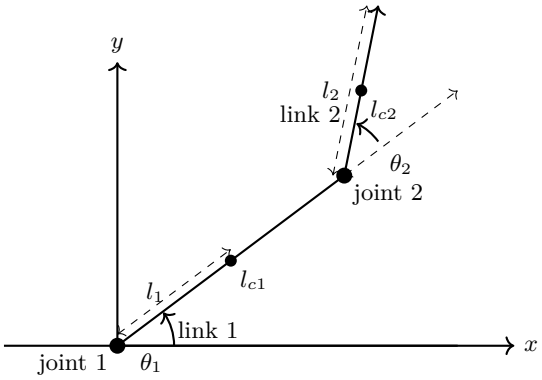
Report #3

Student GU JUN
ID number 6132230056

Problem Statement

Simulate the motion of a 2DOF open link mechanism under PID control. PID control is applied to active joints 1 and 2. Use appropriate values of geometrical and physical parameters of the manipulator.



PID Control

The control law for the PID controller is given by:

$$\tau = -K_p \cdot \mathrm{e} - K_d \cdot \omega - K_i \cdot \int \mathrm{e}\, dt$$

where:

- τ is the control input (torque) applied to the joints.
- K_p is the proportional gain.
- K_d is the derivative gain.
- K_i is the integral gain.
- e is the difference between the desired and actual position.
- ω is the angular velocity of the joint.
- integral is the accumulated error over time.

The PID controller aims to minimize the error by adjusting the control input τ based on the proportional, derivative, and integral terms. The proportional term $K_p \cdot \mathrm{e}$ corrects the error based on the current difference between the desired and actual position. The derivative term $K_d \cdot \omega$ provides a damping effect by considering the rate of change of the error, thus reducing overshoot and oscillations. The integral term $K_i \cdot \int \mathrm{e}\, dt$ accounts for the accumulated error over time, ensuring that the steady-state error is minimized.

Matlab Code

Based on the given example code and the PID control law, the following Matlab code was implemented to simulate the motion of the 2DOF open link mechanism under PID control.

```
function dotq = open_mechanism_2DOF_PID_params (t, q, robot, thetad, Kp, Ki, Kd)
% equation of motion of 2DOF open mechanism under PID control
  disp(num2str(t,"%8.6f"));

  theta = q(1:2);
  omega = q(3:4);
  integral = q(5:6);

  robot = robot.joint_angles(theta, omega);
  [inertia_matrix, torque_vector] = robot.inertia_matrix_and_torque_vector;

  dottheta = omega;
  error = theta - thetad;
  tau = -Kp.*error - Kd.*omega - Ki.*integral;
  dotomega = inertia_matrix \ (torque_vector + tau);

  dotq = [dottheta; dotomega; error];
end
```

To achieve PID control, I add the K_i term as the new parameter and add **integral** to the state description **q**. The error is calculated as the difference between the desired and actual position. The control input τ is then computed based on the PID control law. The derivative of the state vector **q** is returned as the output.

PID Parameters

After some trial and error, the following PID parameters were chosen to achieve stable and accurate control of the 2DOF open link mechanism:

K_p^T	(2.5, 2.5)
K_i^T	(0.8, 0.8)
K_d^T	(0.5, 0.5)

Table 1: PID Parameters

Simulation Results

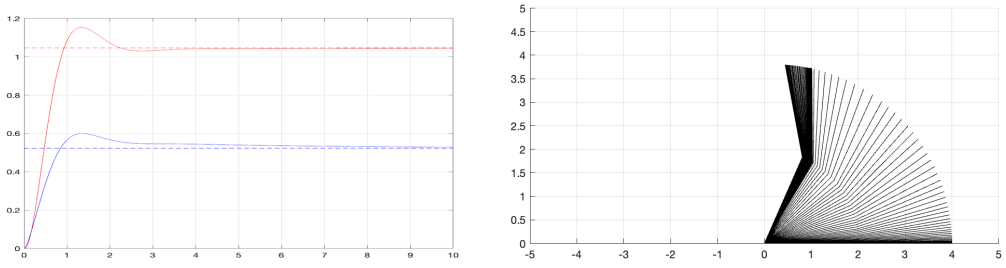


Figure 1: 2DOF Open Link Mechanism under PID Control, left side: joint angles, right side: mechanism drawing

As shown in Figure 1, the 2DOF open link mechanism under PID control exhibits stable and accurate motion than normal PD control. The joint angles converge to the desired values, and the mechanism follows the desired trajectory effectively. The chosen PID parameters provide a good balance between tracking performance and stability, resulting in smooth and precise control of the mechanism. A video demonstration of the 2DOF open link mechanism under PID control can be viewed at the following link:
`https://mushroomspace.notion.site/video-for-report-3-1436031a26338051985ec594ad084aa9`