Table of Contents

run simulation

```
% simulation configuration
tspan = [0 10];
dt = 1e-2;
ode_opts = odeset( 'InitialStep', dt, 'MaxStep', dt, 'Events', @myEvent);

[t, Y] = ode23t(@(t, y) candp_6dof(t, y), tspan, Y0, ode_opts);
```

plot state variables and trajectory

```
mk_plots( t, Y, Y0);
```

frisbee 6dof ODE equations of motion

```
function dydt = candp_6dof( t, y)
% NOTE: per simulation of a spin-stabilised sports disc and
%       crowther & potts ~2003 source code

    % get workspace variables
    disc = evalin( 'base', 'disc');
    amb = evalin( 'base', 'amb');

    % unpack state variables
    % x = st(1); y = st(2); z = st(3);         % X1
    phi = y(4); theta = y(5); psi = y(6);       % TH1
    u = y(7); v = y(8); w = y(9);               % dX2dt
    p = y(10); q = y(11); r = y(12);            % dTH2dt

    % X1 = [x y z]';                            % X1
    TH1 = [phi theta psi]';                     % TH1

    % body reference variables
    dX2dt = [u v w]';                           % dX2dt
    dTH2dt = [p q r]';                          % dTH2dt

    w_tilde = [0 -r q; r 0 -p; -q p 0];
```

```
% calculate earth reference variables

Ta21 = Ta( TH1);
Ta12 = Ta21';
Tr21 = Tr( TH1);

dX1dt = Ta21 * dX2dt;                       % dX1dt
dTH1dt = Tr21 * dTH2dt;                     % dTH1dt

mg = (disc.m .* amb.g .* [0 0 1]');

% calculate zero sideslip body axes variables
beta2 = atan2( v, u);                       % sideslip angle

Ta32 = Ta( [0 0 beta2]');
Ta23 = Ta32';

dX3dt = Ta23 * dX2dt;                       % dX3dt
dTH3dt = Ta23 * dTH2dt;                     % dTH3dt

% calculate aerodynamic coefficients
u3 = dX3dt(1); w3 = dX3dt(3);
p3 = dTH3dt(1); q3 = dTH3dt(2); r3 = dTH3dt(3);

alpha3 = atan2( w3, u3);                    % angle of attack

Ta34 = Ta( [0 alpha3 0]');
Ta43 = Ta34';

v_inf = norm( dX2dt);                       % freestream wind speed

CDrag = Cdrag_fcn( alpha3, disc);
CSide = Cside_fcn( r3, v_inf);
CLift = Clift_fcn( alpha3, disc);

CL = CL_fcn( p3, v_inf, disc);
CM = CM_fcn( alpha3, q3, v_inf, disc);
CN = CN_fcn( r3, v_inf, disc);

% calculate zero sideslip wind axes forces and moments
q_inf = 0.5 * amb.rho * v_inf^2;            % dynamic pressure
qS = q_inf * disc.S;
qSc = qS * disc.c;

F4 = qS .* [-CDrag CSide -CLift]';          % F4
M4 = qSc .* [CL CM CN]';                    % M4

% calculate body forces and moments
Ta42 = Ta32 * Ta43;

F2aero = Ta42 * F4;                         % F2 aero
F2g = Ta12 * mg;                            % F2 g
F2 = F2aero + F2g;                          % F2
M2 = Ta42 * M4;                             % M2
```

```
    % calculate body accelerations
    dX2dt2 = (F2 .* (disc.m^-1)) - (w_tilde * dX2dt);
    dTH2dt2 = disc.I^-1 * (M2 - w_tilde * (disc.I * dTH2dt));

    dydt = [dX1dt' dTH1dt' dX2dt2' dTH2dt2']';
end
```

event handler that will stop simulation when the frisbee hits the ground

```
function [value, isterminal, direction] = myEvent(t, st)
    % stop simulation when disc hits the ground
    z = st(3); z_thr = 0;
    value = (z >= z_thr);
    isterminal = 1;
    direction = 0;
end
```

reference transformation functions

```
% attitude transformation per Etkin and Reid 1995
function T = Ta( TH) %(phi, theta, psi)
    phi = TH(1); theta = TH(2); psi = TH(3);
    T = [
        cos(theta)*cos(psi), sin(phi)*sin(theta)*cos(psi) - cos(phi)*sin(psi),
 cos(phi)*sin(theta)*cos(psi) + sin(phi)*sin(psi);
        cos(theta)*sin(psi), sin(phi)*sin(theta)*sin(psi) + cos(phi)*cos(psi),
 cos(phi)*sin(theta)*sin(psi) - sin(phi)*cos(psi);
        -sin(theta), sin(phi)*cos(theta), cos(phi)*cos(theta)
        ];
end

% euler angle rate transformation per Etkin and Reid 1995
function T = Tr( TH) %(phi, theta, psi)
    phi = TH(1); theta = TH(2); % psi = TH(3);
    T = [
        1, sin(phi)*tan(theta), cos(phi)*tan(theta);
        0, cos(phi), -sin(phi);
        0, sin(phi)*sec(theta), cos(phi)*sec(theta)
        ];
end
```

aerodynamic coefficient functions

```
function C = Cdrag_fcn( alpha3, disc)
%     Cdrag = interp1( disc.aoa_data, disc.Cdrag_data, alpha3, 'spline');
%     C = disc.CD0 + Cdrag * (alpha3 - disc.alpha0)^2;
    C = disc.CD0 + disc.CDa * (alpha3 - disc.alpha0)^2;
end
```

```
function C = Cside_fcn( r3, vinf)
    C = -r3 * vinf * 0;                          % <-- 0 == no side force
end

function C = Clift_fcn( alpha3, disc)
%     Clift = interp1( disc.aoa_data, disc.Clift_data, alpha3, 'spline');
%     C = disc.CL0 + Clift * alpha3;
    C = disc.CL0 + disc.CLa * alpha3;
end

function C = CL_fcn( p3, vinf, disc)
    C = disc.CLp * p3 * disc.c / (2 * vinf);
end

function C = CM_fcn( alpha, q3, vinf, disc)
%     CM = interp1( disc.aoa_data, disc.CM_data, alpha, 'spline');
%     C = disc.CM0 + CM * alpha;
    C = disc.CM0 + disc.CMa * alpha;
    C = C + (disc.CMq * q3 * disc.c / (2 * vinf));
end

function C = CN_fcn( r3, vinf, disc)
    C = disc.CNr * r3 * disc.c / (2 * vinf);
end
```

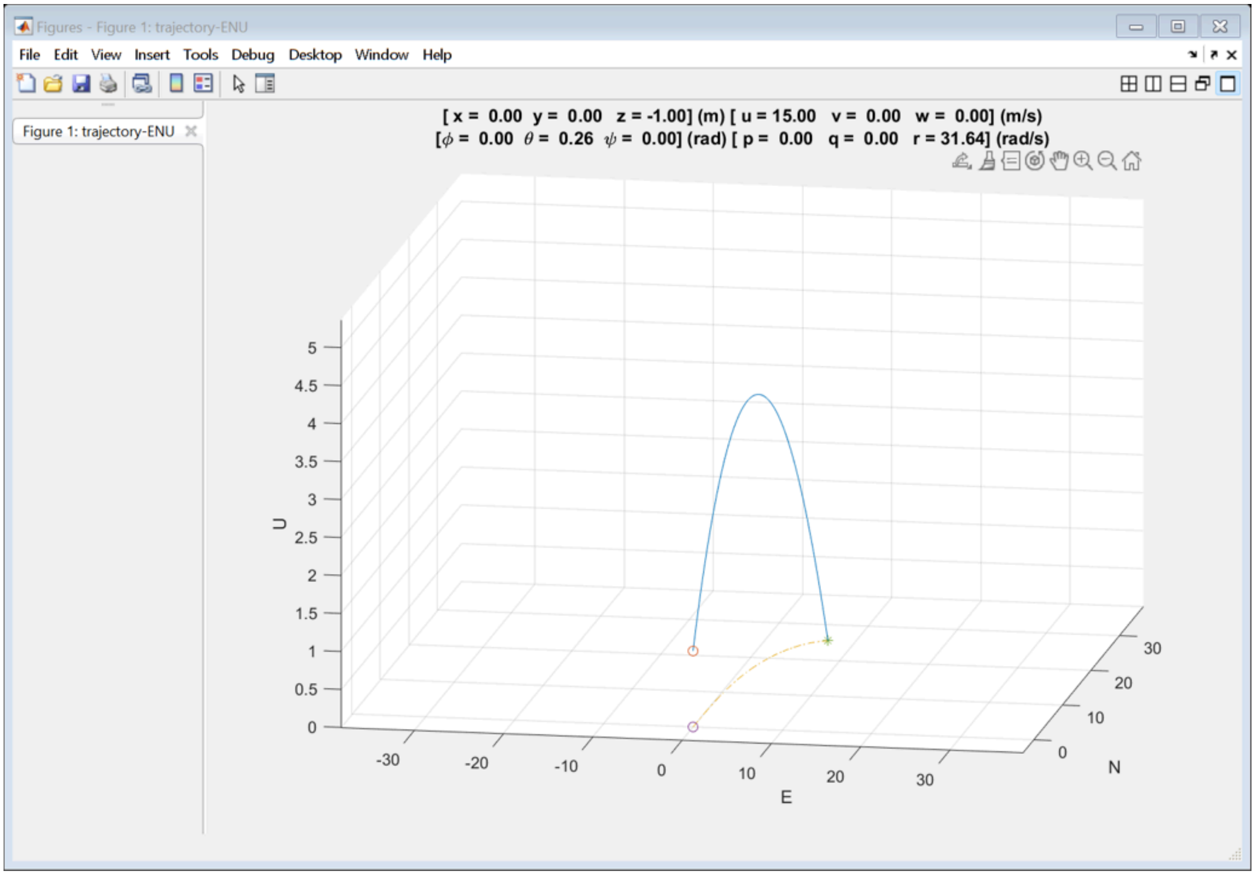
plot function

```
function mk_plots( t, y, y0)

title_str = sprintf( '[%4s %5.2f %4s %5.2f  %4s %5.2f] (m) [%4s %5.2f  %4s
 %5.2f  %4s %5.2f] (m/s)\n[%4s %5.2f  %4s %5.2f  %4s %5.2f] (rad) [%4s %5.2f
 %4s %5.2f  %4s %5.2f] (rad/s)\n', ...
'x =', y0(1), 'y =', y0(2), 'z =', y0(3), ...
'u =', y0(7), 'v =', y0(8), 'w =', y0(9), ...
'\phi =', y0(4), '\theta =', y0(5), '\psi =', y0(6), ...
'p =', y0(10), 'q =', y0(11), 'r =', y0(12) );

C = evalin( 'base', 'C');
```

trajectory

```
new_formatted_fig( 'trajectory-ENU');
hold on;
plot3( y(:,2), y(:,1), y(:,3).*-1);
plot3( y0(2), y0(1), -y0(3), 'o');
plot3( y(:,2), y(:,1), y(:,3).*0, '-.');
plot3( y0(2), y0(1), 0, 'o');
plot3( y(end,2), y(end,1), y(end,3).*0, '*');
hold off;
xlabel( 'E'); ylabel( 'N'); zlabel( 'U');
xy_lim = 1.5*max( max( abs(max(y(:,1))), abs(min(y(:,1)))),
 max( abs(max(y(:,2))), abs(min(y(:,2)))));
z_lim = 1.5*max( abs(max(y(:,3))), abs(min(y(:,3))));
```

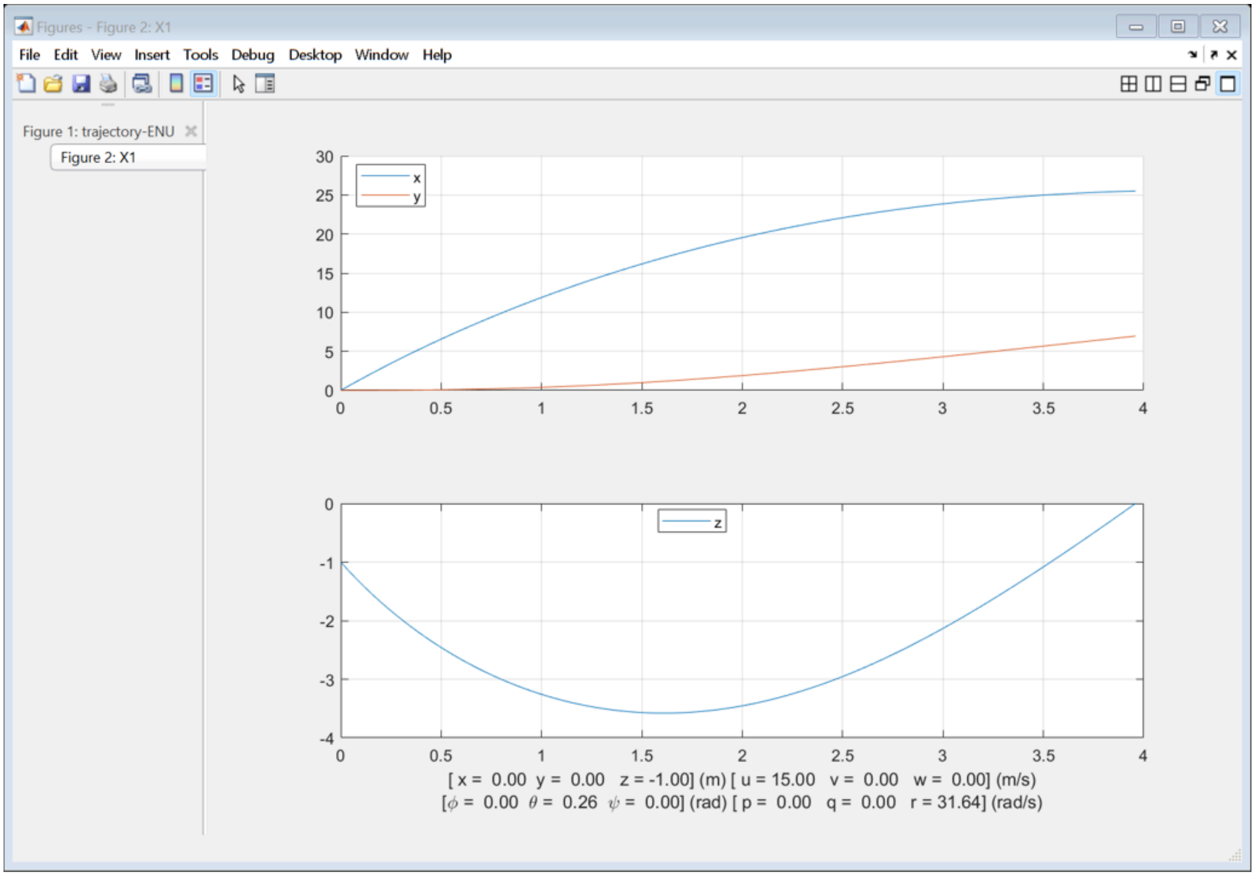
```
xlim( [-1 1] .* xy_lim);
ylim( [-0.1 1] .* xy_lim);
zlim( [0 1] .* z_lim);
grid on
title( title_str);
view( 10, 20);
snapnow;
```


Figures - Figure 1: trajectory-ENU
File Edit View Insert Tools Debug Desktop Window Help
Figure 1: trajectory-ENU
[x = 0.00 y = 0.00 z = -1.00] (m) [u = 15.00 v = 0.00 w = 0.00] (m/s)
[ϕ = 0.00 θ = 0.26 ψ = 0.00] (rad) [p = 0.00 q = 0.00 r = 31.64] (rad/s)
U
0
0.5
1
1.5
2
2.5
3
3.5
4
4.5
5
E
-30
-20
-10
0
10
20
30
N
0
10
20
30

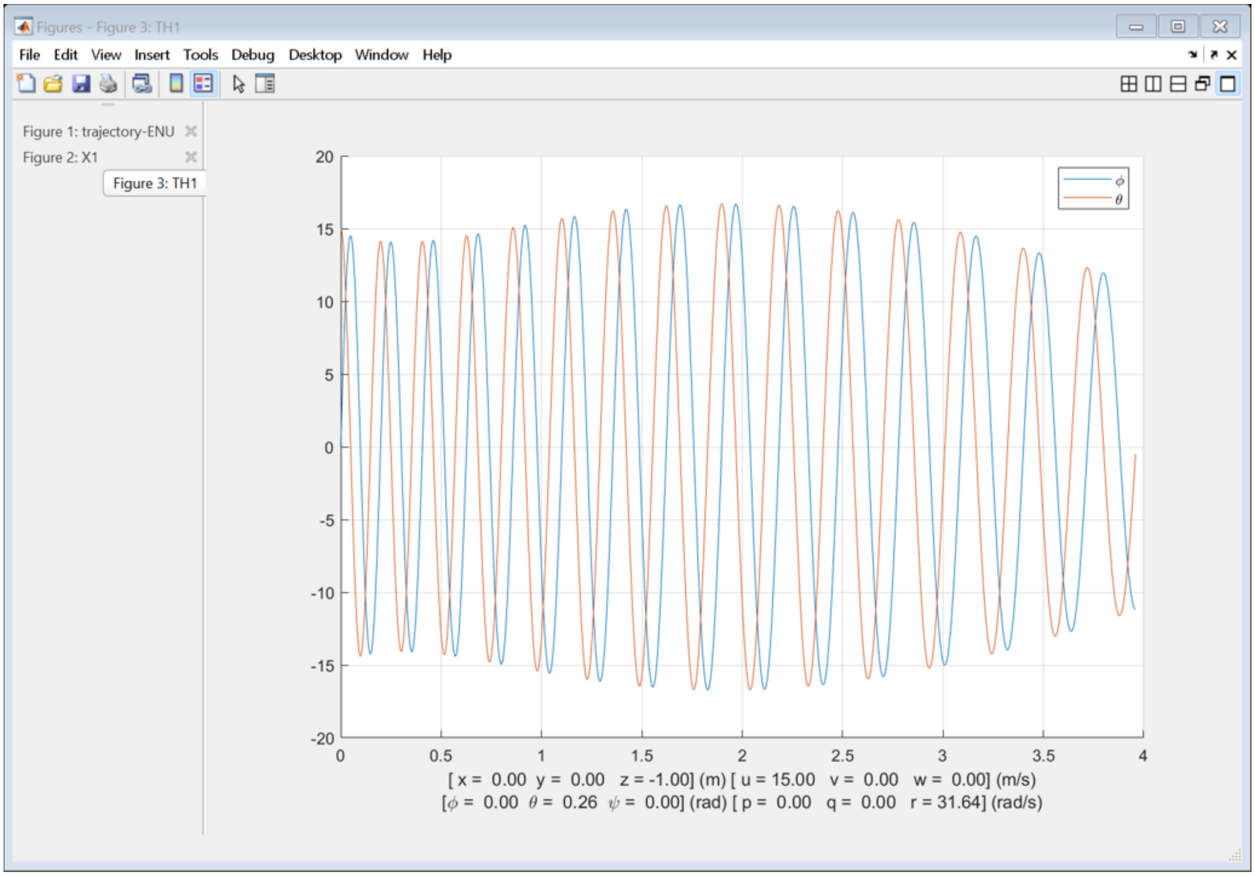
states

```
new_formatted_fig( 'X1');
subplot(211);
hold on;
plot( t, y(:,1));
plot( t, y(:,2));
hold off;
legend( 'x', 'y', 'Location','best');
grid on;
subplot(212);
plot( t, y(:,3));
legend( 'z', 'Location','best');
grid on;
xlabel( title_str);
snapnow;
```

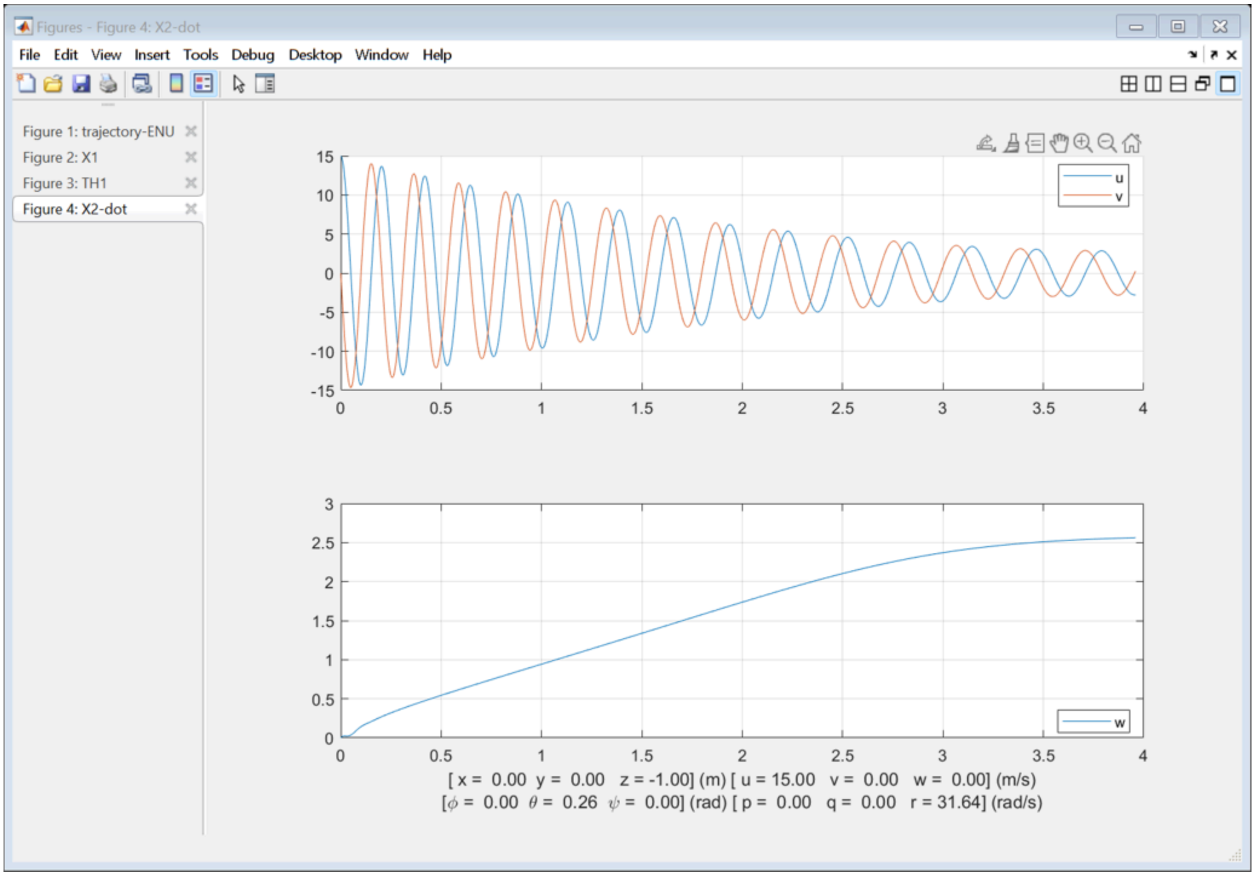
```
new_formatted_fig( 'TH1');
hold on;
plot( t, y(:,4) .* C.RAD2DEG);
plot( t, y(:,5) .* C.RAD2DEG);
legend( '\phi', '\theta', 'Location','best');
hold off;
grid on;
xlabel( title_str);
snapnow;

new_formatted_fig( 'X2-dot');
subplot(211);
hold on;
plot( t, y(:,7));
plot( t, y(:,8));
hold off;
legend( 'u', 'v', 'Location','best');
grid on;
subplot(212);
plot( t, y(:,9));
legend( 'w', 'Location','best');
grid on;
xlabel( title_str);
snapnow;

new_formatted_fig( 'TH2-dot');
subplot(211);
hold on;
plot( t, y(:,10));
plot( t, y(:,11));
hold off;
legend( '\phi\prime', '\theta\prime', 'Location','best');
grid on;
subplot(212);
plot( t, y(:,12));
legend( '\psi\prime', 'Location','best');
grid on;
xlabel( title_str);
snapnow;
```



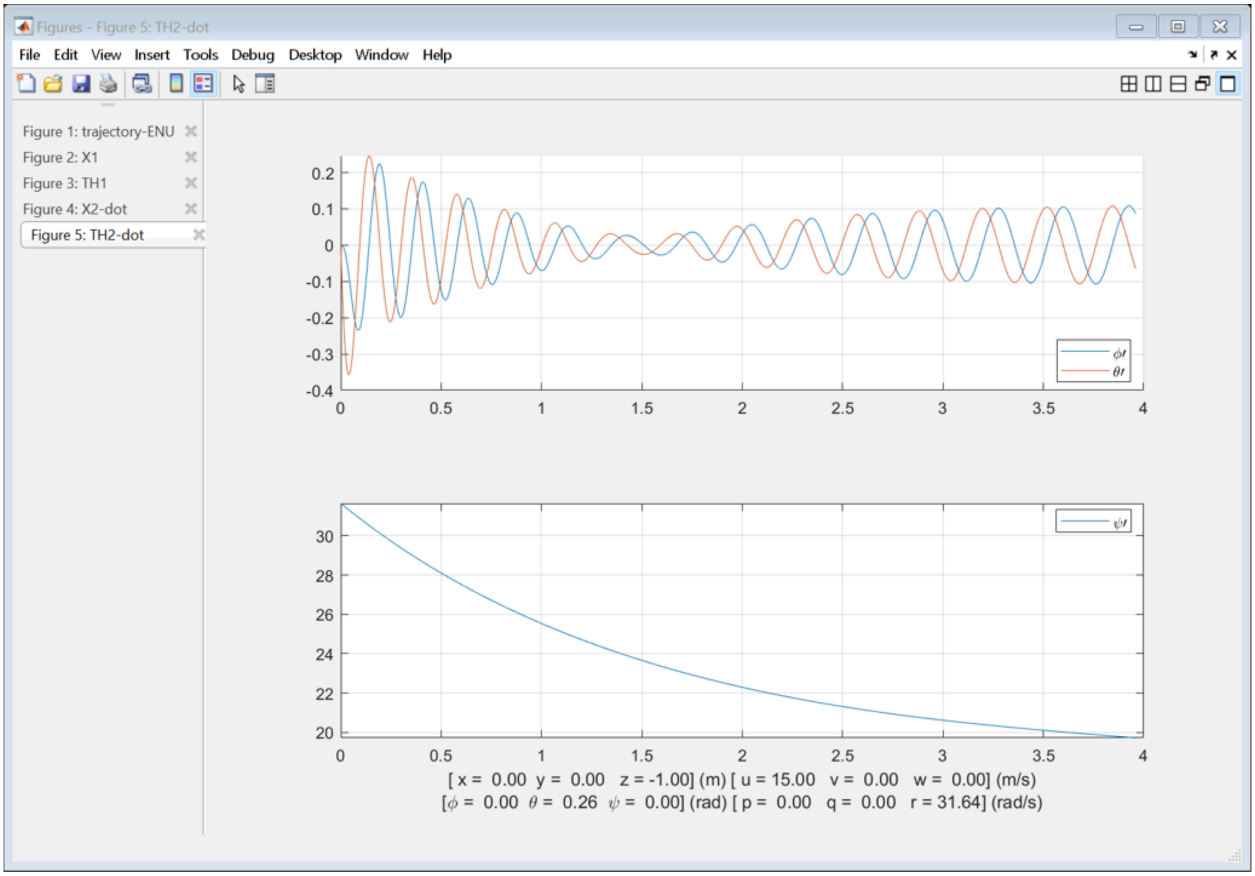
```
[ x =  0.00  y =  0.00   z = -1.00] (m) [ u = 15.00   v =  0.00   w =  0.00] (m/s)
[φ =  0.00  θ =  0.26  ψ =  0.00] (rad) [ p =  0.00   q =  0.00   r = 31.64] (rad/s)
```



[x = 0.00 y = 0.00 z = -1.00] (m) [u = 15.00 v = 0.00 w = 0.00] (m/s)
[ϕ = 0.00 θ = 0.26 ψ = 0.00] (rad) [p = 0.00 q = 0.00 r = 31.64] (rad/s)



[x = 0.00 y = 0.00 z = -1.00] (m) [u = 15.00 v = 0.00 w = 0.00] (m/s)
[ϕ = 0.00 θ = 0.26 ψ = 0.00] (rad) [p = 0.00 q = 0.00 r = 31.64] (rad/s)



[x = 0.00 y = 0.00 z = -1.00] (m) [u = 15.00 v = 0.00 w = 0.00] (m/s)
[ϕ = 0.00 θ = 0.26 ψ = 0.00] (rad) [p = 0.00 q = 0.00 r = 31.64] (rad/s)

```
end
```

Published with MATLAB® R2021b