

---

## Table of Contents

function [] = get_EOM() .....	1
Initialise variables .....	1
Finding mn_Jvn_JvnT .....	1
Finding Jwn_In_JwnT .....	2
Finding Matrix M .....	3
Finding Matrix G .....	3
Finding Matrix B and C .....	3
Finding EOM .....	4

## function [] = get\_EOM()

```
function [] = get_EOM()

clc
close all
verbose = 0;

signpost(verbose, 'Start: get_EOM()')
```

## Initialise variables

```
signpost(verbose, 'Variable init')

s=tf('s');           %Laplace variable

%Symbolic Variables
syms g temp
syms theta_1 dtheta_1 ddtheta_1
syms theta_2 dtheta_2 ddtheta_2
syms theta_3 dtheta_3 ddtheta_3

syms l_1 l_2 l_3
syms m_1 m_2 m_3

syms Ixx1 Ixx2 Ixx3
syms Iyy1 Iyy2 Iyy3
syms Izz1 Izz2 Izz3

syms T1 T2 T3

%Angles (relative)
a = theta_1;
b = a + theta_2;
c = b + theta_3;
```

## Finding mn\_Jvn\_JvnT

```
signpost(verbose, 'Finding mn_Jvn_JvnT')
```

---

```

% matrix for Jv1
e11 = -1*l_1*sin(a);
e12 = 0;
e13 = 0;
e21 = l_1*sin(a);
e22 = 0;
e23 = 0;
e31 = 0;
e32 = 0;
e33 = 0;

Jv1 = [e11 e12 e13; e21 e22 e23; e31 e32 e33];
Jv1T = transpose(Jv1);

m1_Jv1_Jv1T = m_1*(Jv1T*Jv1);

% matrix for Jv2
e11 = -1*l_1*sin(a) + -1*l_2*sin(b);
e12 = -1*l_2*sin(b);
e13 = 0;
e21 = l_1*cos(a) + l_2*cos(b);
e22 = l_2*cos(b);
e23 = 0;
e31 = 0;
e32 = 0;
e33 = 0;

Jv2 = [e11 e12 e13; e21 e22 e23; e31 e32 e33];
Jv2T = transpose(Jv2);

m2_Jv2_Jv2T = m_2*(Jv2T*Jv2);

% matrix for Jv3
e11 = -1*l_1*sin(a) + -1*l_2*sin(b) + -1*l_3*sin(c);
e12 = -1*l_2*sin(b) + -1*l_3*sin(c);
e13 = -1*l_3*sin(c);
e21 = l_1*cos(a) + l_2*cos(b) + l_3*cos(c);
e22 = l_2*cos(b) + l_3*cos(c);
e23 = l_3*cos(c);
e31 = 0;
e32 = 0;
e33 = 0;

Jv3 = [e11 e12 e13; e21 e22 e23; e31 e32 e33];
Jv3T = transpose(Jv3);

m3_Jv3_Jv3T = m_3*(Jv3T*Jv3);

```

## Finding Jwn\_In\_JwnT

```
signpost(verbose, 'Finding Jwn_In_JwnT')
```

---

```

Jw1 = [0 0 0; 0 0 0; 1 0 0];
Jw2 = [0 0 0; 0 0 0; 1 1 0];
Jw3 = [0 0 0; 0 0 0; 1 1 1];

Jw1T = transpose(Jw1);
Jw2T = transpose(Jw2);
Jw3T = transpose(Jw3);

I1 = [Ixx1 0 0; 0 Iyy1 0; 0 0 Izz1];
I2 = [Ixx2 0 0; 0 Iyy2 0; 0 0 Izz2];
I3 = [Ixx3 0 0; 0 Iyy3 0; 0 0 Izz3];

Jw1_I1_Jw1T = Jw1T*I1*Jw1;
Jw2_I2_Jw2T = Jw2T*I2*Jw2;
Jw3_I3_Jw3T = Jw3T*I3*Jw3;

```

## Finding Matrix M

```

signpost(verbose, 'Finding Matrix M')

M = Jw1_I1_Jw1T + Jw2_I2_Jw2T + Jw3_I3_Jw3T + m1_Jv1_Jv1T +
    m2_Jv2_Jv2T + m3_Jv3_Jv3T;

```

## Finding Matrix G

```

signpost(verbose, 'Finding Matrix G')

g1 = [0; m_1*g; 0];
g2 = [0; m_2*g; 0];
g3 = [0; m_3*g; 0];

Jv1_g1 = (Jv1)*(g1);
Jv2_g2 = (Jv2)*(g2);
Jv3_g3 = (Jv3)*(g3);

G = -1*(Jv1_g1) + -1*(Jv2_g2) + -1*(Jv2_g2);

```

## Finding Matrix B and C

```

signpost(verbose, 'Finding Matrix B and C')

C = [temp temp temp; temp temp temp; temp temp temp];
c = [temp temp temp];

theta_list = [theta_1, theta_2, theta_3];

for i = 1:3
    for j = 1:3
        for k = 1:3

            Mij = M(i,j);
            Mik = M(i,k);

```

---

```

        Mjk = M(j,k);

        theta_a = theta_list(k);
        theta_b = theta_list(j);
        theta_c = theta_list(i);

        dMij = diff(Mij, theta_a);
        dMik = diff(Mik, theta_b);
        dMjk = diff(Mjk, theta_c);

        c(k) = dMij + dMik + dMjk;

    end

    C(i,j) = c(1) + c(2) + c(3);
end
end

```

## Finding EOM

```

signpost(verbose, 'Finding EOM')

T = [T1;T2;T3];

theta    = [theta_1;   theta_2;   theta_3];
dtheta   = [dtheta_1;  dtheta_2;  dtheta_3];
ddtheta  = [ddtheta_1; ddtheta_2; ddtheta_3];

signpost(verbose, 'Done: get_EOM()')

```

*Published with MATLAB® R2015b*