Payload optimization for a multi-stage launcher SSO mission using the HJB approach

Olivier Bokanowski
University Paris Diderot, LJLL and Ensta ParisTech

joint work with
Eric Bourgeois, Anya Désilles, Hasnaa Zidani

Outline

Outline

SSO mission - Ariane V

Aim

Maximize the payload m_0 to be steered from the Earth (Kourou) to a Sun-Synchronous Orbit (SSO).

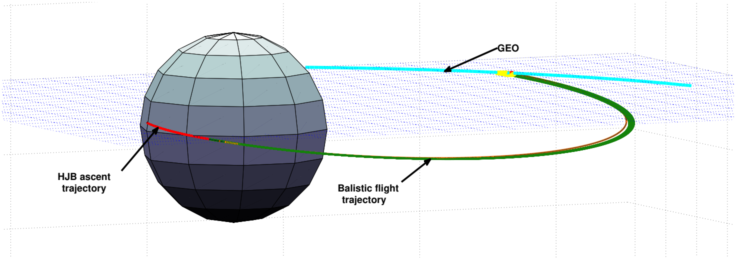
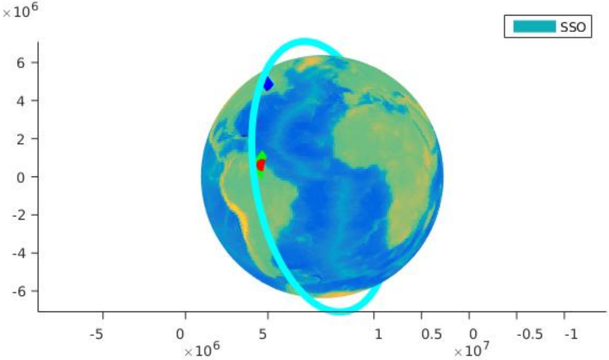


Figure: GEO mission with a ballistic flight



Sun-Synchonous Orbit (SSO)

from GEO to SSO

- Previous successfully solved mission : GEO target (equatorial), reference trajectory given by CNES
- This SSO mission : **no reference trajectory**.

References:

- GEO: Bokanowski - Bourgeois - Désilles - Zidani
  Global optimization approach for the climbing problem of multi-stage launchers
  Preprint HAL 2016
- SSO: Bokanowski - Bourgeois - Désilles - Zidani
  IFAC 2017

First old HJB attempts in our team

- 3D/4D models: Bokanowski, Cristiani, Zidani and A-J. Varin (CNES), $\simeq$ 2010
- use of the "Ultra Bee" antidiffusive scheme for solving front propagation, with efficient sparse dynamic structure for encoding the front.

- The physical model involves 6+1 state variables, the position $\overrightarrow{X}$ of the launcher in the 3D space, its velocity $\overrightarrow{V}$ and its mass M:

$$\mathbf{y} := (\mathbf{X}, \mathbf{V}, \mathbf{M}).$$

- The forces acting on the rocket are: Gravity $M\overrightarrow{g}$, Thrust $\overrightarrow{F_T}$, Drag $\overrightarrow{F_D}$, and Coriolis forces due to the relative reference frame attached to the Earth and which is non-inertial.

$\Rightarrow$ Newton's Law: $\dfrac{d\overrightarrow{X}}{dt} = \overrightarrow{V}$ and

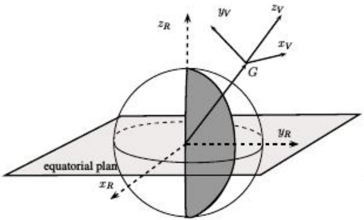
$$\frac{d\overrightarrow{V}}{dt} = \overrightarrow{g} + \frac{\overrightarrow{F_T}}{M} + \frac{\overrightarrow{F_D}}{M} - 2\overrightarrow{\Omega} \wedge \overrightarrow{V} - \overrightarrow{\Omega} \wedge (\overrightarrow{\Omega} \wedge \overrightarrow{X}),$$

- The launcher is controlled by means of:

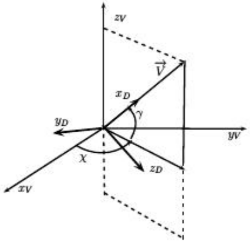
  - *launch parameters* $p = (\psi, \omega)$
  - *incidence and sideslip angles* $\alpha(t), \delta(t)$

- No bank angle : $\mu(t) \simeq 0$

Referential frame : spherical coordinates



(a) Orientation of the local vertical frame $\mathcal{R}_V$



(b) Dynamic frame $\mathcal{R}_D$

$$\overrightarrow{OG} \equiv (r, L, \ell) \qquad \overrightarrow{V} \equiv (v, \chi, \gamma)$$

Vertical frame $\mathcal{R}_V$

$r = \|\overrightarrow{OG}\| =$ altitude, $L =$ longitude, $\ell =$ latitude

Dynamic frame $\mathcal{R}_D$

$v = \|\overrightarrow{V}\|$ velocity modulus

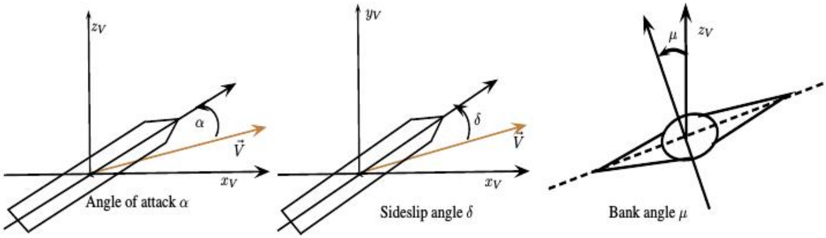


Figure 3. Angles of the launcher

Assumptions:

- $\mu = 0$
- the thrust force coincide with the axis of the launcher.

Controls:

- α: incidence angle $=$ angle between thrust $\overrightarrow{F_T}$ and $\overrightarrow{V}$.
- δ: sideslip angle $=$ the angle between $\overrightarrow{F_T}$ and $\overrightarrow{k_v}$.

The related equation - spherical coordinates

$$
\begin{aligned}
\frac{dr}{dt} &= v \sin\gamma \\
\frac{dL}{dt} &= \frac{v}{r}\frac{\cos\gamma \sin\chi}{\cos\ell} \qquad \text{Longitude's dynamics} \\
\frac{d\ell}{dt} &= \frac{v}{r}\cos\gamma\cos\chi \\
\frac{dv}{dt} &= -g_r \sin\gamma + g_\ell \cos\gamma\cos\chi + \frac{F_T(r)\cos\alpha\cos\delta}{M(t)} + \frac{F^D(r,v,\alpha)}{M(t)} + F_v^C \\
\frac{d\gamma}{dt} &= \cos\gamma\left(\frac{v}{r} - \frac{g_r}{v}\right) - \sin\gamma\cos\chi\frac{g_\ell}{v} - \frac{F_T(r)\sin\alpha}{M(t)v} + F_\gamma^C \\
\frac{d\chi}{dt} &= -\frac{g_\ell \sin\chi}{v\cos\gamma} - \frac{v\cos\gamma\tan\ell\sin\chi}{r} + \frac{F_T(r)\cos\alpha\sin\delta}{M(t)\,v\cos\gamma} + F_\chi^C \\
\frac{dM}{dt} &= -\beta(t) \qquad \text{Mass's dynamics}
\end{aligned}
$$

where

. *Drag forces F_D vanish ($F_D \simeq 0$) out of the atmosphere*

. *g_r, g_ℓ are components of the gravitational field with J_2 corrections*

. *$(F_v^C, F_\gamma^C, F_\chi^C)$ are Coriolis' forces in the dynamic frame $\mathcal{R}_D$.*

The related equation - spherical coordinates

$$\frac{dr}{dt} = v \sin\gamma$$

$$\begin{aligned}
\frac{d\ell}{dt} &= \frac{v}{r}\cos\gamma\cos\chi \\
\frac{dv}{dt} &= -g_r \sin\gamma + g_\ell \cos\gamma\cos\chi + \frac{F_T(r)\cos\alpha\cos\delta}{M(t)} + \frac{F^D(r,v,\alpha)}{M(t)} + F_v^C \\
\frac{d\gamma}{dt} &= \cos\gamma\left(\frac{v}{r} - \frac{g_r}{v}\right) - \sin\gamma\cos\chi\frac{g_\ell}{v} - \frac{F_T(r)\sin\alpha}{M(t)v} + F_\gamma^C \\
\frac{d\chi}{dt} &= -\frac{g_\ell\sin\chi}{v\cos\gamma} - \frac{v\cos\gamma\tan\ell\sin\chi}{r} + \frac{F_T(r)\cos\alpha\sin\delta}{M(t)\,v\cos\gamma} + F_\chi^C \\
\frac{dm_0}{dt} &= 0
\end{aligned}$$

where

. *Drag forces F_D vanish ($F_D \simeq 0$) out of the atmosphere*

. *g_r, g_ℓ are components of the gravitational field with J_2 corrections*

. *$(F_v^C, F_\gamma^C, F_\chi^C)$ are Coriolis' forces in the dynamic frame $\mathcal{R}_D$.*

g_r and g_ℓ are components of the gravitational field

$$g_r := \frac{\mu}{r^2}\left(1 + J_2\left(\frac{r_T}{r}\right)^2 (1 - 3\sin^2\ell)\right)$$
$$g_\ell := -2\frac{\mu}{r^2} J_2\left(\frac{r_T}{r}\right)^2 \sin\ell\cos\ell,$$

$(F_v^C, F_\gamma^C, F_\chi^C)$ are Coriolis' forces

$$\begin{aligned}
F_v^C &:= \Omega^2 r\cos\ell(\sin\gamma\cos\ell - \cos\gamma\sin\ell\cos\chi) \\
F_\gamma^C &:= 2\Omega\cos\ell\sin\chi + \frac{\Omega^2 r}{v}\cos\ell(\cos\gamma\cos\ell + \sin\gamma\sin\ell\cos\chi) \\
F_\chi^C &:= \frac{\Omega^2 r}{v}\frac{\sin\ell\cos\ell\sin\chi}{\cos\gamma} - 2\Omega\left(\sin\ell - \tan\gamma\cos\ell\cos\chi\right)
\end{aligned}$$

The state is represented by $(x, m) = (r, \ell, v, \gamma, \chi, m) \in \mathbb{R}^6$.

Constraints and target set

- Low altitude target orbit $\Rightarrow$ special constraint on the dynamic thermal flow has to be satisfied during the phase 2 of the flight (starting at ignition of E_2):
$$0.5\,\rho(r)v^3 \leq 555 \;Wm^{-2} \tag{1}$$
where $\rho(r)$ is the density of the atmosphere at altitude r. Then the set of state constraints, in $\mathbb{R}^6$, is defined by:
$$\mathcal{K} \quad := \quad \Big\{ y = (x, m) \in \mathbb{R}^6, \quad 0.5\,\rho(r)v^3 \leq 555 \Big\}. \tag{2}$$
- The target set, in $\mathbb{R}^6$, is defined by:
$$\mathcal{C} \quad := \quad \Big\{ y = (x, m) \in \mathbb{R}^6, \text{ s.t.}$$
$$e(x) = 0,\; a(x) = 800,\; i(x) = 98.6^\circ,\; m \geq 0 \Big\}$$
where eccentricity $e(x)$, major semi-axis $a(x)$ and inclination $i(x)$ are known functions of the position $x = (r, \ell, v, \gamma, \chi) \in \mathbb{R}^5$.

Flight sequence

- **Phase 0 - Atmospheric flight**: The trajectory's profile depends only on the shooting azimut ψ and the angular velocity ω: $p = (\psi, \omega) \in P$.
- **Phase 1&2 - First boost until GTO**: The trajectory depends on the control input $\mathbf{u} := (\alpha(\cdot), \delta(\cdot))$. Drag force $F_D = 0$. The duration of the first boost of the second engine E_2 is unknown
- **Phase 3 - Ballistic flight**: All engines are off. The duration of this phase, τ_B, is unknown.
- **Phase 4 - Second boost** starts when the engine E_2 is ignited again and it lasts until the total consumption of the propellant of E_2. (The final time $\mathbf{t_f}$ is unknown, but is determined by the previous durations.)

Mass dynamics

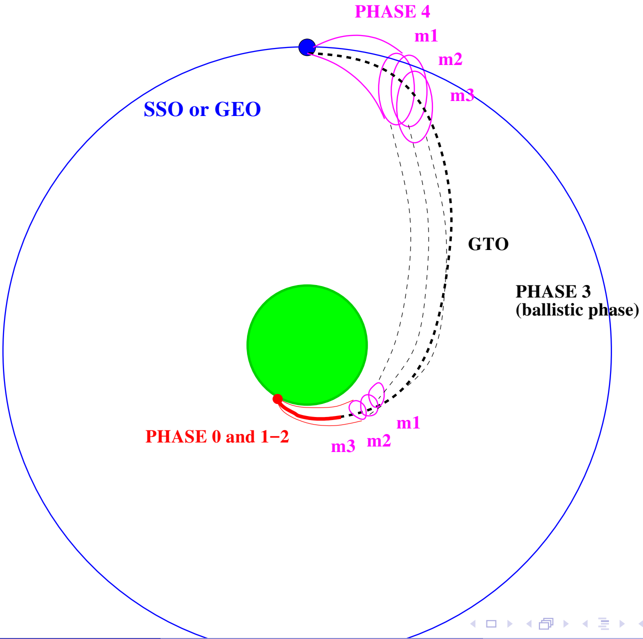
- The evolution of the mass can be summarized as follows

	Phase 0 (atmosph.)	Phase 1	Phase 2	Phase 3 (ballistic)	Phase 4
$\dot{m}_1(t)$	$-\beta_{2B}$	0	0	0	0
$\dot{m}_2(t)$	$-\beta_{E1}(t)$	$-\beta_{E1}(t)$	0	0	0
$\dot{m}_3(t)$	0	0	$-\beta_{E2}$	0	$-\beta_{E2}$
$(time)$		t_0	t_1	t_2	t_3

  where β_{2B}, β_{E1} and β_{E2} are the mass flow rates for the boosters, the first and the second stage.

- At the changes of phases, we have a (not negligible) discontinuity in the rocket's mass (corresponding to the ejection of the boosters or of the E1 stage)

Summary of the phases



Optimization problem

The considered problem is to determine :

- the shooting parameters (ψ, ω),
- the duration of the first boost of the second engine $t_2 - t_1$
- the duration of the ballistic flight τ_B
- the control laws (during phases 1, 2 and 4)

in order to

maximize the payload mass m_0, reach the SSO orbit.

In particular all the propellant mass has to be consumed at the end of the mission.

Outline

- **Phase 0**: First, we consider the set of all possible positions at t_0:

$$X_0 := \{\mathbf{y}^p(t_0);\ \text{for}\ \ p = (\psi, \omega) \in P\}.$$

- **Phases 1,2&4:** The total consumption time T is known. Hence t_f and τ_B must satisfy

$$t_f = t_0 + T + \tau_B.$$

We introduce a consumption time variable s such that $t \in [0, t_f] \to s \in [0, T]$:

$$s(t) := \begin{cases} t - t_0 & \text{if } t \in [t_0, t_2[\\ s_* := t_2 - t_0 & \text{constant, if } t \in [t_2, t_2 + \tau_B[\\ t - \tau_B - t_0 & \text{if } t \in]t_2 + \tau_B, t_f] \end{cases}$$

Set $s_* := t_2 - t_1$ be the duration of the first boost.

- **Phase 3-Ballistic flight**: The launcher's motion is governed by an uncontrolled and autonomous ODE $\dot{z}(t) = \varphi(z(t)), z(0) = z_0$. Let Φ be the transfer function, i.e $z(t) = \Phi(t, z_0)$.

$$y(s_*^+) = \Phi(\tau_B; y(s_*^-)).$$

The control problem $(\mathcal{P})$ can be formulated as (for a given $y \in X_0$):

$$\sup \ \mathbf{m}_y^{\mathbf{u}}(T),$$

$$\text{there exists } s_* \in [s_2^{min}, s_2^{max}], \quad \tau_B \in [\tau_B^{min}, \tau_B^{max}].$$

$$\begin{cases} \dot{\mathbf{y}}_y^{\mathbf{u}}(s) = f(s, \mathbf{y}_y^{\mathbf{u}}(s), \mathbf{u}(s)), & s \in [0, s_*[\\ \mathbf{y}_y^{\mathbf{u}}(s_*^+) = \Phi(\tau_B, \mathbf{y}_y^{\mathbf{u}}(s_*^-)), & \\ \dot{\mathbf{y}}_y^{\mathbf{u}}(s) = f(s, \mathbf{y}_y^{\mathbf{u}}(s), \mathbf{u}(s)), & s \in]s_*, T] \\ \dot{\mathbf{y}}_y^{\mathbf{u}}(0) = y & \end{cases}$$

$$\mathbf{y}_y^{\mathbf{u}}(s) \in \mathcal{K}, \quad \forall s \in [0, T],$$

$$\mathbf{y}_y^{\mathbf{u}}(T) \in \mathcal{C},$$

$$\mathbf{u}(s) \in U \ \text{ a.e. } s \in [0, T]$$

- The target $\mathcal{C}$ corresponds to the GEO orbit
- $\mathcal{K}$ represents an admissible constraints set on $(0, T)$ (bounds on the heat flux and other physical constraints)

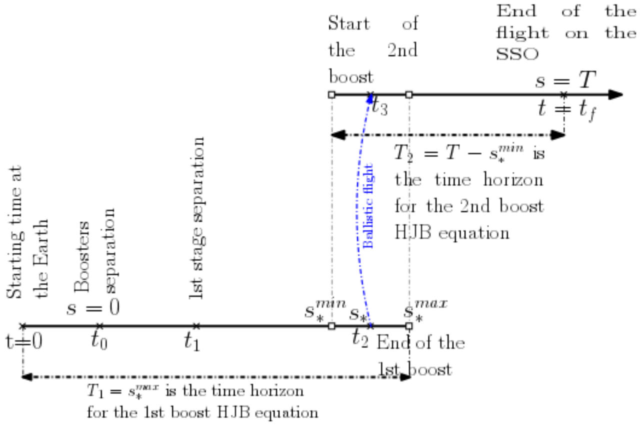


Figure: Relation between physical time t and "consumption" time s

Outline

Level set approach for reachability:

- *Osher, Sethian* - J. Comput. Phys., 1988
  Fronts propagating with curvature-dependent speed: algorithms based on Hamilton-Jacobi formulations
- *Mitchell, Bayen, Tomlin* - IEEE Trans. Automat. Control, 2005
  A time-dependent Hamiliton-Jacobi formulation of reachable sets for continuous dynamic games
- *O. Bokanowski, N. Forcadel and H. Zidani* SICON, 2010
  "Reachability and minimal times for state constrained nonlinear problems without any controllability assumption"
- *Assellaou, Bokanowski, Desilles, Zidani* - IFAC Proceedings 2016
  "Windshear problem"
  $\Rightarrow$ SIMPLE BOUNDARY CONDITIONS FOR THE HJ-PDE

Consider the controlled system:

$$\begin{cases} \dot{\mathbf{y}}^{\mathbf{u}}_{s,y}(\xi) = f(\xi, \mathbf{y}^{\mathbf{u}}_{s,y}(\xi), \mathbf{u}(\xi)), & \xi \in (s, T), \\ \mathbf{y}^{\mathbf{u}}_{s,y}(s) = y, \end{cases} \tag{3}$$

$$\mathbf{u}(\xi) \in U, \quad \text{a.e } \xi \in (s, T).$$

where U is a compact set in $\mathbb{R}^2$.

We use **level set functions** to represent feasibility:

We design $\varphi : \mathbb{R}^6 \to \mathbb{R}$ such that

$$\varphi(y) \leq 0 \quad \Leftrightarrow \quad y \in \mathcal{C}.$$

In the same way, we design an "obstacle" function $g : \mathbb{R}^{\to}\mathbb{R}$ such that

$$g(y) \leq 0 \quad \Leftrightarrow \quad y \in \mathcal{K}.$$

Ex: $\varphi(y) = d_{\mathcal{C}}(y)$, $g(y) := d_{\mathcal{K}}(y)$ (signed distance functions).

Consider the controlled system:

$$\begin{cases} \dot{\mathbf{y}}^{\mathbf{u}}_{s,y}(\xi) = f(\xi, \mathbf{y}^{\mathbf{u}}_{s,y}(\xi), \mathbf{u}(\xi)), & \xi \in (s,T), \\ \mathbf{y}^{\mathbf{u}}_{s,y}(s) = y, \end{cases} \tag{3}$$

$$\mathbf{u}(\xi) \in U, \quad \text{a.e } \xi \in (s,T).$$

where U is a compact set in $\mathbb{R}^2$.

We use **level set functions** to represent feasibility:

We design $\varphi : \mathbb{R}^6 \to \mathbb{R}$ such that

$$\varphi(y) \leq 0 \quad \Leftrightarrow \quad y \in \mathcal{C}.$$

In the same way, we design an "obstacle" function $g : \mathbb{R}^{\to}\mathbb{R}$ such that

$$g(y) \leq 0 \quad \Leftrightarrow \quad y \in \mathcal{K}.$$

Ex: $\varphi(y) = d_{\mathcal{C}}(y)$, $g(y) := d_{\mathcal{K}}(y)$ (signed distance functions).

Consider the controlled system:

$$\begin{cases} \dot{\mathbf{y}}^{\mathbf{u}}_{s,y}(\xi) = f(\xi, \mathbf{y}^{\mathbf{u}}_{s,y}(\xi), \mathbf{u}(\xi)), & \xi \in (s,T), \\ \mathbf{y}^{\mathbf{u}}_{s,y}(s) = y, \end{cases} \tag{3}$$

$$\mathbf{u}(\xi) \in U, \quad \text{a.e } \xi \in (s,T).$$

where U is a compact set in $\mathbb{R}^2$.

We use **level set functions** to represent feasibility:

We design $\varphi : \mathbb{R}^6 \to \mathbb{R}$ such that

$$\varphi(y) \le 0 \quad \Leftrightarrow \quad y \in \mathcal{C}.$$

In the same way, we design an "obstacle" function $g : \mathbb{R}^{\to}\mathbb{R}$ such that

$$g(y) \le 0 \quad \Leftrightarrow \quad y \in \mathcal{K}.$$

Ex: $\varphi(y) = d_{\mathcal{C}}(y)$, $g(y) := d_{\mathcal{K}}(y)$ (signed distance functions).

A reachability problem for the second boost on $[s_*, T]$

➢ Consider the following control problem:

$$w_0(x,t) = \inf_{\mathbf{u}\in\mathcal{U}_{ad}} d_{\mathcal{C}}(\mathbf{y}^{\mathbf{u}}_{s,y}(T)) \bigvee \max_{\xi\in(s,T)} d_{\mathcal{K}}(\mathbf{y}^{\mathbf{u}}_{s,y}(\theta))$$

The function w_0 is Lipschitz continuous, and, on $[s_*^{min}, T] \times \mathbb{R}^6$:

$$\min\Big(-\partial_s w_0(s,y) + H(s,y,D_y w_0(s,y)), w_0(s,y) - d_{\mathcal{K}}(y)\Big) = 0,$$

$$w_0(T,y) = d_{\mathcal{C}}(y) \bigvee d_{\mathcal{K}}(y),$$

where $H(s,y,q) := \max_{u\in U} \big(-f(s,y,u)\cdot q\big)$

➢ Moreover, we have:

$$w_0(s,y) \le 0 \quad \Leftrightarrow \quad \forall\varepsilon > 0,\ \exists \mathbf{u}_\varepsilon \in \mathcal{U}_{ad},\ d_{\mathcal{C}}(\mathbf{y}^{u_\varepsilon}_{s,y}(T)) \le \varepsilon \text{ and } d_{\mathcal{K}}(\mathbf{y}^{u_\varepsilon}_{s,y}(\xi)) \le \varepsilon\ \forall \xi \in [s,T].$$

A reachability problem associated to (P)

Now, consider the following control problem:

$$w(s,y) = \inf_{\substack{\mathbf{u}\in\mathcal{U}_{ad},\\ \tau_B\in[\tau_B^{min},\tau_B^{max}],\\ s_*\in[s_*^{min},s_*^{max}],}} \left\{ d_{\mathcal{C}}(\mathbf{y}^{\mathbf{u}}_{s,y}(T)) \bigvee \max_{\xi\in[s,T]} d_{\mathcal{K}}(\mathbf{y}^{\mathbf{u}}_{s,y}(\xi)) \right\}.$$

where

$$\begin{cases} \dot{\mathbf{y}}^{\mathbf{u}}_y(s) = f(s,\mathbf{y}^{\mathbf{u}}_y(s),\mathbf{u}(s)), & s\in[0,s_*[\\ \mathbf{y}^{\mathbf{u}}_y(s_*^+) = \Phi(\tau_B,\mathbf{y}^{\mathbf{u}}_y(s_*^-)), & \\ \dot{\mathbf{y}}^{\mathbf{u}}_y(s) = f(s,\mathbf{y}^{\mathbf{u}}_y(s),\mathbf{u}(s)), & s\in]s_*,T] \\ \dot{\mathbf{y}}^{\mathbf{u}}_y(0) = y\in X_0\times[0,+\infty[& \end{cases}$$

Let the following operator:

$$\mathcal{M}w_0(s,y) := \min_{\tau \in [\tau_B^{min}, \tau_B^{max}]} w_0(s, \Phi(\tau, x)).$$

Theorem

- The function w is Lipschitz continuous on $[0, T] \times \mathbb{R}^6$

Let the following operator:

$$\mathcal{M}w_0(s,y) := \min_{\tau \in [\tau_B^{min}, \tau_B^{max}]} w_0(s, \Phi(\tau, x)).$$

Theorem

- The function w is Lipschitz continuous on $[0, T] \times \mathbb{R}^6$
- $w = w_0$ on $[s_*^{max}, T] \times \mathbb{R}^6$

Let the following operator:

$$\mathcal{M}w_0(s,y) := \min_{\tau\in[\tau_B^{min},\tau_B^{max}]} w_0(s,\Phi(\tau,x)).$$

Theorem

- The function w is Lipschitz continuous on $[0,T]\times\mathbb{R}^6$
- $w = w_0$ on $[s_*^{max},T]\times\mathbb{R}^6$
- Then w is the unique continuous viscosity solution of the following HJB equation on $[0,s_*^{max}]\times\mathbb{R}^6$:

$$\min\left\{\max\left(-\partial_s w + H(s,y,D_y w),\ w-\mathcal{M}w_0(s,y)\right),\ w-d_{\mathcal{K}}(y)\right\} = 0,$$
$$\text{on } (s_*^{min},s_*^{max})\times\mathbb{R}^6,$$

$$\min\left(-\partial_s w + H(s,y,D_y w), w-d_{\mathcal{K}}(y)\right) = 0, \quad \text{on } (0,s_*^{min})\times\mathbb{R}^6,$$

$$w(s_*^{max},y) = w_0(s_*^{max},y), \quad y\in\mathbb{R}^6.$$

Procedure for solving $(\mathcal{P})$

- *STEP 1.* Compute the set X_0 for a large sample of parameters (ψ, ω).

- *STEP 2.* Solve the first HJB equation to get an approximation of w_0.

- *STEP 3.* Solve the HJB equation to obtain an approximation of w.

- *STEP 4.* Define on the set X_0 the function

$$m^*(x) = \sup\{m \mid w(0,(x,m)) \leq 0\}. \tag{4}$$

This function corresponds to the biggest payload mass that is possible to steer to the GEO starting from x. Finally, the *optimal mass* is given by:

$$m_{opt} = \sup_{x \in X_0} m^*(x).$$

- *STEP 5: Reconstruction of an optimal trajectory.*

Outline

1) Spherical - Cartesian (SC) coordinates

Spherical coordinates for the position $X \equiv (r, L, \ell)$

cartesian coordinates for $\overrightarrow{V}$ in the vertical frame $\mathcal{R}_V$: $V \equiv (v_r, v_L, v_\ell)$, equivalently:

$$V = v_\ell i_v + v_L j_v + v_r k_v$$

$$\begin{aligned}
\frac{d\ell}{dt} &= \frac{v_\ell}{r} \\
\frac{dL}{dt} &= \frac{v_L}{r\cos\ell} \\
\frac{dr}{dt} &= -v_r \\
\frac{dv_\ell}{dt} &= \frac{F_T(r)}{M}\cos\theta\,\cos\mu + g_\ell - \Omega^2 r\cos\ell\,\sin\ell - 2\Omega v_L\sin\ell - \frac{v_L^2\tan\ell - v_\ell v_r}{r} \\
\frac{dv_L}{dt} &= \frac{F_T(r)}{M}\cos\theta\,\sin\mu + 2\Omega(v_r\cos\ell + v_\ell\sin\ell) + \frac{v_L(v_\ell\tan\ell + v_r)}{r} \\
\frac{dv_r}{dt} &= -\frac{F_T(r)}{M}\sin\theta + g_r - \Omega^2 r\cos^2\ell - 2\Omega v_L\cos\ell - \frac{v_L^2 + v_\ell^2}{r} \\
\frac{dm_0}{dt} &= 0
\end{aligned}$$

1) Spherical - Cartesian (SC) coordinates

Spherical coordinates for the position $X \equiv (r, L, \ell)$

cartesian coordinates for $\overrightarrow{V}$ in the vertical frame $\mathcal{R}_V$: $V \equiv (v_r, v_L, v_\ell)$, equivalently:

$$V = v_\ell i_v + v_L j_v + v_r k_v$$

$$\frac{d\ell}{dt} = \frac{v_\ell}{r}$$

$$\begin{aligned}
\frac{dr}{dt} &= -v_r \\
\frac{dv_\ell}{dt} &= \frac{F_T(r)}{M}\cos\theta\,\cos\mu + g_\ell - \Omega^2 r\cos\ell\,\sin\ell - 2\Omega v_L \sin\ell - \frac{v_L^2\tan\ell - v_\ell v_r}{r} \\
\frac{dv_L}{dt} &= \frac{F_T(r)}{M}\cos\theta\,\sin\mu + 2\Omega(v_r\cos\ell + v_\ell\sin\ell) + \frac{v_L(v_\ell\tan\ell + v_r)}{r} \\
\frac{dv_r}{dt} &= -\frac{F_T(r)}{M}\sin\theta + g_r - \Omega^2 r\cos^2\ell - 2\Omega v_L\cos\ell - \frac{v_L^2 + v_\ell^2}{r} \\
\frac{dm_0}{dt} &= 0
\end{aligned}$$

2) Analytic expression for the Numerical Hamiltonian

$$
\begin{aligned}
H(t,x,z,q) &= \max_{u=(\alpha,md)} \Big(-F(x,u)\cdot q \Big) \\
&= \max_{\substack{\alpha\in[\alpha_{min},\alpha_{max}] \\ \delta\in[\delta_{min},\delta_{max}]}} \Big(b_1\cos(\alpha)\cos(\delta) + b_2\cos(\alpha)\sin(\delta) + b_3\sin(\alpha) \Big) \\
&\qquad + C(t,x,z,q)
\end{aligned}
$$

where

- $b_1 \equiv \frac{F_T(r)}{M} q_3, \quad b_2 \equiv \frac{F_T(r)}{M\,v} q_4, \quad b_3 \equiv \frac{F_T(r)}{M\,v\,\cos\gamma}\, q_5.$
- $C(t,x,z,q)$ does not depend neither on α nor δ,
- q involves finites differences (ENO2) estimates of derivatives ($q \simeq Dw$)

$\Rightarrow$ **A simple analytical expression for (α^*, δ^*) is obtained.**

3) State constraints & domain reduction

REF: The HJB approach for the optimal control of an abort landing problem, CDC 2016, 55th IEEE, Assellaou, Bokanowski, Desilles, Zidani.
Idea: domain reduction technique

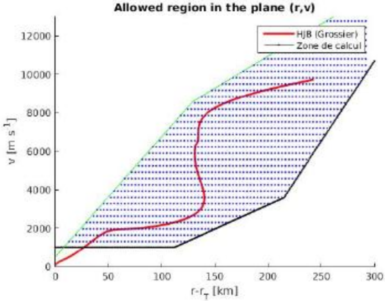


Figure: A priori domain reduction

Details for boundary conditions

Consider the simplified problem $y(t) \in \mathbb{R}$ with state constraint to be enforced:

$$y(t) \in [a, b]$$

Introduce some $\eta > 0$ and the computational domain

$$\Omega_\eta := [a - \eta, b + \eta].$$

Define the L.S. function $g : \mathbb{R} \to R$

$$g(x) := \min(\epsilon, \max(x - b, a - x)).$$

Define the OCP

$$w(t, x) = \inf_u \varphi(y^u_{t,x}(T)) \bigvee \max_{\theta \in (t,T)} g(y^u_{t,x}(T))$$

Then

$$\min(-w_t + H(x, \nabla w),\ w - g) = 0, \quad x \in \mathbb{R}$$
$$w(T, x) = \varphi(x) \bigvee g(x)$$

Furthermore, assuming that $\varphi(.) \leq \eta$, it holds

$$x \notin \Omega_\eta = (a - \eta, b + \eta) \quad \Rightarrow \quad \eta \geq w(t, x) \geq g(x) = \eta.$$
$$\Rightarrow \quad w(t, x) = \eta$$

Efficient computing

EFF-1/ Scalable scheme
EFF-2/ Minimize tests
EFF-3/ Parallelizability
EFF-4/ Avoid boundary testing

EFF-1/ Scalable scheme
Consider the PDE:

$$\min(v_t + H(x, \nabla v),\ v - g(x)) = 0$$

FD Scheme on mesh (x_i):

$$\min\left(\frac{u_i^{n+1} - u_i^n}{\Delta t} + h(u^n)_i,\ u_i^{n+1} - g(x_i)\right) = 0,$$

which leads to

$$u_i^{n+1} = \max\Big(\underbrace{u_i^n - \Delta t\, h(u^n)_i}_{u_i^{n+1,FD}},\ g(x_i)\Big).$$

Numerical hamiltonian h**:** typically, a finite difference scheme of ENO type.
Example: ENO of second order needs only 5 points $(i, i \pm 1, i \pm 2)$ in each direction, total = $1 + 4d = O(d)$ neighboring points

EFF-2/ Minimize tests in coding:

```
void HJB_FD::ENO2_RK1(double t, double deltat, double* vin, double* vout)
{
  // INITIALISATION // PERIODICITY & BORDER PREPARATION
  for(j=0;j<ranksize;j++){

    //- Dvnum is global

    i    = rank[j];      //- using tab rank
    vi   = vin[i];
    for(d=0;d<dim;d++){
      v1 = vin[i -   mesh->out_neighbors[d]];
      v3 = vin[i - 2*mesh->out_neighbors[d]];
      v2 = vin[i +   mesh->out_neighbors[d]];
      v4 = vin[i + 2*mesh->out_neighbors[d]];
      h  = divdx[d];
      vv = (v2-2.*vi+v1);
      Dvnum[2*d]   = ((vi-v1) + .5*minmod((vi-2.*v1+v3),vv))*h;
      Dvnum[2*d+1]= ((v2-vi) - .5*minmod((vi-2.*v2+v4),vv))*h;
    }
    double *xx=(mesh->*(mesh->getcoords))(i);
    vout[i] = vi - deltat * (*this.*Hnum)(xx,Dvnum,t);
  }
  return;
}
```

EFF-3/ Paralellizability: OpenMP

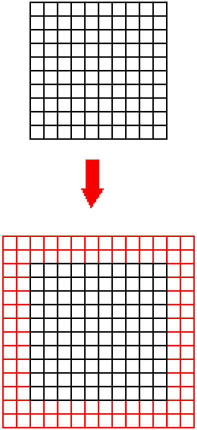
```
void HJB_FD::ENO2_RK1_omp(double t, double deltat, double* vin, double* vout)
{
  // INITIALISATION // PERIODICITY & BORDER PREPARATION

  #pragma omp parallel for num_threads(OMP_NUM_THREADS)
  private(d, i, j, vi, v1, v2, v3, v4, vv, h)
  shared(t,deltat,vin,vout) default(none)

  for(j=0;j<ranksize;j++){"
    double dvnum[2*dim];

    i   = rank[j];
    vi  = vin[i];
    for(d=0;d<dim;d++){
      v1 = vin[i -   mesh->out_neighbors[d]];
      v3 = vin[i - 2*mesh->out_neighbors[d]];
      v2 = vin[i +   mesh->out_neighbors[d]];
      v4 = vin[i + 2*mesh->out_neighbors[d]];
      h  = divdx[d];
      vv = (v2-2.*vi+v1);
      dvnum[2*d]   = ((vi-v1) + .5*minmod((vi-2.*v1+v3),vv))*h;
      dvnum[2*d+1]= ((v2-vi) - .5*minmod((vi-2.*v2+v4),vv))*h;
    }
    double *xx=(mesh->*(mesh->getcoords))(i);
    vout[i] = vi - deltat * (*this.*Hnum)(xx,dvnum},t);
  }
  return;
}
```

EFF-4/ Avoid boundary testing: boundary ENLARGMENT



Numerical scheme

- We use the ROC-HJ c++ solver

  http://uma.ensta-paristech.fr/files/ROC-HJ/

and in particular a finite difference scheme, with Open-MP parallelization techniques. (developpers: O.B., J. Zhao, A. Desilles, H. Zidani)

- Other solvers available: I. Mitchell's Matlab toolbox, ...

Outline

Approximation of the optimal trajectories

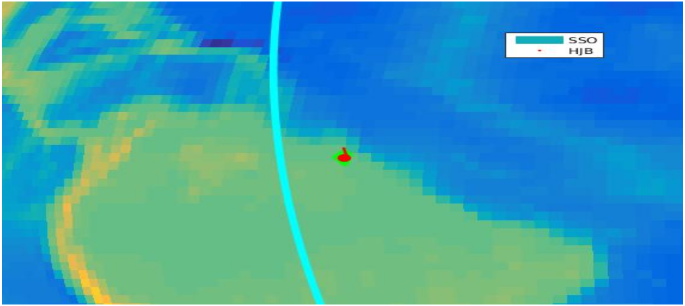
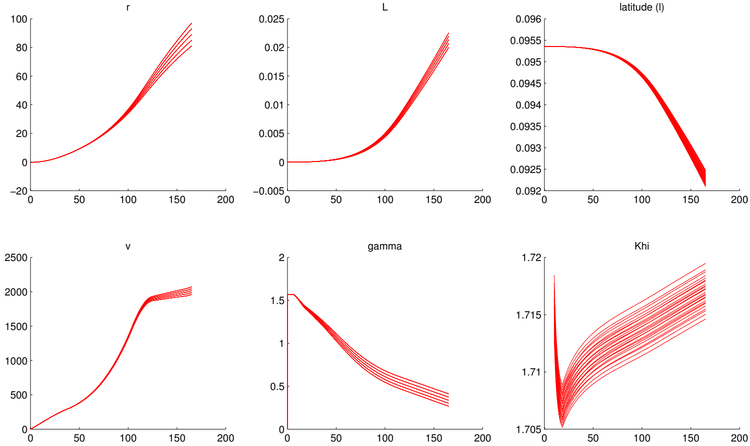


Figure: trajectory - atmospheric part

Approximation of X_0 (Step 1)

$$X_0 := \{\mathbf{y}^p(t_1) \mid p \in P,\ \ \dot{\mathbf{y}}^p(t) = f(p, \mathbf{y}^p(t)),\ \mathbf{y}^p(0) = y_0\},$$



Optimization of the shooting parameter p

Assume we have the knowledge of the value $w(0, x, z)$:

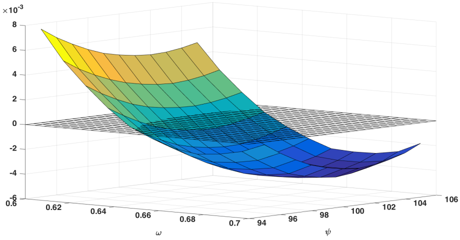


Figure: Values of $w(0, x, m^*(x))$ with $x = \Gamma(p)$, for different shooting parameters $p = (\psi, \omega)$.

$\Rightarrow$ we determine an optimal mass $m_{opt} = m^*$ and corresponding shooting parameters $p^* = (\psi^*, \omega^*)$.

HJB : connecting different box computations for the different phases

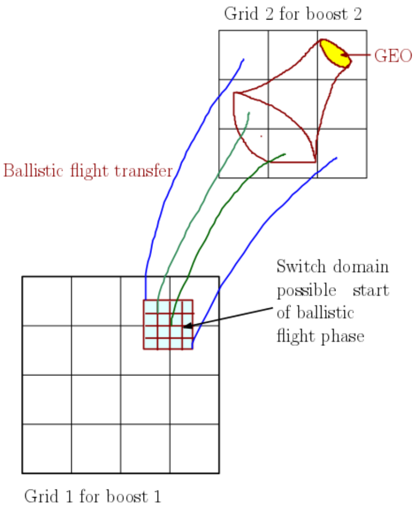


Figure: Connecting different HJB computations

HJB - Numerical computations

Grid (B1-1)	Number of points	CPU (s)
Grid 1	$20 \times 30 \times 10 \times 10 \times 8 \times 3$	900
Grid 2	$30 \times 40 \times 15 \times 15 \times 12 \times 4$	3520
Grid 3	$40 \times 60 \times 20 \times 20 \times 16 \times 5$	18900

Table: Grid sizes (for B1-1) and CPU times

Grid (B1-1)	ψ (deg)	ω (deg s^{-1})	s_* (s)	τ_B (s)	m_{opt} (kg)
Grid 1	105.00	0.69	956.15	2605.82	15449.90
Grid 2	105.00	0.69	956.44	2625.82	15563.13
Grid 3	103.99	0.69	955.41	2605.82	**15624.87**

Table: Optimal initial parameters, phase durations and payload mass

Grid	ν (deg)	r_a (km)	r_p (km)	i (deg)
Grid 1	61.48	826.32	148.60	105.25
Grid 2	61.37	848.93	142.89	102.36
Grid 3	68.98	780.66	133.36	100.28

Table: Optimal transfer orbit parameters (for ballistic phase)

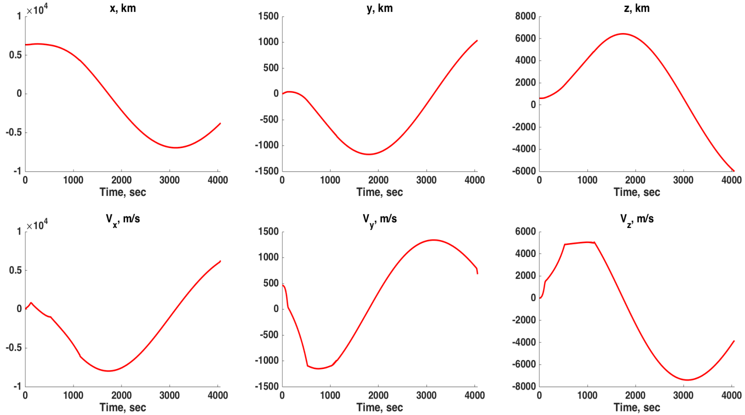


Figure: Optimal trajectory with HJB (in inertial frame)

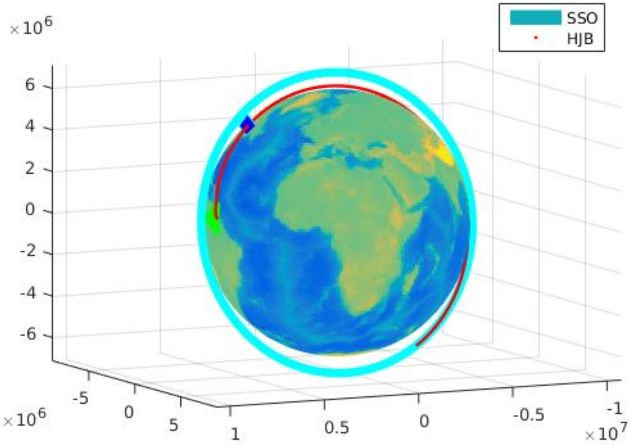


Figure: Optimal trajectory with HJB

Conclusion - Going further

- Trajectories for the SSO pb where obtained using the HJB approach

Conclusion - Going further

- Trajectories for the SSO pb where obtained using the HJB approach
- Non discussed issues: memory, diffusive/non-diffusive aspects of FD schemes, level set functions used for target and state constraints, trajectory reconstruction from state-constraint value function, ...

Conclusion - Going further

- Trajectories for the SSO pb where obtained using the HJB approach
- Non discussed issues: memory, diffusive/non-diffusive aspects of FD schemes, level set functions used for target and state constraints, trajectory reconstruction from state-constraint value function, ...
- Going further: try using the HJ computation to initialize PMP / shooting method (Cristiani-Martinon JOTA 2010)