CENTRALISED CONTROL ALGORITHMS FOR SMART GRID OPERATION

AN ELEC3875 INDIVIDUAL ENGINEERING PROJECT REPORT
SUBMITTED TO THE SCHOOL OF ELECTRONIC AND
ELECTRICAL ENGINEERING
OF UNIVERSITY OF LEEDS
IN PARTIAL FULFILLMENT OF THE REQUIREMENTS
FOR THE DEGREE OF
BACHELOR OF ENGINEERING

Jialeng Guo
May 2019

Abstract

Maintaining the frequency at the nominal value in the power grid is one of the most elusive and long-standing challenges in smart grids. This project tackles the problem of frequency changes: how to design an algorithm to build a centralised Secondary Frequency Control (SFC) and to analyse the performance of the smart gird. On the one hand, we think that our SFC algorithm can maintain the frequency. On the other hand, we would need an efficient and concise strategy to analyse the performance of the system if we want to build an optimal controller to ensure that the frequency of the electricity network is always restored to its nominal value when disturbances occur in the system.

In this project, we focus on PI Control: the most common control algorithm by far and the standard algorithm in SFC. Compared to traditional grids without SFC, this algorithm has proven to be more effective in maintaining the frequency.

This project consists of two parts. In the first part, we aim to understand the physical theory behind SFC and PI control and present our efforts at building effective SFC models.

In the second part of this project, we test our algorithm in RAMSES [1] [1], [2] based on Nordic Grid scenario. In particular, 1) how we test our system in a low time delay; 2) how we analyse the impact of different time delays; 3) how we analyse Emergency Control.

[1] RAMSES is a time-domain dynamic simulator for future electric power systems. RAMSES document: https://ramses.paristidou.info

Acknowledgments

A special thanks to my supervisor Dr Petros Aristidou. He always has a very insightful, high-level view of the field while he is also uncommonly detail oriented and understands the nature of the problems very well. More importantly, Dr Aristidou is an extremely caring and supportive supervisor that I could not have asked for more.

Collaboration is a valuable lesson that I have learnt, which is also a precious part of my undergraduate stage. I thank Sultan Alghamdi for never reserving his minds when I tried to seek for his help. I am glad that my research can help part of his doctoral subject. It was a very unique and rewarding experience for me.

I would like to thank Dr Zoran Ikonic. His passionate lecture (ELEC2540 Control Systems) and impressive experiment last year has enhanced my understanding of PI control.

I am greatly appreciative to Walter Roberson from MATLAB for helping me find the clue to solve the problem with plotting a 3D triangle surface plot without a clearly relationship between variables.

Lastly, I would like to thank other staffs and my fellow schoolmates in the School of Electronic and Electrical Engineering, without them, this project would not be finished smoothly: Anna de Jong, Nathan Smith, Zikang Qian and Al Dabashi.

Contents

List of Tables

List of Figures

Chapter 1

Introduction

1.1 Motivation

The motivation for doing this project contains a vision of restructuring and sustainability of energy in the future.

1.1.1 Problems in the Grid

1.1.1.1 Energy Crisis

The energy crisis is one of the most essential and critical crises in the 21st century. Nowadays, non-renewable resource still consists of a large proportion in the energy system. Non-renewable resource, or finite resource, is depleting. Although we might not meet the complete depletion of non-renewable resources in the future 50 years, based on Hotelling's "Economics of Exhaustible Resources", David Ricardo proposed that [3] as the historical production stock accumulates, higher grade ores get depleted and the producer resorts to lower grade ores, sustaining greater extraction costs. It means, the extraction costs rise, and the price of the products based on ores rise. Thus, we can assume that the price of most of the non-renewable resources, like oil, coal and gas, rise since these have similar properties with ores.

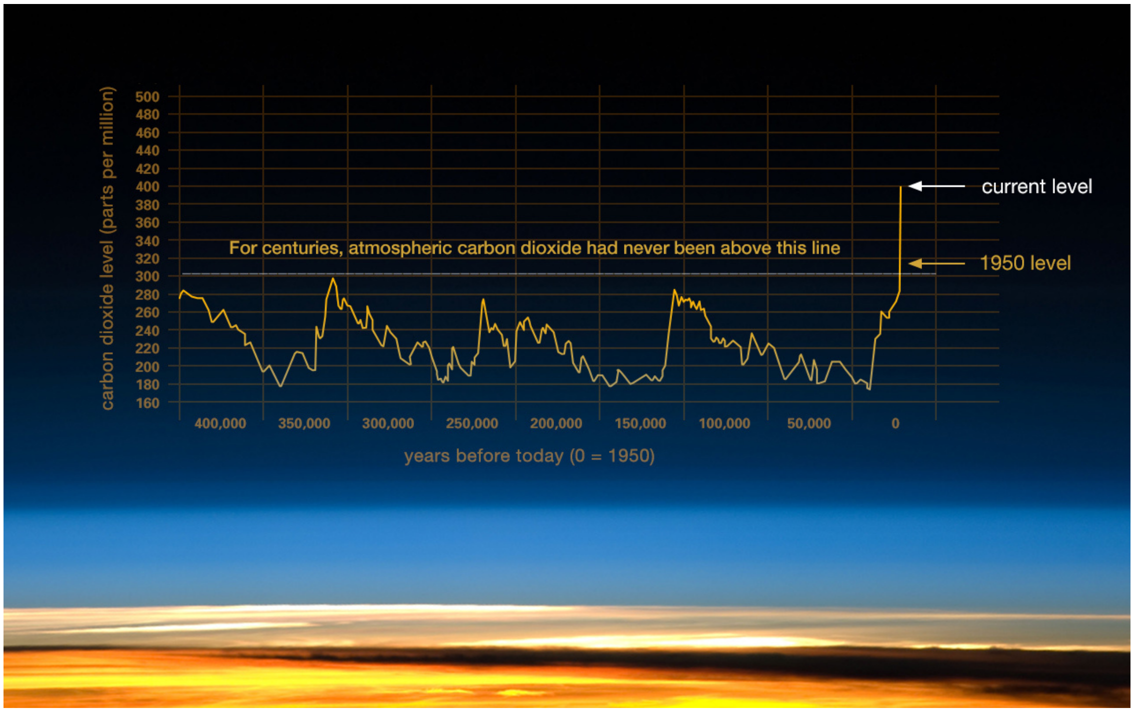


Figure 1.1: The evidence that atmospheric CO2 has increased since the Industrial Revolution began. Image courtesy: https://climate.nasa.gov/evidence

1.1.1.2 Climate Change

According to the paper from Nature, [4] climate change happened in the past 70 years. It has already had effects on the environment around us. Glaciers are shrinking and ices are breaking up earlier on the lakes and rivers. Most climate scientists agree that [5] it is the human expansion that causes the global warming. As we know, carbon dioxide (CO2) is a significant component of the atmosphere. [5] Atmospheric CO2 concentration has been increased by more than a third since the Industrial Revolution began. More importantly, atmospheric carbon dioxide has exceeded the highest level in the past 400,000 years.

1.1.1.3 Conflicts and Wars

An unbalanced energy distribution causes conflicts and wars. The wars, like Gulf War, are more or less derived from energy issues since WW2.

1.1.1.4 Power Interruption

With the use of unreliable electrical grids, power interruption occurs especially when a natural disaster, such as typhoon, earthquake, and wildfire, happens. Smart grids are more reliable than traditional grids. It's possible to be build a dynamic technology, like Secondary Frequency Control (SFC), that grid operators need. Self-healing is possible when the storms hit, or physical attack occurs. Therefore, consumers do not encounter power interruptions.

1.1.1.5 Equal Rights to Use Natural Power

Everyone should have his/her right to use natural resources equally. However, in most countries, the fact is that a few large companies dominate public funds and gradually form a monopoly market. Entrepreneurs use their centralised power to restrict consumers' right to disagree.

Technologies should begin with the customers. It is connected with intelligent devices that always communicate with each other. The system creates and stores energy throughout the day and any extra energy flows back onto the grid to power neighbours and businesses. Those smart energy buildings will then power the entire communities. It's fairly distributed, clean, and more cost-effective.

1.1.1.6 New Issues from Clean Energy

Using clean energy from renewable resource could be one of the ways to solve the problems above! In fact, recently, California Assembly passed a bill requiring 100 percent of the state's electricity to come from carbon-free sources by the end of 2045. In China and Germany, renewables are outgrowing their grids. The public is accepting the idea of using electric cars instead of fuel cars.

However, these renewable energy sources which interfaced with national or state power systems has introduced new issues on stability, resilience, and reliability. Thus, power networks are under modification.

For countries like Switzerland, where 62% of electricity comes from renewable sources, it's another situation. Although it's really friendly to the environment, energy instability has also increased. Sun doesn't always shine, wind doesn't always blow, and water doesn't always flow.

1.1.2 Solution: Secondary Frequency Control

As these highly variable sources come to represent a growing portion of the grid, it becomes more and more important to develop an accurate and validated model to represent these units in the computational tools used to analyse the ancillary services of smart grids.

Smart grid powers the modern city, and Secondary Frequency Control is one of the most important parts in smart grids to ensure the energy security in electricity systems and to allow the increasing renewable energy penetration. Secondary Frequency Control ensures the frequency of the electricity network is always restored to its nominal value when disturbances occur in the system. The frequency is one of the key "health" indicators of smart grids. Actively monitoring and controlling can ensure system security. This is done by remotely controlling the power output of generating units (both conventional and renewables) through a communication network.

1.2 Thesis Outline

This project consists of two parts: PART I Algorithms and PART II Test Case Scenario (Nordic).

PART I focuses on the task of understanding and building Secondary Frequency Control (SFC) model and PI control algorithm so that we are able to test our cases in PART II.

In Chapter 2, we present an overview of Frequency Control, including Primary Frequency Control (PFC), Secondary Frequency Control (SFC) and Tertiary Frequency Control (TFC).

In Chapter 3, we formally focus on the physical theory behind Secondary Frequency Control (SFC) which is the base of the whole project. We also briefly discuss PID control and argue that we should use PI control to reduce the probability of risk. We then discuss how to build a communication layer on top of an existing smart grid simulator and how to design a centralised controller to stabilise the system. We also describe the algorithms we built, the tuning methodology, and some implementations. Finally, we define the acceptable results based on the official report from Nordic, and with that, we build our own analytical tools. With these tools, we can plot a 2D and even a 3D diagram, find the eligible results and give feedback to our controller in PART II.

PART II views testing in a specific test case scenario (Nordic) as an important part such as the impact of different time delays. Detailedly,

In Chapter 4, we focus on testing the system in a low time delay and discuss how to choose an appropriate generator as a breaker and how to tune the range of gain. Before simulating, we predict some expected results based on the physical theory behind Secondary Frequency Control (SFC). We then present a comprehensive evaluation on the simulation results through detailed description and comparison. We also discuss why my prediction had deviation or missing. Then, we show a 2d plot and a simulation result. Lastly, we add the element of ramp rate to make sure a scientifically correct.

In Chapter 5, we discuss the impact of different time delays. We increase the delay and use the range of kp and ki in Chapter 4. We also explain why we think it is reasonable to continue using the range of gain in Chapter 4. Then we predict the simulation results based on both the physical theory behind Secondary Frequency Control (SFC) and the results in Chapter 4. Then we present a comprehensive evaluation on the complicated simulation results. We describe the results with a plotted 3d graph. We discuss some unpredictable results and some seemingly irregular data. We discuss the risk of the generators, like what we do in Chapter 4, and how to remove unacceptable results. We also show a 3d plot and a best simulation result. Finally, we add the element of ramp rate to make sure a scientifically correct.

In Chapter 6, we test the Emergency Control and discuss the impact of different time delays. We firstly define Emergency Control. We predict some results based on Secondary Frequency Control (SFC) and the conclusions in Chapter 4 and Chapter 5. After implementing and analysing the results, we discuss why my predictions are "wrong". Finally, we add the element of ramp rate to make sure a scientifically correct.

We finally conclude in Chapter 7.

1.3 Contributions

The contributions of this thesis are summarised as follows:

- I built a communication layer on top of an existing Smart Grid simulator and designed a centralised controller to stabilise the system.
- I researched situations that cause disturbances, including generators interruption, high time delays and blackout. I built a series of efficient models for analysing the performance of the system and remove any unacceptable results. After hundreds of manual data proofreading, these models have demonstrated superior analytical ability.
- I built an optimal controller to stabilise the frequency based on the centralised controller and the analytical models above. This optimal controller has demonstrated superior performance of restoring frequency when disturbances occur in the system.

Part I

Algorithm

Chapter 2

An Overview of Frequency Control

Every country has its nominal frequency of the oscillations of alternating current in an electric power grid transmitted from a power station to the end-user. For instance, the nominal value is 50Hz [6] in the UK while it's 60Hz [7] in USA.

However, [8] if a load is suddenly connected or disconnected to the system, or if the protection equipment suddenly disconnects a generating, there be a distortion in the power balance between that delivered by the turbines and that consumed by the loads. This imbalance is initially covered from [8] the kinetic energy of rotating rotors of turbines, generators and motors and, as a result, the frequency in the system change.

If there is a mismatch between the generation and the demand, for instance, due to the outage of one generating unit, then the frequency starts to drop down.

If no control is applied, the frequency will largely deviates and then reaches a meagre and steady-state value, due to which the electrical grid is shut down.

The mechanism of primary frequency control is to restore the active power balance in a power system. After primary control takes place, the power balance is restored at a lower or higher frequency. Normally it takes seconds and responses from 5 seconds to 30 seconds. It is a partly Automatic Generation Control.

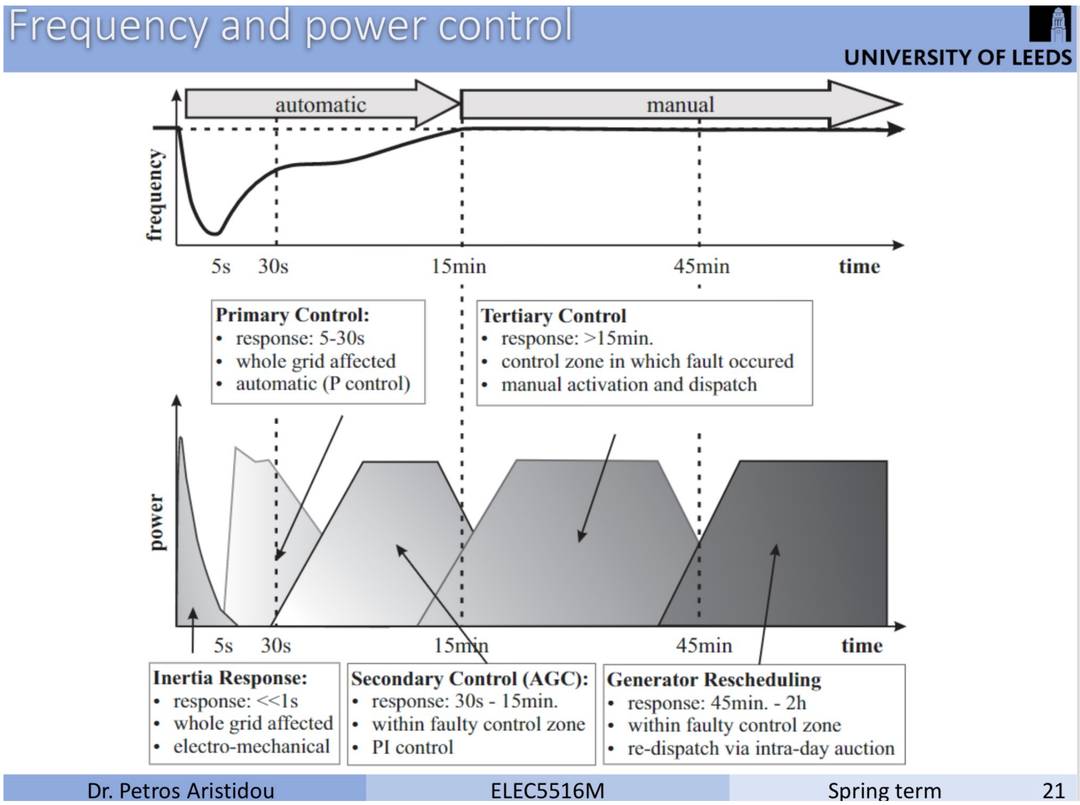


Figure 2.1: Frequency and power control.

However, the frequency does not go back to its nominal value and remains at a steady-state value below or above the nominal one. To avoid damages to equipment and loads, we need secondary frequency control to [8] restore the frequency balance to its nominal value or to eliminate the steady-state error/frequency error. Normally it takes minutes and responses from 30 seconds to 15 minutes. It is a fully Automatic Generation Control.

Tertiary Frequency Control encompasses actions taken to capture current and future emergencies by getting resources. Alternate deployment and recovery after a disturbance are common types of Tertiary Control. Normally it takes dozens of minutes and responses longer than 15 minutes. It is a fully manual control.

Chapter 3

PID Control Algorithm

3.1 The Physical Theory behind Secondary Frequency Control

From Chapter 2, we mentioned that disturbances occur in the system, there be a distortion in the power balance between that delivered by the turbines and that consumed by the loads. The imbalance is from the kinetic energy of rotating rotors of turbines, generators and motors and, thus, the frequency in the system change. If no control is applied, the frequency largely deviates and then reaches a meagre and steady-state value, due to which the electrical grid is shut down.

We also mentioned that the system will start Primary Frequency Control firstly and then the frequency will remain at a steady-state value below or above the nominal one. After that, Secondary Frequency Control starts.

The mechanism of Secondary Frequency Control is to restore the frequency to the nominal one.

Secondary Frequency Control (SFC), or Load Frequency Control (LFC), or Automatic Generation Control (AGC), is an automatic control that [8] restores the frequency back to

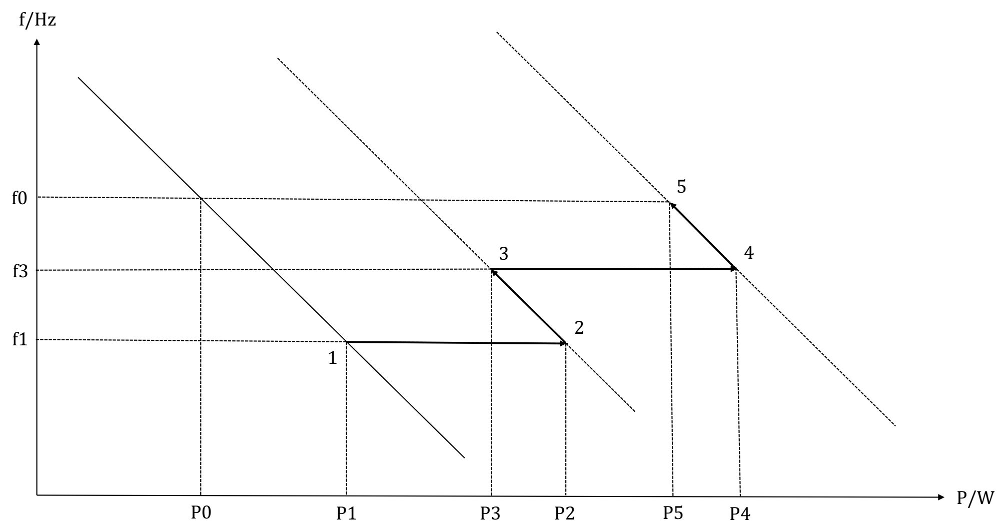


Figure 3.1: Equilibrium points for an increase in the power demand.

its nominal value in a centralised way. It's implemented that is activated after the primary frequency control. Typically, it takes 30 seconds to 15 minutes.

According to Figure 3.1, frequency value rises when reference power rises. Assumed point 1 is the situation after primary frequency control happens and point 5 has the nominal value of frequency. When trying to raise the reference power a little bit, point 1 shift to point 2. Due to the power rise, the frequency of the system rises, so point 2 will move to point 3. Changing more reference power of individual governors will move the overall generation characteristic of the system upwards. Eventually, this leads to the restoration of the rated frequency ($f0$).

As seen in Figure 3.2, the frequency (f) is measured in the local network and is compared with the reference frequency to produce an amplified signal (ΔP_f) that is proportional to the frequency deviation (Δf). For instance, if the frequency is smaller the reference frequency, then the signal ΔP_f will be negative. Thus, the input signal ACE is positive according to the functional diagram. Therefore, the output signal ΔP_f is positive and it adjusts the system by raising the reference value of the power. Then the system has

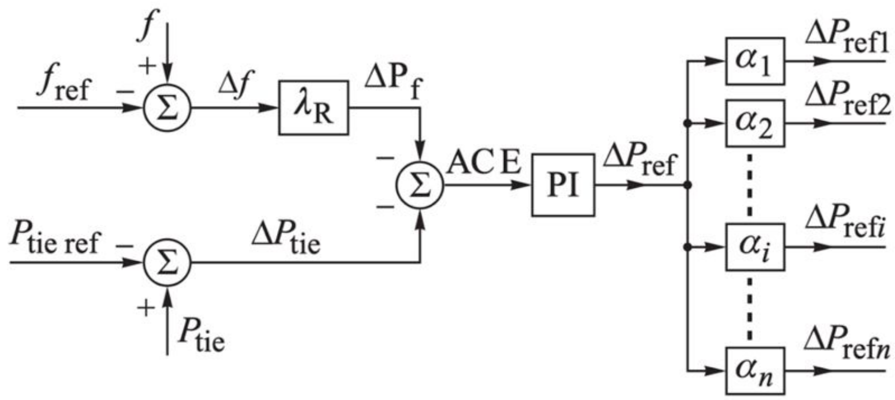


Figure 3.2: Functional diagram of a central regulator.

a new frequency value (f_{new}) that go through the functional diagram again to compare the difference with the value of the reference frequency. ACE won't be zero and the frequency won't be stopped adding until we remove any error.

In this standard case, which ignores the existence of tie-line interchange error, the only condition to remove errors is when the frequency deviation (Δf) equals to zero.

However, it would be more difficult if considering the situation of tie-line interchange error. In interconnected power systems, AGC is implemented in a way where each subsystem has its own regulator. As shown in Figure 3.3, the power system is in equilibrium if the total power generation (P_T), the total power demand (P_L) and the net tie-line interchange power (P_{tie}) satisfy the condition in each subsystem:

$$P_T - (P_L + P_{tie}) = 0 \tag{3.1}$$

The objective of each regulator of the subsystem is to [8] maintain frequency at the nominal level and to maintain net tie-line interchanges from the given area at the scheduled values. If there is a disturbance in one subsystem, then regulators in each subsystem should try to restore the frequency and net tie-line interchanges. Each subsystem regulator

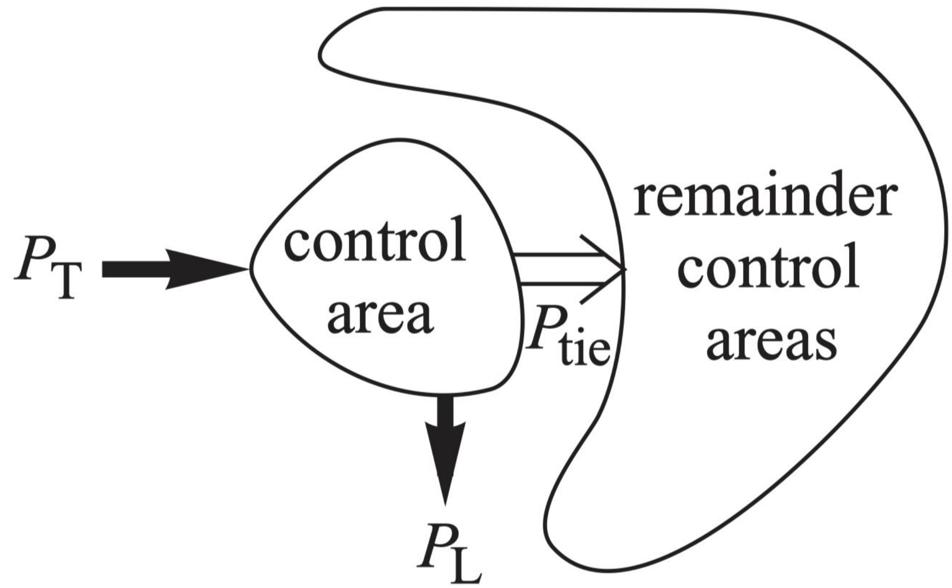


Figure 3.3: Power balance of a control area.

should enforce an increased generation covering its own area power imbalance and maintain planned net tie-line interchanges.

As shown in Figure 3.2, to obtain a signal proportional to the tie-line interchange error (ΔP_{tie}), the information on power flows in the tie-lines is sent via telecommunication lines to the central regulator which compares it with the reference value. Then the signal (ΔP_f) is added to the net tie-line interchange error (ΔP_{tie}) so that ACE is

$$ACE = -\Delta P_f - \Delta P_{tie} \tag{3.2}$$

The situation here is similar to the situation above, where we ignore tie-line interchange error, except for the condition to remove errors. In this book [8], it shows us that zeroing of errors can be achieved in two ways: zeroing of both errors ($P_{tie} = 0$ and $f = 0$) and achieving a compromise between the errors ($\Delta P_f + P_{tie} = 0$ or $P_f = -\Delta P_{tie}$).

3.2 PID Controller

PID controller can be written in the following equation:

$$u(t) = k_p e(t) + k_i \int_0^t e(\tau) d\tau + k_d \frac{de}{dt} \tag{3.3}$$

where u is the control signal and e is the error. The nominal value is also called the reference value or the setpoint. The controller's output is therefore summed by three terms: the P-term, the I-term, and the D-term. The P-term is proportional to the error and its amplification factor is k_p. The I-term is proportional to the integral of the error and its amplification factor is k_i. The D-term is proportional to the derivative of the error and its amplification factor is k_d.

The function of the P-term is trying to send a control signal proportional to the error. For example, there is a signal whose value is 90 and we hope it can approach to 100 using P-term only. The error now is (100 - 90 = 10). Assumed that k_p equals to 0.5, then control signal becomes (10 x 0.5 = 5). The new signal becomes (90 + 5 = 95). If we continue to repeat this flow, we find the new signal turns to 97.5, 98.75 and 99.375. It seems like the new signal is approaching to our nominal value and it approaches to the nominal value if we continue updating the signal.

However, there are two situations that we cannot ignore. Firstly, steady-state error occurs if k_p is not set well. For example, if k_p equals to 2, the error be 20 and the new signal be 110, 90, 110, 90,... The signal can not be restored to the nominal one forever.

The second situation is that, in reality, the signal is not perfect and it has its own errors. In the first example above, we choose 0.5 as our amplification factor (k_p) and thus the first new signal be 95. However, it is a possible the new signal will be lost its signal by 5 units every time after the control. Finally, the signal will remain 90 and the error will be 10.

In reality, whatever it is the automobile control system, the electronic compass control

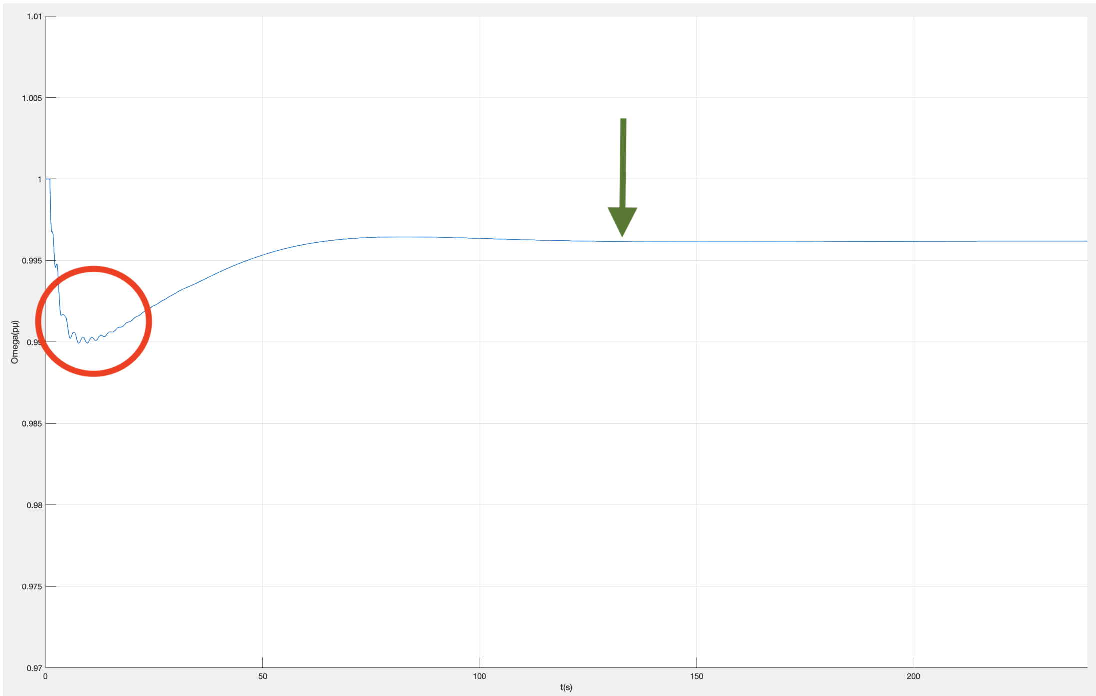


Figure 3.4: Steady-state error and high frequency component in a signal.

system or the Automatic Generation Control, the steady-state errors occur if we use P control only.

We introduce the I-term to remove such a steady-state error. Briefly, the integral of the error accumulate the previous errors and the I-term is proportional to the integral of the error, which means the I-term is proportional to previous accumulated errors. The integral of the error will not be zero, unless the error is removed. Gradually, the controlled signal will be larger and the new signal will approach to the nominal value.

To the D-term, it is proportional to the derivative of the error. In another way, the D-term is proportional to the gradient of the signal. Thus, the D-term amplifies the high frequency component in the signal and will disturb the system.

3.3 PI Control Model

At this point, we are equipped with all the building blocks. Let's first recap a PI control model. In PI control, we need to know the value of error and the value of integral of the error. Error is defined as the difference between the nominal value and the signal. In AGC, the nominal value is the nominal frequency in the country (50 Hz or 60 Hz) and the signal is the actual frequency sending into the controller.

To the integral of the error, it should be realised that it is helpful to use discrete mathematics. In discrete mathematics, the value of integral of the error is the accumulated error. Thus, in Python, we can use time-step to help calculating the integral of the error. Detailedly, in program, we can represent output as

```python
error = nominal_frequency - actual_frequency
errSum += error * agcTimeStep
output = float(kp) * float(error) + float(ki) * float(errSum)
```

Figure 3.5: Python: PI control algorithm.

where nominal frequency is 1.0 and we can directly get actual frequency from the simulation.

Next, we should send output signal to generators through the command,

```python
command = 'CHGPRM TOR ' + gensName + ' Tm0 ' + str(output*gensWeight) + ' 0'
```

Figure 3.6: Python: send corrections to the generators.

where, 'gensName' is the name of generator such as g1, g2 or g9. '1/gensWeight' is α in the Figure 3.2. We use α to ask power proportional to these generators' nominal power.

Furthermore, we need to consider deadband control. The deadband refers to the range of input signal when output signal is zero in the domain of transfer signal. In our case, as shown in Figure 3.7, it refers to the range of time when frequency does not change.

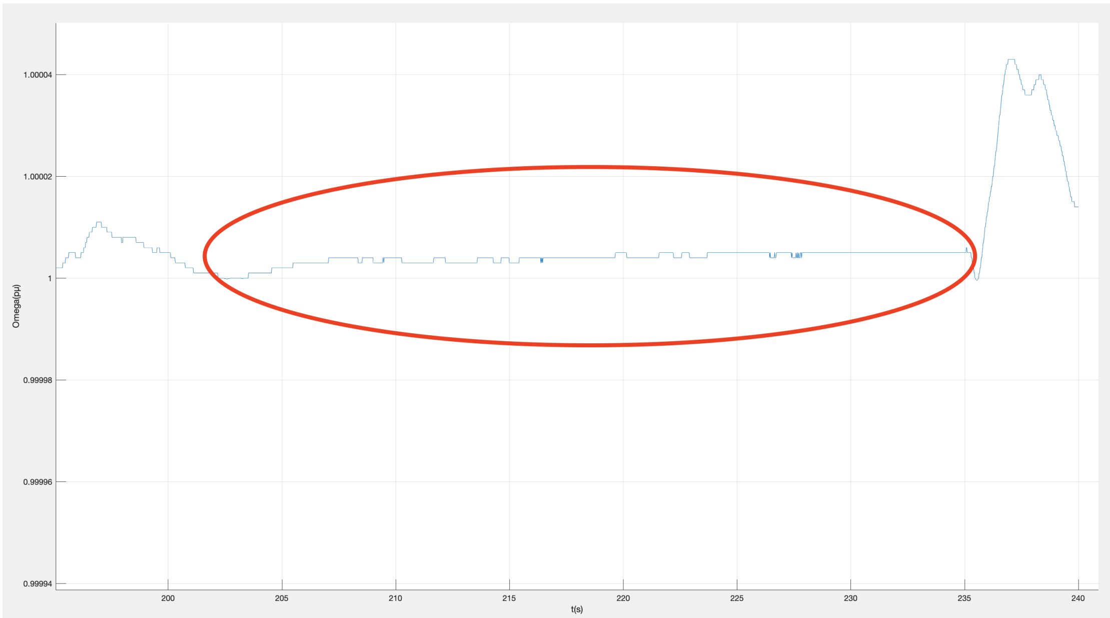


Figure 3.7: Deadband in SFC.

However, in theory, it is impossible to remove deadband totally. The aim to use deadband control is:

(1). Make the frequency value as small as possible in the deadband area, so we can assume that tiny error is no error;

(2). Make the deadband area as long as possible, so we can keep the frequency in the system stable in a relatively long time.

Thus, we need to choose two parameters in our deadband control: acceptable frequency range in deadband area and control results.

The acceptable frequency range is a suitable tiny frequency range that we allow deadband occurs.

For control results, we have three situations. One of them is without deadband control. Another situation is to make error be zero in the deadband area. The last situation is to make the error and the sum of error be zero at the same time.

```python
if abs(error)<0.00001:
    error = 0.0
    errSum = 0.0
```

Figure 3.8: Deadband controller.

Finally, as shown in the Figure 3.8, we choose the frequency range in the deadband from -0.000001 Hz to 0.000001 Hz and we make the error and the sum of error be zero at the same time after the deadband control starts. Figure 3.9 shows the comparison between one without deadband control and one with deadband control. It shows that our deadband control is a more suitable solution for the system.

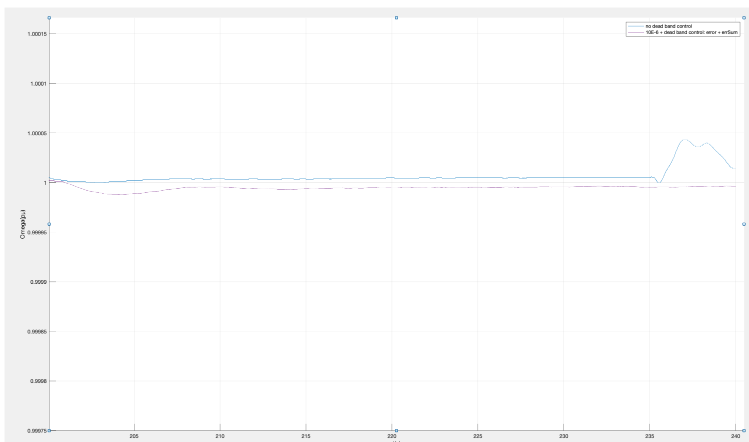


Figure 3.9: Comparison: with and without deadband control.

We get the input from the existing smart grid simulator and send output to the existing smart grid simulator via our controller. Thus, a communication layer is formed.

Multiple hard-coded parameters, like the disconnected generator, the generators we asked for their power, amplification factor and time delay, are removed and added into main function so we can test our system comprehensively and easily.

3.4 Tuning Methodology

3.4.1 Tuning Model

Imagine if we have a 2D plane with the horizontal axis kp and the vertical axis ki, we can use the points on the plane to represent all the acceptable combination of kp and ki.

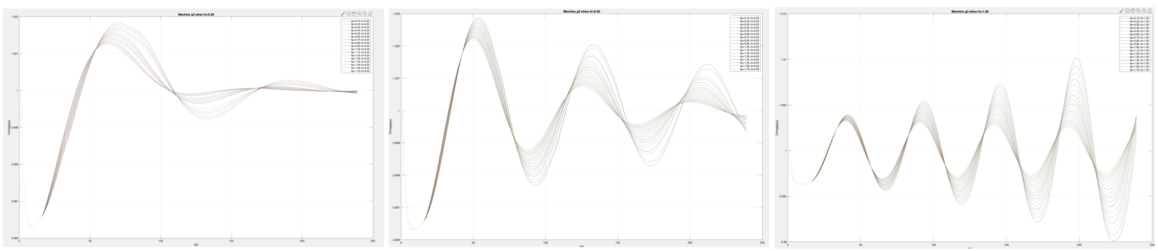


Figure 3.10: PI Control: ki = 0.2; ki = 0.5; ki = 1.3.

The reason for this is that neither kp nor ki can be infinitely magnified. Kp is the amplifier factor of D-term and it represents how quickly an error can be corrected. If kp is larger, the overshoot of the corrected signal will be larger. Kp has a limit because there is an acceptable range of frequency with official document. For instance, in Nordic test case scenario, the limit is $\pm0.2\%$. In other words, the range of signal is from 0.998 $p\mu$ to $1.002p\mu$.

Ki is the amplifier factor of the I-term. The signal has too many oscillations and it would not be settled if we keep increasing the value of ki.

However, the signal needs to be settled before a specific time in reality and it makes sure the signal has no chance of excessive oscillation. Thus, ki is limited.

One of the objectives of this project is to find the best combination of amplifier factors, i.e. kp and ki. To obtain it, we need to find the limit of kp and ki and simulate possible situations in the range.

The idea of bisection method is used to find the limit of kp and ki.

Firstly, we tune a large kp, and we test if a new signal is acceptable. To simplify the model, we choose ki being from 0.1 to the value of kp and the step of ki is dependent to the value of kp. For instance, if kp is 500.1, the range of ki is between 0.1 and 500.1. From the part we discussed before, the signal has great chance to have a large overshoot. Thus, ki can be given a large step, like 100.0. We do not need to worry about such a large step will filter some acceptable results. The purpose of this step is not to find all the acceptable results. The purpose of tuning kp and ki is not finding all the possible values. We hope to find a general relationship between kp and ki and it will be helpful to find the impact of time delay afterwards.

Secondly, we check if the signal is acceptable if we have simulated data. The purpose of this step is to find the limit of ki.

If ki exists (i.e. there is one acceptable result at least), for instance, ki equals to 200.1, we need to make sure there are no solutions when increasing ki by its step. Then, we can judge that the limit of the kp is between (200.1 and 200.1 + the step of kp). Thus, we have a range of kp which is between (0.1 and 200.1 + the step of kp). If the step of kp is 10, the range of kp will be from 0.1 to 210.1. Normally, we can directly tune kp and ki between 0.1 and 210.1.

If ki exists in another way, for instance, kp equals to 200.1 and ki equals to 5.1 when the step of ki is 5.0. The result is acceptable but we need to increase the value of kp to make sure ki does not exist when kp equals to (kp + the step of kp).

If ki does not exist, for instance, there is no acceptable ki when kp is 500.1. According to bisection method, we can test the simulations and tune kp between 0.1 and 250.1. We repeat the method above until we find the limit of kp and ki.

3.4.2 Analytical Models

The first analytical model generates two different kinds of excel files. One of them is named as ori.xlsx. This is a file saving all the acceptable results. For another kind of files, their names are related to the value of time delay. For instance, if time delay is 0.01 seconds, the file name be td_0.01s_xlsx. It stores all the acceptable results when delay is 0.01 seconds.

Importantly, we define 'SettlingTimeThreshold', a MATLAB variant, to 0.02, so the time taken for the error between the response and the steady-state response will fall within 2% of nominal value.

```matlab
info = stepinfo(f,t,1.0,'SettlingTimeThreshold',0.02);
```

Figure 3.11: MATLAB: stepinfo function.

In our case, the system starts from the zeroth second. However, in the analytical model, we can not put all the data into the stepinfo function because the data should be analysed from the beginning of SFC. Thus, the data stepinfo function gets should starts from the SFC control starting, which is the 150th seconds.

```matlab
startingTime = 150.0;

% find the index so it is the 150th sec
for index = 1:length(t)
    if t(index)>=startingTime
        break
    end
end

% shift time-axis
tchopped = t(index:end,1) - startingTime;
fchopped = f(index:end,1);

% set steady-state value (y_final) to nominal value & SettlingTimeThreshold to 2%:
info = stepinfo(fchopped,tchopped,1.0,'SettlingTimeThreshold',0.02);
settlingTime = info.SettlingTime;
```

Figure 3.12: MATLAB: chopped the time from the 150th sec.

In the first analytical model named i_cur_plot.cur , we check if simulation results are acceptable via our MATLAB program if we have collected data. Firstly, we import data from simulations. Then, we limit both the overshoot and settling time as required via a MATLAB module: stepinfo. The data will be filtered if the overshoot or the settling time is not in the required range.

```matlab
if info.Overshoot <= nordic_limit && settlingTime < required_settlingTime
    txt = ['kp = ', num2str(kp,'%.2f'), ', ki = ', num2str(ki,'%.2f'), ...
           ', Delay = ', num2str(delay,'%.2f'), ...
            ' sec, Settling Time = ', num2str(settlingTime+startingTime,'%.4f'), ' sec'];
    plot(t, f, 'DisplayName',txt, 'LineWidth',1)
    KP = [KP; kp];
    KI = [KI; ki];
    DELAY = [DELAY; delay];
    SETTLINGTIME = [SETTLINGTIME; settlingTime+startingTime];
```

Figure 3.13: MATLAB: filter unacceptable tuning results.

As you can see from Figure 3.13, if the signal is acceptable, its parameters, i.e. kp, ki and time delay, will be saved in prepared lists. The lists will be transferred into the two kinds of excel files.

The second analytical model helps finding which combination of kp and ki has a minimum settling time with a fixed delay. The idea is as follows. Firstly, we import an excel file, whose name is related to the value of delay, that is generated by the last step. Then, we find the minimum settling time. Finally, we find related kp, ki and delay and export them into another excel file.

The third analytical model helps finding the best combination of kp and ki. Firstly, we import data from ori.csv. Then, we calculate the average settling time for each combination of kp and ki. However, it is invalid that a combination is not acceptable for all the time delay. Thus, we need to add a limitation factor to make sure the best combination is valid for every delay. For instance, if there are 21 different time delays, we can use

```python
if i != 21: # have 21 kinds of delays
    settlingTime = 999999
```

Figure 3.14: Python: filter out the tuning results that cannot settled in all delay

to filter some unacceptable results. Finally, we have all the average settling time so we can find the minimum average settling time.

The fourth analytical model helps sorting out the combination's settling time with different time delay. It is one of the inputs of the fifth analytical model.

The fifth analytical model helps plotting a 3D triangle surface diagram. It draws a 3d plot from the data in ori.xlsx. We use:

```
K = boundary(x, y, z, 1);
trisurf(K, x, y, z, 'FaceAlpha', 0.1)
```

Figure 3.15: MATLAB: 3D triangle surface algorithm.

to draw a triangle surface plot. Besides, it highlights the results from the second analytical model and the fourth analytical model. With a 3D triangle surface plot, we can clearly understand the impacts of the time delay.

3.5 Rate of Generators Change

Generators have a limit on the output power (in MW) and another limit on the rate of change of power (in MW/minute). The rate of change of power is called ramp rate. It expresses how quickly a power plant's power output is changing.

$$Rate = \frac{\Delta P}{\Delta t} \tag{3.4}$$

According to National Renewable Energy Lab., Golden, CO (US), [9] large thermal units are usually able to ramp around 1% of their capacity per minute. For instance, if a generator's nominal power is 360 MW, its ramp rate will be 3.6 MW/min or 0.06 MW/sec.

In this project, benefiting from PYRAMSES [1], it's possible to get generators' power output directly from the simulator. Thus, through the MATLAB STEPINFO [2] module, we

[1] PYRAMSES is a Python library for RAMSES dynamic simulator: https://anaconda.org/apetros/pyramses

[2] STEPINFO contains RISE TIME, SETTLING TIME, and other step-response characteristics. Document: https://www.mathworks.com/help/control/ref/stepinfo.html

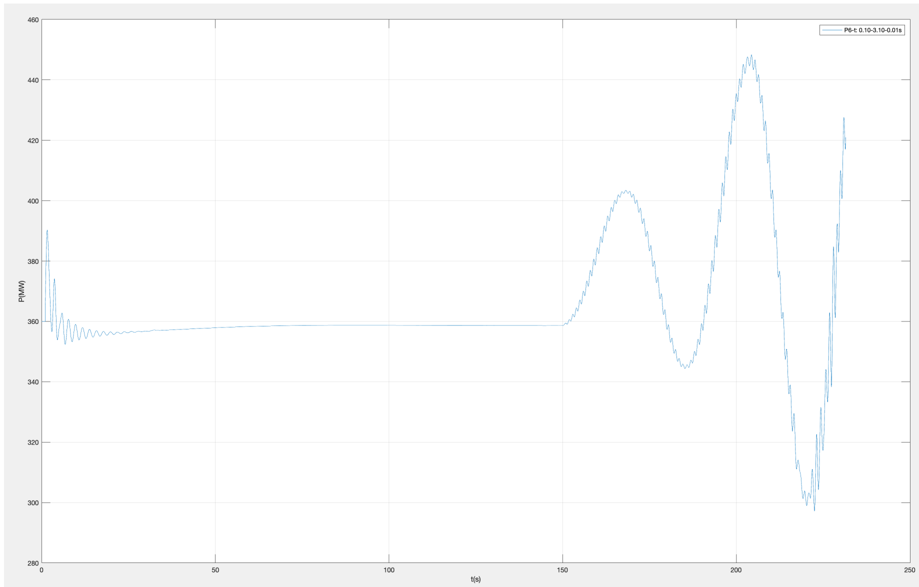


Figure 3.16: g6's power output: kp=0.1, ki=3.1

can use PEAK and PEAKTIME to calculate the ramp rate:

$$Rate = \frac{P_{\text{PEAK}} - P_o}{\text{PEAKTIME}} \tag{3.5}$$

where P_o is the original power output of a generator when SFC starts (i.e. $P_{t=0}$ when the SFC starts).

I have to explain why I don't use RISETIME to calculate the ramp rate. It seems like RISETIME ensures the rate's reliability from its definition: "Time it takes for the response to rise from 10% to 90% of the steady-state response". However, unlike frequency, generators do not have steady-state response (y_{final}). Traditionally, we can use the final value of the signal as steady-state response (y_{final}). However, in this case, as shown in Figure 3.16 and in Figure 3.17, different tuning results may bring different y_{final}. It makes using y_{final} as steady-state response nearly impossible. Choosing PEAK and PEAKTIME that are not related with steady-state response is necessary.

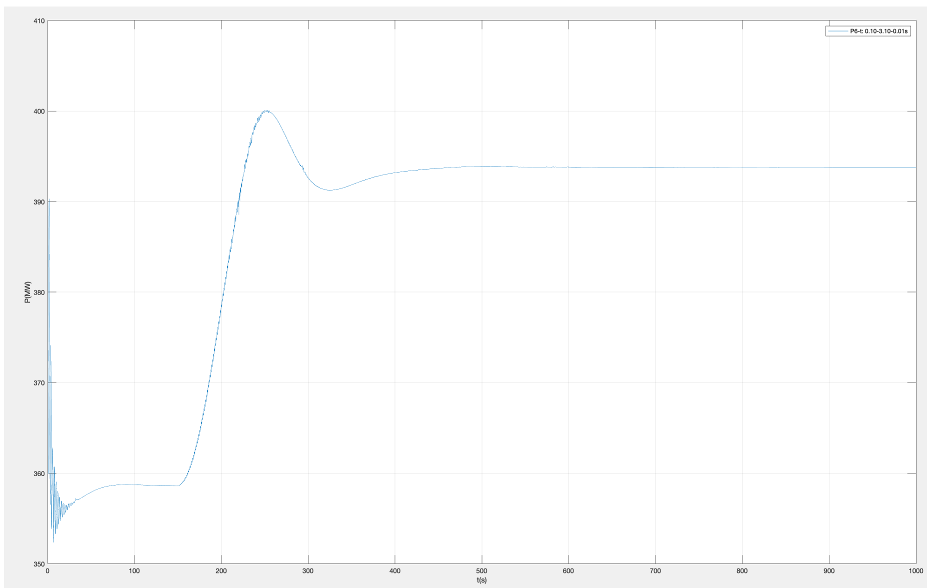


Figure 3.17: g6's power output: kp=0.1, ki=0.1

Part II

Test Case Scenario (Nordic)

Chapter 4

Low Time Delay

4.1 Hypothesis

4.1.1 Hardware Hypothesis

In this Chapter, we use the algorithms described in the Chapter 3 to test the well-known Nordic system. One-line diagram of the Nordic test system is shown in Figure 4.1.

Firstly, we assume g2 as our monitor. The monitor sends its frequency, also the frequency in the system, to the communication layer. Then, we choose five thermal generators (g6, g7, g14, g15, g16) in the CENTRAL area. There are three reasons I choose these five generators that produce the power to the system when disturbances occur in the system. First of all, these are thermal generators not the generators in the NORTH area which are equipped with hydraulic turbines. We only have strict mathematical proof of thermal generators. Another reason is they have nice nominal power as shown the Table 4.1 below. The second reason is the difference of them is up to 900 MW which is good for researching the function of generators comprehensively. The last reason is that I collaborated with a PhD student. We researched the NORDIC TEST SYSTEM with different algorithms. He used distributed method and I used centralised method. For convenience, we used the same generators.

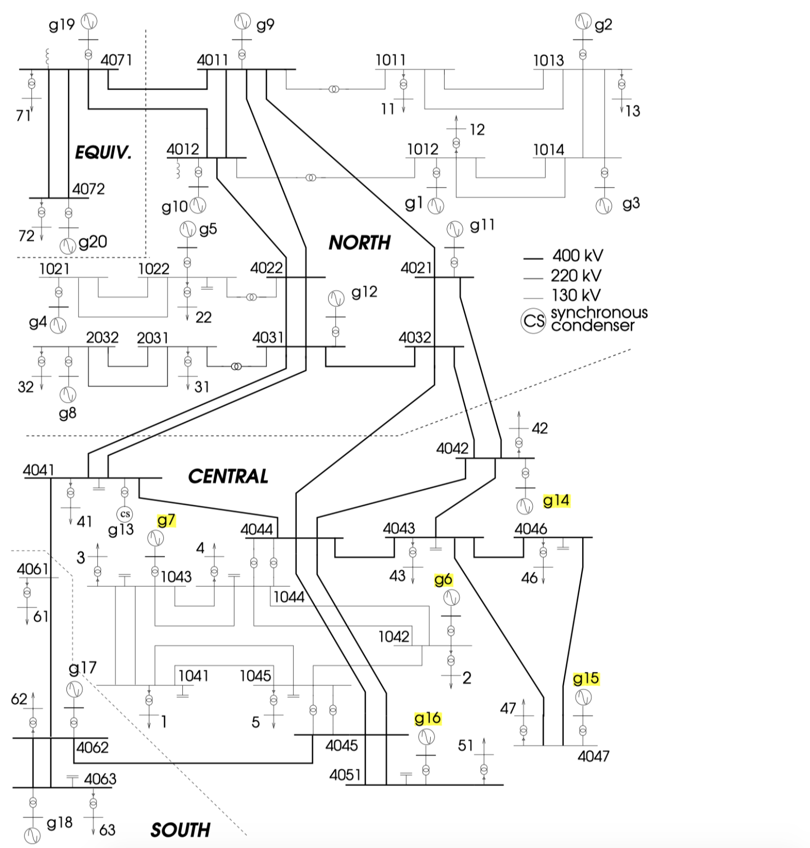


Figure 4.1: One-line diagram of the test system

Generator	Nominal Power (MW)
g1	760.0
g2	570.0
g3	665.0
g4	570.0
g5	237.5
g6	360.0
g7	180.0
g8	807.5
g9	950.0
g10	760.0
g11	285.0
g12	332.5
g13	0.0
g14	630.0
g15	1080.0
g16	630.0
g17	540.0
g18	1080.0
g19	475.0
g20	4275.0

Table 4.1: Generators Nominal Power in Nordic.

After choosing the generators, we need to select a breaker. The breaker is a generator that can be disconnected to the system. Thus, a breaker is one of the most important impact factors to the grid system. However, at first, I was not sure how to choose a suitable breaker but then I tried to disconnect all of them one by one without Secondary Frequency Control. Results are as Figure 4.2.

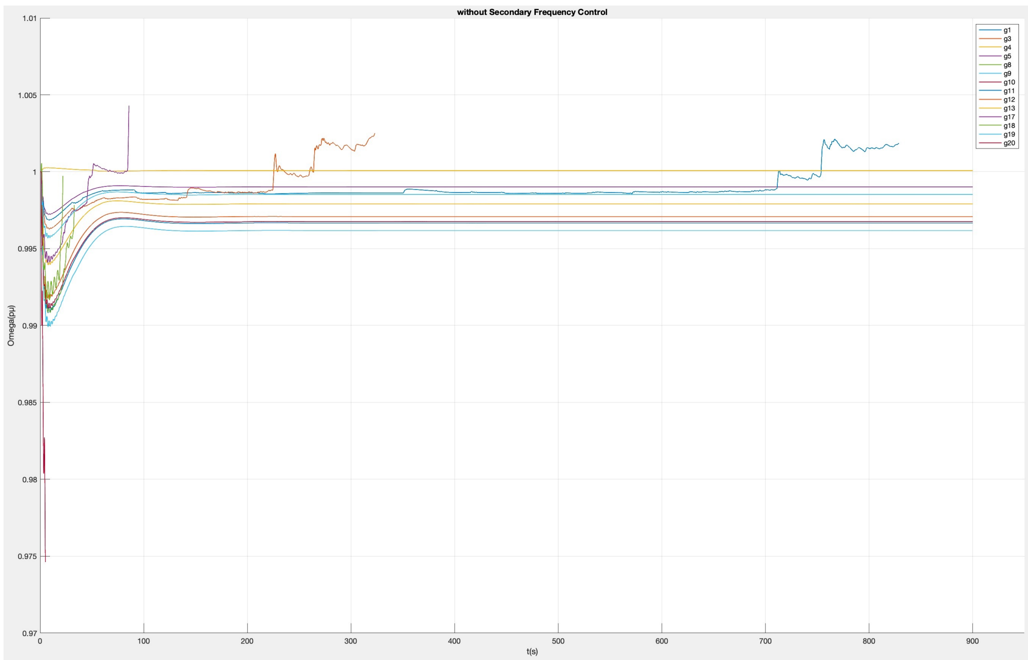


Figure 4.2: Simulation results with different breakers, without SFC.

As we can see from Figure 4.2, g8, g17, g18 and g20 are extremely unstable before 150 seconds and g13 have a zero frequency (because it has a zero nominal power, (Table 4.1)). Thus, we choose a breaker out of them to make sure we have a smooth testing environment.

Finally, we chose g9 because it has a large steady-state error, as shown in Figure 4.3. Thus, the turbines will send more power to the system. It is easier to see the changes of kp and ki and to plot a 2D diagram of them.

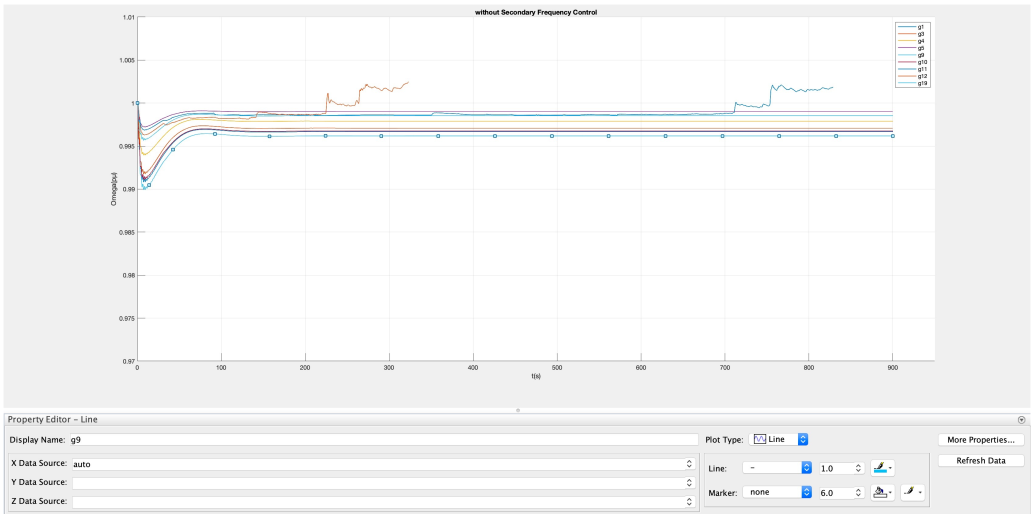


Figure 4.3: MATLAB figure: choose g9 as breaker.

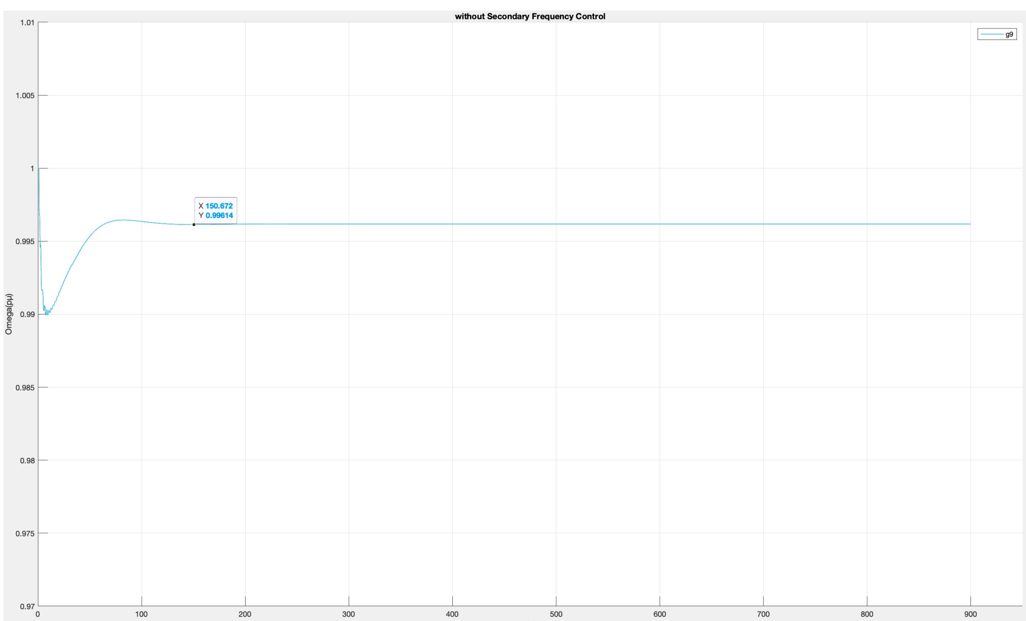


Figure 4.4: MATLAB figure: choose start time for SFC.

4.1.2 Software Hypothesis

Next hypothesis is about start time and end time. In the definition of Primary Frequency Control (PFC), its power balance will be restored at a lower or higher frequency. Thus, from Figure 4.4, we can observe that, at the 150th seconds, the frequency is stable. Thus, I choose the 150th second at the beginning of Secondary Frequency Control. In the definition of Secondary Frequency Control, the response time will take up to 15 minutes, thus, I choose the 900th second as my temporary end time.

Since it is a low time delay testing, we can assume that delay is 0.01 seconds. The reason for not choosing an ideal situation (i.e. delay = 0 sec) is that, in reality, there be no such a zero-delay scenario.

Next step, we start tuning kp and ki. Following the idea in Section 3.4, firstly, we assumed a large value as a limit value of kp. Detailedly, we assumed the range of kp and ki are both from 0.1 to 300.1 and their steps are both 50.0.

Then, we used the designed MATLAB program to check whether the simulation results are acceptable. Detailedly, we set overshoot smaller than 0.2% because it is required that the frequency error should be in the range of ± 0.1 Hz from Section 4.1.1 in the Nordic official document.

We also need to make sure the signal is really settled from 0.9998 to 1.0002 before the end time (i.e. the 900th second), thus, it is necessary to extend the simulation time and give the program more data to check whether it's settled before 15 minutes. Thus, we set 1000 sec as our end time.

Finally, the program sketched plots of all the acceptable results and gave related information in the legend bar.

However, as you can see from Figure 4.5, only two simulations are acceptable. The

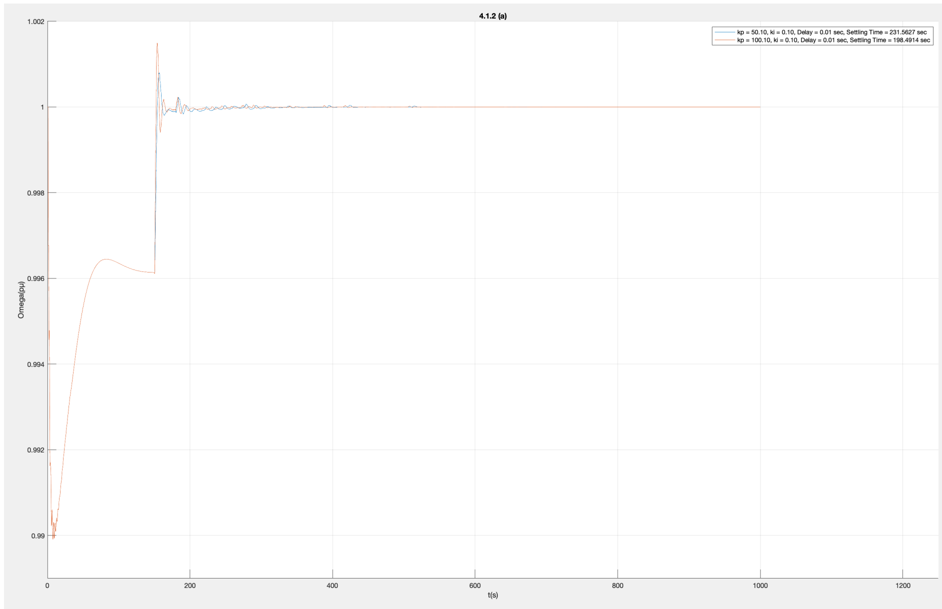


Figure 4.5: Low time delay tuning 1: Disconnect generator g9 when delay is 0.01 sec, kp is from 0.1 to 300.1 (step: 50), ki is from 0.1 to 300.1 (step: 50), start time is the 150th sec, end Time is the 1000th sec, and required settling time is the 900th sec.

maximum value of kp is 100.1 and the maximum value of ki is 0.1.

According to the regulations from the last chapter, we need to use bisection method to increase the number of effective amplifier factor (i.e. kp and ki).

For instance, the new range of kp is between 0.1 and 150.1 and the new range of ki is between 0.1 and 50.1. To make more accurate results, we need to decrease the step at the same time. We set step be 10.0.

Finally, we have another plot as shown in Figure 4.6.

Apparently, we have more acceptable results than the last simulation because we shrink the step of kp. We find that the simulations are unacceptable if kp equals to 140.1 or 150.1. Thus, we can finally fix the range of kp between 0.1 and 140.1 and keep the step of kp to 10. We can set ki in a range of 0.1 and 10.1 with the reason above. However, it is possible

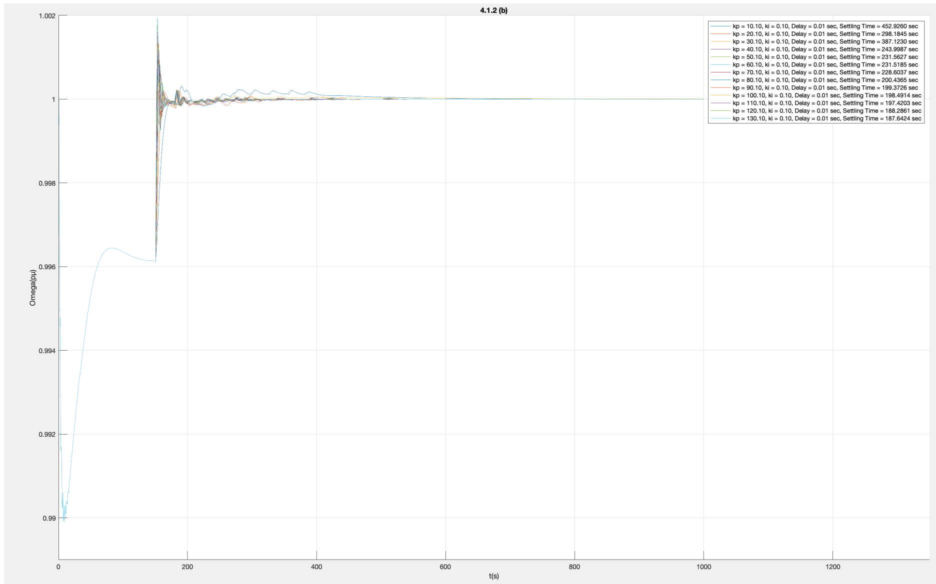


Figure 4.6: Low time delay tuning 2: Disconnect generator g9 when delay is 0.01 sec, kp is from 0.1 to 150.1 (step: 10), ki is from 0.1 to 50.1 (step: 10), start time is the 150th sec, end Time is the 1000th sec, and required settling time is the 900th sec.

that the maximum value of ki is still 0.1 if we set the range of ki from 0.1 and 10.1. If so, then the simulations are meaningless.

Thus, we need to apply bisection method on the range of ki and find the meaningful maximum ki. Detailedly, we range kp from 0.1 to 140.1 and set ki equals to 5.1. The purpose of doing this is to find whether there are acceptable results if ki is in the middle of 0.1 and 10.1.

The filtered results are as Figure 4.7.

The result shows there are acceptable simulations if ki equals to 5.1.

Thus, we can set the range of kp between 0.1 to 140.1 (step: 10.0), the range of ki between 0.1 and 10.1 (step: 1.0). The controller will start from 150 seconds and will end at 1000 seconds. We hope that the signal can be settled before 900 seconds.

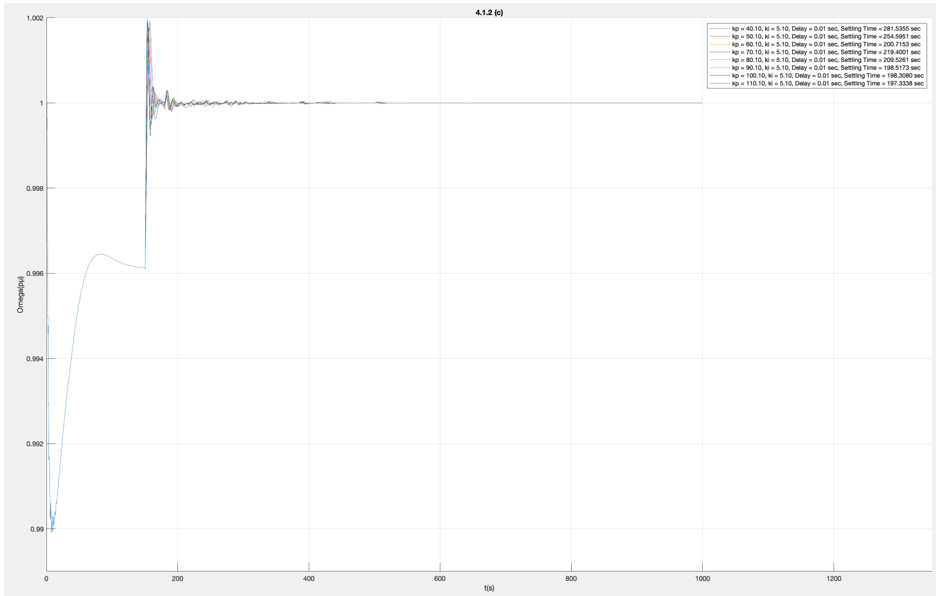


Figure 4.7: Low time delay tuning 3: Disconnect generator g9 when delay is 0.01 sec, kp is from 0.1 to 140.1 (step: 10); ki is 5.1, start time is the 150th sec, end Time is the 1000th sec, and required settling time is the 900th sec.

4.2 Expected Outcome

From discussions before, we know that both kp and ki can not be expanded indefinitely since we have specialised limit for kp and ki in MATLAB program. Thus, firstly, I hope to see a clear borderline that separates the acceptable and unacceptable results. The acceptable results will be expressed as blue points in Excel.

Secondly, I hope to see a relationship between the best point, which has minimum settling time among all the acceptable results, and other acceptable points. I expect the best point will not appear in the borderline because the points in the borderline are likely to be unstable if the delay increases.

I also do not hope the best point has a large or small kp and ki. A larger kp produce a larger overshoot although it would be relatively faster. A small kp has a larger probability of producing a steady-state error and it is relatively harder to approach a settled statement.

A larger ki produces more oscillations, as Figure 3.10 shows, and thus it is also hard to approach a settled statement. A small ki does not help fixing the steady-state error along with a small kp.

4.3 Implement

One of the most important idea to implement the testing environment is to modularise the function (Figure 4.8). We modularised our functions in a python file so we can import them as a library and call them directly (Figure 4.9). Additionally, it is necessary to remove hard-coded parameters so it is easy to change the parameters like start time, end time, prepared folder address and the list of generators (Figure 4.9).

Another noteworthy detail is we need to keep two significant digits (Figure 4.9) for kp, ki and time delay. It provides an unified format that helps exporting data into analytical algorithms.

The script to generate the feasible kp-ki-td parameters is automatic and well documented (Figure 4.9).

```python
# Load library
import numpy as np
import PyRAMSES

# Load models
from examples_models import end_simulation
from examples_models import move_file
from examples_models import sfc

# Load a simulator instance
ram = PyRAMSES.sim()

# Load saved test-case & Add more observation(s)
case = PyRAMSES.cfg('cmd.txt')
case.addRunObs('MS g2')  # will plot a frequency-time diagram by bus g2
```

Figure 4.8: Python: import related libraries.

```python
# Tuning
if __name__ == '__main__':
    ######### parameters #########
    start_time = 150.0
    end_time = 1000.0
    agcTimeStep = 1.0
    monitor = ['g2']
    breaker = 'g9'  # (dst file)
    list_of_gens = ['g6', 'g7', 'g14', 'g15', 'g16']
    weight_of_gens = [1/8,1/16,7/32,3/8,7/32]
    prepared_folder_address = 'D:/OneDrive - University of Leeds/Nordic/4.3'

    #### tuning delay & kp & ki: ####
    flagTd = 'ones'
    for td in np.arange(0.01, 0.02, 0.01):  # td: 0.01s
        list_of_td = td * np.ones(5)
        for kp in np.arange(0.1, 140.2, 10.0):  # kp: 0.1~140.1, step: 10
            for ki in np.arange(0.1, 10.2, 1.0):  # ki: 0.1~10.1, step: 1.0
                kp = "{0:.2f}".format(round(float(kp),2))
                ki = "{0:.2f}".format(round(float(ki),2))
                td = "{0:.2f}".format(round(float(td),2))
                print("kp = " + str(kp))
                print("ki = " + str(ki))
                print("td = " + str(td))

                #### Run sfc: ####
                sfc(ram, case, start_time, end_time, agcTimeStep, monitor, kp, ki, list_of_gens, weight_of_gens,
    list_of_td, prepared_folder_address, breaker, flagTd)
                pass
            pass
```

Figure 4.9: Python: tune PI control.

4.4 Results

4.4.1 Results and Analysis

The implement results are as follows. Figure 4.10 shows all the acceptable results.

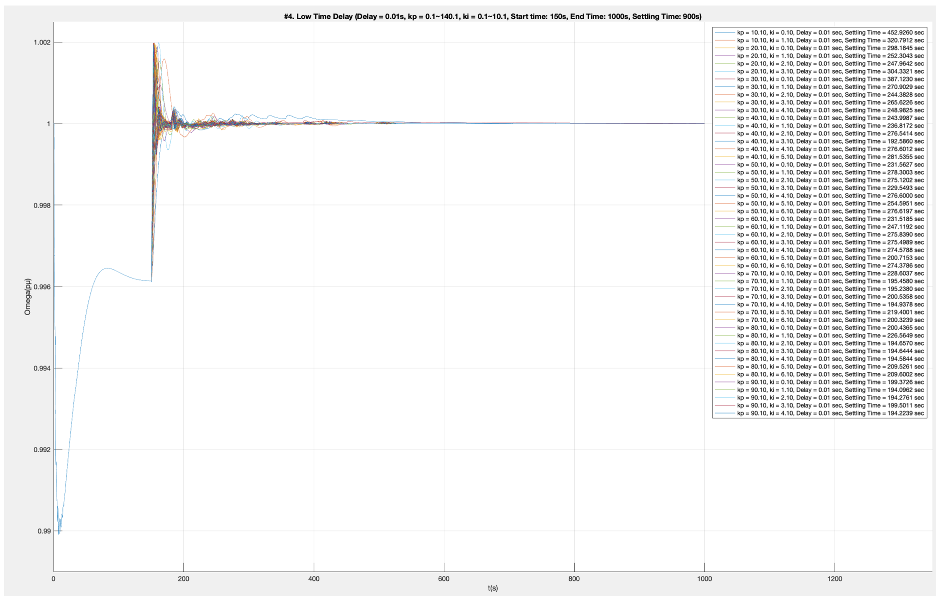


Figure 4.10: Low time delay tuning results.

From Figure 4.10, we can roughly draw a conclusion that the signals are indeed within a reasonable range and it seems that they have finally reached the nominal value.

However, we need to check further by choosing one of the signals. For instance, I choose the signal whose kp is 90.1 and ki is 7.1. The reason I choose this signal is that it is located at the borderline. Results are as follows.

From the Figure 4.11, we can see that the overshoot does not exceed 1.002 obviously. But for the settling time, we can not make any conclusion until we see the details.

From Figure 4.12, we can judge that its overshoot does not exceed the maximum allowable value, i.e. 1.002 $p\mu$, after the SFC starts.

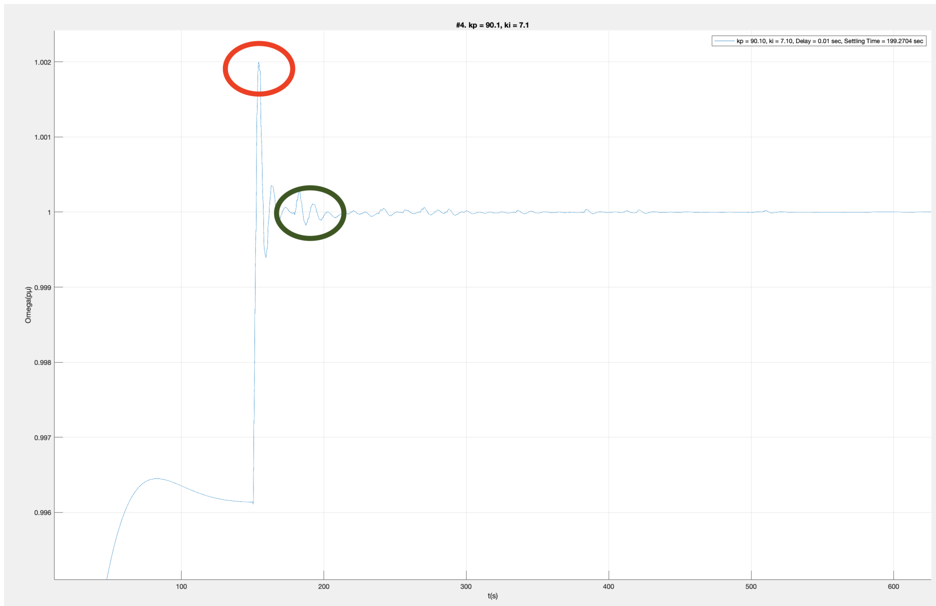


Figure 4.11: A random low time delay tuning result

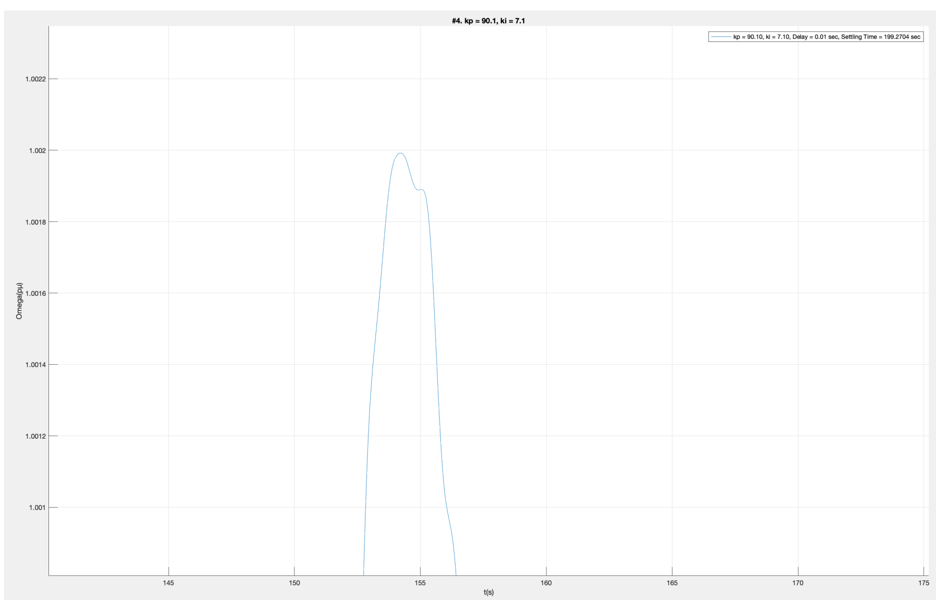


Figure 4.12: Detail 1 for a random low time delay tuning result: frequency limit.

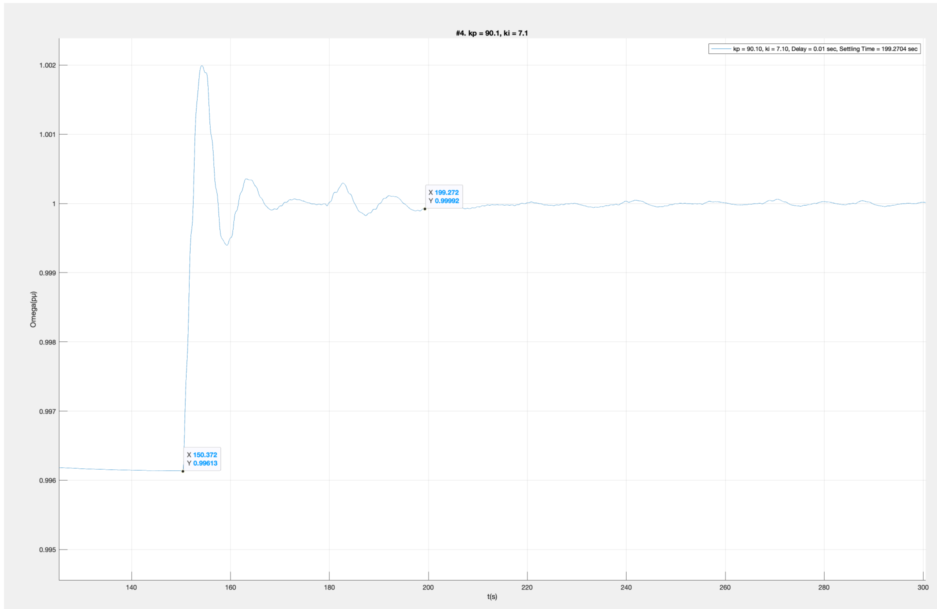


Figure 4.13: Detail 2 for a random low time delay tuning result: settling condition.

According to the definition of settling time from MATLAB,

$$\Delta f_{threshold} = 2\% * (f_{nominal} - f_0) \tag{4.1}$$

where $\Delta f_{threshold}$ is the threshold for the settled results.

From Figure 4.13, we find that, after the signal settled, the signal is between the settling time threshold, i.e. 0.9992 $p\mu$ and 1.0008 $p\mu$.

With the prove above, we can finally say that our assumed acceptable simulation results are acceptable.

Next, we show you the relationship between kp and ki.

Table 4.2 shows the parameters that make up the acceptable signals, i.e. kp, ki, delay and settling time. It has been ranked by the settling time.

KP	KI	DELAY	SETTLINGTIME (sec)	KP	KI	DELAY	SETTLINGTIME (sec)
130.1	0.1	0.01	187.6424	60.1	5.1	0.01	200.71535
120.1	1.1	0.01	187.70885	80.1	5.1	0.01	209.5260889
120.1	0.1	0.01	188.2861333	80.1	6.1	0.01	209.60025
120.1	2.1	0.01	188.304	70.1	5.1	0.01	219.4000571
110.1	1.1	0.01	192.01492	80.1	1.1	0.01	226.5649143
120.1	3.1	0.01	192.15436	70.1	0.1	0.01	228.6036571
110.1	2.1	0.01	192.39484	50.1	3.1	0.01	229.5492571
110.1	3.1	0.01	192.4948	60.1	0.1	0.01	231.51848
40.1	3.1	0.01	192.5859619	50.1	0.1	0.01	231.5627333
100.1	6.1	0.01	193.47516	40.1	1.1	0.01	236.8172
90.1	1.1	0.01	194.09616	40.1	0.1	0.01	243.99872
90.1	4.1	0.01	194.22394	30.1	2.1	0.01	244.38276
90.1	2.1	0.01	194.27608	60.1	1.1	0.01	247.11916
80.1	4.1	0.01	194.5844	20.1	2.1	0.01	247.96416
80.1	3.1	0.01	194.64444	30.1	4.1	0.01	248.98248
80.1	2.1	0.01	194.65696	20.1	1.1	0.01	252.30432
70.1	4.1	0.01	194.93776	50.1	5.1	0.01	254.5950667
70.1	2.1	0.01	195.23796	30.1	3.1	0.01	265.6226
70.1	1.1	0.01	195.45804	30.1	1.1	0.01	270.90288
110.1	5.1	0.01	197.3338	60.1	6.1	0.01	274.3786
110.1	4.1	0.01	197.37752	60.1	4.1	0.01	274.5788
110.1	0.1	0.01	197.4202667	50.1	2.1	0.01	275.1202
100.1	5.1	0.01	198.308	60.1	3.1	0.01	275.49892
100.1	3.1	0.01	198.3430667	60.1	2.1	0.01	275.83904
100.1	4.1	0.01	198.39056	40.1	2.1	0.01	276.54144
100.1	0.1	0.01	198.4914222	50.1	4.1	0.01	276.59996
100.1	1.1	0.01	198.5091556	40.1	4.1	0.01	276.60116
90.1	5.1	0.01	198.5172727	50.1	6.1	0.01	276.61972
100.1	2.1	0.01	198.5241818	50.1	1.1	0.01	278.30032
90.1	6.1	0.01	198.6748615	40.1	5.1	0.01	281.5354667
90.1	7.1	0.01	199.2704	20.1	0.1	0.01	298.18448
90.1	0.1	0.01	199.3726	20.1	3.1	0.01	304.3321333
90.1	3.1	0.01	199.5010909	10.1	1.1	0.01	320.7912
70.1	6.1	0.01	200.3238667	30.1	0.1	0.01	387.123
80.1	0.1	0.01	200.4365143	10.1	0.1	0.01	452.926
70.1	3.1	0.01	200.53584				

Table 4.2: Part of the acceptable results ranked by settling time.

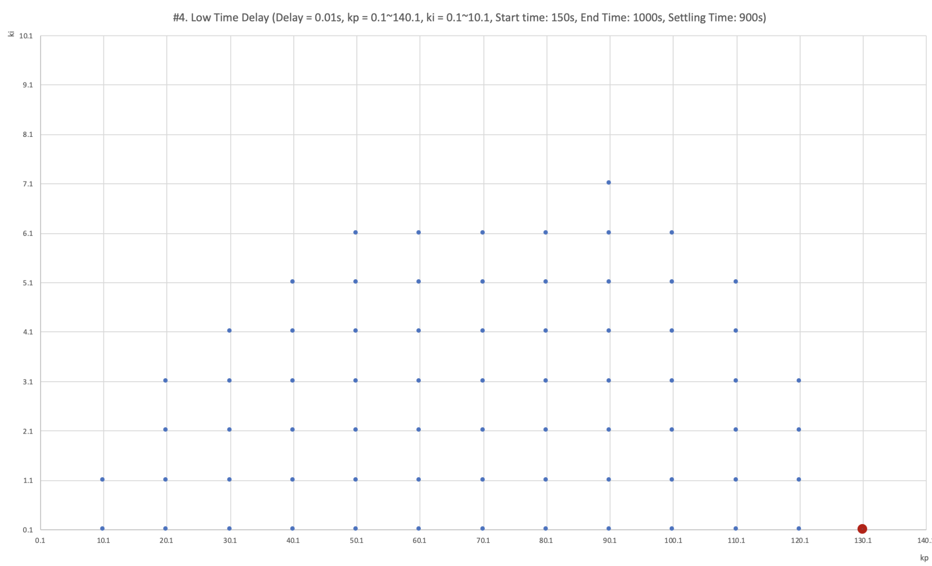


Figure 4.14: 2D plot: All the acceptable tuning results in low time delay.

Table 4.14 draws a 2D graph where kp is horizontal axis and ki is vertical axis. Besides, we highlight the point which gives the minimum settling time as a reference.

Analysing the Table 4.14, we can find:

- The 2D graph shows a clear borderline which separates the acceptable results and unacceptable results. This is in line with the previous expectation from Section 4.2;
- The best point is (130.1, 0.1);
- The best point is in the borderline. This is not in line with the previous expectation from Section 4.2;
- The best point's kp value is too large. This is not in line with the previous expectation from Section 4.2;
- The best point's ki value is not too large. This is in line with the previous expectation from Section 4.2;

- The worst point, i.e. kp is 10.1 and ki is 0.1, appears in the borderline. This is in line with the previous expectation from Section 4.2.

Although, for the best point, its kp is large enough to keep the signal approaching the settling status as fast as possible, it is dangerous that it would be removed by a higher time delay.

However, the best point's ki value is 0.1, which is "unexpected" before. There are two explanations for this small ki value. The first explanation is that this "small" ki value is not small enough or is not the smallest because it is assumed that 0.1 is the minimum value of the range of ki at the beginning of the simulation. We do not need to find the real limit value of ki in this case because it will increase the amount of computation by an order of magnitude. There might be acceptable results whose kp is smaller than 0.1 but it is not necessary to find them in this case.

Another explanation is that the function of the I-term is to help fixing the steady-state error. In fact, steady-state error will not be always shown if there is only one suitable kp being tuned. Besides, a larger ki will produce more oscillations and it makes a longer settling time.

4.4.2 The Best Tuning Result

4.5 Ramp Rate Analysis

From Section 3.5, we understand the importance of using ramp rate in power system. The aim is to be realistic. This is a real constraint in physical systems. The control cannot be used in reality if do not set the limit of it.

In Section 3.5, we discussed the reason to use PEAK and PEAKTIME to calculate the ramp rate. Equation is as follow.

$$Rate = \frac{P_{\text{PEAK}} - P_o}{\text{PEAKTIME}} \tag{4.2}$$

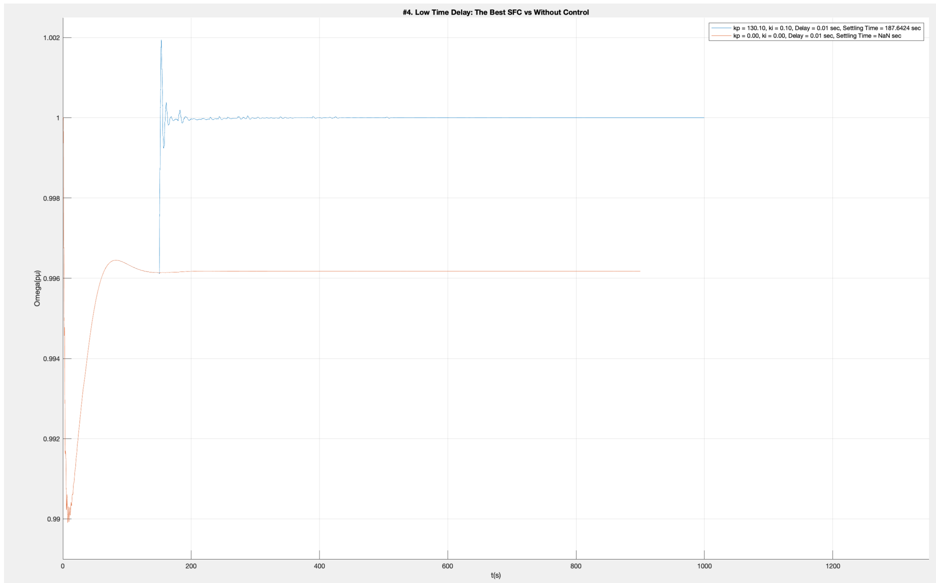


Figure 4.15: The best tuning signal

Next, we can calculate the limit of ramp rate of generators. According to National Renewable Energy Lab., Golden, CO (US), [9] large thermal units are usually able to ramp around 1% of their capacity per minute. Thus, we have Table 4.3:

Generator	Pnom [1] (MW)	Ramp Rate (MW/min)	Ramp Rate (MW/sec)
g6	360	3.6	0.06
g7	180	1.8	0.03
g14	630	6.3	0.105
g15	1080	10.8	0.18
g16	630	6.3	0.105

Table 4.3: Nominal ramp rate of generators

Then we can use the limit of ramp rate from Table 4.3 to remove some tuning results. The source code [2] is open to public.

Finally, we should have a figure that similar to Figure 4.14. However, as you can see from Table 4.4, none of tuning results can meet the requirement of the limit of ramp rate

[1] Pnom (nominal power) is original from: https://github.com/realgjl/sfcNordic/blob/master/examples/dyn_A.dat
[2] See https://github.com/realgjl/sfcNordic/blob/master/analysis/4.5/risk_lowTd.m

PARAMETERS				RAMP RATE (MW/min)				
kp	ki	Delay (sec)	Settling time (sec)	g6	g7	g14	g15	g16
50.1	3.1	0.01	229.5493	2.8490	2.1494	8.5368	26.1632	8.4787
30.1	3.1	0.01	265.6226	2.9639	1.6352	9.0306	26.2311	8.9688
100.1	10.1	0.01	197.4007	3.2202	164.8239	19.2880	7718.4360	19.2674
30.1	1.1	0.01	270.9029	3.2223	2.1412	18.8363	54.8532	18.8157
50.1	7.1	0.01	276.4196	3.4218	1.6093	18.8287	54.8261	18.8080
50.1	5.1	0.01	254.5951	3.4286	2.3714	27.7560	80.8243	19.0440
90.1	0.1	0.01	199.3726	3.5154	1.6284	19.0810	55.5505	19.0552
90.1	6.1	0.01	198.6749	3.5633	141.3201	19.3475	56.3445	19.3264
110.1	2.1	0.01	192.3948	3.5644	154.3812	19.2601	56.0877	19.2399
100.1	3.1	0.01	198.3431	3.6146	146.0997	19.1514	55.7720	19.1300
50.1	6.1	0.01	276.6197	3.6985	1.6605	19.4779	82.8241	19.4564
80.1	3.1	0.01	194.6444	3.7887	2.4101	19.1095	55.6475	19.0890
70.1	1.1	0.01	195.4580	3.8829	1.6604	11.7007	34.0736	11.6883
60.1	6.1	0.01	274.3786	3.8993	2.3916	18.8550	81.4826	18.8341
30.1	0.1	0.01	387.1230	4.8450	1.2569	14.6000	42.5168	14.5847
20.1	3.1	0.01	304.3321	6.0353	1.5498	18.1892	52.9735	18.1706
30.1	4.1	0.01	248.9825	6.1444	1.5787	18.5641	53.9345	18.5005
80.1	1.1	0.01	226.5649	6.1496	1.5794	18.5359	53.9779	18.5157
100.1	5.1	0.01	198.3080	6.1672	151.2526	18.5886	81.2217	18.5685
50.1	0.1	0.01	231.5627	6.1762	1.5782	18.6235	54.2320	18.5988
40.1	2.1	0.01	276.5414	6.1788	1.5787	18.4820	53.8201	18.4618
40.1	10.1	0.01	296.4604	6.1794	73.0790	18.6271	54.2451	18.6066
50.1	1.1	0.01	278.3003	6.1860	1.5922	18.6463	54.3010	18.6261
40.1	0.1	0.01	243.9987	6.1959	1.5953	18.6726	54.3760	18.6530
80.1	0.1	0.01	200.4365	6.1970	1.6014	18.6686	54.3632	18.6486
90.1	7.1	0.01	199.2704	6.2036	144.0517	18.8568	55.1432	18.8361
100.1	2.1	0.01	198.5242	6.2070	143.4520	18.7075	54.4759	18.6874
70.1	0.1	0.01	228.6037	6.2190	1.5888	18.5273	53.9544	18.5072
90.1	9.1	0.01	198.5086	6.2272	149.6897	18.7770	54.6747	18.7501
80.1	6.1	0.01	209.6003	6.2296	1.5921	18.7767	54.6791	18.7563
60.1	2.1	0.01	275.8390	6.2345	1.6084	18.7919	54.7213	18.7712
110.1	1.1	0.01	192.0149	6.2355	151.8984	18.7944	54.7310	18.7741
110.1	3.1	0.01	192.4948	6.2414	156.9059	18.8112	55.4817	18.7911
60.1	0.1	0.01	231.5185	6.2440	1.6043	18.8193	54.8033	18.7968
60.1	5.1	0.01	200.7154	6.2452	1.6052	18.8406	54.8115	18.8024
40.1	4.1	0.01	276.6012	6.2519	2.1444	18.8433	82.0929	18.8229
30.1	7.1	0.01	311.5328	6.2521	2.3920	18.8438	54.8741	18.8236
70.1	3.1	0.01	200.5358	6.2522	1.6103	19.0914	55.6008	19.0711
70.1	8.1	0.01	244.3974	6.2535	93.7337	18.8450	54.8809	18.8258
110.1	0.1	0.01	197.4203	6.2564	149.4185	18.8584	54.9158	18.8375
30.1	8.1	0.01	311.7186	6.2577	62.4280	18.8379	54.8553	18.8177
50.1	4.1	0.01	276.6000	6.2603	1.6097	18.8726	54.9691	18.8502
40.1	1.1	0.01	236.8172	6.2612	1.5911	18.8707	54.9554	18.8510

Table 4.4: Some generators' real ramp rates, ranked by g6's ramp rate.

(i.e. the ramp rates in the test system are larger than the limit). After repeated confirmation, I do not think there are any logic mistake in my algorithm.

In fact, our assumption is based on the materials from the United States. Although it's positive to reference an official document, the situation might be different between the United States and Nordic countries.

Wärtsilä Oyj Abp, a Finnish corporation which manufactures and services power sources and other equipment in the marine and energy markets, says, [10] ”Ramp rates of most industrial frame gas turbine models are advertised as 10 MW/min up to 100 MW/min, with an average of about 25 MW/min”. The maximum allowed ramp rate or the average one are significantly larger than the limit we set.

Another paper is also mentioned the similar information that, in Germany, [11] the maximum ramp can be up to 11%.

Thus, it's reasonable to believe that, in Nordic grid, the official ramp rate should be larger than 1% of the nominal power per min.

Another conclusion that we can sum up from Table 4.4 is that storage ramp rates are rapid (i.e. output can change quite rapidly). Power devices with a slow response time tend to have a slow ramp rate. This is the philosophical inspiration of Smart Grid operation: a fast response time would exceed the physical limit of a turbine which forces you to slow the response time to keep a balance between the design and the requirement.

Chapter 5

Impact of Time Delay

5.1 Hypothesis

The hypothesis in this chapter is basically based on the hypothesis in Section 4.1.

One of the different hypothesis is that it has several delays here, i.e. delay ranges between 0.01 seconds and 0.21 seconds, and the step is 0.01 seconds.

It allows having the similar hypothesis with Section 4.1 because, in this chapter, it can be seen as an extension of last chapter. In the last chapter, Nordic system is tested in a low delay while, in this chapter, we increase the delay.

An important assumption to allow keeping other parameters the same when increasing the time delay is that a larger time delay declines the performance of Nordic grid.

5.2 Expected Outcome

From the assumption of a larger delay declining the performance of Nordic grid and the results from Section 4.4, I expect in this chapter,

- The acceptable points will be less if the time delay is increased.
- To the best result of different delay, kp should be larger when time delay is increased.

I make this assumption based on my experience of waiting for traffic light. At first I didn't notice the green light until the car behind reminded me. In order to make up for the few seconds I missed, I stepped on the accelerator to increase the speed.

With the same reason, the signal facing a time delay problem should make up the errors by accelerating the speed of it. Thus, kp will increase if the time delay increases.

5.3 Implement

The only difference with Section 4.3 is we need to change the range of time delay. Details are shown as Figure 5.1.

```python
for td in np.arange(0.01, 0.22, 0.01):  # td: 0.01s ~ 0.21s, step: 0.01s
```

Figure 5.1: Implement different time delays

5.4 Results

5.4.1 Results and Analysis

All the acceptable points can be seen as Figure 5.2.

As we can see from Figure 5.4, overall, the shape of the 3D plot shows kp and ki are shrunk together when the time delay increases. An increasing time delay removes some larger kis. Detailedly, if a controller has a larger kp and a large ki, the simulation results are recognised as unacceptable points in high probability. If a controller has a high kp but a small ki when the time delay is increased, the simulation results will not be recognised as

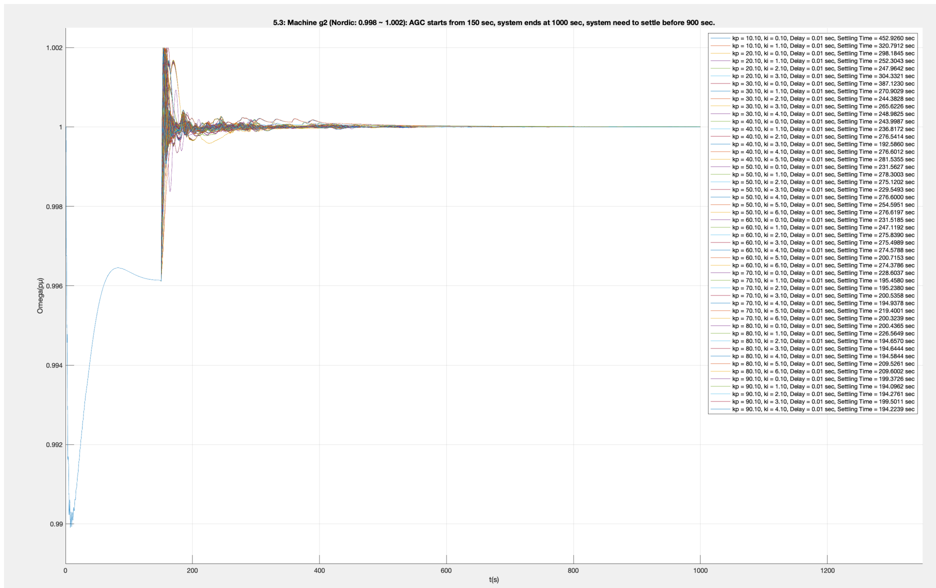


Figure 5.2: Impact of time delay: all the acceptable points

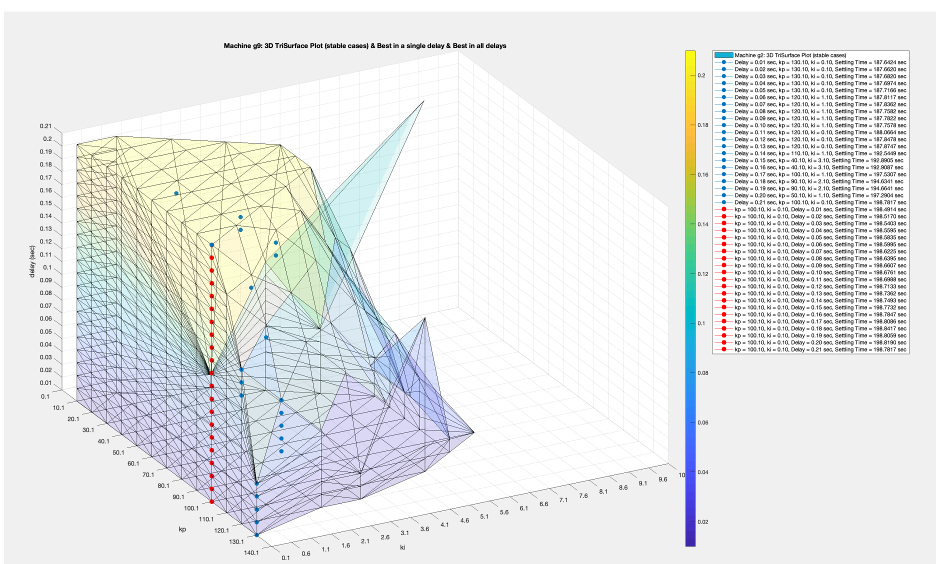


Figure 5.3: 3D plot: contain outliers

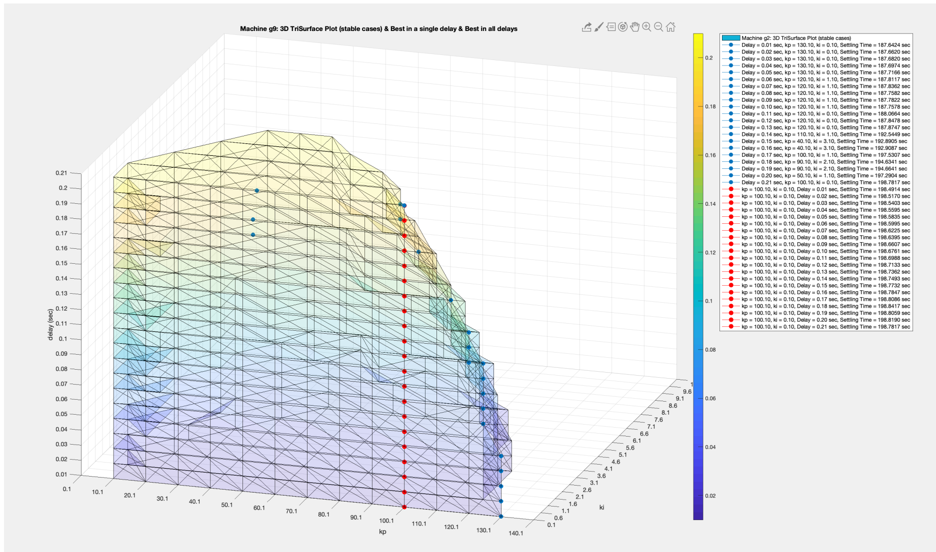


Figure 5.4: 3D plot front side: without outliers

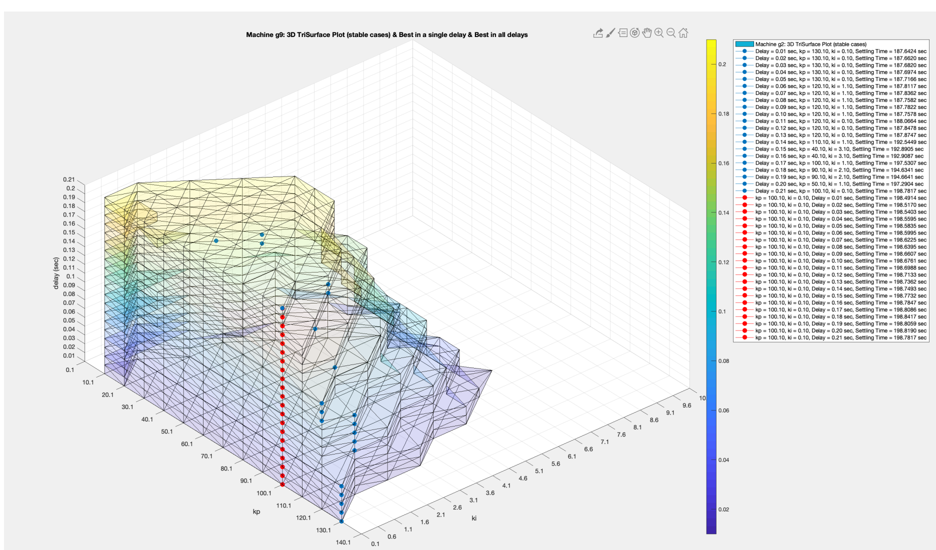


Figure 5.5: 3D plot right side: without outliers

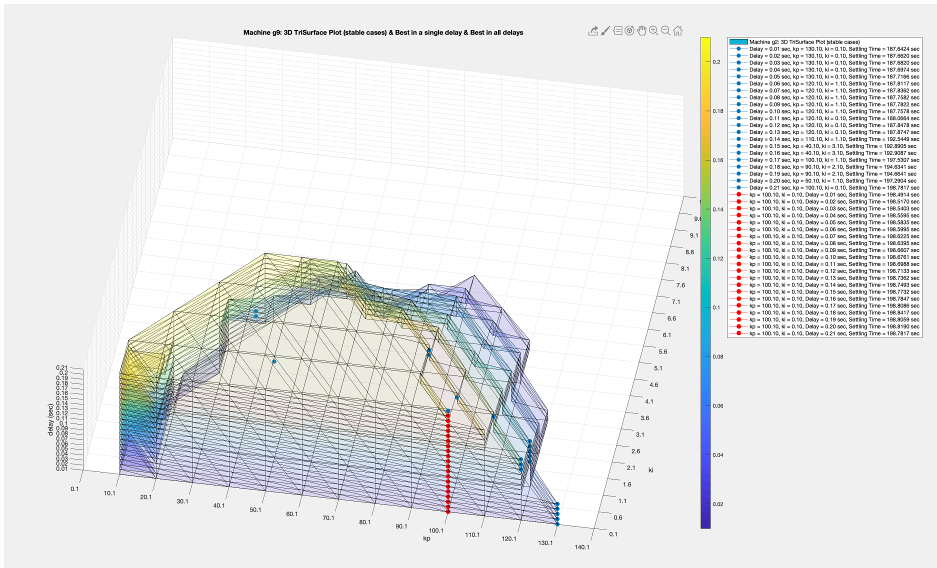


Figure 5.6: 3D plot top side: without outliers

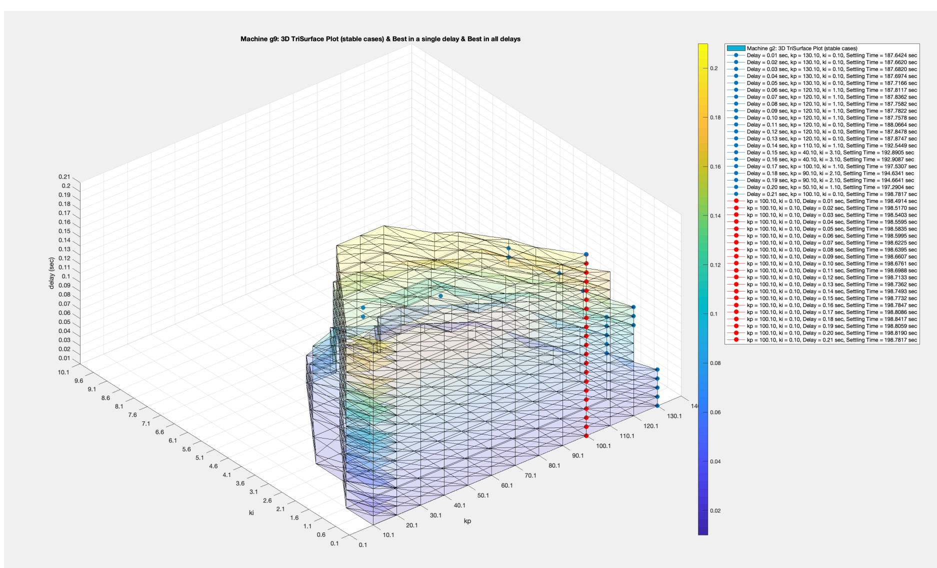


Figure 5.7: 3D plot left side: without outliers

unacceptable points in a large probability.

We rank the average settling time for every tuning situation. We finally find that when kp is 100.1 and ki is 0.1, the signal can approach fastest to the nominal value in average.

Besides, we find that for the same ki and ki value, for instance, when kp is 100.1 and ki is 0.1, the settling time has a trend of increasing when time delay is increased. This directly prove the previous expectation on the impact of the time delay from Section 5.2 and shows the time delay weakens the stability of the control.

Another important result is shown in the legend bar in the Figure 5.4. The highlighted blue points are the best result for their delay. For example, "Delay = 0.01 sec, kp = 130.1, ki = 0.10, Settling Time = 187.6424sec" in the legend bar means when time delay is 0.01 seconds, the signal fastest approach to the nominal value if its kp is 130.1 and its ki is 0.10. We did not find that the kp value will increase if time delay increases as expected in Section 5.2. However, the reference of ki is changing: when delay is 0.01 seconds, the best ki is 0.01 however, when delay is 0.09 seconds, the best ki is 1.1, etc. The reason for this case might be the threshold for frequency (i.e. the frequency threshold should be in the range of 0.2% in Nordic system) is too small for the signal. A large kp will first exceeds the limit of frequency threshold, then this large kp and its ki will have no chance appearing in the acceptable results.

5.4.2 The Best Tuning Result

The best tuning result is compared with the signal without SFC control and is shown in Figure 5.8.

5.5 Ramp Rate Analysis

Similar to Section 4.5, it's important to consider the aspect of ramp rate to make sure a scientifically correct although I didn't simulate the results for this Chapter due to the results

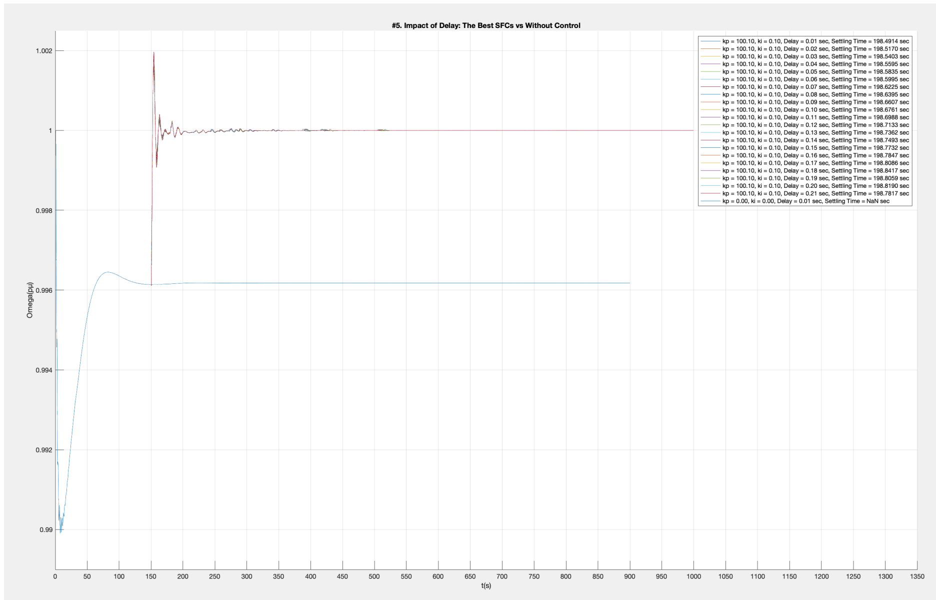


Figure 5.8: Impact of Delay: The Best SFCs vs Without Control

in Section 4.5.

However, we can still draw a conclusion that the limit of ramp rate will remove some fast tuning results from one of the conclusions in Section 4.5. More than that, we can reasonably infer the following conclusions: due to the fact that some higher-kp tuning results has been removed by a larger time delay (Figure 5.5), ramp rate affects the tuning results of higher time delay less than the tuning results of lower time delay.

Chapter 6

Emergency Control

6.1 Overview

Kundur and Morison discussed about emergency control in TECHNIQUES FOR EMERGENCY CONTROL OF POWER SYSTEMS AND THEIR IMPLEMENTATION. In that paper, they introduced alert state before describing emergency control. They think, [12] an alert state will be entered if the security level is below ”a certain limit of adequacy”, or if the probability of heavy disturb increases due to the weather conditions. However, in an alert state, all constraints are met. Emergency state is at state that one or more constraints are not met. Thus, the emergency control tries to do prevent the emergency state happening and an emergency control will triggered at the alert state.

In all, emergency control should be established in emergency conditions to minimise the risk of further uncontrolled separation, loss of generation, or system shutdown. In reality, [12] alert thresholds can be established. If the system exceeds one of the alert thresholds, emergency control can start.

In our case, emergency control starts from the lowest point of frequency in Primary Frequency Control and use Secondary Frequency Control to restore its frequency to prevent the risk of black out.

6.2 Hypothesis and Implement

Referencing Figure 4.3 in Subsection 4.1.1, we find one of the lowest points to simulate one of the worst situation we can find in Noridc system, as shown in the Figure 6.1, as our start point of Emergency Control.

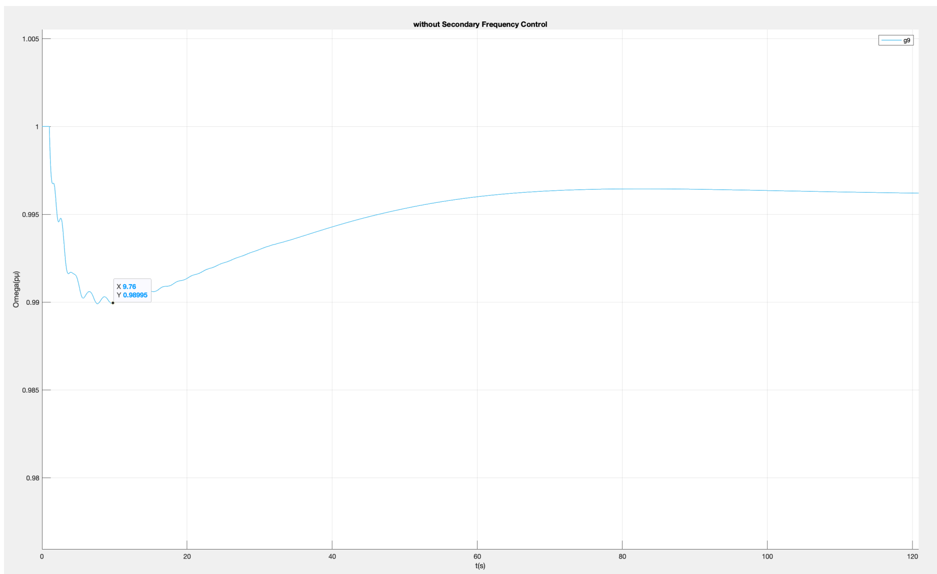


Figure 6.1: Start time of Emergency Control.

We start the Emergency Control from the 9.76th second and tuned kp, ki and time delay, i.e. kp is from 0.1 to 140.1 (step is 10.0), ki is from 0.1 to 10.1 (step is 1.0) and time delay is from 0.01 seconds to 0.21 seconds (step is 0.01sec), which are the same tuning conditions in Chapter 5. The aim of such hypotheses is to test the impact of time delay under the condition of Emergency Control and compare the simulation results with SFC. The simulation lasts for 1000 seconds.

6.3 Expected Outcome

What triggers an alert state will disturb a power grid system and may [13] even blackout mojor grids in several areas while time delay will affect the stability of the control (Subsection 5.4.1).

In all, emergency control will be affected by time delay and the controller's stability will be weakened. Besides, due to an alert state, the system will be disturbed, thus the stability will be weakened again. I predict that there are less acceptable results for emergency control.

6.4 Results

6.4.1 Results and Analysis

All the acceptable results can be seen as Figure 6.2. Compared to Figure 5.2, we can see there are less acceptable tuning results in the emergency control.

Separately, when delay is 0.01 seconds, as shown in Figure 6.2 and in Figure 6.3, there are only four acceptable results for the emergency control. Besides, emergency control removes high-kp values to make sure the overshoot is in the set range. Comparing Figure 6.4, Figure 6.6 and Figure 6.8, the system is more unstable when the time delay increases.

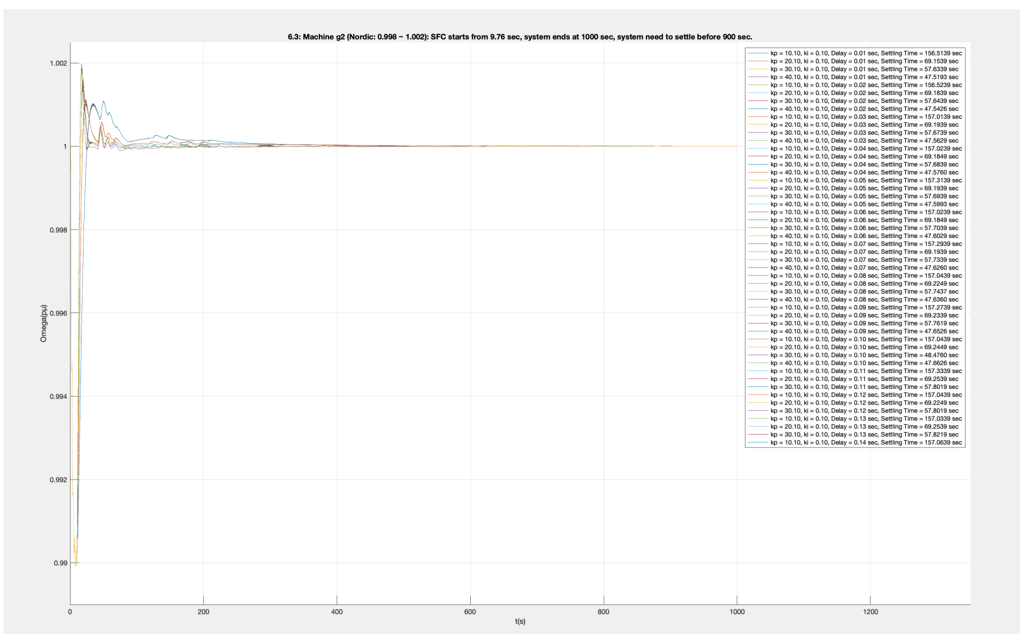
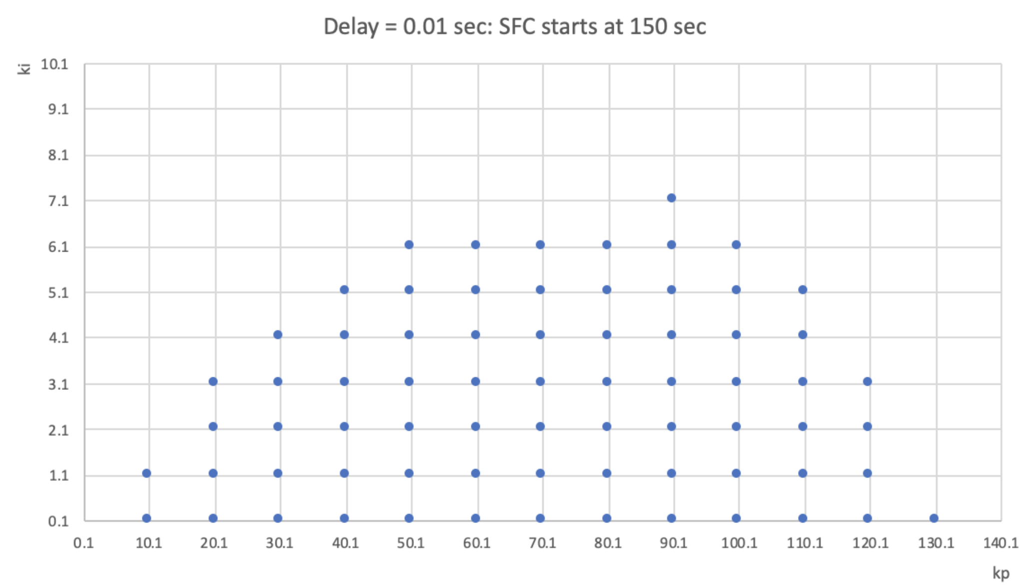


Figure 6.2: Acceptable results of Emergency Control.



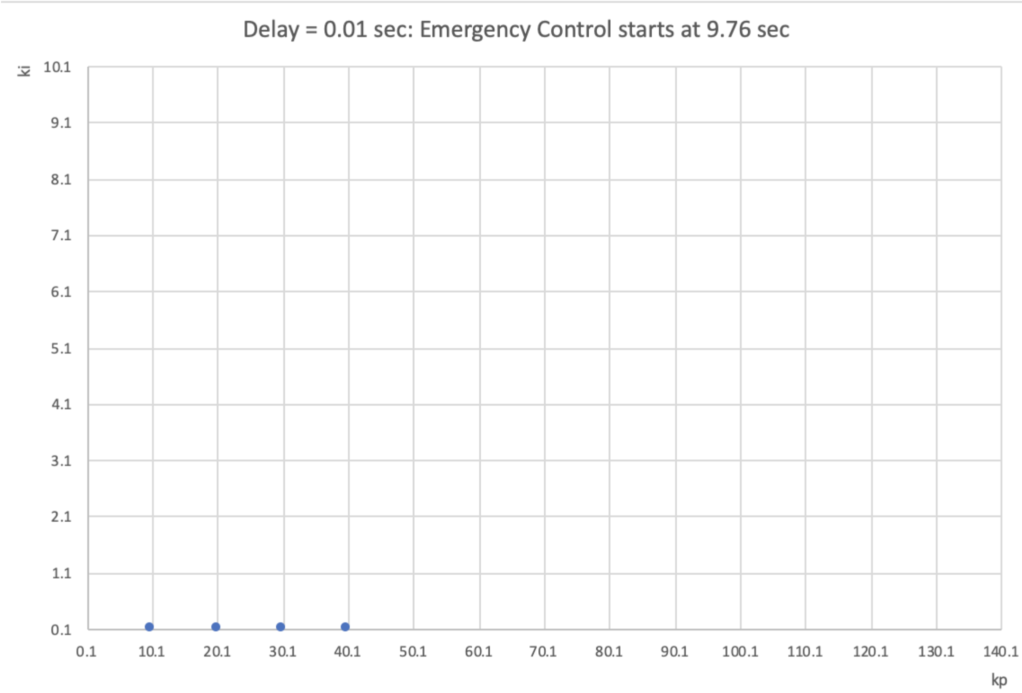


Figure 6.3: Comparison: Delay is 0.01 sec

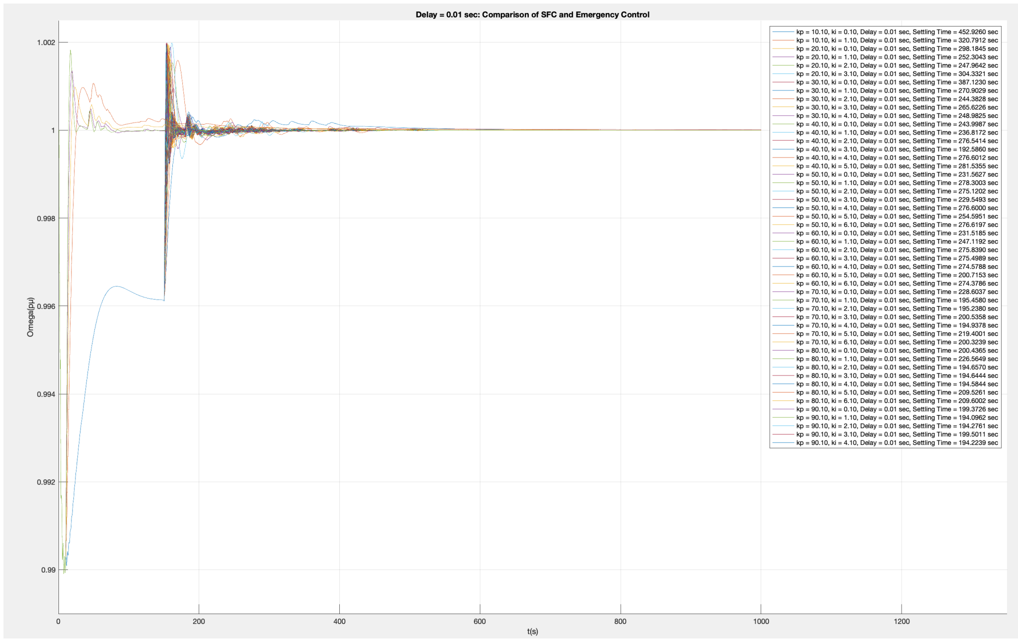


Figure 6.4: Sketches for SFC and Emergency Control: Delay is 0.01 sec

6.4.2 The Best Tuning Result

The best tuning result is compared with the signal without SFC control and is shown in Figure 6.9.

6.5 Ramp Rate Analysis

Like what we do in Section 4.5 and in Section 5.5, we need to find the ramp rate in the emergency control. Results are as Table 6.1. Comparing Table 4.4 and Table 6.1, we can find that the grid needs a larger ramp rate in the emergency control which shows that an alert state does weaken the stability of the system.

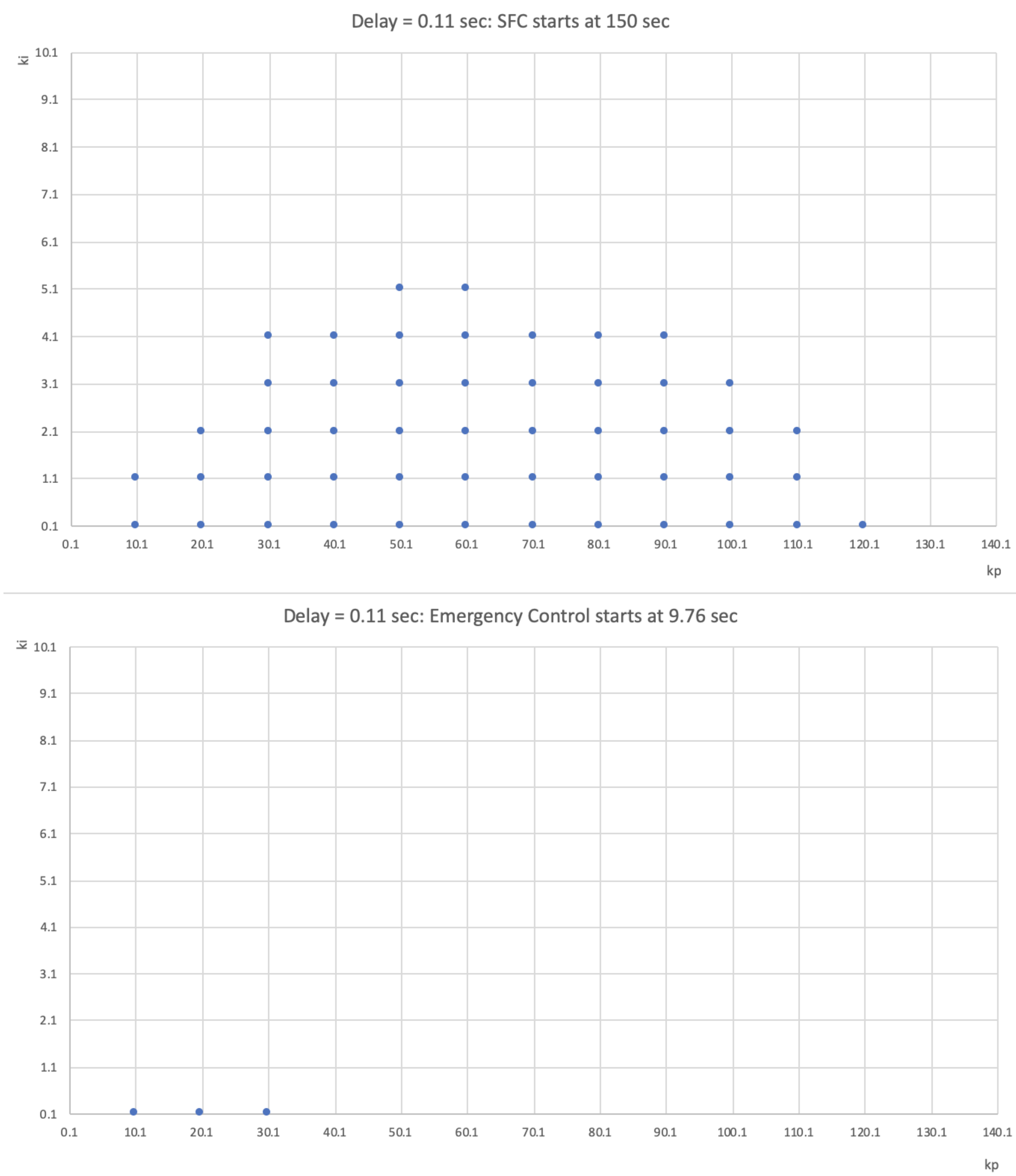


Figure 6.5: Comparison: Delay is 0.11 sec

PARAMETERS				RAMP RATE (MW/min)				
KP	KI	DELAY (sec)	SETTLINGTIME (sec)	g6	g7	g14	g15	g16
0.1	0.1	0.01	216.0308	39.3739	11.0131	120.8436	341.6526	117.3243
10.1	0.1	0.01	156.5139	103.1334	25.3490	338.3861	977.3432	317.4613
10.1	7.1	0.01		178.6439	137.4221	548.4813	1665.9473	537.6921
0.1	1.1	0.01		185.2809	47.9584	563.4105	1711.7261	558.2290
10.1	1.1	0.01	122.5678	185.7758	45.5807	610.6689	1801.5655	602.2359
20.1	0.1	0.01	69.1539	226.9924	51.7346	720.8555	2410.3649	710.0906
10.1	2.1	0.01	87.8304	261.1636	63.9143	809.9728	2594.6601	800.8921
20.1	1.1	0.01	57.5939	273.1597	62.8920	860.7161	2882.4697	849.5901
0.1	2.1	0.01		304.8005	79.0852	922.2324	2685.9862	911.9675
20.1	2.1	0.01	49.0939	308.1996	77.9355	1067.6620	3230.6748	1053.6556
10.1	3.1	0.01	169.2139	333.0774	81.2400	1032.2398	3368.0729	1019.9471
20.1	3.1	0.01	52.1529	369.1920	86.7042	1165.3831	3963.0098	1148.7566
30.1	0.1	0.01	57.6339	381.9509	82.9905	1086.3220	3855.5082	1071.0816
10.1	4.1	0.01	257.0999	386.9496	95.2370	1199.4026	3968.1442	1186.2328
0.1	3.1	0.01		389.2433	99.6157	1203.1748	3652.9045	1205.4600
10.1	5.1	0.01	888.4539	405.8005	111.2012	1380.4250	4627.0644	1364.1282
30.1	1.1	0.01	57.7139	424.7082	92.5645	1199.4765	4268.7561	1183.4528
20.1	4.1	0.01	71.4229	444.6857	103.1471	1389.9621	4835.7601	1372.0536
0.1	4.1	0.01		458.4640	116.2877	1441.7190	4402.9113	1439.1193
30.1	2.1	0.01	48.6629	463.6436	101.7606	1298.9049	4631.4775	1282.2416
10.1	6.1	0.01		466.4620	130.4226	1437.8227	4807.1216	1420.0287
20.1	5.1	0.01	81.2579	469.5291	110.8371	1469.8462	5133.4152	1451.4619
40.1	0.1	0.01	47.5193	487.2553	116.1599	1573.7708	6169.7847	1554.3084
20.1	6.1	0.01	129.3339	489.6735	129.0187	1728.6360	5375.9589	1708.9875
30.1	3.1	0.01	47.9529	498.4043	110.5302	1582.3456	4941.5245	1561.3353
0.1	5.1	0.01		519.1297	131.3588	1595.6927	5137.1947	1583.8335
40.1	1.1	0.01	57.2729	520.2965	124.6185	1676.7190	6564.8858	1656.5071
30.1	4.1	0.01	46.5060	530.2170	118.7860	1678.2056	6218.0761	1657.1749
40.1	2.1	0.01	47.7429	551.4328	133.0771	1773.4577	6942.6801	1752.5240
30.1	5.1	0.01	50.9518	559.1271	126.5235	1769.7339	6561.3747	1743.5392
20.1	7.1	0.01	136.2504	573.8505	137.7249	1804.3393	5525.7482	1783.8580
40.1	3.1	0.01	47.2060	580.7121	141.9224	1864.0236	7303.0228	1842.4242
0.1	6.1	0.01		583.2017	146.8523	1783.4183	5355.3437	1771.2938
30.1	6.1	0.01	50.4381	585.2393	133.6340	1847.3300	6878.0007	1820.7210
20.1	8.1	0.01	179.3539	594.6671	145.1665	1868.2641	6713.2441	1846.1036
30.1	7.1	0.01	55.4139	606.9717	140.1626	1915.9090	7143.6266	1888.7611
20.1	9.1	0.01	200.7214	608.1554	170.0947	1920.1625	6928.5437	1891.6607
40.1	4.1	0.01	46.5429	608.2092	150.3688	1948.5746	7646.6520	1926.3334
30.1	8.1	0.01	55.4526	627.7776	164.0650	1975.7122	7408.6481	1947.9462
40.1	5.1	0.01	51.2808	633.9012	158.7517	2027.1486	7972.9745	2004.3154
0.1	7.1	0.01		640.1096	163.2624	1999.7925	6300.1633	1990.3871
10.1	8.1	0.01		656.9294	165.6534	1957.4121	6528.2415	1949.5629
40.1	6.1	0.01	45.5429	657.8388	167.0058	2099.8722	8247.0341	2076.4102

Table 6.1: Emergency Control: Some generators' real ramp rates, ranked by g6's ramp rate.

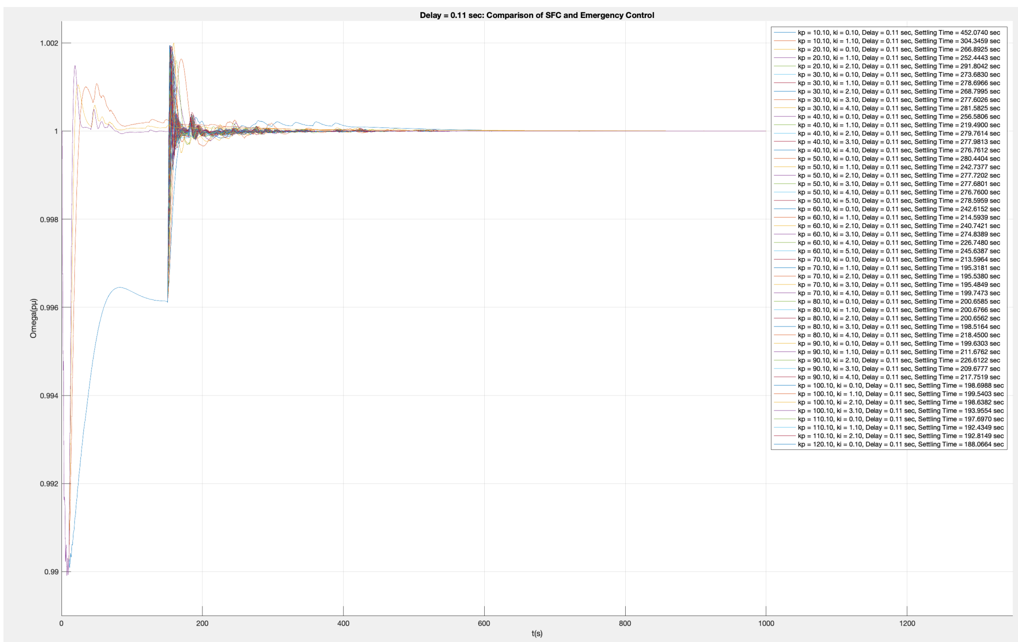


Figure 6.6: Sketches for SFC and Emergency Control: Delay is 0.11 sec

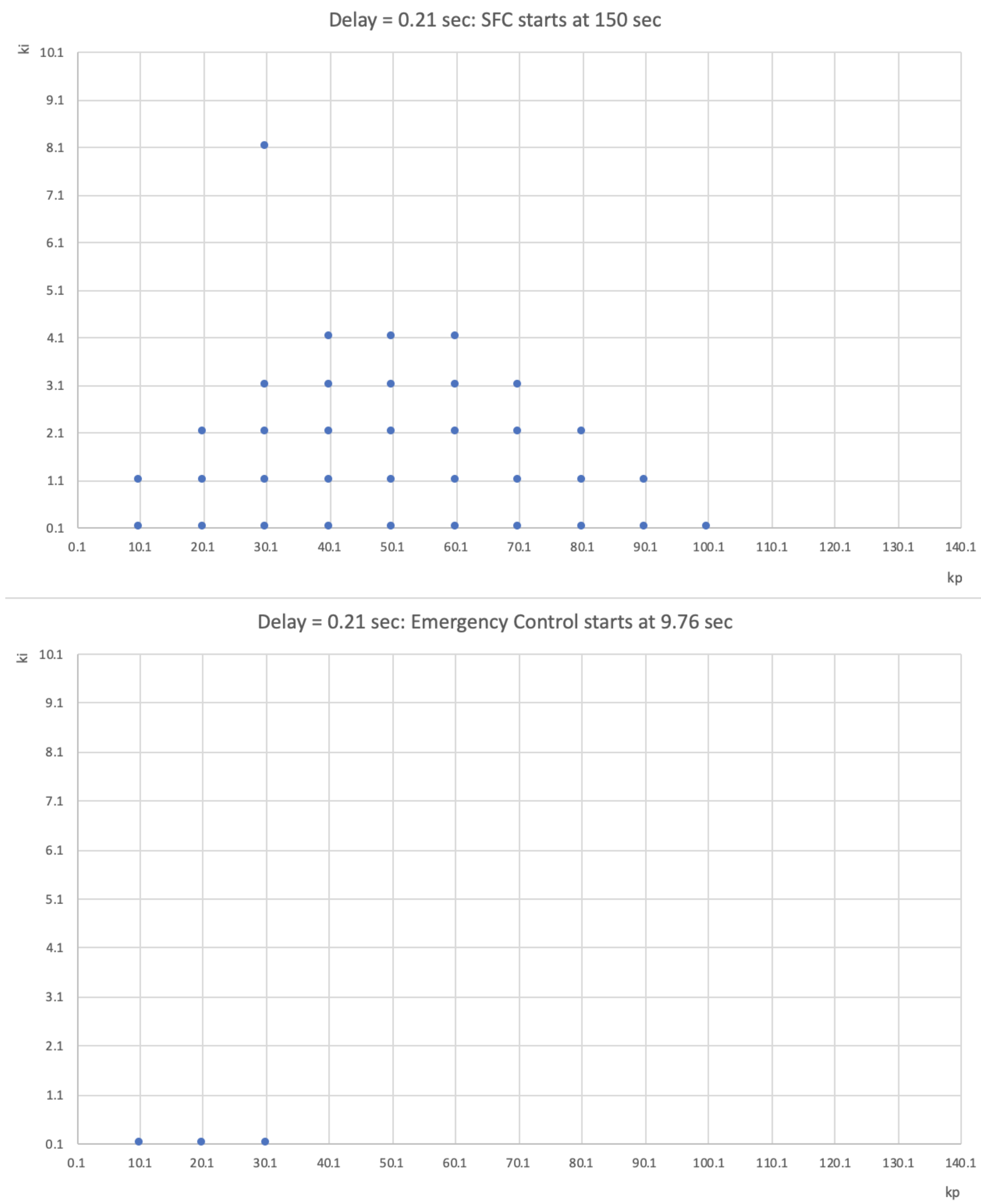


Figure 6.7: Comparison: Delay is 0.21 sec

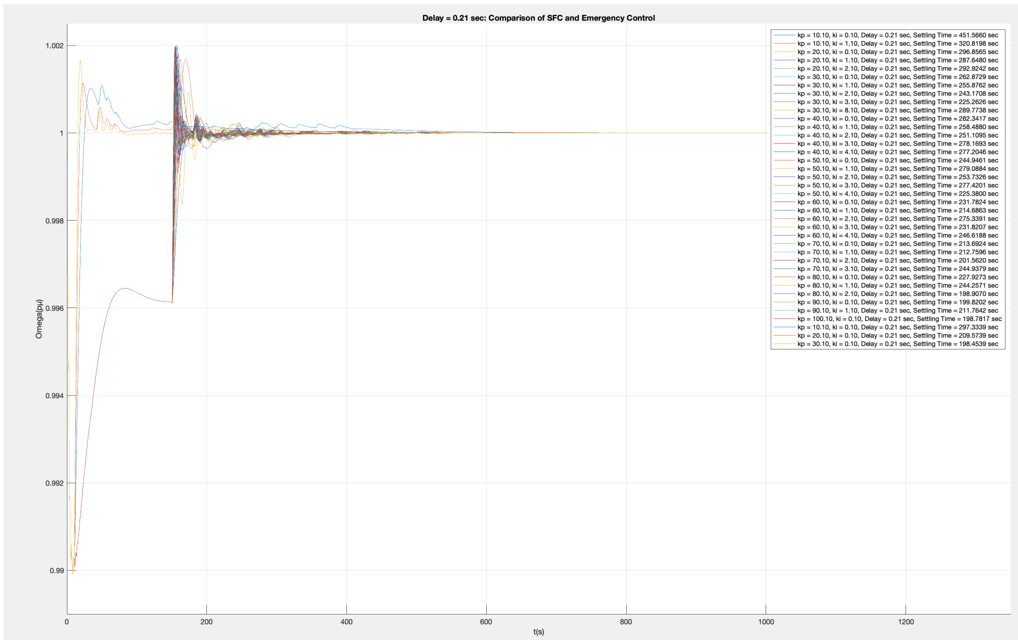


Figure 6.8: Sketches for SFC and Emergency Control: Delay is 0.21 sec

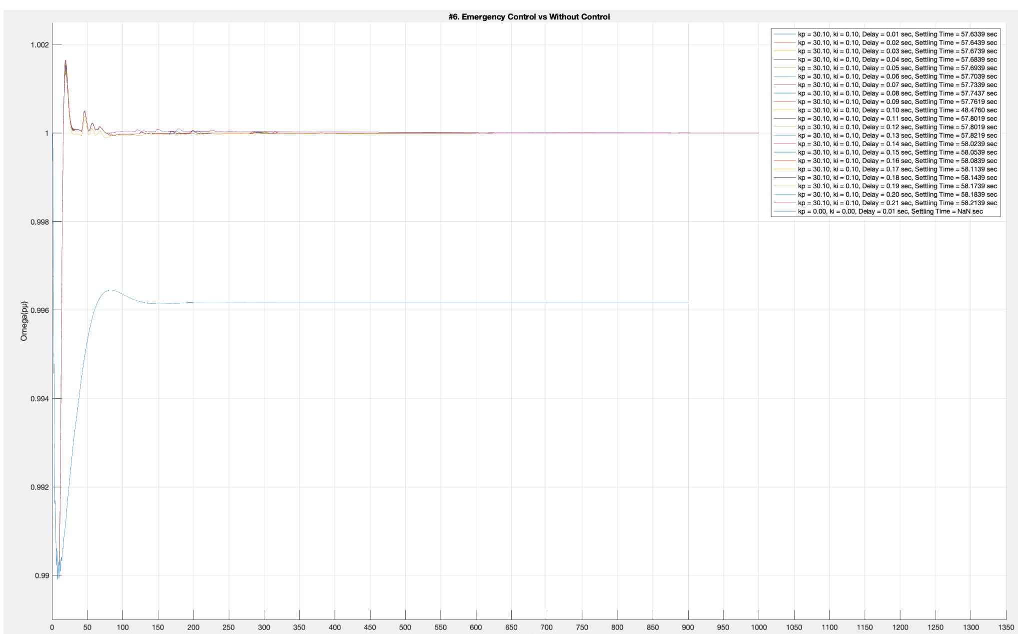


Figure 6.9: Best Emergency Control vs Without Control

Chapter 7

Conclusion

In this project, we gave readers a thorough overview of Secondary Frequency Control and PI control: the algorithms (PART I) and the test case scenario (Nordic) (PART II), as well as how we contributed to the analytical tools to connect the theories and testings.

In Chapter 2, we walked through the overview of Frequency Control, including Primary Frequency Control, Secondary Frequency Control and Tertiary Frequency Control. We knew the reasons we need Frequency Control and the mechanisms of the three Frequency Controls. Equally important is that we knew the response time and the feature of the three Frequency Controls.

In Chapter 3, we covered the theory elements of Secondary Frequency Control and PI control. We introduced the physical theory behind Secondary Frequency Control and the logic of a central control algorithm. We explained we need to simplify the model to remove some conditions that do not affect the simulation results. We introduced the mathematical and physical theory of PID control. Through the frequency problems in the reality, we understood the reasons of using P-term and the I-term in SFC and the reasons of not using D-term because of amplifying the high component parts in the signal. Then, we introduced the core codes in PI control in Python and how we sent the information into the generators to fix the frequency problems. Besides, we introduced the deadband error and discussed the deadband control algorithm. In the last part of this chapter, we introduced our tuning methodology. First, we analysed the impacts of ki and then we introduced bisection method into tuning models. Then, we introduced analytical models written in Python and MATLAB and finally they can be used in filtering unacceptable results according to signal requirements, in generating 2D diagram, and in generating 3D triangle surface plot with multiple needs.

In PART II, the key questions we wanted to answer are: Is there any feature of a PI control under the condition of low time delay? What are the impacts of time delay to such a PI controller? Where is the best tuning result? What are the impacts of the rate of change? What are the impacts of time delay to the emergency control? Is there any impact on the stability of the system?

In Chapter 4, we put forward how to assume the conditions, i.e. hypothesis, of the Nordic system and finally, we choose suitable value of kp, ki, delay, start time, end time etc. Then, we discussed the expected outcome based on physical theory of SFC before the simulation starts. Then, we introduced how to tune PI control and finally we showed you the simulation results. We thought a larger kp responses faster but at the same time it can potentially exceed the speed limit of power output. Finally, we introduce the ramp rate which is significant in real life. It is concluded that the ramp rate can remove some unacceptable results but we need to set the limit of ramp rate reasonably.

In Chapter 5, like in Chapter 4, we put forward how to assume the conditions of the Nordic system and finally, we choose the range of time delay and keep the condition in Chapter 4. Then, we discussed the expected outcome based on physical theory of SFC before the simulation starts. Then, we introduced how to tune time delay and finally we showed you the simulation results. We think there are some outliers that will interfere the analysis so we decided to filter them. We concluded that acceptable results are less and kp and ki are shrank to each other with a larger time delay.

In chapter 6, we discussed the impact of time delay under the condition of Emergency

Control. We concluded that there is no difference for Emergency Control and a SFC interns on the impact of time delay: the controllers are affected by the time delay. Besides, we analysed the ramp rate of the emergency control and compare it with the SFC. We concluded that the emergency control requires a larger ramp rate than the SFC needs. This poses a challenge to the stability of the grid in an emergency.

I am really excited about the progress that has been made for the past one academic year. At the same time, we also deeply believe that there is still a long way to go towards smart grid algorithms, and we are still facing challenges and a lot of open questions that need to be addressed in the future. For instance, we need spend more time on developing reasonable limits of ramp rate. We need to analyse every possible emergency situation and find a way to predict it and prevent it. Another challenge is that we need to keep designing more advanced algorithms. Algorithms like Artificial Neural Network (ANN) [14], [15], Genetic Algorithms [14], [16], [17] and Particle Swarm Optimisation (PSO) Algorithms [14], [18] has been discussed in the area for years, but so far we haven't found a universal perfect algorithm yet.

Bibliography

[1] P. Aristidou and T. Van Cutsem, "A parallel processing approach to dynamic simulations of combined transmission and distribution systems," *International Journal of Electrical Power & Energy Systems*, vol. 72, pp. 58–65, 2015.

[2] P. Aristidou, D. Fabozzi, and T. Van Cutsem, "A schur complement method for dae systems in power system dynamic simulations," in *Domain Decomposition Methods in Science and Engineering XXI*. Springer, 2014, pp. 719–727.

[3] S. Devarajan and A. C. Fisher, "Hotelling's" economics of exhaustible resources": Fifty years later," *Journal of Economic Literature*, vol. 19, no. 1, pp. 65–73, 1981.

[4] C. Parmesan and G. Yohe, "A globally coherent fingerprint of climate change impacts across natural systems," *Nature*, vol. 421, no. 6918, p. 37, 2003.

[5] R. K. Pachauri, M. R. Allen, V. R. Barros, J. Broome, W. Cramer, R. Christ, J. A. Church, L. Clarke, Q. Dahe, P. Dasgupta, N. K. Dubash, O. Edenhofer, I. Elgizouli, C. B. Field, P. Forster, P. Friedlingstein, J. Fuglestvedt, L. Gomez-Echeverri, S. Hallegatte, G. Hegerl, M. Howden, K. Jiang, B. J. Cisneroz, V. Kattsov, H. Lee, K. J. Mach, J. Marotzke, M. D. Mastrandrea, L. Meyer, J. Minx, Y. Mulugetta, K. O'Brien, M. Oppenheimer, J. J. Pereira, R. Pichs-Madruga, G.-K. Plattner, H.-O. Pörtner, S. B. Power, B. Preston, N. H. Ravindranath, A. Reisinger, K. Riahi, M. Rusticucci, R. Scholes, K. Seyboth, Y. Sokona, R. Stavins, T. F. Stocker, P. Tschakert, D. van Vuuren, and J.-P. van Ypserle, *Climate Change 2014: Synthesis Report. Contribution of Working Groups I, II and III to the Fifth Assessment Report of the Intergovernmental Panel*

on Climate Change, R. Pachauri and L. Meyer, Eds. Geneva, Switzerland: IPCC, 2014.

[6] P. Papadopoulos, L. Cipcigan, N. Jenkins, and I. Grau, "Distribution networks with electric vehicles," in *2009 44th International Universities Power Engineering Conference (UPEC)*. IEEE, 2009, pp. 1–5.

[7] P. Komarnicki, C. Dzienis, Z. Styczynski, J. Blumschein, and V. Centeno, "Practical experience with pmu system testing and calibration requirements," in *2008 IEEE Power and Energy Society General Meeting-Conversion and Delivery of Electrical Energy in the 21st Century*. IEEE, 2008, pp. 1–5.

[8] J. Machowski, J. Bialek, and J. Bumby, *Power system dynamics: stability and control*. John Wiley & Sons, 2011.

[9] B. Kirby and M. Milligan, "Method and case study for estimating the ramping capability of a control area or balancing authority and implications for moderate or high wind penetration: Preprint," pp. 3,7, 2005.

[10] Wärtsilä. Combustion engine vs gas turbine: Ramp rate. [Online]. Available: https://www.wartsila.com/energy/learning-center/technical-comparisons/combustion-engine-vs-gas-turbine-ramp-rate

[11] M. Huber, "Flexibility in power systems-requirements, modeling, and evaluation," Ph.D. dissertation, Technische Universität München, 2017.

[12] P. Kundur and G. Morison, "Techniques for emergency control of power systems and their implementation," *IFAC Proceedings Volumes*, vol. 30, no. 17, pp. 639–644, 1997.

[13] G. Andersson, P. Donalek, R. Farmer, N. Hatziargyriou, I. Kamwa, P. Kundur, N. Martins, J. Paserba, P. Pourbeik, J. Sanchez-Gasca, *et al.*, "Causes of the 2003 major grid blackouts in north america and europe, and recommended means to improve system dynamic performance," *IEEE transactions on Power Systems*, vol. 20, no. 4, pp. 1922–1928, 2005.

[14] S. K. Pandey, S. R. Mohanty, and N. Kishor, "A literature survey on load–frequency control for conventional and distribution generation power systems," *Renewable and Sustainable Energy Reviews*, vol. 25, pp. 318–334, 2013.

[15] D. Chaturvedi, P. Satsangi, and P. Kalra, "Load frequency control: a generalised neural network approach," *International Journal of Electrical Power & Energy Systems*, vol. 21, no. 6, pp. 405–415, 1999.

[16] P. Dangprasert and V. Avatchanakorn, "Genetic algorithms based on an intelligent controller," *Expert systems with applications*, vol. 10, no. 3-4, pp. 465–470, 1996.

[17] D. Rerkpreedapong, A. Hasanovic, and A. Feliachi, "Robust load frequency control using genetic algorithms and linear matrix inequalities," *IEEE Transactions on Power Systems*, vol. 18, no. 2, pp. 855–861, 2003.

[18] S. Hosseini and A. Etemadi, "Adaptive neuro-fuzzy inference system based automatic generation control," *Electric Power Systems Research*, vol. 78, no. 7, pp. 1230–1239, 2008.