Hamilton-Jacobi-Bellman approach for the climbing problem for multi-stage launchers[*]

O. Bokanowski[†] E. Cristiani[‡] J. Laurent-Varin[§] H. Zidani[¶]

Abstract

This study aims to investigate the Hamilton-Jacobi-Bellman approach for solving an optimization problem for space launchers. We consider a simplified (realistic) flight mission of the European launcher Ariane 5 to the Geostationary Transfert Orbit, and aim to minimize the fuel consumption. We consider the complete flight including stage separations and dynamic pressure constraint. Numerical experiments are performed with data given by the Centre National d'Études Spatiales (CNES) - the French space agency, and show the relevance of our approach. By using an anti-diffusive scheme, we obtain in small computing time the backward reachable set and the optimal trajectory. Comparisons with a reference trajectory of CNES are presented.

AMS Classification: 34H05, 49J15, 49L20

Keywords: multi-stage launchers, trajectory optimization, minimum time problem, Hamilton-Jacobi-Bellman approach

[*] This work is partially supported by Centre National d'Études Spatiales (CNES), under the grant CNES-81783/00

[†] Laboratoire J.-L. Lions, Université Pierre et Marie Curie 75252 Paris Cedex 05, France, and UFR de Mathématiques, Site Chevaleret, Université Paris-Diderot, 75205 Paris Cedex, France. boka@math.jussieu.fr

[‡] Ensta & Inria-Saclay (Projet Commands), 32 Bd Victor, 75739 Paris. emiliano.cristiani@gmail.com

[§] CNES, direction de lanceurs, Rond Point de l'Espace, 91000 EVRY, Julien.Laurent-Varin@cnes.fr

[¶] Ensta & Inria-Saclay (Projet Commands), 32 Bd Victor, 75379 Paris Cx 15, Hasnaa.Zidani@ensta.fr

Nomenclature.[1]

$\mathcal{B}$	=	computational domain
C_s	=	coefficient for the state/control constraint (3.5×10^4 Pascal·degree)
C_x, C_{xx}	=	aerodynamic coefficients
F_D	=	drag force
F_g	=	gravitational force
F_T	=	thrust force
$I_{sp,EAP}$	=	specific impulse of boosters (310 s)
$I_{sp,E1}$	=	specific impulse of first stage (429 s)
$I_{sp,E2}$	=	specific impulse of second stage (465 s)
L	=	longitude
ℓ	=	latitude
m	=	mass of the launcher
m_f	=	desired final mass
M_{EAP}	=	mass of boosters without propellant ($2 \times 38 \times 10^3$ kg)
M_{E1}	=	mass of first stage without propellant (17×10^3 kg)
M_{E2}	=	mass of second stage without propellant (6.2×10^3 kg)
M_L	=	mass of launcher without propellant
M_P	=	mass of the propellant
$M_{P,EAP}$	=	mass of propellant in boosters
$M_{P,E1}$	=	mass of propellant in first stage
$M_{P,E2}$	=	mass of propellant in second stage
M_{PL}	=	mass of the payload
M_0	=	computed optimal mass
P	=	atmospheric pressure
Q	=	dynamic pressure
r	=	distance of the launcher to the center of the Earth
r_T	=	Earth's mean radius (6378×10^3 m)
S_r	=	reference surface (23.345 m^2)
S_{EAP}	=	exit nozzle area of boosters (2×7.0 m^2)
S_{E1}	=	exit nozzle area of first stage (3.6 m^2)
S_{E2}	=	exit nozzle area of second stage (0 m^2)
$\mathcal{T}$	=	minimum time function
v	=	modulus of velocity of the launcher
α	=	angle of attack
β_{EAP}	=	flow rate for boosters (2×2000 kg s^{-1})
β_{E1}	=	flow rate for first stage (320 kg s^{-1})
β_{E2}	=	flow rate for second stage (40 kg s^{-1})
γ	=	path inclination
ϑ	=	reachability function
μ	=	Earth's gravitational constant (3.986×10^{14} m^3 s^{-2})
ρ	=	atmospheric density
Ω	=	Earth's angular velocity (7.29×10^{-5} s^{-1})
χ	=	Azimuth

[1] Units are given only when the numerical value is specified.

1 Introduction

This paper deals with the resolution of a climbing problem by the Hamilton-Jacobi-Bellman (HJB) approach.

Trajectory optimization for space launchers is a classical problem in optimal control, see for instance [1, 2] and the references therein. The pioneering Goddard [3] problem is perhaps the simplest model. It consists in maximizing the final altitude of the rocket, for a vertical flight, with a given initial propellant allocation. In one dimension this model is described by three state variables: the altitude r of the launcher, its velocity v and its mass m. The system is submitted to the aerodynamic force (the drag $\overrightarrow{F_D}$) and is controlled via the thrust force $\overrightarrow{F_T}$. Since this work, several studies were made on theoretical properties of the optimal trajectories [1, 4, 5] and numerical methods allowing to calculate these trajectories [1, 5, 6, 7, 8, 9, 10], and in particular [11, 12, 13, 14] for the ascent problem.

Several numerical methods have been developped for trajectory optimization of space launchers. The first class is based on a discretization of the control problem leading to the resolution of a finite dimensional nonlinear optimization problem. Special discretization and powerful solvers of nonlinear optimization are here needed to take into account the physical properties of the original problem, to handle properly the state constraints, and in order to obtain an accurate solution. An other class of methods usually used for trajectory optimization is shooting algorithms. These methods aim to solve the first order necessary optimality conditions (Pontryagin Maximum Principle or PMP). In some situations, the optimality conditions can be reduced to a two-point boundary problem for the state and the co-state. Of course, a solution to the first order necessary optimality conditions may not be the optimal solution to the original problem. Then second order optimality conditions (like Legendre-Clebsch conditions) have to be used to check optimality. Moreover, it is often required to have a good knowledge of the global behavior of the solution before implementing shooting methods (existence of singular arcs, number of commutations and so on).

Both classes of methods mentioned above present several difficulties when the optimization problem is non-convex. Indeed in this case the above methods are not able to avoid local minima, especially for shooting methods that have the reputation to have a small convergence radius. The initialization of these methods can also be very hard.

In control theory, there is another approach called Hamilton-Jacobi-Bellman (HJB) and which is based on the Dynamic Programming Principle (DPP) studied by R. Bellman [15]. It leads to a characterization of the value function as a solution of an HJB equation which is a first order nonlinear Partial Differential Equation (PDE) in dimension d, where d is the number of variables involved in the problem. The HJB equation may be viewed as a differential form of the DPP, and appears to be well-posed in the framework of *viscosity solutions* introduced by Crandall and Lions [16, 17, 18]. These tools allow us to perform numerical analysis of the approximation schemes. The theoretical and numerical contributions in this direction do not cease growing, see the book of Bardi and Capuzzo-Dolcetta [19], and the Appendix A by Falcone in the same reference.

An interesting by-product of the HJB approach is the synthesis of the optimal control in feedback form. Once the HJB equation is solved, for any starting point, the reconstruction of the optimal trajectory can be performed in real time. Also the method gives a global optimum and do not need any initialization procedure, see [20]. Another advantage of the HJB approach is that state or mixed state-control constraints can be taken into account. Of course, we should be careful when handling viscosity solutions for the HJB equation associated with such problems, see [21, 22].

Although the theoretical framework of the HJB approach is well known, this approach is also known to be difficult [23]. It is usually not used in practice because of the numerical complexity for discretizing the non-linear PDE when the dimension d of the state is large (classical finite difference methods are not very accurate and are CPU-time consuming). In many applications, the computation of the solution of the HJB equation is considered to be difficult to perform. For guidance purpose, an application of the HJB equation can be obtained around a reference trajectory [23]. But this approach connot be used for a general control problem where the optimal solution is not known.

In this work we aim to show that combining several new techniques for the HJB approach we can obtain efficient solutions to a fully nonlinear control problem.

More precisely, we investigate the potential of the HJB approach for the climbing problem in the case of the European Ariane 5 launcher developped by the French space agency "Centre National d'Études Spatiales" (CNES). For a given payload (fixed final mass), we aim at steering the launcher to the Geostationary Transfert Orbit (GTO) with *minimal* propellant consumption under a dynamic pressure constraint.

We assume that the launcher trajectory evolves in a plane (equatorial plane) and its position is detected by the altitude r and longitude L, the velocity is defined by its modulus v and the flight path angle γ. The dynamics of these state variables depend on the mass m. The latitude and azimuth variables are supposed to be fixed. These assumptions lead to a simplified model involving only 4 state variables (r, v, γ, m), the longitude L being deduced from the other variables (details are given in Section 2.2). The control variable is the angle for the direction of the thrust force. Our simplified model is in the same line as in [4], it means that there is no out-of-plane steering optimization considered in the problem, and the approximated trajectory remains planar afterwards.

The payload beeing fixed, the mass is a deterministic function of $t_f - t$ where t_f is the unknown optimal minimal time (see Section 2.3). The control problem is then formulated as a nonautonomous minimal time problem:

$$(\mathcal{P}_x) \qquad \left\{ \begin{array}{l} \mathcal{T}(x) := \text{minimize} \;\; t_f, \\[1ex] \text{with} \left\{ \begin{array}{l} \dot{y}_x(t) = f(t_f - t, y_x(t), \alpha(t)), \;\; t \in [0, t_f], \\ y_x(0) = x, \end{array} \right. \\ \qquad\quad t_f \geq 0, \qquad \alpha(t) \in \mathcal{A} \quad \text{for a.e. } t \in [0, t_f], \\ \qquad\quad y_x(t_f) \in \mathcal{C}, \qquad \Psi(y_x(t), \alpha(t)) \leq 0 \;\; \text{for a.e. } t \in [0, t_f], \end{array} \right.$$

where $y = (r, v, \gamma)$ belongs to $\mathbb{R}^3$, $\mathcal{A} \subset \mathbb{R}^m$ is the set of admissible control values, $f : \mathbb{R}_+ \times \mathbb{R}^d \times \mathcal{A} \to \mathbb{R}^d$ is the dynamics, $\mathcal{C} \subset \mathbb{R}^d$ is the target and $\Psi : \mathbb{R}^d \times \mathcal{A} \to \mathbb{R}$ is the mixed state/control constraint function. In order to compute $\mathcal{T}(x)$, we first consider the "reachability function $\vartheta(t, x)$" which takes a value of 0 if there exists a trajectory that can reach the target $\mathcal{C}$ at time t and from starting point x, otherwise takes a value of 1. It is shown in [24] that ϑ is a lower semi continuous (l.s.c.) function, and that it is the unique l.s.c. *viscosity solution* of the following HJB equation

$$\left\{ \begin{array}{ll} \vartheta_t(t, x) + \max\limits_{\alpha \in \mathcal{A}, \Psi(x, \alpha) \leq 0} \{ -f(t, x, \alpha) \cdot \nabla_x \vartheta(t, x) \} = 0 & t > 0, \; x \in \mathbb{R}^3 \\ \vartheta(0, x) = \Phi(x) & x \in \mathbb{R}^3 \end{array} \right. \qquad (1)$$

where ϑ_t represents the *time derivative*, and $\nabla_x \vartheta$ denotes *the gradient* with respect to the variable x *(the time derivative and the gradient are taken in a general sense)*. Moreover, $\mathcal{T}(x)$ can be recovered from $\vartheta(t,x)$ (see Proposition 3.1).

The complexity of a climbing mission (separation of stages, strong variations of the state variables) makes this HJB equation a long-time evolutive PDE where different scales are involved.

However, in our approach, the reachability function ϑ takes only values 0 and 1 and its computation corresponds to the determination of the front Γ_t that encloses the backward reachable set $\Omega_t := \{x, \vartheta(t,x) = 0\}$, and which requires only local calculations around Γ_t. For this aim, we discretize (1) by the Ultra Bee scheme which has the nice property to compute the front with a good accuracy without numerical diffusion (see [25, 26], see also [27, 28] for convergence results). Following [20], the Ultra Bee scheme is implemented on a sparse data structure in which the values of ϑ are stored. This leads to an algorithm where the CPU time grows as $O(N)$ where N is the number of nodes around the front.

They are several advantages of the present HJB approach. Mainly, it does not necessitate any new initialization procedure. Moreover, our algorithm is not specific to the Ariane 5 launcher. It could be used for more complex launcher missions with state constraints. Let us stress again that the HJB approach gives robust feedback control laws. This is illustrated in the numerical section where once the value function is computed a feedback control can be obtained even for perturbed data. On several numerical examples, we show the relevance of our approach, and in particular we obtain good results even on a relatively coarse mesh. Here the use of an anti-diffusive scheme plays a key role.

The paper is organized as follows. Section 2 is devoted to the presentation of the physical model and the related optimal control problem. In Section 3, we give some results on the Hamilton-Jacobi-Bellman approach in a general mathematical framework. Numerical results are given in Section 4.

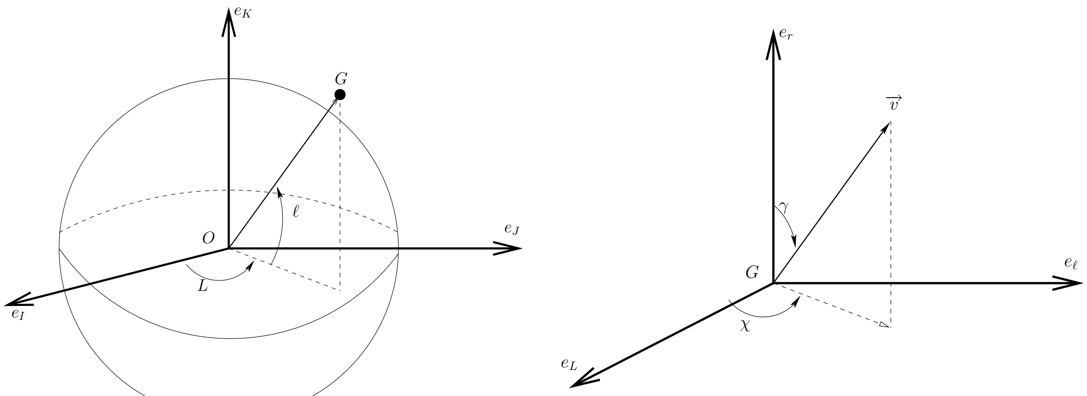
2 Problem statement

2.1 Physical model

In order to model the problem, we first define a suitable frame to describe the state variables. Let O denote the center of the Earth, e_K be the North-South rotation axis of the Earth, and Ω be the angular velocity oriented along e_K. We denote by $\mathcal{R} := (e_I, e_J, e_K)$ a quasi-inertial frame centered at O, rotating around e_K with angular speed Ω, and e_I is chosen to intersect the Greenwich meridian.

Let r_T denote the Earth's mean radius and G the mass center of the vehicle. The spherical coordinates of G are (r, L, ℓ), where r is the distance to O, L is the longitude and ℓ is the latitude. The moving frame $\mathcal{R}_1 = (e_r, e_L, e_\ell)$ is centered at G and defined such that e_r is the local vertical direction and (e_L, e_ℓ) is the local horizontal plane with e_ℓ lies in the plane (e_r, e_K) and pointing to the North (i.e., $e_\ell \cdot e_K > 0$), see Figure 1.

Let $X : t \longmapsto (x(t), y(t), z(t))$ be the trajectory of G in the quasi-inertial frame (e_I, e_J, e_K) and let $\overrightarrow{v} := \dot{x}e_I + \dot{y}e_J + \dot{z}e_K$ be the relative velocity. We parametrize $\overrightarrow{v}$ by its modulus v



Projection of $\overrightarrow{v}$ on the frame (e_L, e_ℓ)

Figure 1: setting of the frame

and two angles: the path inclination (or flight angle) γ which is the angle between e_r and $\overrightarrow{v}$; the azimuth χ which is the angle between e_L and the projection of $\overrightarrow{v}$ on (e_L, e_ℓ), see Figure 1.

In addition to the frames $\mathcal{R}$ and $\mathcal{R}_1$, we also introduce the orthonormal frame $\mathcal{R}_2 = (i, j, k)$ defined such that i has the same direction as the velocity $\overrightarrow{v}$ (i.e., $i = \frac{\overrightarrow{v}}{v}$), j is the unitary vector in the plane (i, e_r) perpendicular to i and satisfying $j \cdot e_r > 0$ and $k = i \wedge j$. This new frame $\mathcal{R}_2$ will be very useful to describe the evolution of the mass center G. From now on, the system is represented in the coordinates $(r, L, \ell, v, \gamma, \chi)$.

During its motion the engine is submitted to:

- Gravitational force: $\overrightarrow{F_g} = m\overrightarrow{g}$, where m is the mass of the vehicle and $\overrightarrow{g} = -g(r)e_r = -g(r)(\cos\gamma\ i + \sin\gamma\ j)$ is the gravitational field. The term $g(r)$ is given by

  $$g(r) := \frac{\mu}{r^2},$$

  where μ is Earth's gravitational constant (we neglect the J_2 correction term, and others high order term in the harmonic expansion of the gravitational field).

- Drag force: $\overrightarrow{F_D} = -F_D(r, v, \alpha)i$ opposite to the velocity $\overrightarrow{v}$. In this paper, we consider that F_D is given by

  $$F_D(r, v, \alpha) = S_r Q(r, v)(C_x(r, v) + C_{xx}|\alpha|)$$

  where α is the angle of attack which is the angle between the velocity $\overrightarrow{v}$ and the axis of the launcher (in the plane (e_r, i)) and $Q(r, v)$ is the dynamic pressure defined by $Q(r, v) = 0.5\rho(r)v^2$, with $\rho(r)$ the atmospheric density, S_r is the reference surface, C_x and C_{xx} are aerodynamic coefficients. In the sequel the angle α will be considered as a control variable.

- Thrust force: $\overrightarrow{F_T} = F_T(r)(\cos\alpha i + \sin\alpha j)$, with $F_T(r) = \beta g_0 I_{sp} - SP(r)$ with $g_0 = 9.81$ ms^{-2}, where $P(r)$ is the atmospheric pressure, and β (flow rate), I_{sp} (specific impulse) and S (surface) depend on the stage.

- Coriolis force $2m\overrightarrow{\Omega} \wedge \overrightarrow{v}$ and centripetal force $m\overrightarrow{\Omega} \wedge (\overrightarrow{\Omega} \wedge \overrightarrow{OG})$, where $\overrightarrow{\Omega}$ is the Earth's angular velocity. We add these two forces because we are not in an inertial reference frame.

Taking into account all these forces, and using Newton's laws of motion, we get:

$$m\frac{d\overrightarrow{v}}{dt} = \overrightarrow{F_g} + \overrightarrow{F_D} + \overrightarrow{F_T} - 2m\overrightarrow{\Omega} \wedge \overrightarrow{v} - m\overrightarrow{\Omega} \wedge (\overrightarrow{\Omega} \wedge \overrightarrow{OG}). \tag{2}$$

In the frame (i, j, k), the physical law (2) yields to:

$$\frac{dr}{dt} = v\cos\gamma \tag{3a}$$

$$\begin{aligned}\frac{dv}{dt} &= -g(r)\cos\gamma - \frac{F_D(r,v)}{m} + \frac{F_T(r,v,a)}{m}\cos\alpha \\ &\quad +\Omega^2 r\cos\ell(\cos\gamma\cos\ell - \sin\gamma\sin\ell\sin\chi)\end{aligned} \tag{3b}$$

$$\begin{aligned}\frac{d\gamma}{dt} &= \sin\gamma\left(\frac{g(r)}{v} - \frac{v}{r}\right) - \frac{F_T(r,v,a)}{vm}\sin\alpha \\ &\quad -2\Omega\cos\ell\cos\chi - \Omega^2\frac{r}{v}\cos\ell(\sin\gamma\cos\ell - \cos\gamma\sin\ell\sin\chi)\end{aligned} \tag{3c}$$

$$\frac{dL}{dt} = \frac{v}{r}\frac{\sin\gamma\cos\chi}{\cos\ell} \tag{3d}$$

$$\frac{d\ell}{dt} = \frac{v}{r}\sin\gamma\sin\chi \tag{3e}$$

$$\frac{d\chi}{dt} = -\frac{v}{r}\sin\gamma\tan\ell\cos\chi - 2\Omega(\sin\ell - \text{cotan}\gamma\cos\ell\sin\chi) + \Omega^2\frac{r}{v}\frac{\sin\ell\cos\ell\cos\chi}{\sin\gamma} \tag{3f}$$

For a multi-stage heavy launcher (such as Ariane 5), the mass $m(t)$ is given by the sum of the three quantities: payload's mass M_{PL}, launcher's mass M_L and propellant's mass M_P. Note that $M_L(t)$ depends on the flight phase as detailed in Table 1. The propellant mass M_P is split into tree parts of consuming fuel: $M_{P,EAP}$ for the boosters, $M_{P,E1}$ for the first stage, and $M_{P,E2}$ for the last stage:

$$m(t) = M_{PL} + M_L(t) + M_{P,EAP}(t) + M_{P,E1}(t) + M_{P,E2}(t) \tag{4a}$$

The propellant is progressively consumed and induces a variation of the mass of the engine. This variation depends on the flight phase:

- Phase 0: vertical take off (about 10 seconds). During this phase both boosters and the first stage are ignited. The launcher is completely vertical ($\gamma \equiv 0$) and there is nothing to control ($\alpha \equiv 0$). This phase will be not taken into account in our model.
- Phase 1: still during this phase both boosters and the first stage are ignited. It lasts until the boosters separation.
- Phase 2: propulsion is assured only by first stage. This phase lasts until the first stage separation.
- Phase 3: final phase, until the final time, where only the second stage is ignited.

The evolution of the mass can be summarized as follows where β_{EAP}, β_{E1} and β_{E2} are mass flow rates for the boosters, the first and the second stage respectively, and M_{EAP}, M_{E1} and M_{E2} are the mass of the boosters, first and second stages.

Our aim is to minimize the fuel consumption by steering the vehicle from a given initial position on the Earth to a specified orbit (for instance GTO orbit). The payload and the launcher's mass are supposed to be fixed.

Table 1: Evolution of the masses

Phase 0 & 1	Phase 2	Phase 3
$\dot{M}_{P,EAP}(t) = -\beta_{EAP}$	$\dot{M}_{P,EAP}(t) = 0$	$\dot{M}_{P,EAP}(t) = 0$
$\dot{M}_{P,E1}(t) = -\beta_{E1}$	$\dot{M}_{P,E1}(t) = -\beta_{E1}$	$\dot{M}_{P,E1}(t) = 0$
$\dot{M}_{P,E2}(t) = 0$	$\dot{M}_{P,E2}(t) = 0$	$\dot{M}_{P,E2}(t) = -\beta_{E2}$
$M_L(t) = M_{EAP} + M_{E1} + M_{E2}$	$M_L(t) = M_{E1} + M_{E2}$	$M_L(t) = M_{E2}$

(4b)

2.2 A simplified model

We assume here that the plane of motion is the equatorial plane ($\ell \equiv 0$ and $e_\ell \equiv e_K$), and the drag and thrust forces are contained in this plane ($\chi \equiv 0$). Furthermore, since we are aiming to minimize the fuel consumption, the evolution of (r, v, γ, m) does not depend on the longitude L, and the target does not make any constraint on L, the equation (3d) can be dropped out. Therefore, a simplified (nevertheless realistic) model is given by (4) and

$$\frac{dr}{dt} = v \cos\gamma \tag{5a}$$

$$\frac{dv}{dt} = -g(r)\cos\gamma - \frac{F_D(r,v)}{m(t)} + \frac{F_T(r,v)}{m(t)}\cos\alpha + \Omega^2 r \cos\gamma \tag{5b}$$

$$\frac{d\gamma}{dt} = \sin\gamma\left(\frac{g(r)}{v} - \frac{v}{r}\right) - \frac{F_T(r,v)}{v\,m(t)}\sin\alpha - \Omega^2\frac{r}{v}\sin\gamma - 2\Omega \tag{5c}$$

The simplification considered here and leading to (5) is coherent with the real case of launching from a nearly equatorial base to a geostationary transfer orbit.

Remark 2.1 *Although the longitude L is important to localize the launcher, it should not be considered as a state variable since it can be obtained from the variables r, v, γ, by a simple integration of the differential equation*

$$\frac{dL}{dt} = \frac{v}{r}\sin\gamma.$$

2.3 Optimal control problem

The problem is to steer the vehicle from an initial position (r_0, v_0, γ_0) to a target $\mathcal{C}$, which is a terminal manifold representing the GTO orbit. The initial conditions are chosen according to CNES's data. For the climbing problem it is natural to consider a mixed state/control constraint of the form

$$Q(r,v)|\alpha| \leq C_s, \tag{6}$$

where C_s is a constant. The constraint (6) links the incidence angle and the dynamic pressure. The l.h.s term in (6) represents the first term of the transverse load on the launcher. The launcher design minimizes the mass structure, and is strong along the longitudinal axis but not along the transverse axes. For this reason, constraint (6) should be considered along the optimal trajectory.

We assume that the launcher is always at full thrust, therefore, minimizing the fuel consumption corresponds minimizing time. This allows us to set the problem in the framework of minimum time problems

$$(\mathcal{P}) \qquad \begin{cases} \text{Minimize } t_f \\ (r, v, \gamma, m, \alpha) \text{ satisfies } (4) - (5) \\ r(0) = r_0,\ v(0) = v_0,\ \gamma(0) = \gamma_0, \\ \alpha(t) \in [\alpha_{min}, \alpha_{max}] \quad \text{for a.e. } t \in [0, t_f], \\ (r(t_f), v(t_f), \gamma(t_f)) \in \mathcal{C}, \quad m(t_f) = M_{PL} + M_{E2} \\ Q(r, v)|\alpha| \leq C_s \quad \text{for a.e. } t \in [0, t_f]. \end{cases} \tag{7}$$

Remark 2.2 *We point out that the aim of the above problem is not to maximize the payload at the target but rather to give the minimal fuel's quantity to take a given payload on the target.*

Now, we come back to the evolution of the mass. As we said at the end of Section 2.1, the "maximal capacities" of the boosters as well as the stages are supposed to be fixed. This does not mean that we fix the quantity of the fuel, but only the shape of the launcher and the capacities of each part.

The evolution of the mass does not depend on the other state variables. The final payload being fixed, the mass depends only on the final time of the trajectory. Indeed, if the final time t_f is known, the mass could be obtained by a simple backward integration of (4) with the final condition $m(t_f) = M_{PL} + M_{E2}$. Then we would obtain:

$$m(t) = \begin{cases} M_{PL} + M_{E2} + \beta_{E2}(t_f - t) & \text{in } (t_f - t_3, t_f) \\ M_{PL} + M_{E2} + \beta_{E2} t_3 + M_{E1} + \beta_{E1}(t_f - t_3 - t) & \text{in } (t_f - t_3 - t_2, t_f - t_3) \\ M_{PL} + M_{E2} + \beta_{E2} t_3 + M_{E1} + \beta_{E1} t_2 & \\ +M_{EAP} + (\beta_{EAP} + \beta_{E1})(t_f - t_3 - t_2 - t) & \text{in } (0, t_f - t_3 - t_2) \end{cases} \tag{8}$$

where t_3 is the duration of the third phase and t_2 that of the second. Note that in correspondence of each change of phase an empty reservoir is abandoned causing a discontinuity in the total mass. We define the function $\widetilde{m}$ as the function such that $m(t) = \widetilde{m}(t_f - t)$. With this new function, the control problem $(\mathcal{P})$ can be rewritten in the following form:

$$(\mathcal{P}') \qquad \begin{cases} \text{Minimize } t_f \\ \frac{dr}{dt} = v \cos\gamma \\ \frac{dv}{dt} = -g(r)\cos\gamma - \frac{F_D(r,v)}{\widetilde{m}(t_f - t)} + \frac{F_T(r,v)}{v\widetilde{m}(t_f - t)} \cos\alpha + \Omega^2 r \cos\gamma \\ \frac{d\gamma}{dt} = \sin\gamma \left(\frac{g(r)}{v} - \frac{v}{r} \right) - \frac{F_T(r,v)}{v\widetilde{m}(t_f - t)} \sin\alpha - \Omega^2 \frac{r}{v} \sin\gamma - 2\Omega \\ r(0) = r_0,\ v(0) = v_0,\ \gamma(0) = \gamma_0, \\ \alpha(t) \in [\alpha_{min}, \alpha_{max}] \quad \text{for a.e. } t \in [0, t_f], \\ (r(t_f), v(t_f), \gamma(t_f)) \in \mathcal{C}, \quad \widetilde{m}(0) = M_{PL} + M_{E2} \\ Q(r, v)|\alpha| \leq C_s \quad \text{for a.e. } t \in [0, t_f], \end{cases}$$

In this equivalent formulation, the mass does not appear as a state variable but it is explicitly given in function of the final time (which is unknown). This allows us to drop one dimension, greatly speeding up the computation. Note that this is possible because the equation for the mass does not involve the control α and its dynamics does not depend on the other variables.

Remark 2.3 *The problem $(\mathcal{P}')$ is a minimal time problem for a non-autonomous system. Moreover, the dynamics depends on the final time. In the following section, we recall how we can deal with such a problem by using the HJB approach.*

3 Hamilton-Jacobi-Bellman approach

3.1 General framework

The problem $(\mathcal{P}')$ enters into a general setting of time optimal control problems of the form:

$$(\mathcal{P}_x) \qquad \begin{cases} \mathcal{T}(x) := \text{minimize } \; t_f, \\ \text{with } \begin{cases} \dot{y}_x(t) = f(t_f - t, y_x(t), \alpha(t)) \,, \; t \in [0, t_f], \\ y_x(0) = x, \end{cases} \\ \qquad t_f \geq 0, \qquad \alpha(t) \in \mathcal{A} \quad \text{for a.e. } t \in [0, t_f], \\ \qquad y_x(t_f) \in \mathcal{C}, \qquad \Psi(y_x(t), \alpha(t)) \leq 0 \;\text{ for a.e. } t \in [0, t_f], \end{cases}$$

where the initial position x belongs to $\mathbb{R}^d$ (with $d \geq 1$), $\mathcal{A} \subset \mathbb{R}^m$ is the set of admissible control values, $f : \mathbb{R}_+ \times \mathbb{R}^d \times \mathcal{A} \to \mathbb{R}^d$ is the dynamics, $\mathcal{C} \subset \mathbb{R}^d$ is the target and $\Psi : \mathbb{R}^d \times \mathcal{A} \to \mathbb{R}$ is the mixed state/control constraint function. The function $\mathcal{T}$ represents the minimal time function which associates to each $x \in \mathbb{R}^d$ the minimal time t_f needed to reach the target with an admissible trajectory obeying the state/control constraint. Because the final time appears in the dynamics, the above setting of the control problem is not usual.

Let us recall some theoretical results concerning the HJB approach for problem $(\mathcal{P}_x)$. In this section, we assume that f and Ψ satisfy some classical assumptions:

(A1) f is a continuous function, and for every $(t, x) \in \mathbb{R}_+ \times \mathbb{R}^d$ the set $f(t, x, \mathcal{A})$ is closed and convex. There exists $c_0 \geq 0$ s.t. $\sup_{a \in \mathcal{A}} |f(t, \xi, a)| \leq c_0(1 + |\xi|)$. Moreover, for every $R > 0$, there exists $L_R > 0$, such that

$$\forall t \in \mathbb{R}_+, \; \forall \xi, z \in B(0, R), \quad \sup_{a \in \mathcal{A}} |f(t, \xi, a) - f(t, z, a)| \leq L_R |\xi - z|.$$

(A2) Ψ is a continuous function and for every $x \in \mathbb{R}^d$ the set $\{\Psi(x, a) \mid a \in \mathcal{A}, \text{ and } \Psi(x, a) \leq 0\}$ is closed, convex, and nonempty.

Time optimal control problems for autonomous dynamics have been studied in several publications. It is known that in this context the minimum time function satisfies the dynamic programming principle (DPP) and is characterized by a steady HJB equation [19]. Bokanowski et. al. prove in [24] that the minimum time function for the nonautonomous case does not satisfy the DPP and cannot be characterized by an HJB equation. However, still according to [24], function $\mathcal{T}$ is linked to the determination of ”backward reachable sets” of the controlled system. More precisely, define a ”reachability function” $\vartheta(t, x)$ that takes value 0 if there exists an admissible trajectory y_x which reaches the target $\mathcal{C}$ before time t and starting from x, otherwise takes the value 1, i.e.

$$\begin{cases} \vartheta(t, x) := \text{minimize } \; \Phi(y_x(t)), \\ \text{with: } \begin{cases} \dot{y}_x(s) = f(t - s, y_x(s), \alpha(s)) \text{ a.e. } s \in [0, t], \\ y_x(0) = x, \end{cases} \\ \qquad \alpha(s) \in \mathcal{A} \quad \text{for a.e. } s \in [0, t], \\ \qquad \Psi(y_x(s), \alpha(s)) \leq 0 \;\text{ for a.e. } s \in [0, t], \end{cases} \tag{9}$$

where Φ is given by

$$\Phi(y) = \begin{cases} 0 & \text{if } y \in \mathcal{C} \\ 1 & \text{otherwise.} \end{cases}$$

For every $t \geq 0$, the set $\Omega_t := \{x \in \mathbb{R}^d \mid \vartheta(t,x) = 0\}$ is called the "backward reachable set", or also "capture basin", at time t. It corresponds to the whole set of starting points x from which there exists an admissible trajectory reaching the target before time t. The interest for introducing the reachability function ϑ lies in the fact that this new function satisfies the DPP and we have:

Proposition 3.1 *For every $x \in \mathbb{R}^d$, we have:*

$$\mathcal{T}(x) = \min\{t \geq 0,\ \vartheta(t,x) = 0\},$$

with the convention that $\mathcal{T}(x) = +\infty$ whenever $\{t \geq 0,\ \vartheta(t,x) = 0\} = \emptyset$.

We prove in [24] that under assumption (A1)-(A2), the function ϑ, defined in (9), is the unique lower semi-continuous (l.s.c.) solution of the Hamilton-Jacobi-Bellman equation (in *the viscosity sense*)

$$\begin{cases} \vartheta_t(t,x) + \max\limits_{\alpha \in \mathcal{A}, \Psi(x,\alpha) \leq 0} \{-f(t,x,\alpha) \cdot \nabla_x \vartheta(t,x)\} = 0 & t > 0,\ x \in \mathbb{R}^d \\ \vartheta(0,x) = \Phi(x) & x \in \mathbb{R}^d \end{cases} \tag{10}$$

where ϑ_t represents the *time derivative*, and $\nabla_x \vartheta$ denotes *the gradient* with respect to the variable x *(the time derivative and the gradient are taken in a general sense).*

The HJB equation is well known to have complex behavior. Even in the case of smooth initial data the solution can become non differentiable in finite time. In quest of unique solutions, Crandall and Lions [16] introduced the notion of a viscosity solution, which has been extended to the more general case of a l.s.c. initial function Φ, and under more general assumptions on the dynamics f. The notion of a viscosity solution appears to be the right definition for the solution of the HJB equation since it allows us to deal with non differentiable functions.

Now, to obtain the minimum time function $\mathcal{T}$, we first compute the value function ϑ by solving the HJB equation (10). Let us point out that function ϑ takes only values 0 and 1 and its computation corresponds to the determination of the front Γ_t that encloses the backward reachable set Ω_t, which requires only local calculations around the front. To do this, we use the combination of two ingredients. The first one is the discretization of (10) by the Ultra Bee scheme which has the nice property of computing the front with good accuracy without diffusion (see [25, 26]). The second one (as important as the first one) is an efficient data structure in which the values of ϑ are stored. Following [20], we choose the *Sparse semi-dynamic* data structure which leads to an algorithm where the CPU time grows as $O(N)$ where N is the number of nodes around the front.

Finally, we recall that the calculation of ϑ and $\mathcal{T}$ is not an aim by itself. The final goal is the reconstruction from $\mathcal{T}$ of the optimal feedback control $\alpha^*(\cdot)$ [19, Appendix A]. Moreover, as we said before, the most interesting advantage of this method is the fact that it allows us to reach a global minimum of the cost functional, while classical methods as shooting methods provide only local minima.

3.2 HJB approach for problem $(\mathcal{P}')$

In the HJB approach the optimal control problem $(\mathcal{P}')$ can be seen as a front propagation problem in the space (r, v, γ), where the initial front at time $t = 0$ is the target (GTO orbit). The front at time t corresponds to the set of points which can reach the target in time t, but not before. Note that in this interpretation the time flow is reversed with respect to the physical problem since the front propagates from the target to the space while the optimal trajectory goes from the space to the target.

Hence, the reachable function ϑ associated to $(\mathcal{P}')$ satisfies:

$$\begin{cases} \vartheta_t(t,x) + H(t,x,\nabla_x \vartheta(t,x)) = 0 & t > 0,\ x = (r,v,\gamma) \in \mathbb{R}^3 \\ \vartheta(0,x) = \Phi(x) & x = (r,v,\gamma) \in \mathbb{R}^3, \end{cases} \tag{11}$$

where the Hamiltonian $H : \mathbb{R}_+ \times \mathbb{R}^3 \times \mathbb{R}^3 \to \mathbb{R}$ is defined by

$$\begin{aligned} H(t,r,v,\gamma,p_1,p_2,p_3) : \ = \max_{\substack{\alpha\in[\alpha_{min},\alpha_{max}],\\ Q(r,v)|\alpha|\le C_s}} & \Big\{ - p_1 v \cos\gamma \\ & -p_2 \left(-r) \cos(\gamma) - \frac{F_D(r,v)}{\widetilde{m}(t)} + \frac{F_T(r,v)}{v\widetilde{m}(t)} \cos\alpha + \Omega^2 r \cos\gamma \right) \\ & -p_3 \left(\sin\gamma \left(\frac{g(r)}{v} - \frac{v}{r} \right) - \frac{F_T(r,v)}{v\widetilde{m}(t)} \sin\alpha - \Omega^2 \frac{r}{v} \sin\gamma - 2\Omega \right) \Big\}. \end{aligned}$$

Remark 3.1 *Problem $(\mathcal{P}')$ enters into the general setting of Section 3.1. The only difference here is the fact that the dynamics does not satisfy exactly assumption (A1). In particular, the dynamics for problem $(\mathcal{P}')$ is not continuous in time since it has jumps at the end of each phase. However, the HJB approach is still valid and equation* (11) *should be understood in the general viscosity sense given in [29]. We will not enter in more theoretical details and focus only on numerical issues.*

3.3 Numerical approximation of ϑ and $\mathcal{T}$

Although equation (10) is set in the whole space $\mathbb{R}^+ \times \mathbb{R}^d$, to perform computation we have to choose a finite domain $[0,T] \times \mathcal{B}$, where T is an upper bound of the needed time to reach the target (it could be increased during the computation process), and $\mathcal{B}$ is chosen as a d-dimensional box containing the target and the initial point of the optimal trajectory. Then we consider a discretization of the HJB equation (10) on $[0,T] \times \mathcal{B}$. For this we choose a uniform space grid $\mathcal{G}$ on $\mathcal{B}$, with mesh step $(\Delta r, \Delta v, \Delta\gamma)$. On each time t_n and at each node point $x_i \in \mathcal{G}$, we denote by V_j^n an approximation of $\vartheta(t_n, x_j)$. Then the discrete approximation of (10) is given by:

$$\frac{V_j^{n+1} - V_j^n}{\Delta t_n} + \max_{\alpha\in A_{Na},\Psi(x_j,a)\le 0} \{ -f(t_n,x_j,\alpha) \cdot [DV^n] \} = 0, \quad \text{for every } x_j \in \mathcal{G},\ n \ge 0, \tag{12a}$$

$$V_j^0 = \Phi(x_j), \quad \text{for every } x_j \in \mathcal{G}, \tag{12b}$$

where as usual A_{N_a} is a finite set of N_a elements of $\mathcal{A}$ ($N_a > 0$), and where $[DV^n]$ is an approximation of the "gradient". It can be obtained by a finite differences scheme (Upwind, ENO, WENO, ...), by semi-Lagrangian methods, or any other scheme. In this paper, we use for this approximation the first-order HJB Ultra Bee scheme together with an efficient storage technique introduced in [20]. In (12), Δt_n is a variable time step. Recall that for stability reasons, the time step and the grid mesh should satisfy a CFL[2]-like condition, see for example [20].

To compute the minimum time function, we follow the characterization of Proposition 3.1. We first set $\mathcal{T}(x_i) = 0$ for every $x_i \in \mathcal{C}$, otherwise $\mathcal{T}(x_i) = +\infty$. If at time t_n we detect on the cell centered at x_i that V_i^n is "close to zero" and $\mathcal{T}(x_i) = +\infty$, then we set $\mathcal{T}(x_i) = t_n$ (and afterwards the value $\mathcal{T}(x_i)$ cannot be changed anymore). We also store the corresponding mass computed by (8) at that time. Since $\mathcal{T}(x_i)$ corresponds to the minimal time to reach the target from the point x_i with the optimal control, the corresponding mass is the lowest possible allowing to reach the target.

Remark 3.2 *Let us point out that physical reasons allow us to restrict the computational domain, thus decreasing the CPU time. For instance, the region of small altitude with high velocity is clearly not interesting.*

When function ϑ (and then $\mathcal{T}$) is computed everywhere, we use it in order to reconstruct the optimal feedback control law and the corresponding optimal trajectory [19, Appendix A]. In all the tests performed in the next section, we use for the reconstruction of the optimal trajectory a classical fourth-order Runge-Kutta scheme.

4 Numerical simulations

Before attacking the real problem, we focus our attention on a simplified 2-dimensional model in which our technique can be more easily illustrated and the code verified. All computations were done on a 2.6 GHz computer Opteron 275 with 4 GB RAM.

4.1 A model problem in 2D

We consider here the case of a launcher with two stages $E1$ and $E2$, which moves only in the vertical direction and with thrust from two different engines. The final goal here is to find the minimal amount of propellant needed by the launcher to reach a given altitude $3 \le r_{\mathcal{C}} \le 3.1$, with free speed and fixed mass $M_{PL} = 0.1$ (payload). At the beginning the rocket uses only the first engine (phase 1) until its propellent is completely consumed. After that, the first engine and its empty reservoir are lost and the launcher uses only the second engine (phase 2). It is also assumed that one engine is always on, that is a ballistic phase is not allowed. The dynamics we consider are

$$\left\{ \begin{array}{l} \dot{r}(t) = v(t) \\ \dot{v}(t) = \frac{u(t)F_T(t)}{m(t)} - \frac{2}{r^2(t)}, \end{array} \right. \tag{13}$$

where F_T is the thrust force and u is a control parameter. We choose

$$F_T(t) = \left\{ \begin{array}{ll} 2.5 & \text{in phase 1} \\ 0.5 & \text{in phase 2.} \end{array} \right.$$

[2] Courant-Friedrichs-Levy

The mass varies following the law $\dot{m} = -\beta(t)$, where

$$\beta(t) = \begin{cases} 1 & \text{in phase 1} \\ 0.2 & \text{in phase 2.} \end{cases}$$

The control u adjusts the engine's power and it can vary in $\mathcal{A} = [0, 1]$. Note that the control u affects only the lift power, but not the mass consuming as in the Goddard problem. Of course in this case the optimal control is the maximal one and there is no singular arcs. We assume that the weight of the second stage (with its propellant) is 0.2 and the weight of the empty first stage is 0.1. The phase changes when the mass becomes equal to $M_{PL} + 0.2 = 0.3$ and, at that time, the mass immediately drops to 0.3-0.1=0.2 (separation of the first stage).

As we already noted, the HJB approach allows us to deal easily with mixed state/control constraints. To test this feature, we impose that $u \leq 0.6$ if $r < 1.8$. This choice just says that if the altitude is low, it is not allowed to use the maximal engine power. In the present test we have used only the following computational domain

$$\{(r, v) : \ 1 \leq r \leq 3.1, \ -0.5 \leq v \leq 4 \text{ and } v \leq r + 1\}.$$

In Fig.2 we show the level sets of the minimum time function $\mathcal{T}$ which represent the points from which it is possible to reach the target in the same time (and with the same mass). We also show an approximate optimal trajectory (circles) starting from the point $x = (1.6, 0)$ with the optimal mass $m^*(x) = 0.97$ computed by the algorithm, as well as the exact trajectory (solid line), and we observe a good correspondance.

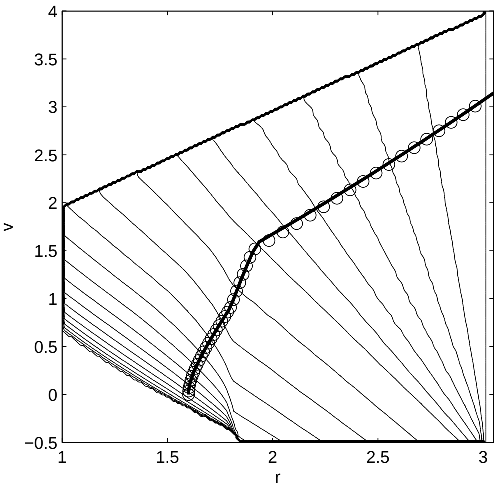


Figure 2: A 2D dimensionless model problem. Level sets of the function $\mathcal{T}$ and the approximate optimal trajectory (circles) compared with the exact optimal trajectory (solid line).

4.2 The real problem

Now we consider the real model corresponding to $(\mathcal{P}')$. Unless specified, the payload will be $M_{PL} = 15.37 \times 10^3$ kg. This means that the final mass of the trajectory should be $m_f = M_{PL} + M_{E2} = 21.57 \times 10^3$ kg (just before the expulsion of the second stage $E2$).

All the following tests will be performed in the domain:

$$(r, v, \gamma) \in \mathcal{B} := [r_T + 450 \ \text{m}, \ r_T + 450 \ \text{km}] \times [60 \ \text{ms}^{-1}, \ 10500 \ \text{ms}^{-1}] \times [0, \pi/2].$$

As stated in Section 3, physical considerations allow to restrict the computational domain. Here, we shall consider local computation on the subdomain K of points (r, v, γ) such that

$$(r, v, \gamma) \in \mathcal{B}, \quad f_{min}(r) < v < f_{max}(r), \tag{14}$$

with

$$\begin{aligned} & f_{min}(r) := \max\left(\max((r - b_1)/a_1, (r - b_2)/a_2), \min(500.0, (r - b_3)/a_3)\right), \\ & f_{max}(r) := (r - b_4)/a_4, \end{aligned}$$

and with constants given by:

$$\begin{aligned} a_1 &= 33.330, \quad b_1 = 6544700; \qquad a_2 = 95.000, \quad b_2 = 6388000; \\ a_3 &= 52.000, \quad b_3 = 6379000; \qquad a_4 = 14.285, \quad b_4 = 6370900. \end{aligned}$$

In all the sequel, we will compare our results to a *reference trajectory*, which is a numerical solution obtained by CNES by using shooting method on the complete 6D physical model.

The variables r and v vary in large intervals. After the take-off, the altitude as well as the velocity of the launcher are small but their variations are large. In order to analyze these variations well, we are tempted to take small discretization steps, requiring considerable computing time. For these reasons, we will make two types of tests:

- In Tests 1 and 2, we will consider uniform grids on $\mathcal{B}$ with reasonable discretization steps. These steps are not sufficiently small to catch the behavior of the trajectory for small values of r and v. Therefore, we will restrict ourselves to the reconstruction of the optimal trajectory starting from the initial point (r_0, v_0, γ_0) :=(10.03 km, 476 ms^{-1}, 0.64 rad), taken from the reference trajectory.

- In Test 3, we propose a change of variable which will make it possible to have fine steps when r is small and large steps elsewhere. This change of variable will enable us to start the simulation from a very low altitude. Moreover, this change of variable will be very useful to deal with the terms $\frac{1}{r}$ and $\frac{1}{v}$ in the dynamics.

Finally, let us describe some notation which will be used: N_r, N_v and N_γ denote number of nodes in the r-axis, in v-axis, and γ-axis, respectively.

Test 1: Thin target problem. Here we consider the problem of reaching a given point $\mathcal{C} := \{(r_T + 353 \ \text{km}, \ 9617\text{ms}^{-1}, \ 1.45 \ \text{rad})\}$ that belongs to a particular GTO target (we consider here the GTO target which has its perigee at $r = 225$ km above earth and with relative velocity $v = 9742.64$ ms^{-1}).

To compute the minimum time function, we solve (12) on a grid with $N_r = 100$, $N_v = 100$, $N_\gamma = 50$, and we use $N_a = 30$ control variables. In Fig. 3 we compare the reference trajectory with our optimal trajectory reconstructed from the approximation of the value function. We can see that, even on a rough grid, the computed optimal trajectory is close to the reference one. The shadowed region represents the backward reachable set, including state constraints. Recall the reference trajectory is obtained by using the complete model (3), while

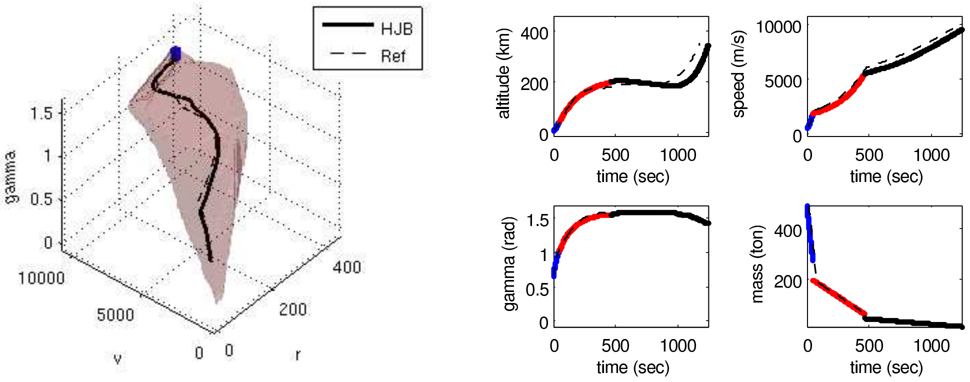


Figure 3: (Test 1) 100 × 100 × 50 meshes, 30 control variables.

the HJB computation is based on the approximated model (5) and its numerical approximation by the Ultra Bee scheme. The whole HJB computation (backward reachable sets, synthesis of the optimal trajectory) takes only 217 s.

In Table 2, we report computations obtained with various mesh sizes. The column "Error/Ref" shows the error between the computed and the CNES reference trajectory in variables (r, v, γ). The column "Error/G" shows the error with the trajectory obtained with the finest grid G: $300 \times 300 \times 100$ (and $N_a = 30$). The error is computed as a normalized Hausdorff's distance.[3] The column "Mass M_0" gives the computed optimal initial mass, and finally, the last column gives the optimal time needed to reach the target. The known optimal reference time for this problem is $T = 1178.4$ s. We have observed that taking N_c greater than 30 does not significantly improve the convergence.

Table 2: Test 1, Comparison with reference trajectory.

(N_r, N_v, N_γ)	CPU time	Error/Ref	Error/G	Mass M_0 (kg)	time (s)
50 50 50	78	0.102	0.105	494.78×10^3	1164.74
100 100 50	217	0.086	0.056	512.73×10^3	1168.90
200 200 75	1029	0.091	0.055	521.67×10^3	1170.97
300 300 100	3571	0.044	0.000	523.92×10^3	1171.49

In Fig. 4 we show optimal mass values (represented with squares) at the starting point computed for different final mass (desired) values (represented with black dots). These computations are performed on a coarse grid with a $100 \times 100 \times 50$ mesh. After this computation, we verify the launcher arrives with the desired mass. Due to the accumulation of numerical errors (computation of the value function and then synthesis of the optimal trajectory), there

[3] If A and B are two sets (two trajectories to be compared), we have considered the normalized Hausdorff distance to be $d_H(A, B) := \max(\delta(A, B), \delta(B, A))$ where $\delta(A, B) := \max\{d_0(x, B),\ x \in A\}$ and

$$d_0(x, B) := \min\{\sqrt{\frac{(x_1 - y_1)^2}{x_1^2} + \frac{(x_2 - y_2)^2}{x_2^2} + \frac{(x_3 - y_3)^2}{x_3^2}}, \quad y = (y_1, y_2, y_3) \in B\}.$$

is a loss of mass and the numerical approximation gives an optimal trajectory of the launcher which arrives to the target with the "effective" final mass represented with triangles. An error on optimal time of about 1% may imply an error on the final mass of about 3×10^3 kg.

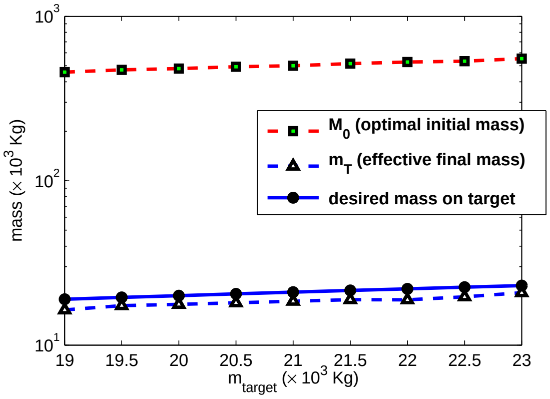


Figure 4: Test 1, optimal mass.

Finally, we come back to the test with the desired final mass $m_f = M_{PL} + M_{E2} = 21.57 \times 10^3$ kg. Now, instead of using the optimal initial mass $M_0 = 512.73 \times 10^3$ kg to reconstruct the optimal trajectory we impose the initial mass to be $\widehat{M_0} = 553.87 \times 10^3$ kg at the starting point (r_0, v_0, γ_0) as in the reference trajectory. In that case, we observe (see Fig. 5) that we are able to recover the reference trajectory with a good accuracy ($d_H = 0.027$), and with a final mass $m_{\mathcal{C}} = 21.036 \times 10^3$ kg.

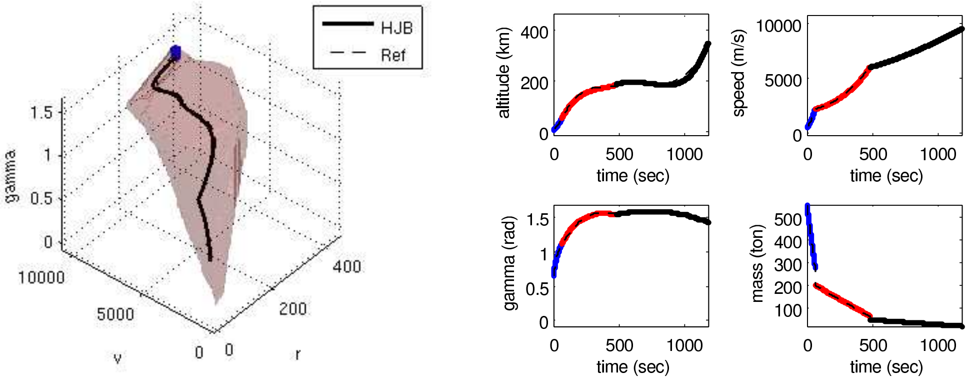


Figure 5: Test 1, trajectory reconstruction with an initial mass $\widehat{M_0} = 553.87 \times 10^3$ kg.

Test 2: GTO Target and constraints. In this test we consider that the target $\mathcal{C}$ is the GTO orbit. This target can be characterized by a set of two equations in variables (r, v, γ), and hence it is a one-dimensional curve that can be numerically approximated. In Fig. 6, we have plotted the numerical solution obtained by HJB on a grid of $100 \times 100 \times 50$ nodes and with $N_a = 30$ control variables (we compare to the reference trajectory plotted in dotted line).

We obtain the best point of the orbit to be reached with an optimal mass $M_0 = 505.01 \times 10^3$ kg and an optimal time of 1167.1 s. These results are consistent with Test 1. Indeed, the optimal time (as well as the fuel consumption) is smaller when the target is the whole GTO orbit than when it is just a fixed point. Notice the capture bassin is bigger than in Test 1, however, the CPU time for computing the value function and synthesizing an optimal trajectory is just of 253 s.

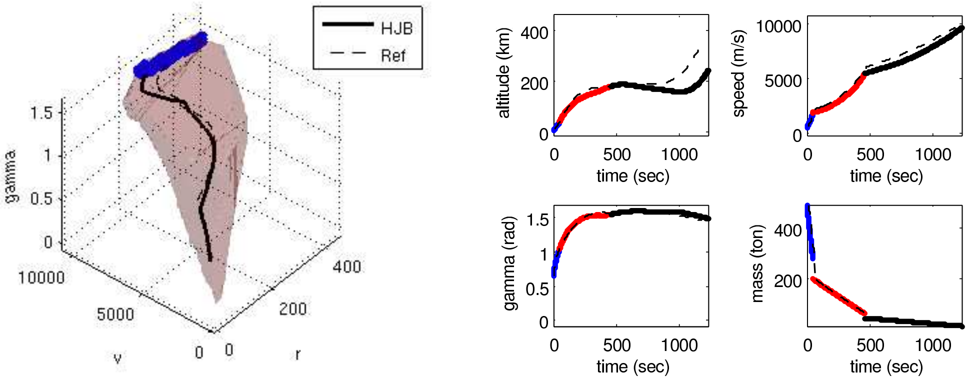


Figure 6: Test 2, 100 × 100 × 50 mesh with 30 control variables.

As in Test 1, we use the optimal control law obtained by the HJB approach to reconstruct a trajectory with the initial mass $\widehat{M_0} = 553.87 \times 10^3$ kg. We observe that this trajectory (see Fig. 7) reaches the GTO target with the final mass $m_{\mathcal{C}} = 22.09 \times 10^3$ kg, slightly above the desired mass ($m_f = 21.57 \times 10^3$ kg).

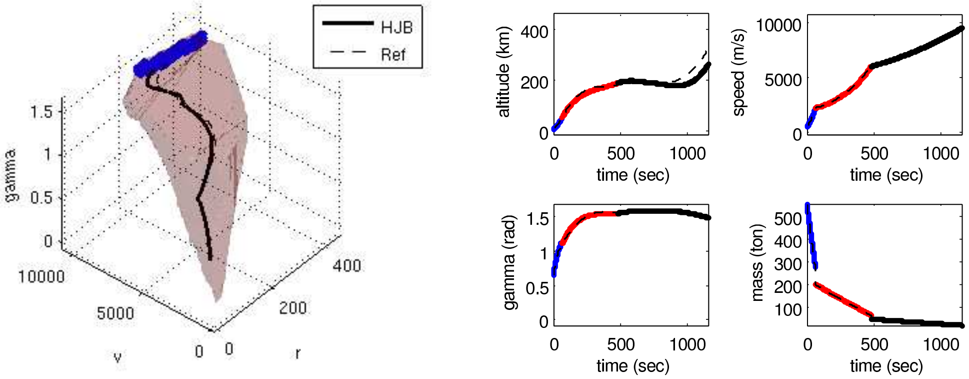


Figure 7: Test 2. Reconstruction of a trajectory corresponding to the optimal control law and with an initial mass $\widehat{M_0} = 553.57 \times 10^3$ kg.

Test 3: Complete trajectory. Here we aim to compute a complete trajectory. We are faced to the following numerical difficulties. First, if we discretize the full computational domain, we observe that computations get very low when the front is getting close to the origin $(r, v) = (0, 0)$ (this is due to the CFL constraint already mentionned in Sec. 3.3. Moreover, the numerical front is not precise enough to reach the starting point. Thus, in order to reach a better accuracy at the beginning of the trajectory, we use a change of variables $(r, v, \gamma) \to (x, y, \gamma)$ defined by:

$$r = K_r(e^x - 1) + r_T, \quad v = K_v(e^y - 1) + v_T,$$

with $K_r = 1.5$ km, $K_v = 1165.6$ ms^{-1}, and $v_T = 10$ ms^{-1}. After this simple change of variables, we solve the HJB equation on a regular grid as before, in variables (x, y, γ). Then we compute the rescaled minimum time function $\mathcal{T}(r, v, \gamma)$. Figure 8 illustrates the mesh used.

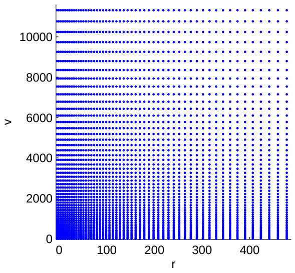


Figure 8: Example of adapted mesh grid points used for Test 3, in the (r, v) plane (here with only 50×50 points represented).

Finally, in Fig. 9 we show the optimal trajectory starting from the initial point

$$(r_0, v_0, \gamma_0) := (501.69 \text{ m}, \ 76.20 \text{ ms}^{-1}, 0.078 \text{ rad}).$$

This point is taken from the reference trajectory and corresponds to the position of the launcher at the end of the take-off (phase 0). We obtain the optimal initial mass $M_0 = 649.77 \times 10^3$ kg corresponding to the optimal time $T = 1200.6$ s. Using the minimal time function to obtain a feedback control law, we can then reconstruct a trajectory leading to a final mass $m = 19.47 \times 10^3$ kg.

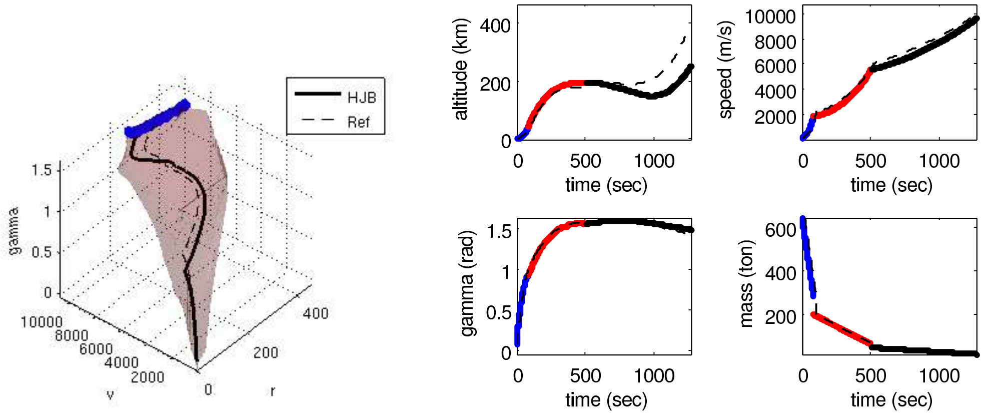


Figure 9: Test 3, Approximation of the optimal trajectory on a grid of $200 \times 200 \times 75$ nodes. CPU time = 57 min.

This approach could be used to initialize a shooting method (which needs a first good approximation in order to work) and then obtain better accuracy. Preliminary numerical work in this direction already shows that this is very successful on various examples, including the Goddard problem [30].

5 Conclusion

In this work we have been able to estimate the minimum time function for the climbing problem by solving an adequate front propagation problem. We deal with a simplified 3D model, including separation of stages and boosters, that gives comparable solutions to a 6D reference model. Our approach uses a particular sparse coding method for data storage together with an antidiffusive scheme, and this allows to perform computations in a reasonable CPU time. The great advantage of the knowledge of the minimum time function is that it gives a robust feedback control law to reach a given target.

To our best knowledge, this is the first study of the applicability of the HJB approach for the climbing problem. As expected, this approach cannot provide a very accurate solution without involving a high CPU time. However, we showed that it can provide a qualitative global view of the backward reachable sets, and should give a good globally optimal solution that can then be used as initial guess for a more precise method. This subject will be studied in an ongoing work.

References

[1] F. Bonnans, P. Martinon, and E. Trélat. Singular arcs in the generalized Goddard's problem. *J. Optimization Theory Applications*, 139 (2):439–461, 2008.

[2] K.P. Zondervan, T.P. Bauer, J.T. Betts, and W.P. Huffman. Solving the optimal control problem using a nonlinear programming technique. part 3: Optimal shuttle reentry trajectories. *Proceedings of the AIAA/AAS Astrodynamics conference, Seattle.*, 1984.

[3] R.H. Goddard. A method of reaching extreme altitudes. *Smithsonian Miscellaneous Collection*, 71(4), 1919.

[4] B. Bonnard, L. Faubourg, and E. Trélat. Optimal control of the atmospheric arc of a space shuttle and numerical simulations with multiple shooting method. *Mathematical Models and Methods in Applied Sciences*, 15(1):109–140, 2005.

[5] J. Laurent-Varin. *Optimal ascent and reentry of reusable rockets.* PhD thesis, Ecole Polytechnique, 2005. Supported by CNES and ONERA.

[6] J.T. Betts. *Practical methods for optimal control using nonlinear programming.* Society for Industrial and Applied Mathematics, Philadelphia, 2001.

[7] H.J. Oberle. Numerical computation of singular control functions in trajectory optimization problems. *Journal of Guidance, Control and Dynamics*, 13:153–159, 1990.

[8] H.J. Pesch. A practical guide to the solution of real-life optimal control problems. *Control and Cybernetics*, 23(1-2):7–60, 1994.

[9] H.J. Pesch. Real-time computation of feedback controls for constrained optimal control problems, Part 2: a correction method based on multiple shooting. *Optimal Control, Applications and Methods*, 10(2):147–171, 1989.

[10] J.T. Betts. Survey of numerical methods for trajectory optimization. *Journal of Guidance Control and Dynamics*, 21(2):193–207, 1998.

[11] P. Martinon, J. F. Bonnans, J. Laurent-Varin, and E. Trélat. Numerical study of optimal trajectories with singular arcs for an Ariane 5 launcher. *Journal of Guidance, Control, and Dynamics*, 32(1):51–55, 2009.

[12] A.J. Calise and P.F. Gath. Optimization of launch vehicle ascent trajectories with path constraints and coast arcs. *J. Guidance, Control, and Dynamics*, 24(2):2001, 296-304.

[13] P. Lu, H. Sun, and B. Tsai. Closed-loop endoatmospheric ascent guidance. *J. of Guidance, Control, and Dynamics*, 26(2):283–294, 2003.

[14] L. Zhang and P. Lu. Fixed-point algorithms for optimal ascent trajectories of launch vehicles. *Engineering optimization*, 40(4):361–381, 2008.

[15] R. Bellman. *Dynamic programming.* Princeton University Press, Princeton, 1961.

[16] M.G. Crandall and P.-L. Lions. Viscosity solutions of Hamilton Jacobi equations. *Bull. American Mathematical Society*, 277(1):1–42, 1983.

[17] M.G. Crandall and P.-L. Lions. Two approximations of solutions of Hamilton-Jacobi equations. *Mathematics of Computation*, 43(167):1–19, 1984.

[18] M.G. Crandall, L.C. Evans, and P.-L. Lions. Some properties of viscosity solutions of Hamilton-Jacobi equations. *Trans. Amer. Math. Soc*, 282(2):487–502, 1984.

[19] M. Bardi and I. Capuzzo-Dolcetta. *Optimal Control and viscosity solutions of Hamilton-Jacobi-Bellman equations.* Birkhäuser Boston, 1997.

[20] O. Bokanowski, E. Cristiani, and H. Zidani. An efficient data structure and accurate scheme to solve front propagation problems. *J. Sci. Comput.*, 42(10):251–273, 2010.

[21] H. Ishii and S. Koike. A new formulation of state constraint problems for first order PDEs. *SIAM J. Control and Optimization*, 34(2):554–571, 1996.

[22] H. M. Soner. Optimal control with state space constraint. *SIAM Journal of Control and Optimization*, 24(3):552–561, 1986.

[23] D. Drake, Ming Xin, and S. N. Balakrishnan. Reusable launch vehicle guidance and control : New nonlinear control technique for ascent phase of reusable launch vehicles. *Journal of guidance, control, and dynamics*, 27(6):938–948, 2004.

[24] O. Bokanowski, A. Briani, and H. Zidani. Minimum time control problems for non autonomous differential equations. *Systems & Control Letters*, 58(10-11):742–746, 2009.

[25] O. Bokanowski and H. Zidani. Anti-diffusive schemes for linear advection and application to Hamilton-Jacobi-Bellman equations. *Journal of Scientific Computing*, 30(1):1–33, 2007.

[26] O. Bokanowski, S. Martin, R. Munos, and H. Zidani. An anti-diffusive scheme for viability problems. *Applied Numer Math*, 56(9):1147–1162, 2006.

[27] O. Bokanowski, N. Megdich, and H. Zidani. Convergence of a non-monotone scheme for Hamilton-Jacobi-Bellman equations with discontinuous initial data. *Numerische Mathematik*, 115(1):1–44, 2010.

[28] O. Bokanowski, N. Forcadel, and H. Zidani. L^1-error estimate for numerical approximations of Hamilton-Jacobi-Bellman equations in dimension 1. *Mathematics of Computations*, 79:1395–1426, 2010.

[29] A. Briani and H. Zidani. Characterisation of the value function of final state constrained control problems with BV trajectories. *Research report,* http://hal.archives-ouvertes.fr/hal-00457804/en/, 2009.

[30] E. Cristiani and P. Martinon. Initialization of the shooting method via the Hamilton-Jacobi-Bellman approach. *J. Optim. Theory Appl. (electronic).*