Optimal Rocket Launch

Abstract

This work formulates a rocket launch as a two dimensional optimal control problem. The object is to maximize the final horizontal velocity subject to terminal state constraints. The steepest ascent approach is used for numerically solving this optimization. Through simulation in MATLAB, the optimal horizontal velocity was found to be 49862.781 ft/s and the optimal control input was observed to have near-linear behavior through time.

1 Introduction

Consider a rocket launch in which we seek to maximize the terminal horizontal velocity over time period $[t_0, t_f]$. The rocket is subject to a specified terminal height and zero terminal vertical velocity constraint. Utilizing a model for the rocket dynamics, this problem can be formulated into an optimal control problem.

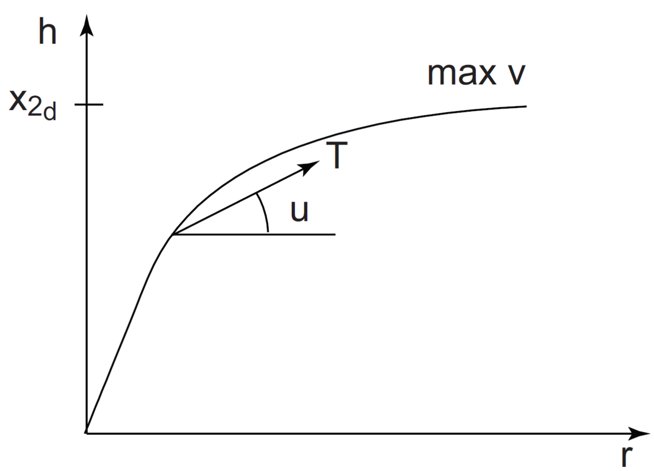


Figure 1: Rocket Launch Illustration

The vehicle dynamics are as follows:

$$\begin{aligned}
\dot{x}_1(t) &= \dot{r} = x_3(t) \\
\dot{x}_2(t) &= \dot{h} = x_4(t) \\
\dot{x}_3(t) &= \dot{v} = T\cos u(t) \\
\dot{x}_4(t) &= \dot{w} = T\sin u(t) - g
\end{aligned} \tag{1}$$

where $x_1(t)$ is the horizontal component of the rocket's position, $x_2(t)$ is the vertical component of the rocket's position, $x_3(t)$ is the horizontal component of velocity, and $x_4(t)$ is the vertical component of velocity, $u(t)$ is the angle modulating the rocket motor's thrust vector from horizontal, g is gravitational acceleration, and T is the constant specific thrust of the rocket motor.

The initial conditions at time $t = t_0$ are assumed to be the following:

$$\begin{aligned}
x_1(t_0) &= x_{10} \\
x_2(t_0) &= x_{20} \\
x_3(t_0) &= 0 \\
x_4(t_0) &= 0
\end{aligned} \tag{2}$$

The problem is to determine control input $u(\cdot)$ over interval $[t_0, t_f]$ to minimize

$$J(u(\cdot); x_0) = -x_3(t_f) \tag{3}$$

subject to

$$x_2(t_f) = x_{2d}, \quad x_4(t_f) = x_{4d} \tag{4}$$

where x_{2d} and x_{4d} are the desired terminal altitude and vertical velocity respectively.

In this project, we consider the case with the following constants:

$$x_{10} = 0 \text{ ft}, \quad x_{20} = 0 \text{ ft}, \quad x_{2d} = 320,000 \text{ ft}, \quad x_{4d} = 0 \text{ ft/s},$$
$$\Big[t_0, \quad t_f\Big] = \Big[0, \quad 900\Big]\, s, \quad g = 32 \text{ ft/s}^2, \quad T = 2g$$

2 Theory and Algorithm

2.1 General Analytical Solution

From *Theorem 4.3.1* [1] if there is an optimal control that minimizes the cost function then there exists a ν that satisfies the following .

$$H_u(x^o(t), u^o(t), \lambda(t), t) = 0 \quad \forall t \text{ in } [t_0, t_f], \tag{5}$$

$$-\dot{\lambda}^T(t) = H_x(x^o(t), u^o(t), \lambda(t), t), \tag{6}$$

$$\lambda^T(t_f) = \phi_x(x^o(t)) + \nu^T \psi_x(x^o(t)) \tag{7}$$

The Hamiltonian of the rocket problem is as follows:

$$H(x, u, \lambda, t) = \lambda_1 x_3 + \lambda_2 x_4 + \lambda_3 T \cos u + \lambda_4 T \sin u - \lambda_4 g \tag{8}$$

In the problem statement, functions $\phi(t_f)$ and $\psi(t_f)$ are defined as follows:

$$\begin{aligned} \phi(t_f) &= -x_3(t_f) \\ \psi(t_f) &= \begin{bmatrix} x_2(t_f) - x_{2d} \\ x_4(t_f) - x_{4d} \end{bmatrix} \end{aligned} \tag{9}$$

Using Equations (5) (6) (7), the general form for the optimal control of the rocket problem can be derived:

$$u^o(t) = \arctan[-\nu_4 - (t_f - t)\nu_2] \tag{10}$$

2.2 Steepest Ascent Method

To solve this rocket problem, the steepest ascent method is used. First, a nominal control u_N is assumed. The initial guess is assumed to have the same form as derived analytically in Equation (10) with $\nu_2 = 0.01$ and $\nu_4 = 0$.

With the assumed nominal control $u_N(t)$, the resulting path $x_N(t)$ can then be calculated. This calculation is done by numerical integration using MATLAB function `ode45`:

$$\dot{x}_N(t) = f(x_N(t), u_N(t), t) \Rightarrow x_N(t) \tag{11}$$

where f is the known dynamics model.

We consider a perturbation δu to the nominal control:

$$u_{N+1}(\cdot) = u_N(\cdot) + \delta u(\cdot) \tag{12}$$

where $\delta u = \epsilon \eta$.

The objective is to eventually use the perturbations to optimize the cost criterion and satisfy the terminal constraints. To do so, influence functions are formulated.
First, the influence function from the terminal constraints, $\lambda^{\psi}(t) \in \mathbb{R}^{n \times p}$:

$$\dot{\lambda}^{\psi}(t) = -f_x^T \lambda^{\psi}(t), \quad \lambda^{\psi}(t_f) = \psi_x^T(t_f) \tag{13}$$

The influence function from the cost criterion, $\lambda^{\phi}(t) \in \mathbb{R}^n$, is as follows:

$$\dot{\lambda}^{\phi}(t) = -f_x^T \lambda^{\phi}(t) - L_x^T, \quad \lambda^{\phi}(t_f) = \phi_x^T(t_f) \tag{14}$$

where $n = 4$ and $p = 2$ in the rocket problem. Following the rocket dynamics and problem constraints,

$$f_x = \begin{bmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}, \quad L_x = 0, \quad \phi_x = \begin{bmatrix} 0 & 0 & -1 & 0 \end{bmatrix}, \quad \psi_x = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

The influence functions are numerically integrated backward in time and their values are stored through time. However, because f_x is time in-variant for the rocket launch, the differential Equations (13) and (14) can also be solved analytically with a matrix exponential:

$$\lambda^{\psi}(t) = e^{-f_x^T(t-t_f)}\lambda^{\psi}(t_f) \tag{15}$$

$$\lambda^{\phi}(t) = e^{-f_x^T(t-t_f)}\lambda^{\phi}(t_f) \tag{16}$$

Next, the desired change in the terminal constraint $\delta\psi(t_f)$ and desired change in cost through choice of ϵ were chosen. For each iteration these were set to the following:

$$\delta\psi(t_f) = \frac{1}{5}\begin{bmatrix} x_2(t_f) - x_{2d} \\ x_4(t_f) - x_{4d} \end{bmatrix} \tag{17}$$

$$\epsilon = 0.01$$

The change in the terminal constraint, $\delta\psi(t_f)$ is specified to vary each iteration. This is to allow for larger changes in the beginning and smaller changes when nearing convergence while retaining linearity. The specifications (17), along with the stored values of the influence functions $\lambda^{\psi}(t)$ (15), $\lambda^{\phi}(t)$ (16), and partial derivative of the dynamics with respect to u, f_u, were then used to formulate Lagrange multiplier ν:

$$\begin{aligned} \nu = &-\left[\int_{t_0}^{t_f} \lambda^{\psi T}(t) f_u(x_N(t), u_N(t), t) f_u^T(x_N(t), u_N(t), t)\lambda^{\psi}(t)dt\right]^{-1} \\ &\times \left[\psi_x(x_N(t_f))\delta x(t_f)/\epsilon + \int_{t_0}^{t_f} \lambda^{\psi T}(t) f_u(x_N(t), u_N(t), t) L_u^T(x_N(t), u_N(t), t)dt \right. \\ &\left. + \int_{t_0}^{t_f} \lambda^{\psi T}(t) f_u(x_N(t), u_N(t), t) f_u^T(x_N(t), u_N(t), t)\lambda^{\phi}(t)dt\right] \end{aligned} \tag{18}$$

where $f_u(x_N(t), u_N(t), t)$ for the rocket problem is defined as follows:

$$f_u(u_N(t), t) = \begin{bmatrix} 0 \\ 0 \\ -T\sin(u_N(t)) \\ T\cos(u_N(t)) \end{bmatrix}$$

The Lagrange multiplier ν is then used to calculate the control perturbation δu:

$$\delta u(t) = -\epsilon \left[\left(\lambda^{\phi T}(t) + \nu^T \lambda^{\psi T}(t) \right) f_u(x_N(t), u_N(t), t) + L_u(x_N(t), u_N(t), t) \right]^T \qquad (19)$$

The control perturbation δu is then used to calculate a new nominal control u_{N+1} following equation (12) and the process is repeated until the measured change in the objective function and terminal constraints are very small i.e. $\delta\phi(t_f) \to 0$ and $\delta\psi(t_f) \to 0$.

3 Results and Performance

Applying methods described in Section 2, a solution was found for the optimal control through time. The optimal control input is shown in Figure 2 and the rocket states which satisfy the terminal state constraints are shown in Figure 3. The optimal control input looks near-linear in time with a regular decrease in radians for the motor angle.

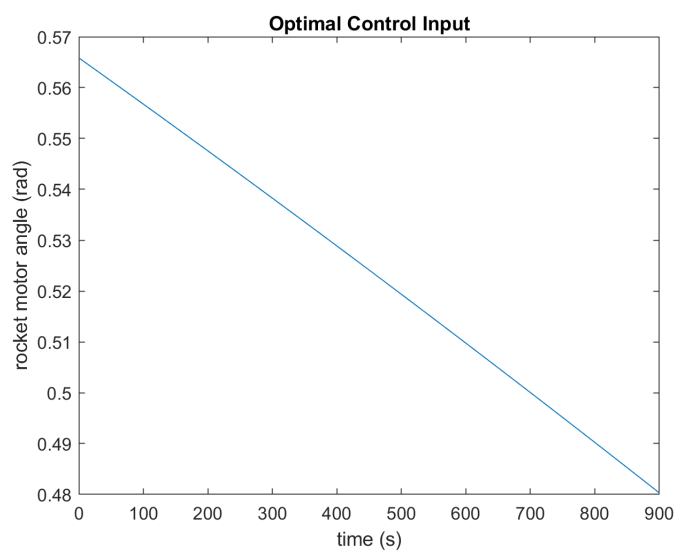
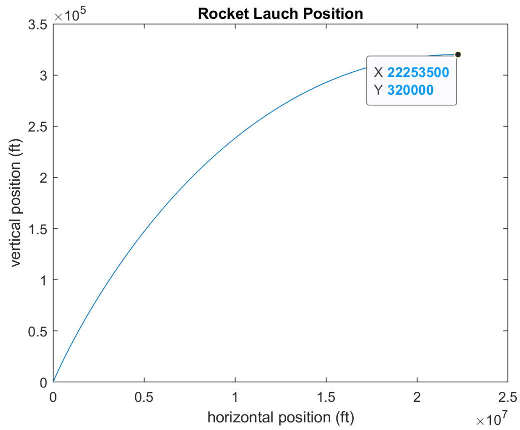
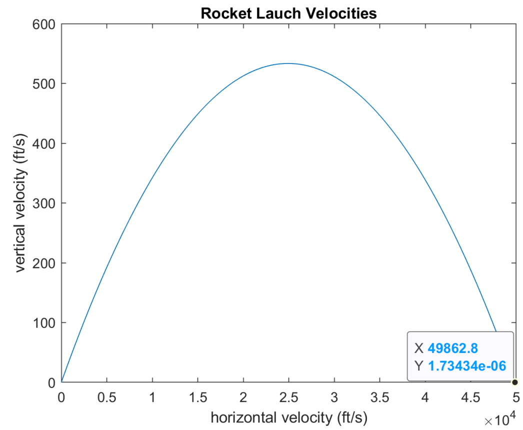


Figure 2: Optimal Control $u^o(t)$



(a) Rocket Position



(b) Rocket Velocity

Figure 3: Rocket State Plots $x^o(t)$

With initial guess of the nominal control being $u_N(t) = \arctan[-(t_f - t) \cdot 0.01]$, the program took 222 iterations to converge to the results shown in Figure 3. The stopping criterion was chosen to be $\delta J \leq 10^{-4}$ and $\delta\psi \leq 10^{-4}$. The constrained terminal conditions are met with $x_2(t_f) = 320000$ ft and $x_4(t_f) = 1.734e{-}0.5$ ft/s (approximately zero).

The change in the cost and terminal constraints are shown in Figures 4 and 5. The specified, and actual values for the change in terminal constraint, $\delta\psi(t_f)$ converge to zero, indicating the terminal constraints being met. Additionally, the change in the objective function, δJ, converge to zero, indicating convergence to an optimal solution.

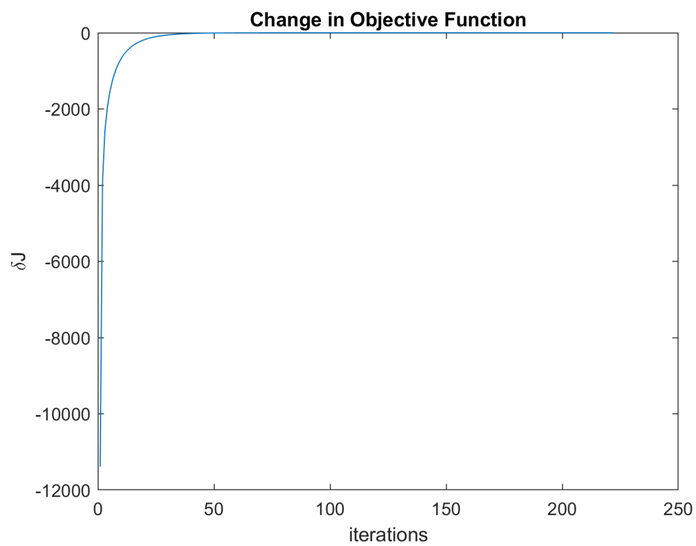


Figure 4: Change in Objective Function δJ

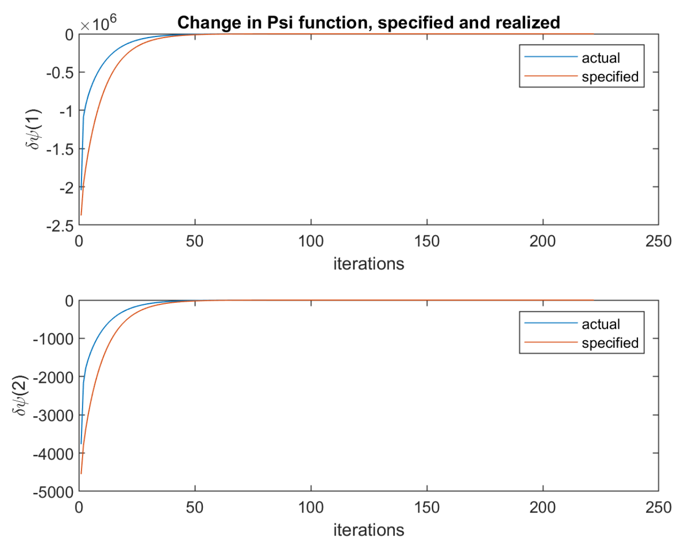


Figure 5: Change in Terminal Constraints $\delta\psi$

The optimal value for the Lagrange multiplier was found to be $\nu = [-0.0001, -0.5210]'$ and the optimal terminal horizontal velocity was 49862.781 ft/s.

4 Conclusion

This project maximizes the horizontal velocity of a rocket while satisfying both terminal constraints in height and vertical velocity. The problem was formulated as an optimal control problem in which laws of weak first-order optimality were used to derive the optimal form of the control input, providing a sufficient first guess of the nominal control, $u_N(t)$. Then, the steepest ascent method was used to converge to an optimal solution. The optimal input was observed to be near-linear and the optimal horizontal velocity, $x_3(t)$ at time t_f was found to be 49862.781 ft/s. Overall, the project provided a practical example of an optimal control problem, which connected to real world applications such as the landing of the lunar module.

References

[1] J. L. Speyer and D. H. Jacobson, Primer on Optimal Control Theory. 2010.

A Code

```
%Helene Levy
%MAE 270C Project
%Optimal Control
clc; clear; close all;

%given parameters
g = 32; %ft/s^2
T = 2*g;
h = 320000; %ft
tf = 900; %s
t0 = 0;

%dimension sizes
p = 2; n = 4;

%partial of dynamics wrt x
f_x = [0 0 1 0; 0 0 0 1; 0 0 0 0; 0 0 0 0];

%initial state
x10 = 0;
x20 = 0;
X_0 = [x10, x20, 0, 0]';

%final conditions
x2d = h;
x4d = 0; %need to check this with speyer
X_f = [0 x2d 0 x4d]';

%assumed intial nu values
nu2 = -0.01;
```

```
nu4 = 0;

% time discretization
dt = 0.5;
t_span= t0:dt:tf;

%% Initial Control Guess
%initial control guess
u_N = atan(-nu4-(tf-t_span)*nu2);
figure;
plot(t_span,u_N);
xlabel('time (s)');
ylabel('angle (rad)');
title('Initial Control Input Guess');

num_it = 500;

∆Phi_actual = zeros(1,num_it);
delPhi = 100; %so while loop runs
∆Psi_actual = zeros(p,num_it);
delPsi = [100; 100]; %so while loop runs
∆Psi_specified = zeros(p,num_it);

it = 1;
troubleshoot = false;
while it ≤ num_it && ((abs(delPhi) > 10^(-4)) || (abs(delPsi(1)) > ...
    10^(-4)) || (abs(delPsi(2)) > 10^(-4)))
    %% State Dynamics Propagation
    %looking at u values to see if modulating
    if troubleshoot
        disp(u_N(1:10));
    end
    %numerically integrating dynamics
```

```
    [t,X] = ode45(@(t,X) dynamics_prop(t,X,u_N,t_span),t_span, X_0);
    t = t';
    X = X';

    %% Lambda Calculation (Exponent Method)
    %size allocations
    lam_psi = zeros(n,p,length(t_span));
    lam_phi = zeros(n,length(t_span));

    %partials of psi and phi
    psi_x_tf = [0 1 0 0; 0 0 0 1];
    phi_x_tf = [0 0 -1 0];

    %terminal constraints
    lam_psi(:,:,end) =  psi_x_tf';
    lam_phi(:,end) =  phi_x_tf';

    for i = 1:length(t_span)-1
        %matrix exponential
        lam_phi(:,i) = expm(-f_x'.*(t_span(i)-tf))*lam_phi(:,end);
        lam_psi(:,:,i) = expm(-f_x'.*(t_span(i)-tf))*lam_psi(:,:,end);
    end

    %% Storing actual ΔPsi and ΔPhi
    if it > 1
        ΔPhi_actual(it-1) = (-X(3,end))-(-X_old(3,end));
        delPhi = ΔPhi_actual(it-1); %for stopping while loop
        ΔPsi_actual(:,it-1) = [X(2,end)-x2d ;X(4,end)-x4d]-Psi_old;
        delPsi = ΔPsi_actual(:,it-1); %for stopping while loop
        ΔPsi_specified(:,it-1) = ΔPsi;
    end

    %% ΔPsi and epsilon
```

```
    epsilon = 0.01;
    ΔPsi = -[X(2,end)-x2d ;X(4,end)-x4d]/5;

    %size allocations
    term1 = zeros(p,p,length(t_span));
    term3 = zeros(p,length(t_span));

    %dynamics partial wrt u
    f_u = ...
        [zeros(1,length(t_span));zeros(1,length(t_span));-T*sin(u_N);T*cos(u_N)];

    for i = 1:length(t_span)
        term1(:,:,i) = lam_psi(:,:,i)'*f_u(:,i)*f_u(:,i)'*lam_psi(:,:,i);
        term3(:,i)= lam_psi(:,:,i)'*f_u(:,i)*f_u(:,i)'*lam_phi(:,i);
    end
    term2 = ΔPsi/epsilon;
    nu = -inv(trapz(term1,3))*(term2+trapz(term3,2));

    %% computing new nominal control
    Δ_u = zeros(1,length(t_span));
    for i = 1:length(t_span)
        Δ_u(i) = -epsilon*((lam_phi(:,i)'+nu'*lam_psi(:,:,i)')*f_u(:,i))';
    end
    u_N = u_N + Δ_u;

    it = it+1;
    X_old = X;
    Psi_old = [X(2,end)-x2d ;X(4,end)-x4d];
end
%% Plotting and Printing Results

fprintf('Number of iterations: %d \n',it)
fprintf('nu: \n');
```

```
disp(nu)
fprintf('Optimal Terminal Horizontal Velocity, %4.3f ft/s\n',X(3,end))

figure;
plot(X(1,:),X(2,:));
ylabel('vertical position (ft)');
xlabel('horizontal position (ft)');
title('Rocket Lauch Position');

figure;
plot(X(3,:),X(4,:));
ylabel('vertical velocity (ft/s)');
xlabel('horizontal velocity (ft/s)');
title('Rocket Lauch Velocities');

figure;
plot(t_span,u_N(:));
ylabel('rocket motor angle (rad)');
xlabel('time (s)');
title('Optimal Control Input');

figure;
subplot(2,1,1);
plot(1:it,ΔPsi_actual(1,1:it));
hold on;
plot(1:it,ΔPsi_specified(1,1:it));
ylabel('\Δ\psi(1)');
xlabel('iterations');
legend('actual','specified');
title('Change in Psi function, specified and realized');

subplot(2,1,2);
plot(1:it,ΔPsi_actual(2,1:it));
```

```
hold on;
plot(1:it,ΔPsi_specified(2,1:it));
ylabel('\Δ\psi(2)');
xlabel('iterations');
legend('actual','specified');

figure;
plot(1:it,ΔPhi_actual(1,1:it));
ylabel('\ΔJ');
xlabel('iterations');
title('Change in Objective Function');

%% Dynamic Propagation function
function dXdt = dynamics_prop(t,X,u_N,t_span)
    g = 32; %ft/s
    T = 2*g;
    u = interp1(t_span,u_N,t);

    dXdt = zeros(4,1);
    dXdt(1) = X(3);
    dXdt(2) = X(4);
    dXdt(3) = T*cos(u);
    dXdt(4) = T*sin(u)-g;
end
```