

Optimal Switching Instants for the Control of Hybrid Systems*

Olivier Mullier^a, Julien Alexandre dit Sandretto^a,
and Alexandre Chapoutot^a

Abstract

The problem of determining the optimal switching instants for the control of hybrid systems under reachability constraints is considered. This optimization problem is cast into an interval global optimization problem with differential constraints, where validated simulation techniques and a dynamic time meshing are used for the computation of its solution. The approach is applied on two examples, one being the well-known example of the Goddard's problem where a rocket has to reach a given altitude while consuming the smallest amount of fuel, the second one considers a system with a higher dimension and a more complex optimization problem.

Keywords: hybrid systems, reachability, optimization, validated numerical integration, set membership computation, Goddard problem

1 Introduction

Since the seminal work from Witsenhausen [20], many real-world processes (*e.g.*, chemical processes, manufacturing processes, *etc.*) have been modeled using hybrid dynamical systems in which both continuous and discrete dynamics occur. These systems can be represented by continuous dynamical systems with discontinuities to model discrete events (see *e.g.* [4]). Usually two kinds of discrete events are considered. The simplest kind of events is the *timed event* which models the switching between two (or more) dynamical systems at particular given time instants. The other kind of events is the *state event* which models the switching between two (or more) dynamical systems as a function of a particular value of the state variables. In this paper, the simplest kind of events, named time event, is considered to model discrete phenomena in hybrid dynamical systems.

*This research benefited from the support of the “Chair Complex Systems Engineering - Ecole Polytechnique, THALES, DGA, FX, DASSAULT AVIATION, DCNS Research, ENSTA ParisTech, Télécom ParisTech, Fondation ParisTech and FDO ENSTA”. This research is also partially funded by DGA MRIS.

^aENSTA ParisTech, 828 bd des Maréchaux, 91762 Palaiseau Cedex, France, E-mail: {mullier,alexandre,chapoutot}@ensta.fr

In its full generality, each subsystem can be driven by a continuous output depending on the time but in our case, only ordinary differential equations will be used to model continuous-time dynamical systems. In consequence, those systems are modeled by *switched systems* consisting of an N -mode system and a switching function driving the subsystem to consider at each switching time [6].

The classical approach to synthesize control law for switched systems is to compute the switching function, that is finding the sequence of modes required to fulfil a given specification. For example, in [11], the sampling time of switched systems is fixed and the control synthesized has to find the sequence of modes to reach a given area of the state space while staying in a safety zone. In contrast to this, the current work addresses problems where the sequence of modes is given (N is fixed), *i.e.*, the global structure of the controller is known, and the problem is to find the time of the switching instants for which a given property is fulfilled. This problem appears in optimal control problems where some optimality criteria have to be achieved. It takes into account a cost function depending on the problem to be addressed which designs some energy consumption (for example fuel consumption for a car) along time (Lagrangian or running cost) and/or some time fixed consumption (terminal cost). It can be solved, for example, by applying Pontryagin's Maximum Principle [13].

The main goal of the work hereafter is to synthesize an optimal control law for switched systems where the sequence of modes is already given. In [6], the problem was tackled using a gradient descent algorithm thanks to the computation of the gradient of the cost function associated to the optimal control problem. In this paper, we provide a validated method to determine the optimal switching instants of a switched continuous-time system under reachability constraints where the objective function can follow a monotonicity property. The method is validated in the sense that it takes into account constraints that are proved to be satisfied. The search space consisting of the times of switch is traveled using a meshing of the time. It results in a sub-optimal solution for the problem with respect to the meshing which tends to become optimal when the meshing parameter ϵ tends to 0. We want a rigorous approach for the case of critical systems where a validated solution is mandatory. The method depicted here makes use of validated numerical integration methods and no longer requires to produce the gradient of the cost function except if the monotonicity of the cost function has to be checked in the improvement also discussed. This validation is made possible using interval analysis [15] which has already been successfully used in the design of optimal and robust controllers when no hybrid system occurs (see, *e.g.* [18]) or when taking into account state-dependent switchings between different friction and hysteresis models [19]. Using interval analysis [15], these methods are mainly based on Taylor series [14, 17] or on Runge-Kutta methods [1, 3] which consider affine arithmetic as well [5]. They allow to produce an outer-approximation of the solution of an *initial value problem (IVP) of ordinary differential equations (ODE)* where bounded uncertainties can be considered on the initial condition or on the parameters of the model designed by the ODE. The presence of bounded uncertainties implies that there is no longer only one trajectory of the IVP-ODE but a set of trajectories.

Runge-Kutta methods suit well with our problem since they are efficient using initial values and parameters as sets and their implementation is embedded into a constraint satisfaction problem framework [2]. Interval analysis has been already used in the control field, for example, in [10] which defines methods to synthesize control laws for linear dynamical systems, to generate safe and robust paths for robots.

This paper is presented as follows. The next section is dedicated to the assumptions and hypotheses we took into account to define the problem being handled. In Section 3 the method used to handle the optimal control problem is described. Section 4 contains the main results in this article. The method is validated on two examples, one using the Goddard problem and another one focusing on the case of a continuous cost function over time. They are described in Section 5. The last section concludes the article and gives some hints on future works.

2 Discussion about the considered system

In this section, the class of the considered system is discussed. Starting from a generic case, some hypotheses are added to focus on the particular class of optimal control problems which is considered in this article.

2.1 Continuous-time nonlinear systems

In this paper, we focus on continuous-time nonlinear dynamical systems described by Ordinary Differential Equations (ODEs)

$$\dot{x}(t) = f(x(t)). \quad (1)$$

More precisely, we are interested in a sub-class of hybrid systems where several ODEs alternate, represented by a sequence of subsystems $\{f_i : i \in [0, N - 1]\}$. A switched system is then an N -mode system consisting of N subsystems that can be described as follows:

$$(\mathcal{S}_i) \begin{cases} \dot{x} = f_i(x(t)) \\ x(t_i) = x_i \end{cases} \quad (2)$$

for all times $t \in [t_i, t_{i+1}]$, for all $i \in [0, N - 1]$. The switching between two ODEs is driven by a specific instant, as described in Figure 1. The global clock, unique, is continuously described by $\dot{t} = 1$ starting at 0. The switching instants t_i are defined such that $t_0 < t_1 < \dots < t_{N-1}$, then $t_i \neq t_j, \forall i \neq j$. The initial state of $\dot{x} = f_i(x)$ is $x(t_i)$, the final state of the previous dynamical system (\mathcal{S}_{i-1}). The flow is then continuous, contrarily to the generic case of hybrid systems. It is though not necessarily continuously differentiable (see Figure 2). This sequence of dynamical systems corresponds to the switching of a control law.

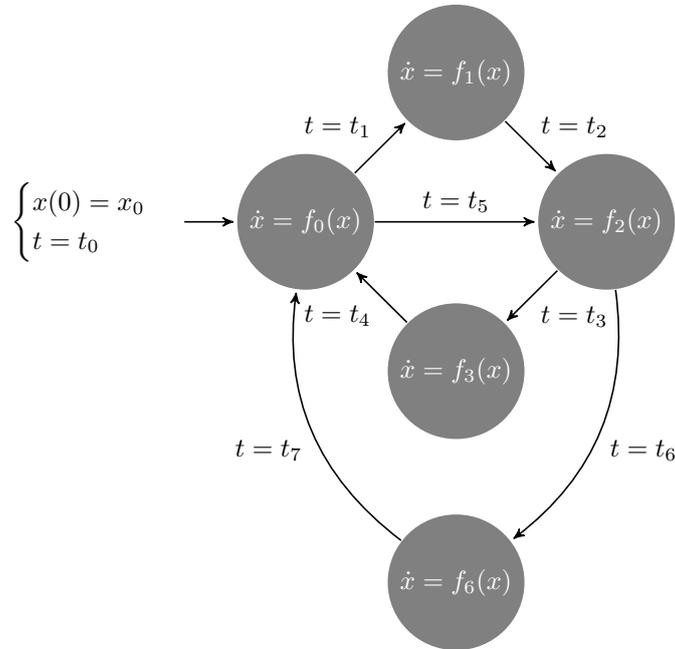


Figure 1: A particular case of hybrid system where some subsystems can occur several times in the sequence ($f_0 = f_4 = f_7$, $f_2 = f_5$).

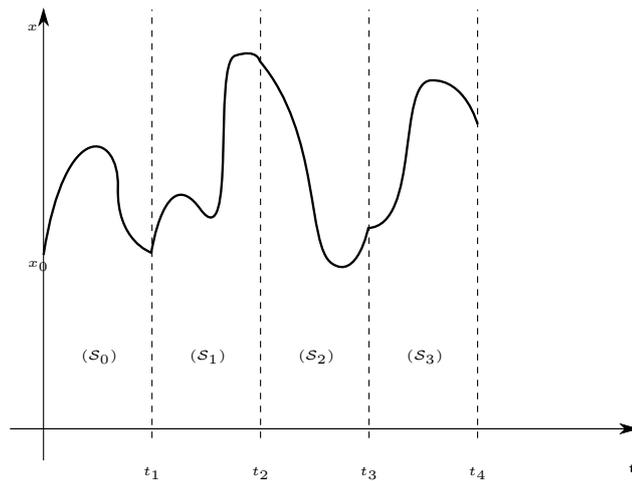


Figure 2: Representation of the trajectory of a 4-mode system as described by Equation (2).

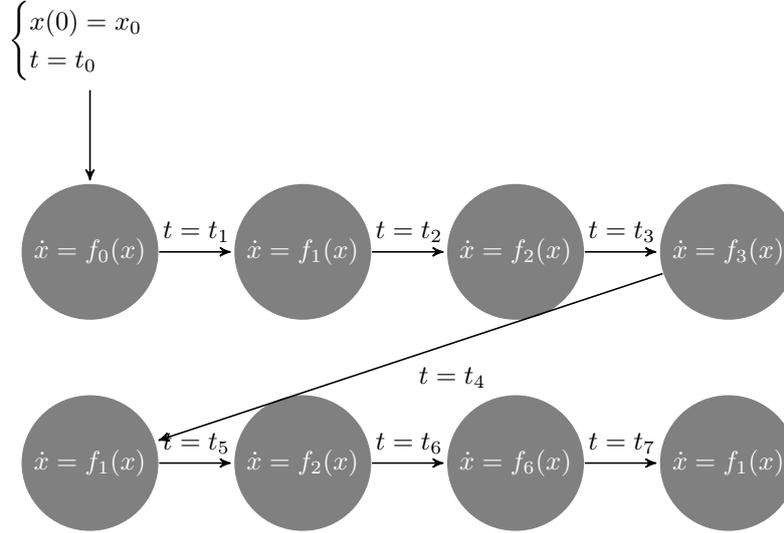


Figure 3: A time-variant system. It corresponds to a flattened view of the hybrid system described in Figure 1

2.2 A sub-class of switched systems

Following in the comparison, the system considered in Figure 1 can be seen as a sub-class of switched systems. Indeed, for a switched system, a rule can be defined to select the sequence of the modes. In our definition, the sequence is stated by the structure of the system. However, by choosing t_i sufficiently close to t_{i+1} : $t_{i+1} - t_i = \epsilon$ with $\epsilon \rightarrow 0$, the time spent in the mode i tends to zero and then, the mode has a negligible effect on the state. It is a way to select to go in another state (at the limit) than the follower. In this paper, this case is not considered since the time meshing forces to spend a minimum amount of time on each state.

2.3 A time-varying system

Finally, it is obvious that the system defined in Figure 1 can be flattened by duplicating some modes. The flattened view of this system is depicted in Figure 3. We can see this system as a time-variant one, described as follows:

$$\begin{cases} \dot{x} = f_0(x), & t_0 \leq t < t_1 \\ \dot{x} = f_1(x), & t_1 \leq t < t_2 \\ \vdots \\ \dot{x} = f_{N-1}(x), & t_{N-1} \leq t < t_N \end{cases} \quad (3)$$

In this paper, we focus on the N-mode switched systems, and more specifically on the optimal control problem involving the systems described in Equation (3). In the following section, we embed these systems onto the optimal control problem.

2.4 The optimal control problem

The optimal control problem aims at providing the set of optimal switching times $\{t_1, \dots, t_{N-1}\}$, we denote hereafter as $t_{[1..N-1]}$, such that a cost function is maximized. It writes as follows

$$\left[\begin{array}{ll} \max_{t_{[1..N-1]}} & J(t_{[1..N-1]}) = \int_0^{t_N} L(x(t))dt + F(x(t_N)) \quad (\text{cost function}) \\ \text{s. t.} & (\mathcal{S}_i), t \in [t_i, t_{i+1}], i \in [0, N-1] \quad (\text{dynamical constraints}) \\ & x(0) = x_0, \varphi(x(t)) \in \Phi, 0 < t \leq t_N \quad (\text{boundary conditions}) \end{array} \right] \quad (4)$$

where the *dynamical constraints* coupled with the boundary conditions represent the sequence of the subsystems; the *cost function* J is a pay-off function where $x(t)$ is the solution for the dynamical constraint; the functions $L : \mathbb{R}^n \rightarrow \mathbb{R}$ and $F : \mathbb{R}^n \rightarrow \mathbb{R}$ are respectively the running cost and the terminal cost. The *boundary conditions* represent constraints on the initial and following states with $\varphi : \mathbb{R}^n \rightarrow \mathbb{R}^m$ and Φ a set included in \mathbb{R}^m . Some uncertainties can occur in the dynamical constraints as well as the cost function. To handle them in a validated manner, we use the following set-membership computation.

3 Validated computation of the optimal control problem

We now describe how to treat the optimal control problem from Equation (4) with validated techniques. The set-membership computation is first introduced and a computation of the dynamical constraints and the cost function are described.

3.1 Set-membership computation

Set-membership computation is used whenever one wants to compute validated results. The most commonly used method is interval analysis [15]. It is designed to produce outer-approximations of computations involving sets of values in a sound manner. Hereafter, an interval is denoted by $[x] = [\underline{x}, \bar{x}]$ with $\underline{x} \leq \bar{x}$ and the set of intervals is $\mathbb{IR} = \{[x] = [\underline{x}, \bar{x}] \mid \underline{x}, \bar{x} \in \mathbb{R}, \underline{x} \leq \bar{x}\}$. The Cartesian product of n intervals in \mathbb{IR}^n is called a box.

The main result of interval analysis is its fundamental theorem stating that the evaluation of an arithmetic expression with intervals always leads to an outer-approximation of the resulting set of values for this expression whatever the values considered in the intervals. In order to deal with interval functions, an interval inclusion function also known as interval extension of a function can be defined.

Solving global optimization can be done using interval analysis coupled with constraint programming techniques. The dynamical constraints in this problem require to integrate differential equations using set membership computation.

3.2 The dynamical constraints

In the problem, we consider the dynamical constraints as ODEs. The solution of an ODE is usually computed using numerical integration since a closed form of the solution of an ODE is not computable in closed form. More precisely, the numerical integration methods [1, 17] are able to solve the IVP of a non-autonomous ODE defined by

$$\begin{cases} \dot{x} = f(x(t)) \\ x(0) \in \mathcal{X}_0 \subseteq \mathbb{R}^n. \end{cases} \quad (5)$$

The set \mathcal{X}_0 of initial conditions is used to model some (bounded) uncertainties. It then requires to bound these uncertainties along the time so validated numerical integration is considered. The goal for a validated (or rigorous) numerical integration method is then to compute the set of solutions of (5), *i.e.*, the set of possible solutions at a point of time t given the initial condition in the set of initial conditions \mathcal{X}_0 :

$$x(t; \mathcal{X}_0) = \{x(t; x_0) \mid x_0 \in \mathcal{X}_0\}. \quad (6)$$

Validated numerical integration schemes using the set-membership framework aim at producing the solution of the IVP-ODE that is the set defined in (6). It results in the computation of an outer approximation of $x(t; \mathcal{X}_0)$.

When considering the set of initial conditions as a box $[x_0]$, the use of the interval technique framework for the problem (5) enables the design of an inclusion function for the computation of an outer approximation of $x(t; [x_0])$ defined in Equation (6). We denote this inclusion function as $[x](t; [x_0])$. To build it, a sequence of time instants t_1, \dots, t_s such that $t_1 < \dots < t_s = t$ and a sequence of boxes $[x_1], \dots, [x_s]$ such that $x(t_{i+1}; [x_i]) \subseteq [x_{i+1}], \forall i \in [0, s-1]$ are computed. From $[x_i]$, computing the box $[x_{i+1}]$ is a classical 2-step method (see [17]):

Phase 1 Compute an a priori enclosure $[\tilde{x}_i]$ of the set

$$\{x(t_k; x_i) \mid t_k \in [t_i, t_{i+1}], x_i \in [x_i]\} \quad (7)$$

such that $x(t_k; [x_i])$ is guaranteed to exist and is unique,

Phase 2 Compute an enclosure of the solution $[x_{i+1}]$ at time t_{i+1} .

Each subsystem (\mathcal{S}_i) of the optimal control problem is then simulated using this technique.

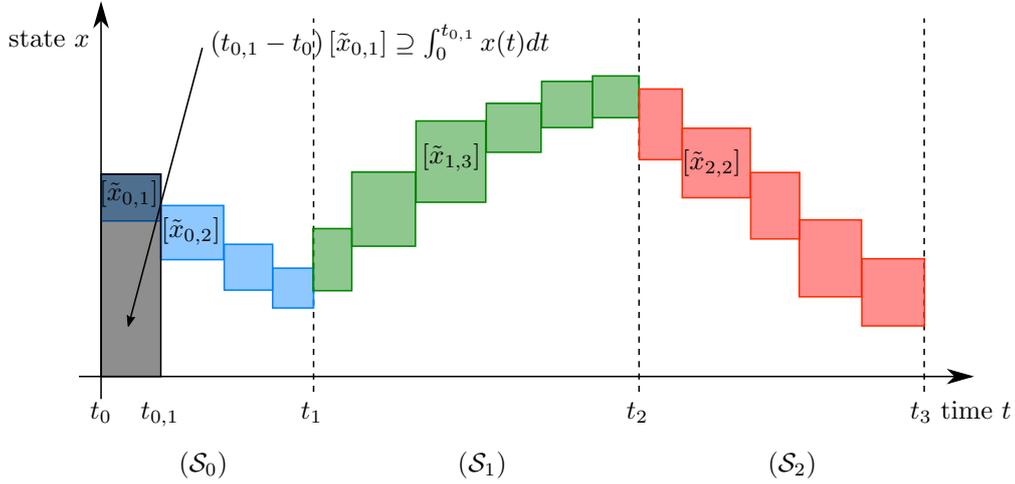


Figure 4: Example of a cost function computation from the boxes obtained by validated numerical integration.

3.3 The cost function

When the cost function is considered in its full generality, it consists of a running cost which is an integral and a terminal cost. They can be computed as follows using interval methods. First we decompose the integral along the different subsystems (\mathcal{S}_i) , $i = 0, \dots, N - 1$:

$$\begin{aligned} J(t_{[1..N-1]}) &= \int_0^{t_N} L(x(t))dt + F(x(t_N)) \\ &= \sum_{i=0}^{N-1} \int_{t_i}^{t_{i+1}} L(x(t))dt + F(x(t_N)). \end{aligned}$$

A validated numerical integration is then applied to produce an outer approximation of $x(t)$ along each time interval $[t_i, t_{i+1}]$. The integral is computed using the decomposition from the validated numerical integration method and by applying the rectangle method:

$$\begin{aligned} J(t_{[1..N-1]}) &= \sum_{i=0}^{N-1} \int_{t_i}^{t_{i+1}} L(x(t))dt + F(x(t_N)) \\ &\in \sum_{i=0}^{N-1} \sum_{k=0}^s (t_{i,k+1} - t_{i,k}) [L]([\tilde{x}_{i,k}]) + [F]([x_N]) \end{aligned}$$

with $[L]$ and $[F]$ the interval inclusions of the running cost L and the terminal cost F and $[\tilde{x}_{i,k}]$ the k -th box in phase 1 for the numerical integration of (\mathcal{S}_i) . Figure 4 represents an example of this computation. In this figure, the continuous cost $L(x)$

is considered as the identity function for visibility and each box is computed in Phase 1 of the validated numerical integration. Our problem is modeled using the following optimization problem

$$\left[\begin{array}{ll} \max_{t_{[1..N-1]}} & J(t_{[1..N-1]}) = \int_0^{t_N} L(x(t))dt + F(x(t_N)) & \text{(cost function)} \\ \text{s. t.} & (\mathcal{S}_i), i \in [0..N-1] & \text{(dynamical constraint)} \\ & h(x(\tau)) > 0, \tau \in [t_{N-1}, t_N] & \text{(reachability constraint)} \end{array} \right] \quad (8)$$

with the decision variables $t_1, \dots, t_{N-1} \in \mathbb{R}_+^{N-1}$ the search space for the different times; $J : \mathbb{R}^m \rightarrow \mathbb{R}$ the cost function, some constraints defined by the dynamical systems (\mathcal{S}_i) and the times t_i ; a reachability constraint using $h : \mathbb{R}^m \rightarrow \mathbb{R}$.

4 Approach

This section describes the approach we choose to address this optimal control problem. A first algorithm can be designed using a brute force approach. When considering the monotonicity of the cost function according to the time vector $t_{[1..N-1]}$, some improvement on the meshing can be made to speed up the resolution of the problem.

4.1 Brute force algorithm

The algorithm consists in a meshing of the time instants and the simulation of the N -mode system for each time instant from this meshing using Equation (8) to check if this sequence of switching times is a candidate for the solution of the problem. It is described in Algorithm 1.

The lines 2 to 6 are the time meshing. After the simulation in Line 7, Line 8 corresponds to the check of the reachability constraint. Then the algorithm checks if the evaluation of the cost function for this candidate provides a better solution for the optimization problem. Eventually, this algorithm provides the best solution according to the meshing parameter ϵ . The solution is then sub-optimal. As for the simulation, each subsystem is simulated using the validated integration method described in Section 3.2 to produce the final state at the time T_{\max} (Algorithm 2).

This algorithm consists in simulating in a guaranteed manner each subsystem (\mathcal{S}_i) , for a particular sequence of switching times (Line 7). This method is similar to the one used in [12]. Based on the obtained result under the form of a tube, the Algorithm 1 checks if the cost function is an optimal candidate at the end of the simulation or not (Line 9). If the reachability constraints are not violated, the switching instants are the current solution of the problem (Line 8).

The algorithm complexity can be expressed with respect to the number of switches $N-1$ and the meshing parameter p (with p the number of trials for one switch, *i.e.*, $p = t_N/\epsilon$), such that $\mathcal{O}(p^{N-1})$. In the next Section, monotonicity of the cost function is used to reduce this complexity.

Algorithm 1 finds the optimal switching times

Funct optimal_switch()

```

1:  $max \leftarrow 0$ 
2: for  $t_1 \leftarrow \epsilon$  to  $t_N - (N - 1)\epsilon$  do
3:    $\vdots$ 
4:   for  $t_i \leftarrow t_{i-1} + \epsilon$  to  $t_N - (N - i)\epsilon$  do
5:      $\vdots$ 
6:     for  $t_{N-1} \leftarrow t_{N-2} + \epsilon$  to  $t_N$  do
7:        $[x_N] \leftarrow \text{simu}(t_1, t_2, \dots, t_{N-1})$ 
8:       if  $[h]([x_N]) > 0$  then the reachability constraints are not violated
9:         if  $[J]((t_1, \dots, t_{N-1})) > max$  then ( $t_1, \dots, t_{N-1}$ ) is the new optimal candidate
10:            $max \leftarrow [J]((t_1, \dots, t_{N-1}))$ 
11:           for  $i \leftarrow 1$  to  $N - 1$  do
12:              $t_{i,max} \leftarrow t_i$ 
13:           end for
14:         end if
15:       end if
16:     end for
17:    $\vdots$ 
18: end for
19:  $\vdots$ 
20: end for
21: return  $t_{1,max}, \dots, t_{N-1,max}$ 

```

Algorithm 2 simulation of the system up to t_N .

Funct simu(t_1, \dots, t_{N-1})

```

1: for  $i \leftarrow 1, \dots, N - 1$  do
2:    $[x_i] \leftarrow$  simulation of  $(\mathcal{S}_i)$  starting at  $[x_i]$  from  $t_i$  to  $t_{i+1}$ 
3: end for
4: return  $[x_N]$ 

```

4.2 Monotonicity based algorithm

In some problems, the objective function can be viewed as monotonic with respect to the final time regarding each switching time of the time vector. More precisely, in the case of maximization, it can occur that for every switching time t_i , if t_i increases, the cost function $J(t_1, \dots, t_i, \dots, t_{N-1})$ decreases. The same argument remains true for minimization with $J(t_1, \dots, t_i, \dots, t_{N-1})$ increasing. For a particular time $\delta \in [t_1, t_n]$ and t_k the greatest switching instant lower than δ , a truncated value $J_\delta(t_1, \dots, t_k)$ of $J(t_1, \dots, t_{N-1})$ can be produced. It consists of computing the mode

switching until reaching the time δ . If $J_\delta(t_1, \dots, t_k)$ is already lower than the current solution, we know that continuing the computation of $J(t_1, \dots, t_i, \dots, t_{N-1})$ will only result in a value smaller than the current solution. In this case, we are able to prune a lot of switching instants that cannot be candidates for the solution of the problem before the simulation reaches the final time t_N . The monotonicity of a differentiable function can easily be checked using Theorem 1. It has been intensively used in global optimization (see for example [9]).

Theorem 1 (Monotonicity). *Let $f : \mathcal{X} \subseteq \mathbb{R}^n \rightarrow \mathbb{R}$ be a differentiable function, f' its derivative over \mathcal{X} , $[x] \in \mathbb{IR}^n$ a box such that $\mathcal{X} \subseteq [x]$ and $[f']$ the interval extension of f' over \mathcal{X} . Then, $0 \notin [f']([x])$ implies that f is monotonic in \mathcal{X} .*

Proof. The proof is simple using the fundamental theorem of interval analysis (see Section 3.1). \square

To take the decreasing monotonicity (for maximization) into account, Algorithms 1 and 2 have to be changed. After each choice of a switching time t_i , the truncated function $J(t_1, \dots, t_i)$ has to be computed incrementally ($J(t_1, \dots, t_i)$ is computed from $J(t_1, \dots, t_{i-1})$). The simulation can be stopped whenever the upper bound of $[J](t_1, \dots, t_i)$ is lower than the lower bound of the current solution. We know for sure that the solution cannot be found if we increase the time instants since the cost function cannot increase.

In the dual case of increasing monotonicity for minimization, if the lower bound of $[J](t_1, \dots, t_i)$ is bigger than the upper bound of the current solution, the simulation can be stopped as well.

The next section illustrates the use of the brute force and the monotonicity based algorithms.

5 Numerical experiments

We now apply our method on two different problems, the first one is the Goddard problem where the cost function strictly decreases monotonically and the second one shows results when the cost function is an integral and no assumption on its monotonicity can be considered.

For both examples, Algorithms 1 and 2 were implemented using DynIbex [1] with its improvement described in [16].

5.1 The Goddard Problem

The Goddard problem [7] models the ascent of a rocket through the atmosphere. The model of the ascent of the rocket can be described as follows:

$$\left[\begin{array}{l} \max \quad m(T) \\ \text{s.t.} \quad \dot{r} = v \\ \dot{v} = \frac{u - Av^2 \exp(k(1-r))}{m} - \frac{1}{v^2} \\ \dot{m} = -bu \\ u(\cdot) \in [0, 3.5] \\ r(0) = 1, v(0) = 0, m(0) = 1 \\ r(T) \geq \mathcal{R}_T \end{array} \right] \quad (9)$$

using the given values for the parameters $b = 2$, $T_{\max} = 0.2$, $A = 310$, $k = 500$, $r_0 = 1$, $v_0 = 0$, $m_0 = 1$, $\mathcal{R}_T = 1.01$. The cost function consists of a terminal cost represented by the mass of the rocket at the final time T . This model takes into account a continuous control $u(t)$. The rocket has to reach a given altitude while consuming the smallest amount of fuel. Physically, the optimal solution is a bang-singular arc-bang controller following the steps: 1) full power to break out 2) an increasing function to compensate for the drag effect and 3) turn off the engine and continue with the impulse. The question is then to find the time instants t_1 and t_2 to switch from one dynamics to the following one. The controls are known so the problem can be modeled using a 3-mode system. The first control is $u(t) = 3.5$ for all $t \in [0, t_1]$, it corresponds to a full thrust. The first mode system is then

$$(\mathcal{S}_0) \left\{ \begin{array}{l} \dot{r} = v \\ \dot{v} = \frac{3.5 - Av^2 \exp(k(1-r))}{m} - \frac{1}{v^2} \\ \dot{m} = -3.5b \\ r(0) = 1 \\ v(0) = 0 \\ m(0) = 1 \end{array} \right. . \quad (10)$$

The second control consists in counterbalancing the drag effect with $u(t) = 3.5 \tanh(1-t)$ for all $t \in [t_1, t_2]$ and the second mode system is as follows:

$$(\mathcal{S}_1) \left\{ \begin{array}{l} \dot{r} = v \\ \dot{v} = \frac{3.5 \tanh(1-t) - Av^2 \exp(k(1-r))}{m} - \frac{1}{v^2} \\ \dot{m} = -3.5b \tanh(1-t) \\ r(t_1) = r_1 \\ v(t_1) = v_1 \\ m(t_1) = m_1 \end{array} \right. , \quad (11)$$

with (r_1, v_1, m_1) the solution of (\mathcal{S}_0) at time t_1 . Finally, the system is then left in

free fall with a control $u(t) = 0$ for all $t \in [t_2, T_{\max}]$. The third mode system is:

$$(\mathcal{S}_2) \begin{cases} \dot{r} = v \\ \dot{v} = \frac{-Av^2 \exp(k(1-r))}{m} - \frac{1}{v^2} \\ \dot{m} = 0 \\ r(t_1) = r_2 \\ v(t_1) = v_2 \\ m(t_1) = m_2 \end{cases}, \quad (12)$$

with (r_2, v_2, m_2) the solution of (\mathcal{S}_1) at time t_2 . Since the cost function is the mass of the rocket over time and the mass strictly decreases monotonically over time, we can use the improvements described in Section 4.2. Figure 5 represents the meshing of the time sequences with the red dots being the optimal candidates. A large part of the search domain is not visited thanks to this property. Our approach provides the controller given in Figures 6, the results agree with [8] such as $t_1 = 0.019$, $t_2 = 0.063$ for a final mass $m = 0.6273$.

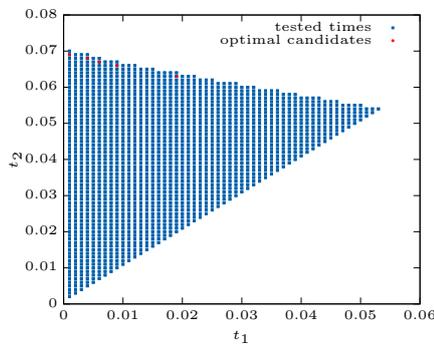


Figure 5: Mesh for t_2 w.r.t. t_1

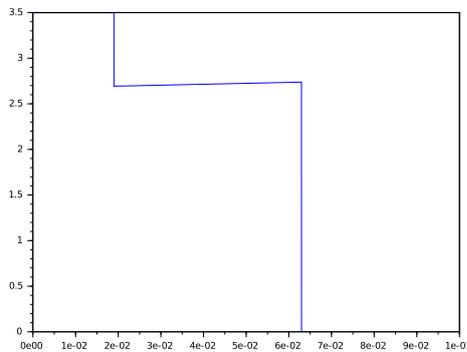


Figure 6: Time t vs. control $u(t)$.

5.2 Second Example

We now exemplify the case of a continuous cost function on a second experiment. This optimal control is a minimization and the method can be easily adapted since a minimization of a cost function is equivalent to the maximization of the function

with opposite sign. The example consists in solving the following problem:

$$\left[\begin{array}{l} \min_{(t_1, t_2, t_3)} \quad J = \frac{1}{2} \int_0^T \|x(t)\|^2 dt \\ \text{s.t.} \quad \dot{x} = A_1 x, t \in [0, t_1] \\ \quad \quad \dot{x} = A_2 x, t \in [t_1, t_2] \\ \quad \quad \dot{x} = A_1 x, t \in [t_2, t_3] \\ \quad \quad \dot{x} = A_2 x, t \in [t_3, T] \\ \quad \quad x(0) = \begin{pmatrix} 1 \\ 0 \end{pmatrix} \end{array} \right] \quad (13)$$

with $T = 1$ and

$$A_1 = \begin{pmatrix} -1 & 0 \\ 1 & 2 \end{pmatrix}, \quad A_2 = \begin{pmatrix} 1 & 1 \\ 1 & -2 \end{pmatrix}.$$

Contrarily to the previous experiment, the objective function is an integral. The objective function is evaluated using the development discussed in Section 3.3. Algorithm 2 is used and the results for different meshing parameters ϵ are given in Table 1. The solution has a smaller cost when meshing is thinner. A dynamic meshing can be considered subject to the gain in the objective function.

Table 1: Solutions for the second experiment using different parameters ϵ in Algorithm 2.

ϵ	switching instants (t_1, t_2, t_3)	J	computation time
0.1	(0.5, 0.7, 0.8)	[0.4770, 0.5274]	0.6s
0.01	(0.53, 0.72, 0.81)	[0.4751, 0.5257]	16mn

6 Conclusions and outlook

This paper deals with the problem of optimal switching instants in the case of an N -mode system where the input consists in the choice of these switching instants. It relies on guaranteed integration schemes and a time meshing to find sub-optimal switching times of these systems.

The method is exemplified using the Goddard rocket problem where the controls are assumed to be known. In this example, the cost function consists of a terminal cost function. A second example with a continuous cost function is also considered.

Since we currently apply a time meshing, the solution we provide is necessarily sub-optimal. Further investigations are to be made though for the guaranteed computation of the solution or at least a guaranteed bounding of it. It can be done considering time intervals instead of a time meshing.

An important improvement would be to compute a guaranteed outer approximation of the Jacobian of the cost function. This will allow to improve the convergence

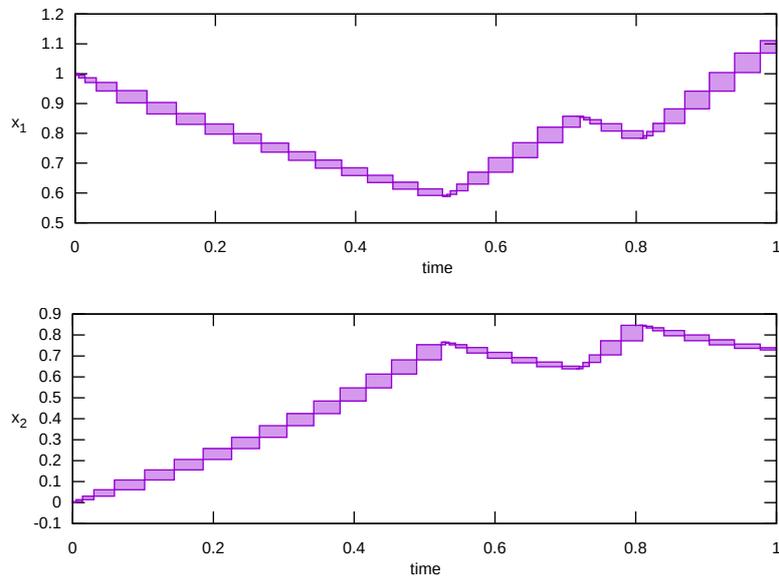


Figure 7: State trajectory over time (top: x_1 , bottom: x_2) for the solution given in Table 1 ($\epsilon = 0.01$).

by applying more efficient methods to explore extrema of the cost function such as the Newton or the Gauss-Seidel algorithms. Another objective could be to consider a continuous input $u(t)$ making this problem a switching instants in the case of N -mode control systems.

References

- [1] Alexandre dit Sandretto, Julien and Chapoutot, Alexandre. Validated explicit and implicit Runge-Kutta methods. *Reliable Computing*, 22:79103, 2016.
- [2] Alexandre dit Sandretto, Julien, Chapoutot, Alexandre, and Mullier, Olivier. Formal verification of robotic behaviors in presence of bounded uncertainties. In *1st IEEE International Conference on Robotic Computing (IRC)*, page 8188, Taichung, Taiwan, April 2017. IEEE. DOI: 10.1109/IRC.2017.17.
- [3] Bouissou, Olivier and Martel, Matthieu. Grklib: A guaranteed Runge Kutta library. In *12th GAMM-IMACS International Symposium on Scientific Computing, Computer Arithmetic and Validated Numerics (SCAN 2006)*, page 88, Duisburg, Germany, September 2006. IEEE. DOI: 10.1109/SCAN.2006.20.
- [4] Branicky, Michael S, Borkar, Vivek S, and Mitter, Sanjoy K. A unified framework for hybrid control: Model and optimal control theory. *IEEE transactions on automatic control*, 43(1):3145, January 1998. DOI: 10.1109/9.654885.

- [5] De Figueiredo, Luiz Henrique and Stolfi, Jorge. Affine arithmetic: Concepts and applications. *Numerical Algorithms*, 37(1-4):147158, 2004. DOI: 10.1023/B:NUMA.0000049462.70970.b6.
- [6] Egerstedt, Magnus, Wardi, Yorai, and Delmotte, Florent. Optimal control of switching times in switched dynamical systems. In *42nd IEEE Conference on Decision and Control*, volume 3, page 21382143, Maui, HI, USA, December 2003. IEEE. DOI: 10.1109/CDC.2003.1272934.
- [7] Goddard, Robert H. A method of reaching extreme altitudes. *Nature*, 105:809811, August 1920. DOI: 10.1038/105809a0.
- [8] Graichen, Knut and Petit, Nicolas. Solving the Goddard problem with thrust and dynamic pressure constraints using saturation functions. *IFAC Proceedings Volumes*, 41(2):1430114306, 2008. DOI: 10.3182/20080706-5-KR-1001.02423.
- [9] Hansen, Eldon and Walster, G William. *Global optimization using interval analysis: Revised and expanded*. CRC Press, DOI: 10.1201/9780203026922, 2003.
- [10] Jaulin, Luc, Kieffer, Michel, Didrit, Olivier, and Walter, Eric. *Applied interval analysis*. Springer, DOI: 10.1137/1.9780898717716, 2001.
- [11] Le Coënt, Adrien, Alexandre dit Sandretto, Julien, Chapoutot, Alexandre, and Fribourg, Laurent. An improved algorithm for the control synthesis of nonlinear sampled switched systems. *Formal Methods in System Design*, 53(3):363–383, Dec 2018. DOI: 10.1007/s10703-017-0305-8.
- [12] Le Cont, Adrien, Alexandre dit Sandretto, Julien, Chapoutot, Alexandre, and Fribourg, Laurent. Control of nonlinear switched systems based on validated simulation. In *International Workshop on Symbolic and Numerical Methods for Reachability Analysis (SNR)*, Vienna, Austria, April 2016. DOI: 10.1109/SNR.2016.7479377.
- [13] Liberzon, Daniel. *Calculus of variations and optimal control theory: A concise introduction*. Princeton University Press, Princeton, NJ, USA, DOI: 10.2307/j.ctvcn4g0s, 2011.
- [14] Lohner, Rudolf J. Enclosing the solutions of ordinary initial and boundary value problems. *Computer Arithmetic*, page 255286, 1987.
- [15] Moore, Ramon E., Kearfott, R Baker, and Cloud, Michael J. *Introduction to interval analysis*. Siam, DOI: 10.1137/1.9780898717716, 2009.
- [16] Mullier, Olivier, Chapoutot, Alexandre, and dit Sandretto, Julien Alexandre. Validated computation of the local truncation error of RungeKutta methods with automatic differentiation. *Optimization Methods and Software*, 33(4-6):718–728, 2018. DOI: 10.1080/10556788.2018.1459620.

- [17] Nedialkov, Nedialko S, Jackson, Kenneth R, and Corliss, George F. Validated solutions of initial value problems for ordinary differential equations. *Applied Mathematics and Computation*, 105(1):2168, 1999. DOI: 10.1016/S0096-3003(98)10083-8.
- [18] Rauh, Andreas and Hofer, Eberhard P. Interval methods for optimal control. In *Variational Analysis and Aerospace Engineering*, page 397418. Springer, 2009, DOI: 10.1007/978-0-387-95857-6_22.
- [19] Rauh, Andreas, Siebert, Charlotte, and Aschemann, Harald. Verified simulation and optimization of dynamic systems with friction and hysteresis. In *Proceedings of ENOC*, Rome, Italy, July 2011.
- [20] Witsenhausen, Hans. A class of hybrid-state continuous-time dynamic systems. *IEEE Transactions on Automatic Control*, 11(2):161167, 1966. DOI: 10.1109/TAC.1966.1098336.