tau,T} \right]([y_\tau], [p_\tau])$ and $\left[R^p_{\tau,T} \right]([y_\tau], [p_\tau])$ of the resolvents can be computed with an analytic formula if it exists, or by integrating ODE (9), which is an enclosure of ODE (6).

$$
\begin{cases}
\dot{y}(t) \ \in [g^y](y(t), p(t), [\xi]) \\
\dot{p}(t) \ \in [g^p](y(t), p(t), [\xi]) \\
\dot{R}^y_{\tau,t} \ \in \left[\dfrac{\partial g^y}{\partial y} \right](y, p, [\xi]) \cdot R^y_{\tau,T} + \left[\dfrac{\partial g^y}{\partial p} \right](y, p, [\xi]) \cdot R^p_{\tau,T} \\
\dot{R}^p_{\tau,t} \ \in \left[\dfrac{\partial g^p}{\partial y} \right](y, p, [\xi]) \cdot R^y_{\tau,T} + \left[\dfrac{\partial g^p}{\partial p} \right](y, p, [\xi]) \cdot R^p_{\tau,T} \\
y(\tau) \ \in [y_\tau] \\
p(\tau) \ \in [p_\tau] \\
R^y_{\tau,\tau} \ = 0 \\
R^p_{\tau,\tau} \ = I_d,
\end{cases}
\tag{9}
$$

The enclosure $[\phi]_{t_0,t_1}([y_0], \operatorname{mid}([p_0]))$ of the flow can be computed by integrating:

$$
\begin{cases}
\dot{y}(t) \ \in [g^y](y(t), p(t), [\xi]) \\
\dot{p}(t) \ \in [g^p](y(t), p(t), [\xi]) \\
y(\tau) \ \in [y_\tau] \\
p(\tau) \ = \operatorname{mid}([p_\tau]).
\end{cases}
\tag{10}
$$

Our OCP solver is an implementation of Krawczyk's method. Algorithm 1 computes the Krawczyk operator, either with analytic formulae or by integrating System (9) and System (10). Algorithm 2 is a Krawczyk's method that outputs $\emptyset$ if there is no solution to $C((y_\tau, p_\tau) = 0$. The convergence criterion is Hausdorff distance, which is a distance between sets particularly adapted to boxes [9].

Algorithm 1 Computes $[K]([p_\tau])$ for given $[y_\tau]$ and $[p_\tau]$

Require: $[y_\tau]$ and $[p_\tau]$

  Compute $\left[R_{\tau,T}^y\right][y_\tau], [p_\tau]$ and $\left[R_{\tau,T}^p\right]([y_\tau], [p_\tau])$

  Compute $[\phi]_{\tau,T}([y_\tau], \mathrm{mid}(p_\tau))$

  Compute $[K]([p_\tau])$ using Formula (8)

Algorithm 2 Computes a thin enclosure $([p_\tau])_k$ of the optimal co-states corresponding to a given state enclosure $[y_\tau]$

Require: An initial searching area $([p_\tau])_0$ and $[y_\tau]$

  $([p_\tau])_k \leftarrow ([p_\tau])_0$

  Compute $[K](([p_\tau])_k)$ using algorithm 1

  while $D_{\mathrm{Hausdorff}}(([p_\tau])_k, ([p_\tau])_k \cap [K](([p_\tau])_k)) > \text{precision}$ **do**

    if $0 \notin C(([y_\tau], [p_\tau]))$ **then**

      Return $\emptyset$

    else

      $([p_\tau])_k \leftarrow [K](([p_\tau])_k) \cap ([p_\tau])_k$

      Compute $[K]([p])$ using algorithm 1

    end if

  end while

Algorithm 2 needs to be initialized with a box $([p_\tau])_0$. This initial box depends on the type of enclosure built. As explained in Section 4.3, the anticipative, open-loop and closed-loop enclosures are computed using Algorithm 2 with different initialization.

4.2 Three enclosures to analyze the problem

For each realization of the initial condition parameters, there are an optimal control and a trajectory that are solution of the OCP. As a consequence, an OCP with interval uncertainties defines infinitely many controls and trajectories.

Similarly to many algorithms in the interval arithmetic literature that search a thin box containing all possible solutions of a problem, the anticipative enclosure contains all trajectories that are solution of an OCP.

Anticipative enclosure: For any parameter function $\xi(.) : [0, T] \to [\xi]$ and any initial state $y_0 \in [y_0]$, the anticipative enclosure contains the trajectory:

$$\begin{cases} \dot{y}(t) = f(y(t), u(t), \xi(t)) \\ y(0) = y_0, \end{cases}$$

with $u(.)$ the solution of the OCP:

$$\min_{u(.) \in \mathcal{U}} \int_0^T \ell(y(t), u(t))dt + \Psi(y(T)) \text{ s.t. } \begin{cases} \dot{y}(t) = f(y(t), u(t), \xi(t)) \\ y(0) = y_0. \end{cases}$$

$\xi(.)$ corresponds to both the actual realization of the parameters and the estimation made by the system. It is an ideal case in which the actual parameters are known with perfect accuracy. As such, this enclosure gives little information on the trajectory of a concrete system. In practice, parameter uncertainties will cause the system to deviate from its optimal trajectory and exit this enclosure.

The following two enclosures are proposed so as to inform on the behavior of the system in practical cases. They are meant to enclose the trajectory of a concrete system that computes its control by solving the OCP with an inaccurate model.

Open loop enclosure : It encloses a system which follows blindly an initial solution that was computed with inaccurate parameters. For any couple of parameters functions $\xi(.), \hat{\xi}(.) : [0, T] \to [\xi]$, and any initial state $y_0 \in [y_0]$, the open loop enclosure contains the trajectory:

$$\begin{cases} \dot{y}(t) = f(y(t), \hat{u}(t), \xi(t)) \\ y(0) = y_0, \end{cases}$$

with $\hat{u}(.)$ the solution of the OCP:

$$\min_{\hat{u}(.) \in \mathcal{U}} \int_0^T \ell(\hat{y}(t), \hat{u}(t))dt + \Psi(\hat{y}(T)) \text{ s.t. } \begin{cases} \dot{\hat{y}}(t) = f(\hat{y}(t), \hat{u}(t), \hat{\xi}(.)(t)) \\ \hat{y}(0) = y_0. \end{cases}$$

$\xi(.)$ corresponds to the actual realization of the parameters, which is unknown. Hence the system has an inaccurate estimation $\hat{\xi}(.)$. This is close to worst case analysis, as the worst case being the application of a control tailored for an extreme scenario to the opposite extreme scenario. This enclosure emphasizes the worst under or over-shooting possible.

It is possible to make a less pessimistic enclosure by taking into account the fact that the system can correct its trajectory using sensor data.

Closed loop enclosure : It encloses a system that uses a perfect measure of its state to recompute the solution of the OCP online (but with inaccurate parameters). Consider a list of recomputation time $(\tau_k)_{k \in 0..K}$, $0 = \tau_0 < \tau_1.. < \tau_K = T$. For any couple of parameters functions $\xi(.), \hat{\xi}(.) : [0, T] \to [\xi]$, and any initial states $y_0 \in [y_0]$, the closed loop enclosure contains the trajectory of the piecewise-defined system:

$$\begin{cases} \dot{y}(t) = f(y(t), \hat{u}_k(t), \xi(t)), t), \forall t \in [\tau_k, \tau_{k+1}] \\ y(0) = y_0, \end{cases}$$

with $\hat{u}_k(.)$ the solution of the OCP:

$$\min_{\hat{u}(.) \in \mathcal{U}} \int_{\tau_k}^T \ell(\hat{y}(t), \hat{u}(t))dt + \Psi(\hat{y}(T)) \text{ such that } \begin{cases} \dot{\hat{y}}(t) = f(\hat{y}(t), \hat{u}(t), \hat{\xi}(t)) \\ \hat{y}(\tau_k) = y(\tau_k). \end{cases}$$

The overall control is made of pieces $\hat{u}_k(.)$ that are computed with an accurate measurement of the state $y(\tau_k)$. As in the open-loop enclosure, $\xi(.)$ corresponds to the actual realization of the parameters, which is unknown, and $\hat{\xi}(.)$ is an inaccurate estimation. This encloses the actual operation of a system with an optimal control regulator. The system does not have access to the value of the parameters, but it compensates using measures of its state.

In this paper, a finite set of recomputation time $(\tau_k)_{k \in 0..K}$ is considered. Under this assumption the closed loop enclosure is a sequence of open loop enclosures. Future works will investigate scenarios in which the control is recomputed at every time, which causes the closed loop to have a more unique geometry.

The computation of these enclosures is presented in Section 4.3.

4.3   The underlying geometry of these enclosures

A simpler system is considered to emphasize the geometry of the enclosures presented in Section 4.2. The following figures are depiction of the OCP:

$$\min_{u(.)\in\mathcal{U}} \int_\tau^T \frac{u^2}{2}dt + K\frac{(y(T)-y_T)^2}{2} \text{ s.t. } \begin{cases} \dot{y}(t) = \xi(t)u, \\ y(\tau) = y_\tau, \\ T \text{ is fixed.} \end{cases}$$

with $\forall t, \xi(t) \in [\xi]$, an uncertain parameter. As the state is one dimensional, state and co-state can be drawn on a single graph to showcase the important sets.
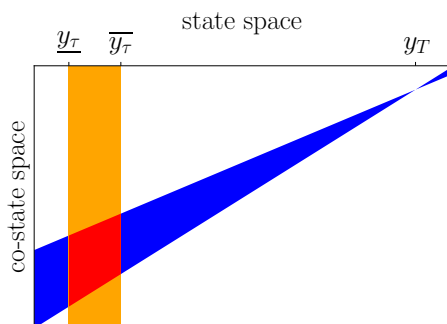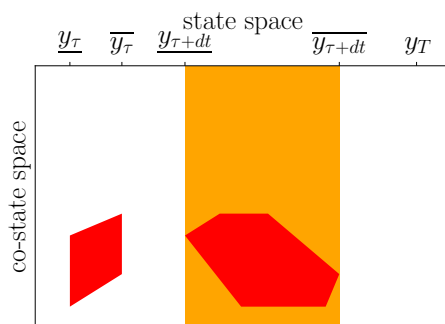

Figure 3: Geometric resolution of the OCP at time τ.

Figure 4: Simulation of the System (4) from time τ to time $\tau + dt$.

Open loop enclosure.   As seen in Section 3.1.2, in the real case, the optimal control and trajectory are found by first solving the IVP (5) then integrating System (4).

To build the open-loop enclosure, an enclosure of the solution of the IVP is computed, then the evolution of System (4) is enclosed. If the initial state is such that $y_\tau \in [y_\tau]$, then the (state, co-state) couples can be anywhere in $[y_\tau] \times \mathbb{R}^n$, the orange vertical strip on Figure 3. The (state, co-state) couples that are solution

of the IVP (5) are enclosed in the blue cone. The solutions of the OCP lie in the intersection (in red) of these two sets. If Algorithm 2 is initialized with a box $([p_\tau])_0$ that is large enough to contain this intersection, then it computes a box enclosure of this intersection.

Once $[p_\tau]$ has been computed, the evolution of System (4) is enclosed using validated simulation during a time range dt. This integration of System (4) is shown on Figure 4. This yields a set of possible state, co-state) couples at time $\tau + dt$, which in turn gives an enclosure of the possible states $[y_{\tau+dt}]$. By going on until the final time T, the open-loop enclosure is computed.
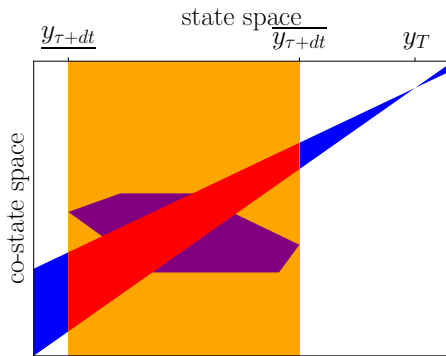

Figure 5: Closed-loop: recomputation of the optimal co-states at time $\tau + dt$.
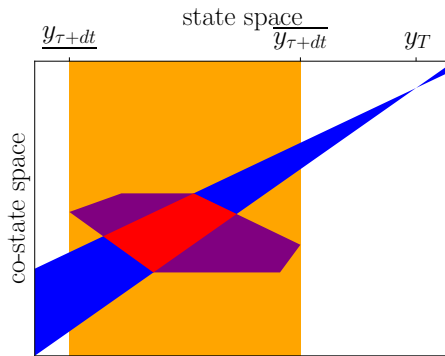
Figure 6: Anticipative: refining of the optimal couples at time $\tau + dt$.

Closed loop enclosure. If the system recomputes its control at $\tau + dt$ using accurate measurements from the sensors, the OCP is solved again with an initial state in $[y_{\tau+dt}]$. This is illustrated in Figure 5, which is similar to Figure 4. An important point is that the red set of recomputed (state, co-state) couples is not contained in the purple set obtained by integrating System (4) from a prior step. This is due to the fact that the prior co-states were computed with the wrong parameters. Replacing them is a correction. Hence, these trajectories do not respect the equations of the PMP: applying a control that is optimal for a faulty model results in a non optimal trajectory.

Anticipative enclosure. To characterize the actual optimal trajectories, the following criterion is applied. If a trajectory is optimal from A to B and C is an intermediate state of that trajectory, then the trajectory is optimal from A to C and from C to B. If A is the state at time τ, B is the state at time T and C is the state at time $\tau + dt$, then its (state, co-state) couple at time $\tau + dt$ satisfies two properties.

- Optimality condition between A and C: the couple lies in the purple polyhedron on Figure 6, which is the set obtained by integrating System (4) from

step τ.

- Optimality condition between C and B: the couple is in the blue cone on Figure 6, which is the set of solutions of the IVP (4) with the initial time $\tau + dt$.

By taking the intersection of these two sets, the set of optimal (state, co-state) couples is refined. This refinement can be done by initializing Algorithm 2 with $([p_{\tau+dt}])_0$ the projection of the purple polyhedron on co-state space.

5   Experimentations

The computation of the three enclosures is showcased on a double integrator with quadratic cost.
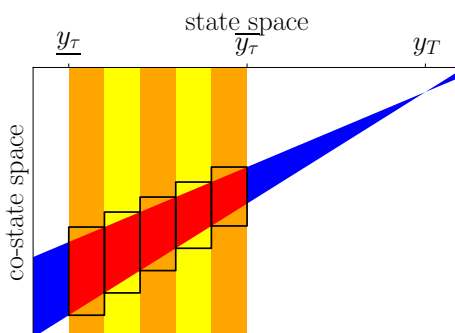
5.1   A naive implementation

Figure 7: Paving the state enclosure to get a thinner approximation.

Since Krawczyk's Algorithm 2 operates with boxes, the state representation has to be a box or a set of boxes. Representing the set of possible states by a box is computationally efficient but causes crippling wrapping effects. Instead, we propose a naive implementation that encloses the state at time t in a tiled bounding box, such as in Figure 2.

To compute an enclosure of state at time $t+dt$, for each tile that might contain a possible state, the OCP is solved (see Figure 7) and then the system is integrated. A first round of OCP resolution and validated integration is done to compute a bounding box of the state at time $t + dt$, then this bounding box is tiled uniformly and a second round finds which tiles might contain a solution.

Each tile requires several calls to Algorithms 1 and 2 , and since the complexity of the tiling is exponential in the dimension, this naive method is not adapted to complicated systems.

5.2 Numerical results on a double integrator

Consider a double integrator in a uniform gravity field and with a reactor thrust. It is subject to a quadratic continuous cost and a quadratic penalization on the final state.

$$\min \int_0^T \frac{||u||^2}{2} dt + K_v \frac{||v(T)||^2}{2} + K_r \frac{||r(T) - r_T||^2}{2} \quad \text{such that} \quad \begin{cases} \dot{r}(t) = v \\ \dot{v}(t) = -\frac{C}{m} u - Ge_2 \\ r(0) = r_0, \quad v(0) = v_0, \end{cases} \quad (11)$$

where G is the normalized gravity field, e_2 is the unit vector of the vertical axis, C is the maximum thrust and m is the mass of the system, which is assumed to be constant.

By using the method in Section 3.1.2, this OCP turns into the following two point boundary value problem:

$$\begin{cases} \dot{r}(t) = v \\ \dot{v}(t) = -p_v \left(\frac{C}{m}\right)^2 - Ge_2 \\ \dot{p}_r(t) = 0 \\ \dot{p}_v(t) = -p_r \end{cases} \quad (12)$$
$$r(0) = r_0, \quad v(0) = v_0$$
$$p_r(T) - K_r r(T) = 0, \quad p_v(T) - K_v v(T) = 0.$$

The values used are inspired by the take-off problem in [4]. The parameter nominal values are the same as in [4]. The initial position of the system corresponds to the final position of the take-off mission. The initial velocity has been chosen to be coherent with a re-entry mission.

The solution of the nominal case is presented on Figure 8.

The initial position has an uncertainty of around 5 km, which is about 2% of the span of the trajectory. The initial speed has a relative uncertainty of 1.7% and the parameter ratio C/m has a relative uncertainty of 0.2%. However, the mission duration is greater than the optimal duration so as to emphasize the differences between the enclosures. The longer the mission, the more the system has to fight gravity, the more uncertainties are accumulated along the vertical axis.

Figures 9, 10 and 11 depict enclosures of the double integrator computed using Algorithm 2 coupled with a tiling of the state enclosure (as shown on Figure 7). The initialization of Algorithm 2 depends on the type of enclosure, as explained in Section 4.3.

Since System (12) is linear time invariant with a strictly triangular matrix, there are analytic formulae for the flow and the resolvents. We use these analytic formulae in Algorithm 1. We have also simulated Systems (9) and (10) with DynIbex [1]: a C++ library for validated simulation. The precision of the results of the simulations matches those of the analytic formulae. However, these simulations drastically increase computation time, which in turn significantly restrict the precision of the tiling. As a consequence, Figures 9, 10 and 11 were done with analytic formulae
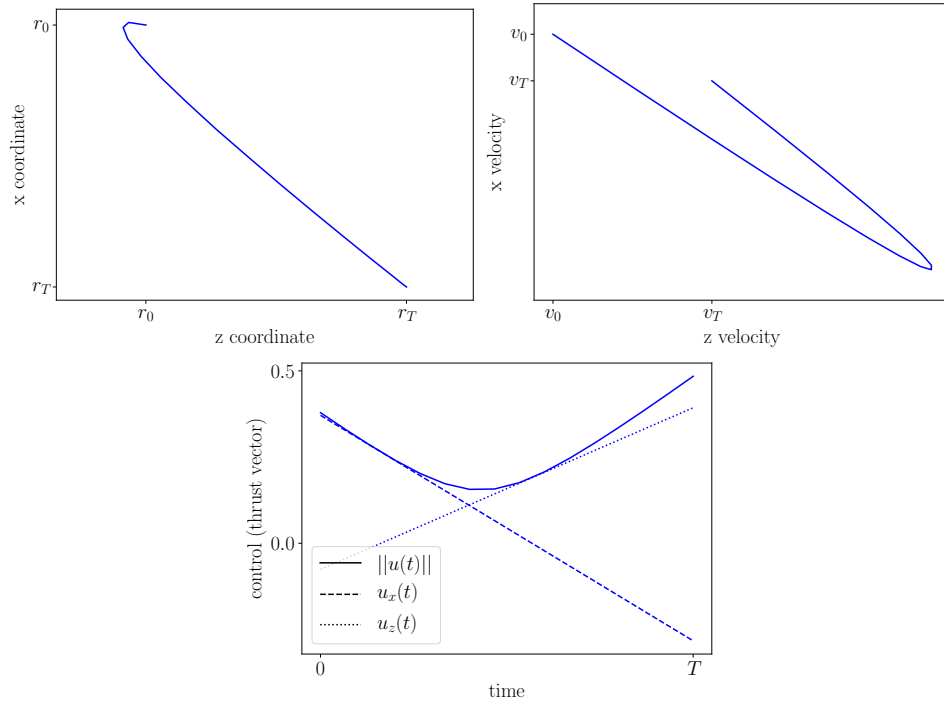
Figure 8: Optimal trajectory, velocity space trajectory and control for the a double integrator with quadratic cost. Unlike Goddard's problem on Figure 1, the control changes continuously.

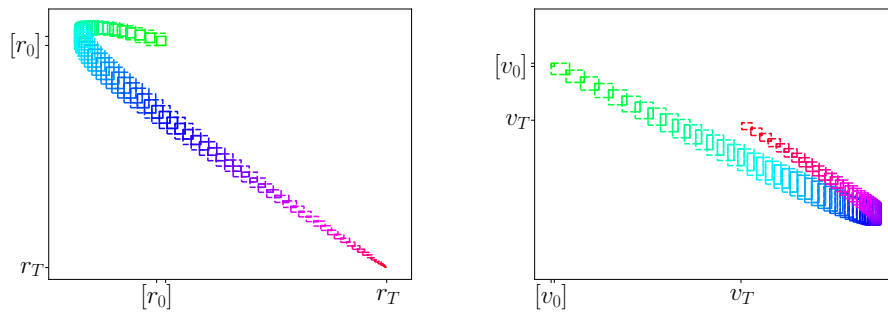rather than validated simulation.

Figure 9: Anticipative enclosure of position and velocity.

The anticipative enclosure on Figure 9 starts wide because of initial uncertainty but gets thinner as all the system converge to the target. As a result, the final box is very thin.

Contrarily, the open-loop enclosure on Figure 10 becomes wider over time and the final box is very big. Systems over-shoot or under-shoot the target because of their lack of correction.

Lastly, the closed loop enclosure on Figure 11 is somewhere in between the two other enclosures. These systems deviate from their optimal trajectory but correct it, leading to a bulge on the middle and end of the trajectory that gets abruptly smaller at the end.
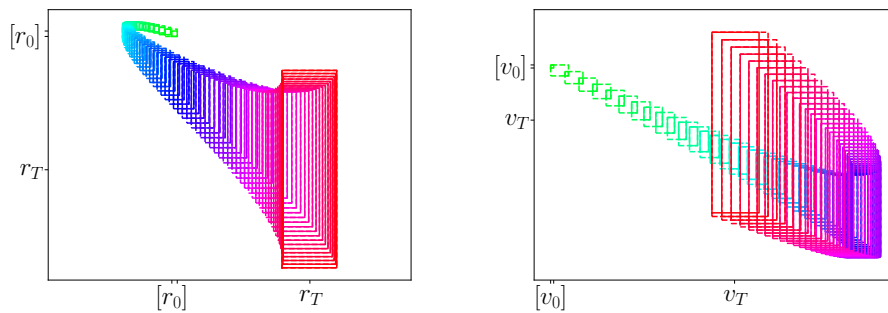

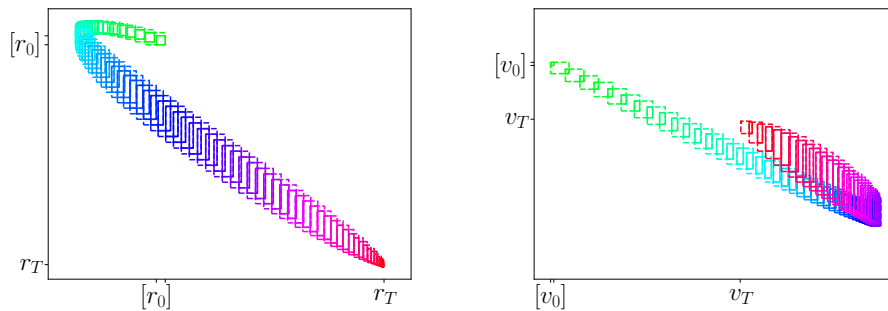Figure 10: Open loop enclosure of position and velocity.

Figure 11: Closed loop enclosure of position and velocity.

6 Discussion

In this section, we propose some uses of the enclosures, then list what needs to be done to address more complex problems.

6.1   Risk assessment and indicating the relevance of an OCP

These enclosures are similar to those developed in [2]. As such, they can have the same application. Consider the example of final constraints: the final state must be in a safe zone. The final position must be in the recuperation platform and the final velocity must be sufficiently small for the system to land. A similar reasoning can be held for a state constraint, or other requirements of a mission.

If the final state box of the closed loop enclosure (Figure 11) is contained in the safe zone, then a system using this OCP in closed loop is guaranteed to satisfy the final constraint. Similarly, it can be guaranteed that applying the solution of the OCP in open loop will satisfy the mission (although this is clearly not the case on Figure 10). Both are strong assessment of robustness. The cost can be enclosed as well, which gives a worst performance index.

When using a floating-point control, the anticipative enclosure will provide an upper bound to the distance to the actual optimal control. As a consequence, this provide a criterion to discuss the validity of a floating-point control. To minimize this criterion, the middle point of the control box of the anticipative enclosure can be used as a control.

These enclosures could be used to design the model or the concrete system. For instance, if the final enclosure of the velocity of the anticipative enclosure (Figure 9) is not entirely contained in the safe zone, that means that there might be a realization of the initial states and the parameters such that the system will crash by following the optimal trajectory defined by the OCP.

The existence of such a realization is not guaranteed as those enclosures are conservative, nevertheless, for critical systems, the mere possibility of the existence of a failure may require the OCP to be changed.

In Problem (11), this could mean increasing the penalization coefficient K_r and K_v, thus making the launch more costly but decreasing the uncertainty on the final state and maybe void the possibility of failure.

Contrarily, if the open-loop or closed-loop enclosures final boxes respect the final constraint by a significant margin, then the penalization could be decreased to achieve better performances on other criteria. Alternatively, the uncertainty could be altered. One could assess the maximum acceptable magnitude of the uncertainties and use it as specification for the creation of an engine or sensors. Lastly, if the open loop enclosure respects all the constraints, then it might not be necessary to elaborate a closed loop regulator.

Of course, these statements hold true as long as the model with uncertainties is accurate. If parameter uncertainties have been under-estimated or if the model has neglected a phenomenon, then there is no guarantee that the system will remain in the enclosures.

6.2   Generalizing to Goddard's problem

The double integrator with quadratic cost was considered because it is used to initialize a continuation method to solve Goddard's problem [4, 5].

In a similar manner to a continuation method, there are two steps to bridge the gap between the double integrator and Goddard's problem. First non linear systems have to be addressed. We need to simulate non linear problems and their resolvent, which is costly. Then state dependent transitions and variable time horizons will need to be modeled and enclosed. This means being able to enclose the evolution of a hybrid system, if possible in a manner that does not induce a significant over-approximation. This also makes the computation of the resolvent more challenging.

In addition to that, Goddard's representation of a spacecraft is in dimension seven. As a consequence, tiling may no longer be used to improve the precision of Algorithm 2. An entirely new method may be needed.

7 Conclusion

In this paper, a resolution tool was proposed for optimal control problems with embedded uncertainties. This resolution tool is used to compute three enclosures, an anticipative enclosure containing all possible solutions of the optimal control problem, and open-loop and closed loop enclosures that bounds the trajectory of a concrete system that uses this problem as a controller. These enclosures can show the deviation caused by the uncertainties, identify the critical zone that could be crossed or more generally assess the relevance of a given optimal control problem.

In later works, we will try to improve our method until it can reliably compute these enclosures for a problem as complex as Goddard's.

Acknowledgment

References

[1] Alexandre dit Sandretto, J. and Chapoutot, A. Validated explicit and implicit Runge–Kutta methods. *Reliable Computing*, 22(1):79–103, Jul 2016.

[2] Althoff, M. *Reachability Analysis and its Application to the Safety Assessment of Autonomous Cars.* PhD thesis, Technische Universität München, 2010.

[3] Blackmore, L. Autonomous precision landing of space rockets. *The Bridge*, 4(46):15–20, 01 2016.

[4] Bonnans, F., Martinon, P., and Trélat, E. Singular arcs in the generalized goddard's problem. *Journal of Optimization Theory and Applications*, 139(2):439–461, 2008. DOI: `10.1007/s10957-008-9387-1`.

[5] Brendel, E., Hérissé, B., and Bourgeois, E. Optimal guidance for Toss Back concepts of Reusable Launch Vehicles. In *EUCASS*, 2019.

[6] Butcher, J. C. Coefficients for the study of runge-kutta integration processes. *Journal of the Australian Mathematical Society*, 3(2):185–201, 1963. DOI: `10.1017/S1446788700027932`.

[7] Caillau, J.-B., Cerf, M., Sassi, A., Trélat, E., and Zidani, H. Solving chance constrained optimal control problems in aerospace via kernel density estimation. *Optimal Control, Applications and Methods*, 39(5):1818–1832, 2018. DOI: `10.1002/oca.2445`.

[8] Coron, J.M. and American Mathematical Society. *Control and Nonlinearity.* Mathematical surveys and monographs. American Mathematical Society, 2007.

[9] Jaulin, L., Kieffer, M., Didrit, O., and Walter, E. *Applied Interval Analysis: With Examples in Parameter and State Estimation, Robust Control and Robotics.* Springer London, DOI: `10.1007/978-1-4471-0249-6`, 2001.

[10] Moore, Ramon E. *Interval analysis*, volume 4. Prentice-Hall Englewood Cliffs, 1966.

[11] Rauh, A. and Hofer, E. P. Interval methods for optimal control. In Buttazzo, G. and Frediani, A., editors, *Variational Analysis and Aerospace Engineering*, chapter 22, pages 397–418. Springer New York, New York, NY, 2009, 10.1007/978-0-387-95857-6_22.

[12] Rauh, A., Minisini, J., and Hofer, E. P. Interval techniques for design of optimal and robust control strategies. In *12th GAMM - IMACS International Symposium on Scientific Computing, Computer Arithmetic and Validated Numerics (SCAN 2006), Duisburg,*, 2006. DOI: `10.1109/SCAN.2006.27`.

[13] Trélat, E. *Contrôle optimal : théorie et applications.* Vuibert, 2005.

[14] Trélat, E. Optimal Control and Applications to Aerospace: Some Results and Challenges. *JOTA*, 154(3), 2012.