Dynamic Formulation Using Recursive Newton Euler Algorithm - DeNOC

COM vectors:

```
d1_vec = formula([d1*cos(theta_1);d1*sin(theta_1);0]);
d2_vec = formula([d2*cos(theta_1+theta_2);d2*sin(theta_1+theta_2);0]);
d3_vec = formula([d3*cos(theta_1+theta_2+theta_3);d3*sin(theta_1+theta_2+theta_3);0]);
r1 = formula([(l1-d1)*cos(theta_1);(l1-d1)*sin(theta_1);0]);
r2 = formula([(l2-d2)*cos(theta_1+theta_2);(l2-d2)*sin(theta_1+theta_2);0]);
```

B matrix:

```
p1 = [e1;cross(e1,d1_vec)];
p2 = [e2;cross(e2,d2_vec)];
p3 = [e3;cross(e3,d3_vec)];
B21 = [Id ZO;vec_cross(-r1-d2_vec) Id];
B32 = [Id ZO;vec_cross(-r2-d3_vec) Id];
B31 = B32*B21;
B = [ZO_1 ZO_1 ZO_1;B21 ZO_1 ZO_1;ZO_1 B32 ZO_1];
```

N_l and N_d matrices:

```
%DeNOC Matrices
Nd = [p1 ZO_c ZO_c;ZO_c p2 ZO_c;ZO_c ZO_c p3];
Nl = [Id_1 ZO_1 ZO_1;B21 Id_1 ZO_1;B31 B32 Id_1];
```

Wrench Matrix and Final equation

Angular Velocity matrix:

```
W1 = [vec_cross(ang1) ZO;ZO ZO];
W2 = [vec_cross(ang2) ZO;ZO ZO];
W3 = [vec_cross(ang3) ZO;ZO ZO];
W = [W1 ZO_1 ZO_1;ZO_1 W2 ZO_1;ZO_1 ZO_1 W3];
```

External wrench:

```
%External Wrench
W1_e = [ZO_c2;m1*g_f];
W2_e = [ZO_c2;m2*g_f];
W3_e = [ZO_c2;m3*g_f];
W_e = [W1_e;W2_e;W3_e];
```

Inertia Matrix:

```
I_1 = [0 0 0;0 0 0;0 0 I1];
I_2 = [0 0 0;0 0 0;0 0 I2];
I_3 = [0 0 0;0 0 0;0 0 I3];
M1 = [I_1 ZO;ZO m1*Id];
M2 = [I_2 ZO;ZO m2*Id];
M3 = [I_2 ZO;ZO m3*Id];
M = [M1 ZO_1 ZO_1;ZO_1 M2 ZO_1;ZO_1 ZO_1 M3];
```

Final equation:

```
eqn = (M*N*q_dd)+(M*N_d*q_d)+(W*M*N*q_d)-W_e;
final_eqn = formula(transpose(N)*eqn);

eqn_theta1 = final_eqn(1);
eqn_theta2 = final_eqn(2);
eqn_theta3 = final_eqn(3);
```

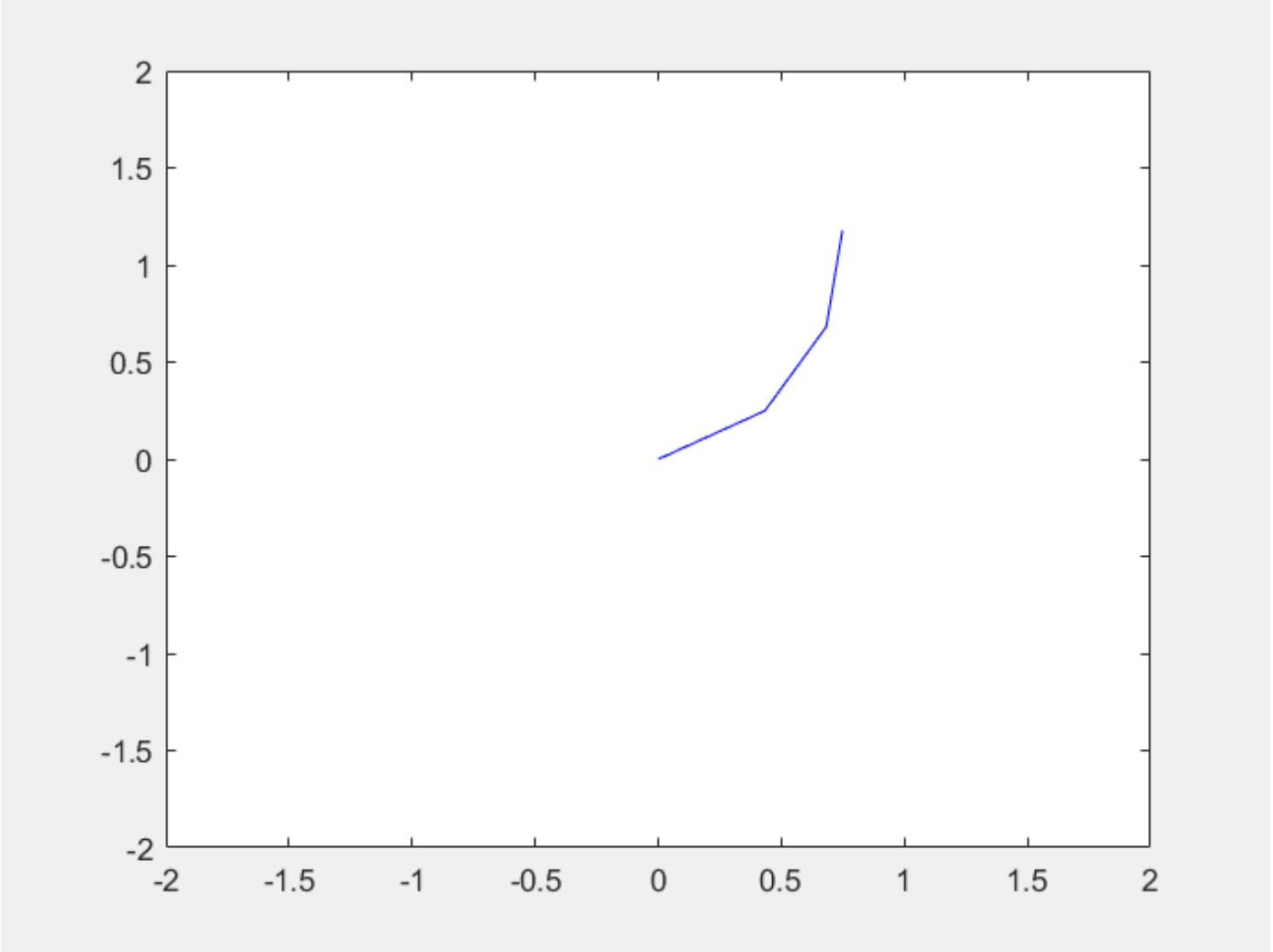
Mass matrix:

```
M1 = simplify(formula([(diff(eqn_theta1,a)) (diff(eqn_theta1,b)) (diff(eqn_theta1,c))]));
M2 = simplify(formula([(diff(eqn_theta2,a)) (diff(eqn_theta2,b)) (diff(eqn_theta2,c))]));
M3 = simplify(formula([(diff(eqn_theta3,a)) (diff(eqn_theta3,b)) (diff(eqn_theta3,c))]));
M = [M1;M2;M3];
```

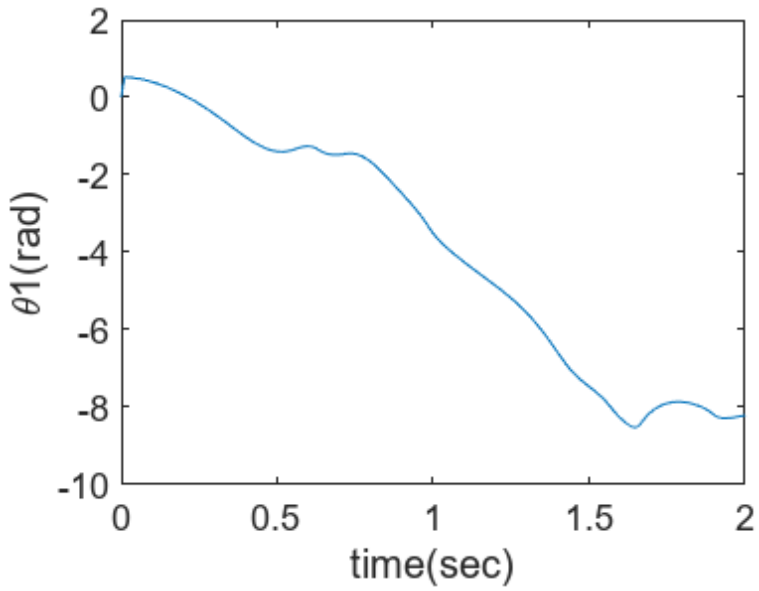
Coriolis Force matrix:

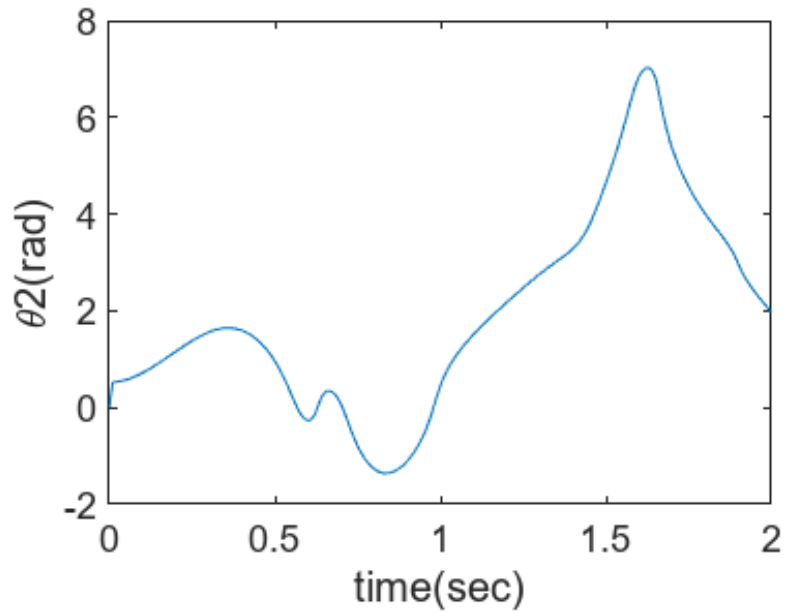
```
for i=1:3
    for j=1:3
        for k=1:3
            temp = simplify(formula(0.5*((diff(M(i,j),q(k,1))) + ( ...
                diff(M(i,k),q(j,1))) - (diff(M(k,j),q(i,1))))*q_dot(k,1)));
            C_matrix(i,j) = C_matrix(i,j) + temp;
        end
    end
end
```

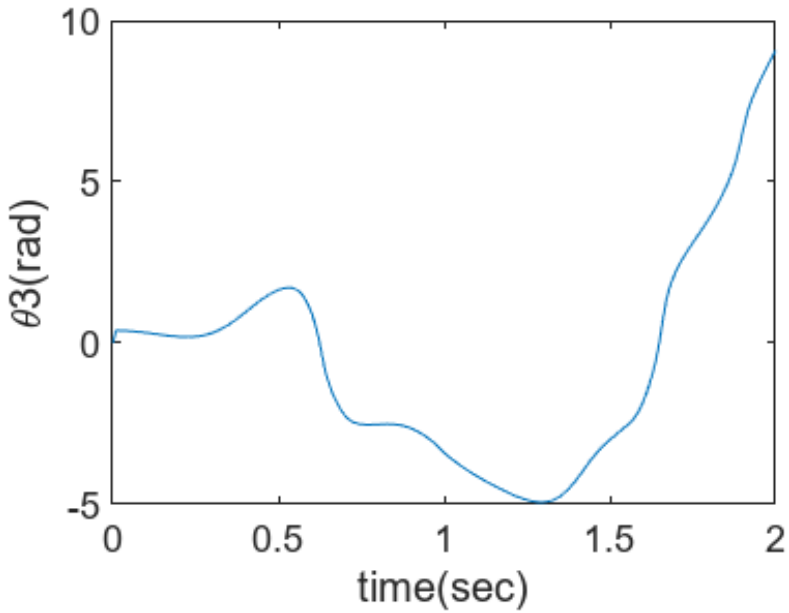
Simulation of 3R system under Free-Fall

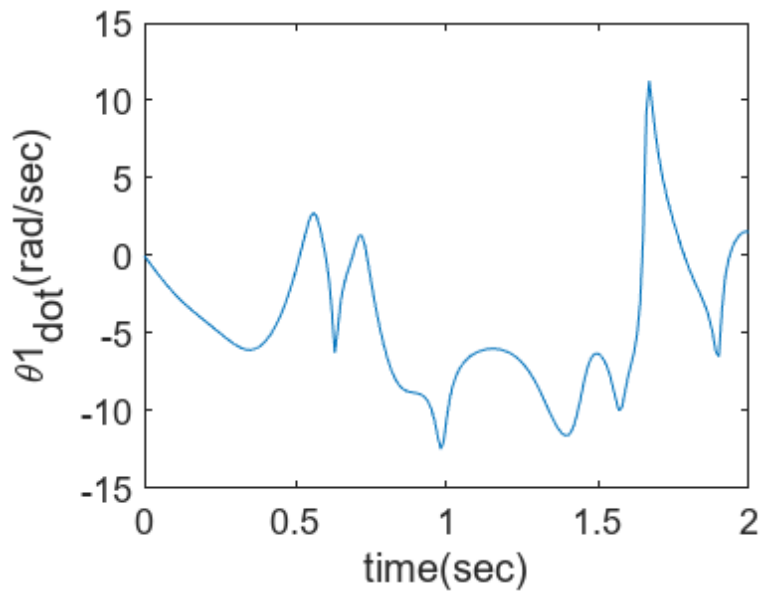


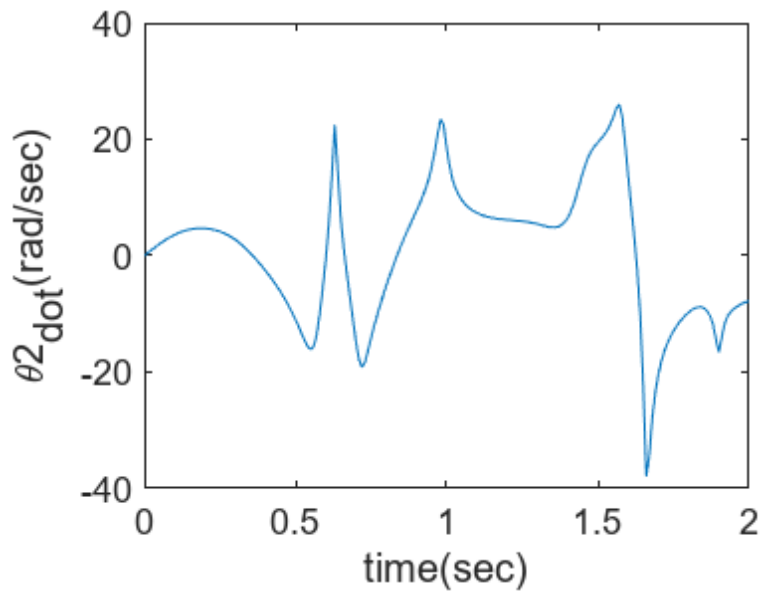
Simulation Results

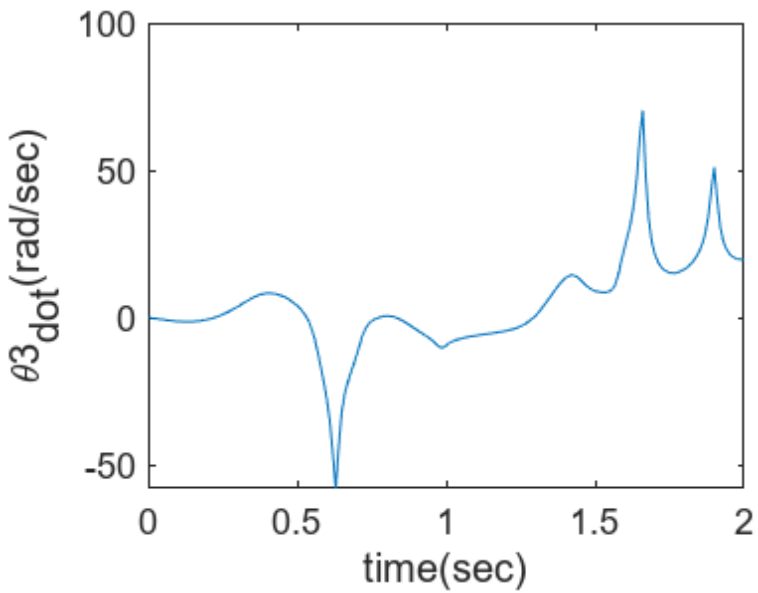












Inverse Dynamics : 3 Bar Mechanism

- a_1=0.5m
- a_2=0.5m
- a_3=0.5m

- Initial and Given conditions :
  - $\theta_1 = \frac{pi}{6}$ *and* $\dot{\theta_1} = 1\frac{rad}{sec}$
  - $\theta_2 = \frac{pi}{6}$ $and \dot{\theta_2} = 1\frac{rad}{sec}$
  - $\theta_3 = \frac{pi}{8}$ and $\dot{\theta_3} = 1\frac{rad}{sec}$