Collocation and inversion for a reentry optimal control problem

Tobias NECKEL [1] Christophe TALBOT [2] Nicolas PETIT [3]

1 - École Polytechnique, 91128 Palaiseau Cedex France
Tobias.Neckel@cnes.fr
2 - Centre National d'Études Spatiales, CNES - DDA/SDT/SP Evry, Rond Point de l'Espace 91023 EVRY Cedex France
Christophe.Talbot@cnes.fr
3 - Centre Automatique et Systèmes, École Nationale Supérieure des Mines de Paris 60, bd Saint-Michel 75272, Paris Cedex 06, France
petit@cas.ensmp.fr

Abstract

The purpose of this article is to provide the reader with an overview of an inversion based methodology applied to a shuttle atmospheric reentry problem. The proposed method originates in the search for computationally efficient trajectory optimization as an enabling technology for versatile real-time trajectory generation. The technique is based on the nonlinear control theory notion of inversion and flatness. This point of view allows to map the system dynamics, objective, and constraints to a lower dimensional space. The optimization problem is then solved in the lower dimensional space. Eventually the optimal states and inputs are recovered from the inverse mapping.

1 Introduction

The purpose of this article is to provide the reader with an overview of an inversion based methodology applied to a shuttle atmospheric reentry problem. This problem has a 6 states, 2 controls nonlinear dynamics with terminal and initial constraints and a terminal cost function. Aerodynamics models (linear for lift and quadratic for drag) are considered. Gravity and air density are modelled according to the classic non rotating spherical earth potential and exponential models.

The proposed method originates in the search for computationally efficient trajectory optimization as an enabling technology for versatile real-time trajectory generation. Trajectory generation of unmanned aerial vehicles is an example where the tools of real-time trajectory optimization can be extremely useful. In [9, 13, 12], this new technique was presented and used to solve such problems. In [11] this methodology was applied to formation flight of microsatellites under J2 gravitational effect. Following the same ideas the real time trajectory generation of a planar missile was addressed [10] with similar drag and lift models.

The technique is based on the nonlinear control theory notion of inversion [7] and flatness [3, 4]. This point of view allows to map the system dynamics, objective, and constraints to a lower dimensional space. The optimization problem is then solved in the lower dimensional space. Eventually the optimal states and inputs are recovered from the inverse mapping.

The example treated in this report has interesting features. First it is more complex in terms of dimensionality and nonlinearities than the previously cited examples. Second the dynamics are not flat. In other words it is not possible to fully invert the system dynamics. This particular situation deserves a careful treatment of the parametrization of the states variables. Numerical results are given, and a comparison with existing techniques for this example [1] is given. In short, the proposed approach appears tracktable, but could be improved further by paying more attention to the choice of the nonlinear programming solver and the finite dimensional representation that are used.

2 Background information

In this section we present the general framework of inversion-based collocation methods for numerical solution to optimal control problems. Most of this material can be found in [13]. We address the simple single-input case which is by far the most easy and emphasizes the role of inversion.

2.1 Optimal Control Problem

Consider the single input nonlinear control system

$$\begin{aligned} &\dot{x} = f(x) + g(x)u, \\ &\mathbb{R} \ni t \mapsto x \in \mathbb{R}^n, \mathbb{R} \ni t \mapsto u \in \mathbb{R} \end{aligned} \quad (1)$$

where all vector fields and functions are smooth functions. It is desired to find a trajectory of (1) $[t_0, t_f] \ni t \mapsto (x, u)(t) \in \mathbb{R}^{n+1}$ that minimizes the cost

$$\begin{aligned} J(x, u) = &\phi_f(x(t_f), u(t_f)) + \phi_0(x(t_0), u(t_0)) \\ &+ \int_{t_0}^{t_f} L(x(t), u(t))dt, \end{aligned}$$

where L is a nonlinear function, subject to a vector of initial, final, and trajectory constraints

$$\begin{aligned} &lb_0 \leq \psi_0(x(t_0), u(t_0)) \leq ub_0, \\ &lb_f \leq \psi_f(x(t_f), u(t_f)) \leq ub_f, \\ &lb_t \leq S(x, u) \leq ub_t, \end{aligned} \quad (2)$$

respectively. For conciseness, we will refer to this optimal control problem as

$$\begin{cases} \min_{(x,u)} J(x,u) \\ \text{subject to} \\ \dot{x} = f(x) + g(x)u, \\ lb \leq c(x,u) \leq ub. \end{cases} \tag{3}$$

2.2 Different approaches

2.2.1 Classical collocation

One numerical approach to solve this optimal control problem is the direct collocation method outlined by Hargraves and Paris in [6]. The idea behind this approach is to transform the optimal control problem into a nonlinear programming problem. This is accomplished using a time mesh

$$t_0 = t_1 < t_2 < \ldots < t_N = t_f \tag{4}$$

and approximating the state x and the control input u as piecewise polynomials $\hat{x}$ and $\hat{u}$, respectively. Cubic polynomial may be chosen for the states and a linear polynomial for the control on each interval represents a good choice. Collocation is then used at the midpoint of each interval to satisfy Equation (1). Let $\hat{x}(x(t_1)^T, ..., x(t_N)^T)$ and $\hat{u}(u(t_1), ..., u(t_N))$ denote the approximations to x and u, respectively, depending on $(x(t_1)^T, ..., x(t_N)^T) \in \mathbb{R}^{nN}$ and $(u(t_1), ..., u(t_N)) \in \mathbb{R}^N$ corresponding to the value of x and u at the grid points. Then one solves the following finite dimension approximation of the original control problem (3)

$$\begin{cases} \min_{y \in \mathbb{R}^M} F(y) = J(\hat{x}(y), \hat{u}(y)) \\ \text{subject to} \\ \dot{\hat{x}} - f(\hat{x}(y), \hat{u}(y)) = 0, \quad lb \leq c(\hat{x}(y), \hat{u}(y)) \leq ub, \\ \qquad \forall t = \dfrac{t_j + t_{j+1}}{2} \quad j = 1, \ldots, N-1 \end{cases} \tag{5}$$

where $y = (x(t_1)^T, u(t_1), \ldots, x(t_N)^T, u(t_N))$, and $M = \dim y = (n+1)N$.

2.2.2 Inverse dynamic optimization

In [15] Seywald suggested an improvement to the previous method (see also [2] page 362 for an overview of this method). Following this work, one first solves a subset of system dynamics in (3) for the the control in terms of combinations of the state and its time derivative. Then one substitutes for the control in the remaining system dynamics and constraints. Next all the time derivatives $\dot{x}_i$ are approximated by the finite difference approximations

$$\dot{\bar{x}}(t_i) = \frac{x(t_{i+1}) - x(t_i)}{t_{i+1} - t_i}$$

to get

$$\left.\begin{matrix} p(\dot{\bar{x}}(t_i), x(t_i)) = 0 \\ q(\dot{\bar{x}}(t_i), x(t_i)) \leq 0 \end{matrix}\right\} \quad i = 0, ..., N-1.$$

The optimal control problem is turned into

$$\begin{cases} \min_{y \in \mathbb{R}^M} F(y) \\ \text{subject to} \\ p(\dot{\bar{x}}(t_i), x(t_i)) = 0 \\ q(\dot{\bar{x}}(t_i), x(t_i)) \leq 0 \end{cases} \tag{6}$$

where $y = (x(t_1)^T, \ldots, x(t_N)^T)$, and $M = \dim y = nN$. As with the Hargraves and Paris method, this parameterization of the optimal control problem (3) can be solved using nonlinear programming.

The dimensionality of this discretized problem is lower than the dimensionality of the Hargraves and Paris method, where both the states and the input are the unknowns. This induces substantial improvement in numerical implementation (see again [15] for an implementation of the Goddard problem).

2.2.3 Proposed Numerical Approach

In fact, it is usually possible to reduce the dimension of the problem further. Given an output, it is generally possible to parameterize the control and a part of the state in terms of this output and its time derivatives. In contrast to the previous approach, one must use more than one derivative of this output for this purpose.

When the whole state and the input can be parameterized with one output, one says that the system is flat [3]. When the parameterization is only partial, the dimension of the subspace spanned by the output and its derivatives is given by r the *relative degree* of this output.

Definition 1 ([7]) A single input single output system

$$\begin{cases} \dot{x} = f(x) + g(x)u \\ y = h(x) \end{cases} \tag{7}$$

is said to have *relative degree* r at point x_0 if $L_g L_f^k h(x) = 0$, in a neighborhood of x_0, and for all $k < r - 1$ $L_g L_f^{r-1} h(x_0) \neq 0$ where $L_f h(x) = \sum_{i=1}^{n} \frac{\partial h}{\partial x_i} f_i(x)$ is the derivative of h along f.

Roughly speaking, r is the number of times one has to differentiate y before u appears.

Result 1 ([7]) *Suppose the system* (7) *has relative degree* r *at* x^0. *Then* $r \leq n$. *Set*

$$\begin{aligned} \phi_1(x) &= h(x) \\ \phi_2(x) &= L_f h(x) \\ &\vdots \\ \phi_r(x) &= L_f^{r-1} h(x). \end{aligned}$$

If r *is strictly less than* n, *it is always possible to find* $n - r$ *more functions* $\phi_{r+1}(x), ..., \phi_n(x)$ *such that the mapping*

$$\phi(x) = \begin{pmatrix} \phi_1(x) \\ \vdots \\ \phi_n(x) \end{pmatrix}$$

has a Jacobian matrix which is nonsingular at x^0 and therefore qualifies as a local coordinates transformation in a neighborhood of x^0. The value at x^0 of these additional functions can be fixed arbitrarily. Moreover, it is always possible to choose $\phi_{r+1}(x), ..., \phi_n(x)$ in such a way that $L_g\phi_i(x) = 0$, for all $r+1 \leq i \leq n$ and all x around x^0.

The implication of this result is that there exists a change of coordinates $x \mapsto z = (z_1, z_2, ..., z_n)$ such that the systems equations may be written as

$$\begin{cases} \dot{z}_1 = z_2 \\ \dot{z}_2 = z_3 \\ \quad \vdots \\ \dot{z}_{r-1} = z_r \\ \dot{z}_r = b(z) + a(z)u \\ \dot{z}_{r+1} = q_{r+1}(z) \\ \quad \vdots \\ \dot{z}_n = q_n(z) \end{cases}$$

where $a(z)$ is nonzero for all z in a neighborhood of $z^0 = \phi(x^0)$.

In these new coordinates, any optimal control problem can be solved by a partial collocation, i.e. collocating only $(z_1, z_{r+1}, ..., z_n)$ instead of a full collocation $(z_1, ..., z_r, z_{r+1}, ..., z_n, u)$. Inverting the change of coordinates, the state and the input $(x_1, ..., x_n, u)$ can be expressed in terms of $(z_1, ..., z_1^{(r)}, z_{r+1}, ..., z_n)$. This means that once translated into these new coordinates, the original control problem (3) will involve r successive derivatives of z_1.

It is not realistic to use finite difference approximations as soon as $r > 2$. In this context, it is convenient to represent $(z_1, z_{r+1}, ...z_n)$ as B-splines. B-splines are chosen as basis functions because of their ease of enforcing continuity across knot points and ease of computing their derivatives.

Both equation from the dynamics and the constraints will be enforced at the collocation points. In general, w collocation points are chosen uniformly over the time interval $[t_o, t_f]$, (though optimal knots placements or Gaussian points may also be considered and are numerically important). The problem can be stated as the following nonlinear programming form:

$$\begin{cases} \min_{y \in \mathbb{R}^M} F(y) \\ \text{subject to} \\ \dot{z}_{r+1}(y) - q_{r+1}(z)(y) = 0 \\ \quad \vdots \\ \dot{z}_n(y) - q_n(z)(y) = 0 \text{ for every } w \\ \quad lb \leq c(y) \leq ub \end{cases} \tag{8}$$

where y represents the unknown coefficients of the B-splines. These have to be found using nonlinear programming.

2.2.4 Comparisons

Our approach is a generalization of inverse dynamic optimization. Let us summarize the presented approaches One could write the optimal control problem with:

- "Full collocation" solving problem (5) by collocating $(x, u) = (x_1, ..., x_n, u)$ without any attempt of variable elimination. After collocation the dimension of the unknowns space is $\mathcal{O}(n + 1)$.

- "Inverse dynamic optimization" solving problem (6) by collocating $x = (x_1, ..., x_n)$. Here the input is eliminated from the equation using one derivative of the state. After collocation the dimension of the unknowns space is $\mathcal{O}(n)$.

- "Flatness parametrization" (Maximal inversion), our approach, solving problem (8) in the new coordinates collocating only $(z_1, z_{r+1}, ..., z_n)$. Here we eliminate as many variables as possible and replace them using the first r derivatives of z_1. After collocation, the dimension of the unknowns space is $\mathcal{O}(n - r + 1)$.

2.3 The ruled manifold criterion

When facing a new system dynamics, it would be interesting to know wether these can be fully inverted or not. The single-input case presented before is the exception. Unfortunately, up today, there does not exist any flatness criterion. Nevertheless the following necessary condition can be a handy tool to check wether one may completely invert a system. This necessary condition for a system to be flat is given by the following criterion [14] (see also [8]).

Result 2 ([14]) *Assume the system $\dot{x} = f(x, u)$ is flat. The projection on the p-space of the submanifold $p = f(x, u)$, where x is considered as a parameter, is a ruled manifold for all x.*

Eliminating u from the dynamics $\dot{x} = f(x, u)$ yields a set of equations $F(x, \dot{x}) = 0$ that defines a ruled manifold. In other words for all $(x, p) \in \mathbb{R}^{2n}$ such that $F(x, p) = 0$, there exists a direction $d \in \mathbb{R}^n$, $d \neq 0$ such that

$$\forall \lambda \in \mathbb{R}, F(x, p + \lambda d) = 0.$$

3 The reentry problem

In this section we present the reentry problem. We detail the nonlinear dynamics, the constraints and the cost function. We show that this system is not flat and explain how to parameterize its trajectories using a reduced number of variables and additional constraints. Finally we give a rewriting of the optimal control problem in terms of this reduced number of unknowns.

3.1 Dynamics

As detailed in Betts [1], the motion of the space shuttle are defined by the following set of equations

$$\dot{h} = v \sin\gamma \tag{9}$$
$$\dot{\phi} = \frac{v}{r}\cos\gamma\sin\psi/\cos\theta \tag{10}$$
$$\dot{\theta} = \frac{v}{r}\cos\gamma\cos\psi \tag{11}$$
$$\dot{v} = -\frac{D(\alpha)}{m} - g\sin\gamma \tag{12}$$
$$\dot{\gamma} = \frac{L(\alpha)}{mv}\cos\beta + \cos\gamma\left(\frac{v}{r} - \frac{g}{v}\right) \tag{13}$$
$$\dot{\psi} = \frac{1}{mv\cos\gamma}L(\alpha)\sin\beta + \frac{v}{r\cos\theta}\cos\gamma\sin\psi\sin\theta \tag{14}$$

where h denotes the altitude, ϕ the longitude, θ the latitude, v the velocity, γ the flight path, ψ the azimuth. The two control are α the angle of attack and β the bank angle.

3.2 Control objective and constraints

Here our problem is to maximize the final value of the θ variable in a *given time* t_f. The initial conditions are prescribed as

$$\begin{aligned} h(0) &= 260000 \text{ ft} \\ \phi(0) &= 0 \text{ deg} \\ \theta(0) &= 0 \text{ deg} \\ v(0) &= 25600 \text{ ft/sec} \\ \gamma(0) &= -1 \text{ deg} \\ \psi(0) &= 90 \text{ deg} \end{aligned}$$

In the numerical example treated in this report the final time t_f equals 2008.59 s. The study is restricted to the trajectory satisfying

$$\begin{aligned} 0 \leq h, -89 \text{ deg} \leq \theta \leq 89 \text{ deg} \\ 1 \leq v, -89 \text{ deg} \leq \gamma \leq 89 \text{ deg} \\ -90 \text{ deg} \leq \alpha \leq 90 \text{ deg}, -89 \text{ deg} \leq \beta \leq 89 \text{ deg} \end{aligned}$$

The final point of the trajectory is defined by the terminal area energy management (TAEM) interface which is defined by the following relations

$$h(t_f) = 80000 \text{ ft}, v(t_f) = 2500 \text{ ft/s}, \gamma(t_f) = -5 \text{ deg}$$

3.3 Physics constants and parameters

We use $\mu = 0.14076539\text{e}17$ as gravitational constant, $Re = 20902900$ ft as the radius of the Earth, $S = 2690$ ft^2 as the aerodynamic reference surface, $h_{ref} = 23800$ ft and ρ_0 =0.002378 for the following physics parameters

$$g = \mu/r^2 \tag{15}$$
$$\rho = \rho_0 \exp(-(r - Re)/h_{ref}) \tag{16}$$

We use $C_L = a_0 + a_1\alpha$ where α is in deg, a_0 =-0.20704, a_1 =0.029244. Lift is then given by

$$L = \frac{1}{2}C_L S\rho v^2 \tag{17}$$

Also we note $C_D = b_0 + b_1\alpha + b_2\alpha^2$, where b_0 =0.07854, b_1 = -0.61592e-2, b_2 = 0.621408e-3 and use it in

$$D = \frac{1}{2}C_D S\rho v^2 \tag{18}$$

The mass of the shuttle was chosen as

$$m = 6309.44 \text{ lbs}$$

3.4 The system is not flat

We use the ruled manifold criterion presented in section 2.3 to prove that the system is not flat.

Eliminating the control from the reentry dynamics yields an equation $F(x, \dot{x}) = 0$. To get this equation we have to solve for the unknowns α and β in terms of the states and its derivatives.

First one may pick equation (12) to get

$$D(\alpha) = -m\dot{v} - mg\sin(\gamma)$$

Then solve according to the physical model (18) to get

$$\alpha = \frac{-b_1 \pm \sqrt{b_1^2 - 4b_2(b_0 + \frac{2m(\dot{v}+g\sin\gamma)}{\rho S v^2})}}{2b_2} \tag{19}$$

On the other hand it straightforward to solve for β using equation (13), equation (14) and the fact that -89 deg $\leq \beta \leq 89$ deg. This gives

$$\beta = \arctan\left(\frac{\cos\gamma(\dot{\psi} - \frac{v}{r\cos\theta}\cos\gamma\sin\psi\sin\theta)}{\dot{\gamma} - \cos\gamma\left(\frac{v}{r} - \frac{g}{v}\right)}\right) \tag{20}$$

Using these last two relations in the reentry dynamics we get the manifold equation $F(x, p) = 0$, where $p = (p_1, p_2, p_3, p_4, p_5, p_6)^T = \dot{x}$ satisfy

$$p_1 = v\sin\gamma \tag{21}$$
$$p_2 = \frac{v}{r}\cos\gamma\sin\psi/\cos\theta \tag{22}$$
$$p_3 = \frac{v}{r}\cos\gamma\cos\psi \tag{23}$$

and Equation (24) Now let us look for a non-zero direction $d = (d_1, d_2, d_3, d_4, d_5, d_6)^T \in \mathbb{R}^6$ such that at a point (x, p) such that $F(x, p) = 0$, for all $\lambda \in \mathbb{R}$, $F(x, p + \lambda d) = 0$.

The first three equations (21), (22), (23) give

$$\begin{aligned} p_1 + \lambda d_1 &= v\sin\gamma \\ p_2 + \lambda d_2 &= \frac{v}{r}\cos\gamma\sin\psi/\cos\theta \\ p_3 + \lambda d_3 &= \frac{v}{r}\cos\gamma\cos\psi \end{aligned}$$

which give

$$d_1 = 0, d_2 = 0, d_3 = 0$$

Equation (24) gives after using the simplification $\sin(\arctan x) = \frac{x}{\sqrt{1+x^2}}$ Equation (25)

This equation must hold for all $\lambda \in \mathbb{R}$. After taking the square of the last expression, the square root in the last expression involving d_4 is the only one that still contains a

$$p_6 = \frac{\rho S v^2}{2mv\cos\gamma}\left(a_0 + a_1\frac{180}{\pi}\frac{-b_1 \pm \sqrt{b_1^2 - 4b_2(b_0 + \frac{2m(p_4 + g\sin\gamma)}{\rho S v^2})}}{2b_2}\right) \times \ldots$$
$$\sin\left(\arctan\left(\frac{\cos\gamma(p_6 - \frac{v}{r\cos\theta}\cos\gamma\sin\psi\sin\theta)}{p_5 - \cos\gamma\left(\frac{v}{r} - \frac{g}{v}\right)}\right)\right)$$
$$+ \frac{v}{r\cos\theta}\cos\gamma\sin\psi\sin\theta \qquad (24)$$

$$p_6 + \lambda d_6 = \frac{\rho S v^2}{2mv\cos\gamma}\left(a_0 + a_1\frac{180}{\pi}\frac{-b_1 \pm \sqrt{b_1^2 - 4b_2(b_0 + \frac{2m(p_4 + \lambda d_4 + g\sin\gamma)}{\rho S v^2})}}{2b_2}\right) \times \ldots$$
$$\frac{\cos\gamma(p_6 + \lambda d_6 - \frac{v}{r\cos\theta}\cos\gamma\sin\psi\sin\theta)}{\sqrt{\left(\cos\gamma(p_6 + \lambda d_6 - \frac{v}{r\cos\theta}\cos\gamma\sin\psi\sin\theta)\right)^2 + \left(p_5 + \lambda d_5 - \cos\gamma\left(\frac{v}{r} - \frac{g}{v}\right)\right)^2}}$$
$$+ \frac{v}{r\cos\theta}\cos\gamma\sin\psi\sin\theta \qquad (25)$$

square root terms in λ. It can not be matched to anything else in the expression. Thus, necessarily,

$$d_4 = 0$$

Taking the square of the last equation gives rise to the following second order polynomial in λ

$$\begin{aligned}
&\lambda^2(d_5^2 + d_6^2) \\
&+ 2\lambda\big(p_5 d_5 - d_5(\cos\gamma\left(\frac{v}{r} - \frac{g}{v}\right))\ldots \\
&+ \cos\gamma^2(p_6 d_6 - a_6(\frac{v}{r\cos\theta}\cos\gamma\sin\psi\sin\theta)\big)\ldots \\
&+ p_5^2 - 2p_5\cos\gamma\left(\frac{v}{r} - \frac{g}{v}\right) + (\cos\gamma\left(\frac{v}{r} - \frac{g}{v}\right))^2 \\
&+ \cos^2\gamma(p_6^2 - 2p_6\frac{v}{r\cos\theta}\cos\gamma\sin\psi\sin\theta\ldots \\
&+ \left(\frac{v}{r\cos\theta}\cos\gamma\sin\psi\sin\theta\right)) \\
&- c\cos^2\gamma
\end{aligned}$$

where

$$\begin{aligned}
c = &\frac{\rho S v^2}{2mv\cos\gamma} \\
&\big(a_0 + a_1\frac{180}{\pi} \\
&\frac{-b_1 \pm \sqrt{b_1^2 + 4b_2(b_0 + \frac{2m(p_4 + \lambda d_4 - g\sin\gamma)}{\rho S v^2})}}{2b_2}\big)
\end{aligned}$$

For this polynomial to be identically zero, necessarily we must have

$$d_5 = 0,\ d_6 = 0$$

Thus the candidate vector for a direction of the ruled manifold is $d = 0$. This shows the manifold is not ruled and so the system is not flat.

3.5 Parameterization

Should the system have been flat, we would have been using only 2 quantities (same number as inputs) for the parametrization of all its variables. As we will see in the following, we need 3 quantities instead. We now use

$$\begin{aligned}
z_1 &= r = h + Re \\
z_2 &= \theta \\
z_3 &= \phi
\end{aligned}$$

where Re is the radius of the Earth. Assuming that around the trajectory -90 deg $< \psi < 90$ deg, we recover from (10) and (11)

$$\psi = \arctan\left(\frac{\dot z_3}{\dot z_2}\cos z_2\right) \qquad (26)$$

Since -90 deg $< \gamma < 90$ deg, we get from (9) and (11)

$$\begin{aligned}
\gamma &= \arctan\left(\frac{\dot z_1}{\dot z_2}\frac{\cos\psi}{z_1}\right) \\
&= \arctan\left(\frac{\dot z_1}{z_1\sqrt{\dot z_2^2 + \dot z_3^2\cos^2 z_2}}\right) \qquad (27)
\end{aligned}$$

and then

$$\begin{aligned}
v &= \sqrt{\left(\frac{z_1\dot z_2}{\cos\psi}\right)^2 + \dot z_1^2} \\
&= \sqrt{\dot z_1^2 + z_1^2\left(\dot z_2^2 + \dot z_3^2\cos^2 z_2\right)} \qquad (28)
\end{aligned}$$

It is convenient in the sequel to solve for the derivatives $\dot v$, $\dot\gamma$, $\dot\psi$. These quantites can be obtained either by direct differentiation of (26) (27) and (28) as

$$\begin{aligned}\dot\psi(1+\tan^2\psi) &= \frac{d(\tan\psi)}{dt}\\ &= \frac{d}{dt}\left(\frac{\dot z_3}{\dot z_2}\cos z_2\right)\\ &= \frac{\ddot z_3}{\dot z_2}\cos z_2 - \dot z_3 \sin z_2 - \frac{\dot z_3 \ddot z_2}{\dot z_2^2}\cos z_2\end{aligned}$$

which gives

$$\begin{aligned}\dot\psi = &\left(1+\frac{\dot z_3^2}{\dot z_2^2}\cos^2 z_2\right)^{-1}\\ &\left(\frac{\ddot z_3}{\dot z_2}\cos z_2 - \dot z_3\sin z_2 - \frac{\dot z_3\ddot z_2}{\dot z_2^2}\cos z_2\right)\end{aligned} \tag{29}$$

and

$$\begin{aligned}\dot v =& \ddot z_1 \sin\gamma + \cos\gamma\cos\psi\,(\ddot z_2 z_1 + \dot z_2 \dot z_1)\\ &+\cos\gamma\sin\psi\times\\ &(\ddot z_3 z_1\cos z_2 + \dot z_3\dot z_1\cos z_2 - \dot z_2\dot z_3 z_1 \sin z_2)\end{aligned} \tag{30}$$

$$\begin{aligned}\dot\gamma =& \frac{1}{v}\ddot z_1\cos\gamma - \frac{1}{v}\sin\gamma\cos\psi\,(\ddot z_2 z_1 + \dot z_2\dot z_1)\\ &-\frac{1}{v}\sin\gamma\sin\psi\times\\ &(\ddot z_3 z_1\cos z_2 + \dot z_3\dot z_1\cos z_2 - \dot z_2\dot z_3 z_1\sin z_2)\end{aligned} \tag{31}$$

The lift is computed from equations (13) and (14) as

$$\begin{aligned}L =& mv\Big((\dot\psi - v/z_1\cos\gamma\sin\psi\tan z_2)\cos\gamma)^2\\ &+(\dot\gamma-(v^2/z_1-g)\cos\gamma/v)^2\Big)^{1/2}\\ &sign(\dot\gamma-(v^2/z1-g)\cos\gamma/v)\end{aligned}$$

which we note after substitution with equations (26), (27), (28), (30) and (29)

$$L = f_L(z_1,\dot z_1,\ddot z_1,z_2,\dot z_2,\ddot z_2,z_3,\dot z_3,\ddot z_3) \tag{32}$$

The bank angle can be recomputed from the previous expression and equation (13)

$$\beta = -\arccos((\dot\gamma-(v^2/z_1-g)\cos\gamma/v/m)v/L)$$

which we note after substitution with equations (27), (28) and (31)

$$\beta = f_\beta(z_1,\dot z_1,\ddot z_1,z_2,\dot z_2,z_3,\dot z_3) \tag{33}$$

Using the linear model for lift (see appendix), we can solve for the angle of attack

$$\alpha = (2L/\rho/v^2/S - a_0)/a_1$$

which we note after substitution with equations (28) and (32) and the air density model for $\rho(z_1)$ given by equation (16)

$$\alpha = f_\alpha(z_1,\dot z_1,\ddot z_1,z_2,\dot z_2,\ddot z_2,z_3,\dot z_3,\ddot z_3) \tag{34}$$

The drag is then recomputed from the law

$$D = \frac{1}{2}\rho S v^2 C_D$$

3.5.1 Parameterization constraints

The reentry dynamics have the same nonlinear structure as the following simple nonlinear system with 3 states and 2 inputs

$$\begin{aligned}\dot x_1 &= -D(u_1)\\ \dot x_2 &= L(u_1)\cos u_2\\ \dot x_3 &= L(u_1)\sin u_2\end{aligned}$$

In general this system is not flat (e.g. if D and L correspond to drag and lift models). In other words, not any time function $t\mapsto(x_1(t),x_2(t),x_3(t))$ is a trajectory of the system. But the trajectories of the system, i.e. time functions $t\mapsto(x_1(t),x_2(t),x_3(t),u_1(t),u_2(t))$ solution to the dynamics, indeed satisfy

$$\tan u_2 = \left(\frac{\dot x_3}{\dot x_2}\right) \tag{35}$$

and

$$L = \sqrt{\dot x_2^2+\dot x_3^2}sign(\dot x_2\cos u_2)$$

These are only necessary conditions. Sufficient extra conditions are that

$$\dot x_1 = -D(L^{-1}(\sqrt{\dot x_2^2+\dot x_3^2}sign(\dot x_2\cos u_2)))$$

In order to solve equation (35), one has to pick the right determination of the angle. In general it can not be assumed that $u_2\in]-\pi/2,\pi/2[$ (it is the case in our example though). Let us call u_2^* this solution (defined up to π). A suitable value has to be such that

$$\begin{aligned}\dot x_2 &= L(u_1)\cos u_2^*\\ \dot x_3 &= L(u_1)\sin u_2^*\end{aligned}$$

To summarize, the trajectories of the system are of the form

$$\begin{aligned}t\mapsto&(x_1(t),x_2(t),x_3(t),\\ &L^{-1}(\sqrt{\dot x_2^2+\dot x_3^2}sign(\dot x_2\cos u_2^*),u_2^*)\end{aligned}$$

where x_1, x_2, x_3, u_2^* are any arbitrary function that satisfy

$$\begin{aligned}\dot x_1 &= -D(L^{-1}(\sqrt{\dot x_2^2+\dot x_3^2}sign(\dot x_2\cos u_2^*)))\\ \dot x_2 &= \sqrt{\dot x_2^2+\dot x_3^2}sign(\dot x_2\cos u_2^*)\cos u_2^*\\ \dot x_3 &= \sqrt{\dot x_2^2+\dot x_3^2}sign(\dot x_2\cos u_2^*)\sin u_2^*\\ \tan u_2^* &= \left(\frac{\dot x_3}{\dot x_2}\right)\end{aligned}$$

Similarly, in our case the following constraints must hold

1. First the drag and the lift must correspond. In other words, the drag that is computed from the lift must be such that

$$m\dot v + g\sin\gamma + D = 0$$

2. Also the sign that appears in the lift expression has to be taken into account. Two additional constraints have to be satisfied to transform the previous necessary condition in a sufficient condition. It is assumed that $\alpha \in]-\pi/2, \pi/2[$. So u_2^* is uniquely defined by the $\arctan$ function. As a summary, the trajectories have to satisfy

$$
\begin{aligned}
&(\dot\psi - v/z_1 \cos\gamma \sin\psi \tan z_2)\cos\gamma) \\
&\quad = L\cos\beta/m/v/\cos\ \gamma \\
&(\dot\gamma - (v^2/z_1 - g)\cos\gamma/v) \\
&\quad = L\sin\beta/m/v
\end{aligned}
$$

3.5.2 Parameterization of the trajectories

The previous relations derived at section 3.5 are necessary conditions. In other words if the time functions

$$t \mapsto (h(t), \phi(t), \theta(t), V(t), \gamma(t), \psi(t), \alpha(t), \beta(t))$$

are solutions of the reentry dynamics then they are of the form

$$
\begin{aligned}
h &= z_1 - R_e \\
\phi &= z_3 \\
\theta &= z_2 \\
v &= \sqrt{\dot z_1^2 + z_1^2\,(\dot z_2^2 + \dot z_3^2 \cos^2 z_2)} \\
\gamma &= \arctan\left(\frac{\dot z_1}{z_1\sqrt{\dot z_2^2 + \dot z_3^2\cos^2 z_2}}\right) \\
\psi &= \arctan\left(\frac{\dot z_3}{\dot z_2}\cos z_2\right) \\
\alpha &= f_\alpha(z_1, \dot z_1, \ddot z_1, z_2, \dot z_2, \ddot z_2, z_3, \dot z_3, \ddot z_3) \\
\beta &= f_\beta(z_1, \dot z_1, \ddot z_1, z_2, \dot z_2, z_3, \dot z_3)
\end{aligned}
$$

Conversely any time function $t \mapsto (h(t), \phi(t), \theta(t), V(t), \gamma(t), \psi(t), \alpha(t), \beta(t))$ computed from the same relations are not solutions to the reentry dynamics. Sufficient extra conditions are that these functions must satisfy the extra conditions

$$
\begin{aligned}
&m\dot v + g\sin\gamma + \frac{1}{2}C_D\rho S\left(\left(\frac{z_1\dot z_2}{\cos z_3}\right)^2 + \dot z_1^2\right) = 0 \\
&(\dot\psi - v/z_1\cos\gamma\sin\psi\tan z_2)\cos\gamma) \\
&= L\cos\beta/m/v/\cos\ \gamma \\
&(\dot\gamma - (v^2/z_1 - g)\cos\gamma/v) \\
&= L\sin\beta/m/v
\end{aligned}
$$

These three relations can be rewritten, after substitution with the necessary conditions (26), (27), (28), (30), (32), (33)

$$F_1(z_1, \dot z_1, \ddot z_1, z_2, \dot z_2, \ddot z_2, z_3, \dot z_3, \ddot z_3) = 0 \tag{36}$$
$$F_2(z_1, \dot z_1, \ddot z_1, z_2, \dot z_2, \ddot z_2, z_3, \dot z_3, \ddot z_3) = 0 \tag{37}$$
$$F_3(z_1, \dot z_1, \ddot z_1, z_2, \dot z_2, \ddot z_2, z_3, \dot z_3, \ddot z_3) = 0 \tag{38}$$

3.6 Rewriting of the optimal control problem

The problem is only to find the best time functions $[0, t_f] \ni t \mapsto (z_1(t), z_2(t), z_3(t))$ so as to maximize $z_2(t_f)$ under the following constraints.

- Initial constraints

$$h(0) = z_1(0) - Re \tag{39}$$
$$\phi(0) = z_3(0) \tag{40}$$
$$\theta(0) = z_2(0) \tag{41}$$
$$v(0) = \sqrt{\dot z_1^2(0) + z_1^2(0)\,(\dot z_2^2(0) + \dot z_3^2(0)\cos^2 z_2(0))} \tag{42}$$
$$\gamma(0) = \arctan\left(\frac{\dot z_1(0)}{z_1(0)\sqrt{\dot z_2^2(0) + \dot z_3^2(0)\cos^2 z_2(0)}}\right) \tag{43}$$
$$\psi(0) = \arctan\left(\frac{\dot z_3(0)}{\dot z_2(0)}\cos z_2(0)\right) \tag{44}$$

- Trajectory constraints (must hold for all $t \in [0, t_f]$)

$$F_1(z_1, \dot z_1, \ddot z_1, z_2, \dot z_2, \ddot z_2, z_3, \dot z_3, \ddot z_3) = 0 \tag{45}$$
$$F_2(z_1, \dot z_1, \ddot z_1, z_2, \dot z_2, \ddot z_2, z_3, \dot z_3, \ddot z_3) = 0 \tag{46}$$
$$F_3(z_1, \dot z_1, \ddot z_1, z_2, \dot z_2, \ddot z_2, z_3, \dot z_3, \ddot z_3) = 0 \tag{47}$$

$$
\begin{aligned}
&0 \le z_1 - Re, -89 \le z_2 \le 89 \\
&1 \le \sqrt{\dot z_1^2 + z_1^2\,(\dot z_2^2 + \dot z_3^2\cos^2 z_2)}, \\
&-89 \le \arctan\left(\frac{\dot z_1}{z_1\sqrt{\dot z_2^2 + \dot z_3^2\cos^2 z_2}}\right) \le 89, \\
&-90 \le f_\alpha(z_1, \dot z_1, \ddot z_1, z_2, \dot z_2, \ddot z_2, z_3, \dot z_3, \ddot z_3) \le 90, \\
&-89 \le f_\beta(z_1, \dot z_1, \ddot z_1, z_2, \dot z_2, z_3, \dot z_3) \le 89
\end{aligned}
$$

- Endpoint constraints

$$h(t_f) = z_1(t_f) - Re \tag{48}$$
$$v(t_f) = \sqrt{\dot z_1^2(t_f) + z_1^2(t_f)\,(\dot z_2^2(t_f)\dot z_3^2(t_f)\cos^2 z_2(t_f))} \tag{49}$$
$$\gamma(t_f) = \arctan\left(\frac{\dot z_1(t_f)}{z_1(t_f)\sqrt{\dot z_2^2 + \dot z_3^2\cos^2 z_2}}\right) \tag{50}$$

4 Numerical results

In this section we give numerical results using the proposed methodology. Details about the initialisation and convergence are given. Accuracy of the method is discussed and comparisons with reference results are given.

$h(t_f)$ (ft)	102600
$v(t_f)$ (ft/sec)	3291.6
$\gamma(t_f)$ (deg)	-3.6479
$\theta(t_f)$ (deg)	31.0802

Figure 1: Initial guess terminal values and cost function value.

$h(t_f)$ (ft)	80182
$v(t_f)$ (ft/sec)	2475.3
$\gamma(t_f)$ (deg)	-5.0179
$\theta(t_f)$ (deg)	33.0656

Figure 2: Terminal values and cost function value after optimisation.

4.1 Numerical setup

4.1.1 Initial guess

The system was initialized with control variables set to α =21 deg for the angle of attack, and $\beta(t) = 75 \times (-1 + t/t_f)$ for the bank angle (in deg). After a careful integration performed with Matlab `ode23`, the corresponding trajectory was found to give the data given in Figure 1.

From these trajectories the unknown coefficients were computed through a least square B-spline approximation. Of course the results depend on the number of coefficients, the order of the B-splines and the multiplicity of their knots and the fitting mesh.

Then we recomputed the control histories from the B-splines representation of the outputs z_1, z_2, z_3 using the formulas given in Section 3.5.

Finally we reintegrated the system dynamics from the same initial condition as before while using the latest control histories. Results are given for a typical case with 40 intervals (44 coefficients) per variable, 60 points mesh, 4^{th} order B-Splines with multiplicity of 3.

$$h^{40\times 60}(t_f) - h_{guess}(t_f) = 55.244 \text{ ft} ,$$
$$v^{40\times 60}(t_f) - v_{guess}(t_f) = -0.7559 \text{ ft/sec} ,$$
$$\gamma^{40\times 60}(t_f) - \gamma_{guess}(t_f) = -0.0266 \text{ deg}$$

Results vary with the number of coefficients and Results are given for a typical case with 100 intervals (104 coefficients) per variable unknown variables, 200 points mesh, 4^{th} order B-Splines with multiplicity of 3.

$$h^{100\times 200}(t_f) - h_{guess}(t_f) = -11.1395 \text{ ft} ,$$
$$v^{100\times 200}(t_f) - v_{guess}(t_f) = 0.5795 \text{ ft/sec} ,$$
$$\gamma^{100\times 200}(t_f) - \gamma_{guess}(t_f) = -0.0216 \text{ deg}$$

In these two cases the mesh was refined around the two boundaries of the domain, to limit the side effects of least square approximation. In fact, a linearly spaced mesh would produce much larger errors. With the 100 intervals and the 200 points linearly spaced mesh the same test gives

$$h^{100\times 200l}(t_f) - h_{guess}(t_f) = 141 \text{ ft} ,$$
$$v^{100\times 200l}(t_f) - v_{guess}(t_f) = 7.49 \text{ ft/sec} ,$$
$$\gamma^{100\times 200l}(t_f) - \gamma_{guess}(t_f) = 0.024 \text{ deg}$$

We were investigating wether the B-Splines were able to provide us with a high degree of accuracy as required for our application. The above numerical investigation suggests that they are well suited provided a sufficiently large number of coefficient is chosen. Also the choice of the mesh matters. In the rest of the report we conduct the tests with a mesh refined around the two ends of the time interval.

4.1.2 Solving the optimal control problem

All the tests were conducted using Matlab 6.5 with the collocation routines from the Splines toolbox and the `fmincon` routine from the Optimisation toolbox.

No analytical gradients were provided, neither for the cost nor for the constraints. This has an impact on the computation times.

Scalings were used for the cost function and the constraints. This helped the nonlinear programming routine to find appropriate search lines. Also nonlinear equality constraints over the time interval (due to the parameterization) were relaxed to help convergence. Eventually the optimisation procedure was restarted once with the previous solution as an initial guess and more stringent values for the relaxation parameter.

We used 40 intervals (44 coefficients per variable) and a 65 nonlinearly spaced points mesh.

With a first run (relaxation parameter set to 1e-4) the obtained solution gave $h(t_f)$ = 79906 ft, $v(t_f)$ =2753.7 ft/sec, $\gamma(t_f)$ =-4.0726 deg, $\theta(t_f)$= 33.5771 deg. This first problem was solved using 104 iterations of `fmincon`, which used 14117 F-count and took approximatively 20 minutes on a Pentium III 1.13 GHz Windows XP based computer.

These results were eventually improved using a new relaxation parameter of 1e-5. Final results are given in Figure 2. The corresponding trajectory is detailed in Figure 3 and Figure 4. This run used 122 iterations of `fmincon`, which used 16703 F-count and took approximatively 50 minutes on the same computer.

5 Conclusions

The numerical results must be compared to the solution given in [1] that gives $\theta(t_f)$ =34.1412 deg, a higher value. The result presented here were obtained by a much different technique. It seems we converged to a different solution. Also it seems that the accuracy could be improved further using more coefficients for the B-splines representation and well adapted meshes. It should be noted that only a simple nonlinear solver was used in this study and that the use of more complex, yet less convenient for implementation, solvers such as `NPSOL` [5] with analytic gradients could help too.

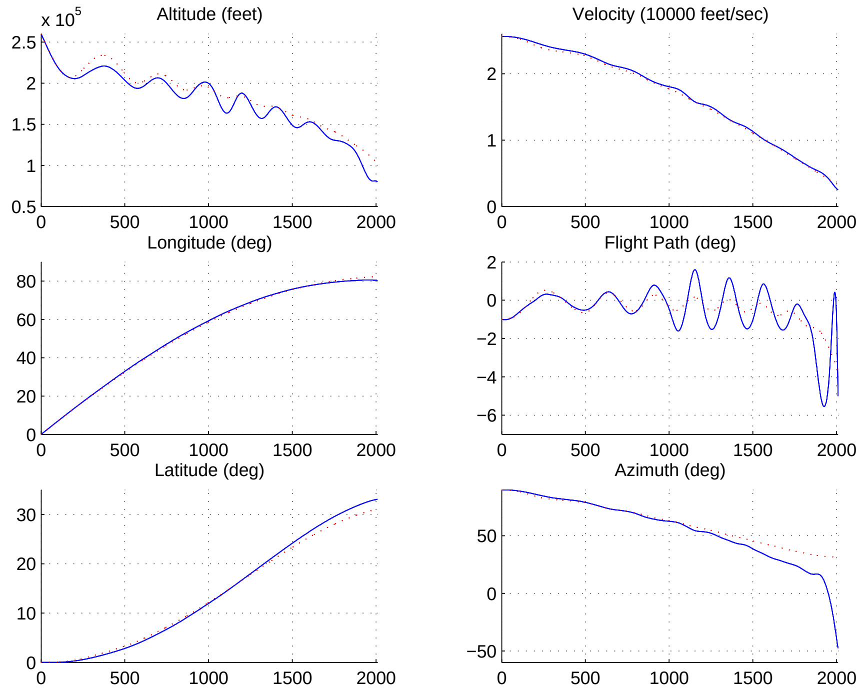

Figure 3: Reentry state variables. Optimal solution (plain) and initialisation (dotted).

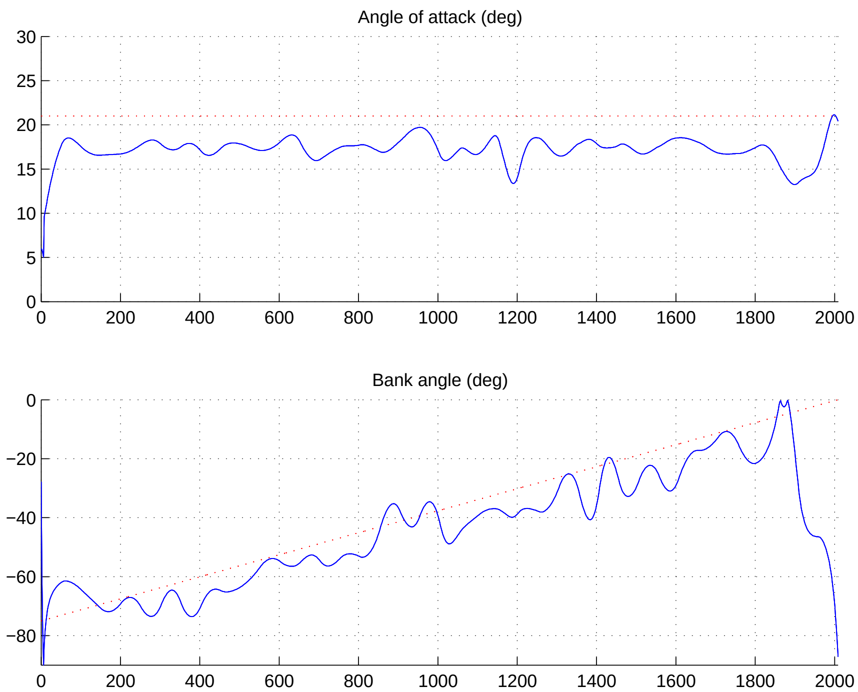

Figure 4: Reentry control variables. Optimal solution (plain) and initialisation (dotted).

Acknowledgement

This work was sponsored by CNES through a ARMINES contract.

References

[1] J. T. BETTS, *Practical Methods for optimal control using nonlinear programming*, SIAM, 2001.

[2] A. E. BRYSON, *Dynamic optimization*, Addison Wesley, 1999.

[3] M. FLIESS, J. LÉVINE, P. MARTIN, AND P. ROUCHON, *Flatness and defect of nonlinear systems: introductory theory and examples*, Int. J. Control, 61 (1995), pp. 1327–1361.

[4] ——, *A Lie-Bäcklund approach to equivalence and flatness of nonlinear systems*, IEEE Trans. Automat. Control, 44 (1999), pp. 922–937.

[5] P. GILL, W. MURRAY, M. SAUNDERS, AND M. WRIGHT, *User's Guide for NPSOL 5.0: A Fortran Package for Nonlinear Programming*, Systems Optimization Laboratory, Stanford University, Stanford, CA 94305, 1998.

[6] C. HARGRAVES AND S. PARIS, *Direct trajectory optimization using nonlinear programming and collocation*, AIAA J. Guidance and Control, 10 (1987), pp. 338–342.

[7] A. ISIDORI, *Nonlinear Control Systems*, Springer, New York, 2nd ed., 1989.

[8] P. MARTIN, R. M. MURRAY, AND P. ROUCHON, *Flat systems*, in Proc. of the 4th European Control Conf., Brussels, 1997, pp. 211–264. Plenary lectures and Mini-courses.

[9] M. B. MILAM, K. MUSHAMBI, AND R. M. MURRAY, *A new computational approach to real-time trajectory generation for constrained mechanical systems*, in IEEE Conference on Decision and Control, 2000.

[10] M. B. MILAM AND N. PETIT, *Constrained tra jectory generation for a planar missile*, tech. report, California Institute of Technology, Control and Dynamical Systems, 2001.

[11] M. B. MILAM, N. PETIT, AND R. M. MURRAY, *Constrained trajectory generation for micro-satellite formation flying*, in AIAA Guidance, Navigation and Control Conference, 2001, pp. 328–333.

[12] R. M. MURRAY, J. HAUSER, A. JADBABAIE, M. B. MILAM, N. PETIT, W. B. DUNBAR, AND R. FRANZ, *Software-Enabled Control, Information technology for dynamical systems*, Wiley-Interscience, 2003, ch. Online control customization via optimization-based control, pp. 149–174.

[13] N. PETIT, M. B. MILAM, AND R. M. MURRAY, *Inversion based constrained trajectory optimization*, in 5th IFAC symposium on nonlinear control systems, 2001.

[14] P. ROUCHON, *Necessary condition and genericity of dynamic feedback linearization*, J. Math. Systems Estim. Control, 5 (1995), pp. 345–358.

[15] H. SEYWALD, *Trajectory optimization based on differential inclusion*, J. Guidance, Control and Dynamics, 17 (1994), pp. 480–487.