EEE4118F 2022
Practical 2: Robust Control System Design

Name	**Student Number**	**Date of Practical**	**Date of the controller Demo**	**Station Number**
Dylan White	WHTDYL001	04/04/2022	25/04/2022	6

1 Introduction

The purpose of this lab was to use quantitative feedback theory (QFT) to design a controller capable of controlling the position of a DC motor in accordance with some user requirements given to us in the lab as shown in APPENDIX B. The transfer functions for this system under various conditions were identified in a previous lab and are included in APPENDIX A

This report details the approach taken to designing, simulating, and testing various controllers to meet these requirements.

2 Setup and Non-linearities

Our chosen controller will be implemented digitally using a series of ADCs to quantify our signals and a DAC with a zero-order hold interpolation on the output. This digital implementation has implications on the design of a controller as described throughout this report.

Our controller was designed for and validated on the DC motor at workstation 6 in the control lab. In our system identification for this plant, we identified that there is a substantial dead-zone nonlinearity for an input range of [-5,4.8] volts as well as a saturation non-linearity at input voltages below -7.55 volts or above 7.51 volts. These non-linearities presented a significant challenge in the implementation of a functional controller.

In addition to the non-linearities, it is noted that we performed the system identification in the linear region in the positive voltage operating region of the DC motor. Due to the nature of physical systems, the transfer function of the motor in the negative operating region will vary from that of the positive region. This was a factor that was noticed in the implementation of our controller and influenced its final implementation.

3 Final Controller: Cascade Controller with Gain Scheduling and Dead-Band Compensation

3.1 Description of final Design

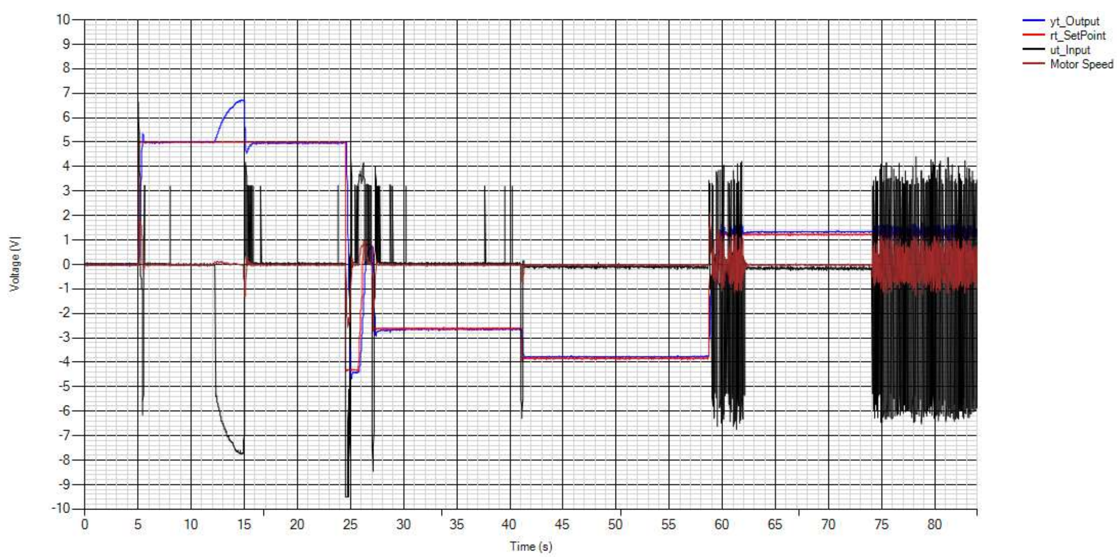
Our implanted controller consisted of a cascaded controller design with an inner velocity control loop and an outer position control loop. This controller was unusual in that it consisted of a simple gain controller in both the inner and outer loop. However, in the inner velocity loop, we wrote a function which attempted (quite successfully) to account for the significant dead band present in the system. Slightly less unusually, we used gain scheduling in the outer, position, control loop which used a different gain depending on the needed direction of the motor's rotation. This was done to overcome the different system behaviour in the positive and negative operating regions.

The performance of our controller is summarised in the following table and image from the laboratory.

Table 1: Summary of final controller performance

Parameter	Symbol	Target	Attained
Speed controller response time	τ_v	$1.2s$	$0.5s$
Overshoot in velocity loop	M_{pv}	25%	30%[1]
Margin for digital design	ϕ_d	$10°$	$8.55°$[2]
Unsaturated step size	y_U	$10°$	$unvalidated$
Overshoot in position loop	M_{pp}	12.5%	8%

Figure 1: Screenshot demonstrating final controller performance



The controller meets all the design requirements. However, our approach to deadband compensation causes controller oscillations at high levels of attenuation as shown above.

---

[1] Simulated only – not validated in lab
[2] Elaborated upon in design explanation

3.2 Design of Cascade Controller

Needless to say, the controller as discussed differs from the one we initially designed. In the design of this controller, we decided to use an Inverse Nichols Chart (INC) to ensure the robust stability of our control system. We would then simulate and implement our controller to evaluate its performance in accordance with the closed loop position requirements as outlined in *Table 1*

We used the MATLAB QFT2 toolbox [1] in order complete the design of the controller by iteratively adding elements to meet the requirements as laid out below.

Another factor to consider is the additional gain and phase lag that arises from using a digital controller with a zero-order hold. To cope with this, we opted to design in the w domain instead of the s domain. To achieve this, we multiplied the transfer functions of our system with "sampling effects" factor: $\left(1+s\left(\frac{T}{2}\right)\right)$. In order to maximise the robustness of our system, we opted to design around a sampling frequency of $T = 0.1$ with the knowledge that we could reduce this if needed.

Our design methodology consisted of using [1] to create templates for our various plants in order to plot nominal design boundaries on an INC where we could perform loop shaping to design our controller. In the loop shaping, it transpired that our critical design frequency happened at $w = 3rad/s$. Our sampling effects factor evaluates to have phase lag of $8.55°$ at this frequency, this is thus our digital design margin. Performing the design on the velocity loop yielded a PI controller with the shaped wave as displayed below.

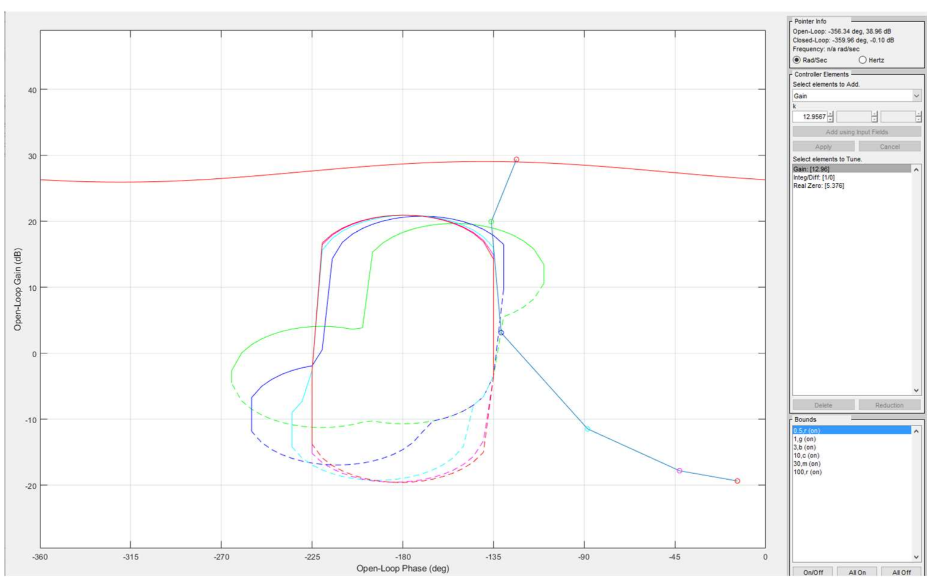


Figure 2: Design of inner (velocity) controller using wave shaping on an INC [1]

Block diagram reduction is then performed on this controller as well as the modified plant to design the position controller using the same technique described above. In this design step we are designing around the nominal bound of the sudo "plant" $P_{nom} = \frac{G_{velocity} \times P \times E_{sampling}}{1+G_{velocity} \times P \times E_{sampling}}$ where $E_{sampling}$ is the effects of the control digitisation as described above. The controller that was designed for position control wound up being a simple lead controller with some gain.

3.3 Simulation of Designed Controller

Our designed controllers can now be simulated using Simulink. The block diagram of the simulated system is shown below:

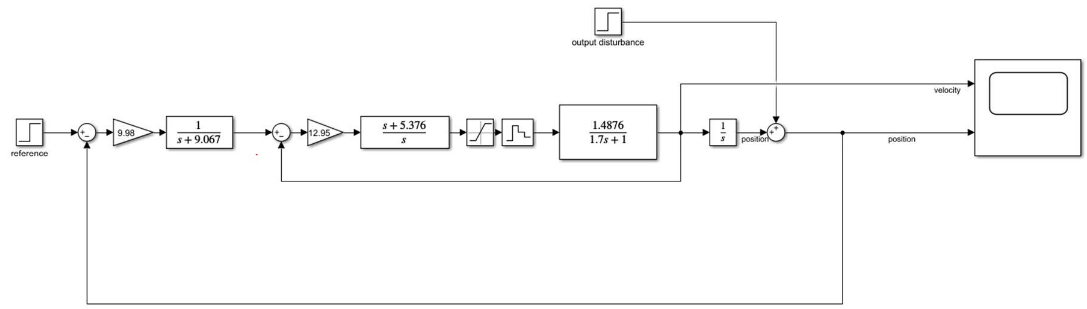


Figure 3: Block Diagram of Cascade Servo Controller

As well as the transfer function and gain blocks used to represent the nominal plant and our controller(s). We used a saturation block and "Zero-order hold" digitisation block at the output of our controller to represent the behaviour of the output DAC of the physical system where these blocks represent its limited output range and sampling rate respectively.

Simulating a step input followed by step output disturbance allows system performance to be predicted:

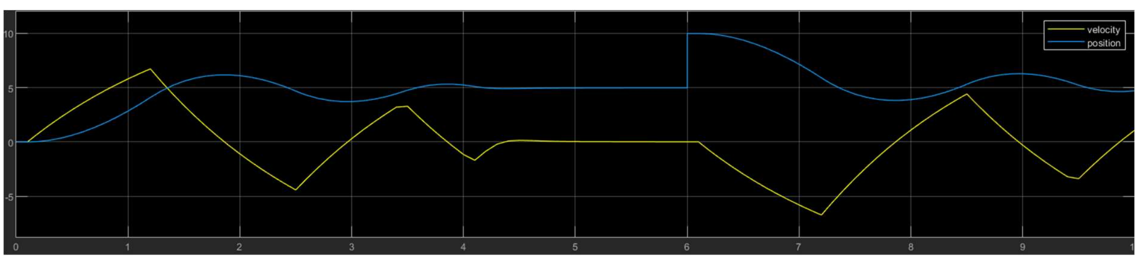


Figure 4: Step Input and Disturbance Responses of Cascade Servo Controller

A step response can also be performed on the velocity loop to complete Table 1: Summary of final controller performance

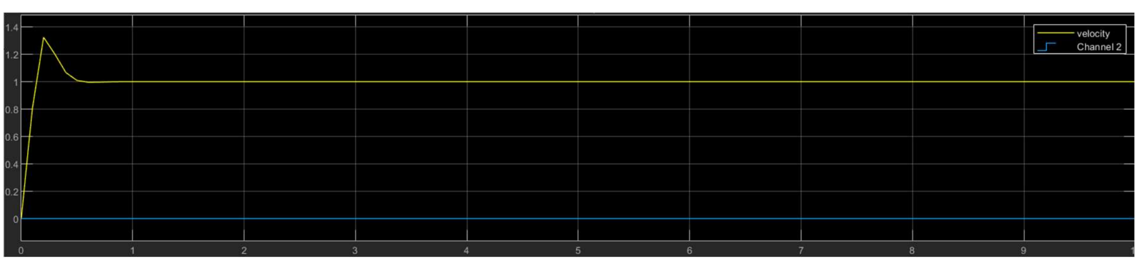


Figure 5: Plot showing Step Response in velocity loop

From these responses we can see that the overshoot in the velocity control causes some oscillatory settling in the position loop. Additionally, the settling time does not meet requirements.

3.4 Implementing a Cascade Controller

In order to implement our controller, we needed to get them in the form of a difference equation. Using MATLAB this is as trivial as calling the *c2d* function to get the transfer functions in the Z domain then performing the inverse Z transform by hand to get the necessary difference equation.

When implementing our controller, we found that it did not perform at all as simulated. The rise time was a lot quicker than predicted but there was oscillation around the setpoint. This was to be expected since we modelled our system as entirely linear, and this is far from the case as described in section 2.

The process of altering our controller to work was closer to an art than science but resulted in a controller as described in section 3.1. A Simulink block diagram of our controller is presented below:

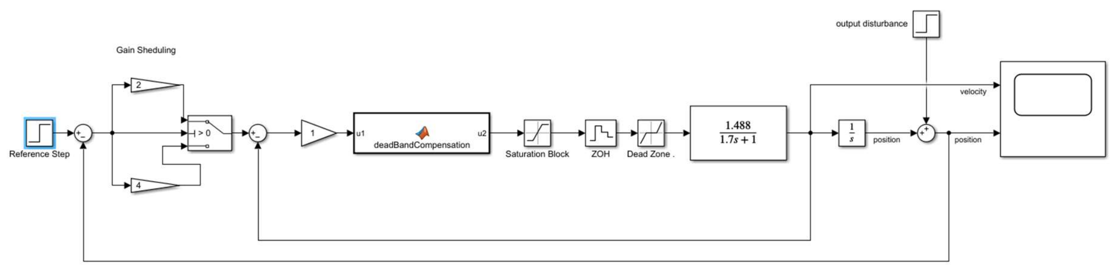


Figure 6: Simulation of implemented controller

The dead zone block attempts to simulate the identified dead zone of the plant dynamics. We only performed system identification in the positive region but if we had performed in both the positive and negative region we could simulate both and allow a separate controller gain to each. As it is, we have simulated the gain scheduling as well as the dead band compensation that we implemented in the laboratory.

The code in the dead band compensator as well as the simulated step responses are displayed in the figure below:

```
function u2 = deadBandCompensation(u1)
    u2=u1;
    if (u1<-0.25&u1>-5)
        u2=u1-5;
    end
    if (u1>0.1&u1<4.8)
        u2=u1+3.1;
    end
end
```

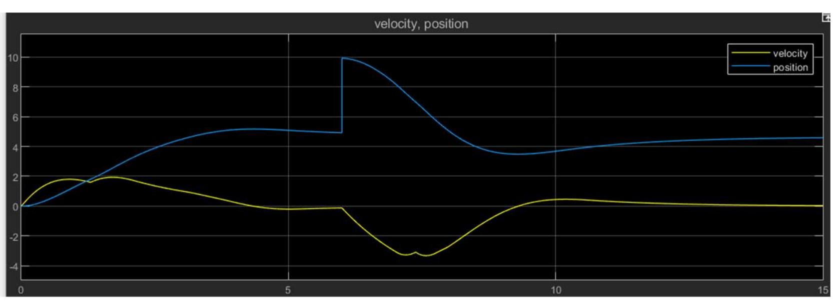


Figure 7: The Dead Band Compensator and the simulated step responses of the implemented controller

4 Lead-Lag Position Controller Designed with QFT

As well as the cascade controller, we attempted to design and implement a lead lag controller designed using INC and QFT methods. This controller is outlined below:

4.1 Process and Robust Stability Design

Much like with the design of the cascade controller, we decided to use an Inverse Nichols Chart in order to ensure the robust stability of our control system. As previously described we used our plant variations (shown in APPENDIX A) as identified in the system identification laboratory. These plants were used with [1] and the specifications outlined below to place nominal bounds on the INC in order to design a lead lag controller.

Our chosen specifications were that the sensitivity of our controlled system must be below -30dB at $\omega = 0.5rad/s$ and below 3dB for all frequencies. As described in previously, when designing on the INC, we need to take account of the additional gain and phase lag effects of signal digitisation. We accounted for this by designing in the w domain as previously described. The INC with the curve of our designed controller is shown below:

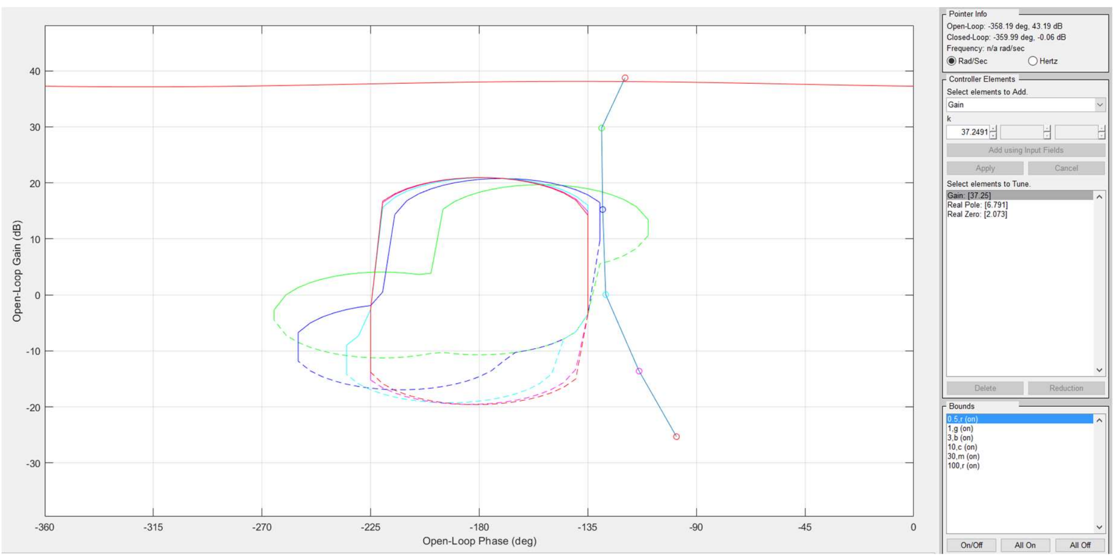


Figure 8: Designing Lead-Lag Controller using QFT wave shaping methods

4.2 Simulation of Lead-Lag Controller

The controller parameters determined above could then be used to simulate the controller. The Simulink block diagram is shown below, the logic behind the chosen blocks has been explained above.

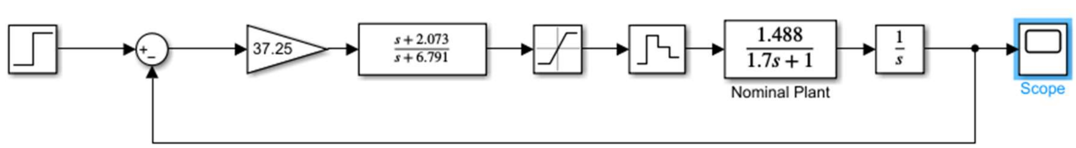


Figure 9: Block Diagram of Lead-Lag position controller

4.3 Results of Simulation

The step response of the controller is displayed below:

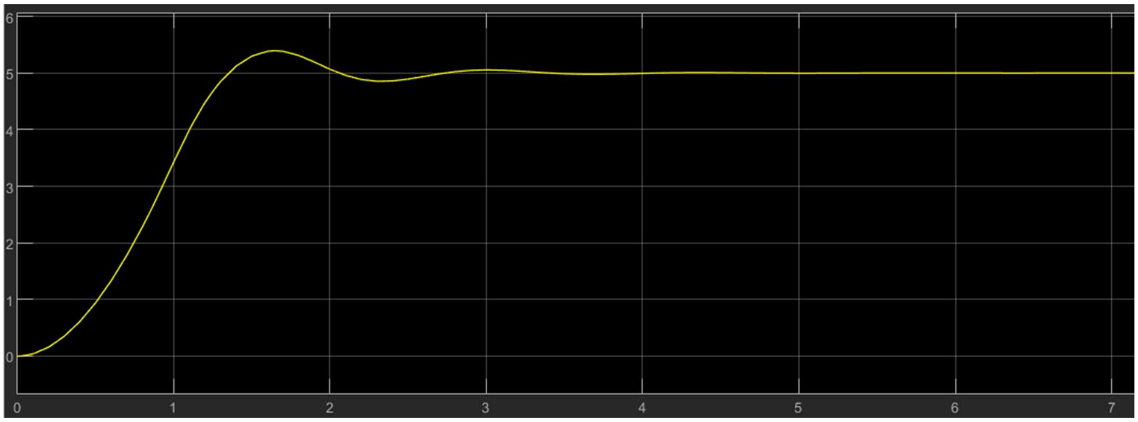


Figure 10: Step response of simulated lead-lag controller

4.4 Results of Implementation

The difference equation describing this controller was determined in the same way as described in section 3.4. However, when testing this controller. We found that the controller oscillated around the set point and did not seem to settle.

4.5 Analysis of Difference between Simulation and Implementation

Once again, we believe the discrepancy between simulation and implementation to be derived from the various non linearities present in the plant as described above. Once again, the most significant of these non-linearities was the substantial dead-band.

5 Conclusion

Above all else, this laboratory work has shown the difficulty that can arise when attempting to control non-linear systems. It has shown that creating a simple linear model of a non-linear system is not conducive to designing an effective controller. Instead, it is clearly necessary to try include these non-linearities in the modelling and especially the simulation process before attempting to implement the designed control solution.

Additionally, this experience has illustrated the importance of robust control design. It has shown that there can be substantial variations in the plant we are trying to control and that even in the modelling of the plant there are inaccuracies that mean we are not working with an accurate model. Thus, it becomes important to design a controller that is robust enough to operate within these plant uncertainties.

6 Appendix A: Transfer Functions of DC Motor

6.1

Plant Conditions	Input→velocity transfer function, $\boldsymbol{P_v(s)}$
Normal Inertia Load	$\frac{1.4876}{s(1.7)+1}$
Normal Inertia Load with Magnetic Braking	$\frac{0.85238}{s(0.35)+1}$
High Inertia Load	$\frac{1.607}{s(11.995)+1}$

7 Appendix B: Time Domain Requirements of Position Controller

Controller Demonstration Marking Guidelines (Total Marks 18)						
Marks	0	1	2	3	4	**MARK**
Set Point Tracking	No Tracking or Large Error	Within 0.6 of Final Value	Within 0.4v of Final Value	Within 0.2v of Final Value	Zero Steady State Error	
Over/undershoot	Unlimited/ Very Large	Within 30%	Within 25%	Within 20%		
Settling Time. No oscillations	>4s OR Oscillations	Within 4s	Within 3s	Within 2s	Within 1s	
Output Disturbance rejection. Return to the previous final value	>4s OR no rejection	Within 4s	Within 3s	Within 2s	Within 1s	
Stability check when plant changes	Unstable	Sustained Oscillations on output		Stable		
Penalty Marks	Controller Output u(t): High amplitude oscillations of > 0.4v 2 Mark penalty	On/Off Bang-Bang Controller Type 8 Mark penalty		Other:		
					TOTAL MARKS	

8 References

[1] *QFT Toolbox*. Terasoft Inc, 2003.