References

[1]Anderson, G. M., "Comparison of Optimal Control and Differential Game Intercept Missile Guidance Laws," *Journal of Guidance and Control*, Vol. 4, No. 2, 1981, pp. 109–115.

[2]Bryson, A. E., and Ho, Y. C., *Applied Optimal Control*, Hemisphere, New York, 1975.

[3]Baron, S., "Differential Games and Optimal Pursuit-Evasion Strategies," Ph.D. Dissertation, Harvard Univ., Cambridge, MA, 1965.

[4]Friedman, A., *Differential Games*, Academic, New York, 1971.

[5]Gill, A., and Sivan, R., "Optimal Control of Linear Systems with Quadratic Costs Which are not Necessarily Positive Definite," *IEEE Transactions on Automatic Control*, AC-14, Feb. 1969, pp. 83–86.

Dense–Sparse Discretization for Optimization and Real-Time Guidance

Renjith R. Kumar* and Hans Seywald†
Analytical Mechanics Associates, Inc.,
Hampton, Virginia 23666-1398

Introduction

A COMMON practice for onboard implementation of optimal or near-optimal guidance strategies is to solve the trajectory optimization problem offline for various boundary conditions. Numerous reference trajectories and the associated gains are then evaluated based on neighboring optimal control concepts[1] and are stored onboard. These reference trajectories and gains are used onboard with state and target observation or estimation to quickly evaluate control actions via simple linear feedback laws. However, this requires extensive ground-based analysis and onboard storage capacity.

For rapid trajectory prototyping, the safest and most robust approaches are the direct methods. These methods rely on a finite dimensional discretization of the optimal control problem to nonlinear programming problems. Even though these methods do not enjoy the high precision and resolution of indirect methods, their convergence robustness makes them the method of choice for most practical applications. Moreover, these methods do not require the advanced mathematical skills required to pose and solve the variational problem. With the advent of high-power computing and excellent nonlinear programming tools, direct methods have been used extensively to calculate optimal trajectories offline. Some of the most common approaches include control discretization, as in the program to optimize simulated trajectories (POST) software,[2] collocation-based methods[3] as in the optimal trajectories by implicit simulation (OTIS) code, and a recently introduced trajectory optimization via differential inclusions (TODI)[4] method.

Although all of the cited direct approaches have been used for offline optimization of trajectories, real-time online guidance strategies using these methods have not yet been developed and implemented. This Note proposes such a guidance scheme. In each control evaluation step a rough near-optimal solution to the current optimal control problem is generated using an intuitive dense–sparse discretization. The direct method employed throughout this paper is the differential inclusion approach in conjunction with NPSOL[5] as the optimization engine, although any other robust optimization technique could be utilized. A minimum time-to-climb problem[6] of an F-15 aircraft is used as an example to illustrate the concept.

Dense–Sparse Discretization

If one is driving from New York to Los Angeles, and the objective is to minimize fuel or time, would it really be necessary to make instantaneous control actions to account for every curve and pothole miles away? How much does the performance index improve by doing this? Although the optimal solution is influenced by every curve and pothole miles away, intuitively it is clear that the degradation in performance by neglecting these would be only marginal.

By experience, TODI captures the general profile of the optimal solution well even with a very small number of nodes. (The larger the number of nodes, the more precise the solution, but the CPU time and memory required to solve the nonlinear programming problem goes up nonlinearly, i.e., approximately as the third power of the number of parameters for a nonsparse solver and as the second power for a sparse solver.) Hence, it is proposed to discretize the trajectory with nodes densely placed near the current time to capture immediate dynamics well and with nodes sparsely placed for the rest of the trajectory to approximate the overall trend. This simple idea is used to determine the near-optimal control action at the current time. Clearly, a whole range of alternate methods to achieve the desired high resolution near initial time can be conceived, but for the ease of presentation, the remainder of this Note is restricted exclusively to the dense–sparse discretization just introduced. The algorithm is discussed in the following section.

Let the speed and computing power available limit the discretization for the problem at hand to, say, eight nodes. The speed and CPU limit may be the limits on a workstation/personal computer performing offline optimization or it may correspond to the limitations of an onboard guidance computer. Place N_D dense nodes (say, four) close to initial time and place N_S sparse nodes (four) for the rest of the trajectory. The nodal density may be chosen by the user. The resulting nonlinear programming problem with eight nodes is then solved. This concludes step 1 of the algorithm as shown in Fig. 1.

During step 2 of the optimization scheme, a displacement integer m is defined that specifies the number of nodes to accept as optimal.

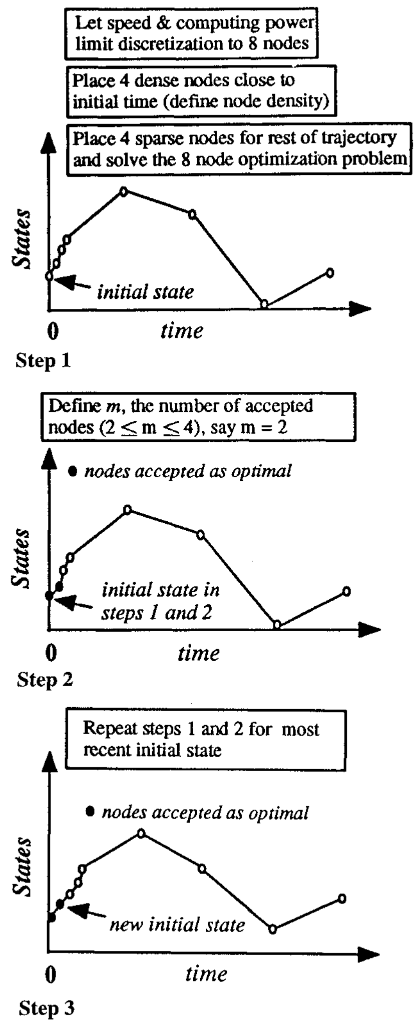

Fig. 1 Steps of optimization/guidance algorithm.

Received Nov. 28, 1994; revision received Aug. 1, 1995; accepted for publication Sept. 25, 1995.

*Senior Engineer. Member AIAA.

†Supervising Engineer. Member AIAA.

This integer m is bounded by $2 \leq m \leq N_D$. Thus, the first m nodes are accepted as optimal. This concludes step 2 of the optimization procedure. The state values at the mth node accepted as optimal are prescribed as the initial conditions for the rest of the trajectory. The remainder of the trajectory is again discretized as shown in step 3. Steps 1 and 2 are again repeated for this remainder trajectory. The algorithm may be stopped using ad hoc schemes. An example of one such scheme is to stop the sequential procedure when the density of dense nodes is less than or equal to the density of sparse nodes.

For real-time onboard guidance, the initial time is synonymous with the current time. The current control action is obtained from the control profile close to the initial time of the solution obtained using the dense–sparse discretization. This control evaluation would be repeated for subsequent times as the guidance scheme proceeds. The initial states prescribed in each optimization subproblem correspond to the observed or estimated current states. Nonstationary targets could cause the target states to change during the guidance scheme.

F-15 Climb-to-Dash Example

The problem of steering an F-15 aircraft using bounded throttle setting $\eta(t)$ and bounded vertical load factor $n(t)$ from prescribed initial conditions to the farmost point to the right of the level flight envelope (dash point) in minimum time is used to test the algorithm proposed in this Note. Explicitly, the problem can be stated in Mayer form as follows.

Minimize t_f subject to the equations of motion

$$\dot{h} = v \sin\gamma \tag{1}$$

$$\dot{E} = (\eta T - D)(v/mg) \tag{2}$$

$$\dot{\gamma} = (g/v)(n - \cos\gamma) \tag{3}$$

the control constraints

$$0 \leq \eta \leq 1 \qquad \text{and} \qquad -n_{\max} \leq n \leq +n_{\max} \tag{4}$$

and the initial and terminal boundary conditions

$$\begin{aligned} h(0) &= 5 & h(t_f) &= 12{,}119.3 \\ E(0) &= 2668 & E(t_f) &= 38{,}029.2 \\ \gamma(0) &= 0 & \gamma(t_f) &= 0 \end{aligned} \tag{5}$$

where the altitude h (in meters), the specific energy E (in meters) replacing velocity v, and the flight-path angle γ (in radians) are the state variables. The mass m (in kilograms) of the aircraft and the gravitational acceleration g (in meters per second squared) are assumed to be constant, namely, $m = 16{,}818$ and $g = 9.80665$. The velocity v is defined by $v = \sqrt{[2g(E-h)]}$. The initial states are picked rather arbitrarily and represent the state of the aircraft shortly after takeoff. The final states represent level flight conditions at the dash point. The atmospheric density and speed of sound are functions of altitude. The lift is proportional to load factor, and the drag D is dependent on velocity and altitude and has quadratic dependence on load factor. The maximum thrust T is also a function of altitude and velocity. The detailed atmospheric, aerodynamic, and propulsive models used for this study can be found in Ref. 6.

Results and Discussion

For the F-15 minimum time-to-climb problem, the dense–sparse discretization method was used in conjunction with TODI to develop a simple heuristic guidance algorithm. It was assumed that the maximum number of nodes was eight. Then four dense nodes (separated by about 5 s, for a total flight time of ~269 s) were placed close to the initial time and four sparse nodes were placed for the rest of the trajectory. The integer m that denotes the number of accepted nodes was chosen as 2, the minimum value. The optimization/guidance problem was simulated as detailed in Fig. 1, until the density of dense nodes was less than or equal to the density of sparse nodes. This results in approximately 46 sequential optimization runs to complete the guidance simulation.

Figure 2 compares the guidance solution of the rapidly changing flight-path angle state history of the F-15 aircraft with a 101-node

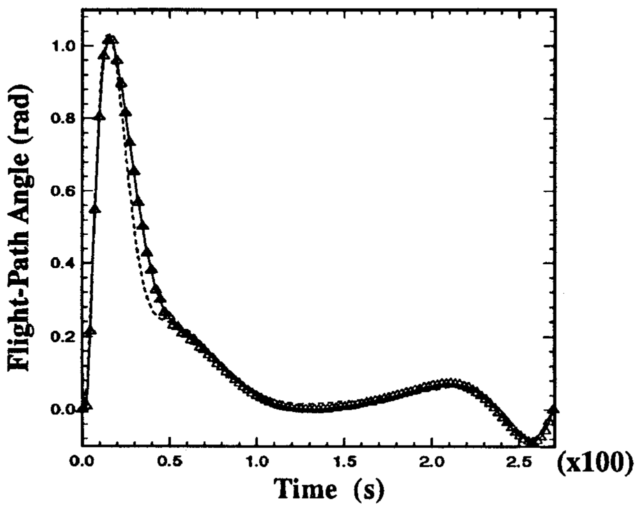

Fig. 2 F-15 flight-path angle profile; optimal (- - -) vs dense sparse (ΔΔΔ).

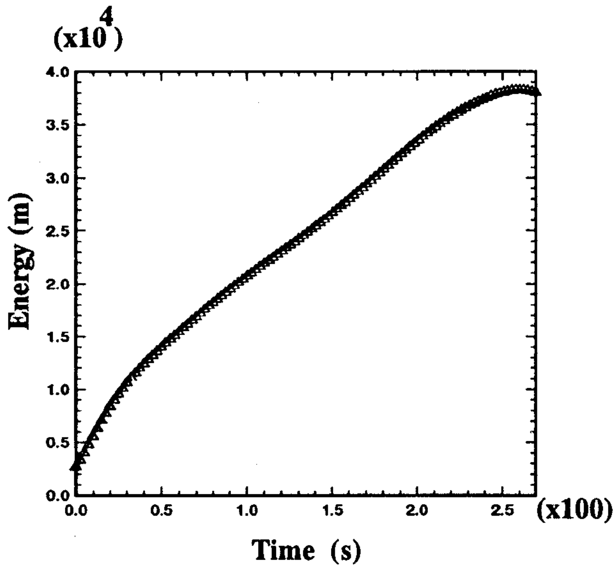

Fig. 3 F-15 energy profile; optimal (- - -) vs dense sparse (ΔΔΔ).

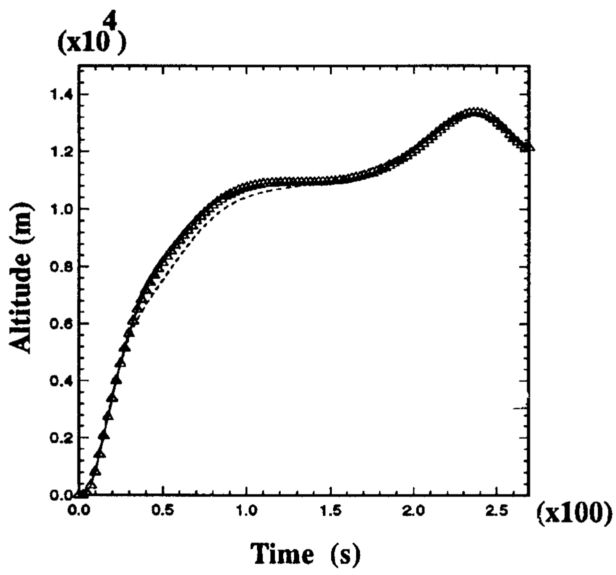

Fig. 4 F-15 altitude profile; optimal (- - -) vs dense sparse (ΔΔΔ).

optimal solution generated by TODI. Note that the 101-node optimal solution could not be distinguished from the variational solution generated by solving the two-point boundary-value problem. It can be clearly seen that the guidance scheme performs extremely well and the loss of performance, which is final time, is only about 1 s (optimal final time is 268.104 s vs guidance strategy final time of 269.148 s). Figures 3 and 4 compare the guidance solution to the optimal solution for the energy and altitude states, respectively. CPU time taken to complete the entire simulation on different computers clearly indicates the feasibility of this guidance scheme for real-time applications: SunSparc1, 101 s; SGI Indigo, 31 s; and SGI Onyx, 19 s.

This optimization/guidance strategy was also tested for the example problem with different values of m between 2 and 4, and the solutions were found to be quite insensitive. The feasibility of onboard implementation of this guidance strategy depends on the problem at hand, the convergence robustness of the algorithm, and the CPU speed.

Conclusions

A simple, heuristic, and intuitive dense–sparse discretization scheme in conjunction with the highly robust differential inclusion method was used to develop an optimization/guidance strategy utilizing a low number of nodes. This concept was tested on a minimum time-to-climb problem of an F-15 aircraft. The optimization/guidance strategy worked extremely well, and the reduction in performance of the near-optimal guidance strategy with respect to the optimal solution was approximately 1 s for a total flight time of 269 s. The CPU time required for computation of optimal solutions clearly indicates feasibility for real-time onboard implementation. The convergence robustness of the algorithm was excellent.

References

[1]Bryson, A. E., and Ho, Y. C., *Applied Optimal Control*, Hemisphere, New York, pp. 177–211.

[2]Brauer, G. L., Cornick, D. E., and Stevenson, R., "Capabilities and Applications of the Program to Optimize Simulated Trajectories (POST)," NASA CR-2770, Feb. 1977.

[3]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.

[4]Seywald, H., "Trajectory Optimization Based on Differential Inclusions," *Journal of Guidance, Control, and Dynamics*, Vol. 17, No. 3, 1994, pp. 480–487.

[5]Gill, P. E., Murray, W., Saunders, M. A., and Wright, M. H., "User's Guide for NPSOL: A Fortran Package for Nonlinear Programming," Dept. of Operations Research, TR SOL86-2, Stanford Univ., Stanford, CA, Jan. 1986.

[6]Seywald, H., "Optimal and Suboptimal Minimum Time-to-Climb Trajectories," *Proceedings of the AIAA Guidance, Navigation, and Control Conference* (Scottsdale, AZ), AIAA, Washington, DC, 1994, pp. 130–136 (AIAA Paper 94-3554).