

---

## Table of Contents

.....	1
Constants .....	1
Extract Parameters .....	1
Add SC MOI to Params .....	1
Initial Conditions .....	1
Simulate .....	2
Post Process Data .....	2
Create Plots .....	2
Animate Attitude .....	2

```
function results = AttitudeSim(params)

% Usage: results = AttitudeSim(params)
%
% Written by Garrett Ailts
%
% Description: Function that simulates the free rotation of a continuous
% rigid body given an initial angular velocity.
%
% Inputs:
%   params - struct of simulation parameters
%
% Outputs:
%   results - struct of simulation results
%
```

## Constants

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

## Extract Parameters

```
Sim

absTol = params.absTol;
relTol = params.relTol;
AttType = params.sc.Attitude_Type;
omega0 = params.sc.omega0;
tf = params.tsim;
```

## Add SC MOI to Params

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

## Initial Conditions

```
if strcmp(AttType, 'quaternion')
```

---

```

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

```

## Simulate

```

options = odeset('AbsTol',absTol,'RelTol',relTol);
tspan = [0 tf];

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

```

## Post Process Data

```

xout = xout';
Post_ProcessAt;
results.t = t;
results.xout = xout;
results.T = T;
results.x0 = x0;
results.eulerAngs = eulerAngs;
results.normDet = normDet;

```

## Create Plots

```

q = zeros(4,length(T));
if strcmp(AttType, 'DCM')
    for i = 1:length(T)
        q(:,i) = DCM2Quat(reshape(xout(1:9,i),[3 3]));
    end
else
    q = xout(1:4,:);
end
PlotterAt;

```

## Animtate Attitude

```

l = params.sc.height;
w = params.sc.side; d = w;
q = q';
AttitudeAnim('figs/AttAnim',q,l,w,d,t,10);

end

```

*Published with MATLAB® R2019b*