JOURNAL OF GUIDANCE, CONTROL, AND DYNAMICS
Vol. 21, No. 6, November–December 1998

Large-Scale Direct Optimal Control Applied to a Re-Entry Problem

J. Frédéric Bonnans* and G. Launay†
Institut National de Recherche en Informatique et Automatique–Rocquencourt, 78153 Rocquencourt, France

We present the numerical solution of an atmospheric re-entry problem for the 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.

I. Introduction

THE technique called direct optimal control (DOC) consists of 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-point boundary-value problem, which can be solved by a multiple shooting algorithm.[1]

The advantages of each method have been discussed thoroughly by many authors, among them Pesch[2] and Betts.[3] 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, which allows one 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.[4] 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. Effective algorithms exist for dealing with this kind of structure, the so-called active set methods.[5] However, parameterizing the control destroys the local structure of the optimal control problems. It is difficult to evaluate how far the solution of the parameterized problem is from the solution of the original problem.

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.[6,7]

In this paper we study the application of a large-scale DOC algorithm to the problem of atmospheric re-entry of the Space Shuttle. In Sec. II, 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 Sec. III, we describe the re-entry problem, which is a highly nonlinear and state-constrained problem. Then, in Sec. IV, 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.

II. Discrete Optimal Control Problem

We consider the following family of optimal control problems[8]:

$$\text{minimize}\ V[y(T), u]$$

$$\frac{\mathrm{d}y}{\mathrm{d}t} = F[y(t), u, v(t), t], \qquad t \in [0, T], \qquad y(0) = y_0$$

$$\underset{\sim}{c}(t) \le c[y(t), u, v(t)] \le \tilde{c}(t), \qquad t \in [0, T]$$

$$\underset{\sim}{c}_f \le c_f[y(T), u] \le \tilde{c}_f, \qquad \underset{\sim}{y}(t) \le y(t) \le \tilde{y}(t)$$

$$\underset{\sim}{u} \le u \le \tilde{u}, \qquad \underset{\sim}{v}(t) \le v(t) \le \tilde{v}(t)$$

in which

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

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

$$\min V(y^{n_t}, u)$$

$$y^k = \Phi(y^{k-1}, u, v^k), \qquad k = 1, \ldots, n_t, \qquad y^0 = y_0$$

$$\underset{\sim}{c}^k \le c(y^k, u, v^k) \le \tilde{c}^k, \qquad \underset{\sim}{c}_f \le c_f(y^{n_t}, u) \le \tilde{c}^f$$

$$\underset{\sim}{y}^k \le y^k \le \tilde{y}^k, \qquad \underset{\sim}{u} \le u \le \tilde{u}, \qquad \underset{\sim}{v}^k \le v^k \le \tilde{v}^k$$

Received Oct. 13, 1994; revision received June 26, 1998; accepted for publication July 1,1998.

*Directeur de Recherche, Projet PROMATH, Domaine de Voluceau, B.P. 105.
†Chercheur Extérieur, Projet Promath, Domaine de Voluceau, B.P. 105; currently Chercheur Extérieur, Laboratoire de Topologie, Université de Bourgogne, B.P. 138, 21004 Dijon, France.

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

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

NLP:

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

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

LP$_k$:

$$\min_{d\in \mathbb{R}^n} f'(x^k)d, \qquad \underset{\sim}{x} \le x^k + d^k \le \tilde{x}$$

$$\underset{\sim}{g} \le g(x^k) + g'(x^k)d^k \le \tilde{g}$$

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 the following:

QP$_k$:

$$\min_{d\in \mathbb{R}^n} f'(x^k)d + \tfrac{1}{2}d^t H^k d, \qquad \underset{\sim}{x} \le x^k + d^k \le \tilde{x}$$

$$\underset{\sim}{g} \le g(x^k) + g'(x^k)d^k \le \tilde{g}$$

Here H^k is an $n\times n$ matrix that 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 Refs. 9 and 10. An early reference for successive quadratic programming applied to optimal control is Ref. 11.

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 Ref. 12. For other implementations of large-scale nonlinear programming algorithms we cite Refs. 13 and 14.

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

$$\nabla f(x) + g'(x)^t\lambda + \mu = 0$$

$$\mu \in \partial I_{[\underset{\sim}{x},\tilde{x}]}(x), \qquad \lambda \in \partial I_{[\underset{\sim}{g},\tilde{g}]}[g(x)]$$

where I_K denotes the indicator of set K

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

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 Ref. 15).

We consider the problem of atmospheric re-entry of the Space Shuttle, the function to be minimized being the integral of thermal flux, subject to some state constraints.

III. Space Shuttle Re-Entry Problem

The dynamic equations are those of flight dynamics without thrust:

$$\frac{dV}{dt} = -g_r \sin\gamma - g_\lambda \cos\gamma\cos\chi - \frac{\rho}{2m}V^2 SC_x + \Omega^2 r(\cos\lambda\sin\gamma - \sin\lambda\cos\gamma\cos\chi)\cos\lambda$$

$$\frac{d\gamma}{dt} = -\frac{g_r\cos\gamma}{V} + \frac{g_\lambda \sin\gamma\cos\chi}{V} + \frac{V(\cos\gamma)}{r} + \frac{\rho}{2m}VSC_z\cos\mu + 2\Omega\sin\chi\cos\lambda + \frac{\Omega^2 r\cos\lambda(\cos\lambda\cos\gamma - \sin\lambda\sin\gamma\cos\chi)}{V}$$

$$\frac{d\chi}{dt} = \frac{g_\lambda\sin\chi}{V(\cos\gamma)} + \frac{V(\cos\gamma\sin\chi\tan\lambda)}{r} + \frac{\rho}{2m}\frac{VSC_z\sin\mu}{\cos\gamma} + \frac{\Omega^2 r\cos\lambda\sin\lambda\sin\chi}{V\cos\gamma} + 2\Omega(\sin\lambda - \cos\lambda\cos\chi\tan\gamma)$$

$$\frac{dr}{dt} = V\sin\gamma, \qquad \frac{d\lambda}{dt} = \frac{V}{r}\cos\gamma\cos\chi$$

with the state variables

r = distance from the center of Earth to the Space Shuttle
V = modulus of velocity
γ = flight-path angle
λ = geocentric latitude
χ = azimuth

The equations include the following fixed parameters:

g = gravitational acceleration, taken equal to 9.81 m/s^2
m = mass of the vehicle
S = reference surface
Ω = velocity of rotation of Earth

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

Two variables appear a priori as controls:

α = angle of attack
μ = 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:

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

There is an integral cost that is the total thermal flux modeled as

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

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

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

so that the cost can be written as $c(T)$. The variables normal acceleration and thermal flux, respectively,

$$n_z := \frac{\rho S V^2 (C_x \sin\alpha + C_z \cos\alpha)}{2mg}, \qquad \varphi := C_q \sqrt{\rho} V^3$$

are constrained as follows: $n_z \le 2.5$ and $\varphi \le 4 \cdot 10^5$ J/s. In addition there are some bound constraints on the state: $\gamma \le 0$, $0 \le \alpha \le 40$ deg, and 1 deg $\le \mu \le 90$ deg; and the control: -1 deg/s $\le \beta \le 1$ deg/s and -6 deg/s $\le \eta \le$ 6 deg/s.

The lower bound of 1 deg 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, where T is a control parameter.

IV. Numerical Experiments

We have used a Newton method (sequential quadratic programming) for constrained problems.[12] 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 Institut National de Recherche en Informatique et Automatique (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. It uses the sparse lower–upper 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 level of expertise and a large amount of time, whereas the aim of optimization techniques is precisely to speed up the design of the trajectory. 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 some tuning itself. A computation made with 50 times intervals used the values listed in Table 1 for the final velocity.

A more accurate discretization is desirable, but it would lead to prohibitive computing times. We prefer to perform the preceding path-following method with a poor discretization and then to 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 a basis at each iteration of the algorithm; 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 $N := \{1, \ldots, n\} \backslash B$, 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, nonbasic, binding, then we have exactly the right number of basic variables for the refined problem. Note (using notations of Sec. II) that for the considered NLP problem $n_y = 8$, $n_v = 2$, $n_c = 1$ with three state constraints distributed on time and one final constraint. Consequently, there are $n = 10n_t + 1$ (that is between 501 and 16,001) variables and $p = 11n_t + 1$ (that is between 551 and 17,601) constraints. Performing SQP, we have found 0 or 1 degree of freedom. This means that the solution is essentially described by constraints.

In Table 2 we give the computing time for each refinement of the grid (computations were made on an IBM R6000/350 workstation). 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 3 the resulting computing times: the advantage of using the refinement technique is clear because computing with 100 time steps is more than three times longer than computing with 50 times steps and performing the doubling method previously described.

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

Table 1 Successive values of final velocity, km/s

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

Table 2 User time for doubling

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

Table 3 Effectiveness of doubling

Computation	User time
Path with n_t = 100	4302 s = 1 h 11 min 42 s
Path with n_t = 50	593 s = 9 min 53 s
Path with n_t = 50 plus one doubling	593 s + 418 s = 1011 s = 16 min 51 s

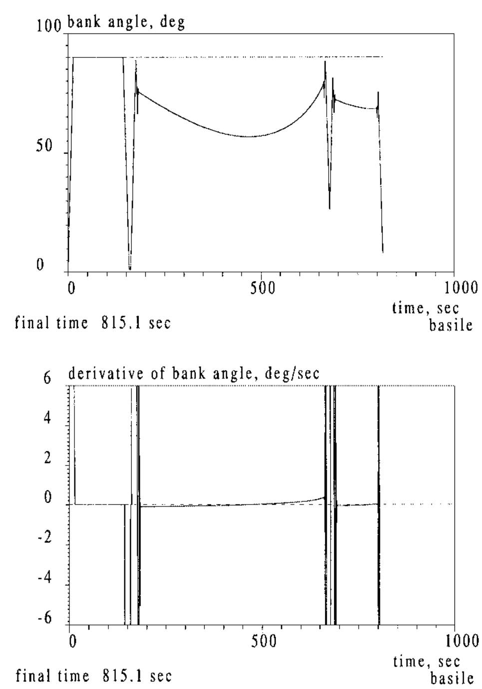

Fig. 1 Bank angle and its derivative.

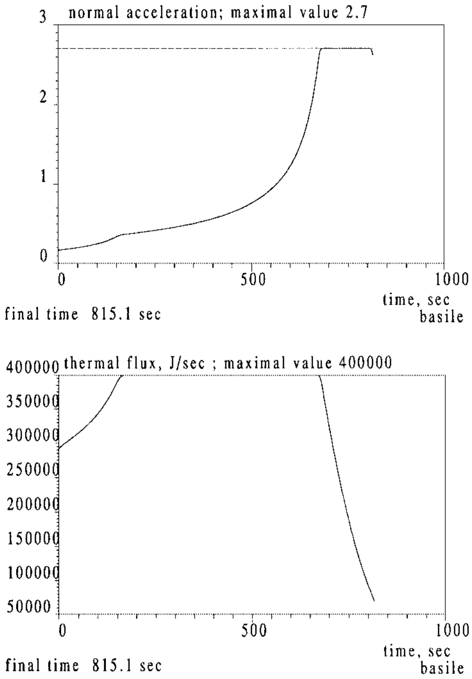

Fig. 2 State constraints.

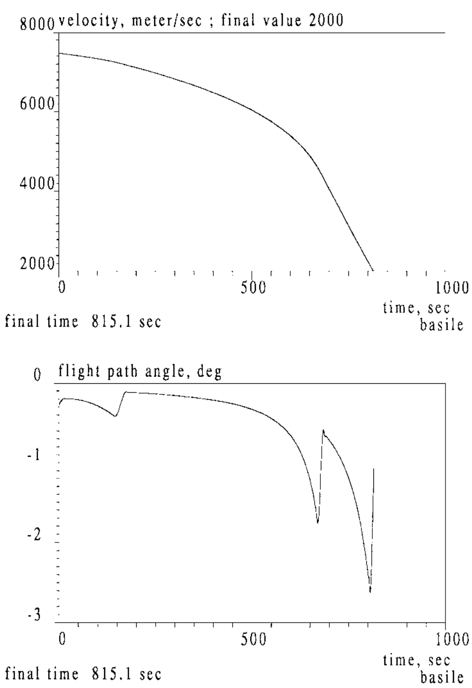

Fig. 3 Velocity and flight-path angle.

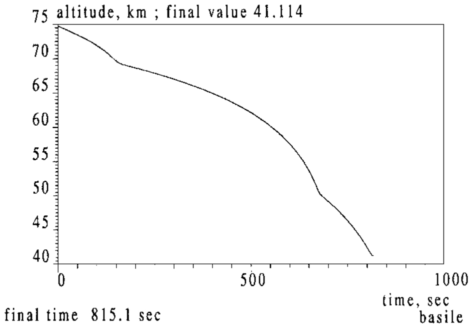

Fig. 4 Altitude.

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 its upper bound. The trajectory is divided in to 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 to reach the upper bound of the bank angle, 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 Samuel[17] where numerical optimization was performed over the last 20 s of the trajectory, by fixing the value of the state at final time minus 20 s. These 20 s were discretized with 160 time steps. What was observed was a kind of damped oscillation behavior of control, which appears, in fact, if one looks carefully at the figures of this paper. No theoretical study of this problem has been made, to our knowledge. Let us also mention the study of the final stage of descent, between 25 and 1 km, by Poisson and Salas y Melia.[18]

V. 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 on 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 because it has at most one degree of freedom; however, successive quadratic programming has been more effective than 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 and 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 of Dassault Aviation and to the Centre National d'Etudes Spatiales, Direction des Lanceurs, Evry, for their support and useful advice.

References

[1] Stoer, J., and Burlirsch, R., *Introduction to Numerical Analysis*, Springer–Verlag, New York, 1980, Chap. 7.

[2] Pesch, H. J., "Off-Line and On-Line Computation of Optimal Trajectories in the Aerospace Field," 12th Course in Applied Mathematics in the Aerospace Field, Erice, France, 1991.

[3] Betts, J. T., "Survey of Numerical Methods to Trajectory Optimization," *Journal of Guidance, Control, and Dynamics*, Vol. 21, 1998, pp. 193–207.

[4] Kraft, D., "Finite-Difference Gradients Versus Error-Quadrature Gradients in the Solution of Parameterized Optimal Control Problems," *Optimal Control, Applications and Methods*, Vol. 2, 1981, pp. 191–199.

[5] Gill, P. E., Murray, W., and Wright, M., *Practical Optimization*, Academic, New York, 1982.

[6] Betts, J. T., and Huffman, W. P., "The Application of Sparse Nonlinear Programming to Trajectory Optimization," *Journal of Guidance, Control, and Dynamics*, Vol. 15, 1992, pp. 198–206.

[7] Betts, J. T., and Huffman, W. P., "Path-Constrained Trajectory Optimization Using Sparse Sequential Quadratic Programming," *Journal of Guidance, Control, and Dynamics*, Vol. 16, 1993, pp. 59–68.

[8] Bryson, A. F., and Ho, Y. C., *Applied Optimal Control*, Hemisphere, Washington, DC, 1987.

[9] Bonnans, J. F., "Local Analysis of Newton Type Methods for Variational Inequalities and Nonlinear Programming," *Applied Mathematics and Optimization*, Vol. 29, 1994, pp. 161–186.

[10]Bonnans, J. F., Gilbert, J. Ch., Lemaréchal, C., and Sagastizábal, C., *Optimisation Numérique : Aspects Théoriques et Pratiques*, Vol. 27, Series Mathématiques et Applications, Springer–Verlag, Paris, 1997, Chap. 3.

[11]Mitter, S. K., "Successive Approximation Methods for the Solution of Optimal Control Problems," *Automatica*, Vol. 3, 1966, pp. 135–149.

[12]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," *Bulletin de la Direction Etudes et Recherches*, Ser. C. EDF, No. 2, 1991, pp. 67–101.

[13]Murtagh, B. A., and Saunders, M. A., "A Projected Lagrangian Algorithm and Its Implementation for Sparse Nonlinear Programming Bases," *Mathematical Programming Study*, Vol. 16, 1982, pp. 84–117.

[14]Shanno, D. F., and Marsten, R. E., "Conjugate Gradient Methods for Linearly Constrained Nonlinear Programming," *Mathematical Programming Study*, Vol. 16, 1982, pp. 149–161.

[15]Hiriart-Urruty, J. B., and Lemaréchal, C., *Convex Analysis and Minimization Algorithms*, Springer–Verlag, Berlin, 1993, Chap. 6.

[16]Reid, J. K., "A Sparsity Exploiting Variant of the Bartels-Golub Decomposition for Linear Programming Bases," *Mathematical Programming*, Vol. 24, 1982, pp. 55–69.

[17]Lestienne, T., and Samuel, S., "Optimisation Locale de la Trajectoire de Descente d'un Planeur Spatial," Mémoire de Majeure, Ecole Polytechnique, Palaiseau, France, March 1995.

[18]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, March 1994.