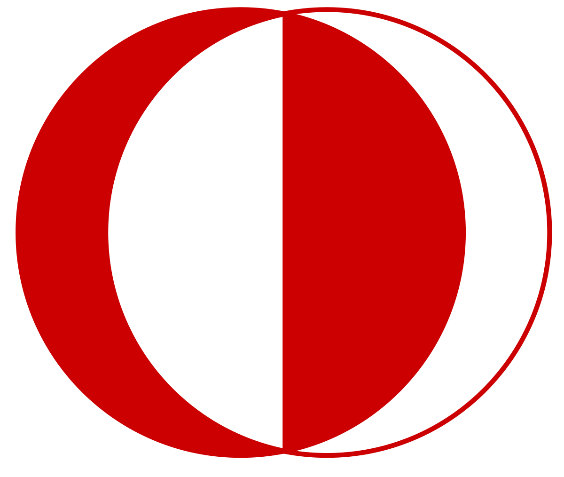
****

**MIDDLE EAST TECHNICAL UNIVERSITY**

**DEPARTMENT OF ELECTRICAL & ELECTRONICS ENGINEERING**

**E463 – PROJECT #3**

**İven Güzel (2030831)**

**05.01.2019**

Table of Contents

[I) INTRODUCTION 3](#_Toc534370712)

[II) QUESTIONS 3](#_Toc534370713)

[QUESTION 1: 3-Phase Thyristor Converter 3](#_Toc534370714)

[QUESTION 2: Buck Converter 3](#_Toc534370715)

[QUESTION 3: Boost Converter (Webench) 3](#_Toc534370716)

[III) CONCLUSION 3](#_Toc534370717)

[*References* 3](#_Toc534370718)

# INTRODUCTION

# QUESTIONS

## QUESTION 1: 3-Phase Thyristor Converter

In this question our goal is to design a proportional-integral (PI) controller that makes the motor speed to follow the reference input speed change by changing the firing angle of a 3-phase thyristor converter. PI controller is used to reduce, or ideally eliminate, the steady-state error between the measured motor speed and reference speed. In Figure Q1.1, the feedback control system is shown.

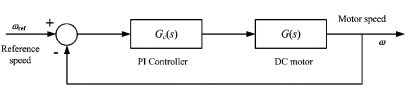


Figure Q1.1: Closed loop feedback control system of DC motor

Transfer function of the PI controller is given by (1). To design a proper controller, the dynamics of the DC motor should be described by a transfer function as well. However, with the help of tuning application methods of Simulink that will not be necessary.

Feedback control system schematics can be seen in Figure Q1.2, plant to be controlled is labeled as DC Motor. Inside the plant can be seen in Figure Q1.3. Upper and lower limits for PI output are included as 90 and 0 to obtain positive voltage mean at the armature terminals.

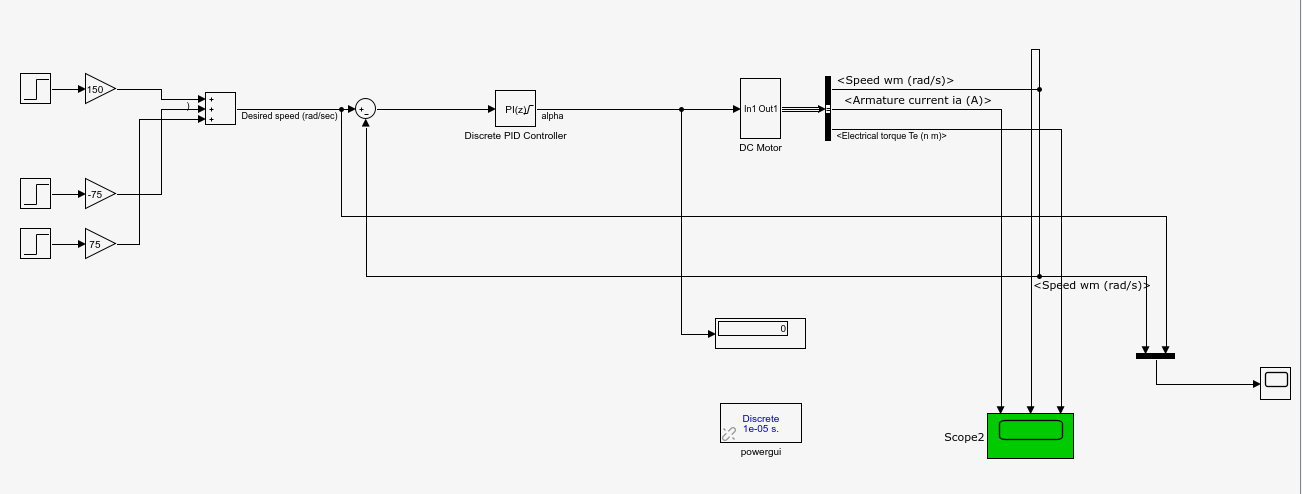


Figure Q1.2: Feedback control system

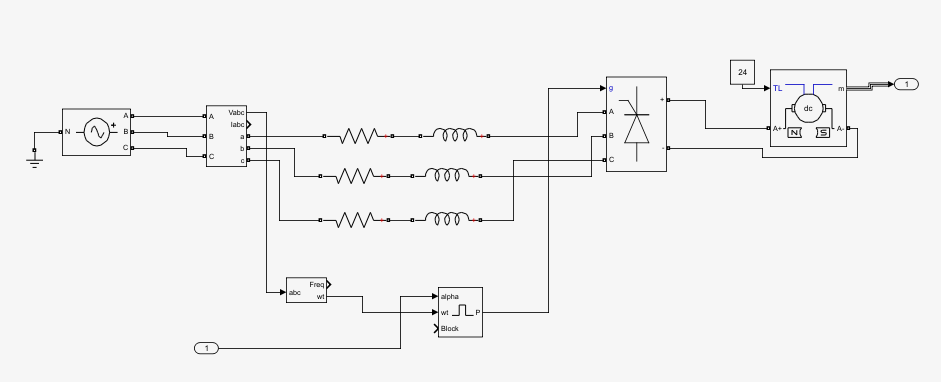


Figure Q1.3: Inside the plant

To model the controller at first, we tried to use Transfer Function Based (PID Tuning) tuning method, which linearizes the plant, calculates PID gains, and opens GUI to adjust respond time however, we could not fix ‘’initial stabilizing controller” error that we received. Therefore; we used Frequency Response Based Tuning Method and found Kp and KI as -2.0474 and -3.7927, respectively. Note that transfer function of the controller is a little different than (1), since it is a discrete time PID controller block. Motor speed and reference speed comparison can be seen in Figure Q1.4, and armature current, speed and torque waveforms can be seen in Figure Q1.5.

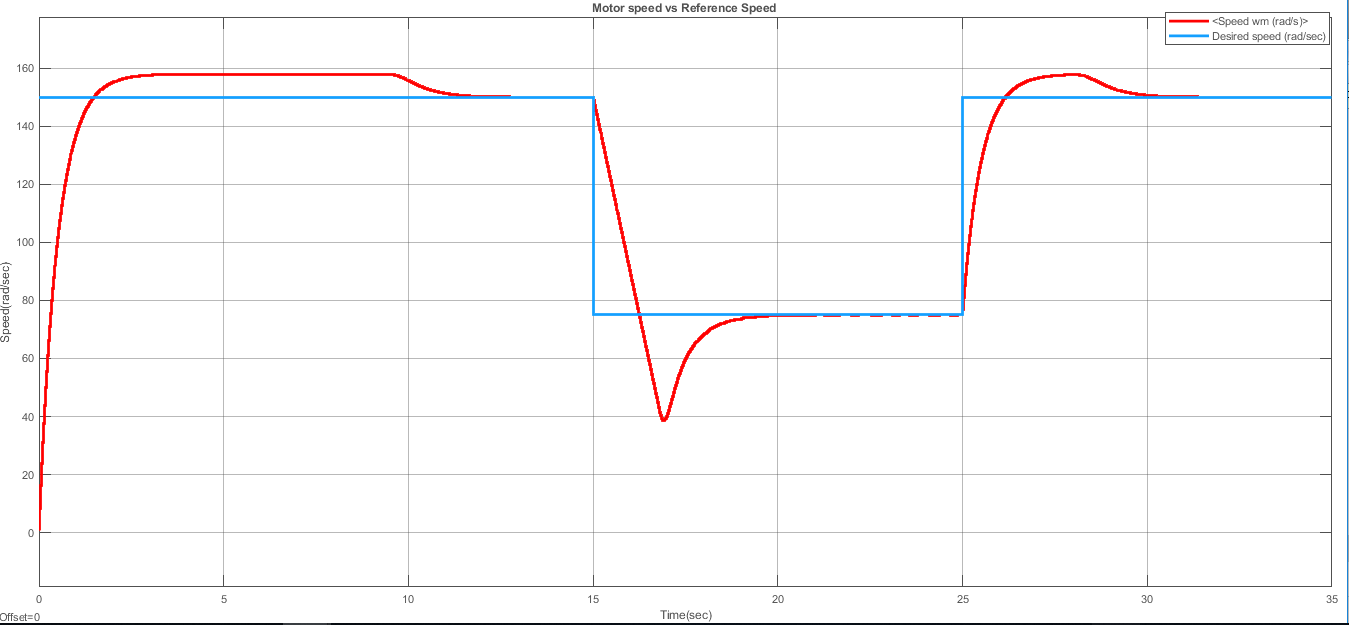


Figure Q1.4: Reference and motor speed comparison

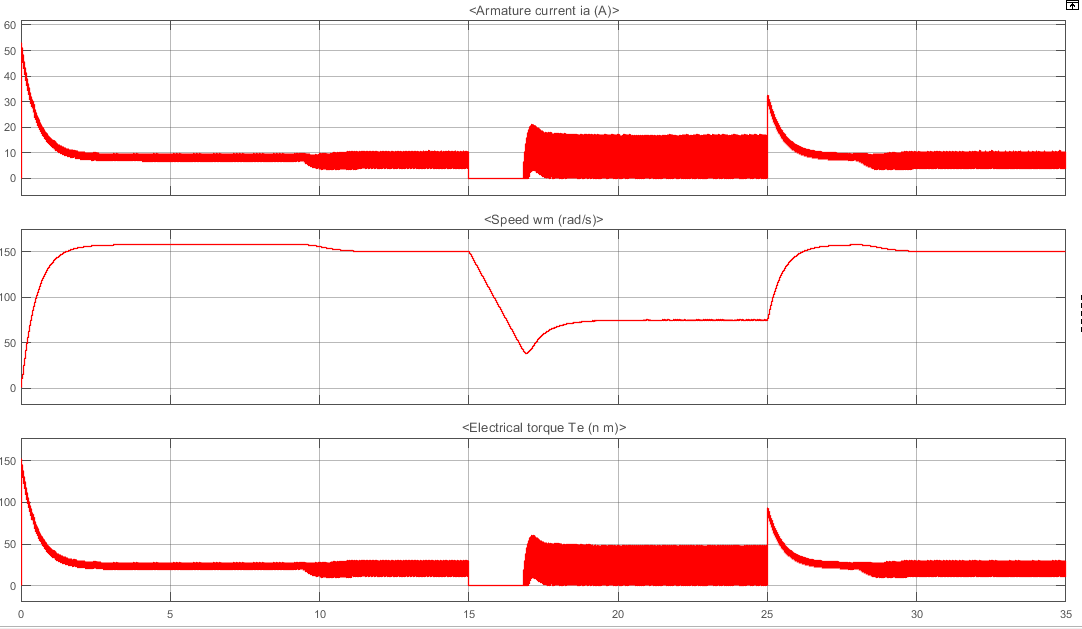


Figure Q1.5: Armature current, speed and torque waveforms

From Figure Q1.4-5 we can see that motor speed reaches zero steady state error eventually due to the integrator in the PI controller. We used various Kp and KI values but the characteristics of these waveform did not change much. However, after only using P controller and PI controller we have figured out that with P controller we cannot reach zero steady state error and with PI controller any amount of KI guarantees overshoot. Also, from Figure Q1.5 we can see that torque waveform persuades armature current waveform oscillations as it was observed in Project 2 as well.

In Figure Q1.4 we can see that motor takes some time to decrease its speed to 150 rad/sec from 157 rad/sec when we first start the motor. Changing Kp and KI values did not help this problem. However, using a different driving technique by gradually increasing the reference speed from zero to 150 rad/sec decreases the starting time from 12 seconds to 8 seconds as it can be seen from Figure Q1.6. Armature current, speed and torque waveforms again repeated for this driving technique in Figure Q1.7.

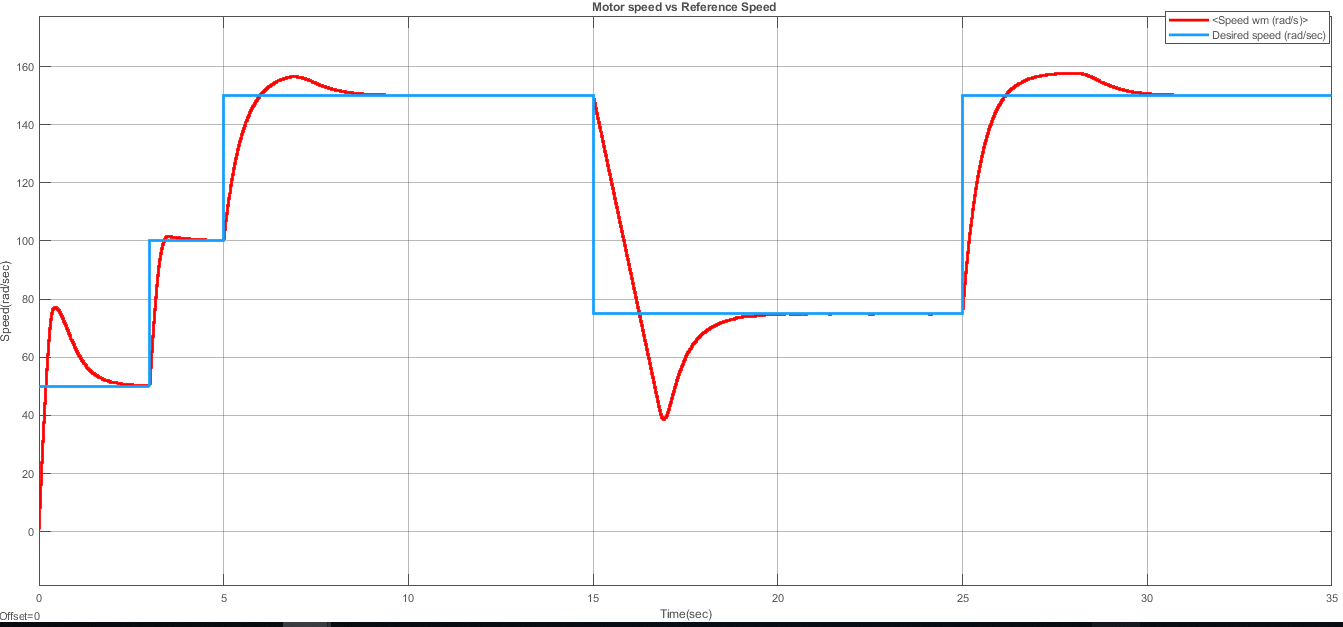


Figure Q1.6: Reference and motor speed comparison

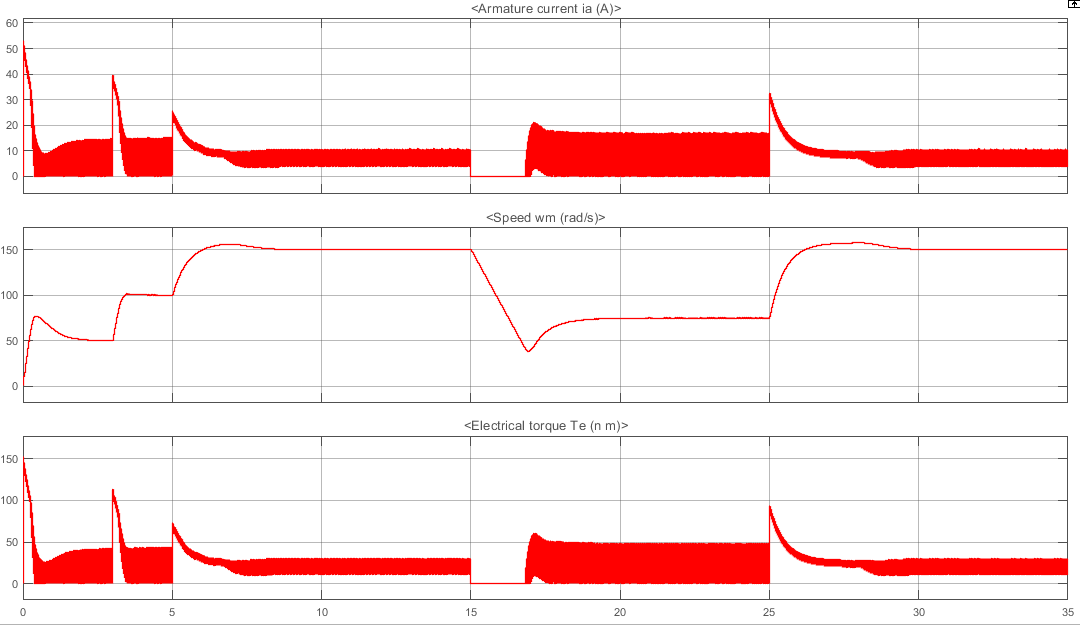


Figure Q1.7: Armature current, speed and torque waveforms

## QUESTION 2: Buck Converter

## QUESTION 3: Boost Converter (Webench)

# CONCLUSION

# *References*