Siddula, Saimanoj

Homework 6

Gravity Turn Rocket

Contents

[Introduction 2](#_Toc444851903)

[Analysis 3](#_Toc444851904)

[Starting Constants 3](#_Toc444851905)

[Orbit Requirements 3](#_Toc444851906)

[Gravity Turn 4](#_Toc444851907)

[Appendix 10](#_Toc444851908)

[Main Code 10](#_Toc444851909)

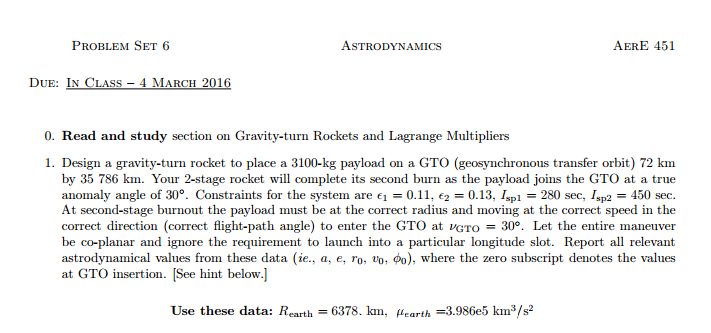
[Gravityassist Function 12](#_Toc444851910)

# Introduction

What is the gravity turn rocket?

The Gravity Turn, sometimes referred to as the zero-lift turn, is a technique used to minimize the amount of fuel needed to get into orbit around Earth or the large body of mass by causing a turn in the rocket and focusing it to orient horizontally while elevating. This decreases the amount of weight the rocket must overcome and makes use of more fuel to use in increasing the vertical distance.

The problem statement is as shown below:



# Analysis

## Starting Constants

Before even solving anything the gives were named:

Mu\_Earth = 3.986012\*10^5

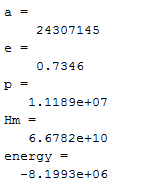
Gravity = 9.81 m/s^2

Radius of the Earth =6378.145 km

After this step before actually solving the problem, in order to avoid unit conflicts I converted all kilometers to meters, because we will be using Newton for Force, which is kg\*m/s^2. To save the confusion between using degrees or radians, I will be using only radians and converting all givens to radians as well.

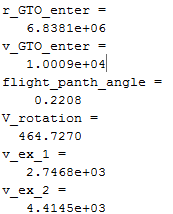
The first step in solving this problem is finding the values of the injection point, which is done using the following equations:

After plugging in these values the results are:



## Orbit Requirements

Once the values for the orbit have been calculated, they were used to determine the required velocity, flight path angle, and the radius to enter the orbit with a true anomaly angle of 30°.



One important thing to decide before solving for any of these is deciding where to launch the rocket from. This is very important because this will decide how much of the Earth’s rotational velocity we will use to our advantage. Assuming the Earth to be spherical, angular velocity of the earth is 7.292E-5(rad/s)

This velocity however is based at the Equator, which as we discovered through past homework will be the maximum amount. We also came up with the equation VEarth\_Rotation, shown above, to help find the rotational velocity at a given Latitude. Since we were given full freedom in choosing where we can launch from, I used braeig.us to find the latitudes of multiple space centers and selected the place closest to the Equator, which ended up being Alcantara, Brazil with a Latitude of 2.3 degrees.

## Gravity Turn

The idea behind the gravity turn rocket as mentioned in the introduction is that the rocket will gain altitude and the velocity rotates along the center of mass of the rocket. This helps the rocket adjust its flight path angle mid-flight. The derived equations to solve for the Distance(X), the height (H), the velocity (V) and the angle (new) is as follows:

The equations for drag and Thrust are:

I ended up creating a function that solves for the above derivatives with respect to time and ran the ODE45 in the main program using this function to solve for all the required values and graphs.

Once I finished this program, the last task came, which was to play with the mass values at the first and second stage and the burn rate at the first and second stage until the optimal rocket mass ratio and the burn rates were found. I first started by using the Isp as a reference in order to go about the problem, estimating the total mass to be around 350,000kg and using the ratio to start somewhere. From there I kept moving to the left and right while keeping the total mass constant the same to notice patterns. The better results were when the first stage and second stage were decently close together but the first stage had a significantly larger value.

The end results were as follows:

Burnrate1: -1400 kg/s

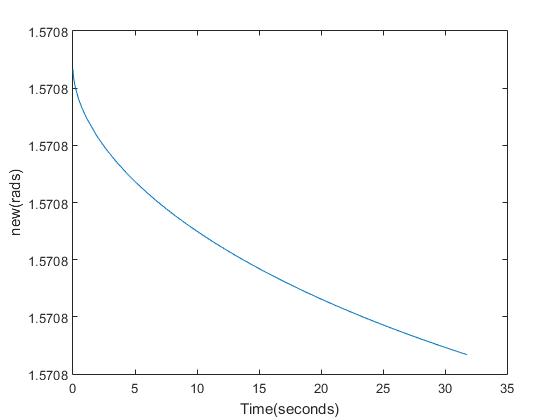
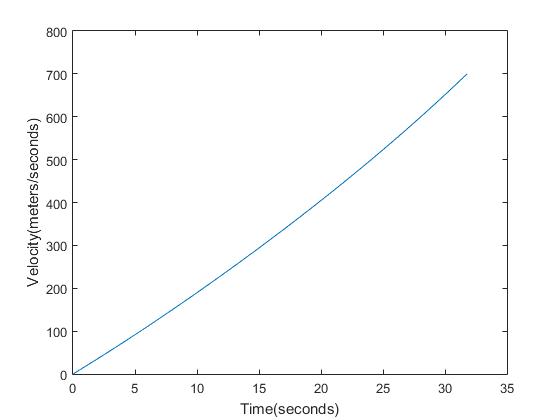
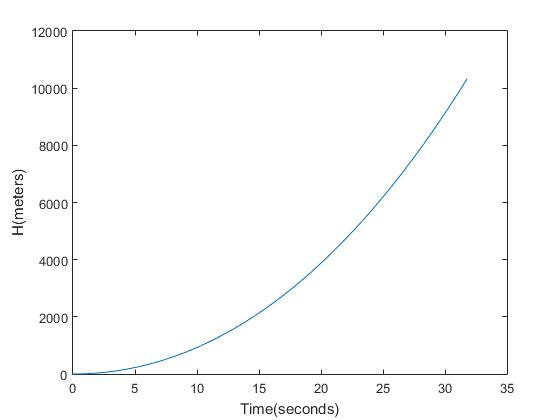
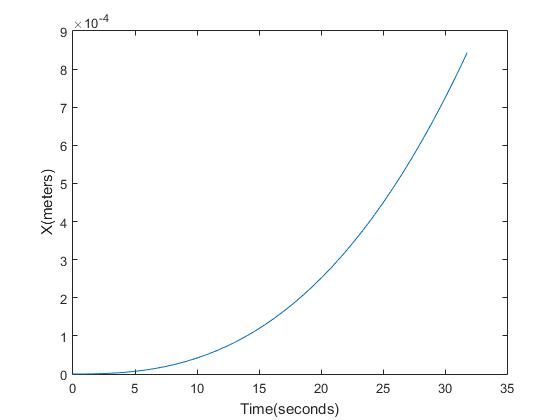
Burnrate2: -200 kg/s

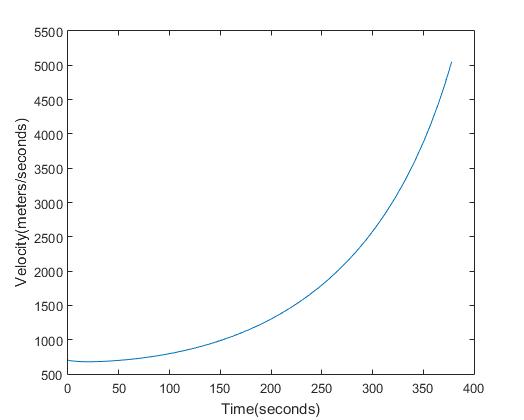
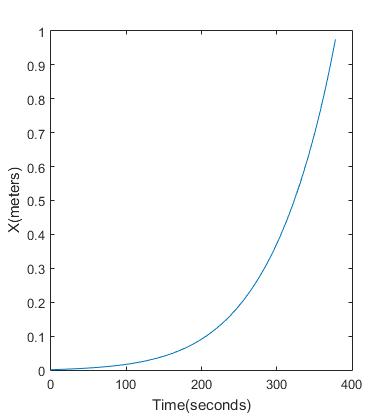
M0 = 90000 kg

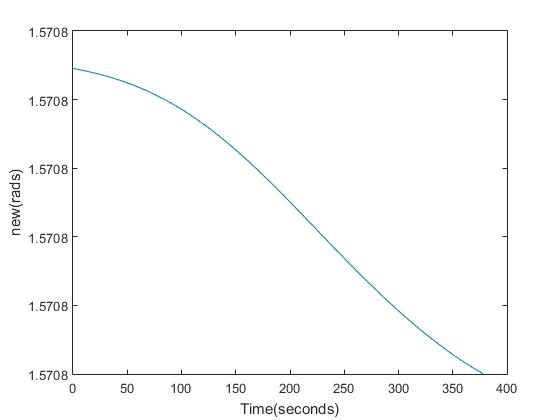
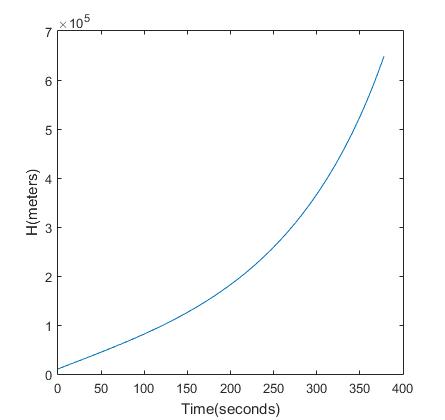
M1 = 140000 kg

## Graphs

Stage1







# Appendix

## Main Code

%Saimanoj Siddula

%AerE 451 Astrodynamics II

%Homrwork 6

clear,clc

meter = 1000;

gram = 1000;

%Defining all constraints

mu = 3.986012\*(10^5)\*meter^3; %m^3/s^2

g = 9.81; %m/s^2

R\_e = 6378.145\*meter; %Radius of earth in meters

mass\_payload\_final = 3100; %kg

epsilon\_1 = .11;

epsilon\_2 = .13;

Isp\_1 = 280; %sec

Isp\_2 = 450; %sec

r\_GTO\_p = 72\*meter+R\_e;%m

r\_GTO\_a = 35786\*meter+R\_e;%m

nu\_enter = 30/180\*pi; %rad

angular\_velocity = 7.292115856\*10^-5;%angular velocity of the earth at equator in rad/sec

%Defining Aerodynamic drag constraints

lamda = 8000; %m

rho\_0 = 1.2;%kg/m^3

C\_d = .155;

A\_front = 20;%m

%Calculating a,e,p, and Hm

a = (r\_GTO\_p + r\_GTO\_a)/2; %semi-major axis(m)

e = 1-(r\_GTO\_p/a);%eccentricity

p = a\*(1-e^2); %semi-latus parameter (m)

Hm = sqrt(p\*mu); %anguar momentum (m^4/s^2)

%Calculating energy of the GTO

energy = -mu/(2\*a);

%Calculating radius and velocity of the GTO entry point

r\_GTO\_enter = p/(1+e\*cos(nu\_enter)) %radius at nu = 12 deg

v\_GTO\_enter = sqrt(2\*(energy+(mu/r\_GTO\_enter))) %

flight\_panth\_angle = acos(Hm/(r\_GTO\_enter\*v\_GTO\_enter)) %flight path angle in rad

%29.5630 N lat for Johnson Space Center, 5.2372 N lat for Guiana Space Center

V\_rotation = angular\_velocity \*R\_e\*cosd(2.3)

%Solving for the exhaust velocity at both stages.

v\_ex\_1 = Isp\_1\*g %Exhaust Velocity at Stage 1 [m/s]

v\_ex\_2 = Isp\_2\*g %Exhaust Velocity at Stage 2 [m/s]

%Solving for the burn times and thrusts.

%burn rates

mass\_total\_2 = mass\_payload\_final; %total mass @stage 2 kg

mass\_total\_1 = 90000; %total mass @stage 1 kg

mass\_total\_0 = 140000; %total mass @ initial stage kg

beta\_1 = -1400; %kg/s

beta\_2 = -200; %kg/s

time\_burn\_0 = 0;

time\_burn\_1 = ((mass\_total\_1-mass\_total\_0)\*(1-epsilon\_1))/(beta\_1); %sec

time\_burn\_2 = ((mass\_total\_2-mass\_total\_1)\*(1-epsilon\_2))/(beta\_2); %sec

thrust\_1 = -v\_ex\_1\*beta\_1; %N

thrust\_2 = -v\_ex\_2\*beta\_2; %N

%setting up for the ODE solver

tburn1 = linspace(0,time\_burn\_1,100);

massburn1 = beta\_1\*tburn1+mass\_total\_0;

tburn2 = linspace(0,time\_burn\_2,100);

massburn2 = beta\_2\*tburn2+mass\_total\_1;

%[t,y] = ode45(odefun,tspan,y0,options);

[T,Y] = ode45(@(t,y) solfunction2(t,y,beta\_1,tburn1,v\_ex\_1,massburn1),[0 time\_burn\_1],[0,0,0.001,pi\*.999999999955/2]);

figure(1)

plot(T,Y(:,1))

xlabel('Time(seconds)')

ylabel('X(meters)')

figure(2)

plot(T,Y(:,2))

xlabel('Time(seconds)')

ylabel('H(meters)')

figure(3)

plot(T,Y(:,3))

xlabel('Time(seconds)')

ylabel('Velocity(meters/seconds)')

figure(4)

plot(T,Y(:,4))

xlabel('Time(seconds)')

ylabel('new(rads)')

fin = length(Y);

[T,Y] = ode45(@(t,y) solfunction2(t,y,beta\_2,tburn2,v\_ex\_2,massburn2),[0 time\_burn\_2],[Y(fin,1),Y(fin,2),Y(fin,3),Y(fin,4)]);

figure(5)

plot(T,Y(:,1))

xlabel('Time(seconds)')

ylabel('X(meters)')

figure(6)

plot(T,Y(:,2))

xlabel('Time(seconds)')

ylabel('H(meters)')

figure(7)

plot(T,Y(:,3))

xlabel('Time(seconds)')

ylabel('Velocity(meters/seconds)')

figure(8)

plot(T,Y(:,4))

xlabel('Time(seconds)')

ylabel('new(rads)')

## Gravityassist Function

function [ dy ] = solfunction2( t,y,b,time,v,m )

%UNTITLED Summary of this function goes here

% Detailed explanation goes here

lamda = 8000; %m

rho\_0 = 1.2;%kg/m^3

C\_d = .155;

A\_front = 20;%m

g = 9.81;

R\_e = 6378.145\*1000;

T = -v\*b;

m = interp1(time,m,t);

D = .5\*C\_d\*A\_front\*(rho\_0\*exp(-y(2)/lamda))\*y(3)^2;

dy = zeros(4,1); %resetting the function dy values

dy(1) = y(3)\*cos(y(4));

dy(2) = y(3)\*sin(y(4));

dy(3) = (T-D-(m\*g-((m\*dy(1)^2)/(R\_e+y(2)))\*sin(y(4))))/m;

dy(4) = (-(m\*g-((m\*dy(1)^2)/(R\_e+y(2))))\*cos(y(4)))/(m\*y(3));

end