Faculty of
Engineering

School of Electrical, Electronic & Computer Engineering

EERI 418
Practical 1

Completed By
MJ Bezuidenhout 24162299

Submitted to:

Prof. K. Uren

May 23, 2017

Innovation through diversity ™

Faculty of Engineering
School of Electrical, Electronic and Computer Engineering

NORTH-WEST UNIVERSITY
YUNIBESITI YA BOKONE-BOPHIRIMA
NOORDWES-UNIVERSITEIT
POTCHEFSTROOM CAMPUS

ABSTRACT

THIS IS THE ABSTRACT, LOCATED IN FRONTBACKMATTER. EDIT OR UN-COMMENT

DECLARATION

I, MJ Bezuidenhout, declare that this report is a presentation of my own original work. Whenever contributions of others are involved, every effort was made to indicate this clearly, with due reference to the literature. No part of this work has been submitted in the past, or is being submitted, for a degree or examination at any other university or course.

Potchefstroom, May 23, 2017

MJ Bezuidenhout, May 23, 2017

CONTENTS

LIST OF FIGURES

LIST OF TABLES

ACRONYMS

PID	Proportional,Integral & differential Controller
PI	Proportional& differential Controller

Ts	Settling Time
PO	Percent Overshoot

INTRODUCTION

LITERATURE STUDY

2.1 MODELING THE DC MOTOR

According to the Prescribed textbook[1, p.160],

> *Time constant* The time interval necessary for a system to change from one state to another by a specified percentage. For a first order system, the time constant is the time it takes the output to manifest a 63.2% change due to a step input.

MODELING

In this section the characteristics of the DC motor need to be interpreted in a way that allows an appropriate controller to built around it. The List of all the measured parameters, and methods are included in the "One page Experiment design" included in Appendix A.Since the specification only requires a first order model of the motor, and since the mechanical time constant overwhelms the electrical time constant, the mechanical time constant is the only parameter necessary to meet the spesifications.

3.1 MEASURING THE MECHANICAL TIME CONSTANT

With regards to the definition of a time constant as discussed in the Literature study, the time constant of the DC motor was calculated by plotting the rotational velocity of the over time after shutting off the power source. Refer to Figure 1 These Measurements were taken by connecting the Tachometer to the PicoScope and recording the values into a .csv file. The Data was prepared for plotting in the following ways:

- The time stamps were offset so that the shutoff occurred at $t = 0$
- The tachometer gave a reading 360mV when the speed was 0 RPM, so the Y axis was offset accordingly
- The Data was smoothed using an average filter. See python script in Appendix B

From the representation of the data in Figure 1, it can be measured that the motor took Approximately 0.77 seconds to reach a speed of 440RPM, A value 63% from the steady state value. The mechanical time constant τ_ℓ is given by:

$$\tau_\ell = 0.77s \qquad (1)$$

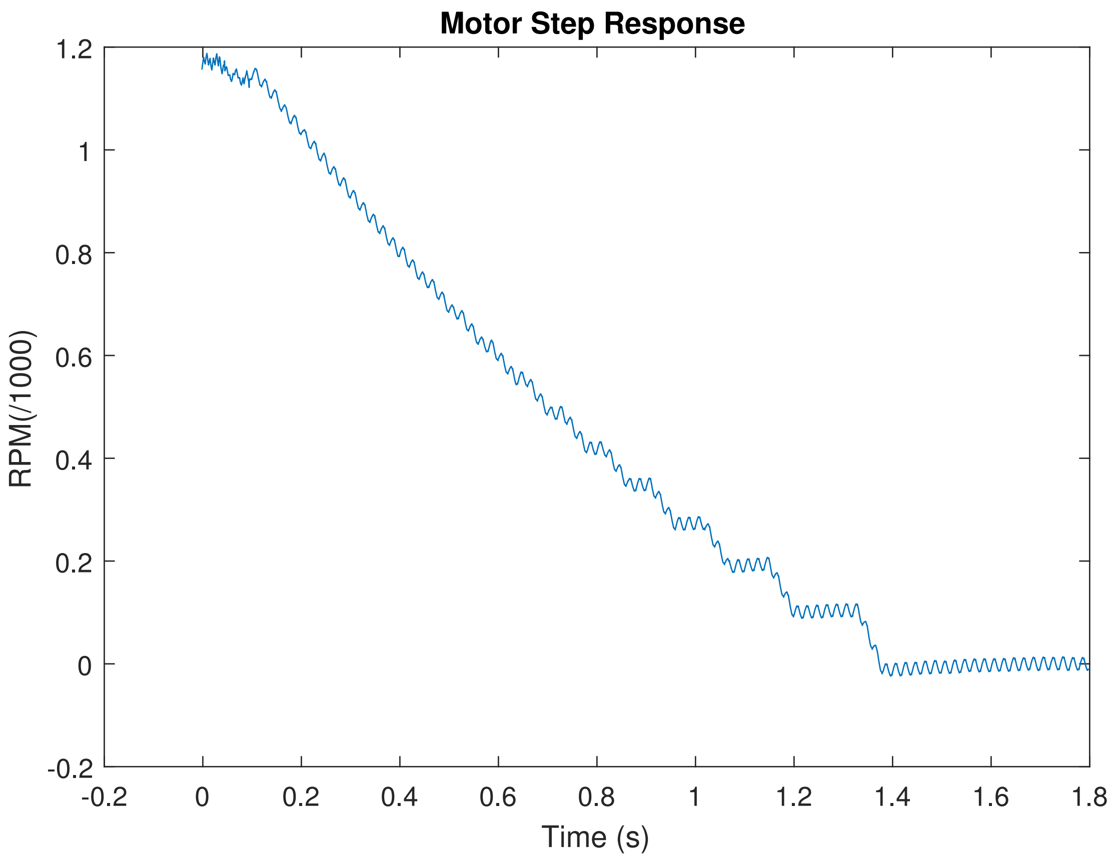

Figure 1: Time Response of the DC motor

3.2 DETERMINING K AND COMPARING THE MODEL WITH THE RESULTS

Having measured the time constant, the time domain model of the DC motor is in the form:

$$\dot{\omega}(t) = Ke^{\frac{t}{\tau_\ell}} \tag{2}$$

Using the value calculated for τ_ℓ and choosing K by means of trial and error, the following model was found to be reasonably close to the measured response:

$$K = 1.2 \tag{3}$$

$$\dot{\omega}(t) = 1.2e^{\frac{t}{0.77}} \tag{4}$$

$$G(s) = \frac{1.2}{0.77s + 1} \tag{5}$$

Refer to Figure 2, where the time domain model is compared to the measured results.

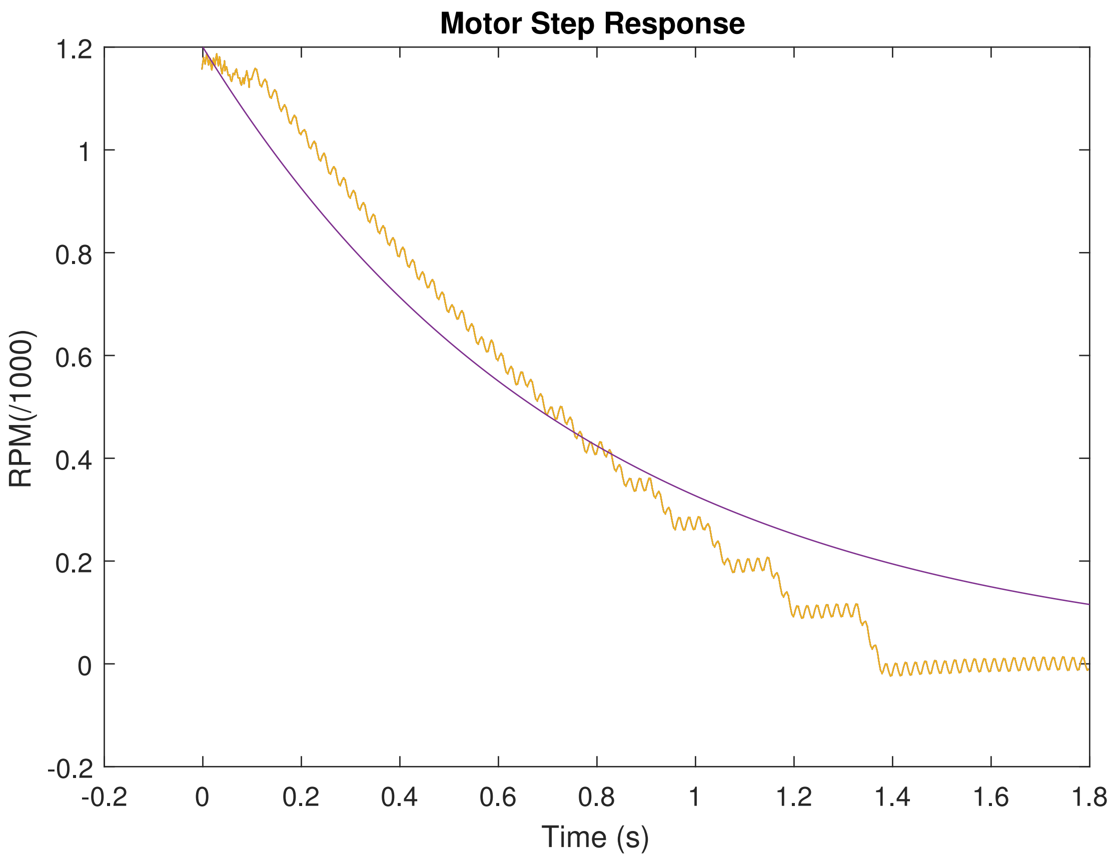

Figure 2: Comparison of time domain model with Laboratory results

3.3 DESIGNING THE PID

The following table shows the specifications that the system must adhere to:

Table 1: Specifications

Parameter	**Maximum Value**
Steady state error	1%
Percent Overshoot(P.O.)	10 %
Settling time	<2s

The transfer function for a PI controller in General is given by:

$$G_c(s) = K_c + \frac{K_i}{s} \tag{6}$$

The equations for Percent overshoot and settling time are given by:

$$P.O. = e^{\left(\frac{-\pi\zeta}{\sqrt{1-\zeta^2}}\right)} \tag{7}$$

$$T_s = \frac{4}{\zeta\omega_n} \tag{8}$$

Furthermore, ζ and ω_n can be used to define a desired characteristic equation in the form:

$$Q(s) = s^2 + 2\zeta\omega_n s + \omega_n^2 \tag{9}$$

The PI controller design can be summed up as the reconciliation of the desired characteristic equation 9 with the characteristic equation of the system, which is given by:

$$Q(s) = G(s)G_c(s) + 1 \tag{10}$$

With respect to the equations 3 and 6

CONTROLLER DESIGN

RESULTS

DISCUSSION OF RESULTS

CONCLUSION

BIBLIOGRAPHY

[1] R. C. Dorf and R. H. Bishop, *Modern control systems*. Pearson, 2011.

APPENDIX A: ONE PAGE EXPERIMENT DESIGN

Table 2: Values to be measured/calculated

Variables and parameters	Symbol and unit
Armature Volatage	$v_a(t)[V]$
Armature Current	$i_a(t)[A]$
Motor Speed	$\omega(t) = \dot{\theta}(t)[rad/s]$
Armature Resistance	R_a $[\Omega]$

The purpose of this lab session is to design experiments to determine the time constants $\tau_a$and τ_l, where:

$$\tau_a = \frac{L_a}{R_a}$$
$$\tau_l = \frac{R_a J}{R_a b + K_b K_m}$$

This will be done using two experiments:

- First, the motor's speed will be measured as it slows down. This makes it possible to determine the moment of innertia and the mechanical time constant
- Second, the Speed will be realted to different values for the armature current and voltage.

These values will be digitally captured and used to determine the nessecary parameters.

APPENDIX B: SOURCE CODE LISTINGS

10.1 PYTHON AVERAGE FILTER

```python
import numpy as np
import csv

reader=csv.reader(open('Motor_start_all.tsv' ,"rb"),delimiter='	')
x = list(reader)
data = np.array(x)
rows = data.shape[0]
arr_time = data[:-(26) ,0].astype(float)
arr_volt = data[: ,1].astype(float)
arr_rpm = data[: ,2].astype(float)

def smooth(y, box_pts):
    box = np.ones(box_pts)/box_pts
    y_smooth = np.convolve(y, box, mode='same')
    return y_smooth

arr_volt_smooth = smooth(arr_volt ,1000)
arr_rpm_smooth = smooth(arr_rpm ,1000)




f1 = open('Motor_start_combined_smoothed.tsv',"wb")
writer = csv.writer(f1 ,delimiter = "	")
for i in range(0,len(arr_time)):
        writer.writerow((arr_time[i]-26,arr_volt_smooth[i], arr_rpm_smooth[i]))
```