Main Script Part 2

```
clear all
close all
clc
format long e
global mu_Earth ae JD AU Target Chasser
mu_Earth = 3.986004415E5;
ae = 6378.136;
JD = 2456296.25;
AU = 149597870.7;

% Setting the initial conditon
%~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~%
for k = 1
    % Target initial conditions(Orbital Elements)
    %~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~%
    % Target initial conditions(Orbital Elements)
    a1      = 1.5E+4; % Semi-major axis in Km
    e1      = 0.1;    % Eccentricity
    inc1    = 10;     % Inclination in deg0
    BigOmg1 = 50;     % RAAN in deg
    LitOmg1 = 10;     % AOP in deg
    f1     = -80;       % True anomaly in deg
    Target = Spacecraft([10 6 2 40 10 0.2 0.5 6 12]); % Describing the
 Chief sailcraft characteristic (refer Spacecraft class definition)

    % Relative orbital elements of the deputy
    %~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~%
    dela = 0;                       % Relative semi-major axis in Km
    deli = 1/(4*a1);                % Relative inclination in deg0
    dele = -1/(2*a1);              % Relative eccentricity, in rad
    delLitOmg = 1/(2*a1);           % Relative RAAN in rad
    delBigOmg = 0;                  % Relative AOP in rad
    delf = 0;                       % Relative true anomaly in rad


    % Chaser initial conditions(Orbital Elements)
    %~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~%
    a2      = a1;           % Semi-major axis in Km
    e2      = e1 + dele;    % Eccentricity
    inc2    = inc1;         % Inclination in deg
    BigOmg2 = BigOmg1;      % RAAN in deg
    LitOmg2 = LitOmg1;      % AOP in deg
    f2      = f1;           % True anomaly in deg
    Chasser = Spacecraft([30 30 4 116 40 0.9 0.9 3000 6000]); %
 Describing the Deputy sailcraft characteristic (refer Spacecraft
 class definition)

    % Converting angles in radians
    inc1 = deg2rad(inc1); BigOmg1 = deg2rad(BigOmg1); LitOmg1 =
 deg2rad(LitOmg1); f1 = deg2rad(f1);
```

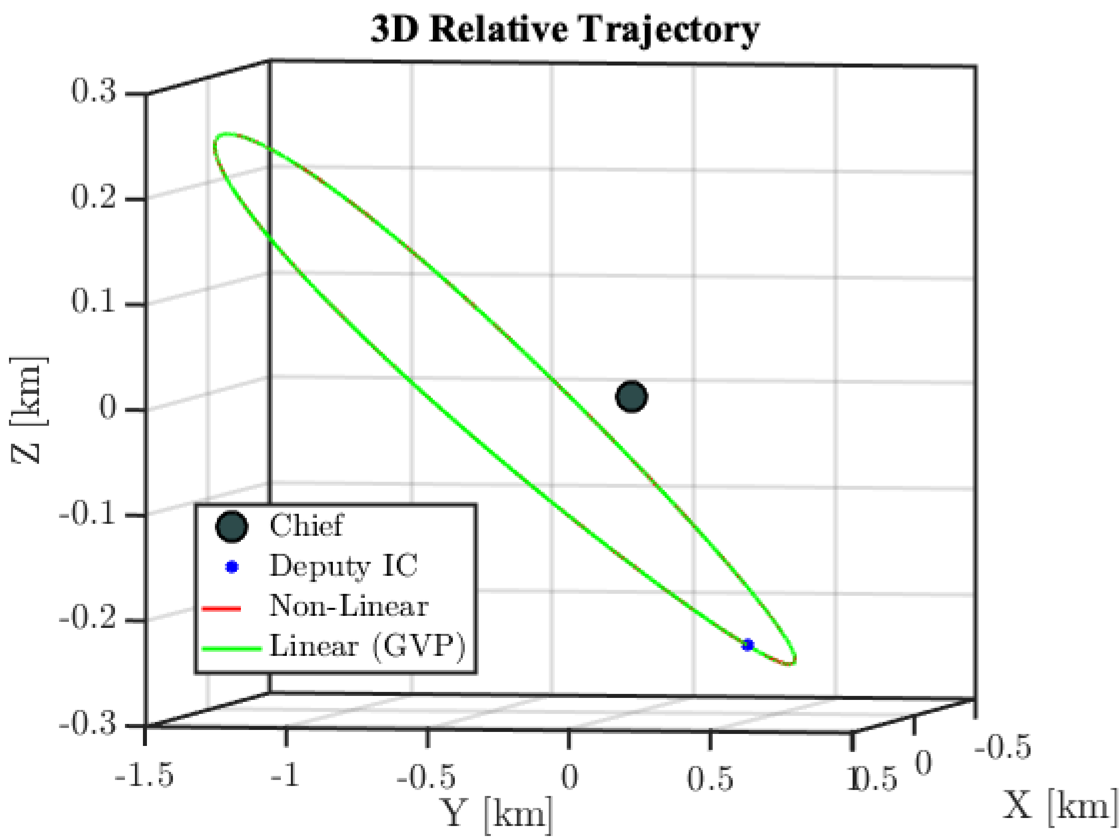
```
    COE1 = [a1,e1,inc1,BigOmg1,LitOmg1,f1];
    [Position_target,Velocity_target]  = COEstoRV(COE1,mu_Earth);

    inc2 = deg2rad(inc2) + deli; BigOmg2 = deg2rad(BigOmg2); LitOmg2 =
 deg2rad(LitOmg2)+delLitOmg; f2 = deg2rad(f2);
    COE2 = [a2,e2,inc2,BigOmg2,LitOmg2,f2];
    [Position_chaser,Velocity_chaser]  = COEstoRV(COE2,mu_Earth);

    % Setting the integrator parameters
    Period = 2*pi*sqrt(a1^3/mu_Earth);
    IntegrationTime = Period;
    tspan  = linspace(0,IntegrationTime,10000);
    options = odeset('RelTol',2.22045e-14,'AbsTol',2.22045e-30);
end

for k=1
    tic
    [~, Chief_OE] =
 ode113(@(t,COE,nu,u,f)GVE(t,COE,Target),tspan,COE1,options);
    toc

    tic
    [~, Deputy_OE] =
 ode113(@(t,COE,nu,u,f)GVE(t,COE,Chasser),tspan,COE2,options);
    toc
    X_rel_linear1 = nan(length(tspan),6);
    R_target2 = nan(length(tspan),3);
    V_target2 = nan(length(tspan),3);

    for j = 1:length(tspan)

        [R_target2(j,:),V_target2(j,:)]  =
 COEstoRV(Chief_OE(j,:),mu_Earth);

        % Computing the relative using the linear maping COE2 =
 [a2,e2,inc2,BigOmg2,LitOmg2,f2];
        % Nonsingular Chief orbital elements
        a_c = Chief_OE(j,1); e_c = Chief_OE(j,2); LitOmg_c =
 Chief_OE(j,5);  f_c = Chief_OE(j,6);
        theta_c = LitOmg_c + f_c;
        inc_c = Chief_OE(j,3);
        q1_c = e_c * cos(LitOmg_c);
        q2_c = e_c * sin(LitOmg_c);
        BigOmg_c = Chief_OE(j,4);

        % Nonsingular Deputy orbital elements
        a_d = Deputy_OE(j,1); e_d = Deputy_OE(j,2); LitOmg_d =
 Deputy_OE(j,5);  f_d = Deputy_OE(j,6);
        theta_d = LitOmg_d + f_d;
        inc_d = Deputy_OE(j,3);
        q1_d = e_d * cos(LitOmg_d);
        q2_d = e_d * sin(LitOmg_d);
        BigOmg_d = Deputy_OE(j,4);
```

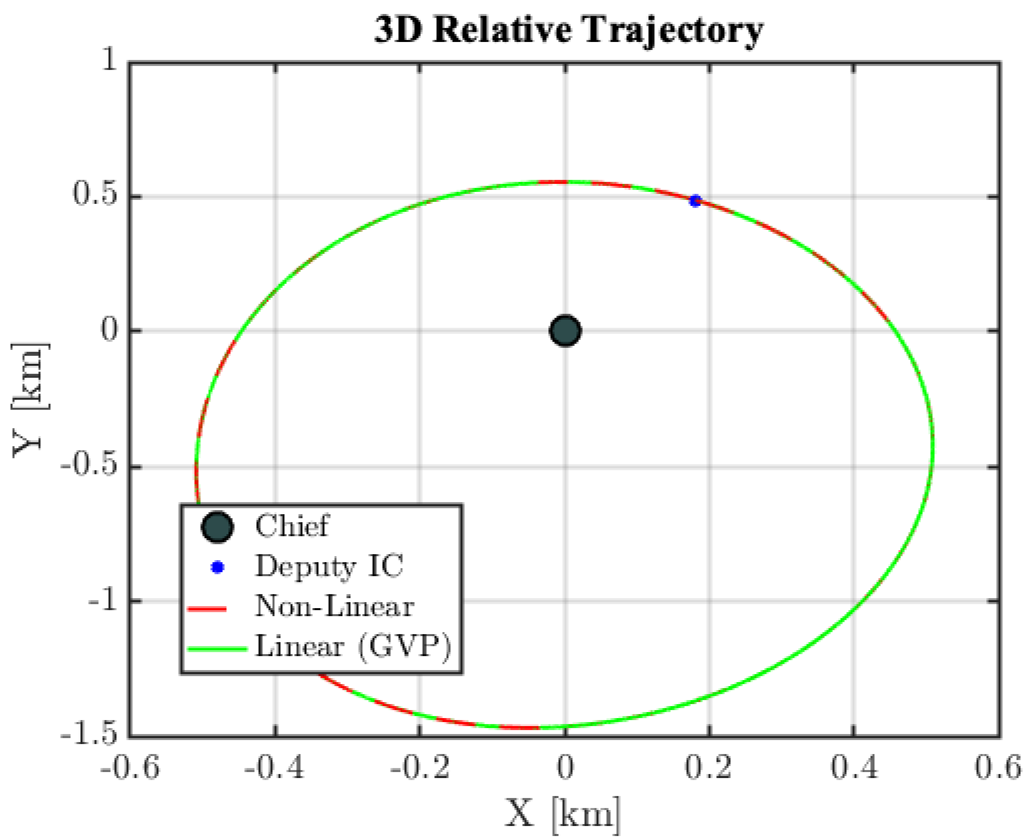
```
        % Relative orbital elements
        del_a = a_d - a_c; del_e = e_d - e_c;  delLitOmg = LitOmg_d -
 LitOmg_c;
        del_theta = theta_d - theta_c;
        del_inc = inc_d - inc_c;
        del_q1 = del_e*cos(LitOmg_c) - e_c*sin(LitOmg_c)*delLitOmg;
        del_q2 = del_e*sin(LitOmg_c) + e_c*cos(LitOmg_c)*delLitOmg;
        del_BigOmg = BigOmg_d - BigOmg_c;

        COE = [a_c, theta_c, inc_c, q1_c, q2_c, BigOmg_c]';
        delCOE = [del_a, del_theta, del_inc, del_q1, del_q2,
 del_BigOmg]';
        [A] = ForwardMapping(COE, mu_Earth);
        Rho_aug = A*delCOE;
        X_rel_linear1(j,:) = Rho_aug';


    end
end

% Integrating the relative motion in the chief's LVLH frame
%~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~%
for i = 1

    TN = DCM(Position_target,Velocity_target); rt_norm =
 norm(Position_target);
    h_vec = cross(Position_target,Velocity_target); h_norm =
 norm(h_vec);
    eh = h_vec/h_norm;
    U_eci_Target  = 0*F_CanonBall(tspan(1),Position_target,Target); %
 SRP force on the Target
    N_nudot = h_vec/rt_norm^2 + dot(U_eci_Target,eh)*Position_target/
h_norm;
    NR_rel = Position_chaser - Position_target; NV_rel2 =
 Velocity_chaser - Velocity_target;
    TR_rel0 = TN*NR_rel; TV_rel0 = TN*(NV_rel2 -
 cross(N_nudot,NR_rel));
    X_aug0 = [Position_target; Velocity_target; TR_rel0; TV_rel0];
    index = 1;
    tic
    [~, X_rel] = ode113(@(t,Xaug)RelativeMotionODE_cannonball(t, Xaug,
 Target, Chasser),tspan,X_aug0,options);
    toc

end


% Plotting the 3D realtive trajectory
%~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~%
for k = 1:3
    % Plotting the 3D trajectories
    c1 = rgb('Crimson'); c2 = rgb('DarkSlateGray');
    figure
    for jj=1
```

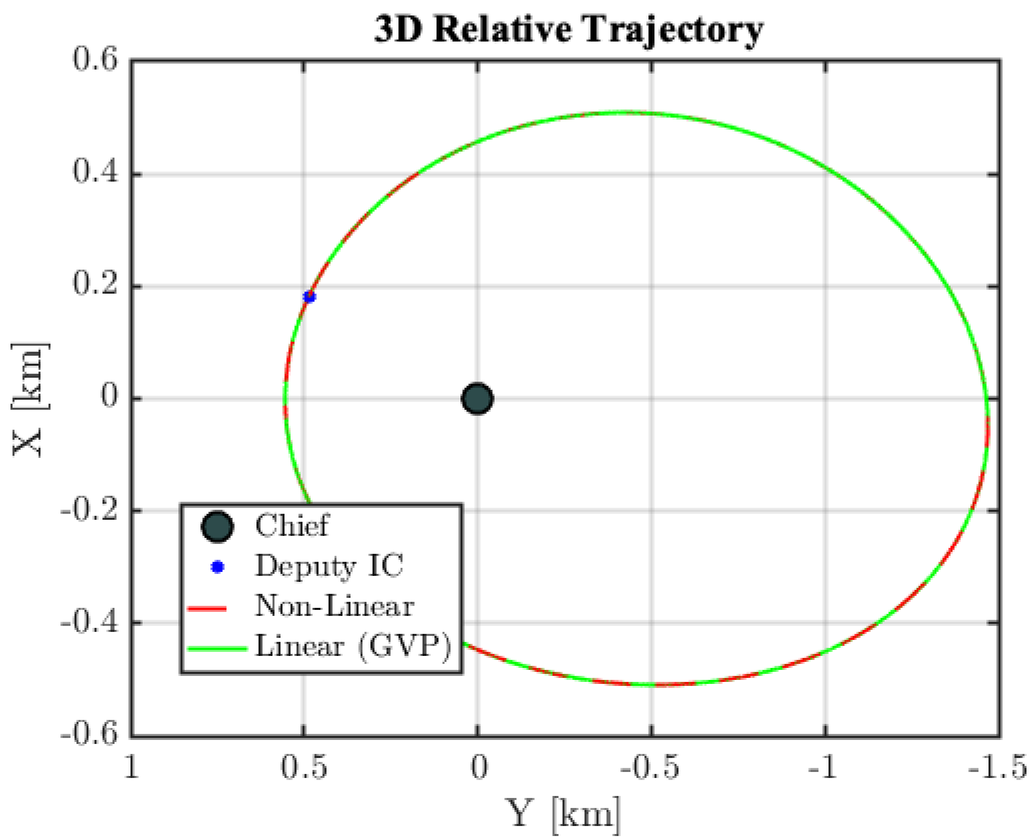
```
        h1 =
 plot3(X_rel_linear1(:,1),X_rel_linear1(:,2),X_rel_linear1(:,3),'g');
        hold on
        h3 = plot3(X_rel(:,7),X_rel(:,8),X_rel(:,9),'r--');

        h4 = plot3(X_rel(1,7),X_rel(1,8),X_rel(1,9),'go',...
            'LineWidth',2,...
            'MarkerEdgeColor','b',...
            'MarkerFaceColor','b',...
            'MarkerSize',5);

        h5 = plot3(0,0,0,'bo',...
            'LineWidth',2,...
            'MarkerEdgeColor','k',...
            'MarkerFaceColor',c2,...
            'MarkerSize',15);
        grid on
        xlabel('X [km]')
        ylabel('Y [km]')
        zlabel('Z [km]')
        title('{\color{black} 3D Relative
 Trajectory}','interpreter','tex')
        h = legend([h5, h4, h3, h1],{'Chief','Deputy IC','Non-
Linear','Linear (GVP)'});
        rect = [0.25, 0.25, 0.1, 0.1];
        set(h, 'Position', rect)
        box on
    end

    if k == 1
        view(100,3)
    elseif k==2
        view(0,90)
    else
        view(-90,90)
    end

end


for jj=1

    DQ1 = [X_rel_linear1(:,1) X_rel_linear1(:,4) X_rel_linear1(:,2)
 X_rel_linear1(:,5) X_rel_linear1(:,3) X_rel_linear1(:,6)]; %
    DQ2 = [X_rel(:,7) X_rel(:,10) X_rel(:,8) X_rel(:,11) X_rel(:,9)
 X_rel(:,12)]; %
    figure
    YLabel={'$\delta x_{_{_{\!\!\!\!\!\!\!\!\!\!\!True}}} - \delta
 x_{_{_{\!\!\!\!\!\!\!\!\!\!\!\!\!\!Approx}}}(km)$',...
        '$\delta \dot{x}_{_{_{\!\!\!\!\!\!\!\!\!\!\!True}}} - \delta
 \dot{x}_{_{_{\!\!\!\!\!\!\!\!\!\!\!\!\!\!Approx}}}(km)$',...
        '$\delta y_{_{_{\!\!\!\!\!\!\!\!\!\!\!True}}} - \delta
 y_{_{_{\!\!\!\!\!\!\!\!\!\!\!\!\!\!Approx}}}(km)$',...
```

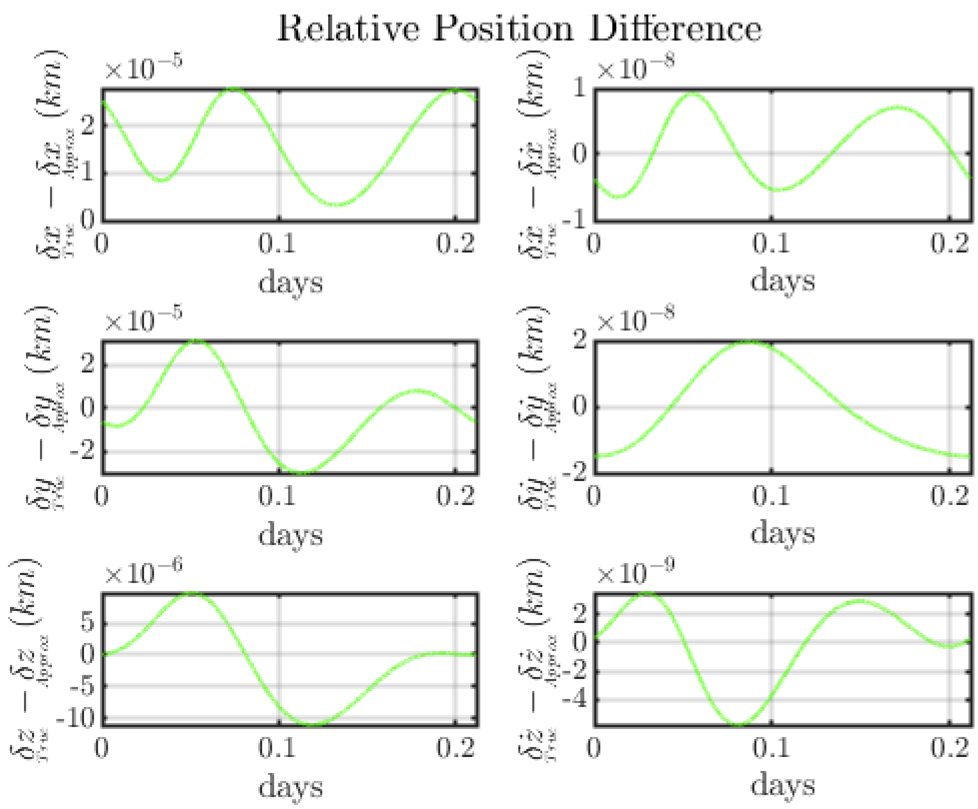
```
        '$\delta \dot{y}_{_{_{\!\!\!\!\!\!\!\!\!\!\!\!True}}} - \delta
 \dot{y}_{_{_{\!\!\!\!\!\!\!\!\!\!\!\!\!Approx}}}(km)$',...
        '$\delta z_{_{_{\!\!\!\!\!\!\!\!\!\!\!\!True}}} - \delta
 z_{_{_{\!\!\!\!\!\!\!\!\!\!\!\!\!Approx}}}(km)$',...
        '$\delta \dot{z}_{_{_{\!\!\!\!\!\!\!\!\!\!\!\!True}}} - \delta
 \dot{z}_{_{_{\!\!\!\!\!\!\!\!\!\!\!\!\!Approx}}}(km)$'};
    for i=1:6
        subplot(3,2,i)
        plot1 = plot(tspan/86400,DQ1(:,i)-DQ2(:,i));
        ylabel(YLabel(i))
        xlabel('days')
        grid on

        plot1.Color(4) = 0.6;
        %              ax = gca;
        %              ax.GridColor = [0.9, 0.9, 0.9];  % [R, G, B]
        %              ax.XColor = 'k';
        %              ax.YColor = 'k';
        %              set(gca,'color','k')
        %              set(gcf,'color',[0.3, 0.3, 0.3])
    end
    ha = axes('Position',[0 0 2.2 1],'Xlim',[0 1],'Ylim',[0
 1],'Box','off','Visible','off','Units','normalized', 'clipping'
 , 'off');
    text(0.13, 0.98,sprintf('Relative Position Difference'))
end
```

Elapsed time is 0.487324 seconds.
Elapsed time is 0.296204 seconds.
Elapsed time is 8.264004 seconds.


3D Relative Trajectory
Chief
Deputy IC
Non-Linear
Linear (GVP)
Z [km]
Y [km]
X [km]


3D Relative Trajectory
Chief
Deputy IC
Non-Linear
Linear (GVP)
Y [km]
X [km]


3D Relative Trajectory
Chief
Deputy IC
Non-Linear
Linear (GVP)
X [km]
Y [km]


Relative Position Difference
days

Published with MATLAB® R2018b