# PID Motor Control

Kiran M V, Manikantan R S
Roll Numbers: 122101019, 122101046

## 1 Aim

To implement a closed loop PID (Proportional-Integral-Derivative) control system for a DC motor using FRDM-KL25Z microcontroller board.

## 2 Abstract and Description

PID Motor Control System is a system designed to regulate the speed or position of a motor by continuously adjusting the motor's input based on the error between the desired output and the actual output. It will give the user/computer the ability to control the speed and direction of the motor.
Here, we are trying to take control of the direction and speed of the axle of the DC motor. We are achieving this by using PWM for controlling the speed of the motor and by using the PID control system for changing the direction of the axle.

## 3 Introduction

### 3.1 FRDM-KL25Z

The hardware platform that we use is the NXP Freedom Development Board (FRDM-KL25Z). The KL25Z is a Cortex-M0+ microcontroller for embedded applications. The FRDM-KL25Z comes with:

- 32-bit ARM Cortex-M0+ core running at 48MHz
- 128KB FLASH
- 16KB RAM
- USB Device
- SPI
- I2C
- ADC
- DAC
- PWM

- Touch Sensor
- GPIOs
- 3-axis Accelerometer
- PWM Controlled RGB LED
- Built-in USB FLASH programmer

ARM Cortex M0+ is based on ARMv6-M architecture. Its processor is built on a highly area and power optimized processor core with von Neumann architecture. The processor delivers exceptional energy efficiency through a small but powerful instruction set and extensively optimized design.

## 3.2 Optical Encoder

An optical rotary encoder is a device used to measure the rotational position and speed of a motor shaft or any rotating object. It provides feedback to the control system, allowing for accurate monitoring and control of the motor's position.

## 3.3 Motor Driver

Here we have used the L298 Motor driver module for driving the DC motor. It is used for controlling DC motors or bipolar stepper motors. It is designed to handle moderate to high currents and provides a convenient way to interface motors with microcontrollers or other control systems.
It can handle a significant amount of current, typically up to 2 amps per channel, with peak currents reaching higher levels for short durations. It also has inputs for controlling the motor's direction and speed. By applying appropriate logic signals to these inputs, you can determine the desired motor direction (forward or reverse) and adjust the motor speed by using Pulse Width Modulation (PWM) signals.

## 3.4 PWM

PWM (Pulse Width Modulation) is used to control the motor's speed. It is a technique that involves rapidly switching a signal on and off at a fixed frequency while varying the width or duration of the pulse.
PWM is used for speed control, i.e., in a speed control system, PWM is used to adjust the average voltage applied to the motor. By varying the width of the pulse, the voltage can be controlled. For example, a longer high pulse will result in a higher average voltage and thus a faster motor speed, while a shorter high pulse will result in a lower average voltage and a slower speed.
PWM is an efficient and commonly used method for motor control because it allows for precise control of the motor's behavior while minimizing power loss and heat generation. By adjusting the duty cycle, the motor's speed, position, or torque can be precisely regulated in a PID control system.

## 3.5 PID

### 3.5.1 Proportional Control

The proportional control component generates an output that is proportional to the current error. It multiplies the error by a constant gain factor (Kp) and provides an immediate response to the error. The larger the error, the larger the corrective action applied to the motor.

### 3.5.2 Integral Control

The integral control component integrates the error over time to eliminate steady-state errors. It sums up the error multiplied by a gain factor (Ki) and provides continuous correction based on the accumulated error. It helps to address any residual error that the proportional control alone cannot eliminate.

### 3.5.3 Derivative Control

The derivative control component calculates the rate of change of the error and multiplies it by a gain factor (Kd). It provides a damping effect by anticipating the future trend of the error. This helps to reduce overshoot and stabilize the system's response.
By adjusting the gains (Kp, Ki, and Kd) of these three components, the control system can be fine-tuned to achieve the desired response. The correction/tuning process involves finding the optimal combination of gains that minimizes overshoot, settles quickly, and maintains stability without oscillations.
Here, the PID control system is used to control the speed of the motor and its position. It is used in places where precise control of motorized systems are required.

# 4 Implementation and Results

## 4.1 Note

We are using Putty for sending data.

- Baud Rate: 9600

  Corresponding register configurations are:

- UART0_BDH = 0x00
- UART0_BDL = 0x0D
- OSR = 0x0E

## 4.2 Motor Interfacing

The DC motor is connected to the microcontroller board. The motor's power supply and ground connections are properly connected. We have connected the L298 motor driver for driving the motor if it requires higher current than what the microcontroller can provide.

### 4.3 Sensor Integration

For the sensor, we are using the Optical Rotary Encoder. It provides feedback on the motor's position or speed. The sensor output is connected to the appropriate input pins of the microcontroller.

### 4.4 PID Algorithm Implementation

The PID algorithm code is written in C programming language for the FRDM-KL25Z microcontroller board. The code includes calculations for the proportional, integral, and derivative terms based on the sensor feedback and desired setpoint. By adjusting the PID gains (Kp, Ki, and Kd), the control system can be fine-tuned to achieve the desired response.

### 4.5 Motor Control using PWM

By using the PWM output pins of FRDM-KL25Z, we can generate the control signals for the motor. The PWM signals will adjust the motor's duty cycle to control its speed.

### 4.6 Feedback Mechanism

By comparing the feedback signals from the sensor with the desired setpoint, we can calculate the error. Feed the error to the code to compute the control output. After getting the output, adjust the motor control signals accordingly.

### 4.7 Fine Tuning

By observing the motor's response, we can make adjustments to the control gains (Kp, Ki, and Kd) to achieve the desired performance.

## 5 Conclusion

We were able to control the position of the motor as we desired. We went deep into embedded C programming and PWM. We also learnt to work with optical rotary encoder.

## 6 Annexure

### 6.1 Videos

- Signals from Optical rotary encoder
- Changing Duty Cycle using PWM controlled by Microcontroller

### 6.2 References

- Brushed DC Motor Control with MC34931/MC33931 - mbed