PID CONTROLLED INDUSTRIAL ROBOTIC ARM

Sankalp Varma, Sandesh Kumar Chandrakar, Srijan Pandey
201020447 , 201020445 , 201020456

Department of Data Science and Artificial Intelligence, Dr. S.P Mukherjee International Institute of Information Technology, Naya Raipur-493661, Chhattisgarh, India
Emails: sankalp20102@iiitnr.edu.in , sandesh20102@iiitnr.edu.in , srijan20102@iiitnr.edu.in

Abstract

—Aim of the project is to implement a PID controlled Industrial robotic arm practically. An articulated robot, often known as an industrial robotic arm, is a machine that closely resembles a human arm. It can have anywhere from two to ten rotary joints that serve as axis points and allow movement. Each extra joint or axis increases the range of motion. A manipulator, also known as a gripper, is commonly attached to the end of a robotic arm to allow it to perform a specific task.
After getting the final mathematical model, the linear transfer function of the system is derived. Using the transfer function, the PID controller is tuned and analyzed using MATLAB software. After getting all the stability criteria, root locus, time domain analysis and frequency domain analysis from MATLAB the system is implemented in real life conditions.

I. Introduction

The control system design process necessitates the use of system modeling. A precise model of a dynamical system allows us to gain a better understanding of the actual system and makes controller analysis and design easier.

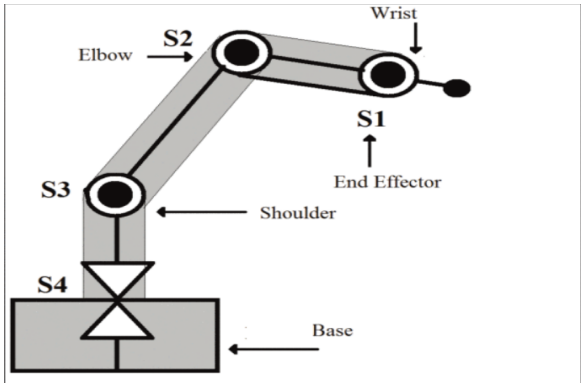
Industrial robots also assist in the removal of workers from hazardous locations and jobs that are physically demanding. As technology advances, the advantages of robots will become more apparent.
So it becomes really important to control the robots to produce the desired output from them. In these areas P.I.D is a good choice. PID controls a process by providing a constant fluctuation of output within a control loop feedback mechanism, removing oscillation and enhancing process efficiency. So in order to control the industrial robotic arm our group has come up with the topic. “**Control of Industrial Robotic Arm**” using a PID controller and joystick and also with keypad angle input.The PID controller constants kp,ki,kd are tuned manually to achieve the stability of the system.

II. Objectives

1. To control an industrial robotic arm using a PID controller.
2. To control the movement of the industrial robotic arm by controlling the joint variables of the robot using servo motors.

3. To make it controllable by the user , using a joystick , able to move and lift objects according to the user.

The control loop is shown in the figure below :



III. MATHEMATICAL MODEL

The first aim here will be to derive the Forward Kinematics equation for the robotic arm. Which will be given to the controller for the working of the actuator (end-effector) of the robotic arm.

To derive the Forward Kinematics Equation, we will be using the DH representation.

The main aim in DH representation is to calculate the overall forward kinematic equation(matrix), by multiplying the intermediate joint transformation matrices (An+1).

To calculate the An+1 matrix for all intermediate joints, we need first to calculate a DH table, having all the joint variables value for all the joints .

JOINT no.	d	a	θ
0-1	0	L1	Θ1
1-2	0	L2	Θ2

The An+1 matrix is calculated as:

$$A_{n+1} = \begin{bmatrix} C\theta_{n+1} & -S\theta_{n+1}C\alpha_{n+1} & S\theta_{n+1}S\alpha_{n+1} & a_{n+1}C\theta_{n+1} \\ S\theta_{n+1} & C\theta_{n+1}C\alpha_{n+1} & -C\theta_{n+1}S\alpha_{n+1} & a_{n+1}S\theta_{n+1} \\ 0 & S\alpha_{n+1} & C\alpha_{n+1} & d_{n+1} \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

For the joint (0-1), An+1 matrix is calculated as:

$$^{0}T_{1}(\theta_{1}) = \begin{pmatrix} C_1 & -S_1 & 0 & L_1C_1 \\ S_1 & C_1 & 0 & L_1S_1 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

For the joint (1-2), An+1 matrix is calculated as:

$$^{1}T_{2}(\theta_{2}) = \begin{pmatrix} C_2 & -S_2 & 0 & L_2C_2 \\ S_2 & C_2 & 0 & L_2S_2 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

Therefore, the overall transform matrix of 2-DOF robot arm obtained by combining the individual transform matrices is $T = {}^0T_2 = {}^0T_1\,{}^1T_2$ is given by

$$T = \begin{pmatrix} C_1C_2 - S_1S_2 & -C_1S_2 - S_1C_2 & 0 & L_2(C_1C_2 - S_1S_2) + L_1C_1 \\ S_1C_2 + C_1S_2 & -S_1S_2 + C_1C_2 & 0 & L_2(S_1C_2 + C_1S_2) + L_1S_1 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \quad (3)$$

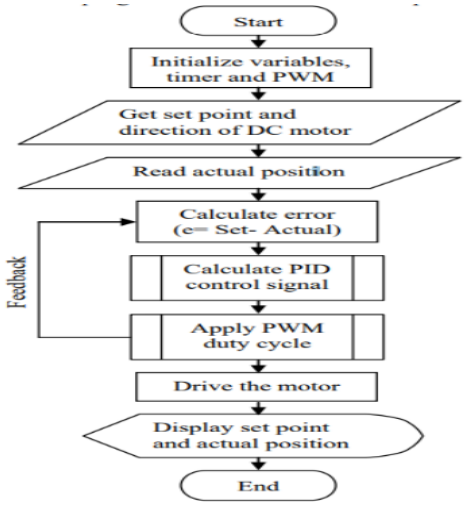
Now applying the inverse Kinematic equations, we will find the joint variables for both the joints:

dx, dy, θx and θy ->

$$\theta_1 = A\tan 2\left(\sqrt{1-\left[\frac{d_xC_2 + d_yL_2S_2}{d_x^{\,2} + d_y^{\,2}}\right]^2}, \frac{d_xC_2 + d_yL_2S_2}{d_x^{\,2} + d_y^{\,2}}\right)$$

$$\theta_2 = A\tan 2\left(\sqrt{1-\left[\frac{(d_x^{\,2} + d_y^{\,2}) - (L_1^{\,2} + L_2^{\,2})}{2L_1L_2}\right]^2}, \frac{(d_x^{\,2} + d_y^{\,2}) - (L_1^{\,2} + L_2^{\,2})}{2L_1L_2}\right)$$

FLOWCHART REPRESENTATION :



IV. TRANSFER FUNCTION

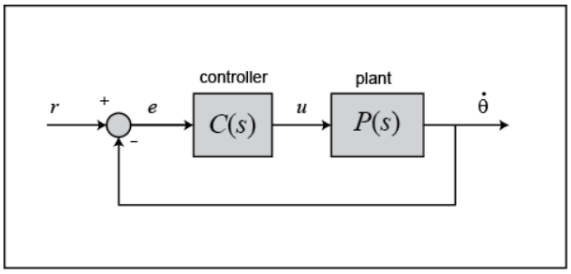
From the main problem, the dynamic equations in the Laplace domain and the open-loop transfer function of the DC Motor are the following.

$$s(Js + b)\Theta(s) = KI(s)$$

$$(Ls + R)I(s) = V(s) - Ks\Theta(s)$$

$$P(s) = \frac{\dot{\Theta}(s)}{V(s)} = \frac{K}{(Js + b)(Ls + R) + K^2} \qquad [\frac{rad/sec}{V}]$$

The structure of the control system has the form shown in the figure below.



For the original problem setup and the derivation of the above equations, please refer to the DC Motor Speed: System Modeling page.

For a 1-rad/sec step reference, the design criteria are the following.

1. Settling time less than 2 seconds
2. Overshoot less than 5%
3. Steady state error less 1%

```
J = 0.01;

b = 0.1;

K = 0.01;

R = 1;

L = 0.5;

s = tf('s');

P_motor = K/((J*s+b)*(L*s+R)+K^2);
```

Recall that the transfer function for a PID controller is:

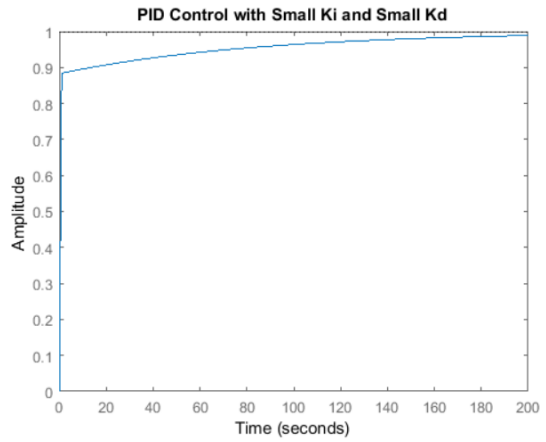
$$C(s) = K_p + \frac{K_i}{s} + K_d s = \frac{K_d s^2 + K_p s + K_i}{s}$$

PID CONTROL

Tuning the gains

1.Let's try a PID controller with small K_i and K_d. :

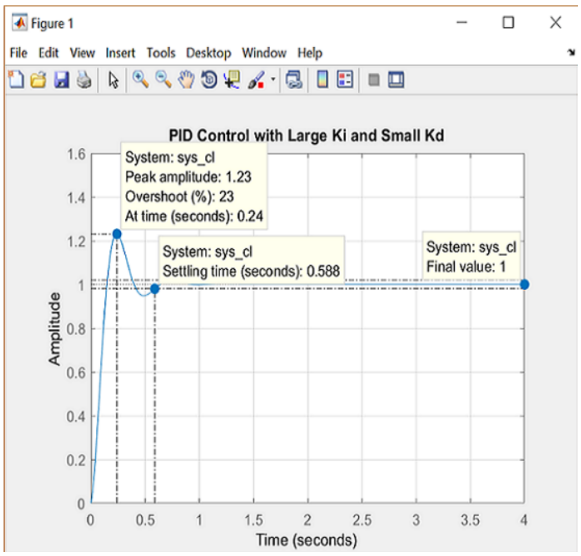
```
Kp = 75;
Ki = 1;
Kd = 1;
C = pid(Kp,Ki,Kd);
sys_cl = feedback(C*P_motor,1);
step(sys_cl,[0:1:200])
title('PID Control with Small Ki and Small Kd')
```



According to the foregoing, the steady-state error for a step input does indeed go to z ero. The time it takes to reach steady-state, on the other hand, is far longer than the needed settling period of two seconds.

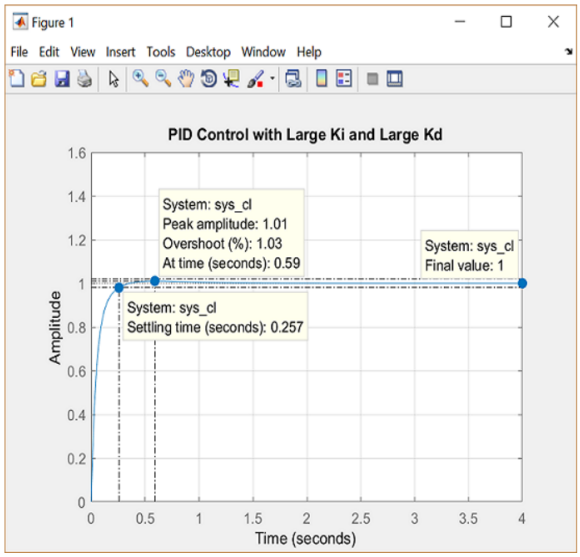
2.This process can be speed up by increasing the value of K_i.

```
Kp = 100;
      Ki = 200;
      Kd = 1;
      C = pid(Kp,Ki,Kd);
      sys_cl = feedback(C*P_motor,1);
      step(sys_cl, 0:0.01:4)
      grid
      title('PID Control with Large Ki and Small Kd')
```



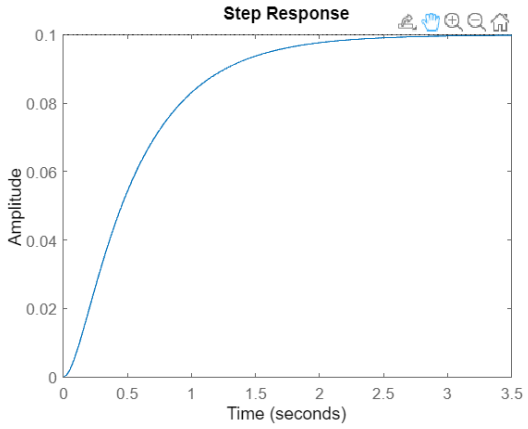
As expected, the steady-state error is now eliminated much more quickly than before. However, the large K_i has greatly increased the overshoot. Let's increase K_d in an attempt to reduce the overshoot.

```
Kp = 100;
      Ki = 200;
      Kd = 10;
      C = pid(Kp,Ki,Kd);
      sys_cl = feedback(C*P_motor,1);
      step(sys_cl, 0:0.01:4)
      grid
      title('PID Control with Large Ki and Large Kd')
```



As we had hoped, the increased K_d reduced the resulting overshoot. Now we know that if we use a PID controller with K_p = 100, K_i = 200, and K_d = 10, all of our design requirements will be satisfied.

So finally :
Kp = 100
Ki = 200
Kd = 10



V. ANALYSIS- TIME RESPONSE, ROOT LOCUS, FREQUENCY DOMAIN

Step response

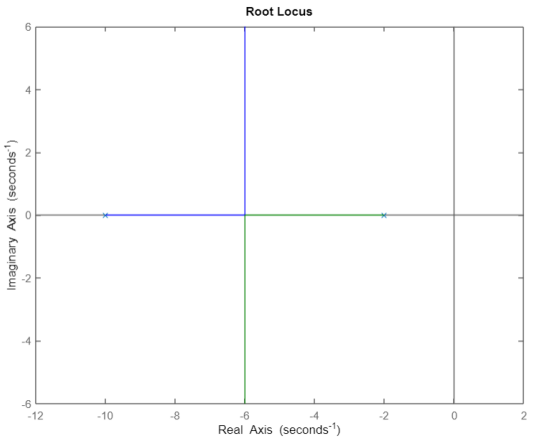
Code: .

```
sys = tf([100],[50,600,1001])

sys =

          100
 ---------------------
 50 s^2 + 600 s + 1001

Continuous-time transfer function.

  stepplot(sys)
```

ROOT LOCUS ANALYSIS:

MATLAB CODE-

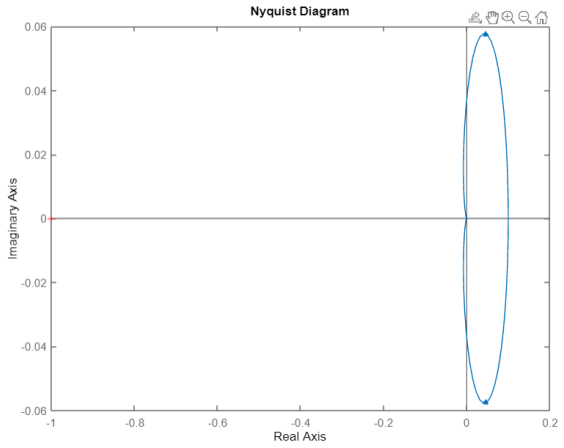
```
J = 0.01;

b = 0.1;

K = 0.01;

R = 1;

L = 0.5;

s = tf('s');

P_motor = K/((J*s+b)*(L*s+R)+K^2);

rlocus(P_motor)
```



FREQUENCY DOMAIN ANALYSIS:

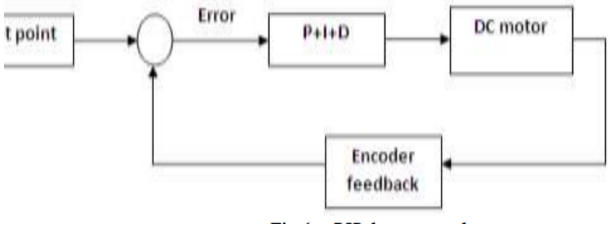
MATLAB CODE -

```
J = 0.01;

b = 0.1;

K = 0.01;

R = 1;

L = 0.5;

s = tf('s');

P_motor = K/((J*s+b)*(L*s+R)+K^2);

nyquist plot(P_motor)
```



V. DESIGN AND TUNING OF CONTROLLER

In industrial control systems, a proportional–integral– derivative controller (PID controller) is commonly utilized. It is a feedback controller that uses a general feedback control loop technology. The PID loop specifies the amount of energy to be provided to the motor at any given time throughout a movement. This is based on the current location of the motor and the projected location. The equation has four parts that define the load, with the three main components referred to as P, I, and D, and the minor friction component referred to as K.



P is the reaction to the present error, I is the reaction to the total of recently appearing errors, and D is the reaction to the rate at which the error changes. The aggregate of all three parts contributes to a control mechanism, such as motor speed control, in which P value depends on present error, I on previous error accumulation, and D predicts future error based on current rate of change. There are numerous PID tuning methods, however only a few are covered here.

PID general formula G1 = Kp + Ki s + Kd ∗ s => $G1$ = Kd [s 2 + Kp Kd ∗ s + Ki Kd] s Second order transfer function equation G2 = ω2 s 2 + 2εωs + ω2 By comparing the G1 and G2 equations we get ω2 = Ki Kd and 2ωε = Kp Kd By assuming Kp=1, and ω=0.16155 (by the DC motor equation) ,we can calculate the values of Ki and Kd as 0.080775 and 3.09501.

VI. ACKNOWLEDGEMENT

We would like to thank Dr.Debanjan Das for his continuous support and guidance. We would like to thank the ECE lab for providing us with the necessary components.

We would like to thank everyone who helped in our project directly or indirectly.

VII. CONCLUSION AND FUTURE SCOPE

Hence an industrial robotic arm is designed using PID controller, capable of making controlled movements (rotatory) , on the command of user , to move or pick different objects .As a future scope in the project we would like to work on increasing the efficiency of the robotic arm , adding different other sensors to it , to make it able to auto detect objects in it's workspace .

VIII. REFERENCES

control - Why have a PID controller for a DC servo motor? - Electrical Engineering Stack Exchange