Table of Contents

..... 1
Constants 1
Extract Parameters 1
Add Spacecraft MOI to Params 2
Assemble Initial Conditions for Spacecraft 2
Simulate 2
Post Process Data 2
Creat Plots 2


```
function results = SpacecraftSim(params)

% Usage: results = SpacecraftSim(params)
%
% Description: Function takes in a struct of simulation parameters and
% returns a struct containing all of the results of the simulation
%
% Inputs:
%   params  -  struct of parameters for spacecraft, planet, and
 general
%              simaultion parameters
%
% Outputs:
%   results  -  struct of simulation results
%
```

Constants

```
deg2rad = pi/180;
rad2deg = 1/deg2rad;  %#ok<NASGU>
```

Extract Parameters

Sim

```
nOrbits = params.nOrbits;
absTol = params.absTol;
relTol = params.relTol;

% Earth
mu = params.Earth.mu_e;
R = params.Earth.Rmean;  %#ok<NASGU>

% Orbit
a = params.sc.sma;
e = params.sc.ecc;
inc = params.sc.inc*deg2rad;

% Spacecraft Initial Conditions
```

```
AttType = params.sc.Attitude_Type;
omega0 = params.sc.omega0;
```

Add Spacecraft MOI to Params

```
params.sc.IB_b = scMOI(params.sc);
```

Assemble Initial Conditions for Spacecraft

```
OMEGA = 0; % assume perigee at equator with RAAN = 0;
omega = 0;
theta = 0;

[r,v,~] = orbEl2rv(a, e, theta, OMEGA, omega, inc, mu); % transform
                                                        % orbital
 elements
                                                        % to pos. and
 vel.

if strcmp(AttType,'quaternion')
    x0 = [r; v; params.sc.qba0; omega0];
elseif strcmp(AttType,'DCM')
    x0 = [r; v; params.sc.Cba0(:); omega0];
else
    error('Incorrect attitude type!\n');
end
```

Simulate

```
options = odeset('AbsTol',absTol,'RelTol',relTol);

T = 2*pi*sqrt(a^3/mu);
tspan = [0 nOrbits*T];

[tout,xout] = ode45(@(t,x) CoupledDyn(t,x,params),tspan,x0,options);
```

Post Process Data

```
xout = xout';
Post_Process_v2;
results.tout = tout;
results.xout = xout;
results.E = E;
results.eulerAngs = eulerAngs;
results.constraint = constraint;
results.x0 = x0;
```

Creat Plots

```
q = zeros(4,length(tout));
```

```
if strcmp(AttType,'DCM')
   for lv1 = 1:length(tout)
       q(:,lv1) = DCM2Quat(reshape(xout(7:15,lv1),[3 3]));
   end
else
    q = xout(7:10,:);  %#ok<NASGU>
end

Plotter_v2;
```

Published with MATLAB® R2019b