

```

%{
Newton Euler Dynamics Algorithm

Calculation of velocities and accelerations from 1->n
Calculation of forces and torques from n->1
%}

clc
clear all

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%% PROBLEM 1 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

syms theta_1
syms theta_2

syms L_1
syms L_2

% a alpha d theta (degrees)
dh_table = [0 0 0 theta_1; L_1 90 0 theta_2; L_2 0 0 0];
syms g
v_dot_0 = [0 0 g].';
syms m1
syms m2
masses = {m1, m2};
inertias = {zeros(3), zeros(3)};

P_c_i = {[L_1 0 0].', [L_2 0 0].'};

[omegas, omega_dots, v_dots, v_dots_cg, F, N, f, n] = newton_euler(dh_table, v_dot_0, masses, inertias, P_c_i);

tau_1 = n{1}(3);
tau_2 = n{2}(3);
disp('***** PROBLEM 1 *****');
disp('tau 1:');
disp(tau_1);

disp('tau 2:');
disp(tau_2);

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%% PROBLEM 2 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
clear all

syms theta_1
syms theta_2

syms L_1
syms L_2

syms Ixx1
syms Iyy1
syms Izz1

syms d2

% a alpha d theta (degrees)
dh_table = [0 0 0 theta_1; 0 -90 d2 0; 0 0 0 0];
syms g
v_dot_0 = [0 0 g].';
syms m1
syms m2
masses = {m1, m2};
inertias = {[Ixx1 0 0; 0 Iyy1 0; 0 0 Izz1], zeros(3)};

P_c_i = {[0 0 0].', [0 0 0].'};

[omegas, omega_dots, v_dots, v_dots_cg, F, N, f, n] = newton_euler(dh_table, v_dot_0, masses, inertias, P_c_i);

tau_1 = n{1}(3);
tau_2 = f{2}(3);

disp('***** PROBLEM 2 *****');
disp('***** Newton Euler Method: *****');
disp('tau 1:');
disp(tau_1);

disp('tau 2:');
disp(tau_2);

syms d_2_dot
syms d_2_double_dot
syms theta_dot_1

disp('***** PROBLEM 3 *****');
disp('***** Cartesian Method: *****');

j = [-d2 0; 0 1];
j_inv_t = (j.')^-1;
M = [m2*d2 + Izz1 0; 0 m2];

```

```
Mx = j_inv_t * M * j_inv_t;  
V = [2*m2*theta_dot_1*d2*d_2_dot; -m2*d2 * theta_dot_1^2];  
Vx = j_inv_t * (V-M*j^-1*[-d_2_dot 0; 0 0]*[theta_dot_1 d_2_dot].');
```

***** PROBLEM 1 *****

```
tau 1:  
L_1*m2*(L_1*t_double_dot1 + L_2*(c2*t_double_dot1 - s2*t_dot_1*t_dot_2) - L_2*s2*t_dot_1*t_dot_2) + L_1^2*m1*t_double_dot1 + L_2*c2*m2*(L_1*t_double_dot1 + L_2*(c2*  
  
tau 2:  
L_2*m2*(L_2*t_double_dot2 + c2*g + L_1*s2*t_dot_1^2 + L_2*c2*s2*t_dot_1^2)
```

***** PROBLEM 2 *****
***** Newton Euler Method: *****

```
tau 1:  
Izz1*t_double_dot1 + d2*m2*(d2*t_double_dot1 + 2*d_dot_2*t_dot_1)
```

```
tau 2:  
m2*(- d2*t_dot_1^2 + d_double_dot2)
```

***** PROBLEM 3 *****
***** Cartesian Method: *****
