INNOPOLIS UNIVERSITY

"Project Task4"

DYNAMICS OF NON-LINEAR ROBOTIC SYSTEMS

By:

Ahmed Mohsen Mohamed Abdelkhalek Elsayed Ali

Date

2-Oct-2021

Home Task 4

This report show the result of code implementation for Home Task 4. There are some code files attached with this report that contain the following:

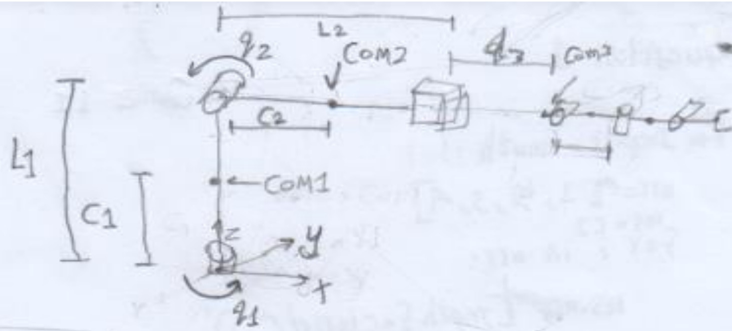
- Dynamics_Symbolic.m show the derivation for M(q), C(q,dq), g used in the dynamics equation. It shows them in symbolic
- Dynamics.m show the values for the above three matrices based in the user input values.

Derivation

1) Center of mass equations

In order to find position equation for each center of mass, its CoM for joint i was assumed to be at distance ci from the local origin. Then using DH, we were able to determine the equations for each CoM as follow:

robot arm:



Center of mass 1:

$x = 0$

$y = 0$

$z = C_1$

Center of mass 2:

$x = -C_2 \cos(q_2) \cos(q_1)$

$y = C_2 \cos(q_2) \sin(q_1)$

$z = l_1 + C_2 \sin(q_2)$

Center of mass 3:

$x = (l_2 + q_3) \cos(q_2) \cos(q_1)$

$y = (l_2 + q_3) \cos(q_2) \sin(q_1)$

$z = l_1 + (l_2 + q_3) \sin(q_2)$

Center of mass 4:

$x = (l_2 + q_3) \cos(q_2) \cos(q_1) - C_4 [C_1 S_2 S_4 - C_1 C_2 C_4]$

$y = (l_2 + q_3) C_2 S_1 - C_4 [S_1 S_2 S_4 - C_2 C_4 S_1]$

$z = (l_2 + q_3)(S_2 + C_4 [C_2 S_4 + C_4 S_2] + d_1$

Center of mass 5:

$x = (l_2 + q_3) c_1 C_2 - C_5 [S_1 S_5 + C_5 [C_1 S_2 S_4 - C_1 C_2 C_4]] + L_4 [C_1 C_2 C_4 - C_1 S_2 S_4]$

$y = C_5 [C_1 S_5 - C_5 [S_1 S_2 S_4 - C_2 C_4 S_1]] + (l_2 + q_3) C_2 S_1 + L_4 [C_2 C_4 S_1 - S_1 S_2 S_4]$

$z = l_1 + (l_2 + q_3) S_2 + L_4 (C_2 S_4 + C_4 S_2) + C_5 C_5 [C_2 S_4 + C_4 S_2]$

COM6:

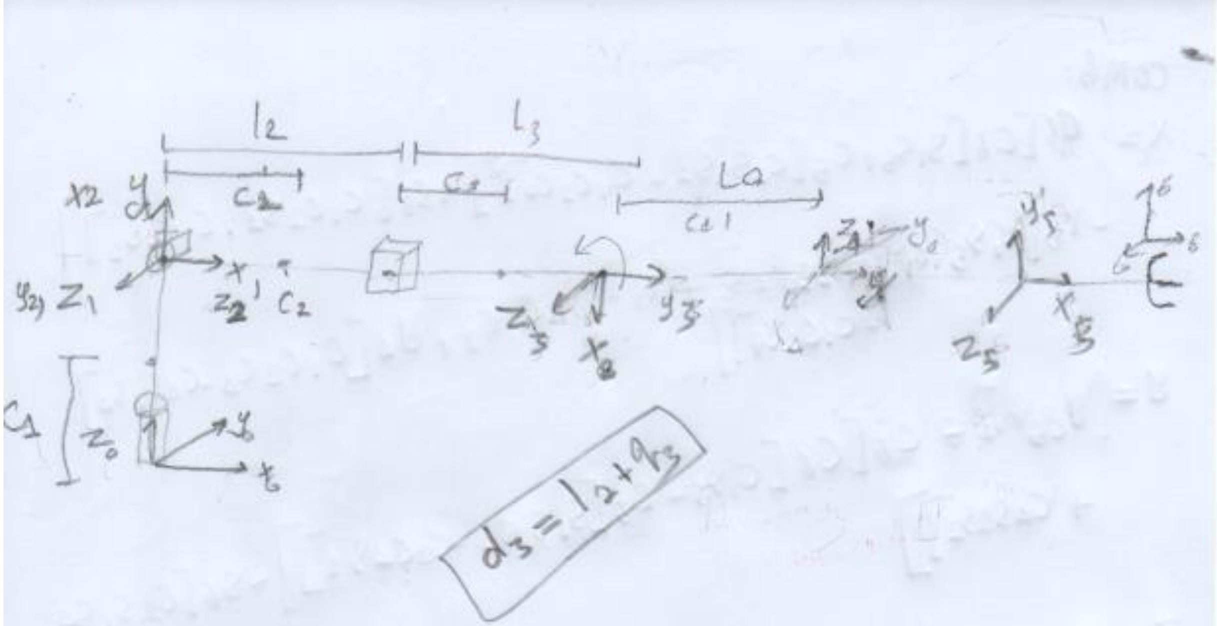
$$x_c = a_6\left[c_6\left[s_1 s_5 + c_5\left[c_1 s_2 s_4 - c_1 c_2 c_4\right]\right] + s_6\left[c_1 c_2 s_4 + c_1 c_4 s_2\right]\right]$$

$$- L_5\left[s_1 s_5 + c_1\left[c_1 s_2 s_4 - c_1 c_2 c_4\right]\right] + (l_2 + q_3)\, c_1 c_2 + d_4\left[c_1 c_2 c_4 - c_1 s_2 s_4\right]$$

$$y = y_{c_5} - a_6\left[c_6\left[c_1 s_5 - c_5\left[s_1 s_2 s_4 - c_2 c_4 s_1\right]\right] - s_6\left[c_2 s_1 s_4 + c_4 s_1 s_2\right]\right]$$

$$z = z_{c_5} - a_6\left[s_6\left[c_2 c_4 - s_2 s_4\right] + c_5 c_6\left[c_2 s_4 + c_4 s_2\right]\right]$$

This is the DH parameters used to calculate CoM equations:



	θ	d	a	α
1	θ_1	d_1	0	90
2	$\theta_2 + 90°$	0	0	90°
3	180°	$l_2 + l_3$	0	90
4	$\theta_4 + \pi$	0	d_4	-90°
5	θ_5'	0	d_5	-90°
6	θ_6	l_6	0	0

2) Mass matrix:

The mass matrix was calculated symbolic as shown in the following code snippet:

```
M_q_lin = m1*transpose(jac_lin_1)*jac_lin_1+m2*transpose(jac_lin_2)*jac_lin_2+m3*transpose(jac_lin_3)*jac_lin_3+m4*transpose(jac_lin_4)*jac_li
M_q_ang = transpose(jac_ang_1)* rot_z1(1:3,1:3) *I* transpose(rot_z1(1:3,1:3))*jac_ang_1+transpose(jac_ang_2)* rot_z2(1:3,1:3) *I* transpose(r
M_q= M_q_lin + M_q_ang;
```

To view the full matrix: write M_q in the command line

3) Corriolis matrix:

This is the equations used to calculate Corriolis matrix

```
C = sym(zeros(6));
for i=1:6
    for j=1:6
        eval(strcat('m',string(i),string(j),' = M_q(',string(i),',',string(j),');')) %mij = M_q(i,j)
        eval(strcat('c',string(i),string(j),' = sym(0);'))%cij = sym(0)
        for k=1:6
            eval(strcat('m',string(i),string(k),' = M_q(',string(i),',',string(k),');'))%mik = M_q(i,k)
            eval(strcat('m',string(j),string(k),' = M_q(',string(j),',',string(k),');'))%mjk = M_q(j,k)

            eval(strcat('c',string(i),string(j),string(k),' = 0.5*(diff(m',string(i),string(j),',theta_',string(k),')+diff(m',string(i),stri
            eval(strcat('c',string(i),string(j),' = c',string(i),string(j),' + c',string(i),string(j),string(k),'*dq',string(k),';'))%cij =
        end
        eval(strcat('Corlios(',string(i),',',string(j),') = c',string(i),string(j),';'))%C(i,j) = cij
    end
end
```

To view the full matrix: write C in the command line.

4) Gravity matrix:

```
%%%%%%%%%%%%%%%%%%%%%CALCULATING    Gravity%%%%%%%%%%%%%%%%%%%%%
g1 = -transpose(jac_lin_1(:,1)) * m1 * g0 -transpose(jac_lin_2(:,1)) * m2 * g0-transpose(jac_lin_3(:,1)) * m3 * g0 - transpose(jac_lin_4(:,1))
g2 = -transpose(jac_lin_1(:,2)) * m1 * g0 -transpose(jac_lin_2(:,2)) * m2 * g0-transpose(jac_lin_3(:,2)) * m3 * g0 - transpose(jac_lin_4(:,2))
g3 = -transpose(jac_lin_1(:,3)) * m1 * g0 -transpose(jac_lin_2(:,3)) * m2 * g0-transpose(jac_lin_3(:,3)) * m3 * g0 - transpose(jac_lin_4(:,3))
g4 = -transpose(jac_lin_1(:,4)) * m1 * g0 -transpose(jac_lin_2(:,4)) * m2 * g0-transpose(jac_lin_3(:,4)) * m3 * g0 - transpose(jac_lin_4(:,4))
g5 = -transpose(jac_lin_1(:,5)) * m1 * g0 -transpose(jac_lin_2(:,5)) * m2 * g0-transpose(jac_lin_3(:,5)) * m3 * g0 - transpose(jac_lin_4(:,5))
g6 = -transpose(jac_lin_1(:,6)) * m1 * g0 -transpose(jac_lin_2(:,6)) * m2 * g0-transpose(jac_lin_3(:,6)) * m3 * g0 - transpose(jac_lin_4(:,6))
```

To view the full matrix: write g_final in the command line.

5) Dynamics Equation:

$$M(q)\begin{matrix} ddq1 \\ ddq2 \\ ddq3 \\ ddq4 \\ ddq5 \\ ddq6 \end{matrix} + C(q,dq)\begin{matrix} dq1 \\ dq2 \\ dq3 \\ dq4 \\ dq5 \\ dq6 \end{matrix} + g(q) = \tau$$

For the following parameters:

```
DH=[0 0 1 0 0 0];  %Enter the values for each joint parameter RRPRRR
L=[1 1 0 1 1 1];  %Robot lengthes
C=[.1 .1  .1 .1 .1]; %Location of each COM relative to the local origin
m = [10 10 10 10 10 10]; %mass of each link
dq=[2 ;2; 4 ;3; 1; 5];%joint velocity
ddq=[1 ;.5 ;1 ;.1 ;.5 ;.6];%joint acceleration
izz=1;  %izz valie
g_0=9.81; % gravity
```

The τ was equal to:

```
T =

 -735.4400
  198.2900
   40.0000
   74.4400
  -58.6200
    5.0200
```