[4]Singh, H., "Regional Pole Placement Methods via Linear Quadratic Regulator Theory," M.S. Thesis, Measurement and Control Engineering Research Center, College of Engineering, Idaho State Univ., Pocatello, ID, Dec. 1994.

Dynamic Boundary Evaluation Method for Approximate Optimal Lunar Trajectories

Matthew L. Rivas* and Bion L. Pierson†
Iowa State University, Ames, Iowa 50011-3231

Introduction

RESEARCH into low-thrust Earth–moon trajectories has increased recently.[1–3] The time and effort required to obtain an optimal low-thrust lunar trajectory depends greatly on the initial estimate of the trajectory used in the optimizing process. Pierson and Kluever[2] present an effective method for finding an initial estimate for a class of Earth–moon trajectories. However, their method requires the setup of tabular data before the initial estimate can be found. This data table must be recomputed for each new initial thrust-to-weight ratio, radius of the initial low Earth orbit (LEO), or radius of the final low lunar orbit (LLO). This Note describes a dynamic boundary evaluation (DBE) method that replaces the data table construction and subsequent interpolation with an on-line iterative numerical process.

The optimal low-thrust lunar trajectory problem treated here involves a planar transfer from a circular LEO to a circular LLO. A specified thrust-coast-thrust sequence is assumed using constant thrust. The spacecraft thrusts away from the LEO, in an increasing spiral, until enough energy is gained to make the lunar rendezvous. The spacecraft then coasts to the vicinity of the lunar sphere of influence, where the spacecraft thrusts, in a decreasing spiral, to the desired LLO. The minimum-fuel solution is the one that accomplishes this transfer with the minimum time spent in the two thrusting periods, because the assumed constant thrust implies a constant mass flow. The Pierson–Kluever method[2] uses maximum-energy spirals, described later, to approximate both the Earth-departure and the lunar-capture thrusting arcs. This composite trajectory, consisting of the maximum-energy spirals plus an optimal translunar coast, is then used to estimate the fully minimum-fuel transfer from a circular LEO to a circular LLO. This estimate is very accurate in terms of performance; i.e., the fuel spent is very close to that for the fully optimal case, and the control time histories for the thrust-steering angle are adequate to ensure convergence to a solution of the full problem.

Implementation of the DBE Method

The approximate optimal lunar trajectory consists of a nonthrusting arc that patches two thrusting arcs together. The two thrusting portions of the trajectory include the departure from LEO and the capture into LLO. These two thrusting phases are modeled by maximum-energy spirals, and patching the trajectories then matches positions and velocities at both the end of the Earth departure and the beginning of the lunar capture. The problem statement for the coast-patch problem can be written as follows: Find r_1 (the radius at the beginning of the coast phase), θ_1 (the polar angle at the beginning of the coast phase), m_f (the final mass), and a nondimensional time parameter $\alpha(t = \alpha\tau)$ that minimize

$$J = t_{\mathrm{dep}} + t_{\mathrm{cap}} \tag{1}$$

subject to restricted three-body dynamics[4] with no thrusting,

$$\frac{\mathrm{d}x}{\mathrm{d}\tau} = \alpha f(x), \qquad 0 \le \tau \le 1 \tag{2}$$

and the boundary conditions

$$r(0) = r_1 \tag{3}$$

$$\theta(0) = \theta_1 \tag{4}$$

$$V_r(0) = f_1(r_1) \tag{5}$$

$$V_\theta(0) = f_2(r_1) \tag{6}$$

$$V_r(1) = g_1(r_2, m_f) \tag{7}$$

$$V_\theta(1) = g_2(r_2, m_f) \tag{8}$$

where $(r, \theta, V_r, V_\theta)$ are the radius, polar angle, and radial and tangential velocities of a standard polar coordinate system, respectively; r_2 is the radius at the beginning of the lunar capture; and t_{dep} and t_{cap} are the departure and capture time durations, respectively.

$$t_{\mathrm{dep}} = t_{\mathrm{dep}}(r_1) \tag{9}$$

$$t_{\mathrm{cap}} = t_{\mathrm{cap}}(r_2, m_f) \tag{10}$$

Use of the time transformation $t = \alpha\tau$ allows us to convert a variable end-time problem in t into a fixed end-time problem in τ. The functions f_1 and f_2 provide the boundary conditions at the end of the Earth departure and require one input parameter, the radius r_1, whereas the functions g_1 and g_2 provide the boundary conditions at the beginning of the lunar capture and require two input parameters, the radius r_2 and the final mass m_f. These functions can be determined in different ways, one of which is to construct, from a set of maximum-energy solutions of fixed duration, a table of time-of-flight and radial and tangential velocity components for the specific initial thrust-to-weight ratio used and then simply to interpolate within this table to find the boundary conditions[2] for the translunar coast trajectory.

On-Line Approach

We first tried to solve these boundary conditions on-line using a root-finding process.[5] The maximum-energy problem mentioned earlier is for a fixed final time and free final radius. The related problem of fixing the final radius and finding the velocity that maximizes the final total energy is actually a maximum final-speed problem. The use of maximum final-speed rather than maximum final-energy trajectories for the escape/capture approximations is possible but beyond the scope of this Note. The problem of finding the maximum-energy spiral that ends at a specified radius is equivalent to finding the time-of-flight t_{dep} that satisfies

$$r_{\mathrm{desired}} - r(t_{\mathrm{dep}}) = 0 = F(t_{\mathrm{dep}}) \tag{11}$$

The expression $r(t_{\mathrm{dep}})$ denotes the final radius found by solving a maximum-energy problem with a time-of-flight of t_{dep}. Equation (11) requires evaluating a function, however complicated, that is the difference between the desired final radius and the calculated final radius. The departure time that satisfies this equation completely determines that trajectory, and the result can be used as the initial conditions for the coast-patch problem, Eqs. (1–8). An iterative root-finding algorithm then can be used to solve Eq. (11). This method proved effective, but its performance was limited by the necessary computational burden of calculating a large number of maximum-energy spiral solutions for each iteration of the optimization. We then derived a new method of finding the boundary conditions by incorporating the root-finding process into the optimal solution of the approximate translunar problem.

A still more direct approach is to treat the radius r as the independent variable rather than time. Then, the final radius is automatically the desired one and we have a fixed end-time problem. This conversion is possible because the radius is a nondecreasing function

Received July 10, 1995; revision received Feb. 19, 1996; accepted for publication March 11, 1996.

*Graduate Research Assistant, Department of Aerospace Engineering and Engineering Mechanics.

†Professor, Department of Aerospace Engineering and Engineering Mechanics. Associate Fellow AIAA.

for the maximum-energy spiral. However, other trajectories near the maximum-energy solution may not have this same property, and the transformed differential equations can therefore become numerically unstable. These trajectories are usually computed iteratively, which allows the possibility of a nonviable trajectory being considered. Also, the transformed equations of motion are singular for $r = r_o$ because the initial radial velocity is zero for circular target orbits. Numerical experiments starting the iterative process at a viable trajectory and with an extremely small nonzero $V_r(r_o)$ have not been encouraging, although an analytical asymptotic treatment remains a possible topic for future research.

An implementation of the previously mentioned iterative on-line method requires an optimization solution to be found within another optimization problem. These types of embedded optimal problems require special consideration and are difficult to solve using the traditional Fortran 77 programming language, because Fortran does not allow recursive function calls. It is necessary either to use another optimization program or to change the way the compiler creates the executable code. Because neither of these options is desirable, a different language, specifically C and then C++, was used for all of the numerical calculations. Care also must be taken that the embedded optimal problem, here the maximum-energy spiral, has a reasonable chance of convergence and that a restart is allowed. This is the primary difficulty involved in implementing the DBE method and the major cause of why the DBE method may not converge to a solution for the overall problem.

Modified Maximum-Energy Method

The solution of the maximum-energy problem usually involves solving the two-point boundary-value problem obtained by applying optimal control theory. The resulting final boundary condition for the costate eqations are[2,5]

$$\lambda_r(1) = -\left(\mu/r_1^2\right) \tag{12}$$

$$\lambda_{V_r}(1) = -V_r(1) \tag{13}$$

$$\lambda_{V_\theta}(1) = -V_\theta(1) \tag{14}$$

A similar boundary-value problem can be formed by analyzing a fixed-final-radius maximum-speed optimal problem: Given r_{desired}, find $u(\tau)$, $0 \le \tau \le 1$, and α that minimize

$$J = -\frac{V_r^2 + V_\theta^2}{2} \tag{15}$$

subject to

$$\frac{\mathrm{d}r}{\mathrm{d}\tau} = \alpha V_r \tag{16}$$

$$\frac{\mathrm{d}V_r}{\mathrm{d}\tau} = \alpha\left(\frac{V_\theta^2}{r} - \frac{\mu}{r^2} + \frac{T}{m}\sin u\right) \tag{17}$$

$$\frac{\mathrm{d}V_\theta}{\mathrm{d}\tau} = \alpha\left(-\frac{V_r V_\theta}{r} + \frac{T}{m}\cos u\right) \tag{18}$$

and

$$\Psi = r(1) - r_{\text{desired}} = 0 \tag{19}$$

The boundary conditions on the costate variables for this problem, obtained after applying optimal control theory, are

$$\lambda_r(1) = \nu - \left(\mu/r_1^2\right) \tag{20}$$

$$\lambda_{V_r}(1) = -V_r(1) \tag{21}$$

$$\lambda_{V_\theta}(1) = -V_\theta(1) \tag{22}$$

where ν is a constant multiplier. The boundary conditions differ from Eqs. (12–14) only in the final value of the radial costate variable λ_r. The resulting trajectory of either the maximum-energy spiral or the restricted maximum-speed problem is completely determined by the costate boundary conditions and the time-of-flight parameter α. The maximum-speed problem then can be used to solve the maximum-energy problem with a specified final radius by enforcing Eqs. (19), (21), and (22). This is the DBE method that is the subject of this Note.

A similar type of maximum-speed problem can be formed for the lunar-capture boundary conditions with the only added complexity being the unknown quantity of the final mass. Two basic ways of dealing with this have been implemented. The first is to treat the final mass as an additional design parameter and add a corresponding constraint to ensure that the linear mass equation is satisfied for the entire trajectory. This method is necessary when tables are used for the boundary-condition evaluation because it is the only way to determine the final mass accurately. The second way of dealing with the final mass, which becomes possible when the boundary conditions are evaluated on-line, is to calculate the mass when needed, using the linear mass relation. This can be done because the mass at the beginning of the lunar capture is known because the exact Earth departure is now known. This has the advantage of decreasing the number of design variables and constraints. However, the technical difficulties in implementing this outweigh any improvement in performance. Therefore, for simplicity, the first method of adding an extra design variable and constraint is recommended.

Comparison of Solution Methods

The main advantage of the DBE method is that it eliminates the necessity of constructing a table of flight times and velocity components for the boundary conditions of the coast-patch problem. The construction of such a table is very labor intensive and can lead to the introduction of extraneous errors. Also, the time and effort required to construct such a table for specific trajectory flight parameters does not help in the solution of other problems with different parameters. Changing the initial thrust-to-weight ratio, for example, requires the formation of an entirely new table that cannot be formed from information contained in the old table. The same can be said of changes in the initial and final circular-parking-orbit altitudes and the initial mass.

A second advantage of the DBE method is that it reduces the overall time required to solve the coast-patch problem. For the tabular method, the total time includes both the time to construct the table and the processing time of finding the solution once the table is available. This can vary a great deal depending upon any prior knowledge of the solution. The table can be smaller and use a finer grid if the general range of the solution is known. However, if no knowledge of the solution is available, the table must be relatively large and use a coarse grid to ensure that the solution boundary conditions lie within the tabular values. The DBE method does not require any time other than that required for solving the maximum-energy problem. Also, the accuracy of the boundary conditions will be ensured as long as it is possible to find a viable solution for the given input parameter.

An example solution illustrates some of these advantages. An approximate solution for a lunar mission is sought starting in a circular LEO of 315 km and terminating in a circular LLO of 100 km. The spacecraft has an initial thrust-to-weight ratio of 0.003, an initial mass of 100,000 kg, and an electric thruster with a specific impulse of 10,000 s. The optimal trajectories were solved by using a nonlinear optimization algorithm to minimize the terminal-state error with a constraint tolerance of 10^{-8}. The flight paths were found using the Bulirsch–Stoer variable-step integration scheme with an accuracy of 10^{-10}. The tabular method's total CPU processing time, which includes the preprocessing time for the table setup, is 898.6 s, where 892.4 s is spent forming the table and only 6.2 s is spent on the solution of the approximation. The DBE method's total processing time, which does not have any preprocessing time, is 780.2 s. The actual solution time is much smaller for the tabular method than for DBE, which is expected because the DBE method requires the solution of two maximum-energy spiral problems for each evaluation of the objective function. The optimization histories of both methods for the solution of the translunar coast problem are nearly identical, i.e., each method required 23 iterations and produced nearly identical states at each iteration. The construction of the table takes many more user-hours, whereas the DBE method is virtually unassisted during its optimization process. We are not reporting these user-hours because of the difficulty in assigning an accurate metric to them.

Conclusions

The DBE method provides a fast and flexible way to evaluate the boundary conditions and performance index for the approximate translunar problem. No reprogramming is required to change specified system parameters, such as the initial thrust-to-weight ratio, and any interpolation errors associated with the tabular method are eliminated.

References

[1]Enright, P. J., and Conway, B. A., "Discrete Approximations to Optimal Trajectories Using Direct Transcription and Nonlinear Programming," *Journal of Guidance, Control, and Dynamics*, Vol. 15, No. 4, 1992, pp. 994–1002.

[2]Pierson, B. L., and Kluever, C. A., "Three-Stage Approach to Optimal Low-Thrust Earth-Moon Trajectories," *Journal of Guidance, Control, and Dynamics*, Vol. 17, No. 6, 1994, pp. 1275–1282.

[3]Kluever, C. A., and Pierson, B. L., "Vehicle-and-Trajectory Optimization of Nuclear Electric Spacecraft for Lunar Missions," *Journal of Spacecraft and Rockets*, Vol. 32, No. 1, 1995, pp. 126–132.

[4]Szebehely, V., *Theory of Orbits: The Restricted Problem of Three Bodies*, Academic, New York, 1967, pp. 8–15.

[5]Rivas, M. L., "Minimum-Fuel Low-Thrust Lunar Trajectories for a Range of Initial Thrust-to-Weight Ratios," M.S. Thesis, Dept. of Aerospace Engineering and Engineering Mechanics, Iowa State Univ., Ames, IA, 1993.

Nonlinear Autopilot Design for Bank-to-Turn Steering of a Flight Vehicle

Liang-Boon Wee*
Defence Science Organization, 118 230 Singapore

I. Introduction

IN bank-to-turn (BTT) steering of a flight vehicle, the control system continuously banks the flight vehicle to minimize the side-slip angle and, hence, the asymmetry of its vortex wake. Consequently, the flight vehicle can maneuver at a higher angle of attack, thus increasing the lift capability and accomplishing a turn in a shorter time. Although the BTT steering offers much theoretical improvements in maneuverability, practical flight vehicles that use this steering are still not able to achieve the desired results. This is because of inherent limitations in the autopilot designs. In recent years, nonlinear autopilot designs based on dynamic inversion have been developed.[1–5] In this Note, we present a nonlinear control law that takes into consideration the full nonlinear dynamics of the flight vehicle. The approach in autopilot design is based on the differential geometric control theory popularized by Brockett.[6] We show that the nonlinear coupling between the kinematics and dynamics can essentially be canceled using feedback and kinematic inversion in the control law. The use of the kinematic inversion essentially removed the nonminimum phase zero associated with the airframe dynamics. The nonlinear control law developed requires only measurements of the roll angle, incidence angle, side-slip angle, and body angular rates.

II. Problem Formulation

We present a new formulation of the dynamical equations of motion. The following assumptions are made in the development of the nonlinear autopilot.

Received April 15, 1995; revision received Nov. 21, 1995; accepted for publication Feb. 9, 1996.

*Engineer, Advanced Systems Division, 20 Science Park Drive. Member AIAA.

Assumption 1. The control surface effectiveness terms are small compared to the aerodynamics derivatives and damping terms.

Assumption 2. The airframe is rigid.

Let α denote the incidence angle and β the side-slip angle. The evolution of incidence and side-slip angles is dependent on the body angular rates. Thus, these angles can be indirectly controlled via the body rates. We define a new set of output variables,

$$y_1 := \phi \tag{1}$$

$$y_2 := \tan\alpha \tag{2}$$

$$y_3 := \tan\beta \tag{3}$$

where ϕ is the roll angle. Given the desired roll angle and the desired incidence and side-slip angles, we can compute the desired output directly using the definitions. Let $\boldsymbol{y} := [y_1, y_2, y_3]^T$; then, the time derivative of the output vector is

$$\dot{\boldsymbol{y}} = \boldsymbol{C}(\boldsymbol{y})\boldsymbol{x} \tag{4}$$

where

$$\boldsymbol{x} := \begin{bmatrix} p \\ q \\ r \end{bmatrix} \tag{5}$$

$$\boldsymbol{C}(\boldsymbol{y}) := \begin{bmatrix} 1 & 0 & 0 \\ -y_3 & 1 + y_2^2 & -y_2 y_3 \\ y_2 & y_2 y_3 & -\left(1 + y_3^2\right) \end{bmatrix} \tag{6}$$

Suppose the system input variables are the equivalent aileron, elevator, and rudder command, denoted by $\boldsymbol{u}$. The evolution of the state vector $\boldsymbol{x}$ is given by Euler's equation and the equations describing the aerodynamic moments

$$\dot{\boldsymbol{x}} = \boldsymbol{f}(\boldsymbol{x}, \boldsymbol{y}) + \boldsymbol{g}(\boldsymbol{x})\boldsymbol{u} \tag{7}$$

where $\boldsymbol{f}(\boldsymbol{x}, \boldsymbol{y})$ and $\boldsymbol{g}(\boldsymbol{x})$ are functions of the aerodynamics and kinematics of the flight vehicle. The equations of motion of the system are completely described by Eqs. (4) and (7). Moreover, the kinematics and dynamics of these equations are tightly coupled and highly nonlinear.

III. Nonlinear Autopilot

We develop the nonlinear autopilot. The nonlinear differential geometric approach we adopt was first popularized by Brockett.[6] We rewrite the equations of motion as follows:

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

$$\dot{\boldsymbol{y}} = \boldsymbol{C}(\boldsymbol{y})\boldsymbol{x} \tag{9}$$

Since the control law does not appear in the output, Eq. (9), we differentiate the output with respect to time, and after some simplification we obtain

$$\ddot{\boldsymbol{y}} = \frac{\partial \boldsymbol{C}(\boldsymbol{y})\boldsymbol{x}}{\partial \boldsymbol{y}}\boldsymbol{C}(\boldsymbol{y})\boldsymbol{x} + \boldsymbol{C}(\boldsymbol{y})\boldsymbol{f}(\boldsymbol{x}, \boldsymbol{y}) + \boldsymbol{C}(\boldsymbol{y})\boldsymbol{g}(\boldsymbol{x})\boldsymbol{u} \tag{10}$$

Thus a nonlinear control law, which completely decouples the output equation, can be written as

$$\boldsymbol{u} = [\boldsymbol{C}(\boldsymbol{y})\boldsymbol{g}(\boldsymbol{x})]^{-1}\left[-\frac{\partial \boldsymbol{C}(\boldsymbol{y})\boldsymbol{x}}{\partial \boldsymbol{y}}\boldsymbol{C}(\boldsymbol{y})\boldsymbol{x} - \boldsymbol{C}(\boldsymbol{y})\boldsymbol{f}(\boldsymbol{x}, \boldsymbol{y}) + \boldsymbol{\nu}\right] \tag{11}$$

where $\boldsymbol{\nu} \in \boldsymbol{R}^3$ is an auxiliary input. The closed-loop equation of motion using the control law is given by

$$\ddot{\boldsymbol{y}} = \boldsymbol{\nu} \tag{12}$$