NEW YORK CITY COLLEGE OF TECHNOLOGY

CITY TECH

Department of Computer Engineering Technology

Final Prototype Design

Course:
CET 4864 Principles of Feedback Control Systems
[redacted]
Semester: SPRING 2023
[redacted]
Final Project
Student: RUYEL RODRIGUES
[redacted]

Fall 2023
Date: 12/20/2023

Table of Content:

Objective

The objective of this laboratory is to improve the output response of the control system provided by a customer. The prototype provided will be used to derive a solution that can be tuned for a desired outcome. The goal is to create a closed loop system with fine tuned controllers that produces a step response with minimal error and an improved steady state response.

Technical details:
The prototype provided displays an RLC circuit that the output taken as the voltage across the Capacitor. Given the 5 volt DC input, a typical capacitor charges exponentially up to the voltage value and then does not continue any other function. Meaning it acts as a short in the DC series system. The RLC series circuit does the same for voltage across the capacitor but in a slower fashion, as the system shows that it's overdamped.

The random nature of values assigned for each component and the fact that they don't follow convention such as having capacitors with microfarads and inductors with millihenries. The output of the system spans the time scale in an odd manner. as in what should take a few milliseconds since step input takes more than 10, 50, 80 seconds to achieve.

This issue of overdamped response can largely be mitigated using a unity feedback controller and a proportional controller to underdampen the response. The error from the system can be used to create a response inversely proportional to the output. Meaning the error is highest at start and reduces over time to 0 as the output reaches the step input value.

Using laplace transform we can convert the time domain function to better come up with a solution. The laplace or S domain is an excellent and effective way to derive our solution since the output response is ultimately independent of the time series. May it be millisecond or a millennium, the solution produces a Linear Time-Invariant system.

To measure the success of a solution we can observe some factors like the rise time, peak time, setting time and the overshooting percentage or OS%. These are indicators of what the system is doing to a step response, and our objective is to match these responses to our criteria. That way we are sure that the solution is effective and the system is producing the desired response.

Theoretical Background

As a control system engineer customers will often request that we improve their product in some ways. Our job is to understand the system and make the necessary additions to improve the performance of the system to the desired outcome.
The requirement for our experiment has the following description:

- The original prototype demonstrates a delayed damped response with a significant steady-state error, indicating that the system's current performance does not meet the desired criteria.
- It is an open-loop system with no feedback or control system in place.
- The goal is to design a closed-loop system to address these issues and improve the system's response and performance.
- Additionally, utilize the components assigned in Fig 2 turning it underdamped and then stabilizing the system with a PID controller.
- Perform the design process outlined in the next section, step by step to complete the objective.

The motivation for this laboratory is to assist the customer with his prototype and make the improvements as outlined above.

Detailed Description Lab procedure

Step by step Procedure

Part 1.

a. Open Loop Block Diagram

Draw the open loop block diagram of the system, including all components and their interconnections.

b. Time Domain ODE

Apply Kirchhoff's voltage law to derive the time domain ordinary differential equation (ODE) for the original physical system.
Express the ODE with respect to the output voltage Vc(t) across the capacitor, given a 5-volt DC power supply as the input.

c. Transfer Function

Use the ODE to derive the transfer function of the system.
Express the transfer function in terms of the Laplace transform of the input and output signals.

d. Poles and Zeros

Find the poles and zeros of the open loop system from the transfer function.
Plot these poles and zeros in the complex plane to visualize their locations.

e. Inverse Laplace Transform

Utilize partial fraction decomposition to express the transfer function in a form suitable for inverse Laplace transform.
Calculate the inverse Laplace transform to obtain the total output response of the system in the time domain.

f. Final Value and Steady State Error

Determine the final value and steady-state error of the system.
Analyze the system's behavior as it approaches a steady-state condition.

g. MatLab Plotting

Use MatLab to plot the system's response based on the derived transfer function and input signal.

Part 2.

a. Controller Tuning

Utilize the closed-loop system in Figure 2 to tune the system's controllers until an underdamped response is achieved.

b. Controller Staging

Stage the controllers and showcase their values or functions.
Explain the rationale behind the chosen controller settings.

c. Closed Loop Transfer Function

Derive the new closed-loop transfer function of the system with a unity feedback controller.

d. Poles and Zeros

Plot the poles and zeros of the closed-loop system to understand its stability and dynamic behavior.

e. PID Control Implementation

Implement PID control to generate a stable underdamped response.
Detail the PID controller settings and their impact on the system's response.

f. Contrast of PID Control

Compare and contrast the effect of PID control on a stable and an unstable response.

Provide a detailed explanation and conclusions based on the observed differences.

g. Total Output Response

Calculate and present the total output response of the final stable underdamped system.

h. Response Characteristics

Calculate the rise time, peak time, settling time, and percent overshoot (OS) of the underdamped response.
Display all relevant calculations.

i. Damping Ratio

Determine the damping ratio of the response to understand its oscillatory behavior.

j. Natural Frequency

Calculate the natural frequency of the underdamped response to further characterize its dynamic properties.

k. MatLab Plotting

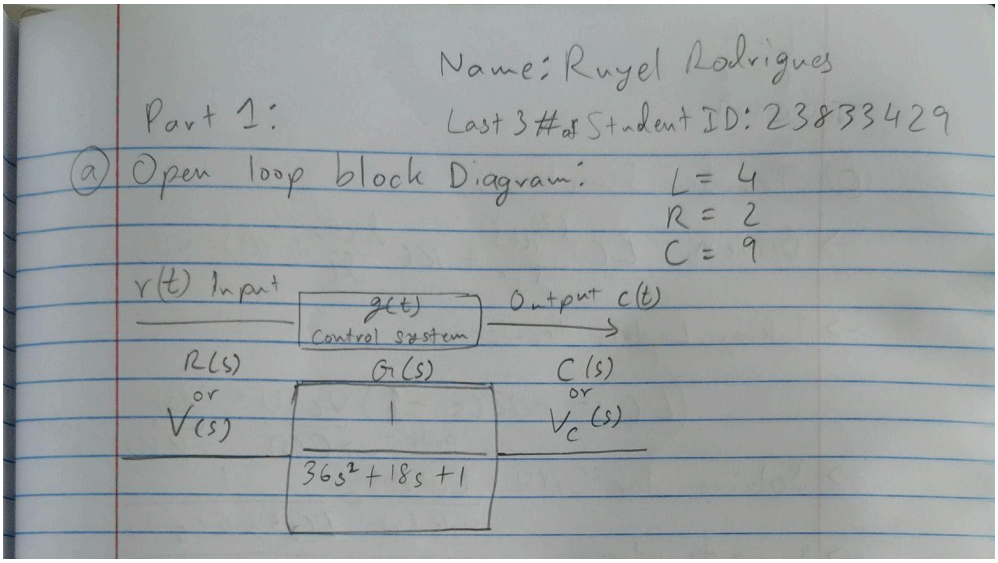
Utilize MatLab to plot all the responses and visualize the system's behavior under the PID control.

Data and Information Tables

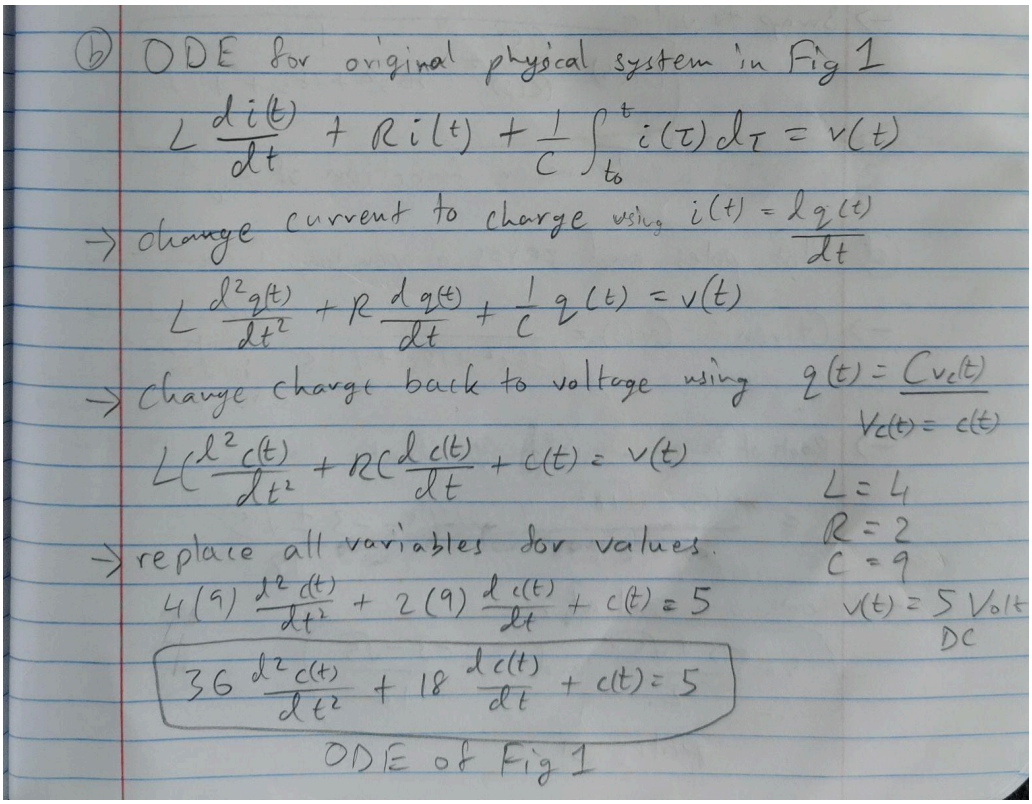
Each of the letters below follow the methodology discussed in the previous section. They are letter coded and by part for easy access. Please consult the table of content by Part then section to navigate. Mathematical equations are included in written form since MS Word editor was not used for the report.

PART 1.

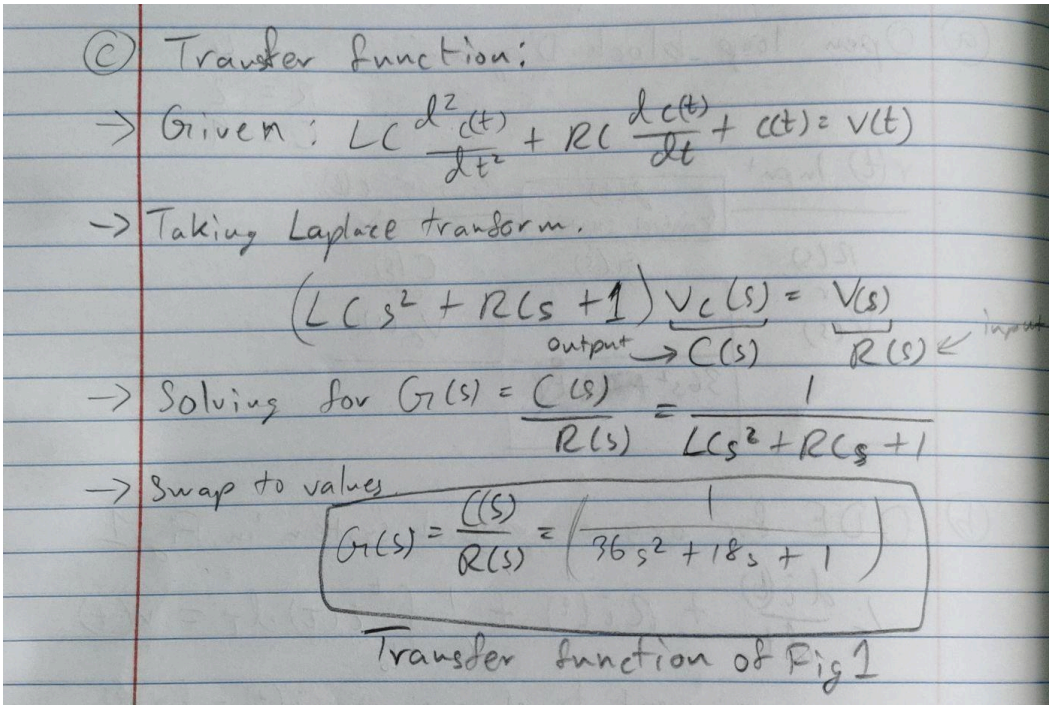
A. Open Loop Block Diagram



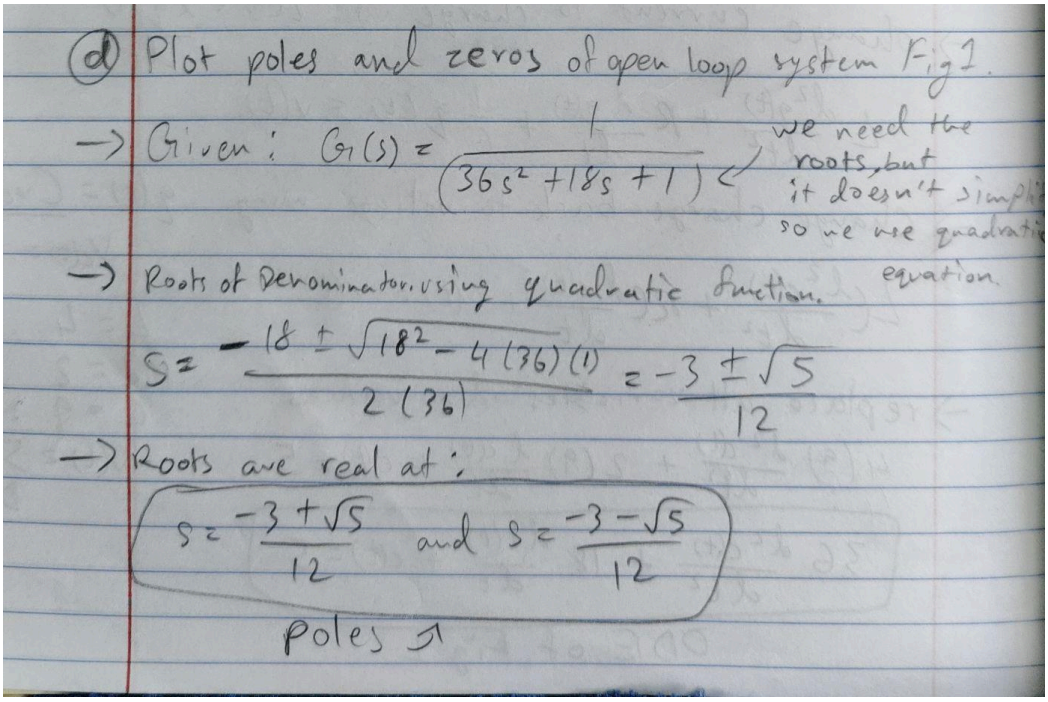
B. Time Domain ODE

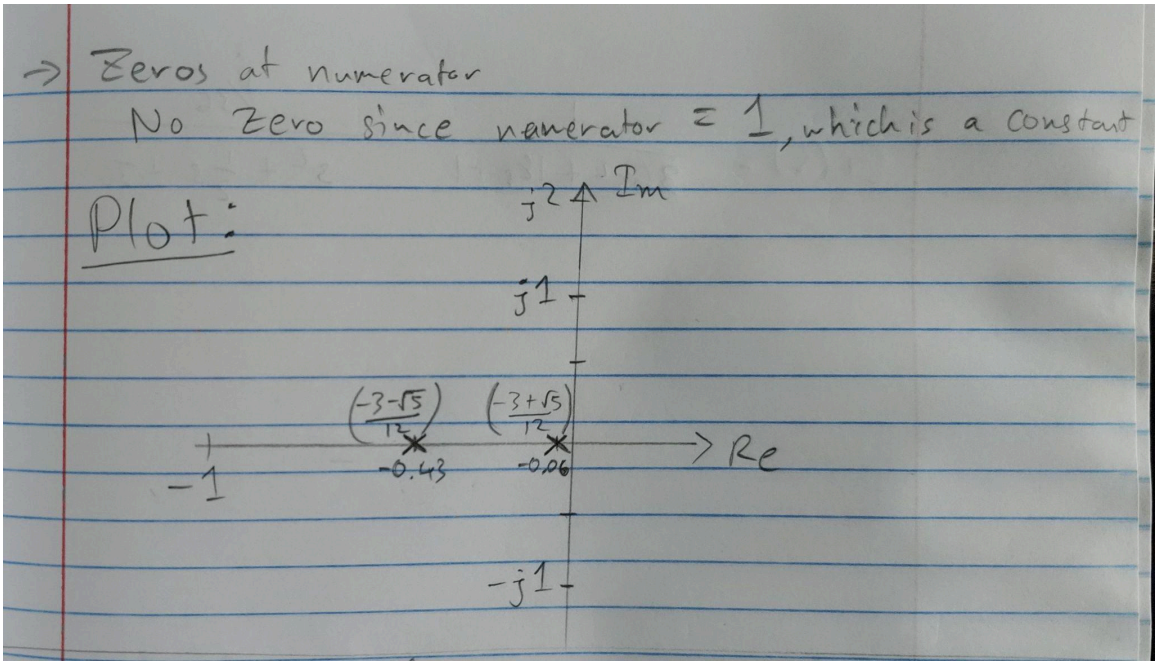


C. Transfer Function



D. Poles and Zeros

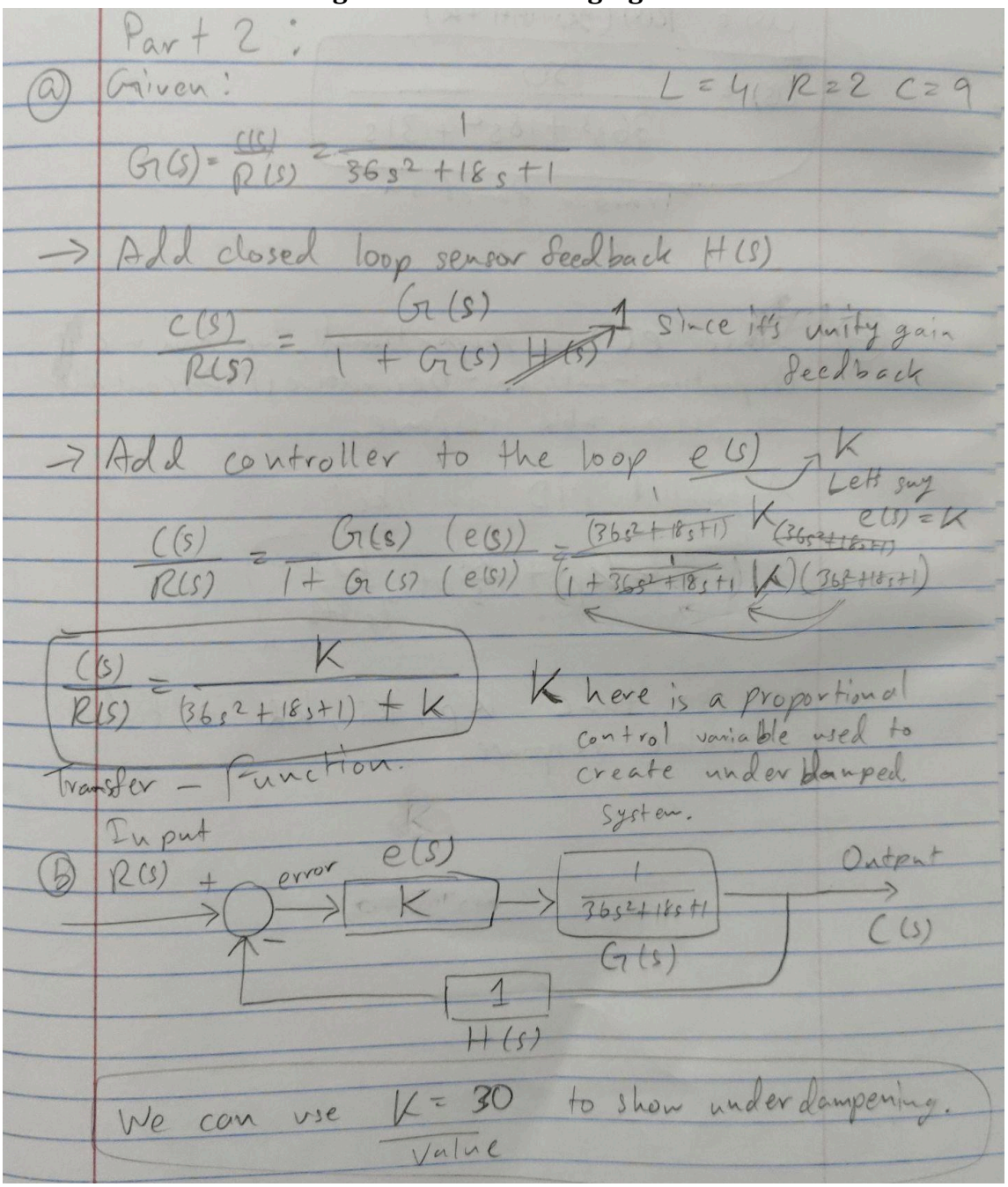


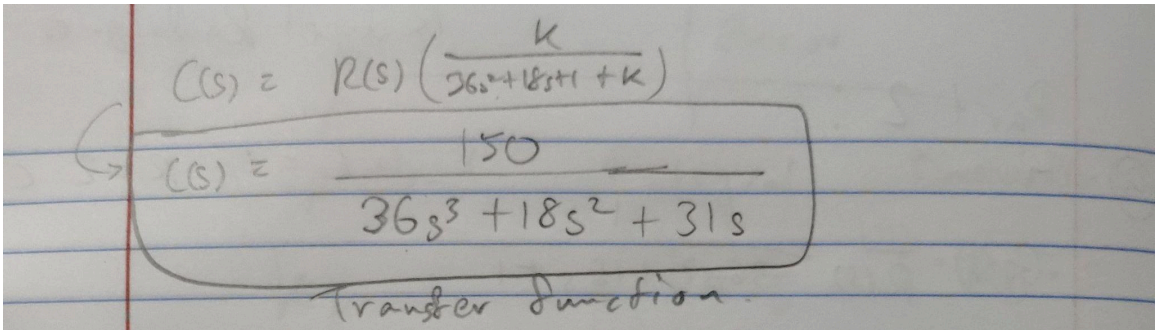


The rest of PART 1 are presented in MATLAB

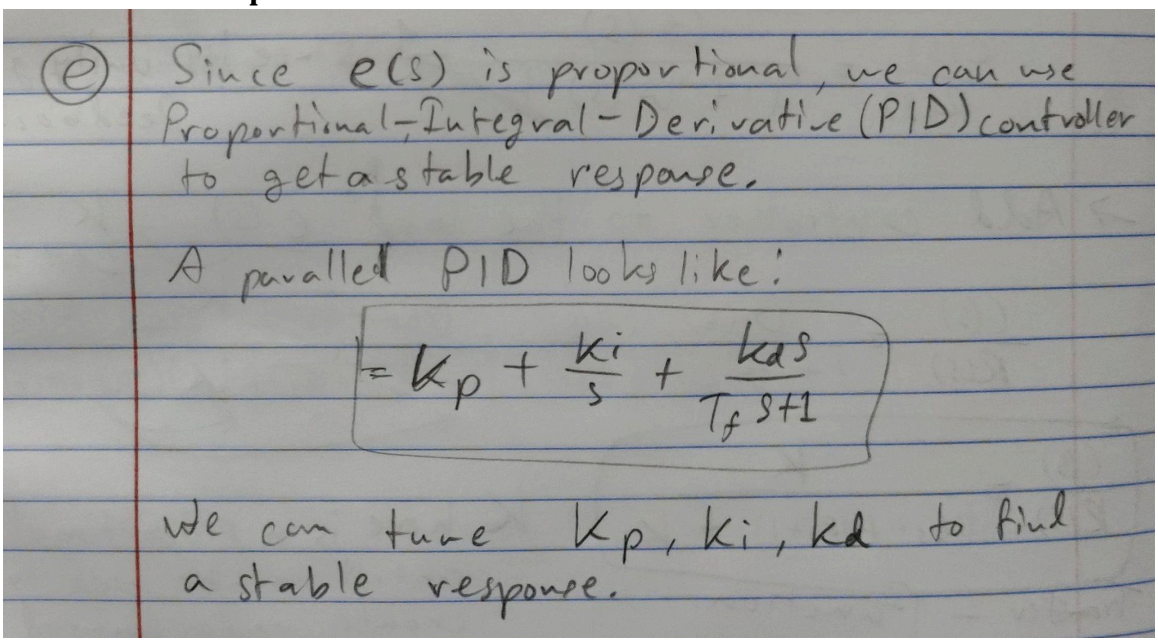
PART 2.

A. and B. Controller Tuning and Controller Staging





E. PID Control Implementation



H. Response Characteristics

Since our output response after controller and PID controller added goes to a 3rd degree polynomial, finding the rise time, peak time, settling time and % OS of our underdamped response is impossible to do by hand. Thus it is recommended that we manually calculate these specifications based on the step response data obtained from the step function and our specific criteria.

<u>Through our observation the following is extracted:</u>

Calculations.

H. Rise time from 10% to 90% Criteria

Observed

$10\% \times 5 = 0.5$ $t \approx 0.00827$ seconds

$90\% \times 5 = 4.5$ $t \approx 0.169$ seconds.

$0.1 c_{final} = 0.00827$

$0.9 c_{final} = 0.169$

Rise Time.

$T_r = 0.16073$ seconds

Peak time

Amplitude peak = 5.04

T_p = time to reach 5.04

$T_p = 0.445$ seconds.

Peak time

Setting time Criteria.

Time for c(t) to reach and stay within ± 2% of find value.

$\pm 2\% \times 5 = \pm 0.1$

Between = [4.9 (-2%), 5.1 (+2%)]

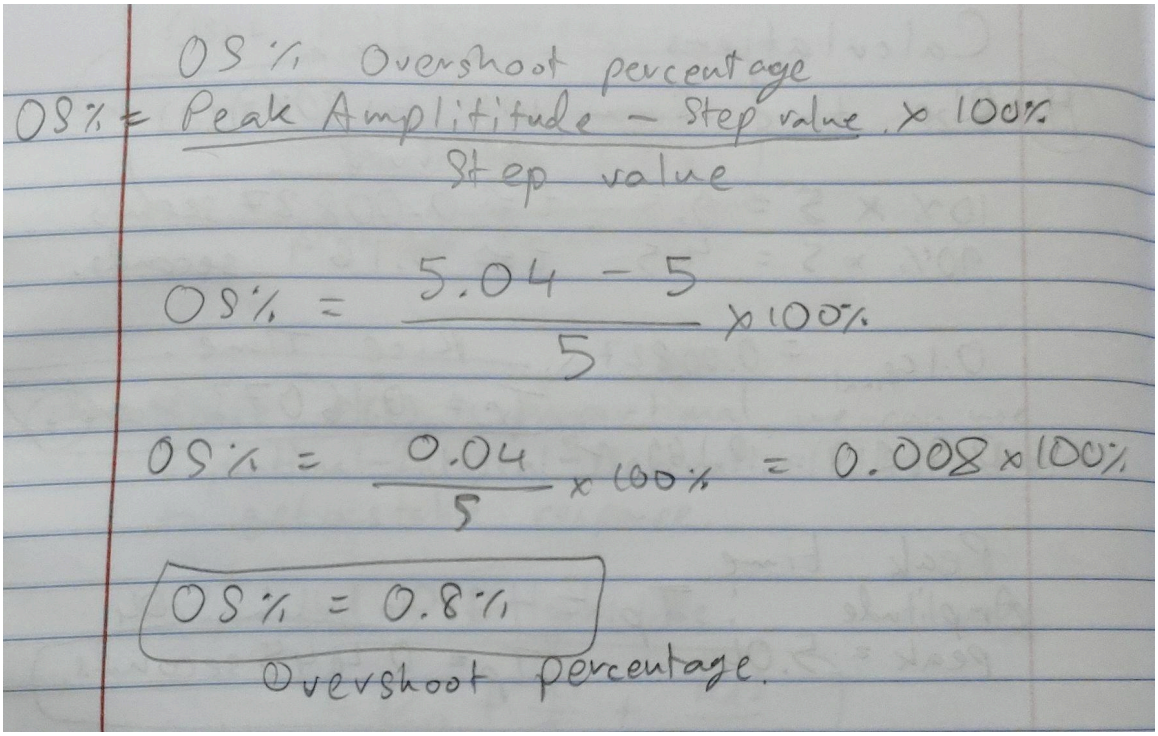
↑

Time to reach this value is: 0.255

and since it stays within ±2%

$T_s = 0.255$

Setting time



Testing Results and Analysis

Use this space to show the MatLab code and explain it in detail. Remember that you can also use comment lines to show the detail of your work.

PART 1.

Since finding a partial fraction is not achievable manually due to the denominator of the transfer function that can't be simplified. Matlab allows us to achieve our goal of creating a partial fraction and the inverse Laplace Transform. Using the code below we can produce the functions and the plots for each section.

```
%% Part 1
clear; clc;close all;
syms t s
L = 4;
R = 2;
C = 9;

Transfer_func = (1/(L*C)) / (s^2 + ((R/L)*s) + (1/(L*C)));
Step_input = 5/s;
Output_response = Transfer_func * Step_input
PartialFraction = partfrac(Output_response)
TimeDomain_response = ilaplace(PartialFraction)
figure;
fplot(TimeDomain_response, [0 100]);
title('Output Response');
xlabel('Time (s)');
ylabel('Output');
```

E. Inverse Laplace Transform

The output partial fraction and the inverse laplace of it is shown below. The total output response is next to TimeDomain_reponse.

```
Output_response =

5/(36*s*(s^2 + s/2 + 1/36))


PartialFraction =

5/s - (180*s + 90)/(36*s^2 + 18*s + 1)


TimeDomain_response =

5 - 5*exp(-t/4)*(cosh((5^(1/2)*t)/12) + (3*5^(1/2)*sinh((5^(1/2)*t)/12))/5)
```

F. Final Value and Steady State Error

Given the plot shown in G. Fig 2, The Final value reaches **5 volt** which is exactly our input and thus it has no steady state error.

G. MatLab Plotting

Fig 1 Demonstrate that it is overdamped

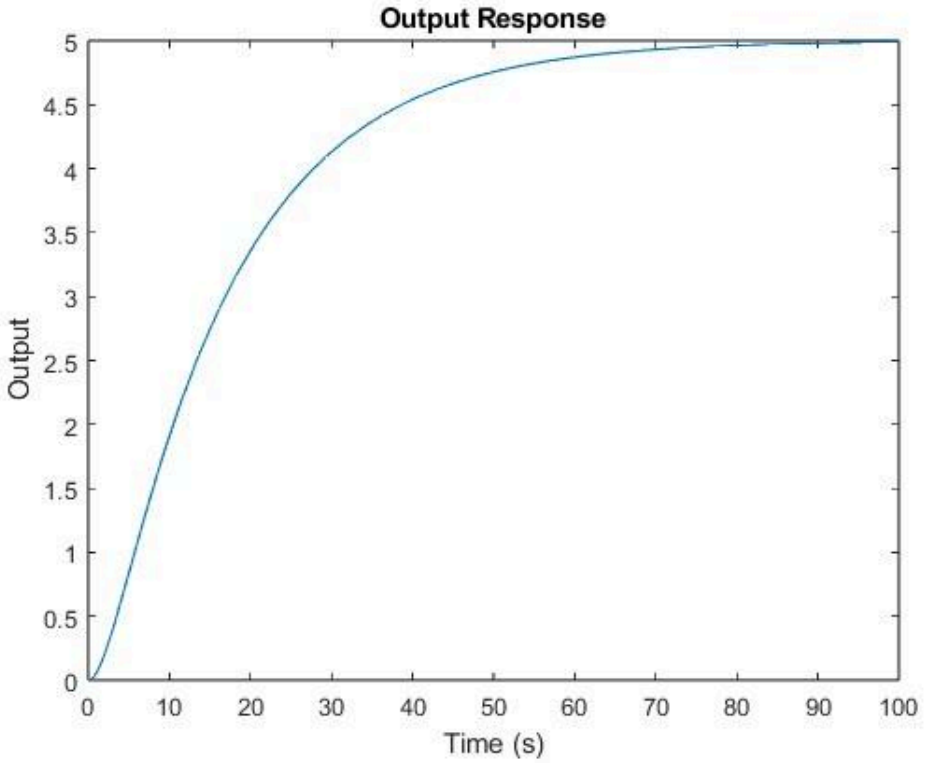
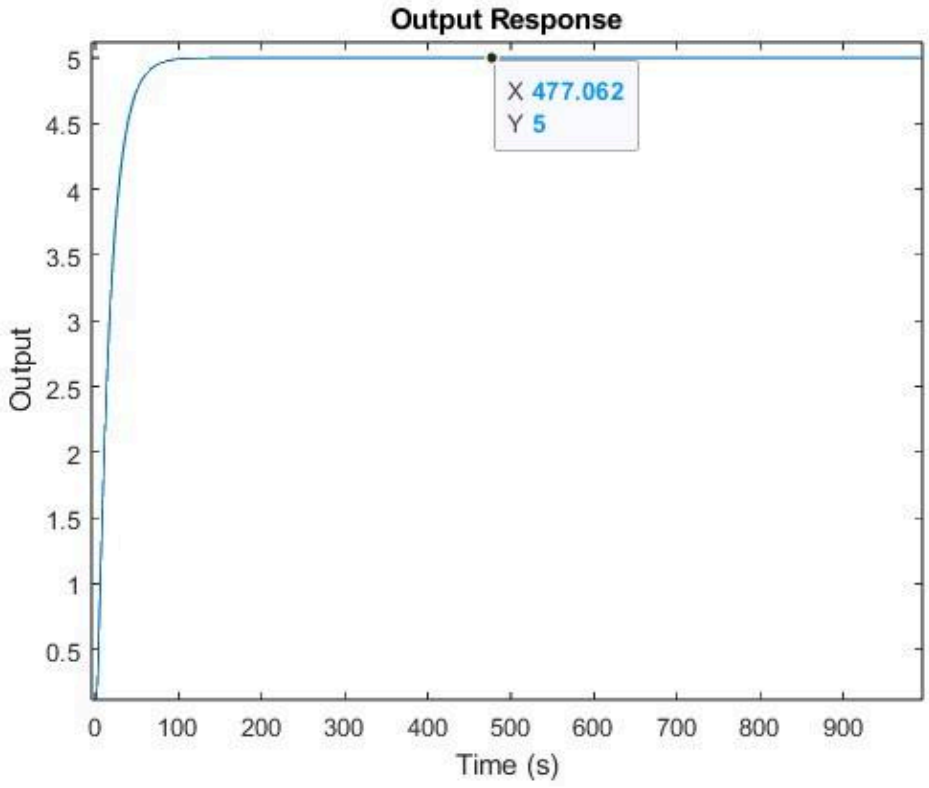


Fig 2 Demonstrate that it reaches 5 volt



PART 2.

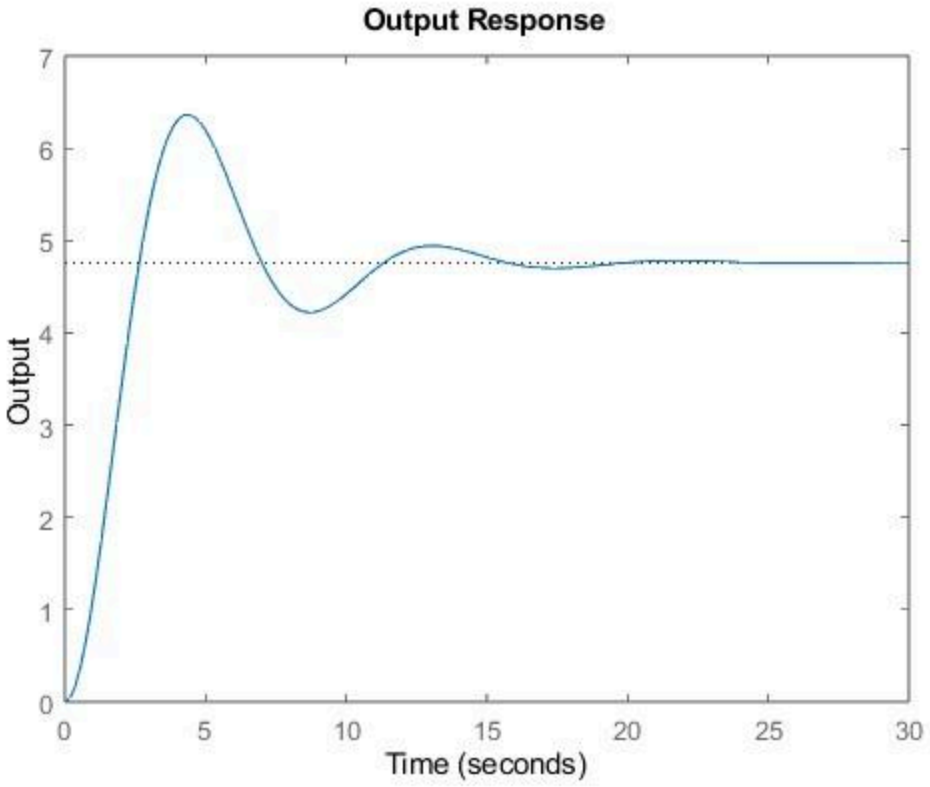
For the following dataset we used the code shown below and variations of it.
The code below shows the final version of our Transfer function that is a closed loop with unity feedback, with input a undamping controller e(s) into a PID stability controller L(s) and finally to the G(s) plant transfer function giving us the output response.

The values used for the tuning of the PID controller is Kp=20, Ki=2, and Kd=23, while the proportional controller uses "k" =20 to underdampen the plant before the PID takes over.

```
%% PID controller

Kp = 20;
Ki = 2;
Kd = 23;
k= 20;
numerator = k;
denominator = [36,18,1+k];
% numerator = 1;
% denominator = [36,18,1];
Transfer_func = tf(numerator,denominator);
PID_controller = pid(Kp, Ki, Kd);
sys_with_pid = series(PID_controller, Transfer_func);
closedLoop = feedback(sys_with_pid, 1)

opt = stepDataOptions;
opt.StepAmplitude = 5;
figure;
step(closedLoop, opt);
title('Output Response');
xlabel('Time');
ylabel('Output');

poles_closedLoop = pole(closedLoop)
zeros_closedLoop = zero(closedLoop)
figure;
pzmap(closedLoop);
```

A. (Demo output response)

Tuning the system's controllers until we get an underdamped response



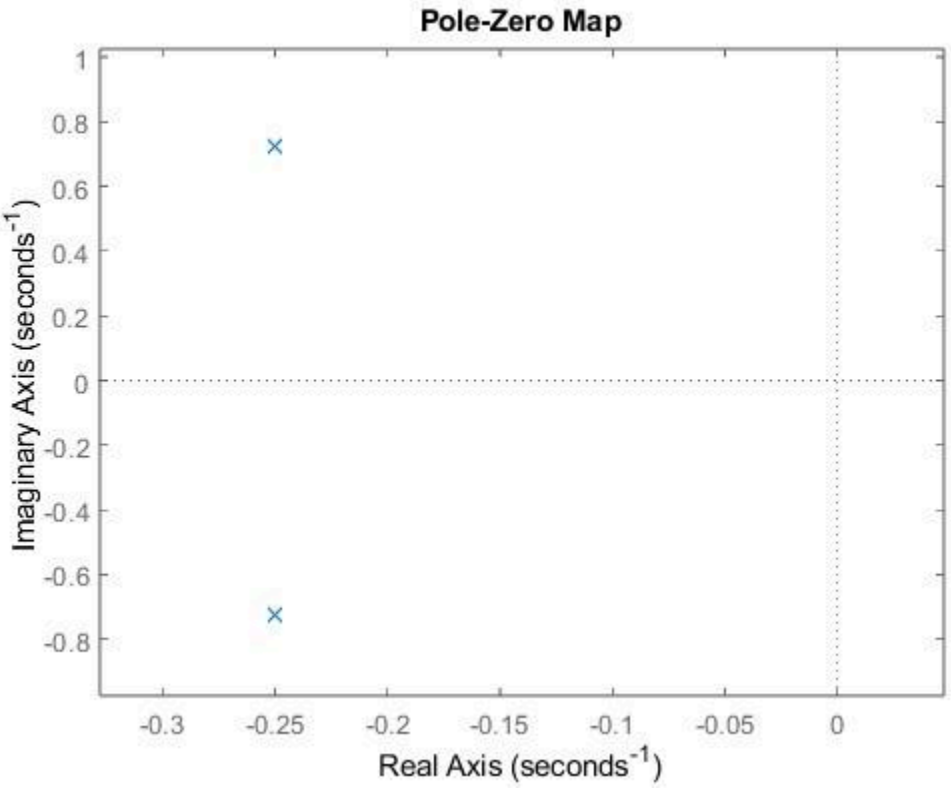
C. Closed Loop Transfer Function

The new closed loop transfer function, using the unity feedback controller

```
Transfer_func =

          20
  ------------------
  36 s^2 + 18 s + 21

Continuous-time transfer function.
```

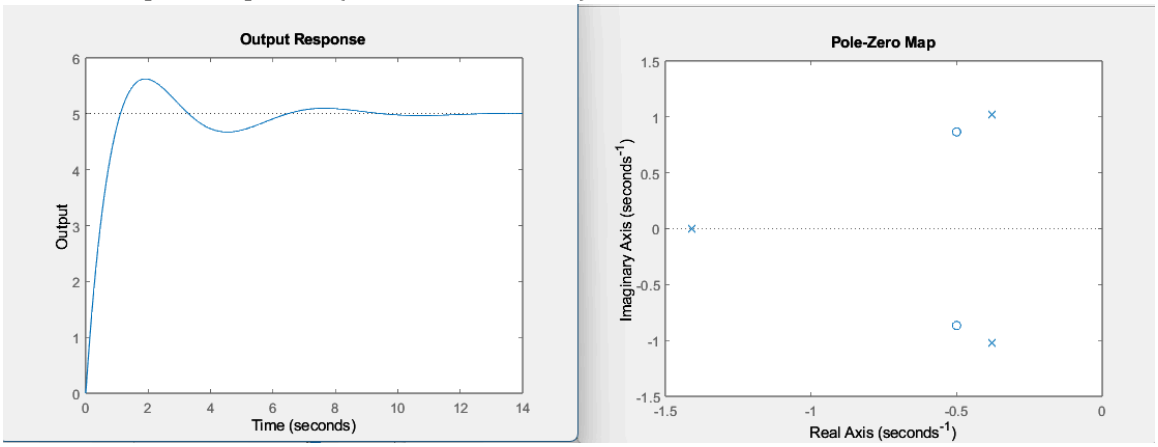
D. Poles and Zeros

The underdamped response's poles and zeros:

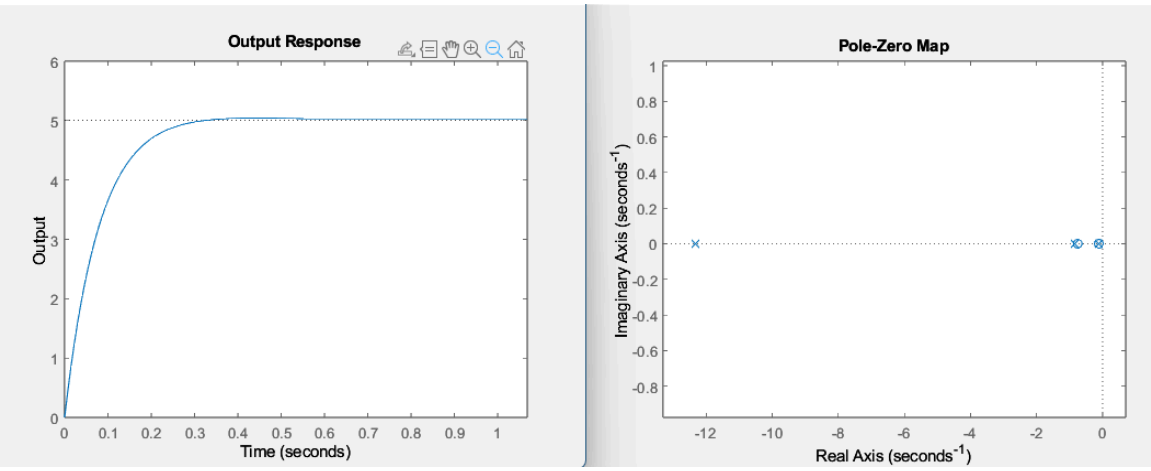


F. Contrast of PID Control

Underdamped response (BEFORE TUNING)



Stabilized response(AFTER PID TUNING)



G. Total Output Response

The total output response of our final stable underdamped response after PID tuning in S domain:

```
closedLoop =

       460 s^2 + 400 s + 40
  ------------------------------
  36 s^3 + 478 s^2 + 421 s + 40

Continuous-time transfer function.
```

H. (Check MATLAB values)

Derived from the step response shown above in Output response After Tuning.

```
rise_time =

    0.1607


peak_time =

    0.4442


settling_time =

    0.2566


percent_overshoot =

    0.8759
```

I. Damping Ratio

The damping ratios of all the poles are shown below. Since our output response has a transfer function that is now a third degree polynomial due to the PID controller, we have 3 damping ratios. They cannot be calculated traditionally as expected from a second degree polynomial.

```
The damping ratio of the transfer function is:
    0.1082
    0.8324
   12.3372
```

J. Natural Frequency

Additionally we have 3 natural frequencies but since they are identical we can assume it's the same across the function using one natural frequency.

```
The natural frequency of the transfer function is:
     1
     1
     1
```

K. MatLab Plotting

All the necessary plots are provided within their sections for PART 2.

Conclusions

To summarize, the project was an excellent gateway to understanding control systems behaviors and how to observe, analyze and develop a solution that makes the system more accurate and responsive. The project was largely successful since all of our queries were answered and we managed to produce a control system that is extremely fast and responsive and reaches the full scope of the input with a minor fractional overshoot. From the initial system taking nearly 80 seconds to reach the step value, we managed to reduce it down to a fraction of a second at 0.26 seconds. Along with the many other advancements to rise time, peak time and more have made this project a great success.

Through this laboratory I discovered how to install and use MATLABs Control System Toolbox to design and analyze control systems. It provides the algorithms and apps needed for systematically analyzing, designing, and tuning linear control systems, with ease.

The primary obstacle for this project was the scale of the complexity when adding the PID controller. In fact given the options I would omit the underdamping controller and use the PID for full control of the system. The higher the order of a system the more complicated the parts become and it becomes more prone to instability. In the future I would like to obtain a 3D printer and a collection of electronic components that would allow me to create systems that can assist our lives. Including but not limited to, smart home appliances like automatic curtains, portable wifi detector, fluid tank refill device etc.