Optimal Re-Entry from Orbit Using Dynamic Programming

Man Hyung Lee

Pusan National University
Department of Precision
Engineering

1. Introduction

This paper demonstrates a dynamic programming solution to the optimal re-entry from orbit problem. The problem takes into account the control of the accelerations experienced by the crew and the heating rate of the surface of the spacecraft. By using a weighting value between these two factors, the re-entry trajectory can be varied to account for differences such as varied thermal protection systems or payload specipic acceleration requirements.

The Mathematical description is presented and the problem is formulated using the Hamiltonian and the minimum principle. A dynamic programming solution is then presented that is similar to the method used by Dreyfuss and Cartino (1) to solve the aircraft minimum time-to-climb problem. A description of a computer program solution is presented with the results to date.

2. Problem Description

In order to successfully return a manned spacecraft from orbit it is necessary to be able to balance the surface heating rate and the acceleration effects experienced by the crew. New technologies are available that provide thermal protection systems that absorb and dissipate a large amount of heat thus allowing reduced accelerations upon the crew. Using an optimal re-entry trajectory allows one to best take advantage of any mission or vechicle specific requirements.

The problem is formulated in a fashion similar to that proposed by Greensite [2]. The spacecraft is considered to be a point mass with forces acting on it as shown in figure 1.

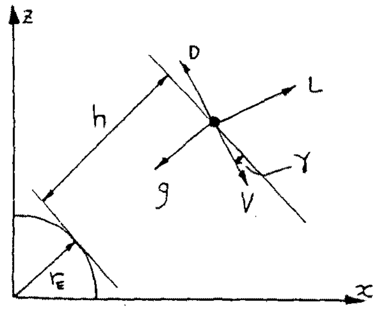

figure 1. Co-ordinate system for re-entry problem

The dynamical equations that describe the motion are as follows:

$$\dot{h} = -V\sin r \tag{1}$$

$$\dot{V} = g\sin r - \frac{D}{m} \tag{2}$$

$$\dot{r} = g\cos r/V - V\cos r/(r_e + h) - L/mV \tag{3}$$

where

h = altitude
V = velocity
r = flight path angle
g = gravitational acceleration
D = drag force
m = mass of vehicle
r_e = radius of spherical earth
L = lift force.

In order to simplify the problem somewhat, a constant flight path angle was assumed.

The rate of heating is given by

$$B = K_h \rho^{1/2} V^3 \qquad (4)$$

where K_h is a heating constant dependent upon the spacecraft and ρ is the atmospheric density. The acceleration sensed by the crew is given by

$$G = (L^2 + D^2)^{1/2}/m^2 \qquad (5)$$

As human tolerance levels depend on magnitude of acceleration and length of time applied, Greensite proposes using the acceleration squared which approximates a linear function for the endurance limit in the region of interest.

Equation (4) and (5) can then be used to form the following performance index:

$$J = \int_{t_0}^{t_f} (K_h \rho^{1/2} V^3 + K_w (L^2 + D^2)/m^2) dt \qquad (6)$$

where K_w is the weighting factor between the heating rate and the acceleration forces. Atmospheric density can be approximated by [3]

$$\rho = K_{sl} \, \mathrm{Exp}(K_e h) \qquad (7)$$

where K_{sl} is a reference density and K_e is an exponential decay factor. Lift and drag forces are given by

$$L = \frac{1}{2} V^2 K_a C_l \qquad (8)$$

$$D = \frac{1}{2} V^2 K_a C_d \qquad (9)$$

where K_a is the reference area and C_l, C_d are the coefficients of lift and drag respectively and are approximated by

$$C_l = K_{cl} \sin\alpha \cos\alpha \qquad (10)$$

$$C_d = K_{cd_1} + K_{cd_2} \sin^2\alpha \qquad (11)$$

where K_{cl}, K_{cd_1}, K_{cd_2} are vehicle dependent constants for lift and drag and α is the angle of attack and control variable.

Using the above equations, the optimal control problem could be formulated using four state variables and one control variables:

$$\dot{x}_1 = h = -V\sin r = -x_2 \sin r = f_1 \qquad (12)$$

$$\dot{x}_2 = V = g\sin r - D/m = f_2 \qquad (13)$$

$$\dot{x}_3 = K_h \rho^{1/2} V^3 = K_h {}^{1/2} x_2^3 = f_3 \qquad (14)$$

$$\dot{x}_4 = (L^2 + D^2)/m^2 = f_4 \qquad (15)$$

$$J = x_3(t_f) + x_4(t_f) \qquad (16)$$

The goal would be to calculate the angle of attack, α, during the trajectory such that equation (16) was minimized. Using the minimum principle, the Hamiltonian could be formed

$$H = \lambda_1 f_1 + \lambda_2 f_2 + \lambda_3 f_3 + \lambda_4 f_4 \qquad (17)$$

To minimize J, α could be determined from

$$dH/dt = 0 \qquad (18)$$

along the trajectory. This sets up a two-point boundary value problem with eight state and costate equations and eight boundary conditions. This presents a difficult problem to solve.

3. Dynamic Programming Solution

By reducing the problem to one of two state variables and one control variable, a dynamic programming iterative solution can be obtained. The method used is similar to that presented originally by [1], in the solution of the aircraft minimum time to climb problem. The problem was also treated by others later [3]. Here the object was to minimize the time required to achieve a given final altitude and velocity from a given initial altitude and velocity. The state space was represented by a grid with velocity and altitude as the abscissa and ordinate, respectively. The problem becomes one of moving from an initial grid point to a final grid point moving in the direction of either constant V or constant h. The theory of dynamic programming shows that an optimal cost can be computed for each grid point by starting at the terminal point and applying the minimum cost control to get to each of the other grid points. The technique uses the priciple of optimality which says that the minimum cost is achived by moving in a direction that minimizes the sum of the cost of going to the next state plus the cost of going from the resulting state to the terminal state. The concept can be represented by the following equation:

$$J(x,t) = \underset{u^{(k)}}{\mathrm{Min}} \; l(x, u^{(k)}, t) \; t + J(x+\Delta x, t+\Delta t) \qquad (19)$$

where $l(x, u^{(k)}, t)$ is the cost per unit time when applying $u^{(k)}$ and $J(x,t)$ is the minimum cost at state x and time t. Once these cost and control values have been calculated for each grid point, the optimal trajectory can be recovered by starting at the inital point and traversing grid such that the cost is minimum for each step. A subtle advantage of this method is what Bellman calls implicit imbedding where an optimal trajectory can be determined from any initial point on the grid. Although the optimal re-entry problem does not try to minimize time, the same principles can still be applied. The state space can be thought of as a grid in the h-V plane. With known terminal conditions as a starting point, the cost and minimum control values can be calculated for each point on the grid by using the recurrence relations in equation (19).

The problem is reduced to state variables and one control variable as follows:

$$f_1 = \dot{x}_1 = h = -x_2 \sin r \qquad (20)$$

$$f_2 = \dot{x}_2 = V = g\sin r - D/m \qquad (21)$$

$$u = \alpha \leq 40^\circ. \qquad (22)$$

For the computer program solution the state equations wre discretized as follows:

$$\Delta x_1 = -x_2 \sin r \cdot \Delta t \qquad (23)$$

$$\Delta x_2 = (g\sin r - D/m)\Delta t \qquad (24)$$

with the discretize versions of the cost funtions as

$$\Delta B = K_h \rho^{1/2} V^3 \Delta t \qquad (25)$$

$$\Delta G = (L^2 + D^2) \cdot \Delta t/m^2 \qquad (26)$$

The admissible controls were discretized to 5° increments

$$u \in \{0^\circ, 5^\circ, 10^\circ, 15^\circ, 20^\circ, 25^\circ, 30^\circ, 35^\circ, 40^\circ\}.$$

The Δt used in the above equations can be determined using an approach by Larson (4,5) called state increment dynamic programming. This method uses fixed size increments of ΔV and Δh but the time variable varies depending upon the state and the control applied at that state. The time interval used is that interval which is required to move one increment in Δx for each admissable $u^{(k)}$. For each control, Δt is calculated as follows:

$$\Delta t^{(k)} = \Delta x_i \,/\, f_i(x, u^{(k)}, t) \qquad (27)$$

where Δx_i is the fixed increment for state variable x_i the fixed increment for state variable x_i and $f_i(\dot{x}, u^{(k)}, t)$ is the state equationfor x_i. The next state is then computed for each u and each x such that

$$x^{(k)} = x + \Delta x = x + f(x, u^{(k)}, t)\Delta t^{(k)}$$

where Δx comes from equations (23) and (24) and Δt comes from equation (27) and $u^{(k)}$ is the control used to obtain $\Delta t^{(k)}$ in equation (27). Using this method each state lies on a boundary of the fixed increment of the state variables where a method using a fixed Δt does not. The optimal cost and control is then calculated for each of these next states using the principle of optimality and equation (19). This process is continued until values are determined for all of the grid points. The grid can then be traversed by starting at the initial point and moving in increments of state so that the cost is minimum. The solution to computer program lends itself well to a structured computer program. The calculations involved can be broken down into separate subroutines or procedures and a main routine can be written to sequence through the procedures as required. A PASCAL program was written that sequence through the dynamical equations for the system but due to problems encountered, a complete that weighed the heating rate and acceleration forces while optimally varying the angle of attack was not finished. The biggest problem encountered was finding a value for the constant fligh path angle that allowed the trajectory to be in the right envelope. In actual flight, the flight path angle varies considerably during re-entry and actually goes negative relative to the angle shown in figure 1. This occurs when the spacecraft begins to enter more dense atmosphere at a high velocity. The effect decreases the rate of increase of the cost funtions at a time when they would otherwise increase rapidly due to high velocity in a more dense atmosphere. In order to overcome the constant flight path angle problem it was nessary to adjust a number of constants in the data base. The following were the desired initial and final conditions:

$$V(t_0) = 24000 \text{ ft/sec}$$
$$V(t_f) = 15000 \text{ ft/sec}$$
$$h(t_0) = 400000 \text{ ft}$$
$$h(t_f) = 250000 \text{ ft}$$

These numbers correspond roughly roughly with the figures used the early portion of the space shuttle re-entry[6]. The actual numbers used however correspond to a space craft weighing about 9600 lbs with a surface area of 66.5 ft^2. The numbers used came from the problem presented by[7] originally but were modified as stated earier.

The intent was to construct a grid based on the initial and final states and then proceed with the algorithm described. Using increments of ΔV of 1000ft/sec and Δh of 10000ft. The grid would be 9x15. These values were chosen in the interest of conserving computer budget and in actual trajectory design would be of smaller magnitude. These increments still cause one to deal with the curse of dimensionality. The admissible control was quantized into 5° increments meaning 8 possible values to cover the 40° range of the angle of attack. This means to calculate all the grid points would require on the order of 9x15x8 or 1080 times through the dynamical equations. This does compare favorably, however, with the alternative brute force method which would require on the order of $(9+15)!/(8!)(16!)$ or 700000 times through the equations to evaluate all possible routes. The run made used a constant angle of attack of 0° the first time through and 40° the the second time through. It was felt this would show the range of values that might be encountered during a trajectory where the angle of attack can be varied from 0° to 40°.

4. Conclusion

The optimal re-entry from orbit problem presenrs some very interesting problems to be solved. A dynamic programming solution is very well suited for the problem compared to a two-point boundary value problem with 8 state and costate equations using the minimum principle. The dynamic programming solution is also more attractive than a brute force grid-type search method. Even so, the problem presented here requires more work and computer time than an end of the project allows.

REFERENCES

1) Cartaino, T. F. and Dreyfuss, S. E., "Application of Dynamic Programming to Airplane Minimum time-to-Climb Problem," Aeronautical Engineering Review, June 1957.

2) Greensite, A. L., Element of Modern Control Theory, Spartan Books, New york, 1970

3) Bellman, R. E. and Dreyfuss, S. E. Applied Dynamic Programming, Princeton University Press, 1962.

4) Larson, R. E., "Dynamic Programming with Reduced Computaional Requirements," IEEE Trans. Auto. Contr., April, 1963.

5) Larson, R. E., State Increment Dynamic Programming, American Elsevier Publishing Co., New York, 1968

6) Duncan, R. C., Dynamic of Atmospheric Entry, McGraw-Hill, New York, 1962.

7) Garcia, F. and Fowler, W. T., "Thermal Protection System Weight Minimization for the Space Shuttle Through Trajectory Optimization," Journal of Spacecraft and Rockets, April, 1974.

Table 1. Constants for Trajectory Program

$r_e = 2.09 \times 10^7$ ft

$g = 32.2$ ft/sec^2

$m = 300.0$ lb-sec^2/ft

$K_h = 1.0 \times 10^{-4}$ $(lb)^{1/2}$sec

$K_{sl}^{1/2} = 0.052$ $(lb)^{1/2}$sec/ft^2

$K_e = -4.26 \times 10^{-5}$ ft^{-1}

$Ka = 66.5$ ft^2

$K_{cl} = 1.8$

$K_{cd1} = 0.274$

$K_{cd2} = 1.2$

$H(0) = h(t_0) = 400000$ ft

$V(0) = V(t_0) = 24000$ ft/sec

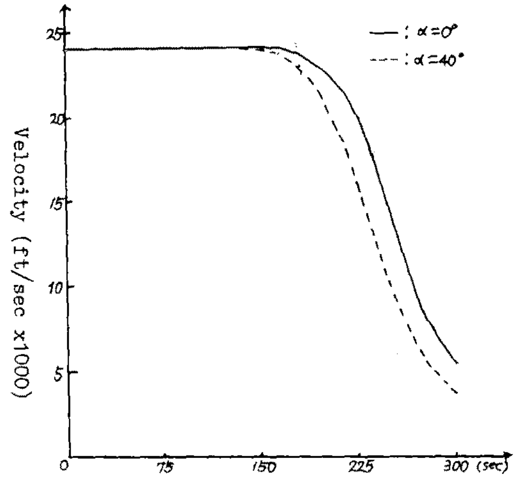

figure 2. Velocity of Spacecraft

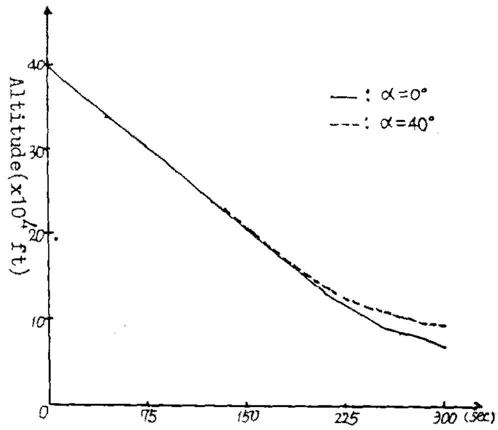

figure 3. Altitude of Spacecraft