2025

Controller Design Methods and Implementations for Simple 3-DoF PUMA Robot

CONTROL METHODS IN ROBOTICS

FATEMEH (NEGIN) HEIRAN

Shiraz - Iran

Contents

The Robot Model

3-Dof Articulated Robot Structure

In an articulated robot, joints are all revolute, similar to a human’s arm. They are the most common configuration for industrial robots. The selected articulated robot is the same as PUMA’s robot first part which yields to the robot end effector position (simplified PUMA). The robot configuration is showed in right side of fig. 1. Accordingly, the robot consists of three revolute joint which implied 3 Dof in task space.

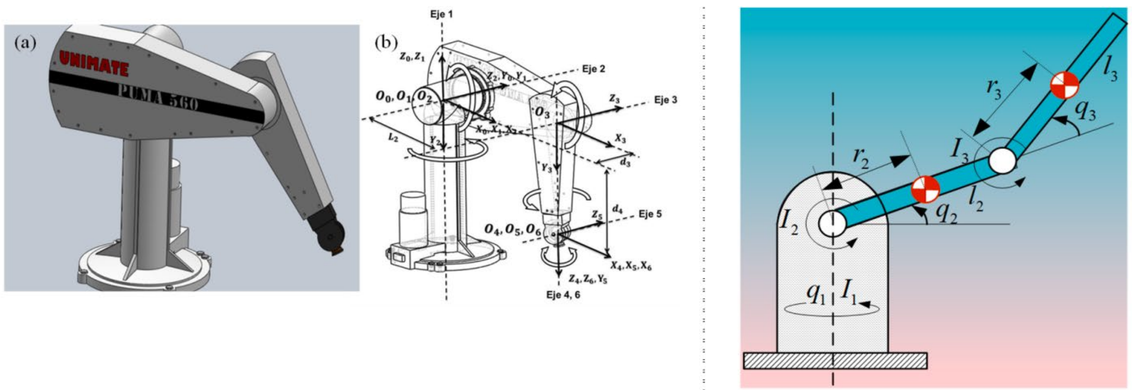

Figure 1- the left side picture is PUMA robot and the right side picture is the simplified PUMA robot [1]

Direct Kinematics

If we state the end effector coordinates of manipulator based on the angles of the joints, it means the forward kinematics. In other word, in forward kinematics, the measures of the joint space are available and we want to determine the measures of coordinate space. In reality, forward kinematics analyzing is a mapping from joint space to the coordinate space. The direct kinematics of this robot comes as follows [2]:

$$P_x = (l_2 \cos\theta_2 + l_3 \cos(\theta_2 + \theta_3)) * \sin\theta_1 \tag{1}$$

$$P_y = (l_2 \cos\theta_2 + l_3 \cos(\theta_2 + \theta_3)) * \cos\theta_1 \tag{2}$$

$$P_z = l_2 \sin\theta_2 + l_3 \sin(\theta_2 + \theta_3) + l_1 \tag{3}$$

Jacobian Matrix

The Jacobian matrix concluded from velocity analysis for the robot and is defined as [2]:

$$\dot{x} = \begin{bmatrix} C_1(l_3C_{23} + l_2C_2) & -S_1(l_3S_{23} + l_2S_2) & -l_3S_{23}S_1 \\ -S_1(l_3C_{23} + l_2C_2) & -C_1(l_3S_{23} + l_2S_2) & -l_3S_{23}C_1 \\ 0 & l_3C_{23} + l_2C_2 & l_3C_{23} \end{bmatrix} \dot{\theta} = J\dot{\theta} \tag{4}$$

Also in some analysis the derivative of Jacobian is needed, therefore it defined as:

$$\dot{j}_{11} = -S_1\dot{\theta}_1(l_3C_{23} + l_2C_2) + C_1(-l_3S_{23}(\dot{\theta}_2 + \dot{\theta}_3) - l_2S_2\dot{\theta}_2) \tag{5}$$

$$\dot{j}_{12} = -C_1\dot{\theta}_1(l_3S_{23} + l_2S_2) - S_1(l_3C_{23}(\dot{\theta}_2 + \dot{\theta}_3) + l_2C_2\dot{\theta}_2) \tag{6}$$

$$\dot{j}_{13} = -l_3C_{23}(\dot{\theta}_2 + \dot{\theta}_3)S_1 - l_3S_{23}C_1\dot{\theta}_1 \tag{7}$$

$$\dot{j}_{21} = -C_1\dot{\theta}_1(l_3C_{23} + l_2C_2) - S_1(-l_3S_{23}(\dot{\theta}_2 + \dot{\theta}_3) - l_2S_2\dot{\theta}_2) \tag{8}$$

$$\dot{j}_{22} = S_1\dot{\theta}_1(l_3S_{23} + l_2S_2) - C_1(l_3C_{23}(\dot{\theta}_2 + \dot{\theta}_3) + l_2C_2\dot{\theta}_2) \tag{9}$$

$$\dot{j}_{23} = -l_3C_{23}(\dot{\theta}_2 + \dot{\theta}_3)C_1 + l_3S_{23}S_1\dot{\theta}_1 \tag{10}$$

$$\dot{j}_{31} = 0 \tag{11}$$

$$\dot{j}_{32} = -l_3S_{23}(\dot{\theta}_2 + \dot{\theta}_3)C_1 - l_2S_2\dot{\theta}_2 \tag{12}$$

$$\dot{j}_{33} = -l_3S_{23}(\dot{\theta}_2 + \dot{\theta}_3) \tag{13}$$

Direct Dynamics

According to [1], in the simplified PUMA560, m_1, m_2, m_3 are the mass of the turntable, the boom and the arm, respectively. The mass of the turntable can be ignored, l_2, l_3 are the length of the boom and the length of the arm, respectively, r_2, r_3 are the distance from the center of mass of the boom to the axis of rotation and the distance from the center of mass of the arm to the axis of rotation. I_1, I_2, I_3 are the moment of inertia about the rotation of each axis. Let:

$$a_1 = m_2r_2^2 + m_3l_2^2,\ a_2 = m_3r_3^2,\ a_3 = m_3r_3l_2\ b_1 = (m_2r_2 + m_3l_2)g,\ b_2 = m_3r_3g$$

Establishing the dynamic equation based on the D-H rule:

$$M(\boldsymbol{q})\ddot{\boldsymbol{q}} + b(\boldsymbol{q},\dot{\boldsymbol{q}})\dot{\boldsymbol{q}} + f(\boldsymbol{q}) = \boldsymbol{\tau} \tag{14}$$

Where,

$$M(\boldsymbol{q}) = \begin{bmatrix} m_{11} & m_{12} & m_{13} \\ m_{21} & m_{22} & m_{23} \\ m_{31} & m_{32} & m_{33} \end{bmatrix} b(\boldsymbol{\theta},\dot{\boldsymbol{\theta}}) = \begin{bmatrix} b_{11} & b_{12} & b_{13} \\ b_{21} & b_{22} & b_{23} \\ b_{31} & b_{32} & b_{33} \end{bmatrix} f(\boldsymbol{q}) = \begin{bmatrix} 0 \\ b_1\cos q_2 + b_2\cos(q_2 + q_3) \\ b_2\cos(q_2 + q_3) \end{bmatrix}$$

$$m_{11} = a_1 \cos^2 q_2 + a_2 \cos^2 (q_2 + q_3) + 2a_3 \cos q_2 \cos(q_2 + q_3) + I_1,\ m_{22} = a_1 + a_2 + 2a_3 \cos q_3 + I_2$$

$$m_{33} = a_2 + I_3,\ m_{23} = m_{32} = a_2 + a_3 \cos q_3,\ m_{12} = m_{21} = m_{13} = m_{31} = 0$$

$$b_{11} = -\frac{1}{2} a_1 \dot{q}_2 \sin(2q_2) - \frac{1}{2} a_2 (\dot{q}_2 + \dot{q}_3) \sin(2q_2 + 2q_3) - a_3 \dot{q}_2 \sin(2q_2 + q_3) - a_3 \dot{q}_3 \cos q_2 \sin(q_2 + q_3)$$

$$b_{12} = -\frac{1}{2} a_1 \dot{q}_1 \sin(2q_2) - \frac{1}{2} a_2 \dot{q}_1 \sin(2q_2 + 2q_3) - a_3 \dot{q}_1 \sin(2q_2 + q_3)$$

$$b_{13} = -\frac{1}{2} a_2 \dot{q}_1 \sin(2q_2 + 2q_3) - a_3 \dot{q}_1 \cos q_2 \sin(q_2 + q_3),\ b_{21} = -b_{12},\ b_{22} = -a_3 \dot{q}_3 \sin q_3,$$

$$b_{23} = -a_3 (\dot{q}_2 + \dot{q}_3) \sin q_3,\ b_{31} = -b_{13},\ b_{32} = a_3 \dot{q}_2 \sin q_3,\ b_{33} = 0$$

Regressor Matrix

```
M = [cos(q(2))^2*ddq(1), cos(q(2) + q(3))^2*ddq(1), 2*cos(q(2))*cos(q(2) +
q(3))*ddq(1), ddq(1), 0, 0, 0, 0, 0;...
    ddq(2), ddq(2) + ddq(3), 2*cos(q(3))*ddq(2) + ddq(3)*cos(q(3)), 0, ddq(2),
0, 0, 0, 0;...
    0, ddq(2) + ddq(3), ddq(2)*cos(q(3)), 0, 0, ddq(3), 0, 0, 0];

C = [-0.5*dq(2)*sin(2*q(2))*dq(1) - 0.5*dq(2)*dq(1)*sin(2*q(2)), - 0.5*(dq(2) +
dq(3))*sin(2*q(2) + 2*q(3))*dq(1)...
    - 0.5*dq(2)*dq(1)*sin(2*q(2) + 2*q(3)) - 0.5*dq(3)*dq(1)*sin(2*q(2) +
2*q(3)), - dq(2)*sin(2*q(2) + q(3))*dq(1)...
    - dq(1)*dq(3)*cos(q(2))*sin(q(2) + q(3)) - dq(2)*dq(1)*sin(2*q(2) + q(3)) -
dq(3)*dq(1)*cos(q(2))*sin(q(2) + q(3))...
    , 0, 0, 0, 0, 0, 0;...
    0.5*dq(1)*dq(1)*sin(2*q(2)), 0.5*dq(1)*dq(1)*sin(2*q(2) + 2*q(3)),
dq(1)*dq(1)*sin(2*q(2) + q(3)) - dq(2)*dq(3)...
    *sin(q(3)) - dq(3)*(dq(2) + dq(3))*sin(q(3)), 0, 0, 0, 0, 0, 0;...
    0, 0.5*dq(1)*dq(1)*sin(2*q(2) + 2*q(3)), dq(1)*dq(1)*cos(q(2))*sin(q(2) +
q(3)) + dq(2)*dq(2)*sin(q(3)),...
    0, 0, 0, 0, 0, 0];

G = [0, 0, 0, 0, 0, 0, 0, 0, 0; ...
    0, 0, 0, 0, 0, 0, sin(q(2))*g/2, - sin(q(2) + q(3))*g/2, - sin(q(2))*g; ...
    0, 0, 0, 0, 0, 0, 0, -sin(q(2) + q(3))*g/2, 0];

Y = M + C + G;
```

Parameters Vector

```
theta = [a1; a2; a3; I1; I2; I3; m2*l2; m3*l3; m3*l2]
```

Controller Design and Simulation

Dynamics

Matrix Form of robot dynamics by considering the effect of the actuators:

$$M(q)\ddot{q} + C(q,\dot{q})\dot{q} + B\dot{q} + g(q) = u$$

Where,

- u is actuator effect.
- g is gravity
- B is the damping matrix and we will take B = 0 for simplicity.

Trajectory Planning

The trajectory planning for the robot joint angles has been take place with assuming the initial and final position, velocity and acceleration between time 0 to 5 second as follows.

$$x_1 = \begin{bmatrix}\theta_{1i} & \dot{\theta}_{1i} & \ddot{\theta}_{1i} & \theta_{1f} & \dot{\theta}_{1f} & \ddot{\theta}_{1f}\end{bmatrix} = [0, 0, pi/12, 2pi/3, 0, 0]$$

$$x_2 = \begin{bmatrix}\theta_{2i} & \dot{\theta}_{2i} & \ddot{\theta}_{2i} & \theta_{2f} & \dot{\theta}_{2f} & \ddot{\theta}_{2f}\end{bmatrix} = [0, 0, pi/12, pi/3, 0, 0]$$

$$x_3 = \begin{bmatrix}\theta_{3i} & \dot{\theta}_{3i} & \ddot{\theta}_{3i} & \theta_{3f} & \dot{\theta}_{3f} & \ddot{\theta}_{3f}\end{bmatrix} = [0, 0, pi/12, 5pi/6, 0, 0]$$

Which yields to the trajectory planning as follows:

```
theta1 = 0.1309*t^2 + 0.0147*t^3 - 0.0085*t^4 + 0.0007*t^5;
theta2 = 0.1309*t^2 - 0.0188*t^3 - 0.001*t^4 + 0.0002*t^5;
theta3 = 0.1309*t^2 + 0.0314*t^3 - 0.0123*t^4 + 0.001*t^5;
```

- *Hint 1: there is a m file named trajectoryPlanning to obtain the trajectory.*

Assumptions

The robot parameters are considered as follows:

```
m1 = 1;
m2 = 1;
m3 = 1;

l1 = 1;
l2 = 1;
l3 = 1;

r1 = l1/2;
r2 = l2/2;
r3 = l3/2;

R1 = 0.05;
R2 = 0.05;
```

```
R3 = 0.05;
g = 9.81;
```

PD Controller

A **Proportional-Derivative (PD) controller** is a type of feedback control system commonly used in robotics to ensure smooth and accurate motion. It adjusts the robot's output (like motor torque or speed) based on two factors: the **proportional term**, which reacts to the current error (difference between desired and actual position), and the **derivative term**, which predicts future error by measuring the rate of change of the error. By combining these two, the PD controller helps the robot reach its target quickly and with reduced overshoot or oscillation, making it especially useful for tasks requiring fast and stable responses, such as joint position or orientation control.

PD

In this controller the terms including gravity (g) were omitted. Fig. 2 show the controller designed in Simulink.

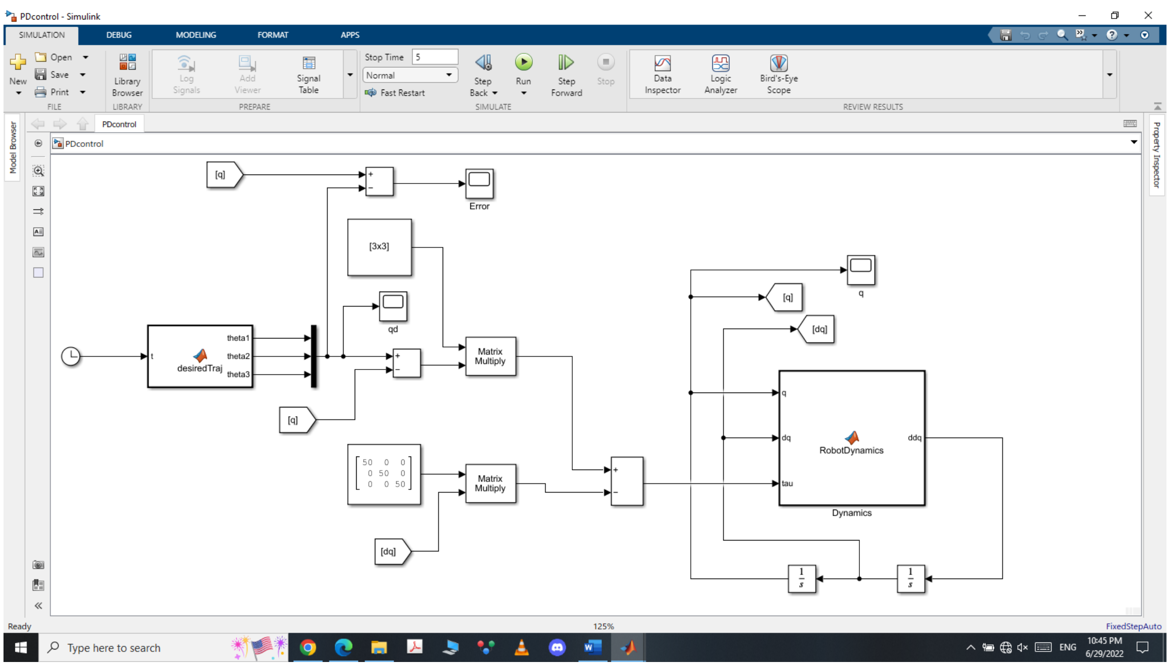

Figure 2- Simulation of PD Controller

According to the simulation, the error is revealed in fig. 3.

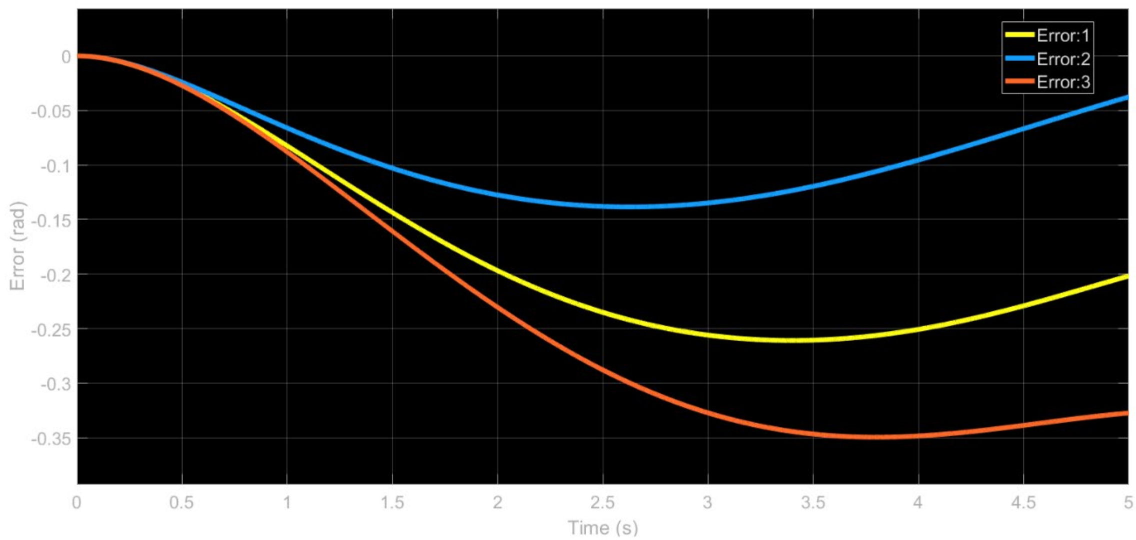

Figure 3- PD controller simulation results

PD + Gravity Controller

This controller is PD type with considering gravity in designing controller. Fig. 4 show the controller designed in Simulink.

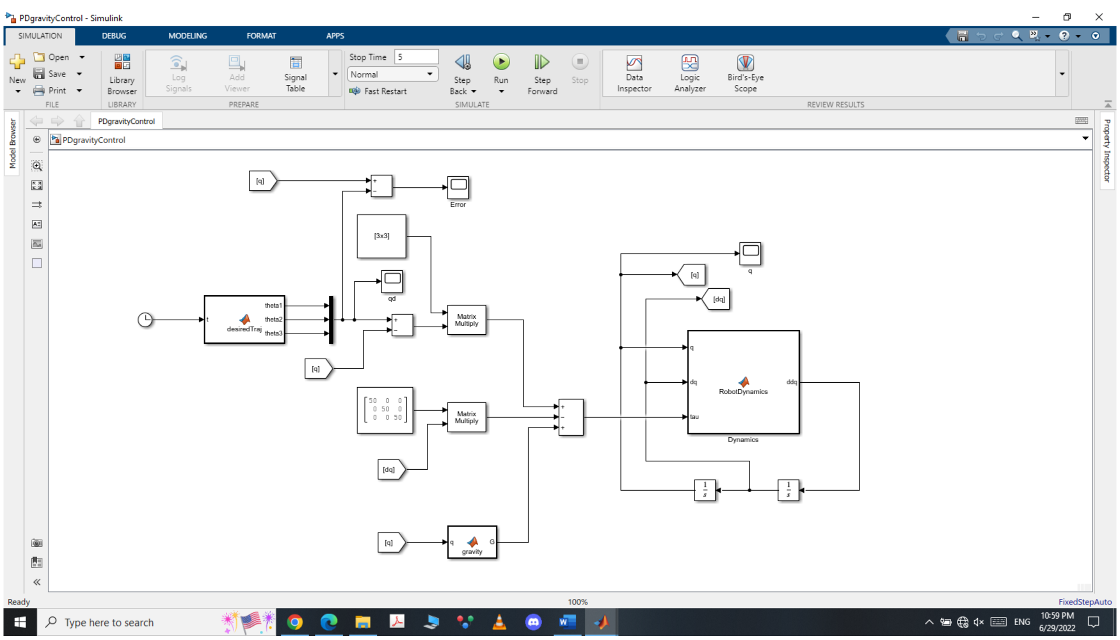

Figure 4- Simulation of PD + gravity Controller

According to the simulation, the error is revealed in fig. 5.

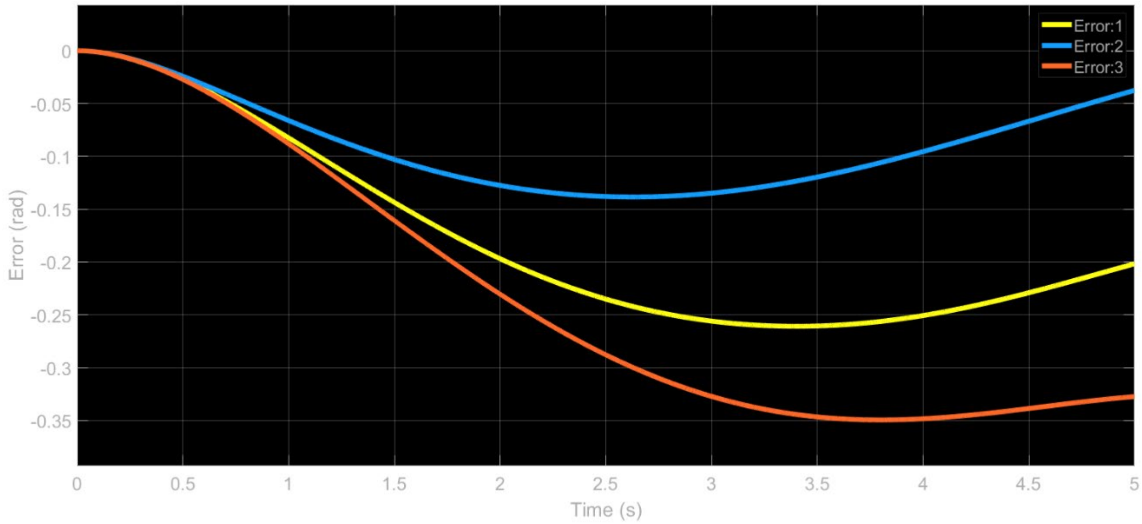

Figure 5- PD + gravity controller simulation results

Inverse Dynamics Control

Inverse Dynamics Control is an advanced control technique used in robotics to precisely control the motion of a robot by directly accounting for its dynamics. It works by computing the required joint torques (or forces) based on the desired motion (position, velocity, and acceleration) and the robot's dynamic model, which includes mass, inertia, Coriolis, and gravity effects. This method "inverts" the robot's equations of motion to calculate the inputs that will produce the desired trajectory. It is especially effective for robots with complex, nonlinear dynamics and is often used in manipulators and humanoid robots for smooth, coordinated movements.

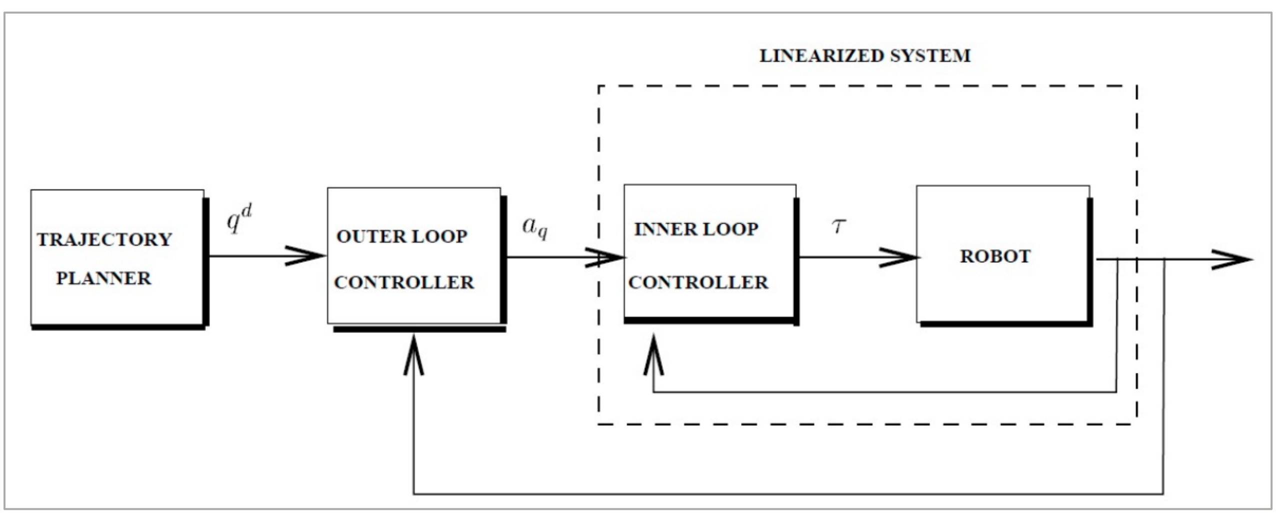

Figure 6- Inverse dynamics control diagram

PD Inverse Dynamics

The fig. 6 shows the simulations of the controller. The control design assumptions are defined as follows:

```
Kp = [1 0 0; 0 1 0; 0 0 1];
Kd = [1 0 0; 0 1 0; 0 0 1];
```

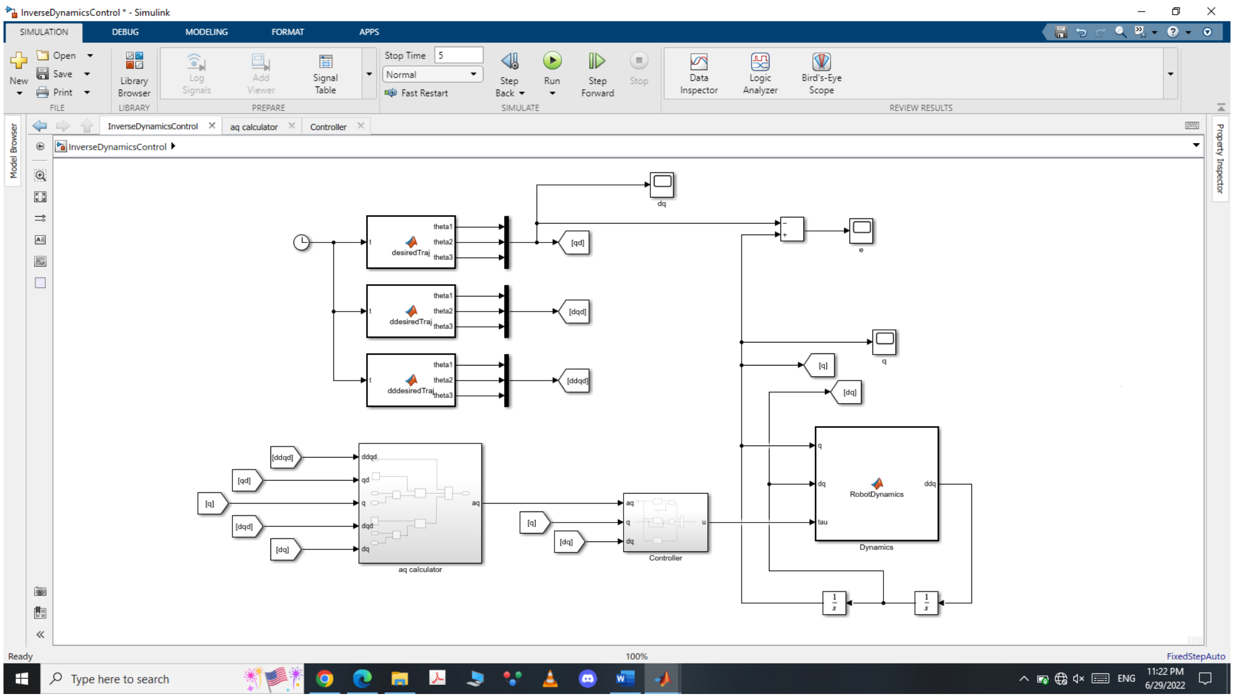

Figure 7- Simulation of PD inverse dynamics Controller

According to the simulation, the error is revealed in fig. 7.

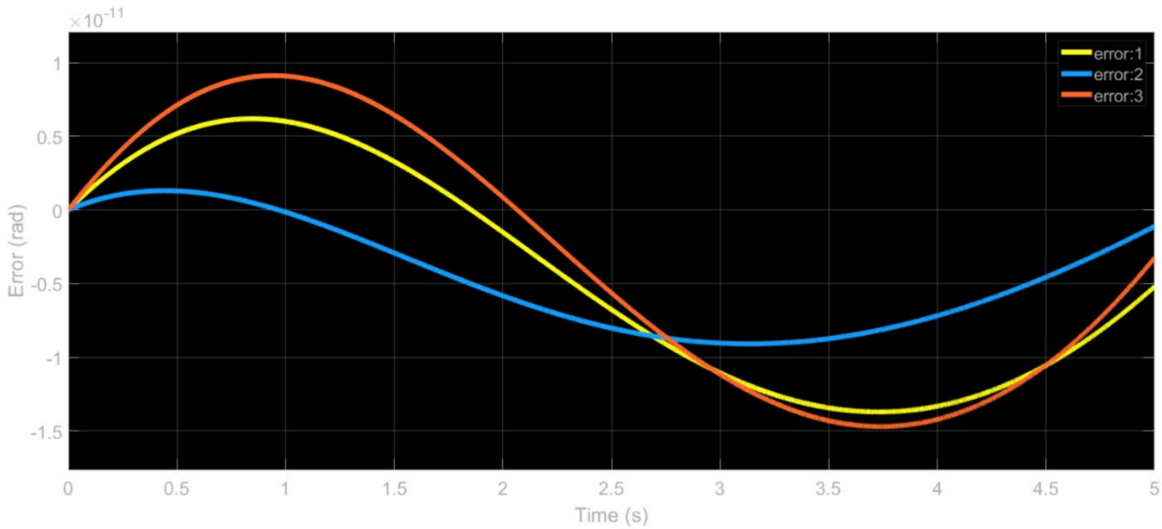

Figure 8- PD inverse dynamics controller simulation results

Robust Inverse Dynamics

The fig. 8 shows the simulations of the controller. The control design assumptions are defined as follows:

```
kd1 = 100; kd2 = 100; kd3 = 100;
kp1 = 100; kp2 = 100; kp3 = 100;
P = [-49 0 0 53 0 0; 0 -49 0 0 53 0; 0 0 -49 0 0 53; 53 0 0 -59 0 0; 0 53 0 0 -
59 0; 0 0 53 0 0 -59];
Rho = 0.01;
Mhat = [3.5 0 0; 0 3.5 0.75; 0 0.75 1.25];
Chat = [2 1.25 1; -1.25 0.5 0.5; -1 0.5 0];
Ghat = [0; 20; 5];
epsilon = 0.1;
```

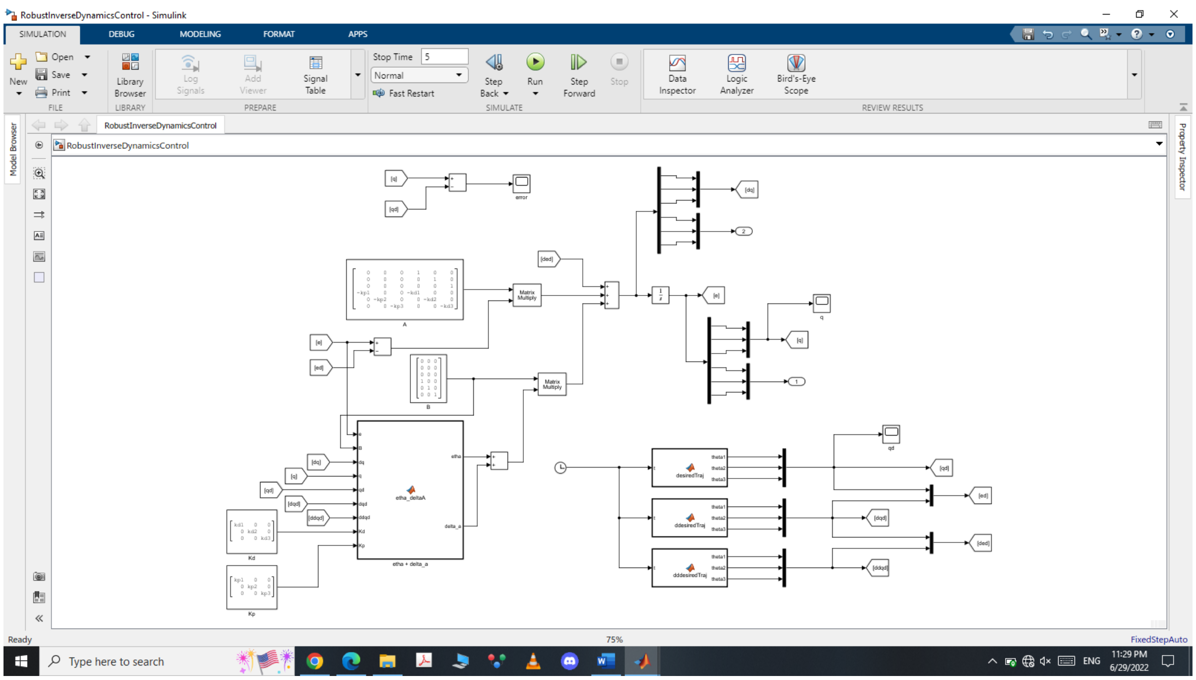

Figure 9- Simulation of robust inverse dynamics Controller

According to the simulation, the error is revealed in fig. 9.

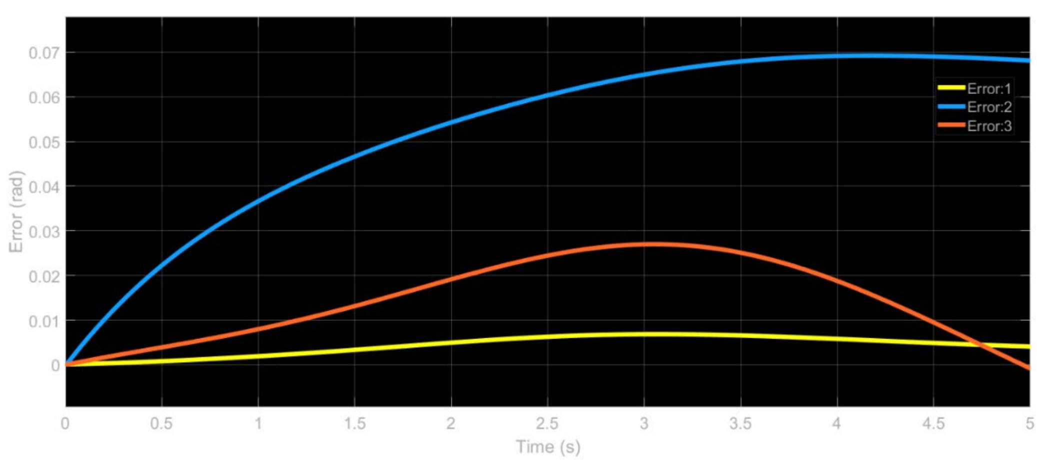

Figure 10- Robust inverse dynamics controller simulation results

Adaptive Inverse Dynamics

The fig. 10 shows the simulations of the controller. The control design assumptions are defined as follows:

```
gamma = 0.00001*[1 0 0 0 0 0 0 0 0; 0 1 0 0 0 0 0 0 0; 0 0 1 0 0 0 0 0 0; 0 0 0
1 0 0 0 0 0; 0 0 0 0 1 0 0 0 0; 0 0 0 0 0 1 0 0 0; 0 0 0 0 0 0 1 0 0; 0 0 0 0 0
0 0 1 0; 0 0 0 0 0 0 0 0 1];
P = [-49 0 0 53 0 0; 0 -49 0 0 53 0; 0 0 -49 0 0 53; 53 0 0 -59 0 0; 0 53 0 0 -
59 0; 0 0 53 0 0 -59];
kd1 = 60;
kd2 = 60;
kd3 = 60;
kp1 = 60;
kp2 = 60;
kp3 = 60;
Mhat = [3.5 0 0; 0 3.5 0.75; 0 0.75 1.25];
theta = [1.25; 0.25; 0.5; 1; 1; 1; 1; 1; 1];
```

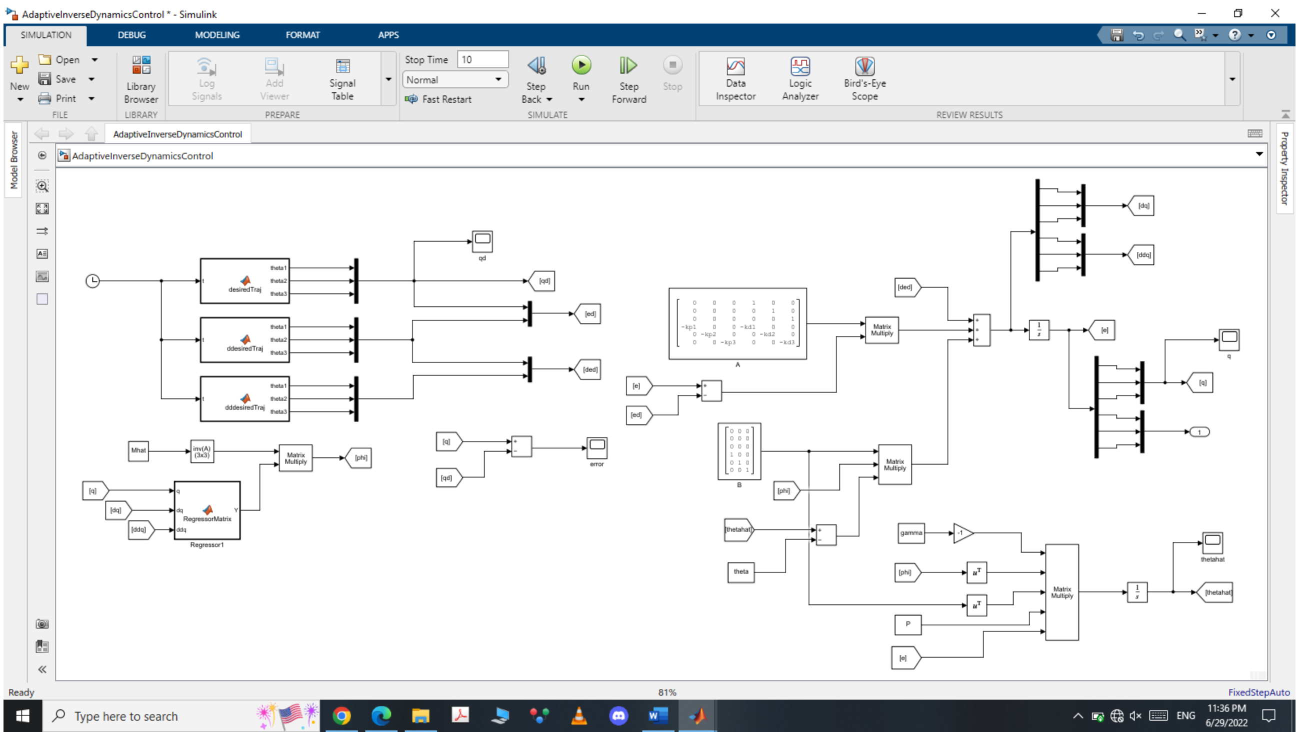

Figure 11- Simulation of adaptive inverse dynamics Controller

According to the simulation, the error is revealed in fig. 11.

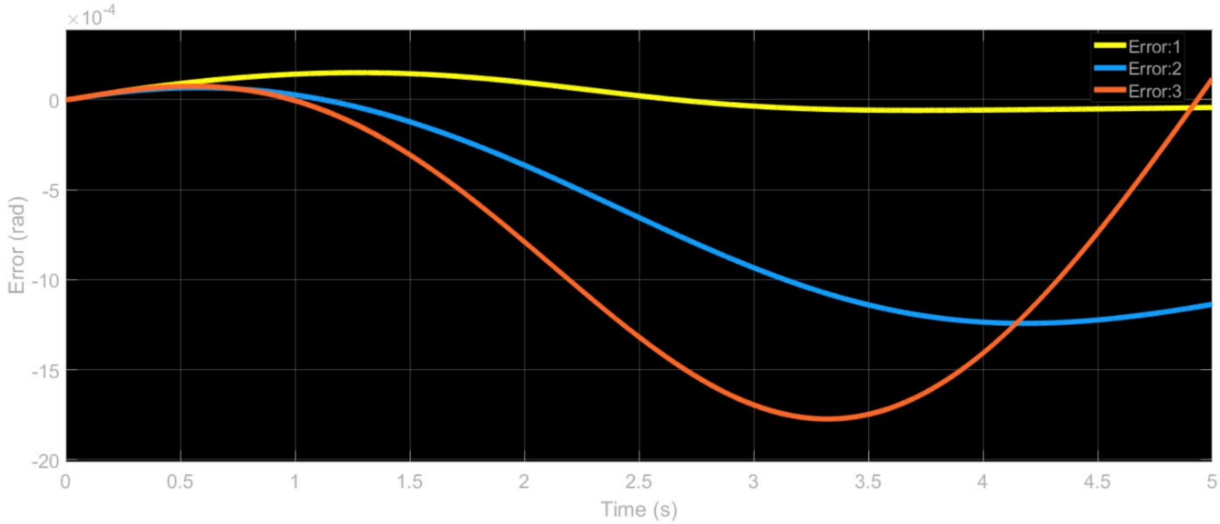

Figure 12- Adaptive inverse dynamics controller simulation results

Passivity Based Control

Passivity-Based Control (PBC) is a control strategy that ensures stability by leveraging the concept of *passivity*, which relates to energy flow in a system. In simple terms, a passive system doesn't generate energy on its own—it can only store or dissipate it. PBC designs controllers that make the robot behave like a passive system, ensuring that interactions (especially with uncertain environments or humans) are safe and stable. It's particularly useful in robotic systems with complex or nonlinear dynamics, such as compliant robots, manipulation tasks, or human-robot interaction, where energy-based stability guarantees are crucial.

PD Passivity Based

The fig. 12 shows the simulations of the controller. The control design assumptions are defined as follows:

```
gamma = 0.1.*[1 0 0; 0 1 0; 0 0 1];
K = 0.01.*[1 1 1; 0 1 0; 0 0 1];
```

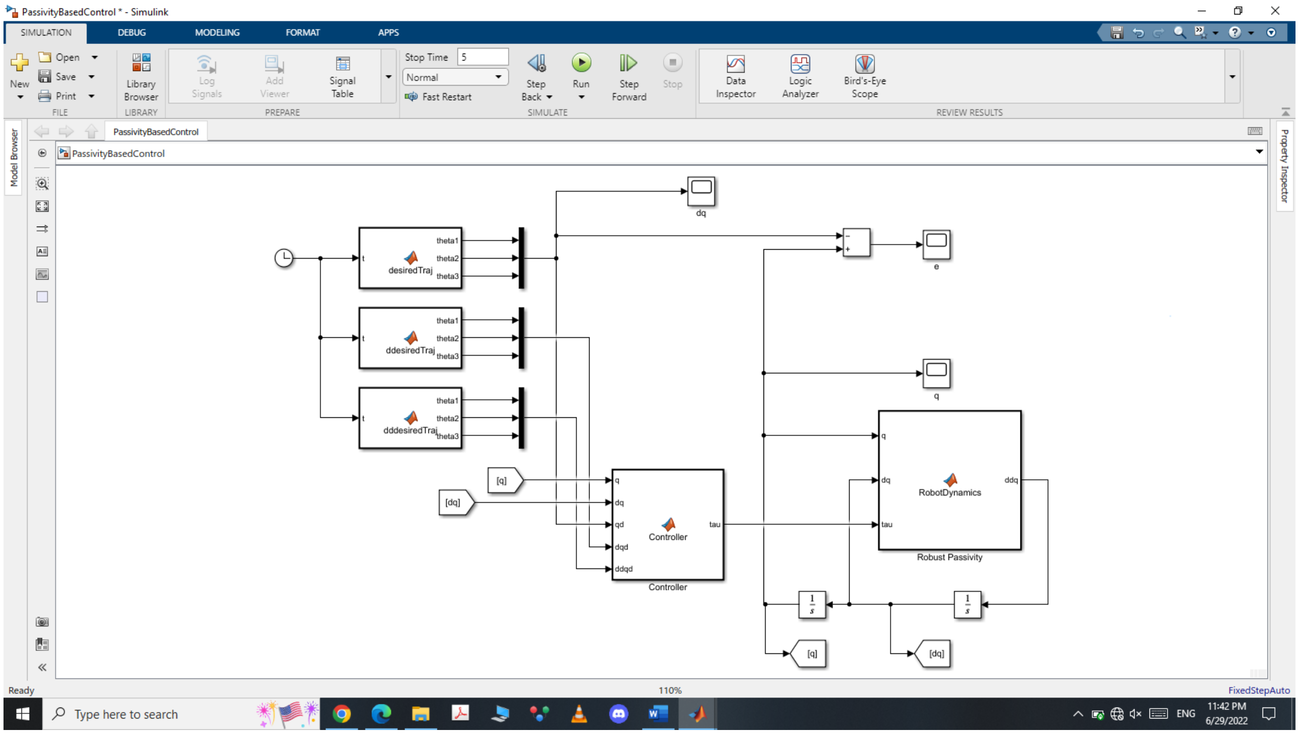

Figure 13- Simulation of PD passivity based Controller

According to the simulation, the error is revealed in fig. 13.

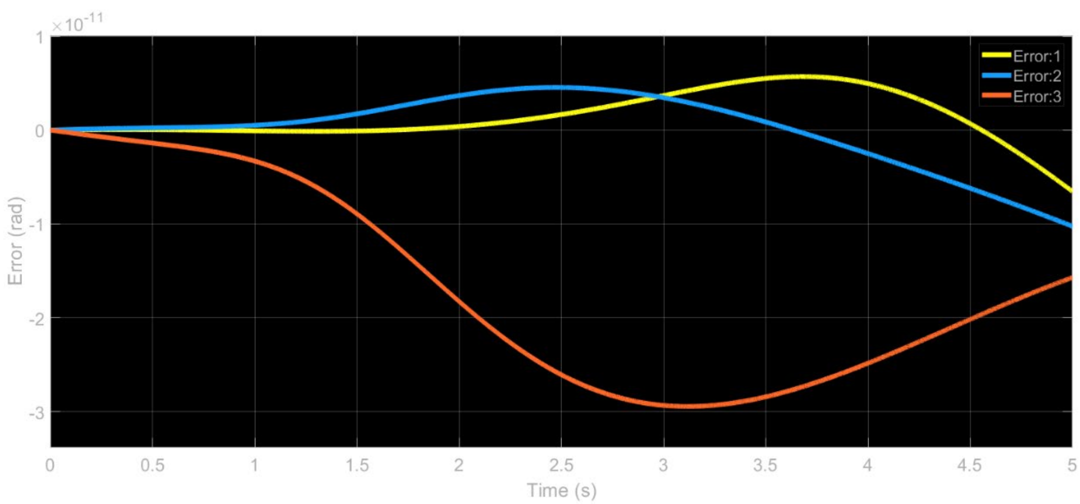

Figure 14- PD passivity based controller simulation results

Robust Passivity Based

The fig. 14 shows the simulations of the controller. The control design assumptions are defined as follows:

```
theta  = [1.25; 0.25; 0.5; 1; 1; 1; 1; 1; 1];
theta0 = theta.*3;
K = 100.*[1 0 0; 0 1 0; 0 0 1];
gamma = 1.*[1 0 0; 0 1 0; 0 0 1];
Rho = 10;
epsilon = 0.1;
```

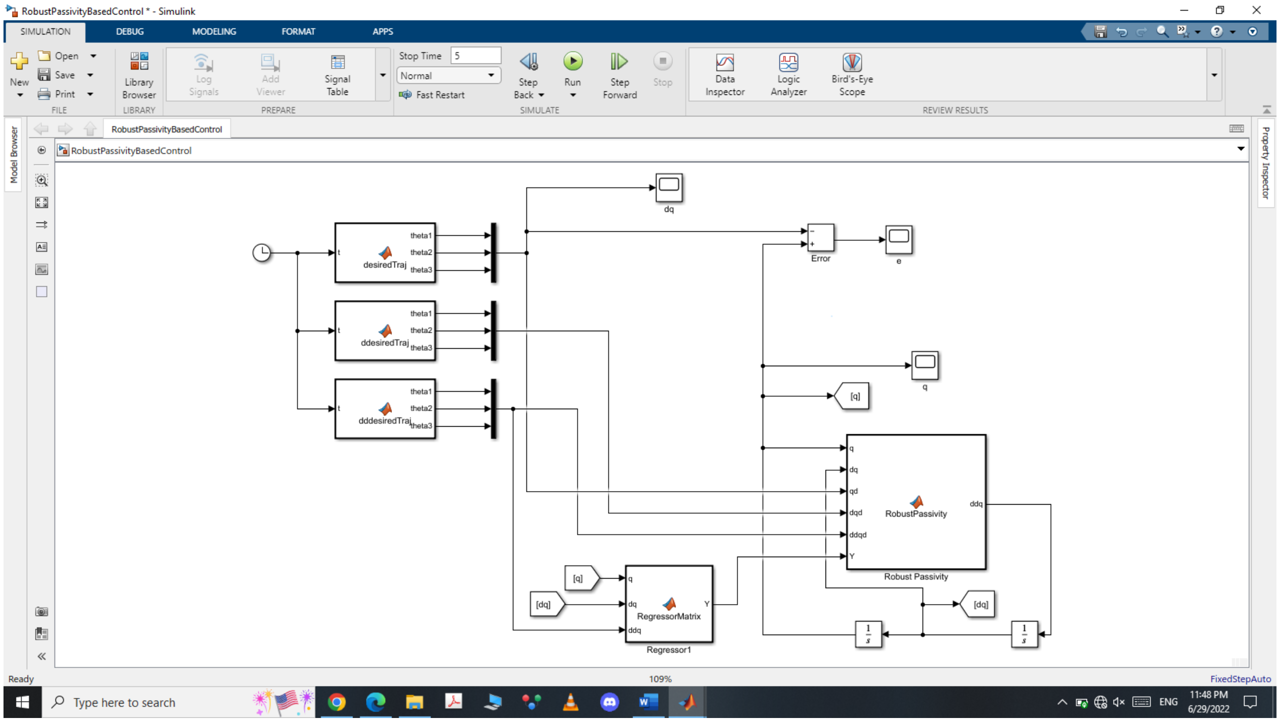

Figure 15- Simulation of robust passivity based Controller

According to the simulation, the error is revealed in fig. 15.

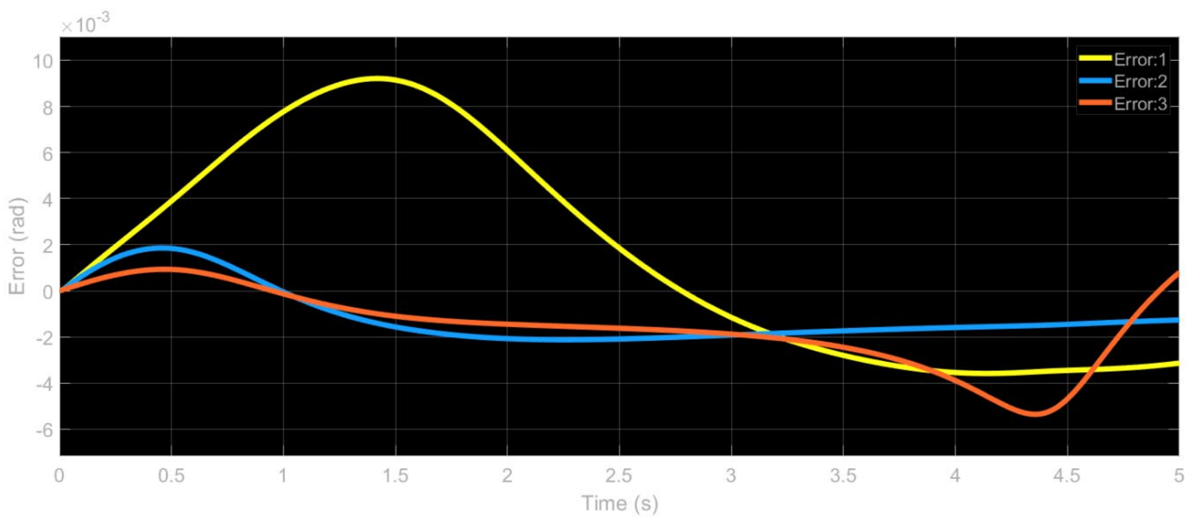

Figure 16- Robust passivity-based controller simulation results

Adaptive Passivity Based

The fig. 16 shows the simulations of the controller. The control design assumptions are defined as follows:

```
K = 100.*[1 0 0; 0 1 0; 0 0 1];
gamma = 1.*[1 0 0; 0 1 0; 0 0 1];
theta  = [1.25; 0.25; 0.5; 1; 1; 1; 1; 1; 1];
```

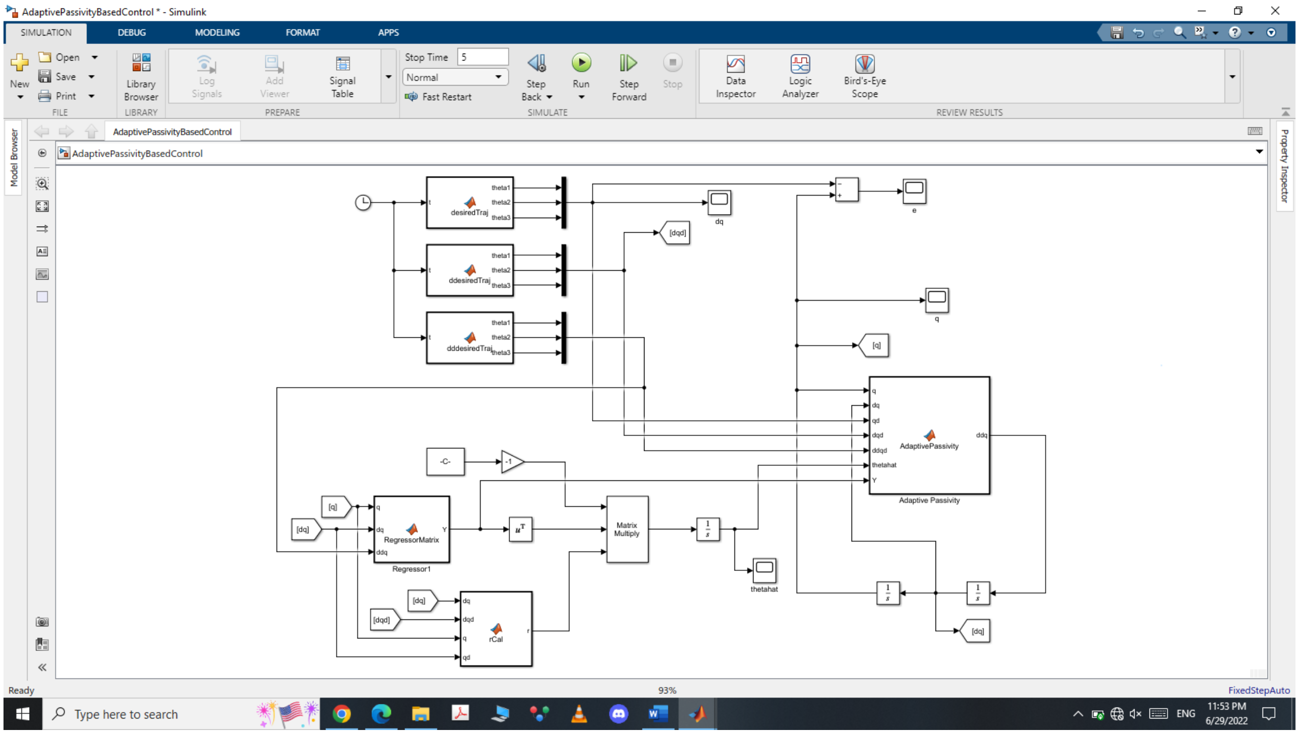

Figure 17- Simulation of adaptive passivity-based Controller

According to the simulation, the error is revealed in fig. 17.

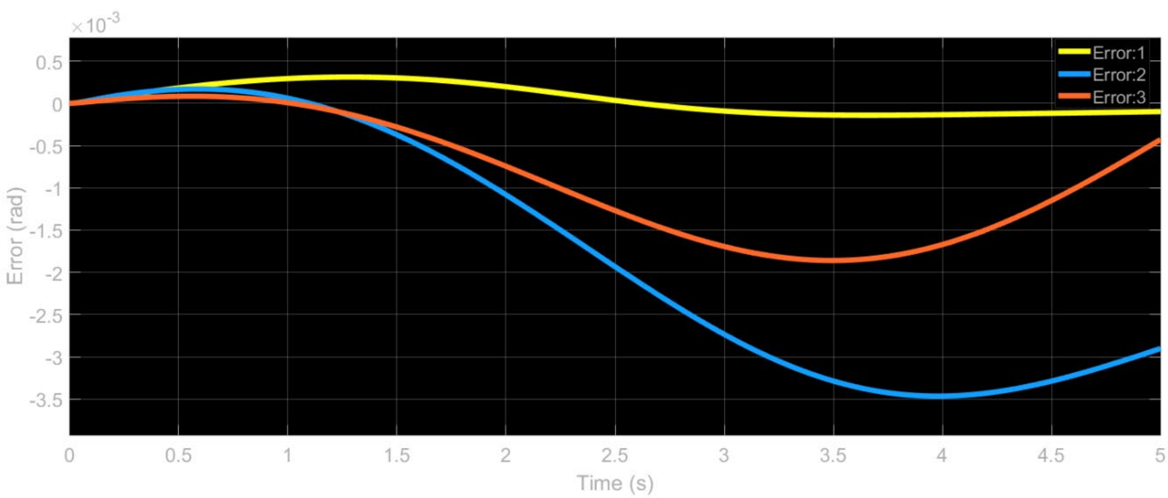

Figure 18- Adaptive passivity-based controller simulation results

References

[1] S. B. Niku, *Introduction to Robotics: Analysis, Control, Applications*, 2nd ed. Hoboken, NJ, USA: Wiley, 2011.

[2] C. Canudas de Wit, B. Siciliano, and G. Bastin, *Theory of Robot Control*. Berlin, Germany: Springer, 1996.