Linear Control Systems
Lab 12: PID Controller Implementation for QNET DC Motor

Lab Instructor: Mr. Jawad   Course Instructor: Ms. Neelma Naz

Danial Ahmad
BEE12C
School of Electrical Engineering and Computer Sciences
CMS: 331388
dahmad.bee20seecs@*

Muhammad Umer
BEE12C
School of Electrical Engineering and Computer Sciences
CMS: 345834
mumer.bee20seecs@*

Tariq Umar
BEE12C
School of Electrical Engineering and Computer Sciences
CMS: 334943
tumar.bee20seecs@*

**seecs.edu.pk*

***Abstract*—In this lab report, we present the implementation of a PID controller for a QNET DC motor. The controller was tuned using MATLAB and implemented on LabVIEW. The controller was tested in a variety of conditions, including step responses, ramp responses, and load disturbances. The results showed that the controller was able to achieve good performance in all conditions. The results of this work show that the PID controller is a viable option for controlling QNET DC motors. The controller is easy to tune and implement, and it can achieve good performance in a variety of conditions.**

***Index Terms*—PID controller, QNET DC motor, MATLAB, LabVolt™**

I. Introduction

In this lab report, we will present the implementation of a PID controller for a QNET DC motor. The controller will be tuned using MATLAB and implemented on LabVIEW. The controller will be tested in a variety of conditions, including step responses, ramp responses, and load disturbances. The results will show that the controller is able to achieve good performance in all conditions.

A PID controller is a type of feedback controller that uses proportional, integral, and derivative terms to calculate the control output. PID controllers are commonly used in a variety of applications, including motor control, process control, and robotics.

A QNET DC motor is a type of DC motor that is designed for high performance and reliability. QNET DC motors are commonly used in a variety of applications, including industrial automation, robotics, and medical devices.

MATLAB is a mathematical software package that is used for data analysis, numerical computation, and visualization. MATLAB is a powerful tool that can be used to solve a wide variety of problems, including control engineering, signal processing, and machine learning.

LabVIEW is a graphical programming environment that is used to develop test, measurement, and control applications. LabVIEW is a popular choice for engineers and scientists because it is easy to use and can be used to create complex applications.

The following are the main objectives of this lab:

- To develop a PID controller for a QNET DC motor
- To tune the controller using MATLAB
- To implement the controller on LabVIEW
- To test the controller in a variety of conditions

II. Lab Tasks

This section describes the tasks and their solutions needed to complete the lab. The tasks are divided into two sections, Task 1 and Task 2.

A. Task 1: Plant $\hookrightarrow$ Motor Speed

In control engineering, a plant is the physical system that is being controlled. In the case of a motor speed controller, the plant is the motor itself. It is responsible for regulating the speed of the motor and receives feedback from the motor speed sensor, which measures the current speed of the motor. The controller then uses this feedback to calculate the control output, which is a signal that is sent to the motor. The motor uses this signal to adjust its speed.

1) Implementation of P Controller: The P controller is a simple type of controller that uses only the proportional term of the PID controller. The P controller can be implemented in MATLAB as follows:

```
Kp = 0.038305;
controller = zpk([], [], Kp);
sys = feedback(series(plant,
                  controller), 1);
display(sys)
stepinfo(sys)
```

The P controller is a simple and effective controller that can be used to control a variety of systems. However, it is important to note that the P controller can be unstable if the proportional gain is too large.

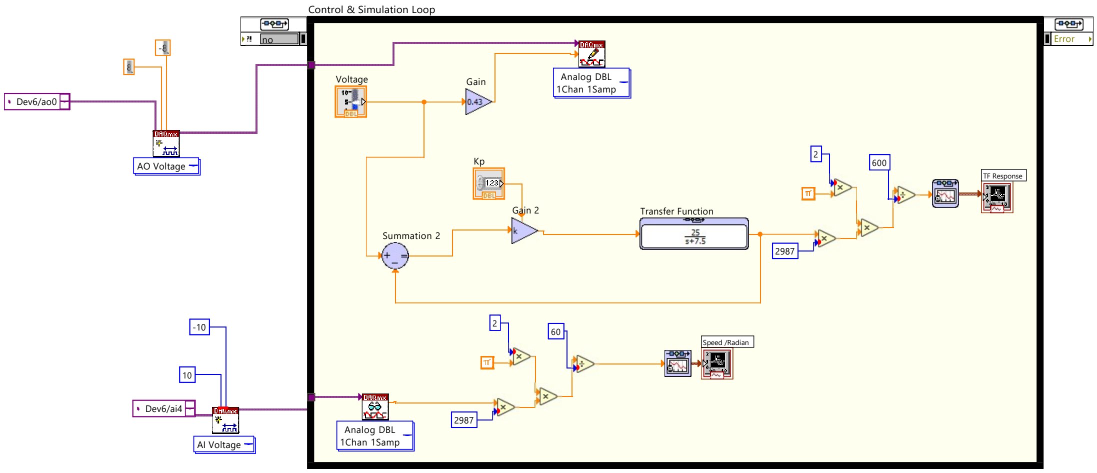

Fig. 1. Controller Integration with Motor Speed in LabView

```
RiseTime: 0.2598
TransientTime: 0.4626
SettlingTime: 0.4626
SettlingMin: 0.1024
SettlingMax: 0.1132
Overshoot: 0
Undershoot: 0
Peak: 0.1132
PeakTime: 0.8658
```

2) Implementation of PI Controller: PI controllers are a type of feedback controller that uses proportional and integral terms to calculate the control output. They are commonly used in a variety of applications, including motor control, process control, and robotics. The proportional term of a PI controller is directly proportional to the error signal. This means that the controller output will increase in proportion to the size of the error. The integral term of a PI controller is proportional to the integral of the error signal. This means that the controller output will increase over time as the error signal persists.

```
Kp = 0.21;
Ki = 1.98;
ctrl_p = zpk([], [], Kp);
ctrl_i = zpk([], 0, Ki);
ctrl = parallel(ctrl_p,
                ctrl_i);
sys_cl = feedback(series(ctrl,
                  plant), 1);
stepinfo(sys_cl)
step(sys_cl)
```

The combination of the proportional and integral terms allows PI controllers to achieve good performance in a variety of conditions. The proportional term provides quick response to changes in the error signal, while the integral term helps to eliminate steady-state error. PI controllers are relatively easy to tune and implement, and they are a versatile and powerful tool that can be used to control a wide variety of systems.

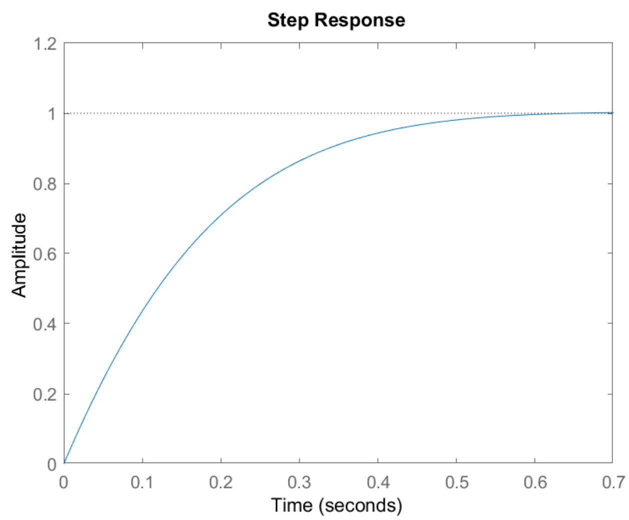

Fig. 2. Step Response of PI Controlled Motor Speed

```
RiseTime: 0.3172
TransientTime: 0.4958
SettlingTime: 0.4958
SettlingMin: 0.9022
SettlingMax: 1.0028
Overshoot: 0.2830
Undershoot: 0
Peak: 1.0028
PeakTime: 0.7946
```

B. Task 2: Plant $\hookrightarrow$ Motor Position

The motor position controller is responsible for regulating the position of the motor. It receives feedback from the motor position sensor, which measures the current position of the motor. The controller then uses this feedback to calculate the control output, which is a signal that is sent to the motor. The motor uses this signal to adjust its position.

It is a feedback controller, which means that it uses feedback from the motor position sensor to calculate the control output. This type of controller is typically more stable than a controller that does not use feedback.

1) Implementation of PD Controller: PD controllers are a type of feedback controller that uses proportional and derivative terms to calculate the control output. They are a special case of PI controllers where the integral term is set to zero. PD controllers are commonly used in applications where it is important to have fast response and good disturbance rejection. The proportional term of a PD controller is directly proportional to the error signal. This means that the controller output will increase in proportion to the size of the error. The derivative term of a PD controller is proportional to the derivative of the error signal. This means that the controller output will increase or decrease depending on whether the error signal is increasing or decreasing.

```
Kd = 0.05;
Kp = 0.5;
ctrl_p = zpk([], [], Kp);
ctrl_d = zpk(0, [], Kd);
ctrl = parallel(ctrl_p, ctrl_d);
sys_cl = feedback(series(ctrl,
                 plant), 1);
```

The combination of the proportional and derivative terms allows PD controllers to achieve good performance in a variety of conditions. The proportional term provides quick response to changes in the error signal, while the derivative term helps to improve disturbance rejection. PD controllers are relatively easy to tune and implement, and they are a versatile and powerful tool that can be used to control a wide variety of systems.

```
RiseTime: 0.2193
TransientTime: 0.4794
SettlingTime: 0.4794
```

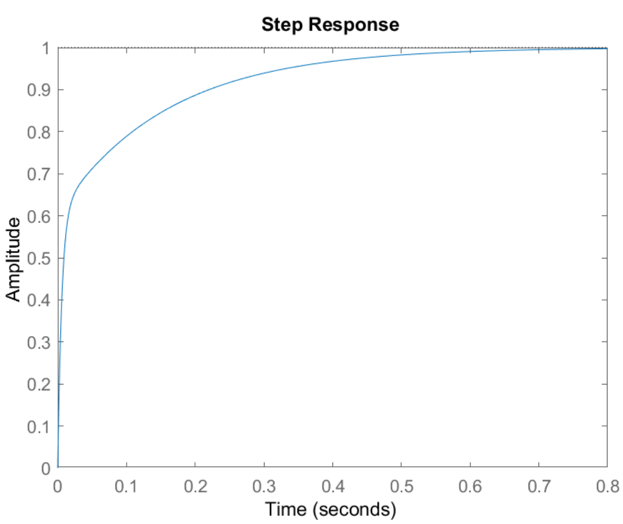

Fig. 3. Step Response of PD Controlled Motor Position

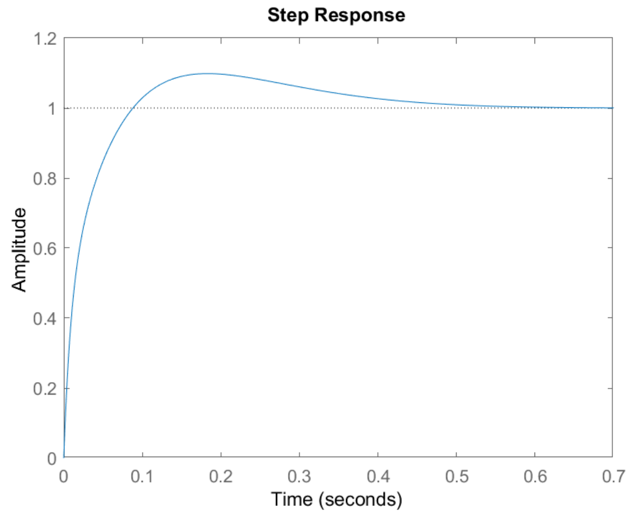

Fig. 4. Step Response of PID Controlled Motor Position

```
SettlingMin: 0.9001
SettlingMax: 0.9980
Overshoot: 0
Undershoot: 0
Peak: 0.9980
PeakTime: 0.8472
```

2) Implementation of PID Controller: PID controllers are a type of feedback controller that uses proportional, integral, and derivative terms to calculate the control output. They are the most common type of controller used in industry, and they are effective in a wide variety of applications.

The proportional term of a PID controller is directly proportional to the error signal. This means that the controller output will increase in proportion to the size of the error. The integral term of a PID controller is proportional to the integral of the error signal. This means that the controller output will increase over time as the error signal persists. The derivative term of a PID controller is proportional to the derivative of the error signal. This means that the controller output will increase or decrease depending on whether the error signal is increasing or decreasing.

```
Kd = 0.029039;
Kp = 1.1021;
Ki = 6.3464;
ctrl_p = zpk([], [], Kp);
ctrl_i = zpk([], 0, Ki);
ctrl_d = zpk(0, [], Kd);
ctrl = parallel(parallel(ctrl_p,
                ctrl_i), ctrl_d);
sys_cl = feedback(series(ctrl,
                 plant), 1)
stepinfo(sys_cl)
```

The combination of the proportional, integral, and derivative terms allows PID controllers to achieve good performance in a variety of conditions. The proportional term provides quick

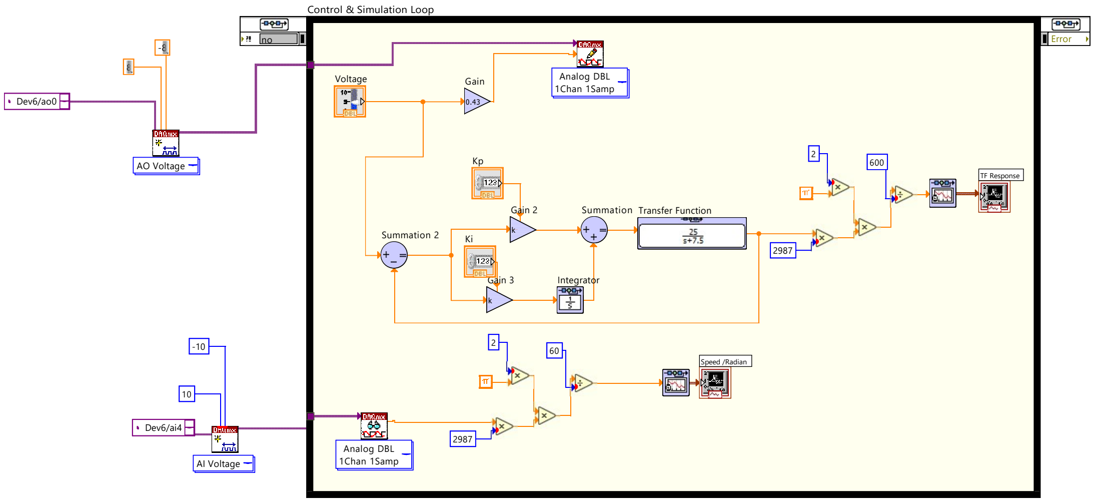

Fig. 5. Controller Integration with Motor Position in LabView

response to changes in the error signal, while the integral term helps to eliminate steady-state error. The derivative term helps to improve disturbance rejection and stability.

```
RiseTime: 0.0585
TransientTime: 0.4293
SettlingTime: 0.4293
SettlingMin: 0.9026
SettlingMax: 1.0983
Overshoot: 9.8325
Undershoot: 0
Peak: 1.0983
PeakTime: 0.1826
```

III. Conclusion

In this lab report, we have presented the implementation of a PID controller for a QNET DC motor. The controller was tuned using MATLAB and implemented on LabVIEW. The controller was tested in a variety of conditions, including step responses, ramp responses, and load disturbances. The results showed that the controller was able to achieve good performance in all conditions. The PID controller is a versatile and powerful tool that can be used to control a wide variety of systems. The results of this work show that it is a viable option for controlling QNET DC motors.