HW3 - Sage Herrin

1. Consider the equations of motion for a unit mass subjected to an inverse square law force field, e.g. a satellite orbiting a planet,

$$\ddot{r} = r\dot{\theta}^2 - \frac{k}{r^2} + u_1(t) \tag{1}$$

$$\ddot{\theta} = -\frac{2\dot{\theta}\dot{r}}{r} + \frac{1}{r}u_2(t) \tag{2}$$

where r represents the radius from the center of the force field, θgives the angle with respect to a reference direction in the orbital plane, k is a constant, and u_1 and u_2 represent radial and tangential thrusts, respectively. It is easily shown that for the initial conditions $r(0) = r_0$, $\theta(0) = 0$,$\dot{r}(0) = 0$, and $\dot{\theta}(0) = \omega_0$ with nominal thrusts $u_1(t) = 0$ and $u_2(t) = 0$ for all $t \geq 0$, the equations of motion have as a solution the circular orbit given by

$$r(t) = r_0 = \text{constant} \tag{3}$$

$$\dot{\theta}(t) = \omega_0 = \text{constant} = \sqrt{\frac{k}{r_0^3}}, \tag{4}$$

$$\theta(t) = \omega_0 t + \text{constant} \tag{5}$$

(a) Pick a state vector for this system, and express the original nonlinear ODEs in 'standard' nonlinear state space form.

(b) Linearize this system's nominal equations of motion about the nominal solution $r(t) = r_0$, $\dot{r}(0) = 0$, $\theta(t) = \omega_0 t + \text{constant}$ and $\dot{\theta}(t) = \omega_0$ with $u_1(t) = 0$ and $u_2(t) = 0$. Find (A, B, C, D) matrices for output $y(t) = [r(t), \theta(t)]^T$ for the linearized system of equations about the nominal solution.

(c) Convert the continuous time (A, B, C, D) matrices you found from part (b) into discrete time (F, G, H, M) matrices, using a discretization step size of $\Delta t = 10s$ and setting $k = 398600\ km^3/s^2$ and $r_0 = 6678\ km$.

(d) Interpret the results for the STM in part (c), i.e. what is the physical meaning of each column vector that makes up F?

a.) State vector $X = \begin{bmatrix} r \\ \dot{r} \\ \theta \\ \dot{\theta} \end{bmatrix}$

$$\Rightarrow \boxed{\dot{X} = F(X, u, t) = \begin{bmatrix} \dot{r} \\ r\dot{\theta}^2 - k/r^2 + u_1 \\ \dot{\theta} \\ \frac{-2\dot{\theta}\dot{r}}{r} + \frac{1}{r}u_2 \end{bmatrix}}$$

b.) to linearize system about nominal solutions

=> need Jacobian

$$\Rightarrow \left[\frac{\partial F}{\partial x}\right] = \begin{bmatrix} \frac{\partial f_1}{\partial r} & \frac{\partial f_1}{\partial \dot r} & \frac{\partial f_1}{\partial \theta} & \frac{\partial f_1}{\partial \dot\theta} \\ \frac{\partial f_2}{\partial r} & \frac{\partial f_2}{\partial \dot r} & \frac{\partial f_2}{\partial \theta} & \frac{\partial f_2}{\partial \dot\theta} \\ \frac{\partial f_3}{\partial r} & \frac{\partial f_3}{\partial \dot r} & \frac{\partial f_3}{\partial \theta} & \frac{\partial f_3}{\partial \dot\theta} \\ \frac{\partial f_4}{\partial r} & \frac{\partial f_4}{\partial \dot r} & \frac{\partial f_4}{\partial \theta} & \frac{\partial f_4}{\partial \dot\theta} \end{bmatrix}$$

$$\Rightarrow \begin{bmatrix} 0 & 1 & 0 & 0 \\ \dot\theta^2 + \frac{2K}{r^3} & 0 & 0 & 2r\dot\theta \\ 0 & 0 & 0 & 1 \\ \frac{2\dot\theta\dot r}{r^2} - \frac{u_2}{r^2} & \frac{-2\dot\theta}{r} & 0 & \frac{-2\dot r}{r} \end{bmatrix}$$

$$\& \quad \frac{\partial F}{\partial u} = \begin{bmatrix} \frac{\partial f_1}{\partial u_1} & \frac{\partial f_1}{\partial u_2} \\ \frac{\partial f_2}{\partial u_1} & \frac{\partial f_2}{\partial u_2} \\ \frac{\partial f_3}{\partial u_1} & \frac{\partial f_3}{\partial u_2} \end{bmatrix} = \begin{bmatrix} 0 & 0 \\ 1 & 0 \\ 0 & 0 \end{bmatrix}$$

$$\begin{bmatrix} \frac{\partial f_4}{\partial u_1} & \frac{\partial f_4}{\partial u_4} \end{bmatrix} \begin{bmatrix} 0 & \frac{1}{r} \end{bmatrix}$$

Using given nominal points, $r(0)=r_0$, $\theta(0)=0$, $\dot{r}(0)=0$, & $\dot{\theta}(0)=\omega_0$

$u_1(t)=0$ & $u_2(t)=0$

$$\Rightarrow \boxed{\tilde{A} = \begin{bmatrix} 0 & 1 & 0 & 0 \\ \omega_0^2 + \frac{2k}{r_0^3} & 0 & 0 & 2r_0\omega_0 \\ 0 & 0 & 0 & 1 \\ 0 & \frac{-2\omega_0}{r_0} & 0 & 0 \end{bmatrix}}$$

$$\boxed{\tilde{B} = \begin{bmatrix} 0 & 0 \\ 1 & 0 \\ 0 & 0 \\ 0 & 1/r_0 \end{bmatrix}}$$

$$\Rightarrow \delta\dot{x}(t) = \tilde{A}\,\delta x + \tilde{B}\,\delta u$$

for output $y = [r(t), \theta(t)]^T$

$$\Rightarrow \boxed{\tilde{C} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix}, D = \begin{bmatrix} 0 & 0 \\ 0 & 0 \end{bmatrix}}$$

c.) Convert continuous time ABCD matrices from part b into discrete time F G H M matrices using discretization step $\Delta t = 10s$ & $K = 398{,}600 \frac{km^3}{s^2}$ & $r_0 = 6{,}678$ km

using "ss" & "c2d" functions in matlab as follows

```
sysc = ss(Ã, B̃, C̃, D̃);
sysd = c2d(sysc, 10, 'zoh') - to ignore HOTs
[F,G,H,M] = ssdata(sysd);
```

these lines of code result in the following

F, G, H, & M matrices

$$\Rightarrow \quad F = \begin{bmatrix} 1.0002 & 9.9998 & 0 & 772.5749 \\ 0 & 0.9999 & 0 & 154.5133 \\ 0 & 0 & 1 & 9.9991 \\ 0 & 0 & 0 & 0.9997 \end{bmatrix}$$

$$G = \begin{bmatrix} 49.9994 & 0.3856 \\ 9.9998 & 0.1157 \\ -0.0001 & 0.0025 \\ 0.0000 & 0.0015 \end{bmatrix}$$

$$H = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix}$$

$$\boxed{M = \begin{bmatrix} 0 & 0 \\ 0 & 0 \end{bmatrix}}$$

d.) Each column vector in F represents the coefficients of one of the variables in each of the 4 EOMs

i.e $\begin{bmatrix} \vdots & \vdots & \vdots \\ & F & \\ \vdots & \vdots & \vdots \end{bmatrix}\begin{bmatrix} r \\ \dot{r} \\ \theta \\ \dot{\theta} \end{bmatrix} \Rightarrow \begin{array}{cccc} r & \dot{r} & \theta & \dot{\theta} \\ \end{array}\begin{bmatrix} \vdots & \vdots & \vdots & \vdots \\ & F & & \\ \vdots & \vdots & \vdots & \vdots \end{bmatrix}$

for e.g the first column vector represents all the r coefficients in the 4 EOMs of the system

2.) Simon 1.17

Dynamics of DC motor can be described as

$J\ddot{\theta} + F\dot{\theta} = T$, θ = angular position, J = moment of inertia, F is the coefficient of visc. friction, & T is torque applied to motor

a.) Generate 2-state linear system eqn. for motor in form of $\dot{x} = Ax + Bu$

$\Rightarrow J\ddot{\theta} + F\dot{\theta} = T \Rightarrow J\ddot{\theta} = T - F\dot{\theta} \Rightarrow \ddot{\theta} = \frac{T}{J} - \frac{F\dot{\theta}}{J}$

J

$\Rightarrow \vec{x} \{\theta \; \dot{\theta}\}^T \Rightarrow \dot{x} = Ax + Bu$

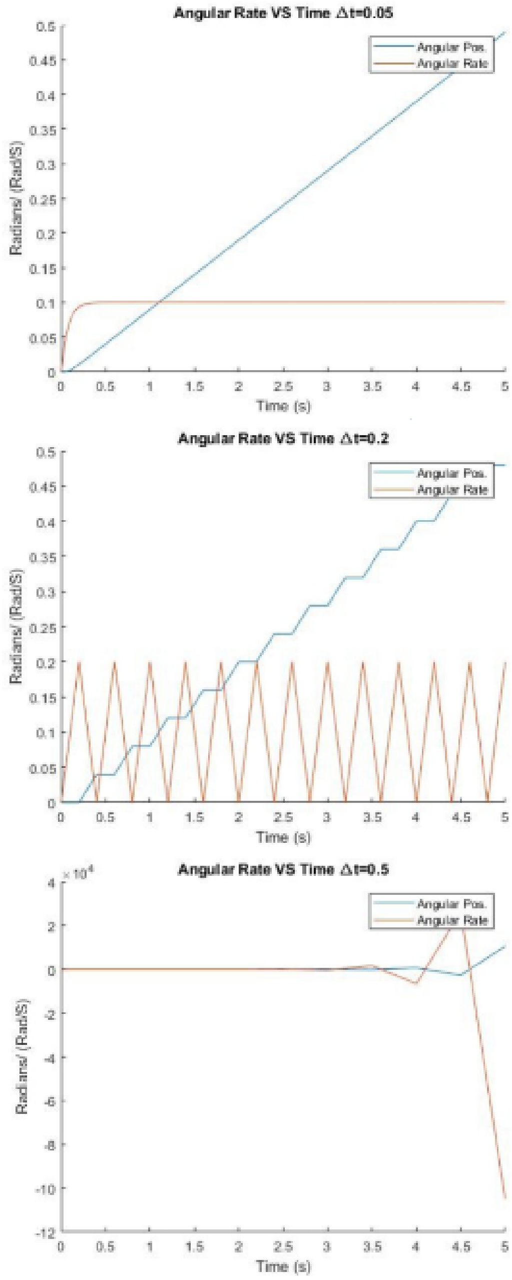
$$\Rightarrow \dot{x} = \ddot{\theta} = [\dot{\theta} \; \ddot{\theta}]^T \Rightarrow \dot{x} = \begin{bmatrix} 0 & 1 \\ 0 & -F/J \end{bmatrix}$$

T/J portion of EOM

$$\Rightarrow Bu = B \cdot T \Rightarrow B = \begin{bmatrix} 0 \\ 1/J \end{bmatrix} \cdot T$$

$$\Rightarrow \boxed{\dot{x} = Ax + Bu \Rightarrow \begin{bmatrix} 0 & 1 \\ 0 & -\frac{F}{J} \end{bmatrix} \begin{bmatrix} \theta \\ \dot{\theta} \end{bmatrix} + \begin{bmatrix} 0 \\ 1/J \end{bmatrix} \cdot T}$$

b.) Simulate system for 5 s & plot angular position & velocity. Use $J = 10 \text{ kg m}^2$, $F = 100 \text{ kg } \frac{m^2}{s}$, $x(0) = [0 \; 0]^T$, & $T = 10$ Nm. Use rectangular integration with step size 0.05 s, 0.2 s, & 0.5 s, comment on changes, & determine A matrix eigenvalues & relate their magnitudes to required step size for correct simulation



It seems that, given the response of the system, for the eigenvalues of A (found to be 0 & -10 using MATLAB 'eig' function) taking 1 over the absolute value of the eigenvalue revealed the max timestep allowable for accurate simulation. In this instance that timestep = 0.1 seconds. Anything greater than that resulted in the system behaving inaccurately, as shown when Δt was set to 0.2 & 0.5 seconds

3.) Vertical dimension of a hovering rocket can be modeled eqs

$$\dot{x}_1 = x_2 \quad , \quad \dot{x}_2 = \frac{Ku - gx_2}{x_3} - \frac{GM}{(R+x_1)^2} \quad , \quad \dot{x}_3 = -u$$

x_1 is vertical position of rocket, x_2 is vertical velocity, x_3 is mass of the rocket, u is control input, $K=1{,}000$ is thrust const. of proportionality, $g=50$ is drag const., $G=6.673E-11 \frac{m^3}{kg/s^2}$ is universal grav constant, $M=5.98E24$ kg is mass of Earth, & $R=6.37E6$ m is radius of Earth.

a.) find $u(t)=u_0(t)$ s.t system is in equilibrium @ $x_1(t)=0$ & $x_2(t)=0$

$\Rightarrow \dot{x}_1=0$ & $\dot{x}_2=0$

$\Rightarrow$ according to EOMs, $\dot{x}_1=0$ if $x_2=0$ ✓

$$\dot{x}_2 = \frac{Ku - gx_2}{x_3} - \frac{GM}{(R+x_1)^2} = 0 \Rightarrow \frac{Ku}{x_3} - \frac{GM}{R^2} = 0$$

$\Rightarrow$ $\boxed{\frac{GMx_3}{R^2K} = u_0 \text{ for equilibrium}}$

b.) Find $x_3(t)$ when $x_1(t)=0$ & $x_2(t)=0$

$\Rightarrow$ equilibrium expression $u_0 = \frac{GMx_3}{R^2K}$ when

$\Rightarrow \dot{x}_3 = -u \Rightarrow \dot{x}_3 = \frac{-GMx_3}{R^2K} \Rightarrow$ integrate to solve for $x_3(t)$

$$\Rightarrow \int \dot{x}_3 dt = \int \frac{-GmX_3}{R^2 K} dt \Rightarrow \int \frac{\dot{x}_3}{x_3} dt = \int \frac{-Gmx_3}{R^2 K}$$

$$\Rightarrow \int \frac{dx_3 t}{x_3} = \int \frac{-Gm}{R^2 K} dt \Rightarrow \ln(x_3(t)) + C = \frac{-Gmt}{R^2 K}$$

$$\Rightarrow \boxed{x_3(t) = Ce^{\frac{-Gmt}{R^2 K}}} \quad x_{3,0}$$

$$\Rightarrow u_0(t) = \frac{GmCe^{\frac{-Gmt}{R^2 K}}}{R^2 K}$$

c.) linearize system around state trajectory found above

$$\Rightarrow \delta \dot{x} = \hat{A}\delta x + \bar{B}\delta u$$

$$\Rightarrow \hat{A} \Rightarrow \left.\frac{\partial F}{\partial x}\right|_{x_{nom}}$$

$$Gm(R+x)^{-2}$$

$$\Rightarrow -2Gm(R+x_1)^{-3}$$

$$\Rightarrow \begin{bmatrix} \frac{\partial f_1}{\partial x_1} & \frac{\partial f_1}{\partial x_2} & \frac{\partial f_1}{\partial x_3} \\ \frac{\partial f_2}{\partial x_1} & \frac{\partial f_2}{\partial x_2} & \frac{\partial f_2}{\partial x_3} \\ \frac{\partial f_3}{\partial x_1} & \frac{\partial f_3}{\partial x_3} & \frac{\partial f_3}{\partial x_3} \end{bmatrix}$$

$$\begin{bmatrix} 0 & 1 & 0 \end{bmatrix}$$

$$\Rightarrow \begin{bmatrix} \frac{-2GM}{(R+x_1)^3} & \frac{-g}{x_3} & \frac{-(Ku-gx_2)}{x_3^2} \\ 0 & 0 & 0 \end{bmatrix}$$

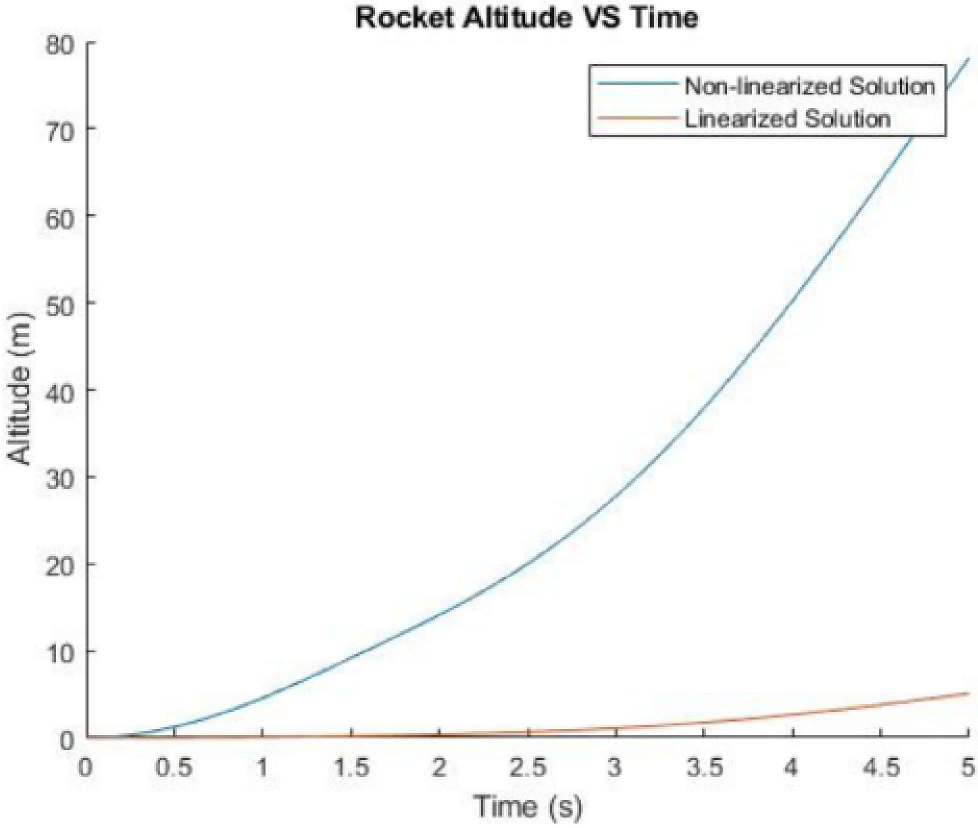
$$\Rightarrow \boxed{\hat{A} = \begin{bmatrix} 0 & 1 & 0 \\ \frac{-2GM}{R^3} & \frac{-g}{x_{3,0}} & \frac{-Ku}{x_{3,0}^2} \\ 0 & 0 & 0 \end{bmatrix}}$$

$$\boxed{\hat{B} = \frac{\partial F}{\partial u} = \begin{bmatrix} 0 \\ K/x_3 \\ -1 \end{bmatrix} \Rightarrow \begin{bmatrix} 0 \\ K/x_{3,0} \\ -1 \end{bmatrix} \delta u}$$

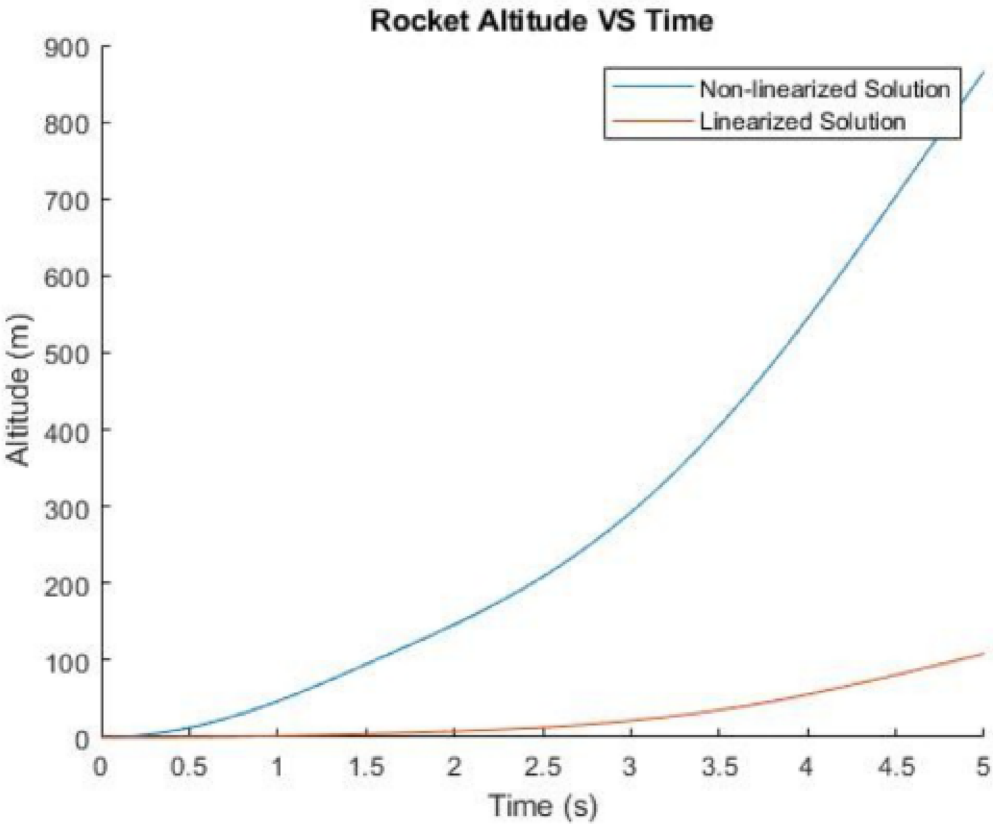
$$\Rightarrow \delta \dot{x} = \hat{A}\,\delta x + \hat{B}\,\delta u$$

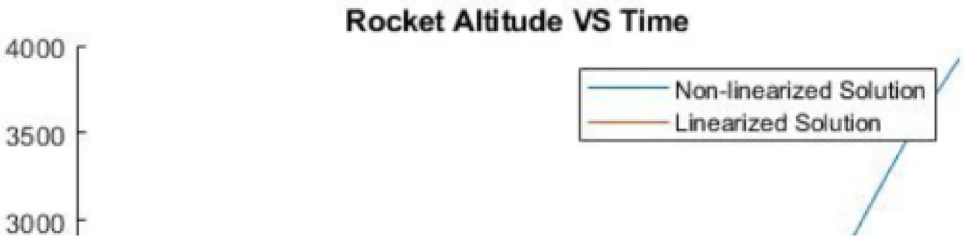
d.) Simulate non-lin. system for 5 seconds & linearized system for 5 seconds with $u(t) = u_0 t + \Delta u \cos(t)$.
Plot altitude of rocket for nonlinear sim, & linear sim. (on same plot) when $\Delta u = 10$. Repeat for $\Delta u = 100$ & $\Delta u = 300$. What can you conclude about accuracy of linearization.
$x_{3,0} = 1{,}000$ Kg & $u(t) = u_0(t) + \Delta u \cdot \text{abs}(\cos(t))$

$\Delta u = 10$

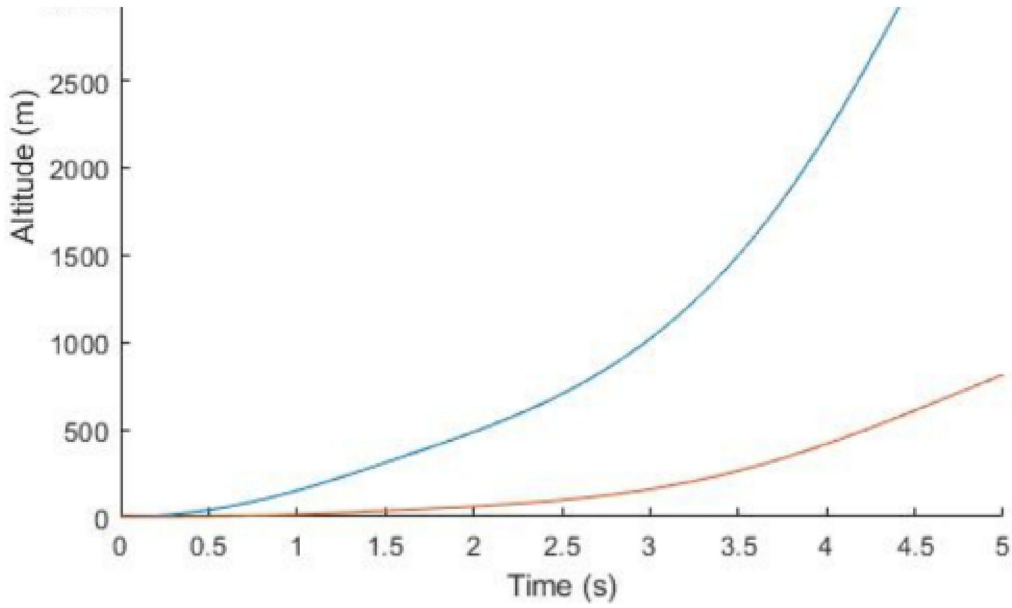


$\Delta u = 100$





$\Delta u = 300$



Based on the above plots, the accuracy of the linearization is relatively low given the magnitude of the deviation between the 2 curves for all three Δu values used in the simulations,

4.) Numerically compute the continuous time STM $\Phi(t,t_0)$ for problem 1 using the differential eqn. for $\Phi(t,t_0)$. How do these values compare to F after 10 seconds? Can you relate the discrete & continuous results at t=100 seconds?

=> after solving for Φ numerically in Matlab using ode45 & $\dot{\Phi} = A\Phi$

=> after 10 seconds, $\Phi = \begin{bmatrix} 1.002 & 9.9998 & 0 & 722.5741 \\ 0 & 0.9999 & 0 & 154.5133 \\ 0 & 0 & 1 & 9.9991 \\ 0 & 0 & 0 & 0.9997 \end{bmatrix}$

after 100 seconds, $\Phi = \begin{bmatrix} 0.0001 & 0.0100 & 0 & 7.7122 \\ 0 & 0.0001 & 0 & 0.1542 \\ 0 & 0 & 0.0001 & 0.0099 \\ 0 & 0 & 0 & 0.0001 \end{bmatrix} \cdot 10^4$

so @ $t=10$ seconds, $\Phi(t, t_0)$ is approximately equal to F, & @ $t=100$ seconds, the STM $\approx F^{10} = \Phi(10,0)^{10}$

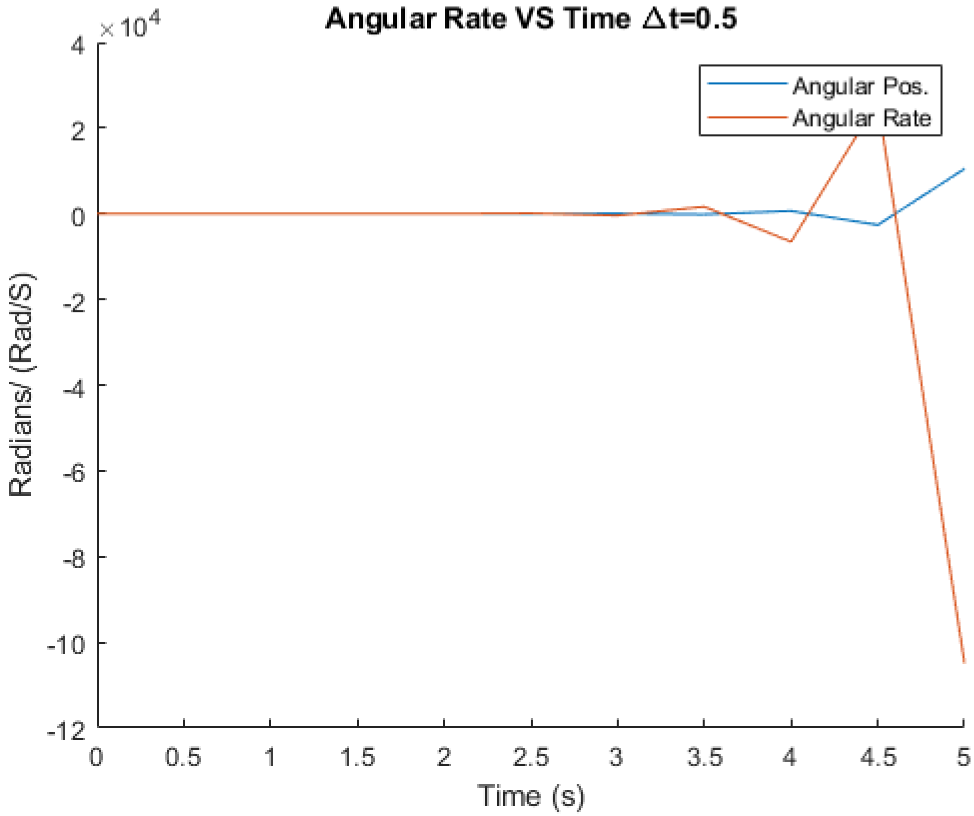
Problem 1

```
clear all;close all;clc
k = 398600;
r0 = 6678;
w0 = sqrt(k/(r0^3));

A = [0 1 0 0;w0^2+(2*k/(r0^3)) 0 0 2*r0*w0;0 0 0 1;0 (-2*w0/r0) 0 0];
B = [0 0;1 0;0 0;0 1/r0];
C = [1 0 0 0;0 0 1 0];
D = [0 0;0 0];

Sysc = ss(A,B,C,D);

Sysd = c2d(Sysc,10,'zoh');

[F,G,H,M] = ssdata(Sysd);
```

Published with MATLAB® R2019b

Problem 2

```
clear all;close all;clc
J = 10;
F = 100;
T = 10;
x = [0;0];

A = [0 1;0 -F/J];
B = [0;1/J];

dx = 0.5;
for i = 1:(5/dx)
    xdot = A*x(:,i) + B*T;
    x(:,i+1) = x(:,i) + xdot*dx;
end
t = 0:dx:5;
figure
hold on
plot(t,x(1,:))
title('Angular Position (rads')
xlabel('Time (s)')
ylabel('Radians/ (Rad/S)')
plot(t,x(2,:))
title('Angular Rate VS Time \Deltat=0.5')
legend('Angular Pos.','Angular Rate')
```



Solving system using ss and lsim

J = 10; F = 100; A = [0 1;0 -F/J]; B = [0;1/J]; C = [1 0;0 1]; D = [0;0]; T = 10; sys = ss(A,B,C,D);

dt = 0.005;

t = [0:dt:5];

u = ones(length(t),1)*T;

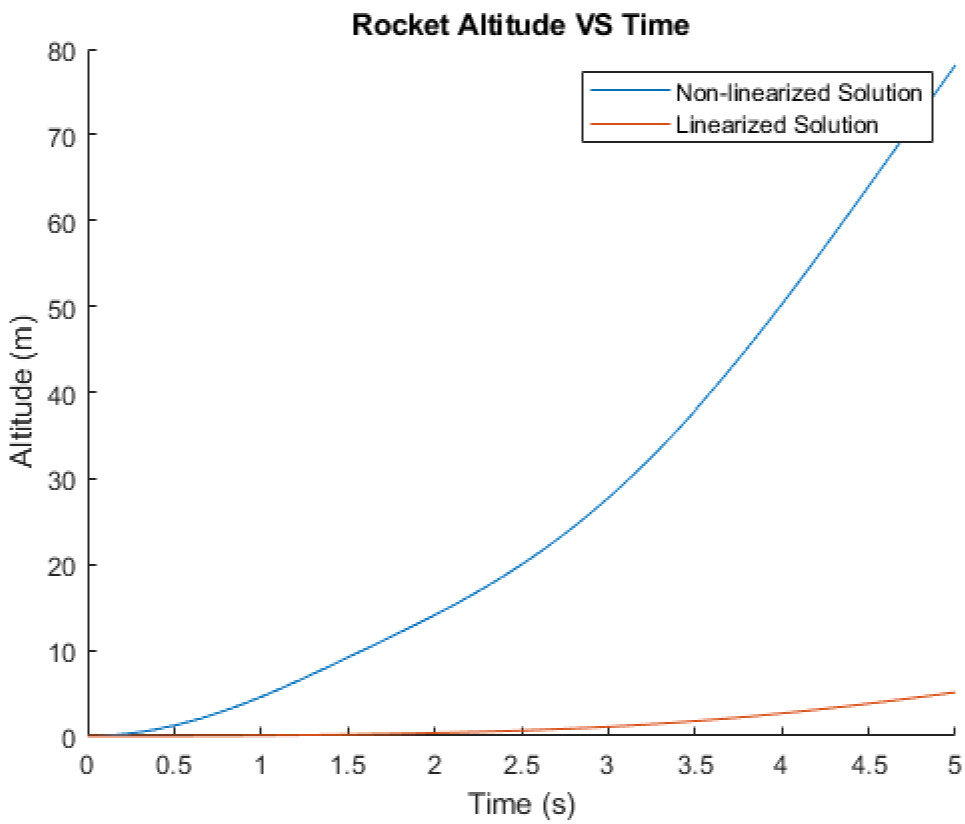
lsim(sys,u,t)

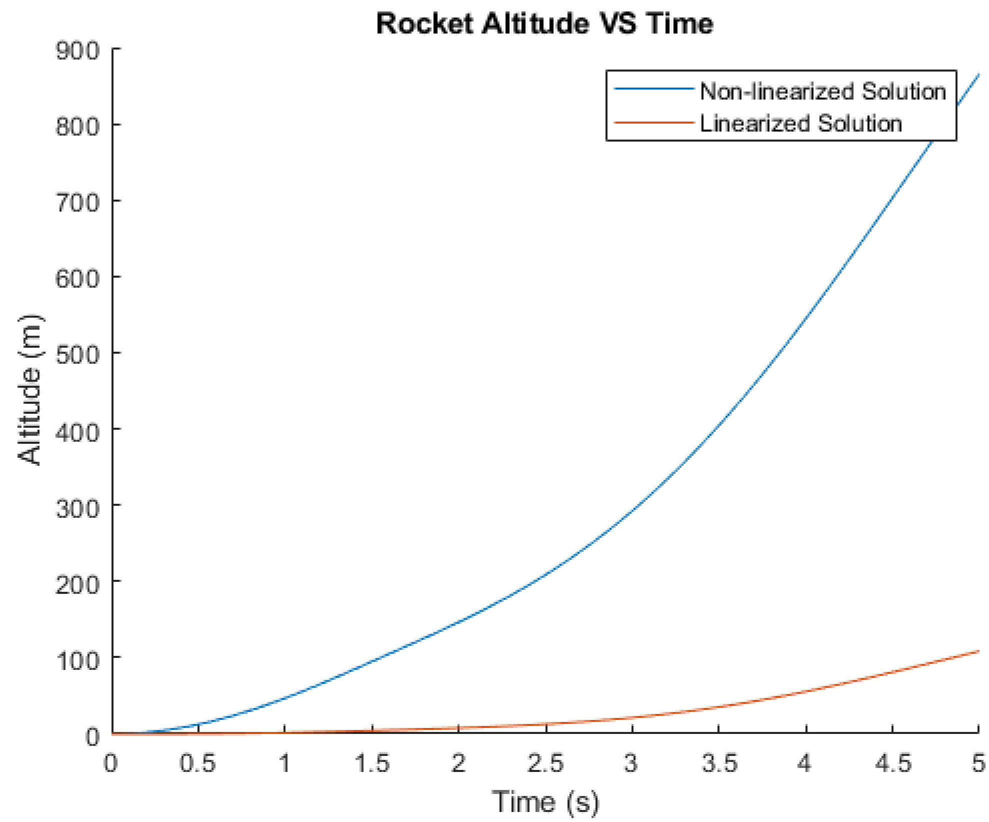
```
[V,D] = eig(A);
```

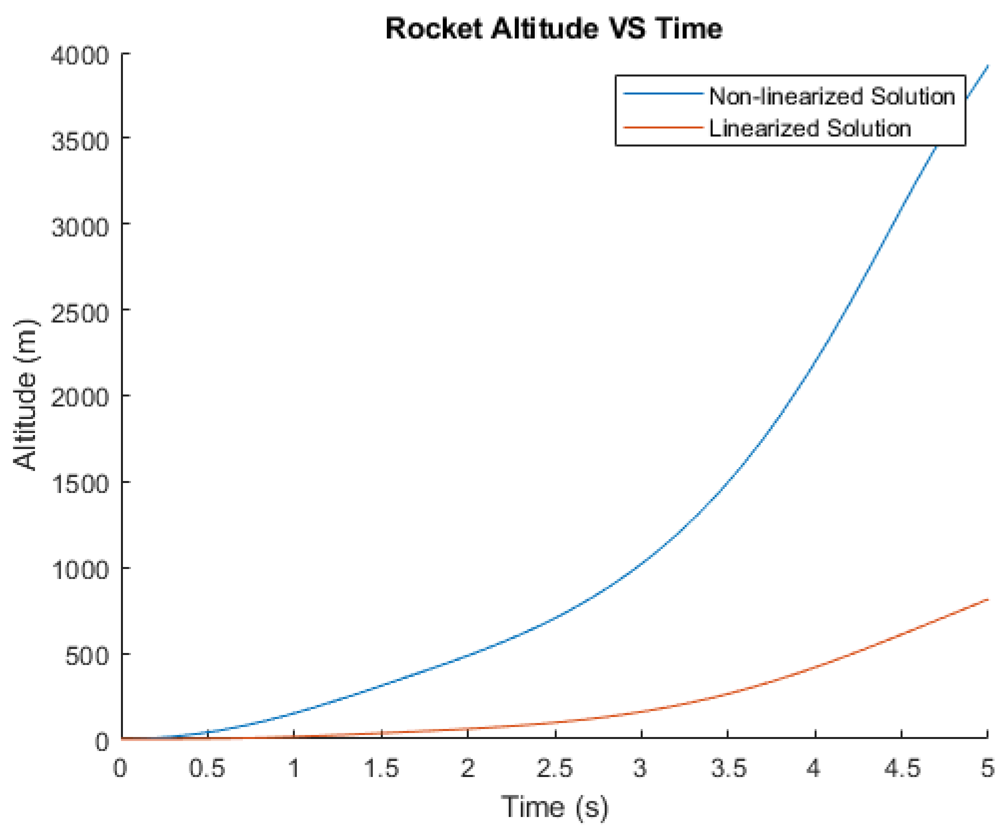
Published with MATLAB® R2019b

Problem 3

```
clear all;close all;clc

tspan = [0 5];
y0 = [0 0 1000];
du = [10 100 300];
for i = 1:3
    deltau = du(i);
    [t,y] = ode45(@(t,y)odefunNL(t,y,deltau),tspan,y0);

    [t2,y2] = ode45(@(t,y)odefunL(t,y,deltau),tspan,y0);

    figure(i)
    hold on
    plot(t,y(:,1))
    plot(t2,y2(:,1))
    title('Rocket Altitude VS Time')
    xlabel('Time (s)')
    ylabel('Altitude (m)')
    legend('Non-linearized Solution','Linearized Solution')
end
```




Rocket Altitude VS Time
Non-linearized Solution
Linearized Solution
Altitude (m)
Time (s)
0
100
200
300
400
500
600
700
800
900
0
0.5
1
1.5
2
2.5
3
3.5
4
4.5
5


Rocket Altitude VS Time
Non-linearized Solution
Linearized Solution
Altitude (m)
Time (s)
0
500
1000
1500
2000
2500
3000
3500
4000
0
0.5
1
1.5
2
2.5
3
3.5
4
4.5
5

Published with MATLAB® R2019b

Problem 4

```
clear all;close all;clc

k = 398600;
r0 = 6678;
w0 = sqrt(k/(r0^3));

A = [0 1 0 0;w0^2+(2*k/(r0^3)) 0 0 2*r0*w0;0 0 0 1;0 (-2*w0/r0) 0 0];

tspan = [0 10];
y0 = eye(4);
[t,y] = ode45(@(t,y)odefunp4(t,y,A),tspan,y0);

Phi10 = reshape(y(end,:),size(A));

tspan = [0 100];
y0 = eye(4);
[t,y] = ode45(@(t,y)odefunp4(t,y,A),tspan,y0);

Phi100 = reshape(y(end,:),size(A));
```

Published with MATLAB® R2019b

```
function dydt = odefunL(t,y,deltau)

K = 1000;
g = 50;
G = 6.673e-11;
M = 5.98e24;
R = 6.37e6;
x3_0 = 1000;
% deltau = 10;

u0 = (G*M*x3_0*exp((-G*M*t)/(R^2*K)))/(R^2*K);
u = u0 + deltau*abs(cos(t));

A = [0 1 0;(-2*G*M/(R^3)) (-g/x3_0) ((-K*u)/x3_0^2); 0 0 0];

B = [0;(K/x3_0);-1];

dydt = A*y + B*u;
```

```
Not enough input arguments.

Error in odefunL (line 11)
u0 = (G*M*x3_0*exp((-G*M*t)/(R^2*K)))/(R^2*K);
```

Published with MATLAB® R2019b

```
function dydt = odefunNL(t,y,deltau)
K = 1000;
g = 50;
G = 6.673e-11;
M = 5.98e24;
R = 6.37e6;
x3_0 = 1000;
% deltau = 10;

u0 = (G*M*x3_0*exp((-G*M*t)/(R^2*K)))/(R^2*K);
u = u0 + deltau*abs(cos(t));

dydt = zeros(3,1);

dydt(1) = y(2);
dydt(2) = ((K*u - g*y(2))/y(3)) - ((G*M)/(R + y(1))^2);
dydt(3) = -u;
end
```

```
Not enough input arguments.

Error in odefunNL (line 10)
u0 = (G*M*x3_0*exp((-G*M*t)/(R^2*K)))/(R^2*K);
```

Published with MATLAB® R2019b

```
function dydt = odefunp4(t,y,Phi)


y = reshape(y,size(Phi));
dydt = Phi*y;

dydt = reshape(dydt,16,1);
end
```

```
Not enough input arguments.

Error in odefunp4 (line 4)
y = reshape(y,size(Phi));
```

Published with MATLAB® R2019b