Lab 1 Report

Rick Yuan

January 17, 2020

Step 1: system dynamics equations The system dynamics for the two states

$$\dot{x}_1 = x_2 \tag{1}$$

$$\dot{x}_2 = -G * m_1 * m_2/r^2 + F_{\text{thrust}} \tag{2}$$

Step 2: force of gravity at surface

$$G_0 = -G * m_1 * m_2/r^2 = -2.9995 * 10^3 \tag{3}$$

Step 3: derivative of gravity

$$\delta F_g/\delta x = 2 * G * m_1 * m_2/(r_a + x)^2 \tag{4}$$

At X = 0:

$$\delta F_g/\delta x = 2 * G * m_1 * m_2/(r_a)^2 = 5.9990 \tag{5}$$

Step 4: linear model of gravity Linear model of gravity

$$F_{\text{Glinear}} = -2.9995 * 10^3 + 5.9990 * x_1 \tag{6}$$

2.022% error at x=80
13.60% error at x=200
100% error at x=500

Step 5: linear state space system

$$\begin{aligned}
\dot{X} &= FX + BU \\
Y &= HX \\
F &= \begin{pmatrix} 0 & 1 \\ 0.006 & 0 \end{pmatrix} \\
B &= \begin{pmatrix} 0 & 0 \\ 0.001 & 0.001 \end{pmatrix} \\
H &= \begin{pmatrix} 1 \\ 0 \end{pmatrix}
\end{aligned} \tag{7}$$

Step 6: discretized state space system at 0.1 seconds

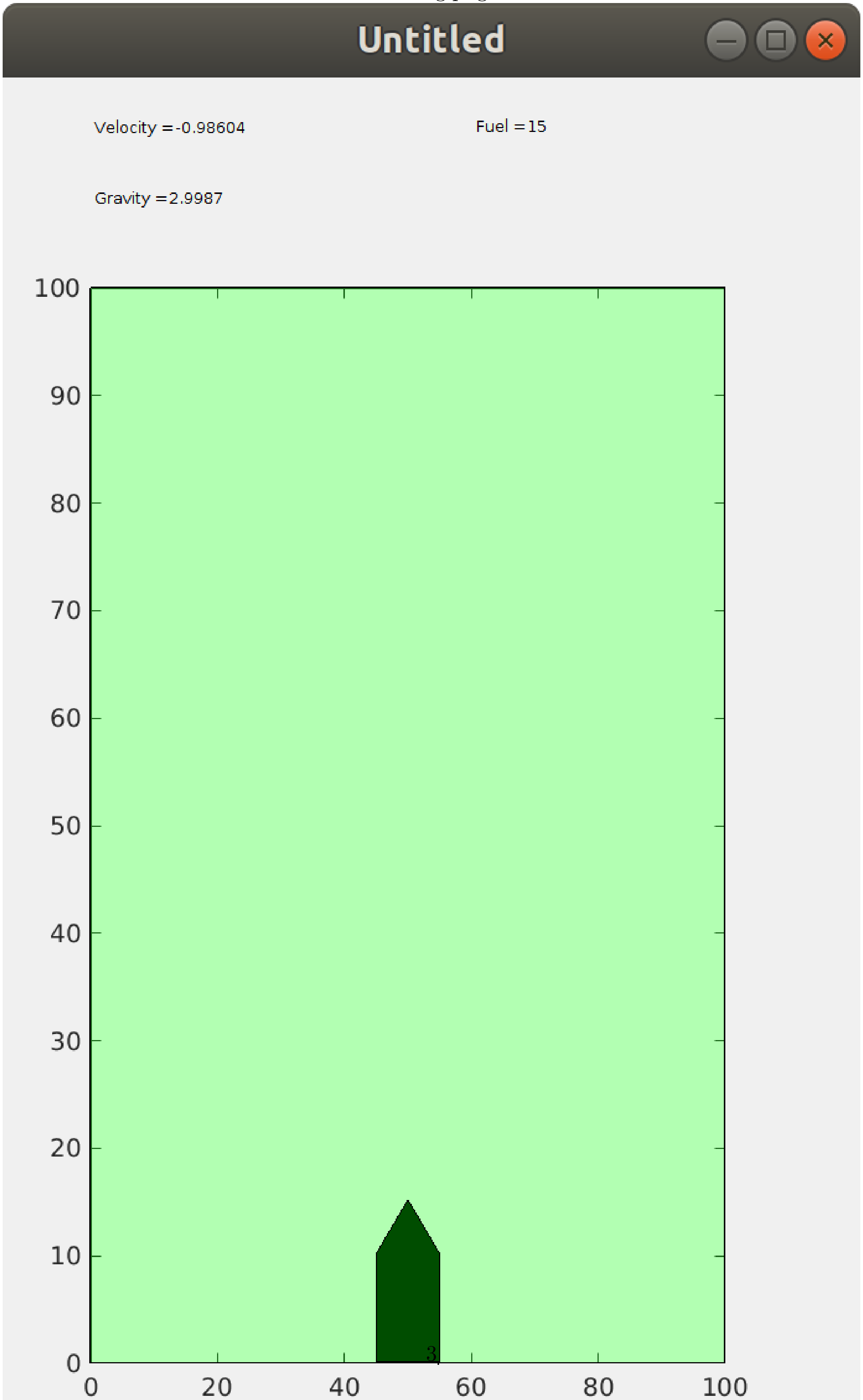
$$\Phi = \begin{pmatrix} 1 & 0.1 \\ 0.0006 & 1 \end{pmatrix}$$
$$B = 10^-3 * \begin{pmatrix} 0.005 & 0.005 \\ 0.1 & 0.1 \end{pmatrix} \tag{8}$$

Step 7: Matlab code to compute next state

```
function x_out = SpaceX_student_function(x_in,thrust)
%UNTITLED2 Summary of this function goes here
%   Detailed explanation goes here
u_step = [thrust;evalin('base','F_grav_0')];

x_out = evalin('base' ,'system_d.A')*x_in + evalin('base','system_d.B')*u_step;
x_out = evalin('base' ,'system_d.A')*x_in + evalin('base','System.B')*0.1*u_step;

end
```

Step 8: successful landing

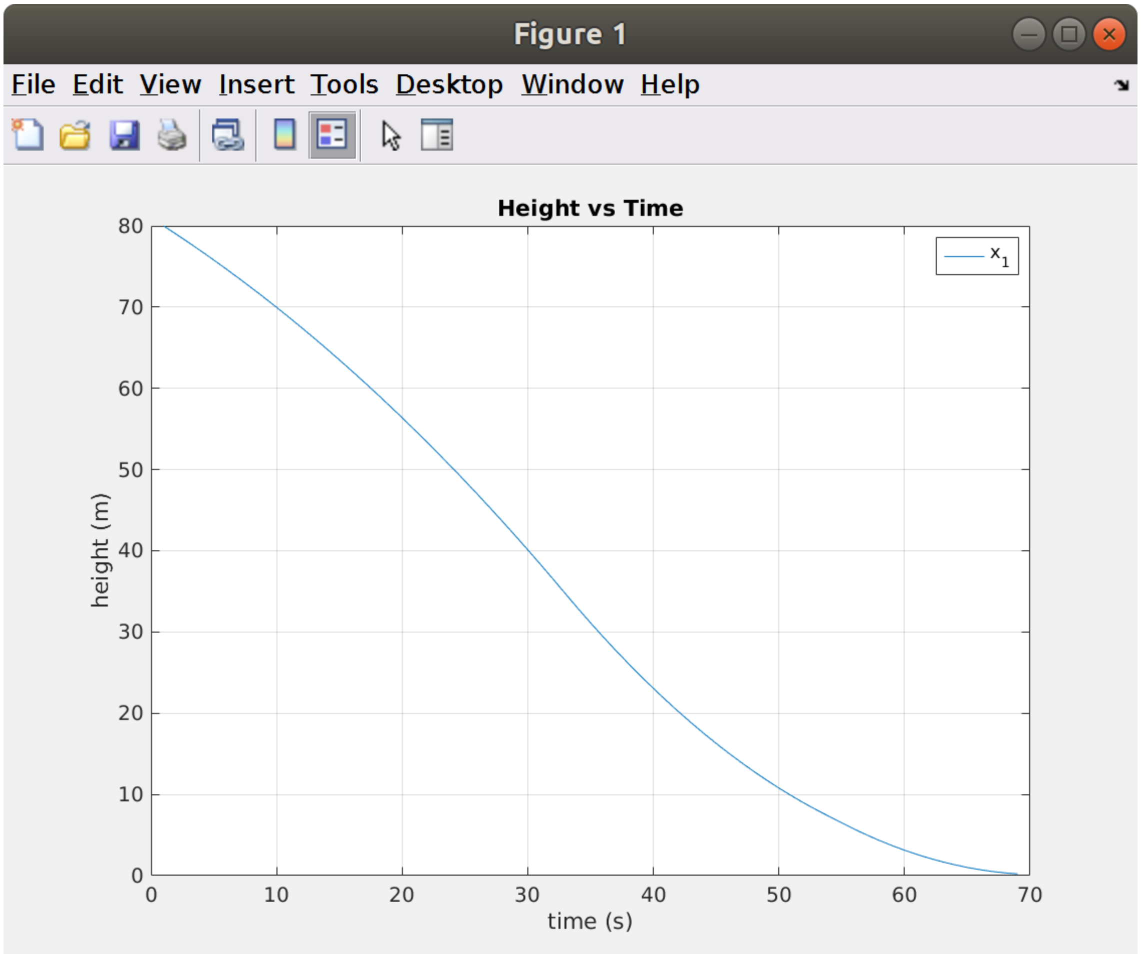
landing.png


Untitled
Velocity =-0.98604
Fuel =15
Gravity =2.9987
100
90
80
70
60
50
40
30
20
10
0
0
20
40
60
80
100

Step 9:

```
%% Lab 1 - Space X Rocket - SpaceX_student_script.m
% Name EENG 765

clear; close all; clc
%% Load data from SpaceX game

Project_1
Thrust_data = load('SpaceX_Winning_Inputs.mat');
Thrust_data = Thrust_data.inputs_saved;
%% Set up variables for simulating the data

dt = 0.1 % [seconds]
tt=0:dt:(length(Thrust_data)*dt)-dt; % Time vector [seconds]
%% Initilize the output vector (Velocity and Position)

Vel_0 =-10; %[m/s]
Pos_0 = 80; %[m]

x = zeros(2,length(tt));
y = zeros(1,length(tt));
x(:,1) = [Pos_0;Vel_0];
%% Loop through the data to get the output vector:
y(1)=H*x(:,1)
for i=2:(length(tt)+1)
    x(:,i)=SpaceX_student_function(x(:,i-1),Thrust_data(i-1));
    y(i) = H*x(:,i);

end
%% Report Final Velocity

disp(sprintf('The Rockets final Velocity was %.4f m/s',x(2,end)));
%% Plot time vs. position
    figure(1)
    plot(x(1,:))
    grid on
    title('Height vs Time')
    legend('x_1')
    xlabel('time (s)')
    ylabel('height (m)')

    figure(2)
    plot(y)
    grid on
    title('Height vs Time (Output)')
    legend('Y')
    xlabel('time (s)')
    ylabel('height (m)')

%% Plot Time vs. Velocity
    figure(3)
    plot(x(2,:))
    title('Velocity vs Time')
    legend('x_2')
    xlabel('time (s)')
    ylabel('velocity (m/s)')

%% Plot Temperature
```

Step 10: plotting



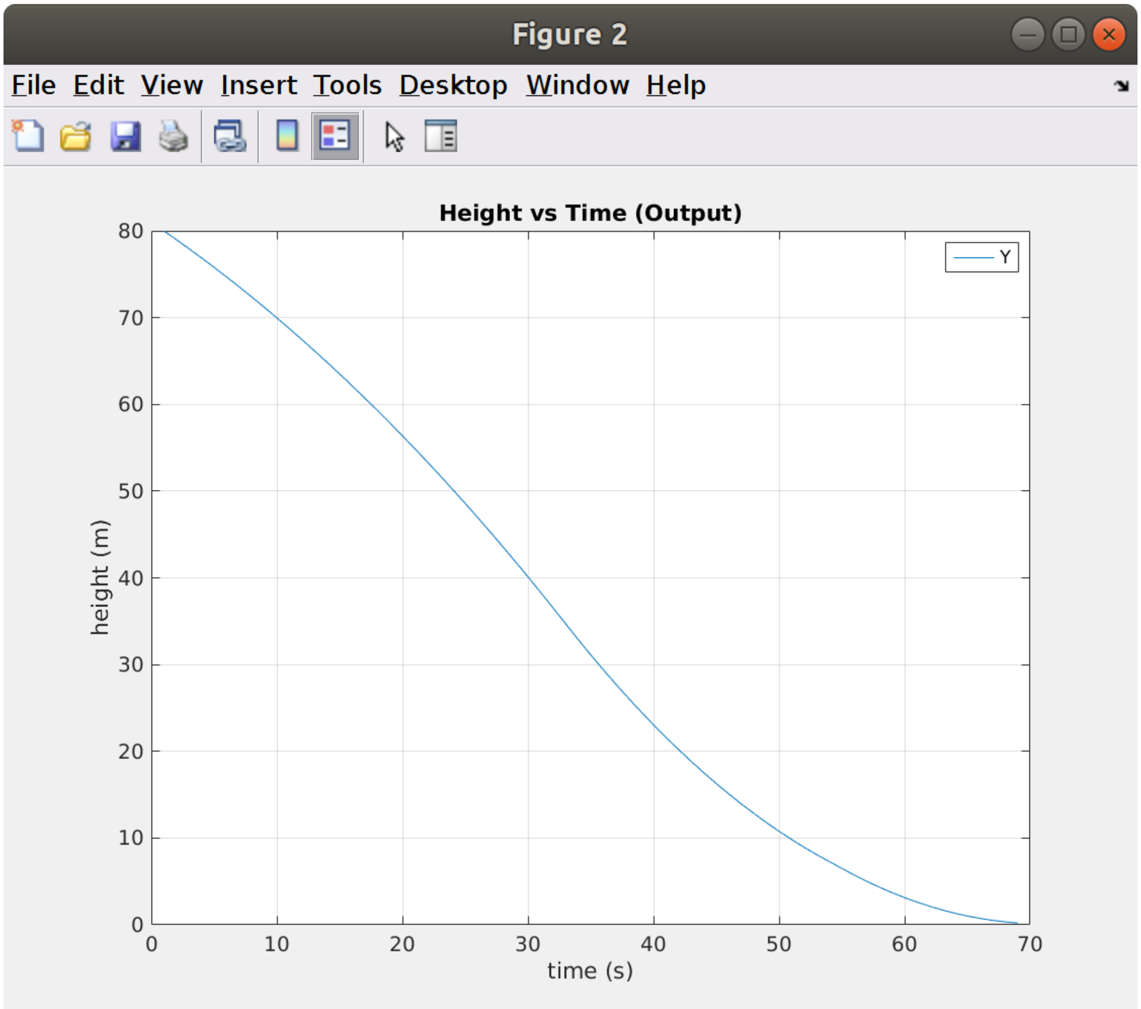
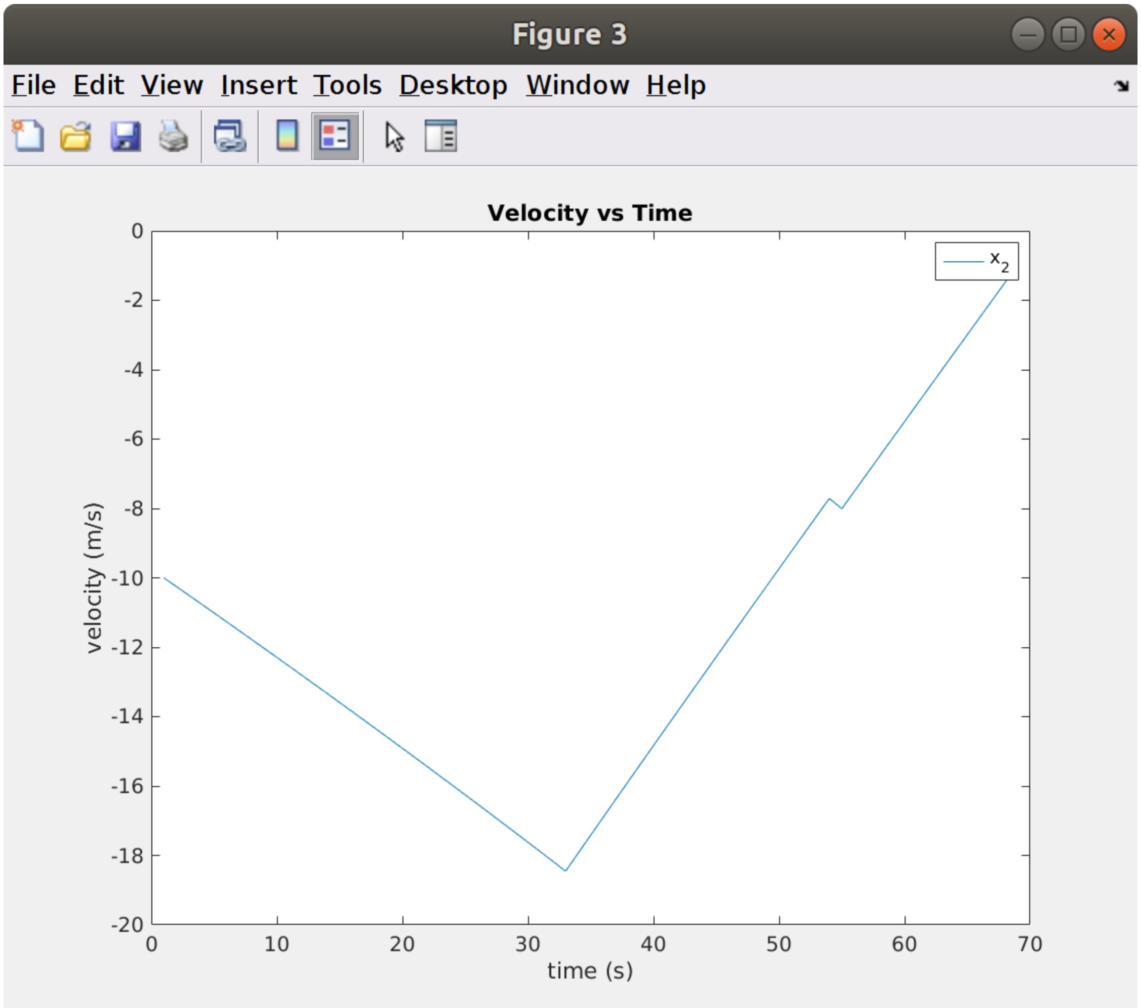


Figure 2
File Edit View Insert Tools Desktop Window Help
Height vs Time (Output)
Y
height (m)
time (s)
0
10
20
30
40
50
60
70
80




```
The output will always be equal to the state x_1 because it is simply multiplied
    by 1 to get the output.

Step 11:
Rerunning the script with the discrete B matrix approximated as B*0.1, the new
    final velocity of the rocket is -0.9800 m/s. This represents a 0.6 percent
    error in the final velocity
```

Other Matlab Code:
Project1.m

```
clear
clc
clf

%%
% Step 1:
%
% x1dot = x2
%
% x2dot = -G*m1*m2/r^2 + F_thrust
%%
% Step 2:
```

```
G = 6.67 *10^-11;
r_a = 1000;
m1 = 1000;
m2 = 4.497*10^16;

F_grav_0 = -G*m1*m2/r_a^2
%%
% Step 3:
%
% f_grav = -G*m1*m2/(r_a+x)^2
%
% d_f_grav/d_x = 2 * G*m1*m2/(r_a + x)^3

d_f_grav = 2 * G*m1*m2/(r_a)^3
%%
% Step 4:

syms x
f_grav = -G*m1*m2/(r_a+x)^2;
f_grav_lin = F_grav_0 + d_f_grav*x;

x = 0
RealGravity = double(subs (f_grav))
LinGravity =double(subs(f_grav_lin))
PercentError=100 * (RealGravity - LinGravity)/(RealGravity)

x = 80
RealGravity = double(subs (f_grav))
LinGravity =double(subs(f_grav_lin))
PercentError=100 * (RealGravity - LinGravity)/(RealGravity)

x = 200
RealGravity = double(subs (f_grav))
LinGravity =double(subs(f_grav_lin))
PercentError=100 * (RealGravity - LinGravity)/(RealGravity)

x = 500
RealGravity = double(subs (f_grav))
LinGravity =double(subs(f_grav_lin))
PercentError=100 * (RealGravity - LinGravity)/(RealGravity)
%%
% Step 5:

F = [0 1;d_f_grav/m1 0]
%2 states for input u1 = thrust u2 = gravity
B = [0 0;1/m1 1/m1]

H= [1 0]

dt= 0.1;

System = ss(F,B,H,0)
system_d = c2d(System,dt)

system_d.A
system_d.B
%%
% Step 7:
```

```
SpaceX_student_function([80;-10],8000)
```