This is the preliminary version of the paper
AIAA J. of Guidance, Control, and Dynamics 21-6(1998), 996-1000

LARGE SCALE DIRECT OPTIMAL CONTROL APPLIED TO A RE-ENTRY PROBLEM

J. FRÉDÉRIC BONNANS* AND G. LAUNAY †

Abstract. We present the numerical solution of an atmospheric reentry problem for a space shuttle. We discretize the control and state with identical grid, and use a large-scale successive quadratic programming technique. With the help of sliding horizon and successive refinement of the discretization, we can solve on a workstation a problem with 1600 grid intervals, an unusually large figure for this kind of real world optimal control problem.

Key Words. Optimal control, Discretization, Path following, Differential equations, Newton's method, Successive quadratic programming.

1. Introduction. The technique called *direct optimal control* consists in discretizing an optimal control problem, and then solving the resulting nonlinear programming problem. It is often opposed to the techniques based on Pontryagin's principle, in which the control is expressed as a function of the state and costate, reducing the optimality system (in the simplest case) to a two points boundary value problem (TPBVP), which can be solved by a multiple shooting algorithm (Stoer and Burlirsch [18]).

The advantages of each method have been discussed thoroughly by many authors, among them Pesch [14] and Betts [1]. It is recognized that multiple shooting is most effective when the starting point (for the state and costate) is good. In terms of complexity, this algorithm is optimal in the sense that the computational effort is (in the case of an integration scheme of order 1) proportional to the number of points used when integrating the differential system. In addition, the integration can be done using a device for controlling the precision. The drawbacks are that the method may have difficulties in converging if the starting point is poor, which may occur often as it is not easy to give good initial values for the costate. In addition, any structural change in the constraints implies a modification of the system of equations to be solved.

The advantage of a priori discretizing an optimal control problem is that it is a general method, not so sensitive to an initial guess for the costate, and which allows to use the software already available for solving nonlinear programming problems. In the past, this kind of technique has often been combined with a low-dimension parameterization of the control (see e.g. Kraft [11]). In that case, the nonlinear programming problem has a small number of variables, and a large number of constraints: the distributed control and state constraints. There exist effective algorithms for dealing with this kind of structure, the so-called active set methods (e.g. Gill, Murray, Wright [9]). However, parameterizing the control destroys the local structure of the optimal control problems. It is difficult to evaluate how far is the solution of the parameterized problem from the solution of the original problem.

*INRIA-Rocquencourt, Domaine de Voluceau, B.P. 105, 78153 Rocquencourt, France
†Laboratoire de Topologie, Université de Bourgogne, B.P. 138, 21004 Dijon, France. This work was done when the second author was with INRIA.

Another possibility is to discretize the control using the same grid intervals as for the state. The aim of this paper is to explore such a possibility. The disadvantage we have to face is the difficulty of solving the resulting large scale nonlinear programming problem. In particular, it seems difficult to obtain the same computational complexity as for multiple shooting. Rather, we may hope to obtain a less precise estimate of the optimal control, but it will be easier to obtain due to the generality of the method. Some results along this line were obtained by Betts and Huffman [2], [3].

In this paper we study the application of a large scale direct optimal control algorithm to the problem of atmospheric reentry of a space shuttle. In section 2, we explain how our optimal control problem is discretized, and how the nonlinear programming problem is solved. In particular, we give a path algorithm that takes into account a poor initial guess for the optimal control, and a method of refinement of the discretization that allows us to compute a more precise solution. In section 3, we describe the reentry problem, which is a highly nonlinear and state-constrained problem. Then in section 4, we give the numerical results. These results tend to show that the resolution of problems by a direct method and with an accurate discretization, is possible at least in some realistic optimal control problems.

2. The discrete optimal control problem. We consider the following family of optimal control problems (see e.g. Bryson and Ho [7]):

$$
\begin{aligned}
&\text{minimize } V(y(T),u);\\
&\frac{dy}{dt} = F(y(t),u,v(t),t) \quad t \in [0,T],\ y(0)=y_0,\\
&\underset{\sim}{c}(t) \le c(y(t),u,v(t)) \le \tilde{c}(t) \quad t \in [0,T],\\
&\underset{\sim}{c}_f \le c_f(y(T),u) \le \tilde{c}_f,\\
&\underset{\sim}{y}(t) \le y(t) \le \tilde{y}(t),\\
&\underset{\sim}{u} \le u \le \tilde{u},\\
&\underset{\sim}{v}(t) \le v(t) \le \tilde{v}(t),
\end{aligned}
$$

in which:

$T > 0$ is the free final time,
$y(t) \in \mathbb{R}^{n_y}$ is the value of the state at time t,
$v(t) \in \mathbb{R}^{n_v}$ is the control,
$u \in \mathbb{R}^{n_c}$ is a set of parameters not depending on time,
y_0 is a given value of the state at time 0,
V is the value function,
F is the dynamics of the problem,
c are the distributed constraints,
c_f are the final constraints.

We discretize the time interval as

$$0 = t_0 < t_1 < \cdots < t_{n_t} = T.$$

We discretize the control variables by taking functions that are of constant value v^k on each time step $[t_{k-1}, t_k[$, $1 \le k \le n_t$. Then we discretize the differential equation using an explicit one-step method (the classical fourth order Runge-Kutta scheme in our implementation). The discrete problem can be formulated as

$$
\begin{aligned}
&\min V(y^{n_t}, u);\\
&y^k = \Phi(y^{k-1}, u, v^k) \quad k = 1, \ldots, n_t,\ y^0 = y_0,\\
&\underset{\sim}{c}^k \leq c(y^k, u, v^k) \leq \tilde{c}^k,\\
&\underset{\sim}{c}_f \leq c_f(y^{n_t}, u) \leq \tilde{c}^f,\\
&\underset{\sim}{y}^k \leq y^k \leq \tilde{y}^k,\\
&\underset{\sim}{u} \leq u \leq \tilde{u},\\
&\underset{\sim}{v}^k \leq v^k \leq \tilde{v}^k,
\end{aligned}
$$

where of course the discrete bounds are discretization of the continuous bounds.

The discrete problem is a nonlinear programming problem of the following form:

$$
(NLP) \qquad \min_{x \in \mathbb{R}^n} f(x)\ ;\ \underset{\sim}{x} \leq x \leq \tilde{x}\ ;\ \underset{\sim}{g} \leq g(x) \leq \tilde{g},
$$

with $f : \mathbb{R}^n \to \mathbb{R}$ and $g : \mathbb{R}^n \to \mathbb{R}^p$; indeed, x is here the vector composed by the discrete state and control variables as well as the parameters. More precisely x is the concatenation of the vectors $(y^k)_{0 \leq k \leq n_t}, u, (v^k)_{1 \leq k \leq n_t}$ whereas $g(x)$ include the state equations as well as the distributed and final constraints of the discrete problem.

Problem (NLP) is, if the time discretization is fine, a large scale problem. An old idea for solving such problems is successive linear programming (SLP) which consists in, given a current point x^k, computing d^k solution of

$$
(LP_k) \qquad \min_{d \in \mathbb{R}^n} f'(x^k)d\ ;\ \underset{\sim}{x} \leq x^k + d^k \leq \tilde{x}\ ;\ \underset{\sim}{g} \leq g(x^k) + g'(x^k)d^k \leq \tilde{g}.
$$

The new point x^{k+1} may be $x^k + d^k$, or a point in the segment $[x^k, x^k + d^k]$ if a line search is used.

Another class of methods is sequential quadratic programming (SQP) in which, at each iteration, a direction d^k is computed as a local solution of

$$
(QP_k) \qquad \min_{d \in \mathbb{R}^n} f'(x^k)d + \frac{1}{2} d^t H^k d\ ;\ \underset{\sim}{x} \leq x^k + d^k \leq \tilde{x}\ ;\ \underset{\sim}{g} \leq g(x^k) + g'(x^k)d^k \leq \tilde{g}.
$$

Here H^k is an $n \times n$ matrix which is an approximation of the Hessian with respect to x of the Lagrangian associated with (NLP), i.e.

$$
\mathcal{L}(x, \lambda, \mu) := f(x) + \lambda^t g(x) + \mu^t x
$$

whose Hessian can be written as $\nabla_x^2 \mathcal{L}(x, \lambda, \mu) = H(x, \lambda)$ with

$$
H(x, \lambda) = \nabla^2 f(x) + \sum_{i=1}^{p} \lambda_i \nabla^2 g_i(x).
$$

For the study of local convergence of successive quadratic programming, we refer to [5, 6]. An early reference for successive quadratic programming applied to optimal control is Mitter [12].

We have compared (SQP) and (SLP) without line searches. For (SLP) the solution of the linear program to be solved at each iteration is computed using a primal simplex algorithm. For (SQP) we use a conjugate reduced-gradient algorithm. Details of the implementation of these optimization algorithms may be found in [4]. For other implementation of large scale nonlinear programming algorithms we cite Murtagh and Saunders [13] and Shanno and Marsten [17].

For future reference we note that the first-order optimality system of (NLP) can be written in the following compact form

$$\begin{cases} \nabla f(x) + g'(x)^t \lambda + \mu = 0, \\ \mu \in \partial I_{[\underset{\sim}{x}, \bar{x}]}(x), \\ \lambda \in \partial I_{[\underset{\sim}{g}, \bar{g}]}(g(x)), \end{cases}$$

where I_K denotes the indicator of set K:

$$I_K(z) := \begin{cases} 0 & \text{if } \; z \in K, \\ +\infty & \text{if not}, \end{cases}$$

and ∂ is the subdifferential in the sense of convex analysis. The subdifferential of the indicatrix coincides with the set of outward normals in the sense of convex analysis (see Hiriart-Urruty and Lemaréchal [8]).

We consider the problem of atmospheric reentry of a space shuttle, the function to be minimized being the integral of thermal flux, subject to some state constraints.

3. The shuttle reentry problem. The dynamic equations are those of flight dynamics without thrust:

$$\begin{aligned}
dV/dt &= -g_r \sin\gamma - g_\lambda \cos\gamma \cos\chi - \tfrac{\rho}{2m} V^2 S C_x \\
&\quad + \Omega^2 r (\cos\lambda \sin\gamma - \sin\lambda \cos\gamma \cos\chi) \cos\lambda \\[2ex]
d\gamma/dt &= -g_r \cos\gamma / V + g_\lambda \sin\gamma \cos\chi / V + V(\cos\gamma)/r \\
&\quad + \tfrac{\rho}{2m} V S C_z \cos\mu + 2\Omega \sin\chi \cos\lambda \\
&\quad + \Omega^2 r \cos\lambda (\cos\lambda \cos\gamma - \sin\lambda \sin\gamma \cos\chi)/V \\[2ex]
d\chi/dt &= g_\lambda \sin\chi / V(\cos\gamma) + V(\cos\gamma \sin\chi \tan\lambda)/r \\
&\quad + \tfrac{\rho}{2m} V S C_z \sin\mu / cos\gamma \\
&\quad + \Omega^2 r \cos\lambda \sin\lambda \sin\chi / (V \cos\gamma) \\
&\quad + 2\Omega (\sin\lambda - \cos\lambda \cos\chi \tan\gamma) \\[2ex]
dr/dt &= V \sin\gamma \\
d\lambda/dt &= \tfrac{V}{r} \cos\gamma \cos\chi
\end{aligned}$$

with the state variables

V the modulus of velocity,
γ flight path angle,
χ azimuth,
r distance from center of earth to shuttle,
λ geocentric latitude.

The equations include the following fixed parameters :
S reference surface,
Ω velocity of rotation of earth,
m mass of the vehicle,
g gravitational acceleration, taken equal to 9.81 $\mathrm{m/sec}^2$.

They also include the intermediate functions
$g_r(r, \lambda)$ and $g_\lambda(r, \lambda)$ radial and tangent component of gravitational acceleration,
$\rho = \rho(r)$ atmospheric density,
$\mathrm{Mach}(V, \rho) = V/V(\rho)$ with $V(\rho)$ the velocity of sound,
$C_x(\alpha, \mathrm{Mach})$ and $C_z(\alpha, \mathrm{Mach})$ drag and lift coefficients.

Two variables appear a priori as controls:
α the angle of attack,
μ the bank angle.

However, their derivatives are subject to bounds, so that we include them as state variables and consider their derivatives as the actual control:

$$\frac{d\alpha}{dt} = \beta,$$
$$\frac{d\mu}{dt} = \eta.$$

There is an integral cost which is the total thermal flux modelled as

$$\int_0^T C_q \sqrt{\rho} V^3 dt.$$

Here $C_q > 0$ is a given constant. In order to comply with the general formulation, we write

$$\frac{dc}{dt} = C_q \sqrt{\rho} V^3,$$
$$c(0) = 0,$$

so that the cost can be written as $c(T)$. The variables

$$\begin{array}{ll} n_z := \rho S V^2 (C_x \sin\alpha + C_z \cos\alpha)/2mg & \text{normal acceleration,} \\ \varphi := C_q \sqrt{\rho} V^3 & \text{thermal flux,} \end{array}$$

are constrained as follows:

$$n_z \leq 2.5 \text{ and } \varphi \leq 4.10^5 \text{ J/sec.}$$

In addition there are some bound constraints on the state:

$$\gamma \leq 0 \; ; \; 0 \leq \alpha \leq 40^o \; ; \; 1^o \leq \mu \leq 90^o$$

and the control

$$-1^o/\text{sec} \leq \beta \leq 1^o/\text{sec} \; ; \; -6^o/\text{sec} \leq \eta \leq 6^o/\text{sec}.$$

The lower bound of 1^o for μ is artificial. Its purpose is to avoid the null value for the bank angle. The symmetry associated with a null bank angle might make difficult the convergence of the algorithm.

Also we take into account a final state constraint on the velocity

$$V(T) = V_T.$$

The final time T is free. Through a change of variable on time we transform the problem into a new one with final time equal to 1, and where T is a control parameter.

4. Numerical experiments. We have used a Newton method (sequential quadratic programming) for constrained problems [4]. For implementation, we have built a fourth order Runge-Kutta integrator, and we compute the exact gradients for the discretized system. We also compute the Hessian of the Lagrangian, using a first-order discretization formula, so that our numerical Hessian is not far from the exact one. We have linked this piece of Fortran 77 code to an INRIA software for solving nonlinear programming problems by a large-scale successive linear or quadratic programming, called SOS-OPSYC; SOS stands for Sparse Optimization Solver. The overall software is called DOC (Direct Optimal Control). It uses the sparse LU factorization of Reid [16].

An essential difficulty in this kind of study consists in finding a reasonable starting point for the optimal control. This may require a high expertise level and a lot of time, whereas the aim of optimization techniques is precisely to speed up the design of the trajectory. In order to deal with this difficulty we decided to optimize first over a small time interval, choosing a target (the final velocity) close to the initial value, and then to decrease the value of the final velocity; the solution computed for a given target is used as the initial point for the new problem with a lower target. In this case the length T of the time interval, being a result of the optimization process, increases automatically. Of course fixing these values of the final velocity needs itself some tuning. Computation made with 50 times intervals used the following values for the final velocity:

Iteration	1	2	3	4	5	6	7	8	9	10	11
Final velocity	6.6	6.3	6.0	5.5	5.0	4.5	4.0	3.5	3.0	2.5	2.0

Successive values of final velocity (km/sec)

A more accurate discretization is desirable, but it would lead to prohibitive computing times. We prefer to perform the above path-following method with a poor discretization, and then refine discretization. The difficulty is to be able to predict a reasonably good value of the set of active constraints for the refined problem. More than that –and here we have to describe a little more in detail the algorithms– reduced gradient methods use at each iteration of the algorithm a basis; this basis is a subset x_B of the components of the variable x, of cardinality $|B| = p$, where p is the dimension of the image space, i.e. $g(x) \in \mathbb{R}^p$. Writing $x = (x_B, x_N)$, where $N := \{1, \dots, n\} \backslash B$, B is chosen in such a way that $\partial g(x)/\partial x_B$ is invertible. This allows computation of displacements d of x such that the linearized constraints of (QP_k) are satisfied. The difficulty is to compute a reasonable basis for the refined problem. This prevents us from choosing an arbitrary refinement. In our experiments we always divided each step by half, so that each state or control variable splits into two variables. Now in our application, the numerical solution has the property that the only control parameter (not distributed on time), i.e. the final time T, is basic whereas there is exactly one active final constraint. It follows that if each variable of the refined problem inherits from the status of the one from which it was created, i.e. basic, non basic, binding, then we have exactly the right number of basic variables for the refined problem. Note (using notations of section 2) that for the considered NLP problem $n_y = 8$, $n_v = 2$, $n_c = 1$ with 3 state constraints distributed on time and 1 final constraint. So there are $n = 10n_t + 1$ (that is between 501 and 16001) variables and $p = 11n_t + 1$ (that is between 551 and 17601) constraints. Performing SQP, we have found 0 or 1 degree of freedom. This means that the solution is essentially described by constraints.

In Table 1 below we give the computing time for each refinement of the grid (computations were made on an IBM R6000/350 work station). In order to have an idea of the effectiveness of refinement, we ran the sliding horizon with 100 time steps (instead of 50 as before). We compare in Table 2 the resulting computing times: the advantage of using the refinement technique is clear since computing with 100 time steps is more than three times longer than computing with 50 times steps and performing the doubling method previously described.

n_t			User-time		
50	$\rightarrow$	100	418 sec	=	6 min 58 sec
100	$\rightarrow$	200	923 sec	=	15 min 23 sec
200	$\rightarrow$	400	19,762 sec	=	5 h 29 min 22 sec
400	$\rightarrow$	800	12,078 sec	=	3 h 21 min 18 sec
800	$\rightarrow$	1600	21,655 sec	=	6 h 55 sec

TABLE 1
User time for doubling

Computation	User-time		
path with $n_t = 100$	4,302 sec	=	1 h 11 min 42 sec
path with $n_t = 50$	593 sec	=	9 min 53 sec
path with $n_t = 50$ plus one doubling	593 sec + 418 sec = 1011 sec	=	16 min 51 sec

TABLE 2
Effectiveness of doubling

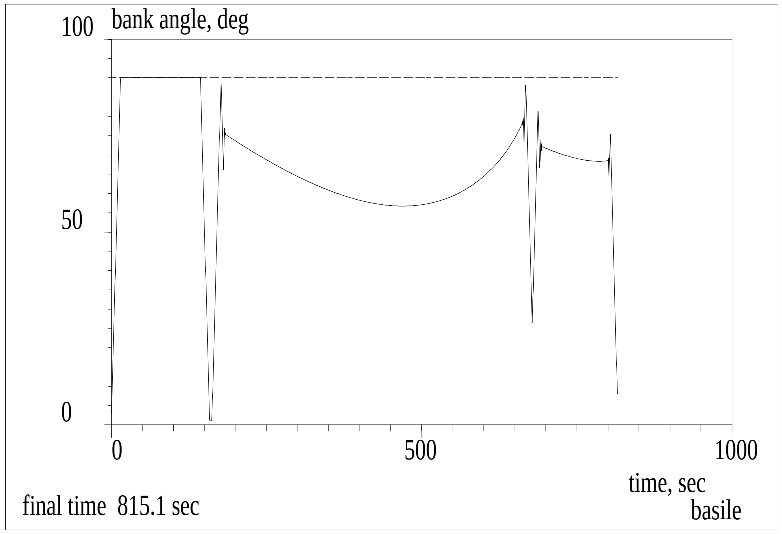
In Figures 1 to 4, we represent the bank angle and its derivative, the nonlinear state constraints, the velocity, flight, path angle and the altitude. These curves are closely related to the active constraints that we describe now.

We note that two state constraints are active: the thermal flux during the first stage of the trajectory, and then the normal acceleration. The optimal control can be described in the following way. The angle of attack remains at to its upper bound. The trajectory is divided in three phases. In the first the bank angle is equal to its upper bound. Then the constraint on the thermal flux becomes active. Finally the constraint on the normal acceleration is active. In addition there are four very short maneuvers, in order to reach the upper bound of the bank angle, and to switch successively to the thermal flux and normal acceleration constraints, and then to reach the final velocity.

The four short maneuvers are somewhat intriguing. One may doubt whether this is a side effect of the discretization or not. Although we have no definite answer to this question, let us mention the study by Lestienne and Tanguy [10] where numerical optimization was performed over the last 20 seconds of the trajectory, by fixing the value of the state at final time minus 20 seconds. These 20 seconds were discretized with 160 time steps. What was observed was a kind of damped oscillation behavior of control, that appears in fact if one looks carefully at the figures of this paper. No theoretical study of this problem has been made, up to our knowledge. Let us also mention the study of the final stage of descent, between 25 and 1 kilometers, by Poisson and Salas y Melia [15].

Conclusion We have performed a numerical computation of the solution of an optimal control problem, using the following tools: a sliding horizon technique for guessing a reasonable starting point, an automatic refinement of the time grid, and a large-scale nonlinear programming solver based successive quadratic programming. When refining the time discretization, it has been possible to guess a “good” initial basis, due to the special properties of the problem. The optimal control is essentially described by constraints, since it has at most one degree of freedom; however, successive quadratic programming has been more effective that our implementation of successive linear quadratic with the same line search. The optimal strategy has a simple physical interpretation, that confirms the intuition of aerospace engineers, i.e. the bank angle is first set to its maximum, then follows the constraint on the thermal flux, and then the one on normal acceleration. The rapid maneuvers between junction points seem not to be due to the discretization.

Acknowledgments. Thanks are due to C. Louis (Dassault Aviation) and CNES (Direction des Lanceurs, Evry) for their support and useful advices.



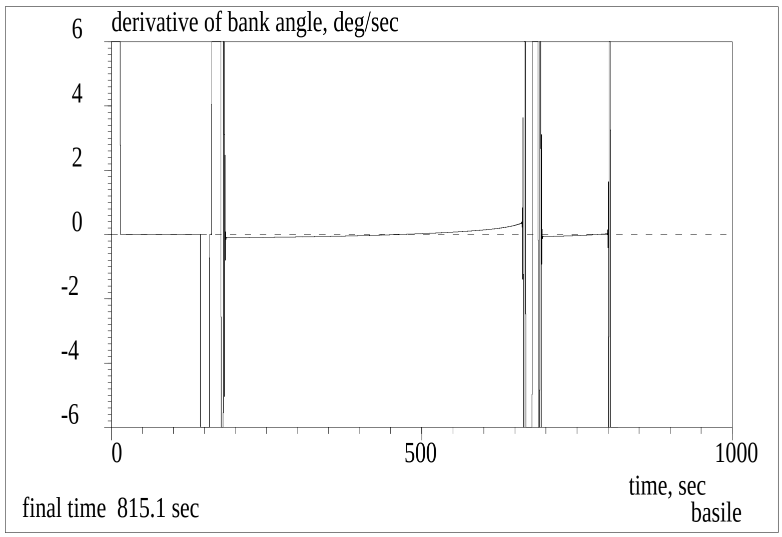
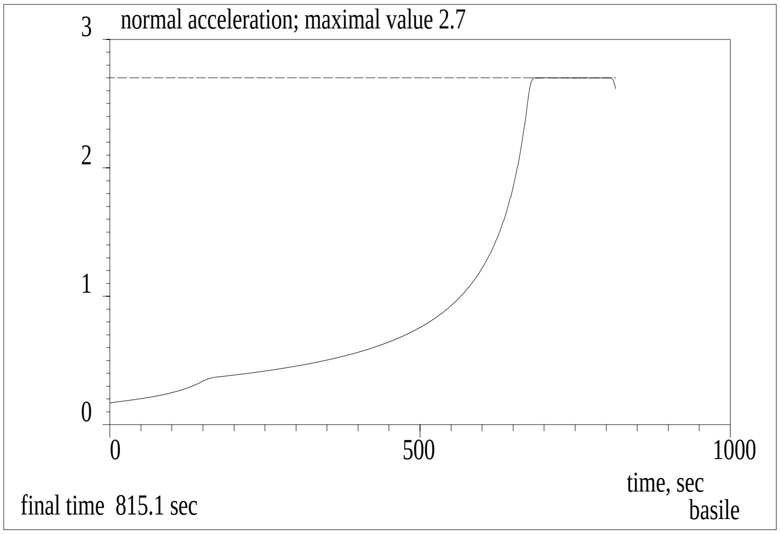


Figure 1: Bank angle and its derivative



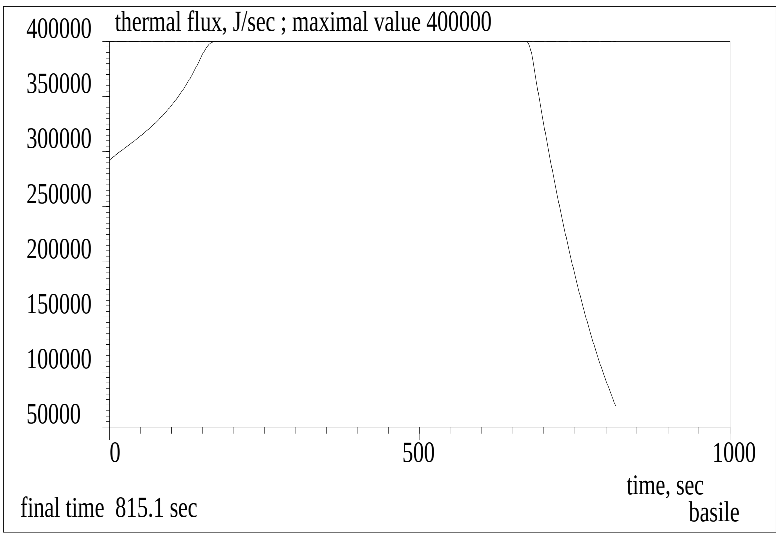
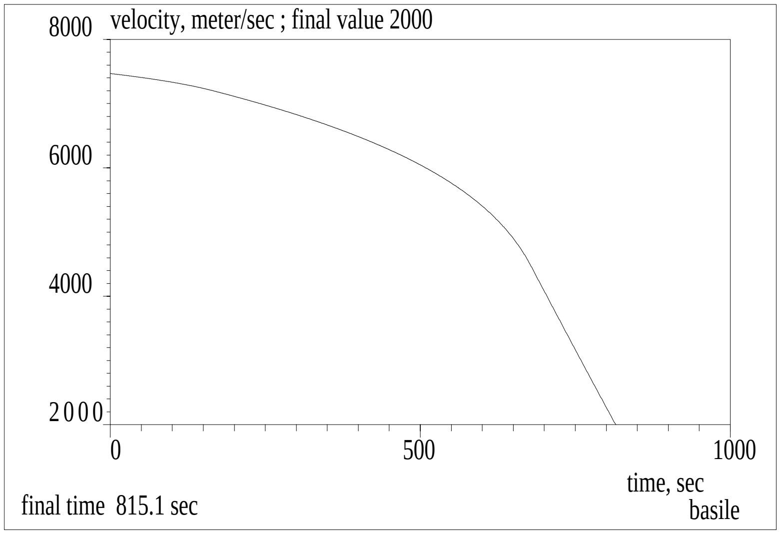


Figure 2: State constraints



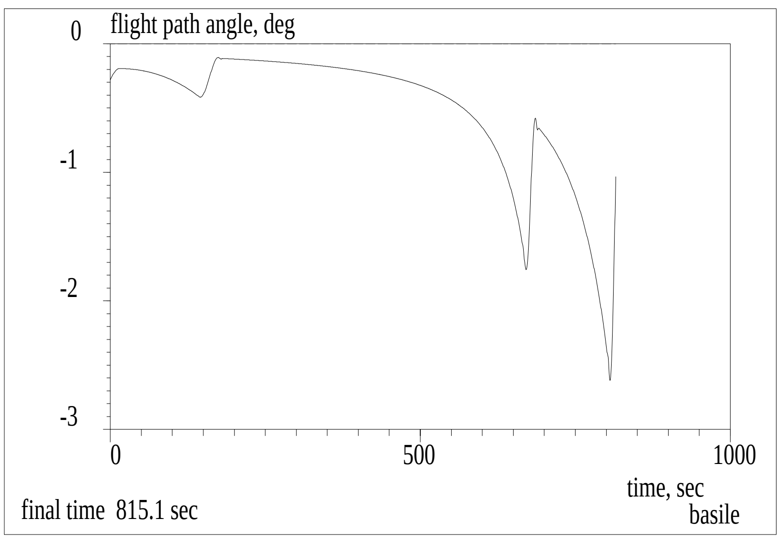


Figure 3: Velocity and flight path angle

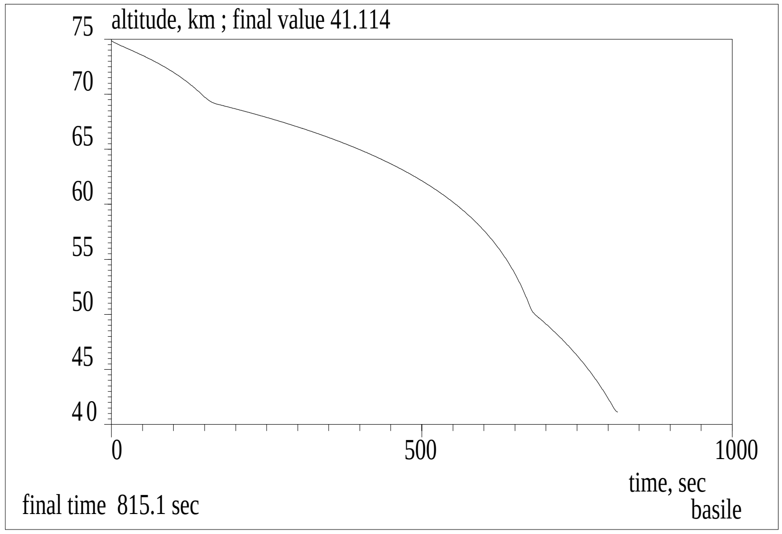


Figure 4: Altitude

REFERENCES

[1] Betts, J.T., "Survey of numerical methods to trajectory optimization". *J. Guidance Control Dynamics*, Vol. 21, 1998, pp. 193-207.

[2] Betts, J.T., and Huffman, W.P., "The application of sparse nonlinear programming to trajectory optimization". *J. Guidance Control Dynamics*, Vol. 15, 1992, pp. 198-206.

[3] Betts, J.T., and Huffman, W.P., "Path-constrained trajectory optimization using sparse sequential quadratic programming". *J. Guidance Control Dynamics*, Vol. 16, 1993, pp. 59-68.

[4] Blanchon G., Bonnans, J.F., and Dodu, J.C., "Application d'une méthode de programmation quadratique successive à l'optimisation des puissances dans les réseaux électriques de grande taille". *Bull. Dir. Etudes et Recherches*, ser. C. EDF, n^{o}2, 1991, pp. 67-101.

[5] Bonnans, J.F., "Local analysis of Newton type methods for variational inequalities and nonlinear programming", *Applied Math. Optimiz.*, Vol. 29, 1994, pp. 161-186.

[6] Bonnans, J.F., Gilbert, J.Ch., Lemaréchal, C., and Sagastizábal, C., "Optimisation Numérique: aspects théoriques et pratiques". Series Mathématiques et Applications, Vol 27, Springer-Verlag, Paris, 1997. (1997).

[7] Bryson, A.F., and Ho, Y.C., "Applied optimal control". Hemisphere, Washington D.C., 1987.

[8] Hiriart-Urruty, J.B., and Lemaréchal, C., "Convex analysis and minimization algorithms". Springer Verlag, Berlin, 1993.

[9] Gill, Murray, and Wright, "Practical optimization", Academic Press, New York, 1982.

[10] Lestienne, T., and Samuel, S., "Optimisation locale de la trajectoire de descente d'un planeur spatial". Mémoire de majeure, Ecole Polytechnique, Palaiseau, France, 1995.

[11] Kraft, D., "Finite-difference gradients versus error-quadrature gradients in the solution of parameterized optimal control problems". *Optimal Control, Appl. and Methods*, Vol. 2, 1981, pp. 191-199.

[12] Mitter, S.K., "Successive approximation methods for the solution of optimal control problems". *Automatica*, Vol. 3, 1966, pp. 135-149.

[13] Murtagh, B.A., and Saunders, M.A., "A projected lagrangian algorithm and its implementation for sparse nonlinear programming bases". *Math. Programming Study*, Vol. 16, 1982, pp. 84-117.

[14] Pesch, H.J., "Off-line and on-line computation of optimal trajectories in the aerospace field". *Proc. 12th Course in Applied Mathematics in the aerospace field*, Erice, 1991.

[15] Poisson, J.H., and Salas y Melia, D., "Optimisation de la phase finale de descente d'un planeur spatial". Mémoire de majeure, Ecole Polytechnique, Palaiseau, France, 1994.
[16] Reid, J.K., "A sparsity exploiting variant of the Bartels-Golub decomposition for linear programming bases". *Math. Programming*, Vol. 24, 1982, pp. 55-69.
[17] Shanno, D.F., and Marsten, R.E., "Conjugate gradient methods for linearly constrained nonlinear programming". *Math. Programming Study*, Vol. 16, 1982, pp. 149-161.
[18] Stoer, J., and Burlirsch, R., "Introduction to numerical analysis". Springer Verlag, New York, 1980.