# CONTROL SYSTEM VIRTUAL LAB – COMPENSATORS

## 18EE810 Project

*Submitted in partial fulfillment for the requirement of B.E. degree in Electrical and Electronics l Engineering of Anna University*

*Submitted by*

***Abinava Kishore.M***
***(18E002)***

***and***

***Steve Antonio D***
***(18E108)***

*Guided by*

***Dr. S. Baskar***
Professor and Dean(R&D)

***Department of Electrical and Electronics Engineering***

**THIAGARAJAR COLLEGE OF ENGINEERING**
(An Autonomous Institution Affiliated to Anna University)

**MADURAI – 625 015**

***April 2022***

# THIAGARAJAR COLLEGE OF ENGINEERING

**(An Autonomous Institution Affiliated to Anna University)**

## MADURAI – 625 015.

## CERTIFICATE

Certified that this is a bonafide record of the 14EE880 Project & Viva Voce done by **Mr. Abinava Kishore.M** (18E002) **and Mr. Steve Antonio D** (18E108) of Eighth Semester B.E. Electrical and Electronics Engineering during the year 2018 - 2019.

Signature of the Guide Signature of H.D.E.E

Station: Madurai Date:

---

Submitted for Viva-Voce examination held at Thiagarajar College of Engineering, Madurai – 625 015, on _____.

INTERNAL EXAMINER EXTERNAL EXAMINER H.D.E.E.

# CERTIFICATE

This is to certify that the **14EE880** Project Work Report entitled ***"Control System Virtual Lab – Performance of Compensators",*** being submitted by Mr. **Abinava Kishore.M** and Mr. **Steve Antonio D** in partial fulfillment for the requirement of **Bachelor of Engineering Degree in Electrical and Electronics Engineering**, is a record of bonafide work. The results embodied in this report have not been submitted to any other university or institute for the award of any degree or diploma.

**Mr. Abinava Kishore.M**
**(B.E. STUDENT)**

**Dr. S. Baskar**
**(GUIDE)**

**Mr. Steve Antonio D**
**(B.E. STUDENT)**

Station: Madurai

Date:

# LIST OF CONTENTS

# LIST OF SYMBOLS

| Symbol | Description |
|---|---|
| A | Cross sectional area of the Water Tank |
| H | Calculated height of water in the Water Tank |
| $H_{ref}$ | Desired Height of the Water Tank |
| $H_m$ | Actual measured height of water in the tank |
| $A_{pipe}$ | Cross sectional area of the outlet pipe |
| g | Acceleration due to gravity (= 981 cm $s^{-2}$) |
| K(tank) | Constant (= $\sqrt{2g}$) |
| K(compensator) | Gain of the Compensator |
| Tc | Time Constant of the Compensator |
| α | Attenuation factor (alpha) of the Compensator |
| U | Output of the compensator (litres/hr) |
| Ur | Randomness added to flow of water |
| Uf | Feed forward term added to make system stable |
| Ud | Custom Disturbance added to the normal flow of water |
| Ts | Sampling time period |
| E | Error ( = $H_{ref} – H_m$) |
| π | Pi (= 22/7) |

# LIST OF FIGURES

# LIST OF TABLES

# ABSTRACT

The need and usage of control and control devices is increasing day by day by in this modern age world. In practice, almost every aspect of human life has been affected by at least one control system in day-to-day life. The main purpose of Control System for any given plant is to improve the gain or bandwidth of the system, and also the stability of the system. In order to achieve these objectives in a feedback control system, there are different types of compensators, which can be used. Two most common types of compensators are the lead and lag compensators. There are also other types of compensators such as lag-lead compensators, feed-forward compensators etc.

The objective of this project is to portray the concepts related to compensators, using a virtually created lab. Virtual Labs can be used as an alternative and innovative tool in providing laboratory experiences. We use simulation and visualization to create realistic scenarios as much as possible. The system which we have chosen is a simple Water Tank model, in which we will be controlling the height of the water. Many years back, level control has been one of the major issues in the industries. Controlling the level of liquid/fluid has been essential in almost every industrial process like food processing, nuclear/thermal power plants, water purification systems, chemical processing, boilers etc.

The simulation of the control of water level in the water tank system is demonstrated using a free, open-source and cross-platform game engine called Godot, which is released under the MIT License. We create a game like environment using the game engine, with a rich and easy to use user-interface. The main part of the project is to simulate the control of water level, using compensators. The user must enter the required parameters for the compensator, and input the desired height in which the level of water should be maintained. The process is controlled by using coding the system and control equations in Godot using its native scripting language called GDScript.

The output of the simulation can be visualized in real-time in terms of height, time taken and error. The response can also be exported as a json formatted file, which can later be used for plotting and analysing. Options for converting the generated json file to the required plot is also given along with the application.

Thus, the virtual lab gives the user a virtual, yet realistic experience of understanding about the concepts of compensator and how it can be used to control various systems around us.

# CHAPTER 1

# INTRODUCTION

## 1.1. CONTROL SYSTEMS

Control Systems can be defined as a system or a group of systems which work together, to regulate the behavior and performance of real-world systems which we use in our day-to-day lives. In the recent years, control systems have played a major role in the development and modernization of our daily activities.

### 1.1.1. Why Control Systems?

As stated above, control systems have become as an important aspect in our daily lives. What makes control systems so important? Here are some of the main features and need of these life-changing systems:

- It gives quick, faster and accurate response, due to which, the overall productivity in industries has increased.
- It minimizes error in the system, which has made it more reliable for the systems which use it.
- There are always some unwanted noise or disturbance present in our systems. These unwanted noise and disturbances can be easily filtered with the help of control systems.
- It reduces the overall cost per product, due to the above advantages.

### 1.1.2. Compensators

There is always a need to adjust the behavior of a system to provide suitable performance. In order to make fine tune these adjustments, we need to receive the current performance of the system, and it is done by getting it through the feedback system. And the feedback, when given to a Compensator helps in improving the

performance of the system. The compensators can be classified into various categories. The most basic and common classification of compensators are:

1. Lead Compensator
2. Lag Compensator

## 1.2. VIRTUAL LABS

In this rapidly developing world of technology, everything has now become accessible through online. Moreover, it is more convenient than the traditional way. Education is not an exception! Technological advancements in education have been a revolution over the years. Gone are the days, when the student has to go in person to labs. Virtual Labs have become the current trend, where students can access them from anytime, anywhere and without any limitations or restrictions.

### 1.2.1. Advantages of Virtual Labs

Virtual Labs have started replacing the traditional old-fashioned labs. And there are plenty of good reasons to support for the same. Some of the advantages of Virtual Lab are:

- The COVID-19 pandemic has led to a dramatic change in the lifestyle of the people, especially students. As the educational institutes were locked down, physical labs could not be conducted. This gave way to the rise of Virtual Labs.
- When physical lab sessions have to be conducted for each and every subject, we are limited by various restriction. These restrictions can be summed up into space, time and economical limitations.
- In physical labs, the student will not get the freedom to modify or adjust the parameters of a system beyond limits. For example, if a student is working in some kind of chip, which has a peak current limitation in its specification, he must not cross the limit, as the chip will get damaged and not usable in the future. Same is not with the case of Virtual Labs, in which the chip is virtual, and can be tried with values greater than what the chip can handle.

- Another good justification for choosing virtual labs is that, we don't need to buy costly equipment needed for the traditional labs. Instead, virtual labs create a simulated environment, where you can get the exact and realistic feel of using the equipment.
- Also, the domain which we have chosen for this project, i.e., control systems, very rarely have virtual labs available over the internet. So, it will be a good addition to the Virtual Labs family.

## 1.3. GODOT

Although there are lots of common options available to create Virtual Labs, we have made an innovative idea to create the game-like environment, which allows for more intuitive and more interesting way to create the Virtual Labs. Godot is a free and open-source gaming engine that runs on a variety of platforms and is licensed under the MIT license.

### 1.3.1. About Godot

Godot promises to provide a complete game development environment. It enables game creators to create a game with no additional tools beyond those needed for content generation. The architecture of the engine is based on the concept of "nodes". Scenes, which are reusable, instanceable, inheritable, and nestable collections of nodes, organize nodes. The latest version of Godot is Godot v4.0 which is still a beta version. The version which we used for this project is Godot v3.3.3.

### 1.3.2. Why Godot?

There are numerous reasons why you can trust Godot. We have listed few of those reasons:

- Godot is beneficial to programmers: Godot is one of the most useful tools for programmers who wish to make games. In terms of API, the Godot game engine is quite dependable because the Godot API exposes most of the

engine's features and displays aspects that are difficult to access directly through code.

- Godot supports multiple languages: C++, C#, and Visual Script are just a few of the programming languages that the Godot game engine supports directly. The Godot engine comes with its dedicated programming language called GDScript, which gives developers an edge when using this platform. This language is almost similar to python.
- Node System: In Godot game engine, each game node is a single object that can inherit from any other node in the game. A scene is a collection of such nodes. Scenes can also pass on their inheritance to one another. Customers can use Godot's node structure to work with objects that aren't readily understood as in game engines like Unreal.
- Godot provides a Custom IDE feature: Godot Engine includes an integrated development environment (IDE) with a number of useful features for users. The best thing about Godot IDE is that it allows users to make their own decisions. If the developer dislikes Godot's IDE or text editor, he or she can continue to work on Godot while using their preferred tools like VS Code.
- Godot Engine is a free and open-source engine: Godot is MIT-licensed open-source software that gives developers access to all relevant tools. Users can customize Godot's gaming engine by adding new and unique features. As a result, these tools created by the community can assist other programmers in the development of games.

## 1.4. SUSTAINABLE DEVELOPMENT GOALS

Out of the 17 Sustainable Development Goals (SDGs), this project satisfies the following 2 goals:

1) Quality Education (Goal No. 4)
2) Decent Work and Economic Growth (Goal No. 8)

### 1.4.1. Quality Education

The main objective of this project is to provide access to labs in a virtual manner, so that, students as well as teachers can use it from wherever they want and whenever they want. Also, as explained earlier, it allows more in-depth grasping of concepts, and a more intuitive way of learning the concepts.


**Figure 1.1 Quality Education**

### 1.4.2. Decent Work and Economic Growth

One of the outcomes of the project is that it will provide lots of knowledge and understanding to the students, so that, they can improve their skills in the respective domain (here, control systems). Also, when this project gets deployed out in the society, the need of physical labs, cost of maintenance of equipments in those labs, and the labor works can be completely stopped. This allows for a free and entirely new way of learning which involves no cost at all.

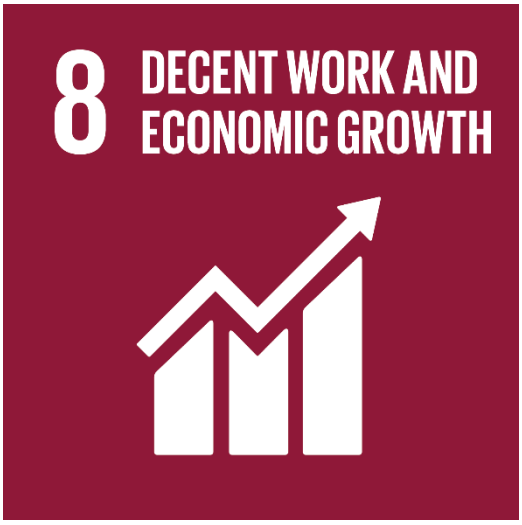



**Figure 1.2 Decent Work and Economic Growth**

# CHAPTER 2

# LITERATURE REVIEW

## 2.1. LITERATURE SURVEY 1

J. R. Brinson published a research article on the topic, “Learning outcome achievement in non-traditional (virtual and remote) versus traditional (hands-on) laboratories: A review of the empirical research” in which he surveyed various literatures of learning outcomes from traditional labs as well as online labs over ten years. He finally concluded that most studies (about 50.89%) showed equal or greater learning outcome in virtual and remote labs as compared to real labs. He considered the following categories for his analysis **[1]**:

- Knowledge and Understanding
- Practical skills
- Social & scientific communication
- Inquiry skills
- Perception
- Analytical skills

## 2.2. LITERATURE SURVEY 2

Mrityunjay Kumar et al. published a journal paper in the 2018 IEEE Conference on the topic, “Usability Analysis of Virtual Labs”. This study looks at virtual labs from a usability standpoint, focusing on vlabs, which is a national effort to offer science and engineering college students in India with access to internet and browser-based virtual laboratories. The usability of user-interface elements (technical usability) has been examined, as well as how they promote learning. The paper mentions that continued investment in virtual labs makes sense, given their cost-effectiveness and ubiquitous access, as well as the fact that they produce equivalent or superior learning outcomes. **[2]**

## 2.3. LITERATURE SURVEY 3

Veljko Potkonjak et al. conducted research and published a journal related to "Virtual laboratories for education in science, technology, and engineering: A review". In the domains of science, technology, and engineering, this document covers the current state of the art in virtual laboratories and virtual worlds. The paper argued about the pros and cons between the virtual laboratory and a physical laboratory. The goal of this work is to expand on this problem space between virtual and physical labs and provide a valuable resource of information that will aid in defining the beginning point for future research. **[3]**

## 2.4. LITERATURE SURVEY 4

M. Travassos Valdez et al. published this IEEE paper in the year 2014. They have mentioned about the features and benefits of using Virtual Laboratories when used in Electrical Engineering. The paper mentions that Virtual laboratories are the way of the future in education. To blend real-world surroundings with data produced by devices, simulations in which environmental measurements and data analysis can be simulated are being developed. According to the paper, Virtual Labs are tasks or challenges that are presented in a virtual environment or context for instructional purposes in Electrical Engineering. The environment and digital objects are meant to encourage inquiry-based learning and conceptual comprehension, so users can explore the environment and investigate digital things. It describes about the interactive simulations, virtual labs, testing as well as internet links in a virtual lab. **[4]**

## 2.5. SUMMARY OF LITERATURE SURVEY

| S.No | Title | Name of authors | Publication | Year |
|---|---|---|---|---|
| **1** | Learning outcome achievement in non-traditional (Virtual and remote) versus traditional (hands-on) laboratories: A review of the empirical research | J.R. Brinson | Computers & Education | 2015 |
| **2** | Usability Analysis of Virtual Labs | Mrityunjay Kumar, Jessica Emory, Venkatesh Choppella | 2018 IEEE 18th International Conference on Advanced Learning Technologies (ICALT) | 2018 |
| **3** | Virtual laboratories for education in science, technology, and engineering: a review | V. Potkonjak, M. Gardner, V. Callaghan, P. Mattila, C. Guetl, V. M. Petrovi, and K. Jovanovic | Computers & Education | 2016 |
| **4** | Virtual labs in electrical engineering education - The VEMA environment | M. Travassos Valdez, C. Machado Ferreira, Maria João M. Martins, F. P. Maciel Barbosa | 2014 Information Technology Based Higher Education and Training (ITHET) | 2014 |

**Table 2.1 Summary of Literature Review**

## 2.6. INFERENCE FROM LITERATURE SURVEY

From the above literature surveys, we can infer that virtual lab outperform the traditional labs in terms of cognitive learning outcomes. Some of the important points are listed below:

### 2.6.1. High Quality Equipment

Virtual labs provide easy access to cutting-edge technology for experimentation for students. Simulations and virtual microscopes, for example, offer science students futuristic solutions.

With a virtual lab in place, learners no longer have to rely on outdated equipment and can instead work with AI-powered laboratories and modern teaching techniques. Using advanced technological solutions, they can easily compete with peers.

Furthermore, with practical applications, these labs can assist the teacher in covering various aspects of the course curriculum. It allows students to better understand concepts, which would otherwise be difficult to provide
with limited equipment and funding.

### 2.6.2. Visual Aid for complex concepts

Teachers can easily explain complex theoretical concepts to students using virtual labs through a visual, immersive experience that can make it easier for students to understand.

Furthermore, it provides the necessary synchronization between explaining theoretical ideas and practical application.

### 2.6.3. Ensures safety of students

In a virtual laboratory, students can conduct a variety of experiments without the risk of injuring themselves or damaging equipment. They can also test multiple scenarios, compare them, and determine which is the most effective without having to try them out in real life.

### 2.6.4. Immediate Feedback

Another benefit of virtual labs is that students can redo the experiments. Experiments in virtual labs, unlike traditional labs, no longer have a single chance option. Students can go over their mistakes, analyze what went wrong, and try again right away.

Maintaining communication between teachers and students becomes more efficient now that all experiment results are recorded.

## 2.7. CONCLUSION FROM LITERATURE SURVEY

Virtual labs are a low-cost and effective way to simulate a real-world environment in the comfort of one's own computer. Researchers and educators are looking into new areas where Virtual Labs can be used as a substitute.

# CHAPTER 3

# PROBLEM DESCRIPTION

## 3.1. PROBLEM ENVIRONMENT

The digital age and the information technology age are terms used to describe the twenty-first century. Students must be proficient in the use of technology in today's virtual era in order to be successful global citizens, and educators must help them achieve this goal. Engineers are troubleshooters who must solve challenges and make things work in the workplace. As a result, the capacity to apply theoretical information in real-world situations is critical. For this cause, laboratories sessions are conducted starting from high school till workplaces. We face lots of issues with the traditional, old fashioned way of laboratory classes.

### 3.1.1. Problems with Traditional Lab

When it comes to Engineering education, students should be prepared for their future tasks. In so-called laboratories, where students conduct experiments and analyze the results, it usually combines theoretical knowledge with practical experience. There are lots of problems which may rise in these laboratories:

- Due to time and safety constraints, only a limited number of people can conduct experiments at the same time.
- As a result, laboratory equipment is in short supply, especially as the number of students enrolled grows.
- In physical labs, the student might not get the complete freedom to set/modify all the parameters of a system. The equipment/component which a student uses has limit restrictions like peak current, max voltage etc. As a result, if these limits were broken, it may cause harm to device or the student himself.

- As technology keeps increasing day by day, the cost of equipments is also rising tremendously. As a result, it becomes economically difficult to keep things updated.
- Tough situations make things tough. The COVID-19 pandemic has been a big lesion for all mankind. It is also important to face such situations, where everything has been put under lockdown, including schools, colleges and most importantly laboratories. During these tough times, virtual labs started gaining popularity and importance.

### 3.1.2. Other Problems

Some other problems which motivated us to take up this project are:

- The domain which we have chosen to make this virtual lab is Control Systems. After surfing the Internet, we understood that Control Systems Virtual Labs were not yet available. So, an initiative is needed.
- The quality and standard of education is not up to the mark. So, an enhancement is needed in the existing teaching plans.
- Although there are quite a few virtual labs existing already, they are not intuitive as well as interactive. Even virtual labs started to become boring for the students. So, an alternative and more interesting method has to found out.

## 3.2. OBJECTIVES

After listing and analyzing the problems mentioned above, we decided to develop a “Control System Virtual Lab”. As Control Systems is a very wide area in the field of Engineering, we will not be able to cover the full domain. Instead, we narrow down it to a single experiment, which will explain a particular control system concept.

The main objectives of this project are:

- To develop a Control System Virtual Lab, which will portray the concept of “Compensators” – more precisely “lead” and “lag” compensators.

- To give the Virtual Lab a game-like feel, to make it more attractive and exciting for the students. For this purpose, we have to use a game engine like Godot, Unity, Unreal etc.
- To control the level of water in a water-tank using compensators.
- To allow students to adjust the parameters of the compensator in real time and see how the response changes.
- To make students design a compensator on his/her own according to their expected response, and to enter those calculated parameters and verify their response.

## 3.3. ASSUMPTIONS

From the developer point of view, we need the following pre-requisites. They are:

- Basic knowledge about coding. Should know atleast any one programming language.
- Should have a good understanding about Control Systems, particularly Compensators.

From the student/user's point of view, following assumptions can be made.

- He/She has studied the concepts of compensators in his theory classes.
- He/She knows how to operate the PC and use the internet.
- He/She has knowledge on how to analyze the response graph.

# CHAPTER 4

# METHODOLOGY / EXPERIMENTAL SET-UP

## 4.1. EXISTING SOLUTION TO THE PROBLEM

There are lots and lots of virtual labs present in the Internet. We have analyzed some of the prominent solutions which are already available in the Internet.

### 4.1.1. IIT KGP Virtual Lab

A robotics and control systems virtual lab are available on IIT KGP's online virtual labs. This lab lacks compensator experiments, and the ones that are available are not very intuitive.

List of control systems experiments available in this virtual lab:

- Open loop characteristics of DC Motor.
- Closed loop characteristics of DC Motor with P, PI, PD and PID control.

### 4.1.2. IIT Kanpur Virtual Lab

IIT Kanpur, in its virtual lab has a few advanced experiments but they are not available for the use of other college students.

### 4.1.3. Dev-mind’s Virtual Lab

Dev-mind’s control system virtual lab has several experiments related to control systems, but it does not have compensator experiments which is going to be done by us.

Some experiments are as described below:

- spring mass
- Rotary spring Mass
- Temperature
- Water level

- Pendulum
- Double Mass
- DC Motor
- Double bar

All of these experiments have animations but are not as intuitive as it should be.

## 4.2. EXPERIMENTAL SET-UP

Following are the set-up which we have made, with the help of Godot Game Engine.

### 4.2.1. Title Screen

Title screen has 4 buttons: Start, How to use, Theory and Exit. Each leading to subsequent screens.

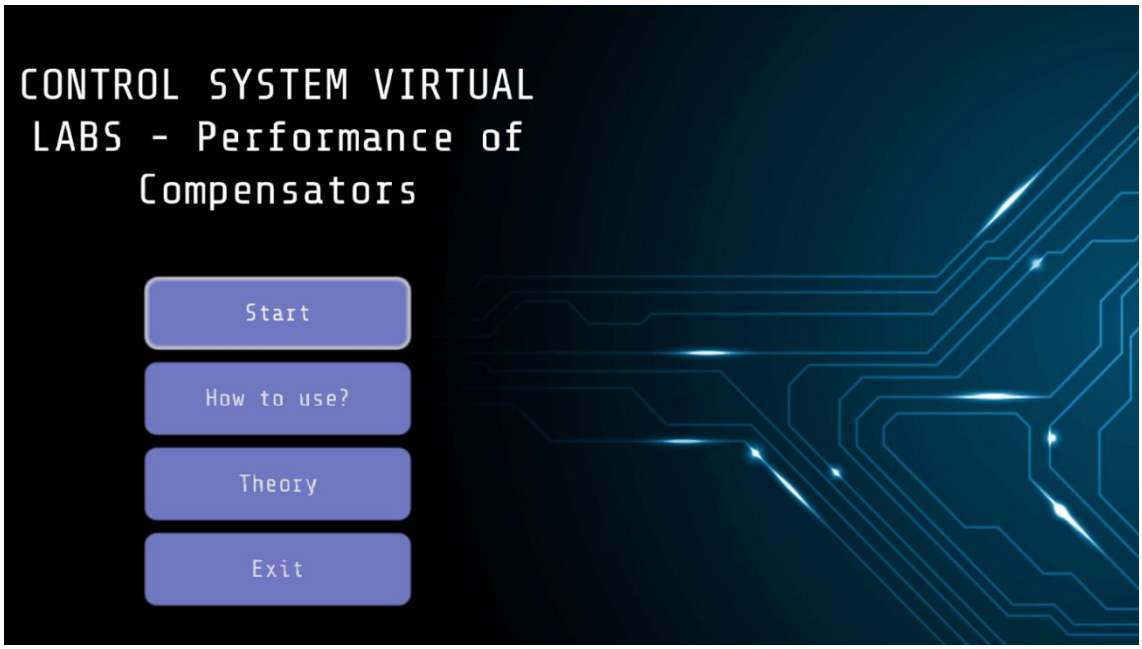



**Figure 4.1 Title Screen**

### 4.2.2. How to Use Screen

**“**How to Use” screen has the rules/instructions to use the virtual lab along with the procedure to plot the graph.

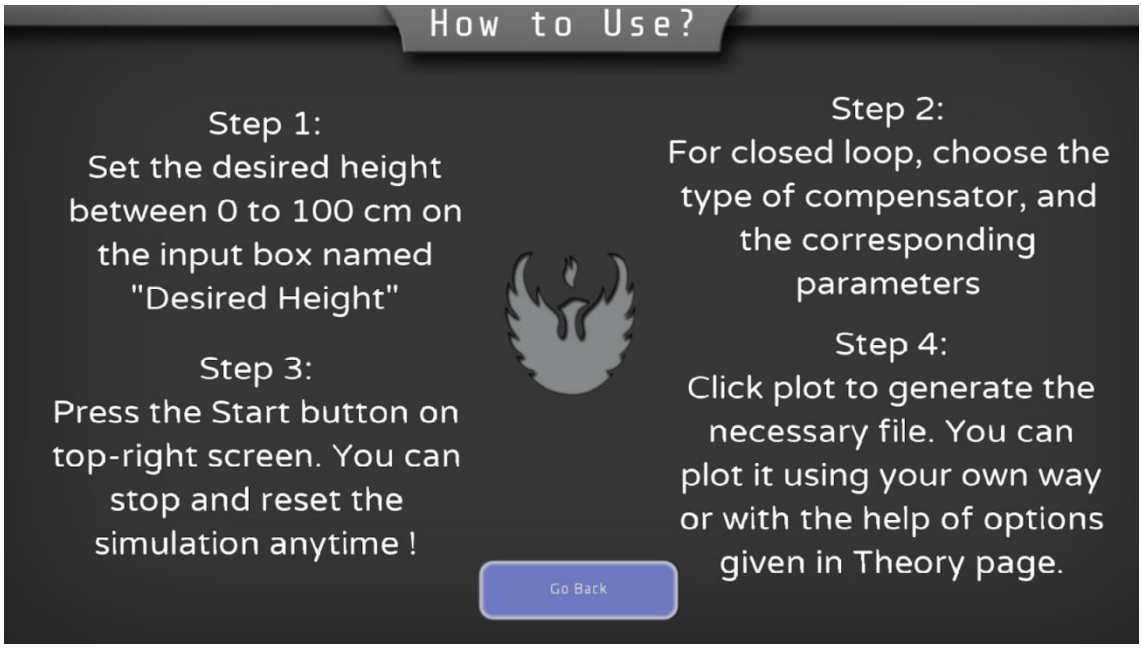



**Figure 4.2 "How to Use" Screen**

### 4.2.3. Theory Screen

The theory screen explains about the theory of the experiment. It depicts the used water tank model as well as the compensator design formula. Students can design the K, Tc, and α parameters using these formulas. It also contains resources at the end to assist with the experiment.

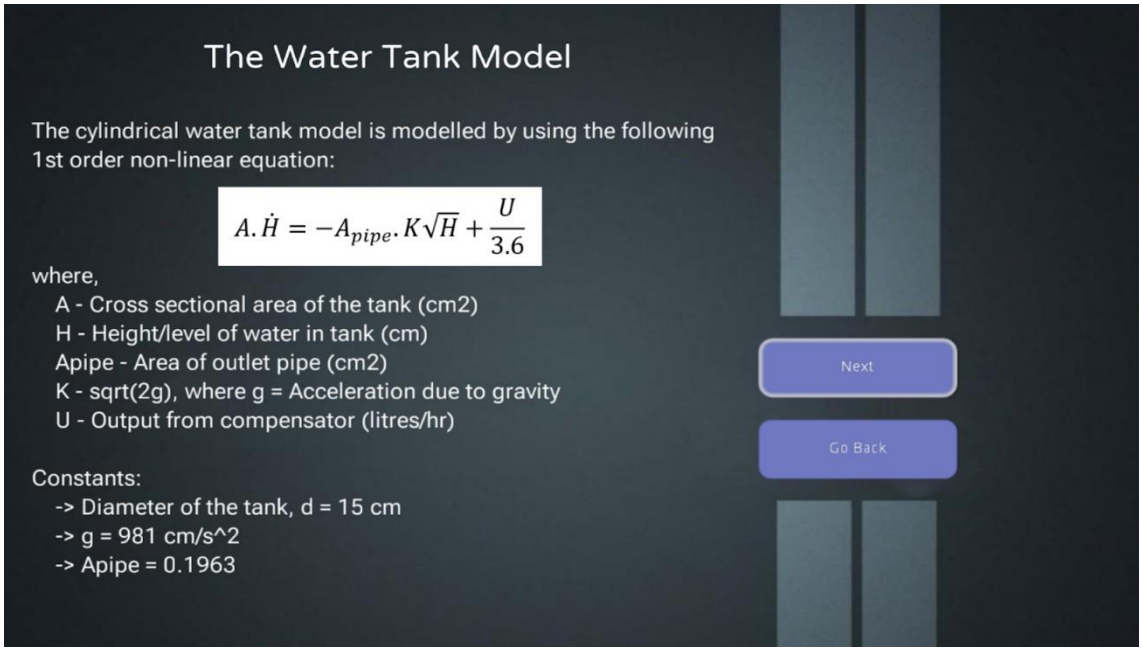



**Figure 4.3 Theory page 1**

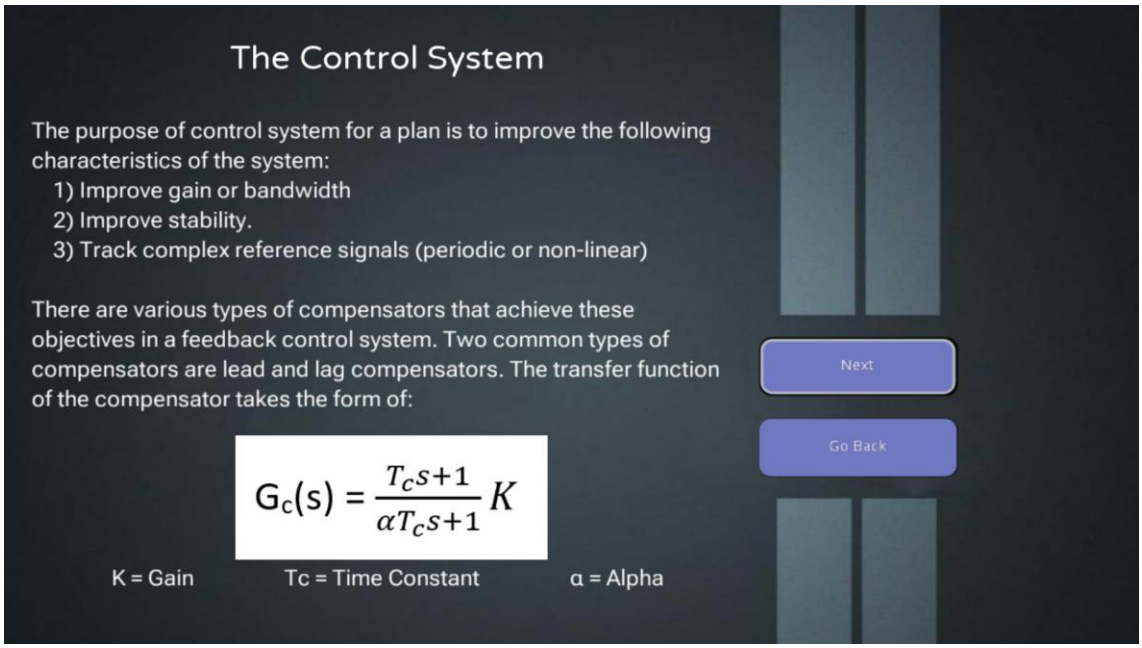



**Figure 4.4 Theory Page 2**

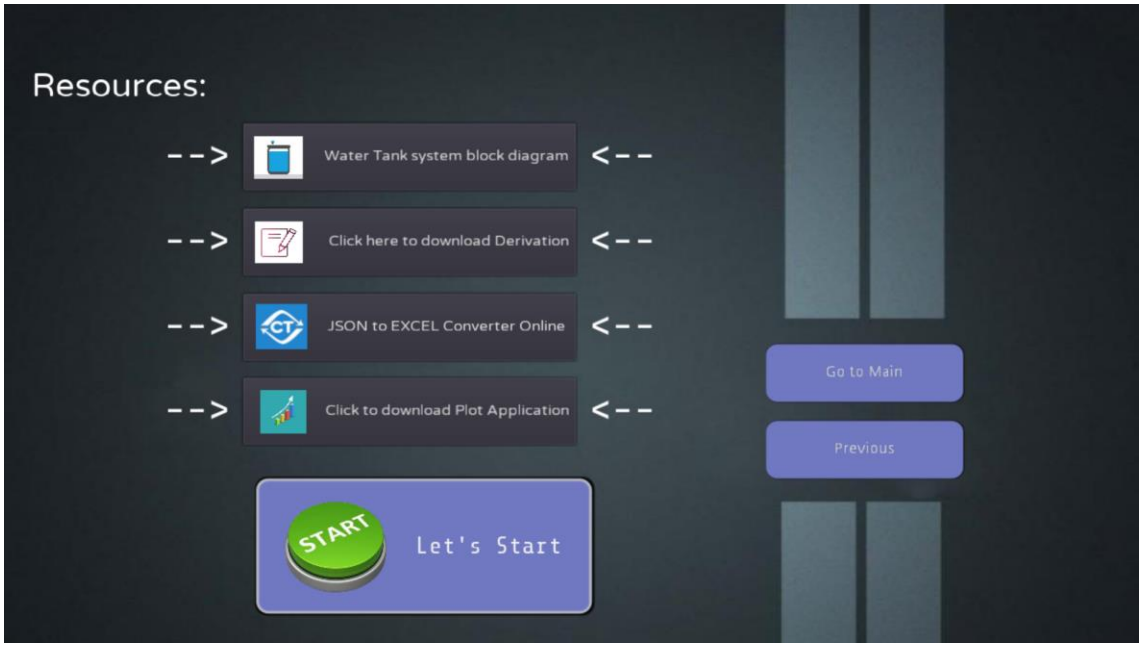



**Figure 3.5 Theory page 3 (Resources)**

## 4.2.4. Lab Screens

The Lab screen is divided into 3 scenes:

- In the first screen, the user has to select the experiment type: Open (or) Closed loop.

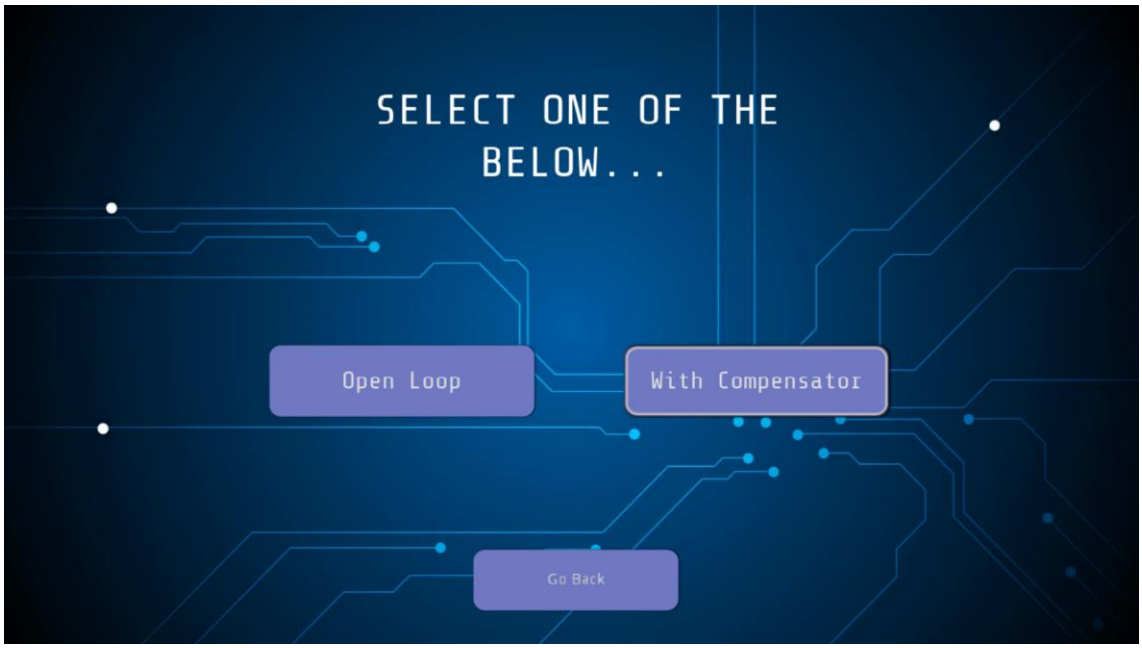



**Figure 4.6 Selection screen**

- In the Open-Loop experiment, the user has to set height and check the time taken to run the experiment.
- In the Closed-Loop (with compensator) experiment, the user has to set height, select the type of compensator, set the values of K, Tc and α and run the experiment.

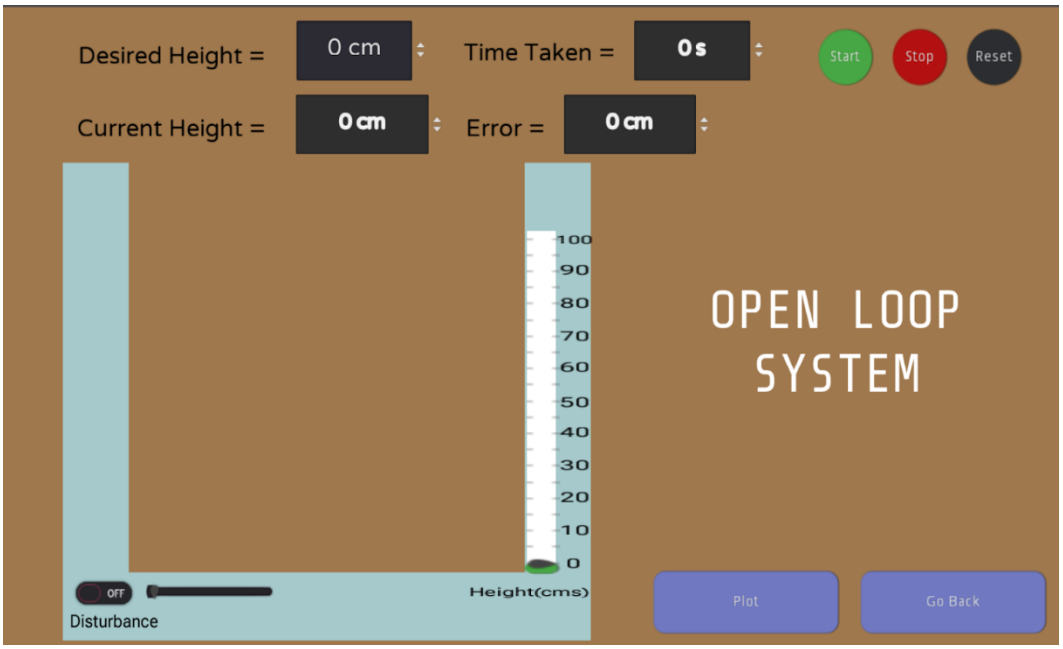



**Figure 4.7 Open-Loop Screen**

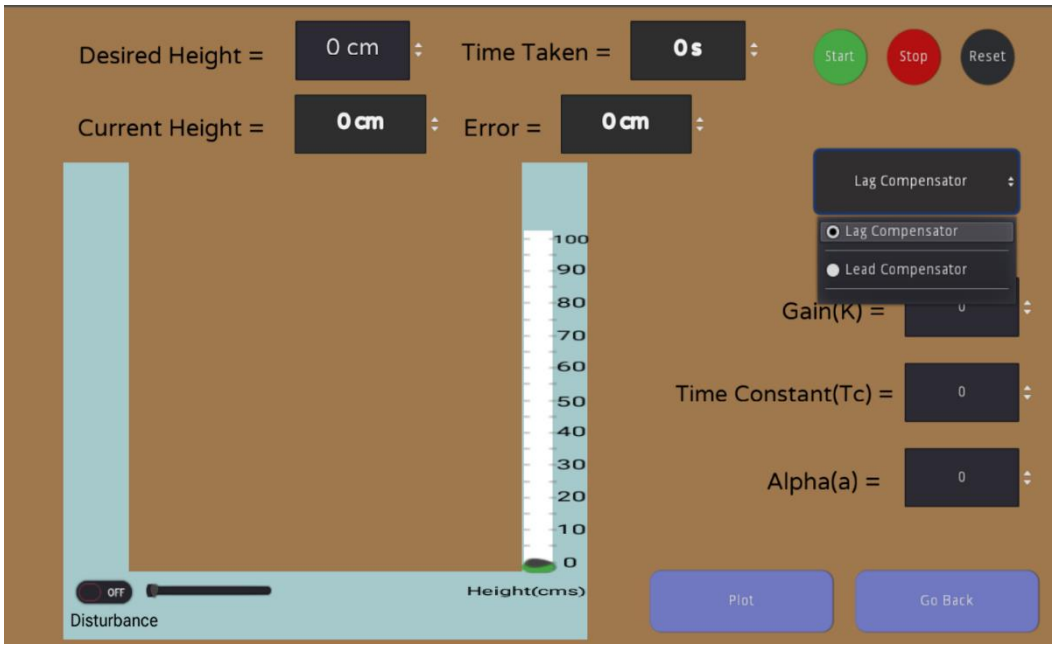



**Figure 4.8 Closed-Loop Screen**

- There is also a plot button available in both Open-Loop and Closed-Loop experiment.

# CHAPTER 5

# PRESENT WORK

## 5.1. CONCEPT

Compensators are additional components added to a control system (in our case, a water tank system) to compensate for deficit performance in the system. They can also improve the performance of a given system.

### 5.1.1. Need for Compensation

When a control system is tuned to increase its performance, it may exhibit unexpected behavior (for example, by increasing the gain value, you might improve stability or even create instability). To get the system to operate the way you want it to, you'll need to modify it and include a compensator. When you add a compensating network to your system, it can also increase the steady state accuracy of your system. Compensators can also minimize overshoot in your systems.

### 5.1.2. Types of Compensators

Compensators are generally classified into 3 categories:

1) Lead Compensator
2) Lag Compensator
3) Lag-lead Compensator

For this project, we only focus on the first 2 types of compensators, i.e., lag compensator and lead compensator. When you understand the performance of the first two types, the third one is understood automatically. It is because, lag-lead compensators are nothing but, it combines the characteristics of lag compensators and lead compensators.

## 5.2. BLOCK DIAGRAM REPRESENTATION

The block diagram representation of the water tank system along with the compensator is shown below:

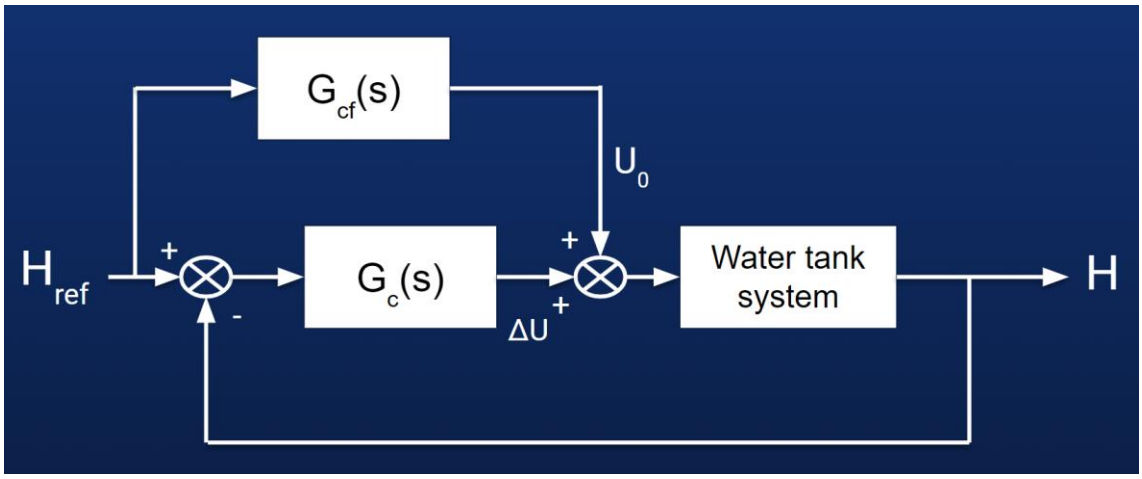



**Figure 5.1 Block Diagram Representation of the system**

Here,

$H_{ref}$ = Desired height/level in which the water in the tank is to be controlled (in cm)

H = Current height/level of water in the tank (in cm)

$G_c(s)$ = Represents the Compensator (either lead or lead) which is connected with the water tank system

$G_{cf}(s)$ = Represents a feed-forward compensation, which is an approach to improve the stability of a feedback control system.

U = Output of the compensator, which controls the flow rate (litres/hr)

### 5.2.1. Process

The above block diagram can be explained in the following way.

- Initially, the desired height ($H_{ref}$) is set. We allow the height between a range of 0 to 100 cm in our project.
- The error is calculated between the desired and current height.
- This error is passed as input to the compensator, $G_c(s)$.
- The compensator processes the error and sends the appropriate compensation (U) to the water tank system.

- Once again, the error is calculated, and this process keeps on repeating until desired height is equal to the current height.

## 5.3. THE WATER TANK SYSTEM

The system which we have chosen for demonstrating the performance of compensator, is a simple first-order, non-linear cylindrical water tank system. The level of water in the tank, i.e., the height of water in the tank, will be controlled by the compensator. The height of water in the water tank will be controlled by controlling the flow-rate of water into the tank. The reason for choosing water tank as the system is that, it is simple and easy to understand and visualize for beginners. As it is a first-order system, we do not have worry about complex dynamics of the system.

### 5.3.1. Design Equations of Water Tank System

The water tank system can be modelled using the following first-order, non-linear equation:

$$A.\dot{H} = -A_{pipe}.K\sqrt{H} + \frac{U}{3.6} \quad \ldots\ldots(5.1)$$

where,

A = Cross sectional area of the tank ($cm^2$)

H = Height/level of water to be controlled by the compensator in the tank (cm)

$A_{pipe}$ = Cross sectional area of the outlet pipe ($cm^2$)

K = $\sqrt{2g}$

g = Acceleration due to gravity

U = Output from the compensator (litres/hr)

### 5.3.2. Constants related to the water tank system

Following are the constants related to the water system model:

- Diameter of the cylindrical tank, d = 15 cm
  - Therefore, Area = $\frac{\pi d^2}{4} = \pi * \frac{15^2}{4}$ = 176.71 $cm^2$

- Area of outlet pipe, $A_{pipe}$ = 0.1963 $cm^2$
- Acceleration due to gravity, g = 981 cm $s^{-2}$

## 5.4. COMPENSATOR DESIGN

The compensator, when connected with the tank system, can control the height/level of water in the tank. It takes the error, i.e., the different between the desired height and current height, as its input. After processing, it tells the system to either increase or decrease the flow-rate in order to control the height of the water tank.

### 5.4.1. Design Equations of Compensator

The general equation of any compensator is given by the following equation:

$$G_c(s) = \frac{T_c s+1}{\alpha T_c s+1} K \qquad \ldots\ldots(5.2)$$

where,

K = Compensator Gain

$T_c$ = Time Constant

α = Attenuation Factor

As $G_c(s)$ is in s-domain, we will not able to use it in the script which we run. So, in order to convert it to a discrete domain, we perform bilinear transformation and then take the inverse Z-transform of it. By doing this, we will now be able to use the resultant discrete equation in our script and run it.

Starting with the general equation of compensator:

$$G_c(s) = \frac{T_c s + 1}{\alpha T_c s + 1} K$$

Perform bilinear transformation by substituting, $s \rightarrow \frac{2}{T_s}\frac{z-1}{z+1}$ (where $T_s$ is the sampling time period)

$$\Rightarrow G_c(z) = \frac{T_c\frac{2\,z-1}{T_s z+1}+1}{\alpha T_c\frac{2\,z-1}{T_s z+1}+1} K$$

Solving the above equation, we get

$$G_c(z) = \frac{2zT_c-2T_c+T_s z+T_s}{2\alpha z T_c-2\alpha T_c+T_s z+T_s} K$$

$$= \frac{K(T_s+2T_c)z + K(T_s-2T_c)}{(T_s+2\alpha T_c)z + (T_s-2\alpha T_c)}$$

Divide numerator and denominator by z,

$$\frac{U(z)}{E(z)} = \frac{K(T_s + 2T_c) + K(T_s - 2T_c)z^{-1}}{(T_s + 2\alpha T_c) + (T_s - 2\alpha T_c)z^{-1}}$$

Cross multiply,

$$T_s.U(z) + 2\alpha T_c.U(z) + T_s.z^{-1}U(z) - 2\alpha T_c.z^{-1}U(z)$$
$$= KT_s.E(z) + 2KT_c.E(z) + KT_s.z^{-1}E(z) - 2KT_c.z^{-1}E(z)$$

Now we take Inverse Z-Transform of the resultant equation:

$$T_s.U(k) + 2\alpha T_c.U(k) + T_s.U(k-1) - 2\alpha T_c.U(k-1)$$
$$= KT_s.E(k) + 2KT_c.E(k) + KT_s.E(k-1) - 2KT_c.E(k-1)$$

$$\Rightarrow \quad (T_s + 2\alpha T_c).U(k) + (T_s - 2\alpha T_c).U(k-1) = K(T_s + 2T_c).E(k) + K(T_s - 2T_c).E(k-1)$$

Finally,

$$U(k) = \frac{K(T_s + 2T_c).E(k) + K(T_s - 2T_c).E(k-1) - (T_s - 2\alpha T_c).U(k-1)}{T_s + 2\alpha T_c}$$

. . . . .(5.3)

So, now we have our compensator equation in discrete domain, and can be easily integrated with our script.

### 5.4.2. Lead Compensator

From the general equation of compensator, we can see that, 3 parameters control how the compensator is going to behave. They are: K, α and $T_c$. Out of these three parameters, α is the one which decides whether the compensator is going to be a lead compensator or a lag compensator.

When α is between 0 and 1, then the compensator will behave as a lead compensator. Therefore, the condition for a compensator to behave as a lead compensator is:

$$0 < \alpha < 1 \qquad \ldots\ldots(5.4)$$

The advantages of using a lead compensator are:

- It improves the overall response of the system. This leads to a faster response.
- It can increase the bandwidth of the system
- The maximum overshoot of the system (if present), can be decreased.

There are also few disadvantages while using a lead compensator. They are:

- The steady state error will not be improved.
- It is susceptible to lot of disturbances and noise.

### 5.4.3. Lag Compensator

When the value of α is greater than 1, then the compensator will behave as a lag compensator. Therefore, the condition for a compensator to behave as a lag compensator is:

$$\alpha > 1 \qquad \ldots\ldots(5.5)$$

The advantages of using a lag compensator are mentioned below:

- The steady state accuracy increases when we use a lag compensator.
- Noises in the high/low frequencies get attenuated.

Some disadvantages when we are using a lag compensator are:

- The response time of the system decreases, and as a result, it is slow.

## 5.5. IMPLEMENTATION IN GODOT GAME ENGINE

After deriving the system equations and compensator equations, as mentioned earlier, a game engine is needed to implement the task. Out of several game engines, we have chosen Godot v3.3.3 as our game engine to proceed further. Godot engine supports both 2D as well as 3D games. For the simulation of water tank system, we chose to go with 2D environment.

Now, the entire work can be divided into 2 sub-tasks:

1) To create a simple and neat user interface for the simulation of water tank system in the Godot Game Engine.
2) Develop the GDScript, which is a dedicated scripting language (almost similar to Python), for the simulation of the water tank system.

### 5.5.1. User Interface

One of the main objectives of this project is to provide a game-like feeling to the student. Godot game engine has the capability to build a very great User Interface, when compared to other gaming engines. Our aim is to create a user interface that is simple to use and interpret, allowing for more thorough game development.

In Godot, nodes are the fundamental building elements for making games. A node is a type of entity that can represent a specific game function. A node can display graphics, perform an animation, or represent a 3D/2D model of an item, among other things. The node also has a set of characteristics that you can use to customize its behavior. The functionality you require will determine the nodes you add to your project.

Scenes allow you to organize the code of your game in whatever way you wish. To develop unique and complicated node types, you can combine nodes. Godot comes with a number of tools for modifying 2D and 3D scenes as well as user interfaces. You can have as many of these scenes as you like in a Godot project. As the main scene of your application, the engine just requires one.

By combining nodes and scenes, along with scripts (i.e., GDScript), Godot will be able to fulfill your requirements.

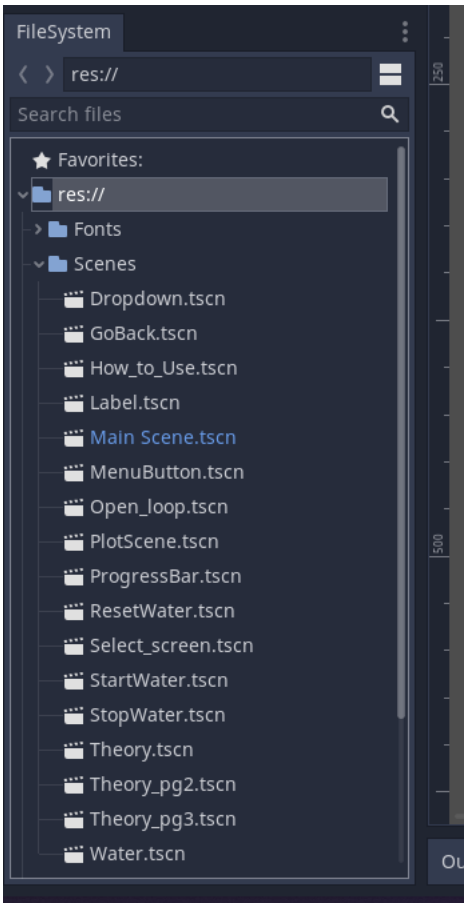



**Figure 5.2 Scenes for this project**

The above image shows the scenes (file extension: .tscn) which were created for this project. Some of the important ones are:

- Main Scene.tscn – This is the scene which loads up at the beginning when we run our project. It contains the title label and menu buttons to navigate to other parts of the game.
- Open_loop.tscn – This scene will run the simulation of the water tank system when in open loop condition (i.e., without any compensator added to the system).
- Water.tscn – This scene will run the simulation of the water tank system, when connected with a compensator. Here, you will have to enter the parameters of the compensators.
- How_to_Use.tscn, Theory.tscn, etc. – These are supporting pages, which students can view and understand how the project works and what should they do.

### 5.5.2. Godot GDScript

Although there are lots of languages, like C++, Visual Script etc., compatible with the Godot Game Engine, we decided to go with Godot's dedicated scripting language called the GDScript. GDScript is a dynamically typed, high-level programming language for creating content. It has a syntax that is identical to Python's. Its purpose is to be optimized for Godot Engine and strongly linked with it, allowing for a lot of content generation and integration flexibility.

There are two main inbuilt functions which we have used in our code:

- func _ready(): This function is called once when a particular node enters the scene. It is generally used for the declarations and initializations of variables.
- func _physics_process(delta): This function contains the logic of our simulation. It runs in an infinite loop at around 60 frames/sec (might vary based on the processor used). Here, delta is the time taken between 2 frames, which will be equal to 1/60 seconds or 0.0167 seconds.

#### 5.5.2.1. Getting the nodes

To reference a node as an object in our code, we can make use the "get_node()" function. Following are the nodes which we require for the simulation scene:

```
2
3  onready var startBut: Button = get_node("Start")
4  onready var stopBut: Button = get_node("Stop")
5  onready var resetBut: Button = get_node("Reset")
6  onready var dropdown: OptionButton = get_node("Dropdown")
7  onready var gain_box: SpinBox = get_node("Gain")
8  onready var tc_box: SpinBox = get_node("Time Constant")
9  onready var alpha_box: SpinBox = get_node("Alpha")
10 onready var progBar: TextureProgress = get_node("ProgressBar")
11 onready var desHeight: SpinBox = get_node("DesiredHeight")
12 onready var currHeight: SpinBox = get_node("CurrHeight")
13 onready var timeBox: SpinBox = get_node("TimeTaken")
14 onready var errorBox: SpinBox = get_node("Error")
15 # onready var timer = get_node("TimeTaken/Timer2")
16 onready var plot: Button = get_node("Plot")
17 onready var yPos: TextureRect = get_node("Cover")
18 onready var popUp: AcceptDialog = get_node("Plot_dialog")
19 onready var lag_alert: AcceptDialog = get_node("Alert_alpha_lag")
20 onready var lead_alert: AcceptDialog = get_node("Alert_alpha_lead")
21 onready var height_meter: VSlider = get_node("VSlider")
22 onready var disturbance_but: CheckButton = get_node("Disturbance_button")
23 onready var disturbance_slider: HSlider = get_node("Disturbance_button/HSlider")
24 onready var disturbance_label: Label = get_node("Disturbance_button/HSlider/dist_slider_val")
25
```

**Figure 5.3 Nodes in simulation scene**

### 5.5.2.2. Water Tank System Implementation

The first order, non-linear water tank system equation which we have derived before, is implemented in the code.

Rearranging Eq. 5.1, we can get “Height” as the output, which will be used to find the error. Following is its implementation in the code:

```
182  #       A = 3.14 * 225/4;
183          A = PI * pow(15,2)/4
184  #       A = 176.71
185          apipe = 0.1963
186
187          if H0 < 0:
188              H0 = 0
189
190          H = H0 + ( (apipe * -1 * sqrt(2*981*H0) + (U+Uf+Ud+Ur)/3.6)/A ) * Ts
191
192          Hm = H + (randf() - 0.5) * 0.5
193          if Hm < 0:
194              Hm = 0
```

**Figure 5.4 Line no. 190 denotes the water tank system equation**

In the above equation, H represents the actual height calculated from the water tank system equation. To make the system more realistic, we have made the height to oscillate plus or minus 5 cms from the height calculated. This is done using the randf() function in line no. 192. Therefore, “Hm” which is the “Measured Height” is going to be the output from the water tank system.

Also, you could see that we have added few extra variables to the original flow-rate, U. These variables are to make the system more realistic and stable. Following are the details about these variables:

- U = Original Flow rate which is controlled by the compensator
- Uf = Feed forward compensating term, which helps in improving the system stability
- Ud = Custom disturbance which can be added, to visualize how the response changes
- Ur = Randomness added to the flow of water, to make the simulation look more realistic

The Custom disturbance is an added additional feature. It denotes the disturbance added to the regular flow rate of water. It must be provided directly

while running the simulation. When you plot the responses, you can compare “With disturbance” and “Without disturbance”.

### 5.5.2.3. Compensator Implementation

For the compensator equation which we have derived before, we use two separate variables to denote the current and previous state of error and flow-rate.

- E(k-1) → E0
- E(k) → E
- U(k-1) → U0
- U(k) → U

The equations are implemented in two separate functions for better readability and accessibility. These equations keep updating the flow rate (U) of the water tank system for every Ts, where Ts = delta * 10. We have fixed the sampling time to be 10 times of delta, so that, the simulation could be a bit faster than actual (this is just to avoid the user waiting long time for the process to end).

```
257
258
259 func lag(Ts):
260     U = ( K*(Ts+2*tc)*E + K*(Ts-2*tc)*E0 - (Ts-2*a*tc)*U0 ) / (Ts+2*a*tc)
261
262 func lead(Ts):
263     U = ( K*(Ts+2*tc)*E + K*(Ts-2*tc)*E0 - (Ts-2*a*tc)*U0 ) / (Ts+2*a*tc)
264
265 #   print("Gain = ", K)
266 #   print("tc = ", tc)
267 #   print("Alpha = ", a)
268 #   print("U = ", U)
269 #   print("E = ", E)
270 #   print("U0 = ", U0)
271 #   print("E0 = ", E0)
272 #   print("Ts = ", T)
273
274 func laglead(T):
275     pass
276
277
```

**Figure 5.5 Compensator implementation**

(Here, laglead() is not yet implemented, which is left for future development, if needed)

### 5.5.2.4. Writing to file

Plotting option is not available inbuilt within Godot. Although there are some external libraries which will allow to plot, we considered a different

approach for plotting. That is, we write the data to a file, which can then be stored and plotted anytime, anywhere. During the simulation (for both open-loop and closed-loop), we simultaneously store the time (in secs) and height (in cm) values in a Godot dictionary as key-value pair.

```
166
167             # Adding to dict
168             dict[timeBox.value] = Hm
169
```

**Figure 5.6 Write data to Dictionary**

When the student presses plot, this dictionary is stored as a JSON formatted file (Godot game engine can write dictionaries as only JSON formatted files, and not in any other extension). The location of the file can also be chosen through the popup dialog box which appears when plot is pressed.

```
233
234     if plot.pressed:
235 #       writeToFile()
236         var x = timeBox.value
237         while x <= 200:
238             x = x+0.1
239             dict[x] = Hm + (randf() - 0.5) * 0.5
240         popUp.popup_centered_ratio(0.5)
241
242 #   print("Height : ", ch)
243
244 func _on_Plot_dialog_file_selected(path: String) -> void:
245     writeToFile(path)
246
247 func writeToFile(path: String):
248
249     if path[len(path) - 1] == '/':
250         path = path + "Compensator.txt"
251
252     var saveFile = File.new()
253     saveFile.open(path, File.WRITE)
254
255     saveFile.store_line(to_json(dict))
256     saveFile.close()
257
```

**Figure 5.7 Write the data to file**

The above three functions work together and write the data to the desired location. Before writing the dictionary, we just pad the final height of the water tank system, till 200 seconds. This is done so that when two graphs are plotted, it

can be conveniently compared when it has been plotted for the same amount of time.

### 5.5.3. Plotting the response

As seen from the previous step, we write the (Time, Height) values in a dictionary as (Key, Value) pairs respectively. This dictionary when gets written as a file, it becomes as a JSON formatted file. There are lots of ways to plot the response, which is now in JSON format. As illustrated in Chapter 4.1.3.3, the resources section contains ways to plot the response.

#### 5.5.3.1. Using Online JSON-XLSX/XLS Converter

There are lots of online converters to convert JSON to XLS/XLSX files, which are nothing but excel files. The JSON formatted file has to be just uploaded, and you can download after the conversion gets over. Once the excel file is downloaded, you can open it in Microsoft Excel or Google Sheets, from which you can easily plot the response.

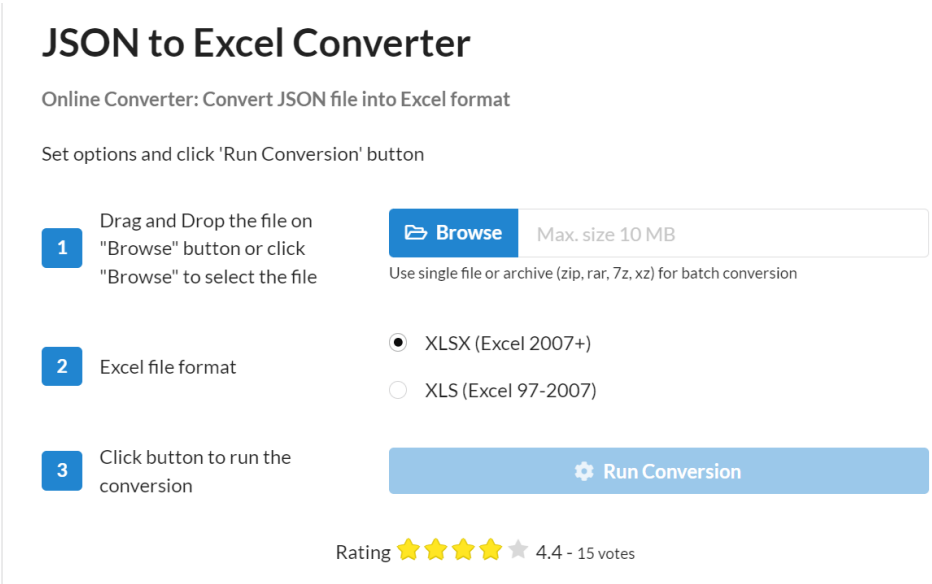



**Figure 5.8 Online JSON-XLSX/XLS Converter**

#### 5.5.3.2. Using Power Query feature in MS Excel

You can directly extract the data in the JSON formatted file into an Excel table. This is done using Power Query Editor in MS Excel. To use this option:

1) Open a blank worksheet in Excel.
2) Go to “Data” tab.
3) Select the “Get Data” dropdown in the “Get & Transform Data” section.
4) Choose “From File” → “From JSON”.
5) Select the JSON formatted file. This will open the Power Query Editor.
6) Just select “Into Table” in the top-left of the editor.
7) Click “Close & Load” in the top-left of the editor.

This will generate the table with the Time and Height values. Now you plot it by selecting “Line Chart” from the “Charts” section in the “Insert” tab.

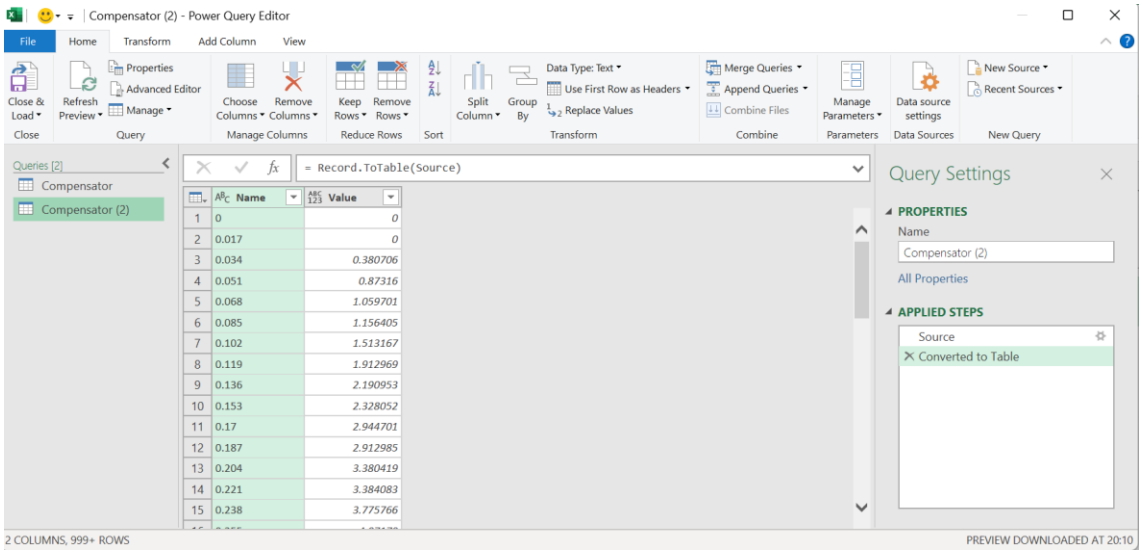



**Figure 5.9 Power Query Editor in MS Excel**

## 5.5.3.3. Using Plot App developed using Python

Plot App is an application which is developed as a part of this project, to make it comfortable for the student to plot the JSON file and perform comparison, analysis and even save it to refer in the future. The app is developed using Python language separately. We use the Tkinter library for the Graphical User Interface, and the matplotlib library for plotting the graph. Any number of files can be chosen for plotting. We have made each plot come one on another, so that it will easy to visualize and compare between two plots.

The python code which gets stored in a .py format is converted to .exe using the pyinstaller library. Then, we compiled the executable file along with other build files into a setup using Inno Compiler, which is a free setup compiler. Finally,

we uploaded the setup file to mediafire, so that it can be shared. Now, the setup can be downloaded from the provided link, by clicking on the respective button in the Resource page of the Theory Scene.

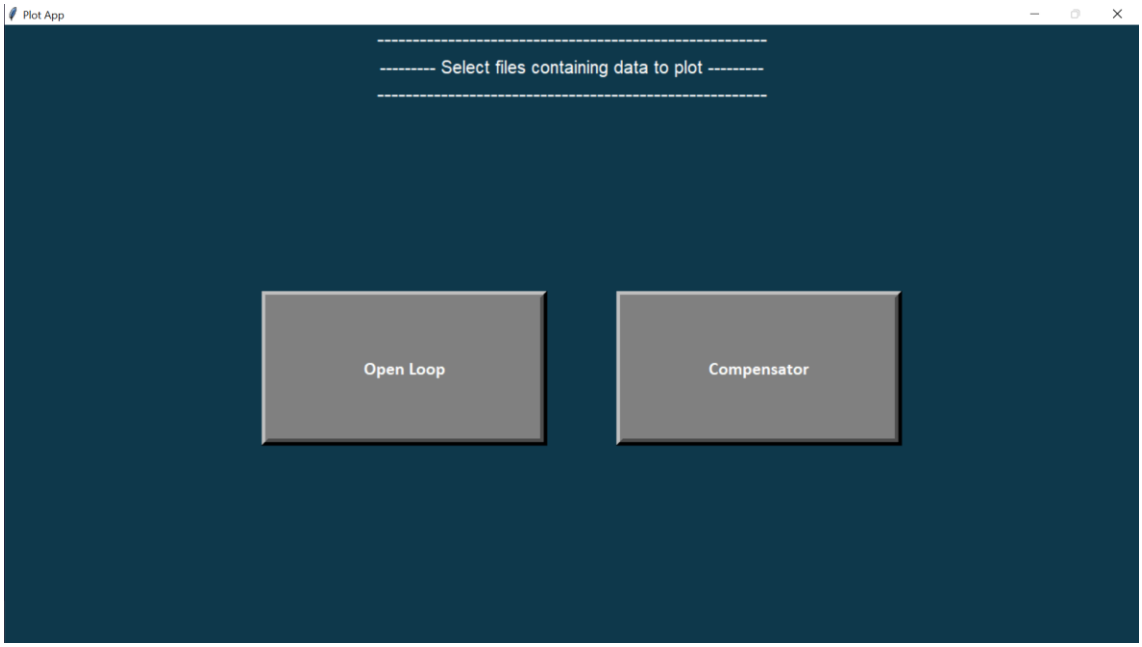



**Figure 5.10 Plot App**

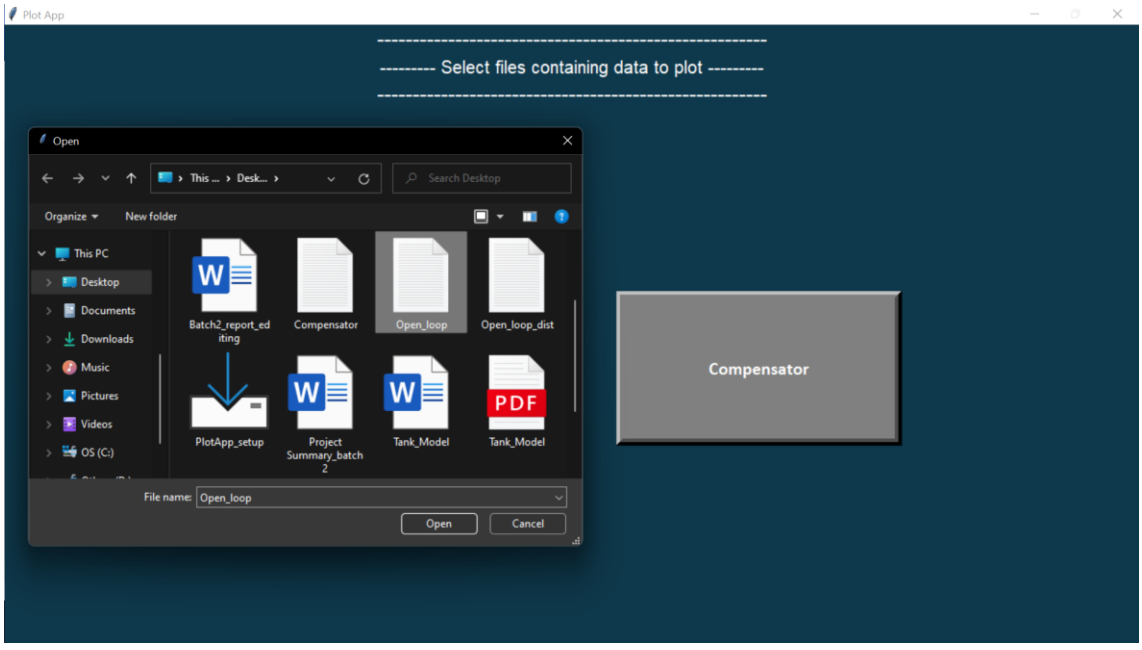



**Figure 5.11 Plot App - Open Dialog Box**

## 5.6. METHODOLOGY

After completing creating the scenes and scripts, the project is now ready to run. When the game is launched, the menu screen comes as the first scene. You can navigate to How to Use/Theory pages to take a look. Then when you press Start, you will get a screen with 2 options: Open Loop and With Compensator. You can then perform your simulation.

### 5.6.1. Calculating parameters of compensator

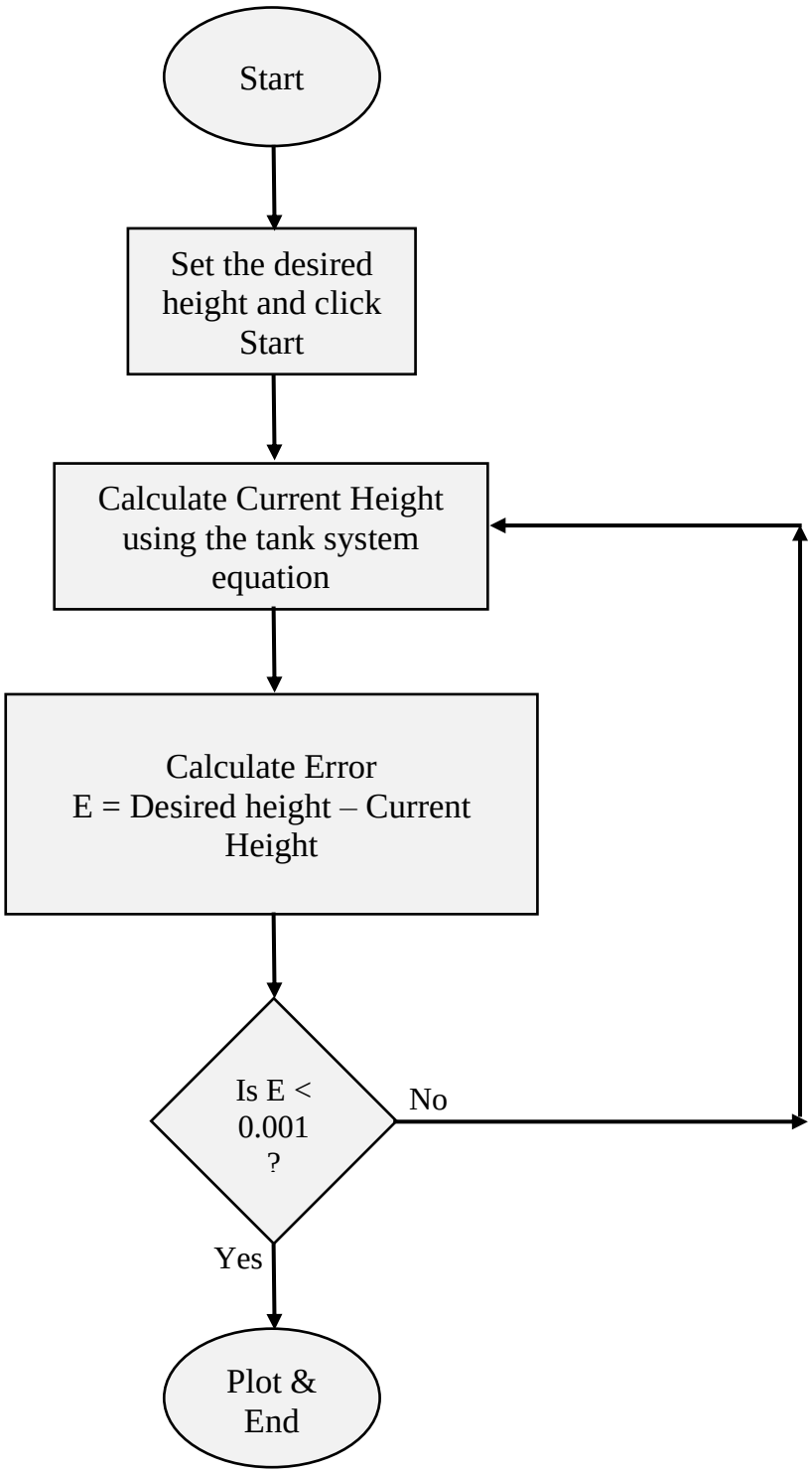
There are mainly 2 ways of using this project.

Firstly, the student can just play around with the parameters and see how the response changes. For example, you can increase the gain of a compensator and check if the response gets faster, or check for the steady state error. So, you can modify the parameters on the go, and analyze it.

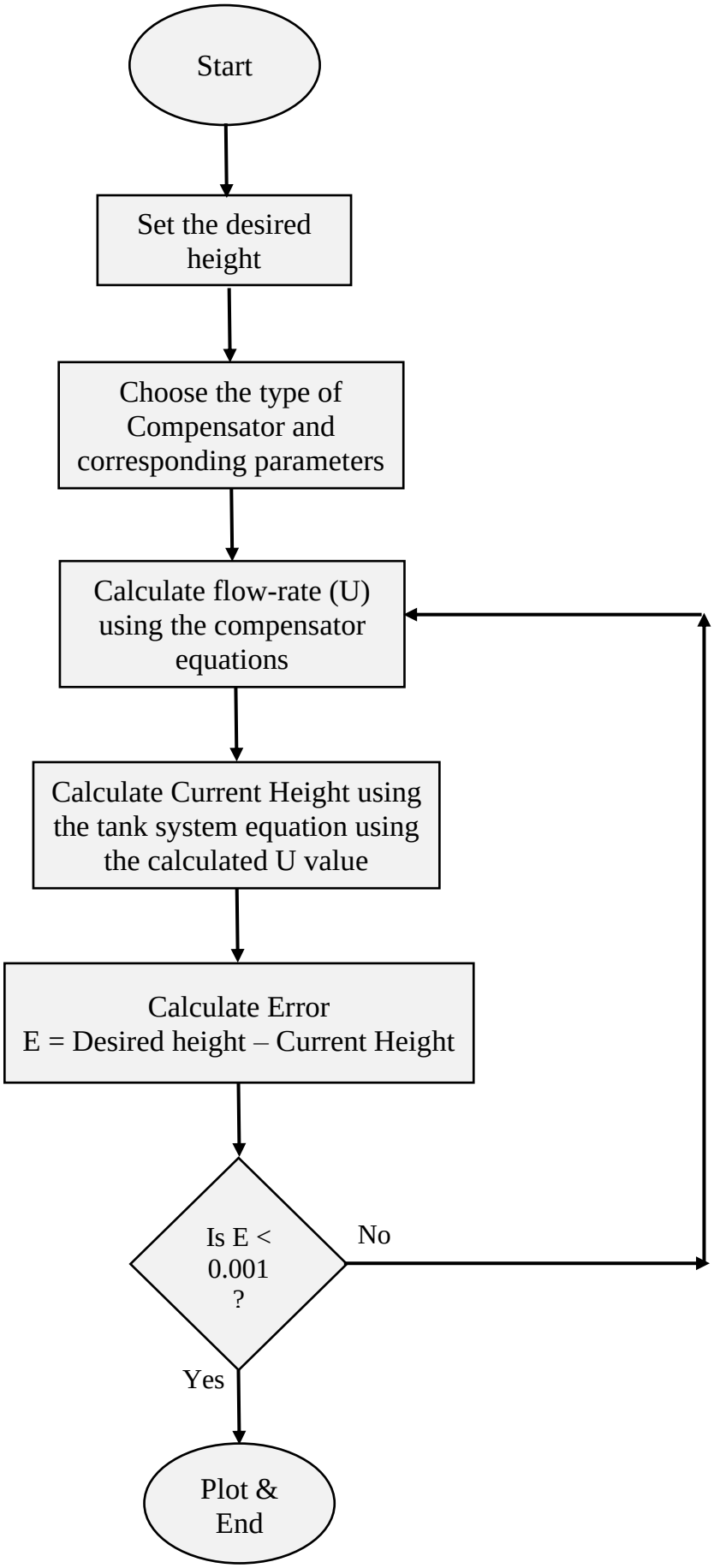
Secondly, the student can fix a desired response which is to be obtained from the simulation. To do this, the student must design the compensator by calculating the parameters, i.e., K, Tc and α, for a desired height and desired response. As the system model is non-linear, the student has to linearize the system, so that, the parameters can be calculated. To linearize the non-linear equation, the student can make use of the Taylor Series Linearization method, which will linearize the system around an operating point. After linearizing the system, the linear transfer function can be used to obtain the parameters. For this, the student can take the help of “sisotool” in MATLAB.

By these ways, the students can analyze their theoretical knowledge, which they acquire during their classroom sessions, and apply them in our Virtual Lab and verify their understandings.

### 5.6.2. Open-Loop Control Flowchart



### 5.6.3. Compensator Control Flowchart



# CHAPTER 6

# RESULTS AND DISCUSSION

## 6.1. GRAPHS

As discussed in the previous chapter, we follow the below procedure to obtain the graph:

1) Run the simulation.
2) Values get stored in dictionary during the simulation.
3) Write the dictionary to a file, which will be a JSON formatted file.
4) Plot the data in the file using any of the method discussed earlier.

### 6.1.1. Plotting in MS Excel

When we use Online JSON-Excel Converter, Power Query Editor, or any other techniques to convert the JSON formatted file into a table in MS Excel for plotting, we can get the response plot in a Line Chart format. There are lots of styles and formatting options available in Excel, which you can use to make your response in a way you want. One of the disadvantages is that, you will not be able to plot two different responses in same area. We get two different graphs separately. This method is also slightly lengthier, compared with the Plot App method.

Following is an example of plot in MS Excel:

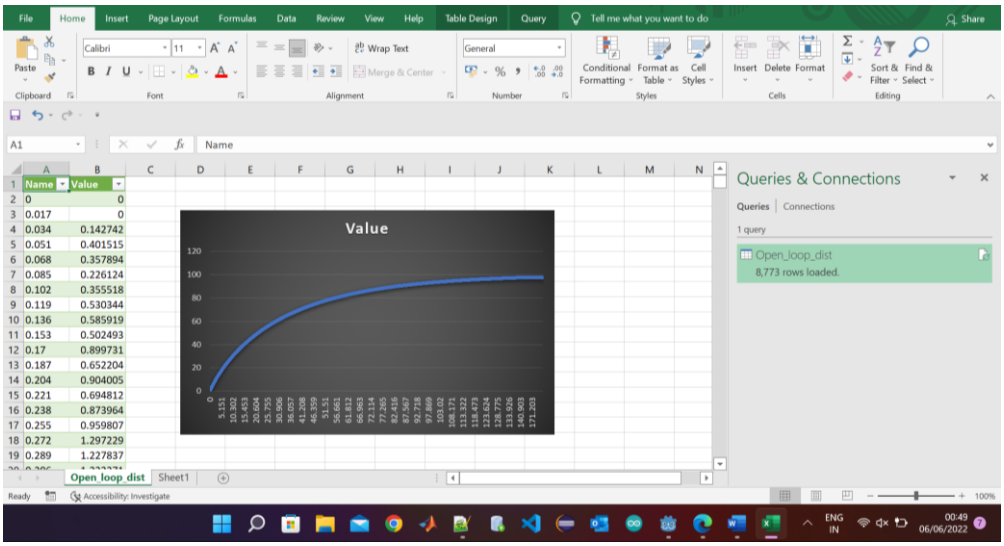



**Figure 6.1 Plot in MS Excel**

### 6.1.2. Plotting using Plot App

Plot App is a python application which is created as a part of this project. This application has some advantages over the other ways. It can plot more than one response in the same plot area. You can save the graph in your local storage and use it whenever you want it. The interface is very simple and easy to use. The setup for this app can be easily downloaded by clicking on the button provided in Resource section in the Theory Scene. Then, you can install it in your local PC, and use it whenever you want.

Following is an example of plot using Plot App:

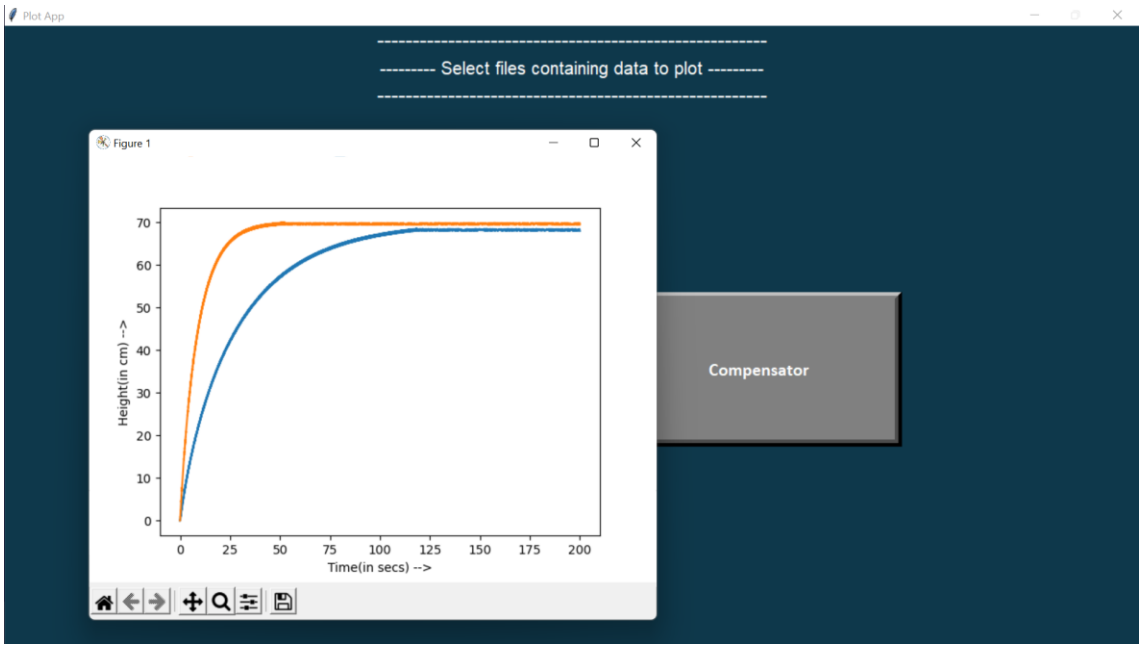



**Figure 6.2 Plot using Plot App**

### 6.1.3. Demonstration

This section will demonstrate the simulation and the resultant response.

- First, we perform simulation in Open-loop.
- Set the Desired Height = 70 cm.
- For open loop, we don't need to set any parameters. So, we simply click start to start the simulation.
- Wait for the Current height to reach near 70 cm, i.e., the desired height, and you will be able to see the error approaching 0 cm.

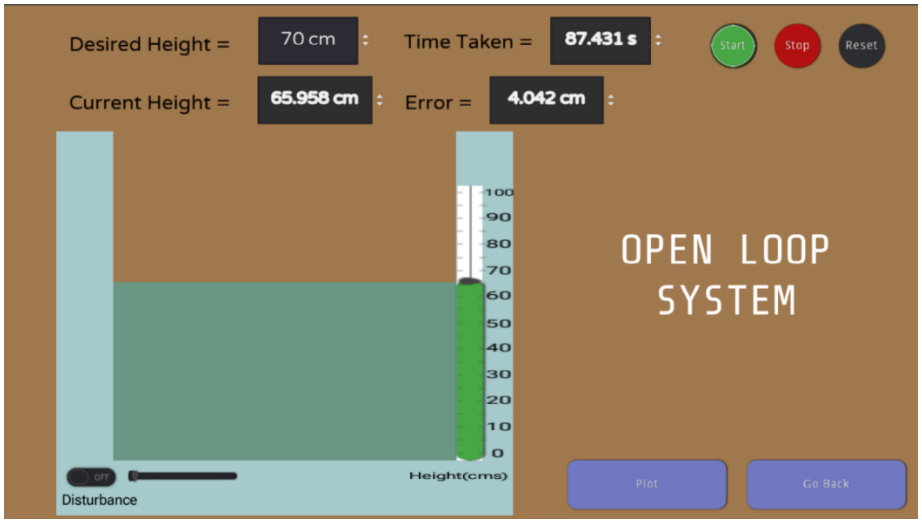



**Figure 6.3 Open-Loop Simulation**

- Click plot and choose the location and name of the file to store the data.
- Next Go back and select With Compensator in the Select screen.
- Here, set Desired Height = 70 cm.
- Choose the type of compensator, say Lag Compensator.
- Enter the parameters:
    - K = 5
    - Tc = 0.1 s
    - α = 1.2

(These are random values, just for demonstration purpose)

- Click on Start and wait for the Current height to reach near 70 cm, i.e., the desired height.

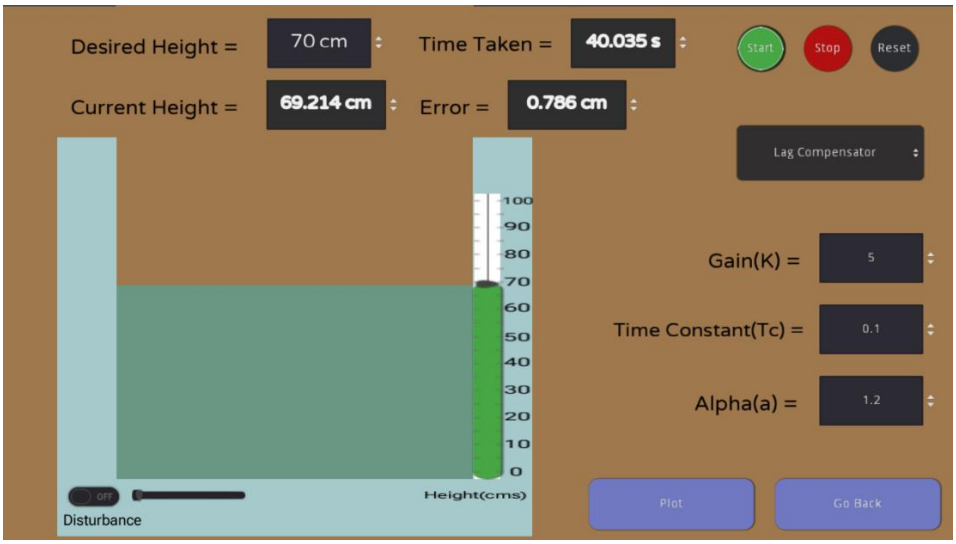



**Figure 6.4 Compensator simulation**

- Once again click plot and choose the location and name of the file to store the data.

The response of these two simulations is shown below:

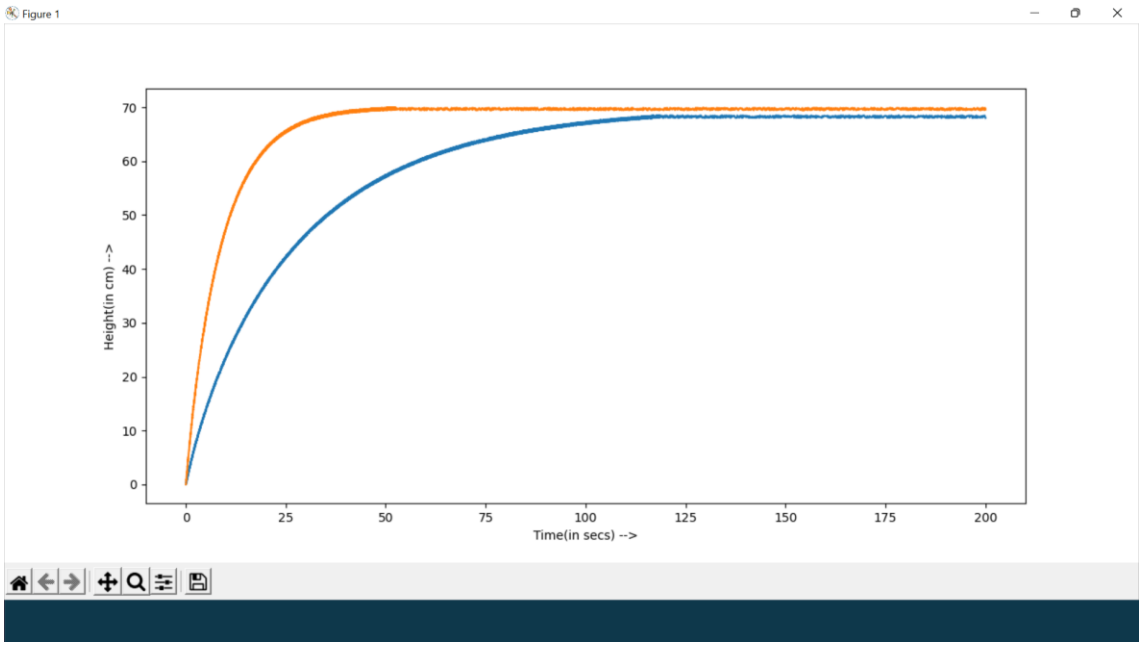



**Figure 6.5 Result of simulation**

### 6.1.4. Demonstration (with disturbance added to normal flow)

As explained earlier (in Chapter 5.5.2.2), you can add some disturbance to the normal flow of water in the tank. Let us see, how does the response get affected when you add disturbance.

- Choose Open Loop and set desired height to 70 cm (you can also try it with compensator).
- Click start to run the simulation.
- Click stop when the Current Height reaches somewhere around 30 – 40 cms
- Turn on the disturbance, and set it to a value, say 100 litres/hr. Note: If the disturbance is negligible, you will not be able to see the difference in the plot.
- After setting the disturbance, click start to resume the simulation.
- Wait for the Current Height to reach near 70 cm.
- Once the simulation is over, click plot to generate the required JSON formatted file.
- Open Plot App, and choose the newly generated file, and then the normal open loop file, which we have generated in the precious section (Chapter 6.1.3).

  Following is the result of the simulation:

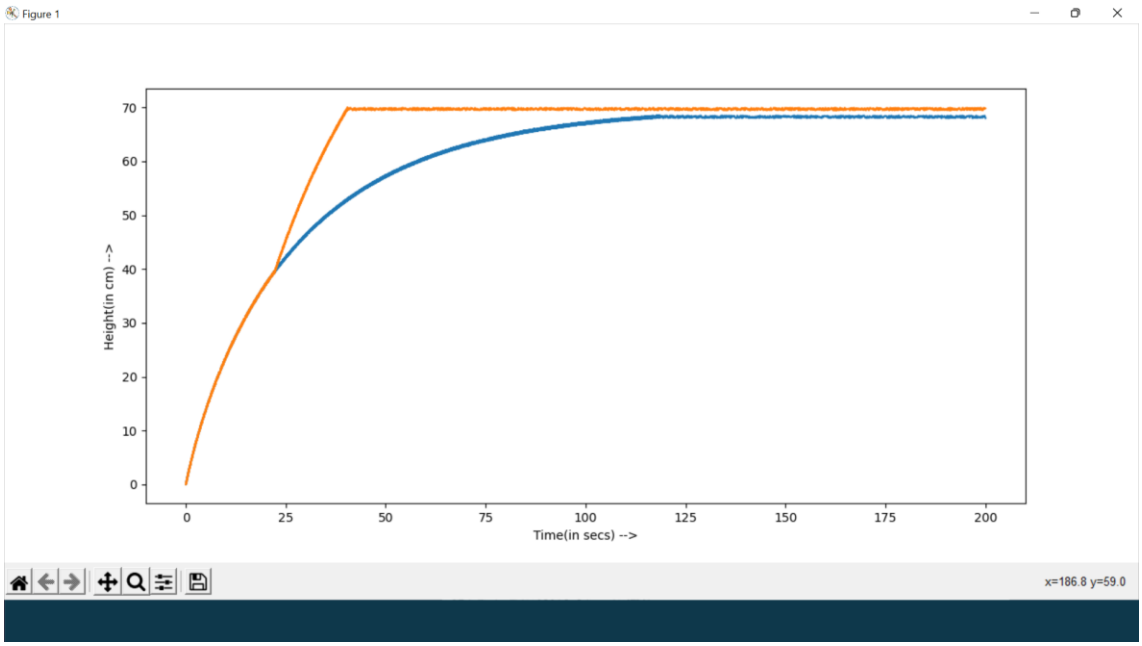



**Figure 6.6 Result of simulation (along with disturbance)**

## 6.2. RESULTS

After completing the simulation, the student must have plotted the response in a graph. Various parameters and scenarios can be tried out and plotted. The graphs can be compared with each other and analyzed accordingly.

From this experiment, the students will get a good practical understanding of how compensators operate, and how does the performance of compensators change, when tried with different parameters. It will enlighten the students with a knowledge of which compensator to use, along with the optimum parameters, in various scenarios. The difference between lag and lead compensators should be inferred by the students. Using a lag compensator, will improve the steady state error, but it will decrease the speed of response of the system. In the same way, using a lead compensator will improve the speed of response of the system, but the error becomes more susceptible.

This project would have given a more interesting as well as intuitive way of understanding concepts and learning things.

## 6.3. DISCUSSIONS

In today's scenario, although the number of virtual labs keeps rising, the reality is that, virtual labs have not yet completely replaced the traditional physical labs. And the reason behind this is quite obvious. Students will not be able to get the feel of touching and seeing the real equipment/components. Virtual lab can just give them visual images and animations, which might not satisfy the needs for a student. Another drawback of a virtual lab is that, it might require some prerequisite skill to operate on the application. You cannot always expect the user to be skillful enough to operate even a PC.

From the above discussion, it is clear that even though Virtual Laboratories are the current trend, we cannot simply close the Physical Laboratories forever. Virtual labs when working alongside physical labs, can make a tremendous impact in helping the students and the staffs. When the physical labs cannot be accessed,

students can always go for the virtual ones. Even in times of doubt, they can experiment with the virtual labs

# CHAPTER 7

# CONCLUSION

By the end of this project, we were able to achieve our goal of creating a Virtual Lab in the Control System domain. We made it look different from the existing solutions by adding a game-feel to our Virtual Lab. The performance of compensators was demonstrated using the simulation and the graphs were plotted. Working with the Godot game engine and learning a new scripting language like GDScript was indeed a wonderful experience. When you want to create a 2D environment, you can always prefer Godot engine rather than others, because Godot is simpler and more efficient than other application engines in the market.

This project can still be developed in the future by adding various other features. Firstly, the lag-lead compensator can be added as a part of this experiment. It would just require an extra time constant parameter. Next, you can add some more systems and make them controllable using compensators. This allows the student to choose a system of his choice and play around with it. Lastly, apart from compensators, other concepts in Control Systems like the PID controller can be added as experiments to the project.

This project is currently only available locally in PCs (if you download or copy the project to your PC) and you need Godot Game Engine to run it. As everyone will not be having the game engine to run it, the project can be exported as a html file and uploaded to a website. This is left for the future scope, when we get the domain rights for the website.

Finally concluding, this Control System Virtual Lab helps not only students but also the teaching staffs understand about the performance of Compensators. And this project also portrays the importance of Virtual Labs in today’s world.

# REFERENCES


[1] J. R. Brinson (2015), “Learning outcome achievement in non-traditional (Virtual and remote) versus traditional (hands-on) laboratories: A review of the empirical research”, Computers & Education, vol. 87, 2015, pp. 218 – 237.

[2] Mrityunjay Kumar et al. (2018), "Usability Analysis of Virtual Labs", 2018 IEEE 18th International Conference on Advanced Learning Technologies (ICALT), July 2018.

[3] V. Potkonjak et al. (2016), “Virtual laboratories for education in science, technology, and engineering: a review”, Computers & Education, Elsevier, vol. 95, April 2016.

[4] M. Travassos Valdez et al. (2014), "Virtual labs in electrical engineering education - The VEMA environment", 2014 Information Technology Based Higher Education and Training (ITHET), September 2014.