Table of Contents


```
%{
Authors: Nevaeh Montoya, Rhiannon Rapplean
Assignment: Project 1 ISS
Creation Date: 10/23/2024
Inputs: Location data (2 sets)
Outputs: Time of closest approach, distance, recommended proximity warnings
Purpose: Analyze risk of two orbital bodies colliding
%}

clear;
clc;
close all;
```

Load data

```
% this is going to load with the names
% here im using variblenamerule and setting it to preserve
% We know this isn't necessary but the warning is annoying
data_ISS_A = readtable('Data_ISS_A.csv', 'VariableNamingRule','preserve');
data_ISS_B = readtable('Data_ISS_B.csv','VariableNamingRule','preserve');

%Setting giving varible names
data_ISS_A.Properties.VariableNames = {'Time_s','X_km', 'Y_km'};
data_ISS_B.Properties.VariableNames = {'Time_s','X_km', 'Y_km'};

%this is taking the columns into sperate varibles
time_a = data_ISS_A.Time_s;
x_a = data_ISS_A.X_km;
y_a = data_ISS_A.Y_km;

time_b = data_ISS_B.Time_s;
x_b = data_ISS_B.X_km;
y_b = data_ISS_B.Y_km;
```

linear fit

```
%least square for x and y
% makes y = mx +b
```

```
%linear model for x and y over time so we can get the velocity for both
%lin fit for A
c_x_a = polyfit(time_a, x_a, 1);
c_y_a = polyfit(time_a, y_a, 1);
u_a = c_x_a(1);
x_a0 = c_x_a(2);
v_a = c_y_a(1);
y_a0 = c_y_a(2);

%lin fit for B
c_x_b = polyfit(time_b, x_b, 1);
c_y_b = polyfit(time_b, y_b, 1);
u_b = c_x_b(1);
x_b0 = c_x_b(2);
v_b = c_y_b(1);
y_b0 = c_y_b(2);
```

time of the closest approach

this is using the formula that was given to cal time when they are close

```
% this will be getting the numerator and demoninator first
n = -((x_b0-x_a0) * (u_b-u_a) + (y_b0-y_a0) * (v_b- v_a));
d = (u_b-u_a)^2 + (v_b- v_a)^2;
T_ca = n / d;
```

min distance at T of the closet approach

```
%T_ca goes back to the postion equation
%positon at T_Ca
x_a_Tca = x_a0 + u_a * T_ca;
y_a_Tca = y_a0 + v_a * T_ca;
x_b_Tca = x_b0 + u_b * T_ca;
y_b_Tca = y_b0 + v_b * T_ca;

% this is getting min distance
D_min = sqrt((x_b_Tca - x_a_Tca)^2 + (y_b_Tca-y_a_Tca)^2);
```

Error propagation

```
%Calculating the standard deviation from our best fit to find uncertainty
%Calculating velocity residuals
u_a_res = (diff(x_a)./diff(time_a))-u_a;
u_b_res = (diff(x_b)./diff(time_b))-u_b;
v_a_res = (diff(y_a)./diff(time_a))-v_a;
v_b_res = (diff(y_b)./diff(time_b))-v_b;
%Taking velocity standard deviations
s_u_a = std(u_a_res);
s_u_b = std(u_b_res);
s_v_a = std(v_a_res);
s_v_b = std(v_b_res);
```

```
%Setting position functions
x_a_func = @(t) u_a*t + x_a0;
x_b_func = @(t) u_b*t + x_b0;
y_a_func = @(t) v_a*t + y_a0;
y_b_func = @(t) v_b*t + y_b0;
%Getting position residuals
x_a_res = x_a - x_a_func(time_a);
x_b_res = x_b - x_b_func(time_b);
y_a_res = y_a - y_a_func(time_a);
y_b_res = y_b - y_b_func(time_b);
%Getting position standard deviations
s_x_a = std(x_a_res);
s_x_b = std(x_b_res);
s_y_a = std(y_a_res);
s_y_b = std(y_b_res);
```

Error propagation for T_ca

```
%Partial derivatives
dt_ca_dxa0 = -(u_b-u_a)/d;
dt_ca_dya0 = -(v_b-v_a)/d;
dt_ca_dxb0 = (u_b-u_a)/d;
dt_ca_dyb0 = (v_b-v_a)/d;
dt_ca_dua = ((x_b0-x_a0)/((u_b-u_a)^(2)+(v_b-v_a)^(2)))-((2*(u_a-u_b)*(-
x_b0+x_a0))-((y_b0-y_a0)*(v_b-v_a)))/((u_b-u_a)^(2)+(v_b-v_a)^(2))^(2);
dt_ca_dub = ((-x_b0+x_a0)/((u_b-u_a)^(2)+(v_b-v_a)^(2)))-((2*(u_b-u_a)*(-
x_b0+x_a0))-((y_b0-y_a0)*(v_b-v_a)))/((u_b-u_a)^(2)+(v_b-v_a)^(2))^(2);
dt_ca_dva = ((y_b0-y_a0)/((u_b-u_a)^(2)+(v_b-v_a)^(2)))-((2*(v_a-v_b)*(-
x_b0+x_a0))-((y_b0-y_a0)*(v_b-v_a)))/((u_b-u_a)^(2)+(v_b-v_a)^(2))^(2);
dt_ca_dvb = ((-y_b0-y_a0)/((u_b-u_a)^(2)+(v_b-v_a)^(2)))-((2*(v_b-v_a)*(-
x_b0+x_a0))-((y_b0-y_a0)*(v_b-v_a)))/((u_b-u_a)^(2)+(v_b-v_a)^(2))^(2);

%uncertainty of tca
s_Tca= sqrt((dt_ca_dxa0* s_x_a)^2 + (dt_ca_dya0*s_y_a)^2 +
(dt_ca_dxb0*s_x_b)^2 +(dt_ca_dyb0*s_y_b)^2 + (dt_ca_dua*s_u_a)^2 +
(dt_ca_dub*s_u_b)^2 + (dt_ca_dva*s_v_a)^2 + (dt_ca_dvb*s_v_b)^2);


%Error prop of dmin
%this will also be with partial dervis
dD_minx = -(x_b_Tca-x_a_Tca)/D_min;
dD_miny = -(y_b_Tca-y_a_Tca)/D_min;

%this is the uncertiany for dmin
s_D_min = sqrt((dD_minx*s_x_a)^2+ (dD_miny*s_y_a)^2 + (dD_minx*s_x_b)^2 +
(dD_miny*s_y_b)^2 );

%display the uncertainty
fprintf('T_ca: %.2f +/- %.2f second\n', T_ca, s_Tca);
fprintf('D_min: %.2f +/- %.2f Km\n', D_min, s_D_min);
```

```
T_ca: 220.95 +/- 1843.53 second
D_min: 36.17 +/- 16.17 Km
```

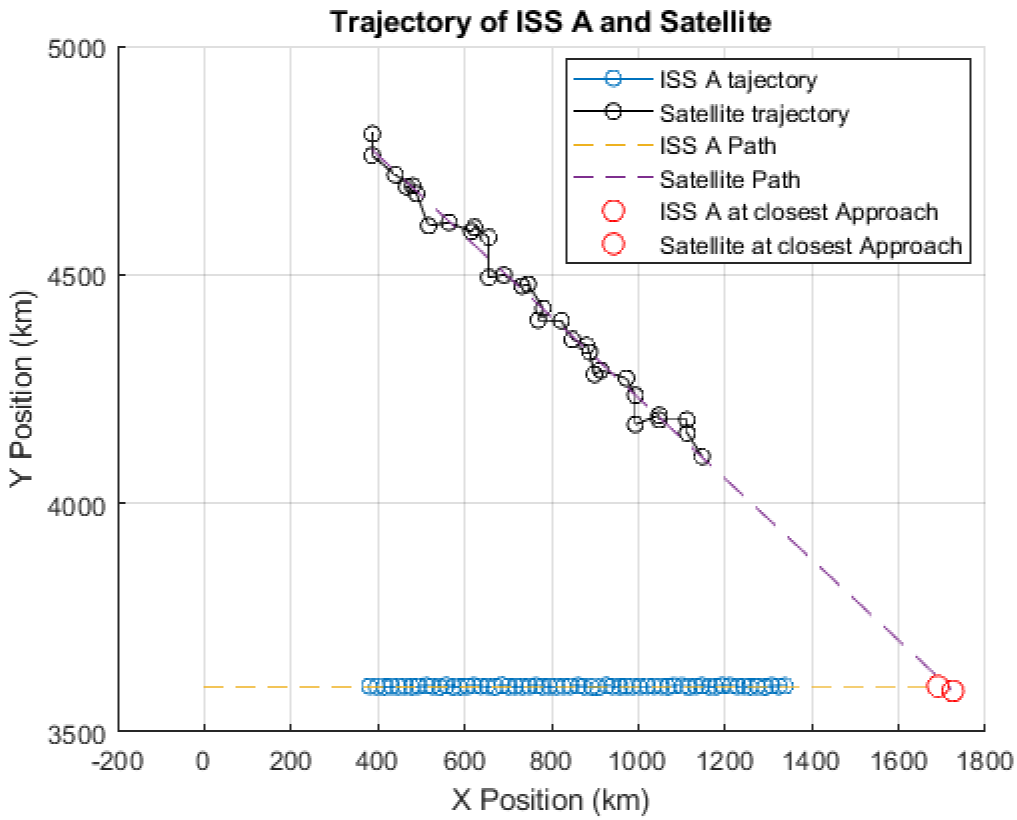
descion making and warning

```
% determining if we should use prevetive manuvers
% use D_min thresholds

% im going to use a if statments for warning
if (D_min-s_D_min) < 1.8
    warning_code = 'Red - Action must be taken';
elseif (D_min-s_D_min) < 28.2
    warning_code = 'Yellow - Plans are devloped';
else
    warning_code = 'Green - All clear';
end


%displaying
fprintf('T_ca: %.2f second\n', T_ca);
fprintf('D_min: %.2f Km\n', (D_min-s_D_min));
fprintf( 'warning code: %s\n', warning_code);
```

```
T_ca: 220.95 second
D_min: 20.00 Km
warning code: Yellow - Plans are devloped
```

Plotting

```
%postiton and trajectories to show closest approach
figure;
hold on;
plot(x_a, y_a,'-o','DisplayName','ISS A tajectory');
plot(x_b, y_b,'-ok','DisplayName','Satellite trajectory');
plot([x_a0,x_a_Tca], [y_a0, y_a_Tca], '--', 'DisplayName','ISS A Path');
plot([x_b0,x_b_Tca], [y_b0, y_b_Tca], '--', 'DisplayName','Satellite Path');
plot(x_a_Tca, y_a_Tca,'ro', 'MarkerSize',8, 'DisplayName','ISS A at closest
Approach');
plot(x_b_Tca, y_b_Tca, 'ro', 'MarkerSize',8, 'DisplayName','Satellite at
closest Approach');
grid on;

%Setting labels & tidying
xlabel('X Position (km)');
ylabel('Y Position (km)');
title('Trajectory of ISS A and Satellite');
legend('Location','best');
hold off;
```



Published with MATLAB® R2023b