[10] Bianchini, R., and Brown, C., "Parallel Genetic Algorithms on Distributed-Memory Architectures," TR 436, Computer Science Dept., The Univ. of Rochester, New York, May 1993.

[11] Raymer, D. P., *Aircraft Design: A Conceptual Approach*, AIAA Education Series, AIAA, Washington, DC, 1989.

Optimal Trajectory for a Minimum Fuel Turn

Ulf Ringertz*
Kungliga Tekniska Högskolan,
SE-100 44 Stockholm, Sweden

Nomenclature

b	=	fuel burn
$\boldsymbol{c}_{\max}$	=	vector of upper bounds
$\boldsymbol{c}_{\min}$	=	vector of lower bounds
D	=	drag
g	=	gravity acceleration
$\boldsymbol{g}_{\max}$	=	vector of upper bounds on algebraic constraints
$\boldsymbol{g}_{\min}$	=	vector of lower bounds on algebraic constraints
h	=	altitude
L	=	lift
m	=	aircraft total mass
m_f	=	fuel mass
T	=	engine thrust
$\boldsymbol{u}$	=	assembled control vector
$\boldsymbol{x}$	=	assembled state vector
x_E	=	distance traveled east
$\boldsymbol{y}$	=	vector of discretized state and control variables
y_E	=	distance traveled north
α	=	angle of attack
γ	=	flight path angle
δ_T	=	power level angle
ϵ	=	thrust angle with respect to body fixed axis
ϕ	=	bank angle
ψ	=	heading

Introduction

FINDING optimal aircraft trajectories by using numerical optimization methods is a well-established area of research. Current methods in widespread use are based on discretization by using Hermite-Simpson collocation and direct solution using nonlinear programming methods. This type of method was pioneered by Hargraves and Paris[1] and is now perhaps the most widely used approach. In some cases, the numerically computed trajectories have also been verified by flight testing.[2]

In most cases numerical methods are demonstrated on problems in two dimensions, namely the vertical plane involving only longitudinal motion. However, many problems are truly three-dimensional and require a more general approach. Performance data models for aircraft are usually rather simple, in particular, the aerodynamic model. There is usually no aerodynamic data for sideslip or unsteady effects. If some restrictions are imposed on the three-dimensional motion of the aircraft, it is possible to solve problems in three dimensions by using only the standard performance data for the aircraft of interest.

The purpose of this Note is to present a point-mass model suitable for solving multistage trajectory optimization problems in three dimensions. The model is first discussed, and the numerical method is described. Finally, the presented method is used to solve an interesting test problem related to in-flight flutter testing.

Performance Model

The equations of motion for the aircraft are obtained by reducing the full six-degrees-of-freedom dynamic model of the aircraft described by Etkin,[3] assuming a point-mass model of the aircraft. Sideslip and unsteady aerodynamic effects are neglected because these effects have very little influence on the type of maneuvering considered here. Consequently, all maneuvering is assumed to take place without sideslip. The resulting equations of motion are given by the system of ordinary differential equations

$$m\dot{V} = T\cos(\alpha+\epsilon) - D - mg\sin\gamma \tag{1}$$

$$mV\dot{\gamma} = T\sin(\alpha+\epsilon)\cos\phi + L\cos\phi - mg\cos\gamma \tag{2}$$

$$mV\dot{\psi}\cos\gamma = T\sin(\alpha+\epsilon)\sin\phi + L\sin\phi \tag{3}$$

$$\dot{h} = V\sin\gamma \tag{4}$$

$$\dot{m}_f = -b \tag{5}$$

$$\dot{x}_E = V\cos\gamma\cos\psi \tag{6}$$

$$\dot{y}_E = V\cos\gamma\sin\psi \tag{7}$$

where a flat, nonrotating earth approximation is assumed.

Multistage Trajectory Optimization

The equations of motion defined by Eqs. (1–5) can be rewritten in brief form as

$$\dot{\boldsymbol{x}} = \boldsymbol{f}(\boldsymbol{x},\boldsymbol{u}) \tag{8}$$

The distances x_E and y_E are defined as algebraic constraints by integrating Eqs. (6–7), giving

$$x_E(t_F) = \int_{t=0}^{t_F} V(t)\cos\gamma(t)\cos\psi(t)\,\mathrm{d}t \tag{9}$$

and

$$y_E(t_F) = \int_{t=0}^{t_F} V(t)\cos\gamma(t)\sin\psi(t)\,\mathrm{d}t \tag{10}$$

Consequently, Eqs. (6) and (7) are not included in Eq. (8).

Additional requirements, such as restrictions on load factor n_z, dynamic pressure, lift coefficient, Mach number, and indicated airspeed V_i are implemented as purely algebraic constraints in the form

$$\boldsymbol{g}_{\min} \le \boldsymbol{g}(\boldsymbol{x},\boldsymbol{u}) \le \boldsymbol{g}_{\max} \tag{11}$$

The differential Eq. (8) and algebraic Eq. (11) are discretized by using Hermite-Simpson collocation[4] with state variables interpolated as piecewise third-order polynomials and control variables as piecewise linear functions. The discretized state and control variables are stored in a finite-dimensional vector $\boldsymbol{y}$, together with the final time variables for each stage. The problem may be solved by using different stages, each involving a different set of state and algebraic equations representing different configurations of the aircraft or different vehicles, such as a combination of an aircraft and a missile. The multistage formulation used was developed in a previous study.[5]

Received 2 April 2000; revision received 28 April 2000; accepted for publication 4 May 2000.

*Professor, Department of Aeronautics. Member AIAA.

By defining an objective function such as minimum time or fuel used for a given mission, it is now possible to formulate the optimization problem as

$$\min_{y} f_0(\boldsymbol{y}) \tag{12}$$

$$\boldsymbol{c}_{\min} \leq \begin{pmatrix} \boldsymbol{c}(\boldsymbol{y}) \\ \boldsymbol{C}\boldsymbol{y} \\ \boldsymbol{y} \end{pmatrix} \leq \boldsymbol{c}_{\max} \tag{13}$$

where f_0 denotes the objective function and $\boldsymbol{c}(\boldsymbol{y})$ denotes the discretized state and algebraic constraints. The matrix $\boldsymbol{C}$ in Eq. (13) define linear constraints. The linear constraints are used to define the interface condition between stages and more complex final conditions. Equality constraints are enforced by simply setting the lower and upper bounds equal.

Test Case

The aircraft model is based on the F-16 model developed in an European Consortium for Advanced Training in Aerospace project[6] at DaimlerChrysler Aerospace AG in 1997. The performance model was developed by using data available in the open literature and computational models used for preliminary aircraft design. The model was then compared with test results from various open sources.[7]

The data are assumed to be relevant for an F-16C with a PW-220 engine but should not be considered accurate enough for assessing the detailed performance of the aircraft. In this study, the model is simply used to demonstrate a trajectory optimization test case for high-performance military aircraft such as the F-16.

Minimum Fuel Turn

Flutter testing is usually performed at low altitude and very high speed to give high dynamic pressure. Because the drag of external stores is quite high, preventing the aircraft from reaching maximum speed, it is not unusual that this type of flight testing occurs without external drop tanks. Consequently, the use of minimum fuel during a flutter test mission is quite important. Because the flutter test itself requires maximum thrust with resulting high fuel burn, most of the savings are possible during the ferry flight to and from the test area and possibly in a turn between test runs in a given area.

The particular problem studied here is to minimize the fuel used for a 180° turn from supersonic to supersonic speed at a given altitude. The scenario is that the aircraft is flying at an altitude of 3 km with indicated airspeed 1350 km/h when approaching the end of the test area. Consequently, the aircraft is required to turn 180° to perform another flutter test on the way back.

The optimization problem is to find the three-dimensional trajectory that uses the minimum fuel for the turn. The initial and final conditions of the turn are given in Table 1.

For comparison, a reference trajectory is also computed where the altitude is maintained constant at 3 km throughout the turn but the remaining state and control variables are obtained by using optimization. The optimal and reference trajectories are shown in a three-dimensional view in Fig. 1. The dashed line shows the reference trajectory, and the optimal trajectory is shown as a solid line. The optimal trajectory involves an immediate reduction of thrust and a fairly steep climb up to an altitude of 10 km. The turn is fairly continuous with a significant bank angle through the entire trajectory as shown in Fig. 2. It could appear from Fig. 2 that the bank angle ϕ is not zero at the first and the last time step. However, the bank angle is zero by definition from the specified initial and final conditions, but the optimal solution gives a significant bank angle (10 deg) in the second time step corresponding to a roll rate of about 100 deg/s. A smoother solution can be obtained at the end points, if a constraint is enforced on the maximum roll rate or if a more elaborate model of the aircraft is used.

Table 1 Initial and final conditions for the turn

Variable	Initial	Final
V_i, km/h	1350	1350
h, km	3	3
m_f, kg	1950	Maximum
γ, deg	0	0
ψ, deg	0	180

Table 2 Results for the optimal and the reference trajectory

Variable	Optimum path	Inplane path
m_{fuel}, kg	101	172
t_f, s	140	72
h, km	$3 \leq h \leq 10.2$	3

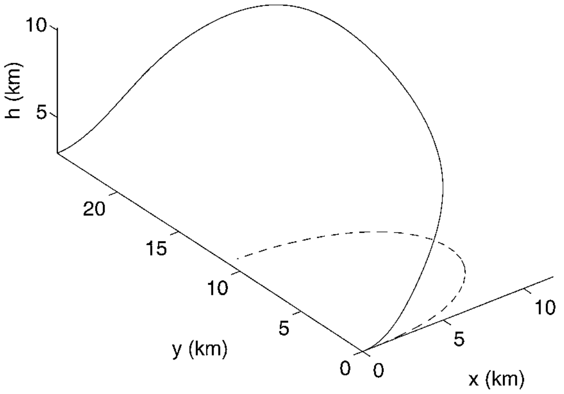

Fig. 1 Three-dimensional view of the optimal and the reference turn.

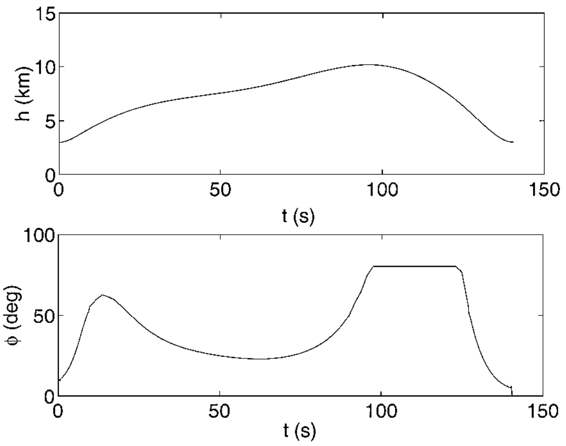

Fig. 2 Optimal state variables.

The most important data for the two trajectories is given in Table 2. The fuel savings obtained by flying the optimal trajectory is quite significant, giving a reduction of approximately 40%. However, the time needed to complete the turn becomes twice as large.

Summary

Solutions to trajectory optimization problems in three dimensions may be quite nonintuitive. In two dimensions it is in many cases quite possible to find close to optimal trajectories by using only a graph of the specific excess power,[8] at least for minimum time to climb problems. In three dimensions and for multistage trajectories, this type of manual approach becomes even less attractive. Consequently, the numerical optimization approach becomes more attractive as the problem complexity increases.

A significant difficulty with trying to test a complex three-dimensional trajectory is the ability of the pilot to follow the computed state and control variables. In two dimensions, this can be accomplished, particularly after some practice in a simulator.[2] This increases the importance of considering the use of an autopilot for flying the trajectory and thus reducing the pilot workload. Currently, optimal trajectories are computed in a desktop computer, usually requiring a skilled operator to solve a given problem. To reach the goal of using an autopilot for following the optimal trajectory, the numerical method needs significant improvement in two areas.

First, although microprocessors, such as the Intel Pentium II/III, quickly become more powerful, it is still not quite possible to solve a problem in real time. Currently, the numerical implementation used in this Note runs on a 400 MHz Pentium laptop running the Linux operating system. At best, a given problem of 500–1000 variables can be solved in 15–20 s. This type of performance is achieved by solving the problem on several discretization grids, starting with a coarse grid and then by using this solution as initial approximation on the finer grids. To significantly improve this performance, it is necessary to use a second-order method with exact second derivatives instead of the quasi-Newton method[9] used here.

Another issue that needs to be resolved concerns the robustness of the solution method. It is in principle impossible to a priori estimate how many iterations are needed to solve a nonlinear problem. Small changes to a given problem may drastically increase the amount of computing necessary to solve the problem. As a first step toward finding the optimal trajectory in real time, one can attempt to update a precomputed trajectory when initial and final conditions are perturbed. This way it may become possible to follow a moving target and also adjust for minor inaccuracies in performance modeling and atmospheric data. In principle this means that the solution of the nonlinear optimization problem approaches a closed-loop control method, as opposed to the open-loop approach used today.

Acknowledgments

This project is financially supported by the Swedish Defense Materiel Administration (FMV) project 61969-LB104302. The author is most grateful for the support provided by the staff of FMV: KCFlyg, in particular Martin Näsman. The author is also most grateful for the support of Philip Gill at University of California at San Diego and Walter Murray and Michael Saunders at Stanford University.

References

[1] Hargraves, C. R., and Paris, S. W., "Direct Trajectory Optimization Using Nonlinear Programming and Collocation," *Journal of Guidance, Control, and Dynamics*, Vol. 10, No. 4, 1987, pp. 338–342.

[2] Ringertz, U. T., "Flight Testing an Optimal Trajectory for the Saab J35 Draken," *Journal of Aircraft*, Vol. 37, No. 1, 2000, pp. 187–189.

[3] Etkin, B., and Reid, L. D., *Dynamics of Flight, Stability and Control*, Wiley, New York, 1996.

[4] Brenan, K. E., "Differential-Algebraic Equations Issues in the Direct Transcription of Path Constrained Optimal Control Problems," The Aerospace Corp., Aerospace Rept. ATR-94(8489)-1, 1993.

[5] Ringertz, U. T., "Multistage Trajectory Optimization Using Large-Scale Nonlinear Programming," TR 99-25, Royal Inst. of Technology, Dept. of Aeronautics, 1999.

[6] Pouvreau, L., "Re-Engineering of Combat Aircraft: Lockheed-Martin F-16C," Daimler-Benz Aerospace, ECATA-Junior Multinational Team Project, 1997.

[7] Webb, T. S., Kent, D. R., and Webb, J. B., "Correlation of F-16 Aerodynamics and Performance Predictions with Early Flight Test Results," *AGARD Conference Proceedings CP-242*, Advisory Group for Aerospace Research and Development, Neuilly sur Seine, France, 1978.

[8] Vinh, N., *Flight Mechanics of High-Performance Aircraft*, Cambridge Aerospace Series 4. Cambridge Univ. Press, Cambridge, England, U.K., 1993.

[9] Gill, P. E., Murray, W., and Saunders, M. A., User's Guide for SNOPT 5.3: A FORTRAN Package for Large-Scale Nonlinear Programming, Rept. NA 97-X, Dept. of Mathematics, Univ. of California, San Diego, CA, 1997.

Computational Analysis of F-15 Forebody Flow at High Alpha

Kenneth E. Wurtzler*
U.S. Air Force Research Laboratory,
Wright–Patterson Air Force Base, Ohio 45433-7913

Introduction

WITH flight envelopes being expanded because of changing tactics and engineering ability, fighter aircraft are designed and expected to fly at higher angles of attack and maintain directional control. However, the vertical tails become surrounded by turbulent, dead air and are limited in their directional control capability at high angles of attack. Relatively small side forces on the nose, even at zero sideslip, can dominate directional stability and create large yawing moments. These small side forces are a result of asymmetrical shedding of the forebody vortices. Small surface imperfections such as radome gaps, dents, and sharp paint depth mismatches can affect the strength and path of the vortices. The resultant net yawing moment can then increase, and the aircraft becomes unstable—an unsteady phenomenon that can be catastrophic.

In this study the F-15 forebody with varying tip geometries was modeled to match the geometry from a wind-tunnel test.[1] The three geometries were bump, bump with strakes, and bump with tabs. The length of the full-scale forebody section was 13.4 ft with the aft end being blunt. The grid-generation package VGRIDns[2] was used to generate an unstructured tetrahedral grid. The bump grid consists of 1.6 millions cells with the bump modeled on the lower left quadrant of the forebody, near the apex, by creating a small ridge of 0.3 in. maximum height on the surface. The bump with strakes grid contains 2.4 million cells with the strakes modeled as thin wedges. Each strake was 10 in. long and 1 in. wide. The bump with tabs grid consists of 1.8 million cells. The tabs consist of the first inch of the strakes. Cells are clustered near the strakes and tabs. Far-field boundaries for all grids are 10 forebody lengths away.

Cobalt$_{60}$ (Ref. 2) is a parallel, implicit unstructured flow solver, which employs Godunov's first-order accurate, exact Riemann method. Second-order spatial accuracy, second-order accurate implicit time stepping, viscous terms, and turbulence models have been added to this procedure. Cobalt$_{60}$ uses a finite volume, cell-centered approach. Arbitrary cell types can be used, and a single grid can be composed of a variety of cell types. The implicit algorithm in Cobalt$_{60}$ was implemented and demonstrated by Tomaro et al.[3] in 1997. The development of the parallel version of Cobalt$_{60}$ was reported by Grismer et al.[4] Domain decomposition is the basis for the parallel code with each processor operating on a subsection (zone) of the original grid.

Results and Discussion

Flow conditions are the same as wind-tunnel test conditions.[1] The dynamic pressure was 10 psf, the corresponding flow velocity was 92 ft/s, and the Reynolds number nearly 5×10^4/in. Flow was assumed to be laminar. For most of the runs, 48 nodes were used, which gave a timing of 13.2 μs/cell/iteration. Several runs utilizing 140 nodes took 4.2 μs/cell/iteration. For a 2-million cell grid this amounts to 26.2 s/iteration for 48 nodes and 8.4 s/iteration for 140 nodes. All force and moment data presented in this Note are in body-axis system and are resolved at the F-15's aerodynamic

Received 31 January 1999; revision received 22 October 1999; accepted for publication 31 October 1999. This material is declared a work of the U.S. Government and is not subject to copyright protection in the United States.

*Aerospace Engineer, Computational Sciences Branch, Air Vehicles Directorate.