
```

function A = stat_od_proj_A(state, consts)
%stat_od_proj_A Calculate A matrix for Stat OD project
fcnPrintQueue(mfilename('fullpath')) % Add this code to code app

% Init A, set up local vars
A = zeros(consts.state_len);
x = state(1);
y = state(2);
z = state(3);
xdot = state(4);
ydot = state(5);
zdot = state(6);
mu = state(7);
J2 = state(8);
Cd = state(9);

Re = consts.Re;
area = consts.area;
rho = consts.rho;
theta_dot = consts.theta_dot;
m = consts.m;

H = 88667; %m

% vars to reduce computations
x2 = x*x;
y2 = y*y;
z2 = z*z;
r = sqrt(x2+y2+z2);
sqrt_r = sqrt(r);
v = sqrt(xdot*xdot+ydot*ydot+zdot*zdot);
rel_wind_x = (xdot + theta_dot*y);
rel_wind_y = (ydot - theta_dot*x);
zdot2 = zdot*zdot;
rel_wind_mag = sqrt(rel_wind_x*rel_wind_x + rel_wind_y*rel_wind_y + zdot2);
Re2 = Re*Re;

rho0 = 3.614e-13; %kg/m3
r0 = 700000+6378136.3; %km
H = 88667.0; %km

% Only a few elements are populated
A(1,4) = 1;
A(2,5) = 1;
A(3,6) = 1;

A(4,1) = (3*mu*x^2)/(r*r*r*r*r) - ...
mu/(r*r*r) + ...
(3*J2*Re2*mu*((5*z2)/(r*r) - 1))/(2*(r)^(5)) - ...
(15*J2*Re2*mu*x2*z2)/(r)^(9) - ...
(15*J2*Re2*mu*x2*((5*z2)/(r*r) - 1))/(2*(r)^(2)) + ...
(Cd*area*theta_dot*rho*rel_wind_x*rel_wind_y)/(2*m*rel_wind_mag) + ...

```

```

        (Cd*area*x*rho*(xdot + theta_dot*y)*rel_wind_mag)/(2*H*m*r);
A1 = [
A2 = [
A3 = [
A4 = [          (3*mu*x^2)/(x^2 + y^2 + z^2)^(5/2) - mu/(x^2 + y^2 + z^2)^(3/2)
A5 = [  (3*mu*x*y)/(x^2 + y^2 + z^2)^(5/2) + (Cd*area*rho0*theta_dot*exp((r0 - (x^2
A6 = [
A(1,:) = A1;
A(2,:) = A2;
A(3,:) = A3;
A(4,:) = A4;
A(5,:) = A5;
A(6,:) = A6;
A4(4);

```

Published with MATLAB® R2013b