```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%  A423_HW7_Student.m
%
%  This script calculates the relative motion of a chase satellite about
%  an elliptical target orbit using the YA state transition matrix.
%
%  Author        : Lt Col Greg Frey                  Sep 2023
%  Modified by   : C2C Unsworth                          2024
%  Inputs        :
%
%  Outputs       :
%    X_traj      : Relative motion trajectory [km, km/rad, s, and rad].
%
%  Coupling
%   YA.m   Propagates initial scaled state for chase forward in time over
%          a specified timespan
%
%  References             :
%    Yamanaka, Koji, and Finn Ankersen. "New state transition matrix for
%    relative motion on an arbitrary elliptical orbit." J Guid Control Dyn
%    25.1 (2002): 60-66.
%    Spacecraft Formation Flying by Alfriend et. al., Section 5.6
%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%


clear all
close all
clc

%% Define constants, target parameters and initial chase relative position and velocity
mu=398600.5;    %Gravitational parameter for Earth

%%*********YOU FILL IN THE FOLLOWING CODE********************************%%
%Define target orbit parameters. Use the same parameters for part (a) and
%(b)
e= 0.5;                %Target eccentricity []
a= 3850;                %Target semi-major axis [km]
nu0= 170;              %Initial true anomaly at t=0, [rad]

%Define chase initial relative position and velocity
%For part (a), chose any initial conditions (keeping in mind that the chase
%and target must be close for the TH equations to yield a good estimate of
%relative motion). For part (b), modify the initial condition you used in
%(a) to yield a periodic relative motion trajectory
x= 5; %[km]
y= 8; %[km]
z= 5; %[km]
```

```
p=a*(1-e^2);                  %Parameter for the target [km]
h=sqrt(mu*p);                 %Specific angular momentum of target [km^2/s]
r0=p/(1+e*cos(nu0));          %Initial orbital radius for target [km]
nudot0 =h/r0^2;

k = 1 + e * cos(nu0);

xdot = 0.0002; %[km/s]
% ydot =- 0.0002;
ydot= (-nudot0 * x - e * sin(nu0) * (xdot - nudot0 * y)) / ( k ) - nudot0 * x; %[km/s]
zdot= 0; %[km/s]
%%*************************************************************************%%

%% Form initial state vectors
% Form initial state vector (relative position and velocity) of the chase
% Initial state in km and km/s
X0i=[ x;    %x
      xdot; %xdot
      y;    %y
      ydot; %ydot
      z;    %z
      zdot];%zdot

% Calculate additional target parameters based on inputs above
p=a*(1-e^2);                  %Parameter for the target [km]
h=sqrt(mu*p);                 %Specific angular momentum of target [km^2/s]
r0=p/(1+e*cos(nu0));          %Initial orbital radius for target [km]
nudot0=h/r0^2;                %Rate of change of true anomaly for target at t=0 [rad/s]

% Calculate the scaled initial state for the chase for use in YA Solution
% to the TH eqns
X0=[X0i(1)*(1+e*cos(nu0));                                  %xbar
    -e*sin(nu0)*X0i(1)+(1+e*cos(nu0))*(X0i(2)/nudot0);      %xbar'
    X0i(3)*(1+e*cos(nu0));                                  %ybar
    -e*sin(nu0)*X0i(3)+(1+e*cos(nu0))*(X0i(4)/nudot0);      %ybar'
    X0i(5)*(1+e*cos(nu0));                                  %zbar
    -e*sin(nu0)*X0i(5)+(1+e*cos(nu0))*(X0i(6)/nudot0)];     %zbar'

%% Define time parameters for simulation
tf=2*2*pi*sqrt(a^3/mu);       %Total simulation time - Default to 2 periods of target
orbit. [sec]
tstep=10;                     %Timestep [sec]
%Define simulation times
tspan=[0:tstep:tf];

%% Propagate the initial state forward using the Yamanaka-Ankersen STM
% Output is a matrix, Xtraj, with 1 row for each time step.
% Each row consists of [xbar xbar' ybar ybar' zbar zbar' t nu]
[Xtraj]=YA(X0,nu0,e,a,tspan);

%Apply inverse coordinate scaling to position components to recover
```

```
%un-scaled relative position
Xtraj(:,1)=Xtraj(:,1)./(1+e*cos(Xtraj(:,8))); %recovers unscaled x(t)
Xtraj(:,3)=Xtraj(:,3)./(1+e*cos(Xtraj(:,8))); %recovers unscaled y(t)
Xtraj(:,5)=Xtraj(:,5)./(1+e*cos(Xtraj(:,8))); %recovers unscaled z(t)

%% Make Plots of the relative position trajectory
plot(Xtraj(:,3), Xtraj(:,1))
xlabel('y [km]')
ylabel('x [km]')
grid on
title('In-plane relative motion trajectory')

figure
plot3(Xtraj(:,3), Xtraj(:,1), Xtraj(:,5))
xlabel('y [km]')
ylabel('x [km]')
grid on
zlabel('z [km]')
title('3-dimensional relative motion trajectory')

figure
subplot(3,1,1)
plot(Xtraj(:,7), Xtraj(:,1))
xlabel('time [sec]')
ylabel('x [km]')
grid on
title('x, y and z components of relative motion trajectory')

subplot(3,1,2)
plot(Xtraj(:,7), Xtraj(:,3))
xlabel('time [sec]')
ylabel('y [km]')
grid on

subplot(3,1,3)
plot(Xtraj(:,7), Xtraj(:,5))
xlabel('time [sec]')
ylabel('z [km]')
grid on
```