Question 2 - Control Systems

Nitin G EE22B041

March 2023

1 Answers

1.1 Part A

Explain the working of Open and Closed Loop Control systems with another example for each.

What is a control system? Simply put, a control system is something that regulates, controls and directs the behaviour of other devices or systems. This is done with the help of control loops.

Control loops are of two types, open and closed. Open loop control systems are those which does not adjust its output with respect to the output of the system. Whereas, in closed loop control system, the output of the controller, i.e the control signal is continuously adjusted by taking the output of the system as feedback.

To understand this better, we shall give a couple examples. A clothes drier is an example of open loop control system. Here the initial state is evaluated and the time required to dry the clothes is evaluated using some algorithm. Then the clothe drier is switched on for this set time regardless of whether the clothes are dried or not. After this time, it automatically switches off even if the clothes are not completely dried. On the other hand, if it is a very sunny day, then even if the clothes are completely dried, the drier is still switched on till this particular time interval is over. That is there is no real time feedback into the system.

If we add a sensor to measure how dry the clothes are and feed that as feedback to the control system. Then we have a closed loop control system. Here the drier will be switched on till the clothes are dry no matter the external conditions. Another example for this is Air conditioners. Here there is a thermostat that measures the outside temperature and adjusts the compressor accordingly. It takes the outside temperature as an input, calculated the error between the required temperature and the current temperature. Then it gives a control signal corresponding to it that works to reduce the temperature towards the required value. This way, there is a live feedback system and the system will continue as long as the output reaches the required setpoint.

The closed loop control system sounds much more useful, but there are some advantages to the open loop systems as well. Open loop control systems are easy to build, easy to maintain and costs lesser. But the closed loop systems are costlier and more complex so they are tough to build and maintain. Open loop systems are generally more stable. They also generally have a faster response time. The disadvantages of open loop control systems is that it is less reliable, it is not robust enough to handle any system disturbance and its calibration for it to work well is difficult to do as it needs precision. Closed loop systems are much better in these regards.

1.2 Part B

You are working in a huge automotive company and tasked with designing a control system to control the motor speed. What are the parameters that will determine whether the designed system is industrially good?

The control system should be:

- Accurate: The system should be able to accurately control the motor speed and maintain it within a specified range.
- Responsive: The system should be able to respond quickly to changes in the input signals and adjust the motor speed accordingly.
- Stable: The system should be stable and not oscillate or overshoot when the motor speed is changed.
- Robust and Reliable: The system should be reliable and able to operate continuously for long periods of time without failure or errors in adverse environments.
- Scalable

The control

1.3 Part C

Read more about PID controllers and explain in brief how each of these parts is significant.

Given below is the block diagram of the PID controller. We can see that the feedback, y(t), is fed back to the system where the error is calculated and the control signal, u(t), is a function of this error. FIgur 1

The basic formula for the operation of the PID control system is:

$$e(t) = r(t) - y(t) \tag{1}$$

Here, e(t) is the error, r(t) is the reference / set-point, and y(t) is the output.

We then get the control signal from the equation:

$$u(t) = k_p e(t) + k_i \int e(t)dt + k_d \frac{de(t)}{dt} \tag{2}$$

I have just thrown a random formula, but why this formula? Why are all these terms required to output a 'nice' control signal. What do each of these terms correspond to?

To start, let us define what a 'nice' control signal is:

The proportional term is the one that drives the control signal and it is very natural to have a proportional term.

The differential term helps to identify the impending motion and helps to eliminate overshooting and oscillation.

The integral term takes care of the slight steady state error, it will make sure by adding this small errors to a large value, that the control system is forced to become steady only when error is 0.

1.4 Part D

We find the values of k_1 and k_2 by curve fitting through the given data. The following ipynb file outlines my method and results.

Part D

We try to find k_1 and k_2 in the following equation:

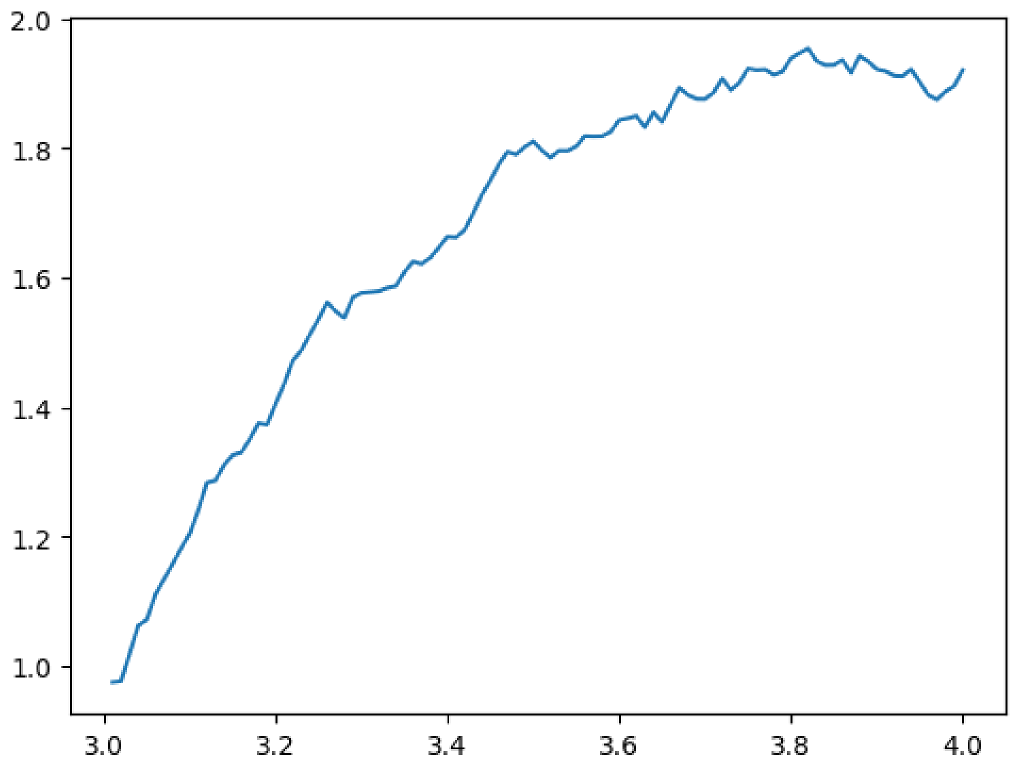
$$\frac{dv}{dt} = k_1 \cdot \text{throttle} - k_2 \cdot v + \text{noise}$$

We find this by trying to fit an equation to the given dataset

In [ ]:
```
#processing the data
datafile = open("dataset_bolt_model.txt", 'r')
data = datafile.readlines()[1:]
processed_data = [list(map(float, line.split())) for line in data]
t_values = [i[0] for i in processed_data] #x-axis
v_values = [i[1] for i in processed_data] #y-axis
#print(x_values, '\n', y_values)
```

In [ ]:
```
import scipy
import numpy as np
import matplotlib.pyplot as plt
```

In [ ]:
```
plt.plot(t_values, v_values);
```



In [ ]:
```
def fitfunc(t, k1, k2):
    throttle = 40
    v0 = 1

    def myode(v,t):
        dvdt = k1*throttle - k2*v
        return dvdt
```

```
    v_sol = scipy.integrate.odeint(myode, v0, t)
    return v_sol[:,0]

k_fit, kcov = scipy.optimize.curve_fit(fitfunc, t_values, v_values, p0 =

print(k_fit)
```

```
[0.1455843  2.89762074]
```

We now check if the k values given by the code looks okay. (Sanity check)

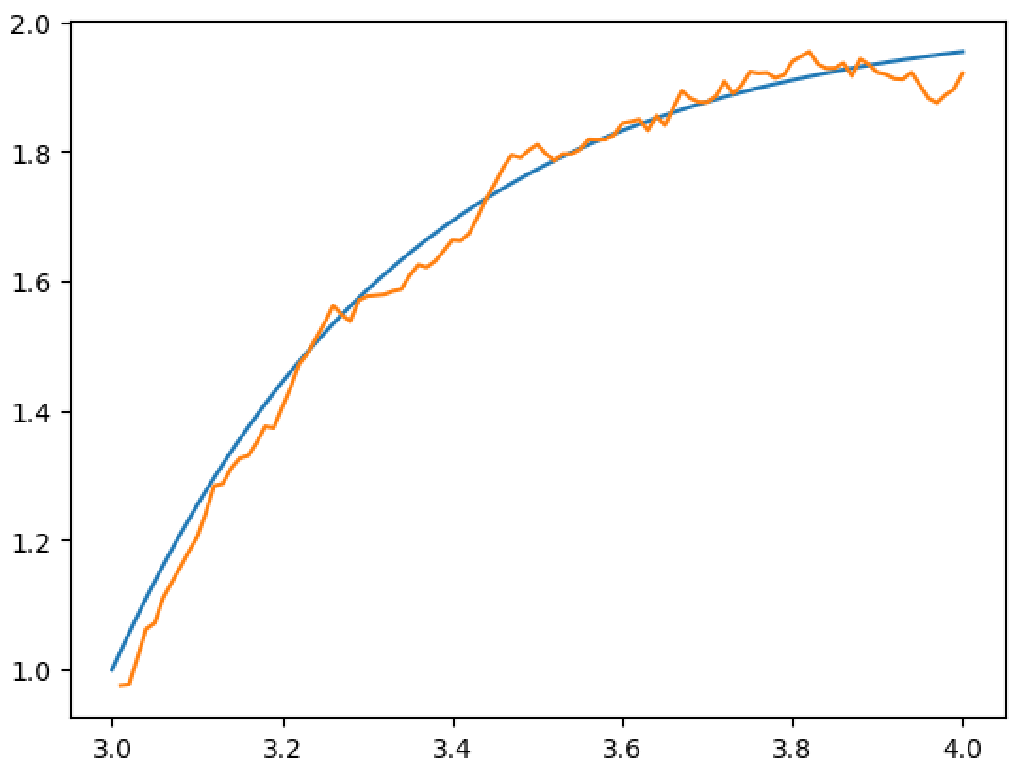
In [ ]:
```
def myode(v,t):
    dvdt = k1*throttle - k2*v
    return dvdt

k1, k2 = k_fit
throttle = 40
v0 =1

x_vals = np.linspace(3, 4, 100)
y_vals = scipy.integrate.odeint(myode, v0, x_vals)

plt.plot(x_vals, y_vals);
plt.plot(t_values, v_values);
print(k1, k2)
```

```
0.145584295473428 2.8976207362313526
```



We have achieved what we wanted to do, i.e found out the values of k1 and k2.

k1 = 0.145584295473428

k2 = 2.8976207362313526

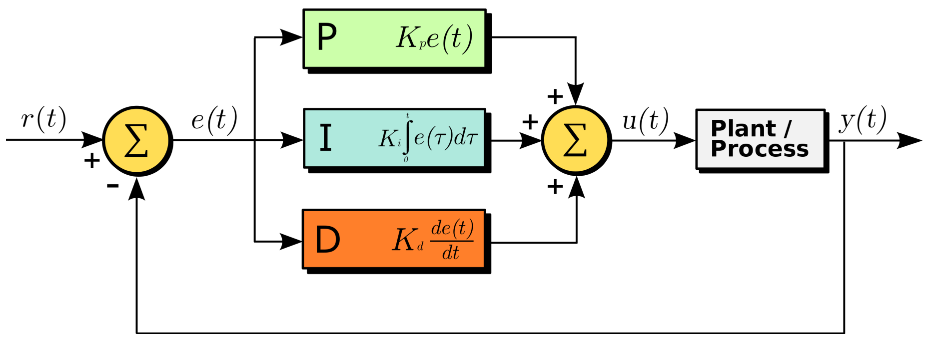


Figure 1: Block Diagram of PID Controller

1.5 Part E

The tuning and explanation is given in a video to see the all the errors clearly.
The drive link to the same is:
https://drive.google.com/drive/folders/176uDs9J1_caNn4HeXiSLsqsheidtjBAW?usp=sharing

2 Sources

2.1 Part A

1. https://circuitglobe.com/difference-between-open-loop-and-closed-loop-system.html
2. https://www.geeksforgeeks.org/difference-between-open-loop-control-system-and-closed-l
3. https://www.elprocus.com/difference-between-open-loop-closed-loop-control-system/
4. https://en.wikipedia.org/wiki/Control_loop
5. https://en.wikipedia.org/wiki/Control_system

2.2 Part C

1. https://en.wikipedia.org/wiki/PID_controller

2.3 Part D

1. https://kitchingroup.cheme.cmu.edu/blog/2013/02/18/Fitting-a-numerical-ODE-solution-to

3 NOTE

To access all the source files, screenshots and ipynb files, etc., please visit this github repo:

`https://github.com/MachanHoid/AbhiyaanElecApp.git`