STUDY ORIENTED PROJECT - REPORT (MF F266)
SUBMITTED BY - SHRUTI SHARMA
SUBMITTED TO - PROF B K ROUT

1. Introduction

1.1 Problem Statement

- Dynamic modelling of a 2-DOF RR manipulator with joint flexibility
- Design of a control system to eliminate the inaccuracy in assembly introduced due to this flexibility.

1.2 What is dynamic modelling

Dynamic modelling of a robotic manipulator refers to the mathematical modelling of it to depict the dynamic response of the manipulator due to input torques.

1.3 Why use a controller?

To eliminate this compliance, we need a feedback control system to adjust the input torque so that the manipulator is able to accurately follow the trajectory.

2. Dynamic Modeling

2.1 Assumptions

- Links are assumed to be rigid
- Joints are modelled as equivalent to a torsional spring
- The manipulator is assumed to be in a gravity-free environment.
- The kinetic energy of the rotor is solely due to rotation about an inertial axis.

2.2 Theoretical Modeling

System description - A 2-DOF RR manipulator arm, with one DC motor at each joint.

Q = Q1, Q3 = angle rotated by a motor at any time t wrt an axis perpendicular to their Centre of Mass.

Theta = Q2, Q4 = angle rotated by the links at any time t wrt an axis perpendicular their Centre of Mass.

According to Euler-Lagrangian formulation we have -

$$B(q)\ddot{q} + c(q,\dot{q}) + g(q) + K(q-\theta) = 0$$

$$B_m\ddot{\theta} + K(\theta - q) = \tau$$

Here, Bm = motor's moment of inertia = J ;

B = moment of inertia of each link;

T= torque applied at joints;
C = 0;
G = 0;

2.3 Algorithmic Modeling

Symbolic Math Toolbox is used to code the above equations as -

```
syms m1; % mass of the rotor
syms r; % radius of the rotor
syms m l; %lenth of the link and mass of the link
J = 0.5*m1*r^2; %moment of inertia of rotor
syms D; %moment of inertia of the link
D = 0.5*m*l^2;
syms k; %rigidity of the joint
syms t q1(t) q3(t); %actuator variables
syms q2(t) q4(t); %link variables
syms dq1 dq3 dq4 dq2; %derivative of joints
dq1 = diff(q1, t);
dq2 = diff(q2, t);
dq3 = diff(q3, t);
dq4 = diff(q4, t);
syms ddq1 ddq2 ddq3 ddq4 ;
ddq1 = diff(dq1,t);
ddq1 = diff(dq1,t);
ddq1 = diff(dq1,t);
ddq1 = diff(dq1,t);
```

```
E1(t) = J*ddq1 + k*(q1-q2) ;
E2(t) = J*ddq3 + k*(q3-q4) ;
E3(t) = D*ddq2 + k*(q2-q1) ;
E4(t) = D*ddq4 + k*(q4-q3) ;
```

```
syms T1 T2;
ode1 = diff(dq1) == T1 + k*(q2-q1)/J;
ode2 = diff(dq3) == T2 + k*(q4-q3)/J;
ode3 = diff(dq2) == k*(q1-q2)/D;
ode4 = diff(dq4) == k*(q3-q4)/D;
odes = [ode1 ode2 ode3 ode4];

S = dsolve(odes);
q1Sol(t) = S.q1
q2Sol(t) = S.q2
q3Sol(t) = S.q3
q4Sol(t) = S.q4
```

The obtained equations are as follows -

```
odes(t) =
```

$$\left(\frac{\partial^2}{\partial t^2}\, q_1(t) = T_1 - \frac{\sigma_2}{m_1 r^2} \quad \frac{\partial^2}{\partial t^2}\, q_3(t) = T_2 - \frac{\sigma_1}{m_1 r^2} \quad \frac{\partial^2}{\partial t^2}\, q_2(t) = \frac{\sigma_2}{l^2 m} \quad \frac{\partial^2}{\partial t^2}\, q_4(t) = \frac{\sigma_1}{l^2 m} \right)$$

where

$$\sigma_1 = 2\,k\,\left(q_3(t) - q_4(t)\right)$$

$$\sigma_2 = 2\,k\,\left(q_1(t) - q_2(t)\right)$$

2.4 Solutions

We use the “dsolve” command to solve the four set of equations, and the values of q1, q2, q3, q4 are obtained as follows -

```
q1Sol(t) =
```

$$\frac{l^2 m \left(C_1 - \frac{T_1 m_1 r^2 t^2}{2 l^2 m}\right)}{\sigma_4} + \frac{l^2 m\, t \left(C_3 + \frac{T_1 m_1 r^2 t}{l^2 m}\right)}{\sigma_4} + \frac{\sqrt{2}\, l^5 m^{5/2} \mathrm{e}^{\sigma_1} \left(C_5 - \frac{\sqrt{2}\, T_1 m_1{}^{3/2} r^3 \mathrm{e}^{-\sigma_1}}{\sigma_3}\right)}{\sigma_2} - \frac{\sqrt{2}\, l^5 m^{5/2} \mathrm{e}^{-\sigma_1} \left(C_6 + \frac{\sqrt{2}\, T_1 m_1{}^{3/2} r^3 \mathrm{e}^{\sigma_1}}{\sigma_3}\right)}{\sigma_2}$$

where

$$\sigma_1 = \frac{\sqrt{2}\,\sqrt{-k}\,t\,\sqrt{\sigma_4}}{l\,\sqrt{m}\,\sqrt{m_1}\,r}$$

$$\sigma_2 = 4\,\sqrt{-k}\,\sqrt{m_1}\,r\,\sigma_4{}^{3/2}$$

$$\sigma_3 = 2\,\sqrt{-k}\,l\,\sqrt{m}\,\sqrt{\sigma_4}$$

$$\sigma_4 = m\,l^2 + m_1 r^2$$

```
q2Sol(t) =
```

$$\frac{l^2 m \left(C_1 - \frac{T_1 m_1 r^2 t^2}{2 l^2 m}\right)}{\sigma_4} + \frac{l^2 m t \left(C_3 + \frac{T_1 m_1 r^2 t}{l^2 m}\right)}{\sigma_4} - \frac{\sqrt{2}\, l^3 m^{3/2} \sqrt{m_1}\, r\, e^{\sigma_1} \left(C_5 - \frac{\sqrt{2}\, T_1 m_1^{3/2} r^3 e^{-\sigma_1}}{\sigma_3}\right)}{\sigma_2} + \frac{\sqrt{2}\, l^3 m^{3/2} \sqrt{m_1}\, r\, e^{-\sigma_1} \left(C_6 + \frac{\sqrt{2}\, T_1 m_1^{3/}}{\sigma_3}\right.}{\sigma_2}$$

where

$$\sigma_1 = \frac{\sqrt{2}\, \sqrt{-k}\, t\, \sqrt{\sigma_4}}{l \sqrt{m}\, \sqrt{m_1}\, r}$$

$$\sigma_2 = 4 \sqrt{-k}\, \sigma_4^{3/2}$$

$$\sigma_3 = 2 \sqrt{-k}\, l \sqrt{m}\, \sqrt{\sigma_4}$$

$$\sigma_4 = m l^2 + m_1 r^2$$

```
q3Sol(t) =
```

$$\frac{l^2 m \left(C_2 - \frac{T_2 m_1 r^2 t^2}{2 l^2 m}\right)}{\sigma_4} + \frac{l^2 m t \left(C_4 + \frac{T_2 m_1 r^2 t}{l^2 m}\right)}{\sigma_4} + \frac{\sqrt{2}\, l^5 m^{5/2} e^{\sigma_1} \left(C_7 - \frac{\sqrt{2}\, T_2 m_1^{3/2} r^3 e^{-\sigma_1}}{\sigma_3}\right)}{\sigma_2} - \frac{\sqrt{2}\, l^5 m^{5/2} e^{-\sigma_1} \left(C_8 + \frac{\sqrt{2}\, T_2 m_1^{3/2} r^3 e^{\sigma_1}}{\sigma_3}\right)}{\sigma_2}$$

where

$$\sigma_1 = \frac{\sqrt{2}\, \sqrt{-k}\, t\, \sqrt{\sigma_4}}{l \sqrt{m}\, \sqrt{m_1}\, r}$$

$$\sigma_2 = 4 \sqrt{-k}\, \sqrt{m_1}\, r\, \sigma_4^{3/2}$$

$$\sigma_3 = 2 \sqrt{-k}\, l \sqrt{m}\, \sqrt{\sigma_4}$$

$$\sigma_4 = m l^2 + m_1 r^2$$

```
Sol(t) =
```

$$\frac{l^2 m \left(C_2 - \frac{T_2 m_1 r^2 t^2}{2 l^2 m}\right)}{\sigma_4} + \frac{l^2 m t \left(C_4 + \frac{T_2 m_1 r^2 t}{l^2 m}\right)}{\sigma_4} - \frac{\sqrt{2}\, l^3 m^{3/2} \sqrt{m_1}\, r\, e^{\sigma_1} \left(C_7 - \frac{\sqrt{2}\, T_2 m_1^{3/2} r^3 e^{-\sigma_1}}{\sigma_3}\right)}{\sigma_2} + \frac{\sqrt{2}\, l^3 m^{3/2} \sqrt{m_1}\, r\, e^{-\sigma_1} \left(C_8 + \frac{\sqrt{2}\, T_2 m_1^{3/2} r^3 e^{\sigma_1}}{\sigma_3}\right)}{\sigma_2}$$

where

$$\sigma_1 = \frac{\sqrt{2}\, \sqrt{-k}\, t\, \sqrt{\sigma_4}}{l \sqrt{m}\, \sqrt{m_1}\, r}$$

$$\sigma_2 = 4 \sqrt{-k}\, \sigma_4^{3/2}$$

$$\sigma_3 = 2 \sqrt{-k}\, l \sqrt{m}\, \sqrt{\sigma_4}$$

$$\sigma_4 = m l^2 + m_1 r^2$$

3. Controller Design

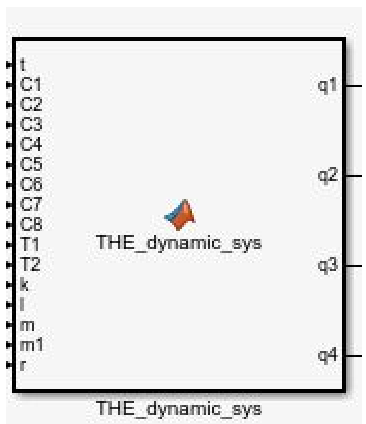
3.1 Algorithmic Modeling

Initially, the code is converted into a SIMULINK block as -

```
tf_i_should_create_SL_block = true;
if(true==tf_i_should_create_SL_block)
    MODEL_NAME = 'model_euler';
    if(4==exist(MODEL_NAME))
        close_system(MODEL_NAME, 0);
        delete(MODEL_NAME);
    end
    new_system(MODEL_NAME)
    open_system(MODEL_NAME)

    % Put BOTH equations into one block
    matlabFunctionBlock( [MODEL_NAME,'/THE_dynamic_sys'], q1Sol, q2Sol(t), q3Sol(t), q4Sol(t) , ...
                         'Optimize', false, ...
                         'Outputs', {'q1', 'q2', 'q3', 'q4'}   );
end
```

The block obtained is as follows -

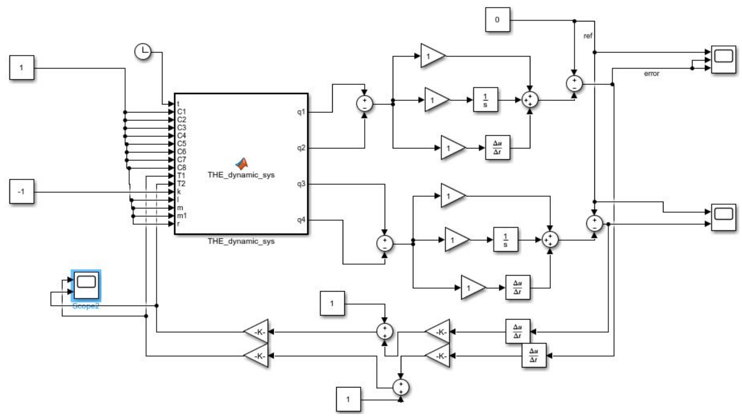


3.2 Theoretical Principles

Now, in order to remove the inaccuracy due to compliance, the torque should be increased with reference to the rate of change of error or the difference between (q1-q2) and (q3-q4) at each time instant. Here, the reference for the proportionality constant is derived from Maxon Geared output, Torque vs speed diagram.

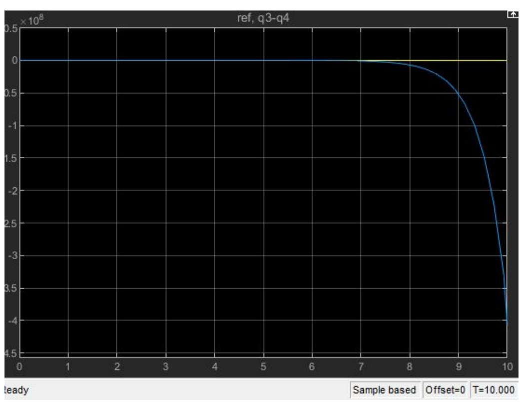
3.3. Design

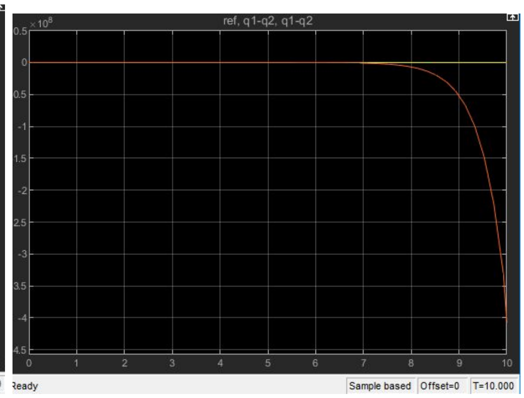
The controller is modelled using Proportional, integral and derivative blocks as follows -

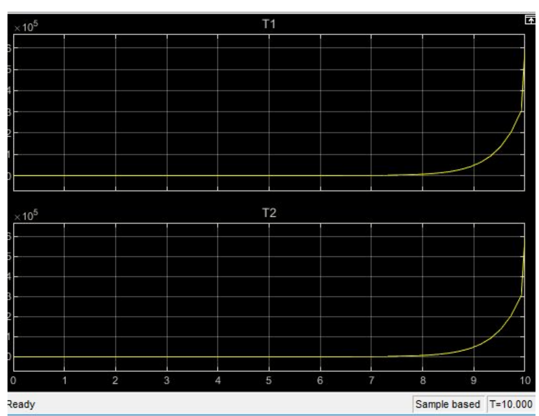


4. Results

The (q1-q2) and (q3-q4) vs time graph are shown below. The corresponding increase in input torque value is ilustrated later.







5. Conclusion

Here, to simplify things, the derivation is based on a number of assumptions. To check its implementation and tuning on a real system, we need to develop a more computationally intensive work with more experimental data.