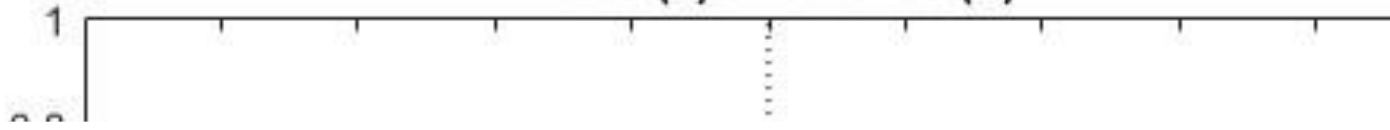


# Poles (x) and Zeros (o)



10/05/2025

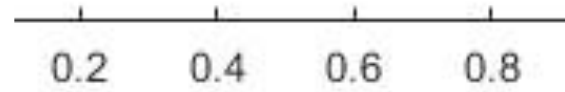
# Digital Control Report

Created by :

Ali Mohamed Aly Mahmoud

EEE Year 3

Cairo University Faculty of Engineering



# Table of Contents

1-Project Objective .....	1
4- Open loop response .....	2
5- PI tuning .....	3
6- Transfer function estimation .....	4
7-Rootlocusfor lead lag compensator .....	5
8-Conclusion .....	7

## 1-Project Objective

The objective of this project is to model a dc motor and design a lead lag compensator to have a Settling time under 2 seconds, OS percentage under 5% and steady state error within 3 percent

## 2-System Description

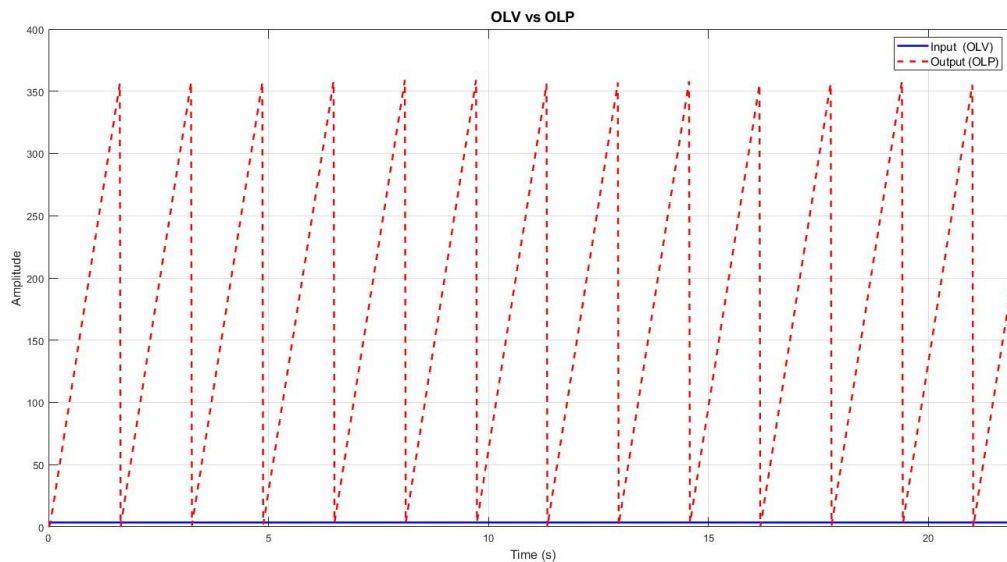
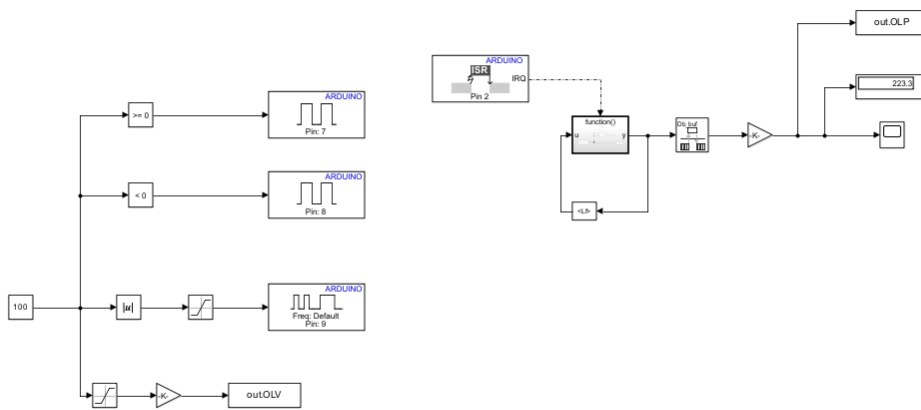
The system is comprised of a DC motor, the plant, with a gear ratio of 1 to 374, with a gear ratio of 34:1. The system is driven by an H bridge driver the TB6612FNG, a modern mosfet based driver with low voltage drop. On the motor is an 11 count quadrature encoder, giving 11 counts per revolution plus quadrature direction identification. Finally, we use the arduino mega for driving the system.

## 3- System Specification

Component	Specification	Description of role
DC Motor	12 v rated voltage 0.3 A rated current 34:1 gear box reduction	Plant of the system to be controlled
Encoder	11 counts per rotation, with reduction given 374 counts per rotation	Necessary for position data
Arduino mega	-	More than sufficient for running and controlling the system, includes sampling and ZOH
TB6612fng H-bridge	Dual H-bridge motor driver (Toshiba) VM: 4.5–15V   VCC: 2.7–5.5V 1.2A cont / 3.2A peak per channel 2x DC motors or 1x stepper Modes: CW, CCW, brake, stop PWM: up to 100kHz Protection: UVLO, thermal shutdown Package: SSOP-24	Converts commands from arduino into power voltage for motor.
Power supply	9v 2 A max	Supplies power for system
Simulink	.02 T sampling time	An adequate sampling time for the given system

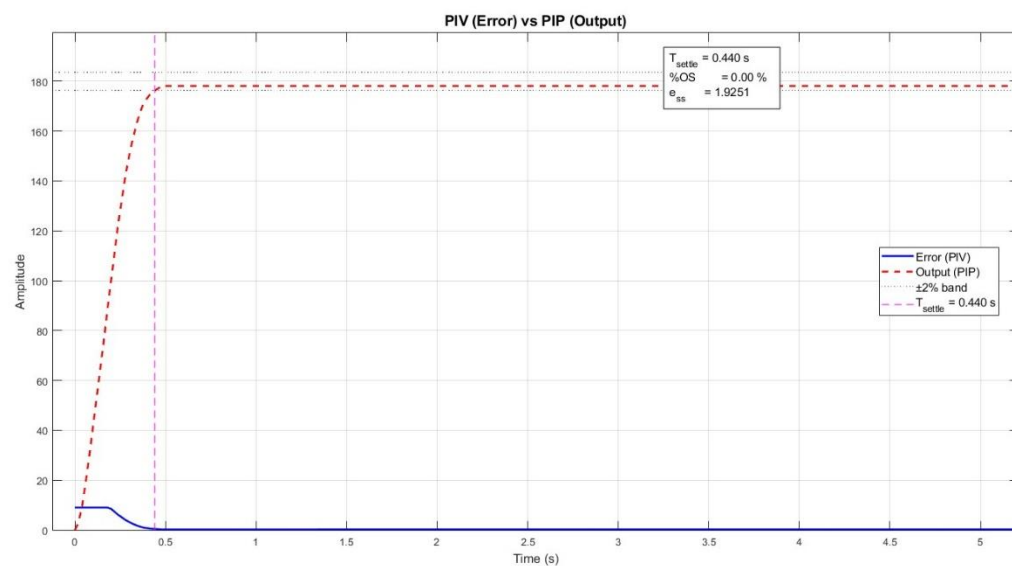
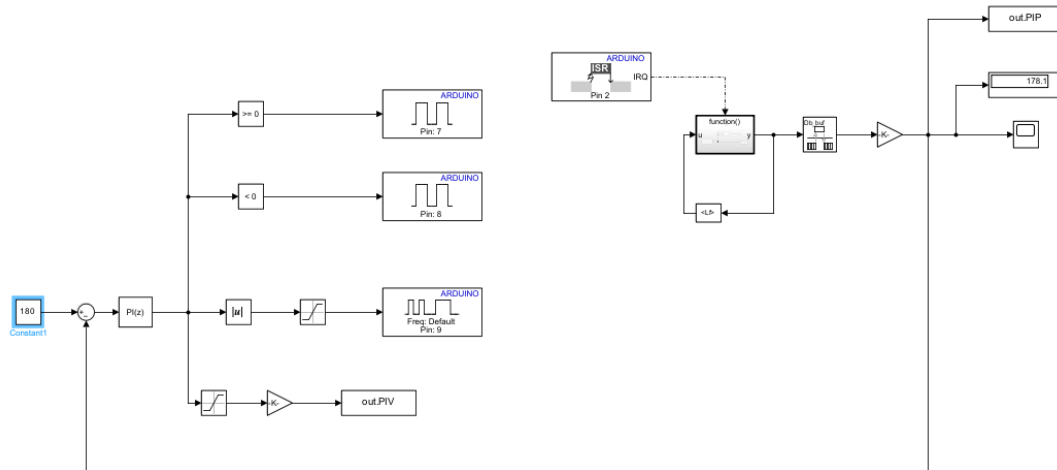
## 4- Open loop response

We define the output position to be cyclic, as in cycling from 0 to 360 degrees instead of being absolutely defined around 0. By doing so, the open loop system's step response is critically stable, instead of unstable. Because the open loop system naturally has no restoring force (spring) it is only stable with feedback. Below is the open loop response, where we observe the oscillating sawtooth wave



## 5- PI tuning

Before Identifying The systems transfer function to properly design a lead lag compensator, we may attempt to tune the system manually using a PI controller. For a system of this type, its intuitive that the system type is 1, as it has a pole at 1 making it critically stable (due to the position definition). As such, an I term is theoretically not necessary, but is added due to non linearity of a real dc motor like the one used.

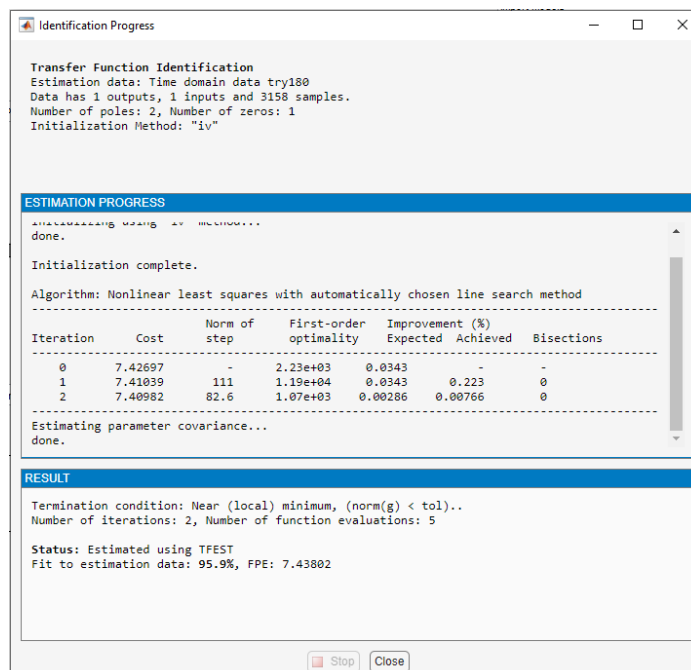
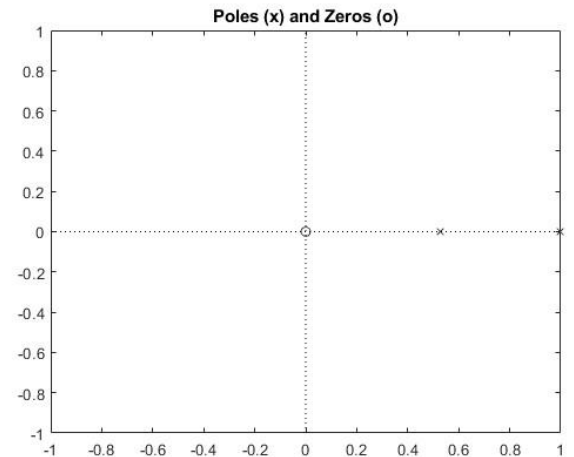
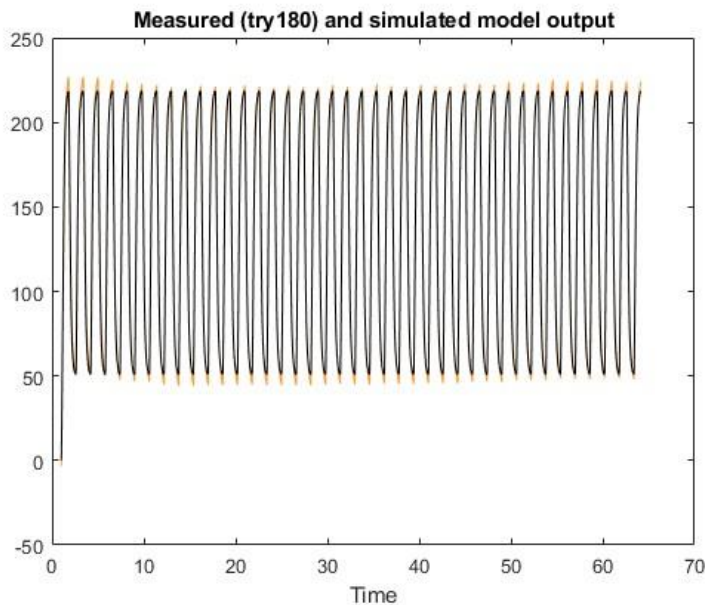


## 6- Transfer function estimation

The transfer function was estimated using the MATLAB system identifier. A transfer function of fit 95.83% was found, which is a very strong fit indicating an accurate transfer function.

$$\frac{\theta}{V} = \frac{.6411z}{z^2 - 1.525z + .5254} = \frac{.6411z}{(z - .999156)(z - .52584)}$$

The pole at .999156 is to be considered stable, and the reason its not exactly at 1, is because of statistical aberrations, however as discussed previously the open loop system is critically unstable. The pole at .52584 is problematic, as it indicates a very fast system. However, due to the very significant fit, we proceed with this model as if it were accurate.



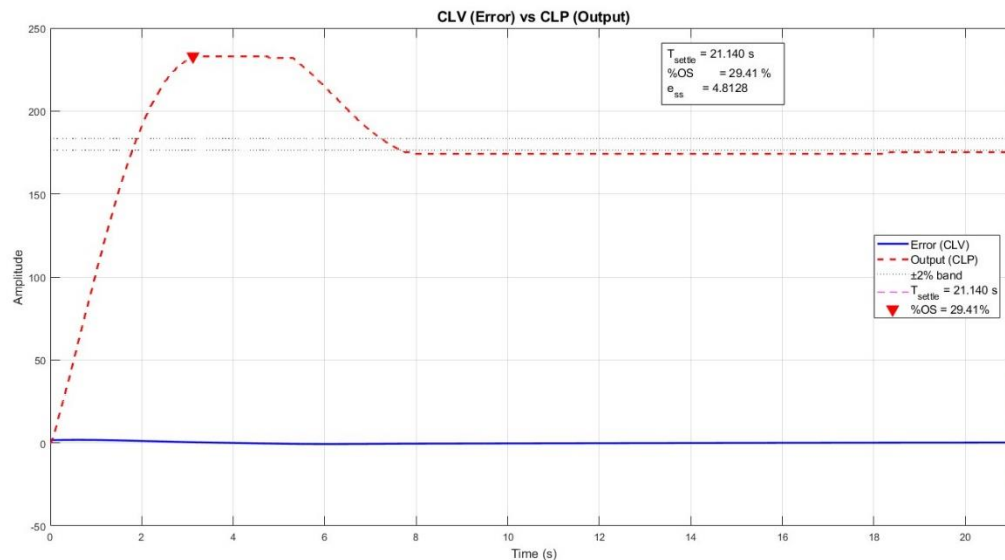
## 7-Rootlocusfor lead lag compensator

The required Parameters of  $T_s$  of 2 seconds or less, OS% of 5% or less, and ess of 3% or less in theory are abysmal relative to our acquired transfer function of fit 95.83%. Since we humour the transfer function, we must acquire new parameters that push our system, so we choose settling time of .175s, 1.15% and ess of 3%. the corresponding  $z^* = 0.6 + 0.203j$

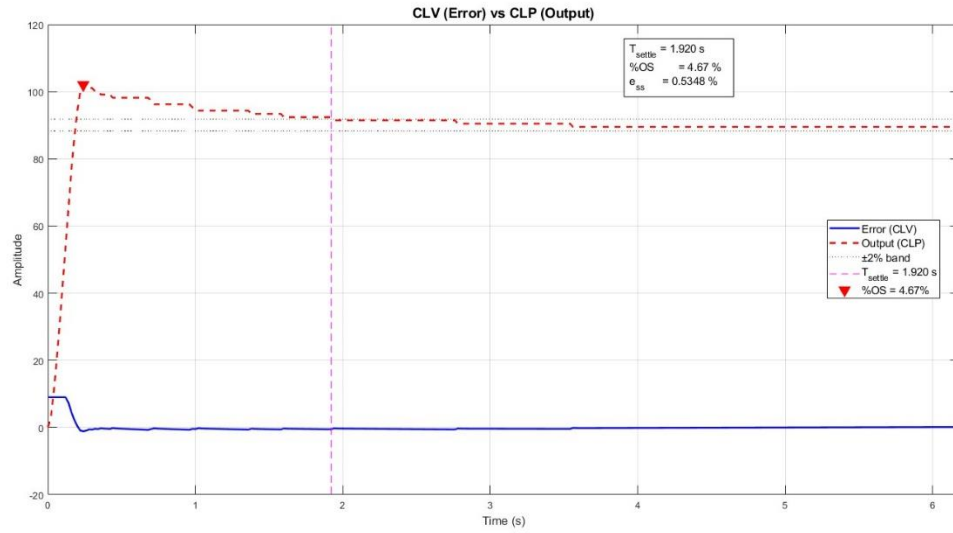
TO design the lead compensator, we place our alpha zero at .525 to cancel the slowest stable pole since we consider .99915 unstable. Then, through the angle condition using matlab, we compute betas angles and respectively its value of .4013. From the magnitude condition, we find K to be 0.32.

Now we notice this K value is tiny, and this is because this K value is prior to Arduino normalization. Multiplying it by 255/9 yields  $K=9.067$  which is the correct value.

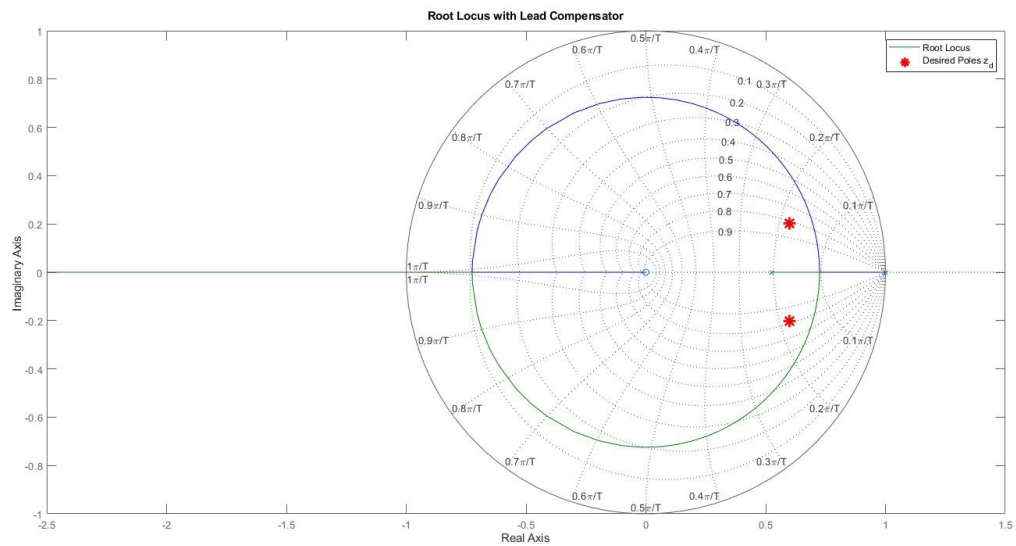
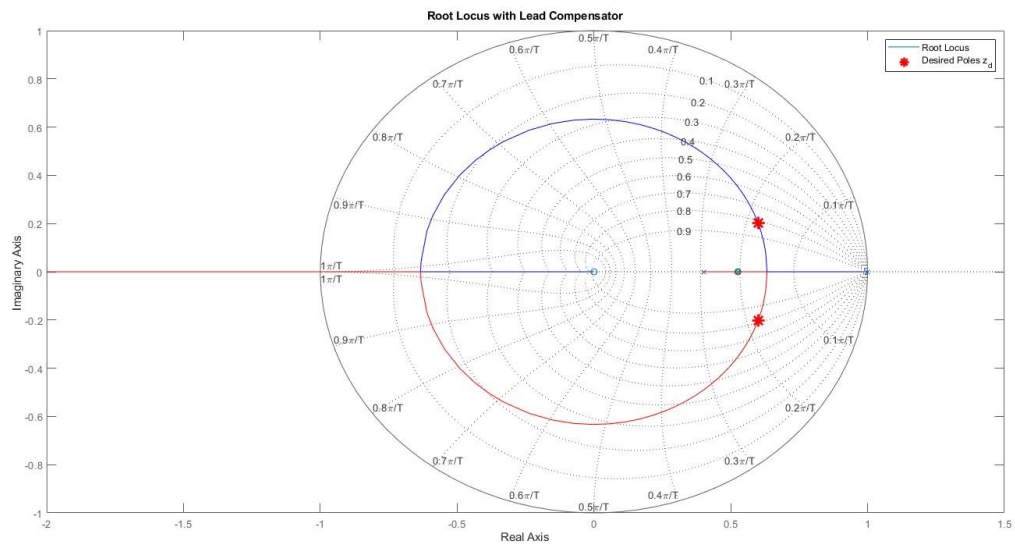
To limit ess, (despite type 1 system), we add a lag compensator of  $\frac{z-.98}{z-.999}$  which satisfies the ess condition with excess.



CLV and CLP without K normalisation shows faulty response



With K compensation, Lead lag compensator meets specification



## 8-Conclusion

The settling time and overshoot and steady state errors match the expected specification due to respecting the theory of modelling and the beautiful 95.9% fit. Notably, these values are comparable to the untuned PI controller, showing the value of modeling the system prior. In conclusion, This report highlights the process of designing a lead lag compensator for a DC motor to achieve some required parameters.