ABDULLAH GÜL UNIVERSITY

Electrical & Electronics Engineering Department

Control System Design

DC Motor Control
PI Speed Control
Report

Mucahit Demirci

110110177

PI CONTROL ACCORDING TO SPECIFICATIONS

In this assignment, as a main goal, we are expected to analyse and learn how to control a DC Motor Control Proportional-Integral process in control systems. Moreover, we are also expected to understand the fundamentals of the PI compensator design according to specifications. We learned the input-output relation in the time-domain for a PI controller.

Speed Response

Damping ratio affects the amplitude of vibrations in our designed system.

$$\zeta = \frac{actural\ damping}{critical\ damping}$$

Natural Frequency

Natural frequency causes to oscillation unless there is damping force, I observed the relationship between these terms to have stable system.

$\omega_0 = \sqrt{\frac{k}{m}}$ is the natural frequency of the system.

Effect of Changing Naturel Frequency

When ζ is zero the system oscillates.

The sharpness of my system depends on the damping ratio, too. In addition, I also see a resonant peak near the natural frequency for underdamped systems when it is 10 rad/s. Simply, large values of natural frequency give large values of control gain.

Effect of Changing Damping Ratio

The effect of changing damping ratio on my system leads to 3 cases; if it is greater than 0.75, it becomes ***overdamped*** whereas if it is between 0 and 1 it behaves as a stable which means ***underdamped***. However, in the system, when I decreased damping ratio, I realised that the system becomes ***undamped***. Therefore, when I increase the value of zeta for 0 to 1 I decrease the damping in response. That is why system response speed up and system becomes stable.

SET-POINT WEIGHT

The set-point weight is used to minimize the overshoot for this DC Motor Control System.

The effect of raising set-point weight

When I increase set-point value, the overshoot increases gradually, too.

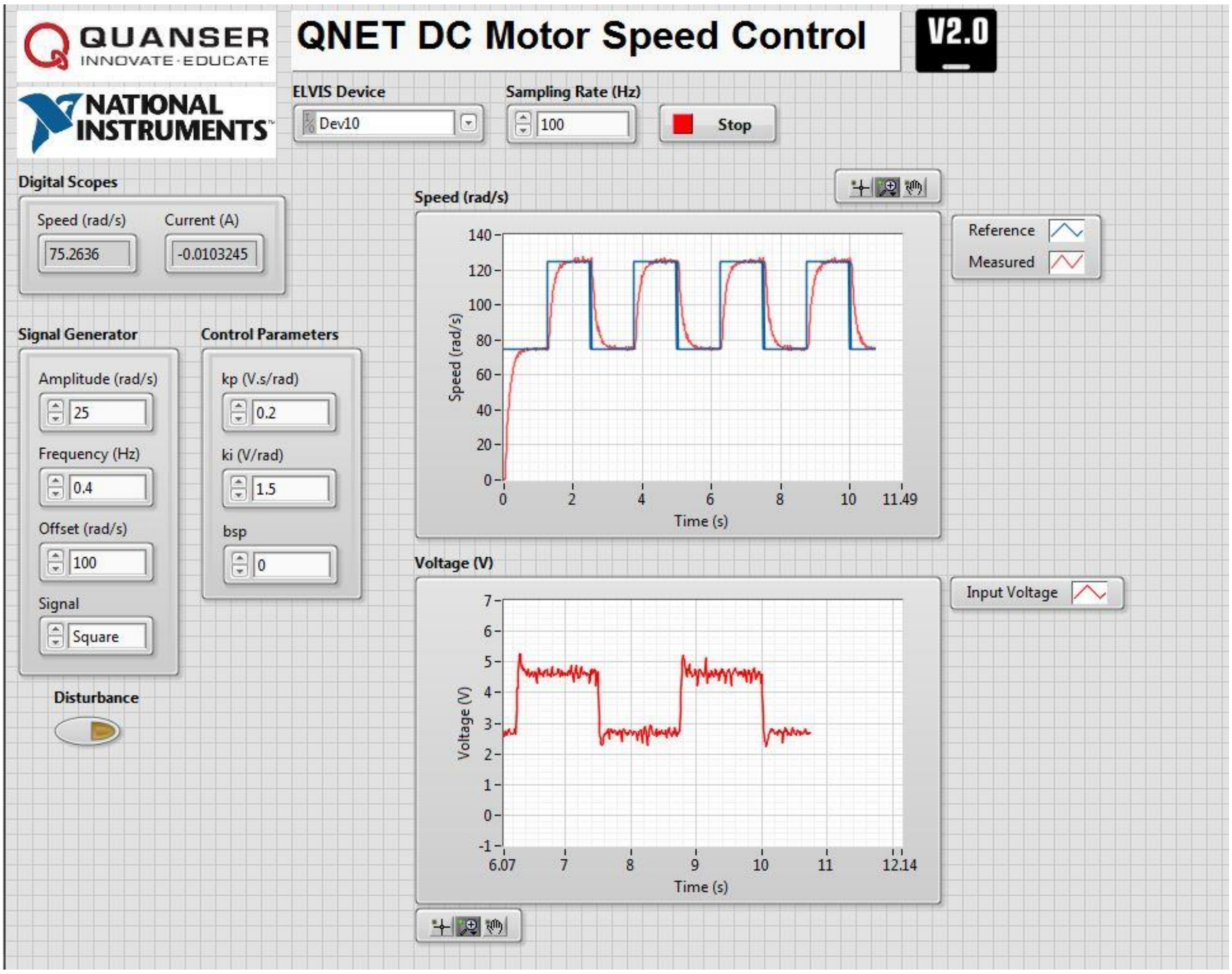

Figure 1 Zero SET-POINT WEIGHT

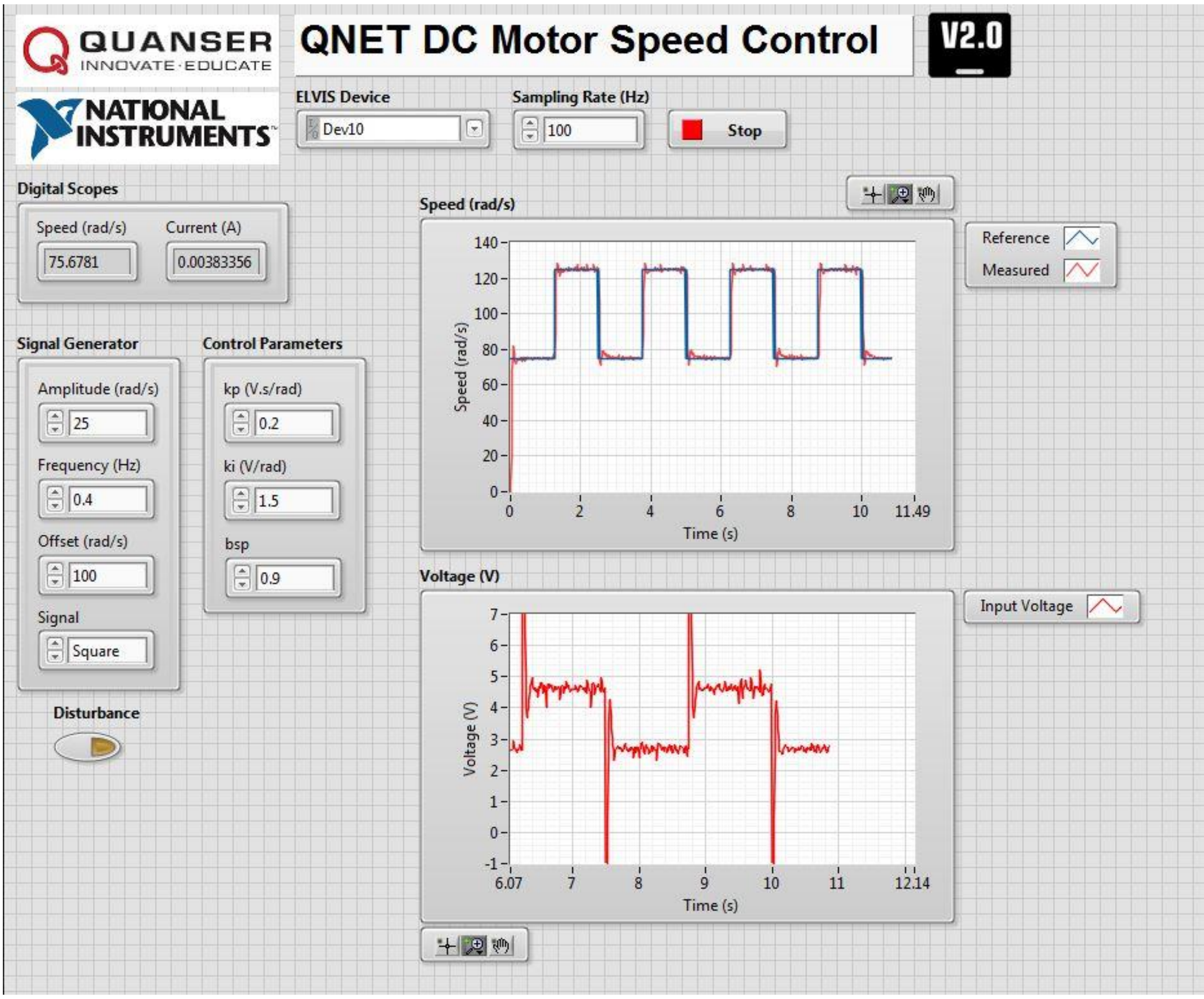

Figure 2 SET-POINT WEIGHT 0.9

RESULTS

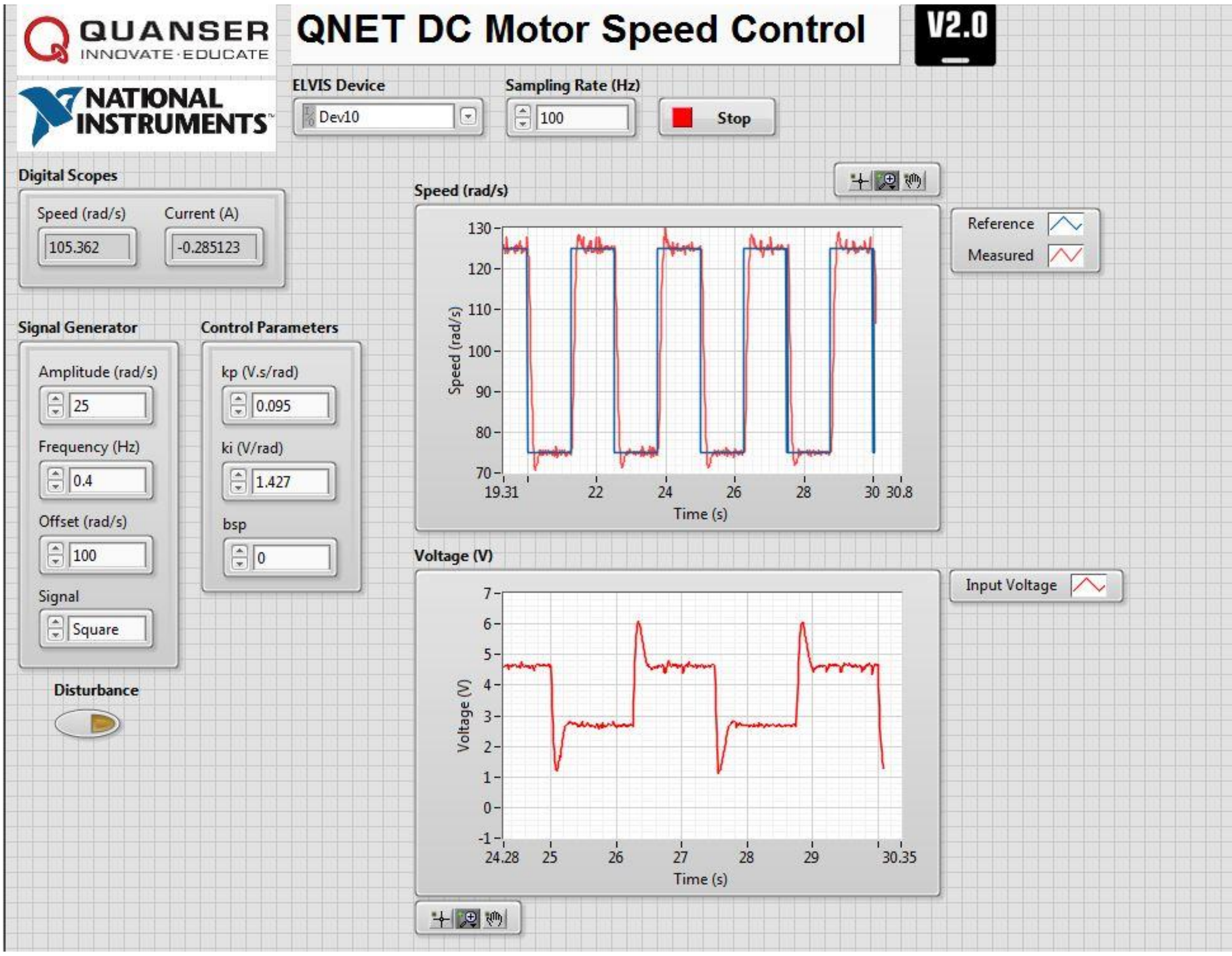

Figure 3 Speed control response plot from step 8

Description	Symbol	Value	Unit
Section 2.1 PI Control Design			
Model gain used	K	26	Rad/s
Model time constant used	τ	0.135	s
Proportional gain	Kp	0.095	V(rad/s)
Integral gain	Ki	1.4276	V/rad
Measured peak time	t_p	0.129	s
Measured Percent Overshoot	M_P	2.83	%

Table 1: DC Motor Modelling Results

ANALYSIS

Calculations for analysis

Percent Overshoot

I have used the formula (0.9) from DC MOTOR PI Control System.

PO = $100 * e^{\frac{-\pi*\zeta}{\sqrt{1-\zeta^2}}}$ = 2.83.

I found the values of Ki and Kp according to the formulas shown in the booklet (1.5, 1.6) Ki = 1.4276, Kp = 0.095

Peak Time

From the initial step time t_0, the time it takes for the response to reach its maximum value is $t_p = t_{max} - t0$.

tp = $\frac{\pi}{\omega n*\sqrt{1-\zeta*\zeta}}$ = 0.29s, tp = 0.129s, measured tp = 0.3s

Effect of changing Proportional and Integral gains

In this experiment, when I increase the proportional gain, the system behaves faster but it can make system unstable. I increased to stop the oscillations since this block reduces the steady state error, however, causes increase in overshoot.

Peak time and Percent Overshoot of Speed Control Response

There is not a big difference between my own calculations and simulation results. So, specifications are mainly satisfied. Therefore, it can be concluded that since there is not an excessive exceeding the maximum value of the step response in step 9. Moreover, having a derivative control would reduce the overshoot in our system.

Effect of changing set-point weight

It can be concluded that in PI Controller, the application of set-point weighting is superior to excessively reduce large overshoot. When I increase set-point value, the overshoot increases gradually, too in step 6.

CONCLUSION

It can be concluded that the desired simulation output and actual motor speed graphs are close to each other. Generally, Steady-state error is simply defined as the difference between input and output of system. Moreover, my system met the specifications on speed controller in Step 7. Furthermore, in my system, steady-state error is calculated using the final value theorem. I observed that steady state error is lower for a step input but higher with a ramp input. To sum up, PI controller is beneficial to reduce the steady state error, but it has negative effects on stability of the system.