Table of Contents


```
function IB_b = scMOI(sc)

% Usage: I = scMOI(sc)
%
% Written by Garrett Ailts
%
% Description: Function takes in a struct of spacecraft and returns
 the
% Moment of Inertia matrix relative to the spacecraft center of mass
% resolved in the body frame
%
% Inputs:
%   sc  -  struct containing stuctural, sensor, and actuator
%          information for the spacecraft
%
% Outputs:
%   IB_b  -  Moment of Inertia matrix relative to the center of mass
%             resolved in the body frame
%
% A more advanced version of the MOI calculation can be done by simply
% packaging the body complexity in the volumetric density, which will
 be a
% function of the location on the body with respect to point z.
```

Extract Parameters

```
side = sc.side;
height = sc.height;
rtuna = sc.rtuna;
htuna = sc.htuna;
rhosc_tuna = sc.rhosc_tuna;
rtuna_mini = sc.rtuna_mini;
htuna_mini = sc.htuna_mini;
sigma = sc.sigma;
```

Assemble Relative Vectors wrt Pt. z

```
%(located on top left corner of cuboid section)
rciz = zeros(3,6);
rciz(:,1) = [height/2 side/2 side/2]';
```

```
rciz(:,2) = [-htuna/2 rhosc_tuna rhosc_tuna]';
rciz(:,3) = [-htuna/2 side-rhosc_tuna rhosc_tuna]';
rciz(:,4) = [-htuna/2 rhosc_tuna side-rhosc_tuna]';
rciz(:,5) = [-htuna/2 side-rhosc_tuna side-rhosc_tuna]';
rciz(:,6) = [-htuna_mini/2 side/2 side/2]';
```

Get Sub-Body Masses

```
mBi = zeros(1,6);
mBi(1) = sigma*side^2*height;
for i = 2:5
    mBi(i) = sigma*pi*rtuna^2*htuna;
end
mBi(6) = sigma*pi*rtuna_mini^2*htuna_mini;
```

Calculate rcz

distance to spacecraft center of mass wrt pt. z

```
rcz = (mBi(1)*rciz(:,1)+mBi(2)*rciz(:,1)+mBi(2)*rciz(:,1)+mBi(2)* ...
                    rciz(:,1)+mBi(2)*rciz(:,1)+mBi(6)*rciz(:,1))/
sum(mBi);
```

Calculate Body Vectors Relative to Center of Mass

```
rhoci = rciz-rcz;
```

Calculate Principal MOI's for Each Sub-Body

```
IBi_b = zeros(3,3,6);
IBi_b(:,:,1) =
 mBi(1)*diag([side^2+side^2,height^2+side^2,height^2+ ...

 side^2])/12;
for i = 2:5
    IBi_b(:,:,i) = mBi(i)*diag([rtuna^2/2,(3*rtuna^2+htuna^2)/12, ...

 (3*rtuna^2+htuna^2)/12]);
end

IBi_b(:,:,6) = mBi(6)*diag([rtuna_mini^2/2,
(3*rtuna_mini^2+htuna_mini^2)/12, ...

 (3*rtuna_mini^2+htuna_mini^2)/12]);
```

Calculate Spacecraft MOI via Parallel Axis Theorem

```
IB_b = zeros(3,3);
```

```
for i = 1:6
    rhocross = crossMatrix(rhoci(:,i));
    IB_b = IB_b+IBi_b(:,:,i)-mBi(i)*rhocross*rhocross;
end

end
```

Published with MATLAB® R2019b