Imperial College London

Department of Electrical and Electronic Engineering

Final Year Project Report 2019

Project Title:	**Data-driven techniques to obtain constraints for optimisation problems**
Student:	**Keerthanen Ravichandran**
CID:	**01195170**
Course:	**EIE3**
Project Supervisor:	**Dr Fei Teng**
Second Marker:	**Dr G. Scarciotti**

1 Abstract

The high penetration of renewable energy sources has led to the decline in systems driven by inertia. This project will look into the frequency stability problem, in particular developing frequency security constraints for Rate of Change of Frequency (RoCoF), nadir and quasi-steady-state frequency in multi-area system with distinct locational frequency signatures. The solution that was developed uses various data driven techniques, in that data points from a simulation were gathered and processed in order to produce a suitable frequency constraint. The project found constrained regression produced the safest constraint however, it was very conservative and compromised on efficiency. The unconstrained regression was not suitable due to the the significant compromise in safety. The main challenges include using appropriate hardware and algorithms to collect enough data points and testing the model rigorously to ensure the accuracy was credible. The solution to this problem will aid an optimisation problem later on.

2 Acknowledgements

I would like to thank Fei Teng and Luis Bernado for providing me with the support required to work on and complete this project. The support they have provided me with has allowed me to gain a better understanding in the field of electronics and invaluable experience in the researching field. I devote this work to my parents who have support me morally and financially throughout my studies.

Contents

3 Introduction

This chapter aims to cover the topics that will be discussed in this project. It describes the purpose of an electrical power system, how recent changes in the sources of energy has an impact on how an electrical power system functions and how this has resulted in the need for fast frequency response services.

3.1 Electrical Power systems

Electrical power systems provide a means of supplying, transferring and using electrical energy. The supply of electrical energy has typically been generated by converting mechanical energy from non-renewable sources such as fossil fuel, into electrical energy using large rotating synchronous generators. Consumers are connected to the power system can use it to obtain a supply of energy. The diagram below shows a simple illustration of the basic structure of an electrical system. The system consists of components that generate electricity (power stations), transmit electricity (transmission lines) and distribute energy (substation step-down transformers). [1]

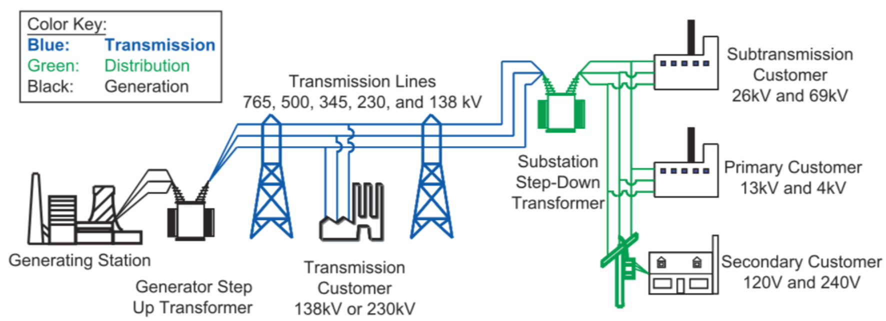

Figure 1: Frequency evolution after a contingency

3.2 Post-fault frequency requirements

System operators of power systems have to consider many factors to ensure that the electrical power system provides a consistent and reliable supply of energy. One important consideration is the balance between the supply and demand of energy. With conventional generation, a imbalance would result in the frequency of rotating masses, which are in all synchronous generators, to speed up (excess power supply) or slow down (excess power demand).[2] This enables the system operator to keep the AC frequency within the limits according to guidelines for that system. In the UK, the AC power is 50Hz with a deviation of 0.5Hz in frequency. [3]

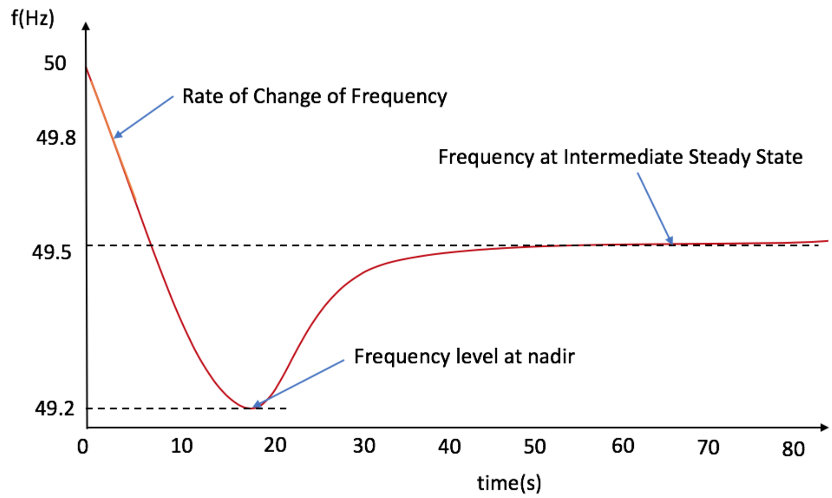

Figure 2: Frequency evolution after a contingency

To ensure the stabilisation of frequency, there are three post-fault requirements that must be met.

1. RoCoF, rate of change of frequency, which is proportional to the power shortage and inversely proportional to the system inertia
2. Frequency nadir, the minimum points achieved by system frequency during the transient period
3. Quasi-steady-state, which occurs after the essential amount of PFR delivered by the generators

The integration of renewable energy sources such as wind, combined with the decline of conventional fossil fuel generation, has led to the decline in the level of system inertia. The level of system inertia will impact the control the system operator has in order to keep the AC frequency within the limits. This leads to the increase in demand of services that provide fast frequency responses. These responses will improve the management of system frequency when a fault occurs, ensuring that the system frequency is closer to 50 Hz under normal operation.

The overall aim of this project is develop a constraint by examining the relationship between system inertia and the primary frequency response for a multi area system. The mixed integer linear programming solved by Teng Fei has optimised the system operation for a single area system. [4] When considering a multi-area system, the process of optimising the system becomes a complex task due to the complex dependency of frequency evolution. The project will aim to create a frequency constraint that can be used in an optimisation problem without comprising the complexity of the problem. Other variable will have an impact on the nadir frequency constraint, however the variables that will be covered are the variables that have the most significant impact.

3.3 Data Driven approaches in power systems

Data driven approaches are becoming an increasingly common way to solve problems in electrical power systems. According to National Audit Office, around 70-75% of meters are expected to be replaced by smart meters.[5] These smart meters can communication information at more regular basis and provide a more reliable and cost-effective way to control power generation. Therefore, data driven approaches such as the methods that will explored in this paper will be relevant in the wider context.

3.4 Structure

The report will have the following structure:

1. **Background-** this part will consist of briefly looking over the research papers that exist in relation to finding the nadir frequency constraint
2. **Requirement Capture-** the primary goals that will be accomplished in this project
3. **Analysis and Design-** looking at the possible methods that could be used to solving the problem
4. **Implementation-** focuses on how the dataset for regression was obtained and the testing of the frequency constraint was completed, including the results
5. **Evaluation-** a summary of the results, covering whether they showed that a frequency constraint was possible and reflecting on the original objectives
6. **Conclusion -** provides a summary the report and any further extensions that can be completed to improve the research

4 Background

This section of the report will cover the published work that was reviewed before solving the problem and provide an insight of the problem that is to be solved. As the primary focus of the project was the data driven element, a high level view of the electrical system is given.

4.1 Decline of inertial response

An inertial response is a property of large synchronous generators, which contains large rotating masses that can overcome an immediate unbalance between power generation and demand in an electrical power system.[2] These rotating masses contain a store of kinetic energy that can be released by slowing the masses down if there is excess demand, or speeding them masses if there is excess power supply.[6] [7] According to [8], this enables the grid controller to keep the AC frequency within a suitable range for that system. For example, the frequency of the power systems around Europe, Asia, Africa and Australia is constantly at 50Hz, whereas in America it is 60Hz.[9] RES are decoupled from the grid and do not contribute to inertia meaning it is difficult to keep stability in terms of immediate power outages and balancing generation and demand. This problem is becoming more significant as there increase interest in utilising RES to meet energy demands. [10] The National Grid UK, proposes that by 2020, 15% of the overall energy is to be met by RES.[11] Hence the inertia of the system will further decrease and will have an impact on wind curtailment. As a result, many researchers are focusing on SUC models incorporating post-fault frequency requirement, taking into consideration fast frequency reserve requirements. [12] [2]

4.2 One Area system

Fei, Teng [4] paper proposes a MILP formulation of stochastic scheduling with inertia-dependent fast frequency requirements by taking into account the post-fault frequency requirements. The constraints that were obtained are as follows:

1. $$H \geq \left| \frac{\Delta P_L^{max}}{2RoCoF_{max}} \right|$$

2. $$|\Delta f_{nadir}| = \Delta f_{DB} + \frac{\Delta P_L^{'}}{D^{'}} + \frac{2R * H}{T_d * D^{'2}} \log\left(\frac{2R * H}{T_d * D^{'} * \Delta P_L^{'} + 2R * H} \right) \leq \Delta f_{max}$$
$$=> \frac{2k*}{T_d} \log\left(\frac{2k*}{T_d * D^{'} * \Delta P_L^{'} + 2k^*} \right)$$
$$= D^{'2}(\Delta f_{max} - \Delta f_{DB}) - D^{'} * \Delta P_L^{'}$$

   where
$$H * R \geq k^*$$

3. $$|\Delta f^{ss}| = \frac{\Delta P_L^{max} - R}{D * P^D} \leq \Delta f_{max}^{ss}$$
$$=> R \geq \Delta P_L^{max} - D * P^D * \Delta f_{max}^{ss}$$

The proposed analytical nadir frequency constraint provided a security region as there is an inverse relationship between H and PFR. This region could be used choose the appropriate values for PFR and H to be used in the optimisation problem. Furthermore, the problem can solved analytically or numerically. Solving the problem analytically shows the inverse relationship of H and R. This relationship was investigated further in Yiapatis, Nikolas [12] where the Matlab simulation for a single is compared to analytically solution. As it can be seen in Figure 2, the points lie extremely close to the analytical solution line proving that the solution proposed by Fei, Teng [4] was the optimum constraint.

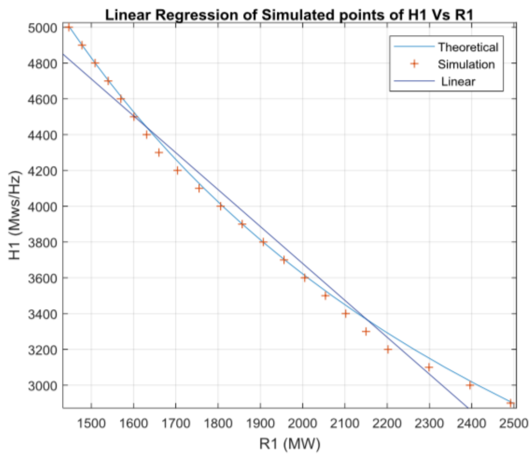

Figure 3: 400Hz input sine wave

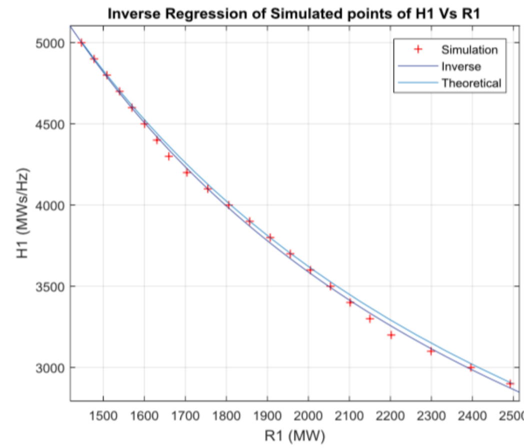

Figure 4: $3.6kHz$ input sine wave

The paper explores regression techniques, such as linear, quadratic, exponential and inverse. As expected, the inverse function was the best fitting curve as shown when the problem is solved analytically. It is ideal to increase the order of polynomial function and try different types of regression to ensure the line fits as close as possible to the operating points. Using regression techniques such as the linear regression provided a safe region, however the discrepancy introduced by having a "above-all-points" function, means that constraint will not provide a optimum solution in a optimisation problem. Therefore, a compromise between the safety and efficiency of the line must be found.

4.3 Multi-area system

The system suggested above makes the assumption that there is a uniform frequency model. However, locational frequencies exist and are significant, as some areas have greater levels of RES compared to others. This is a problem as the inertial responses between the areas will be different. [13] As a result, the frequency evolution seen by the two differential equations (1) and (2) show a complex dependency [14] which means it is not possible to obtain an analytical solution for RoCoF, frequency level at nadir and the quasi-steady-state frequency constraints.

$$2H_1\frac{d\Delta f_1(t)}{dt} + D_1 * P_1^D * \Delta f_1(t) = PFR_1(t) - \Delta P_{L1} + P_1^{transfer}(t) \tag{1}$$

$$2H_2\frac{d\Delta f_2(t)}{dt} + D_2 * P_2^D * \Delta f_2(t) = PFR_2(t) - \Delta P_{L2} + P_2^{transfer}(t) \tag{2}$$

The frequency evolution between two interconnected areas can be modelled in Simulink.[15] This model is allows us to input the parameters and numerical calculate the nadir frequency by running dynamic simulation.

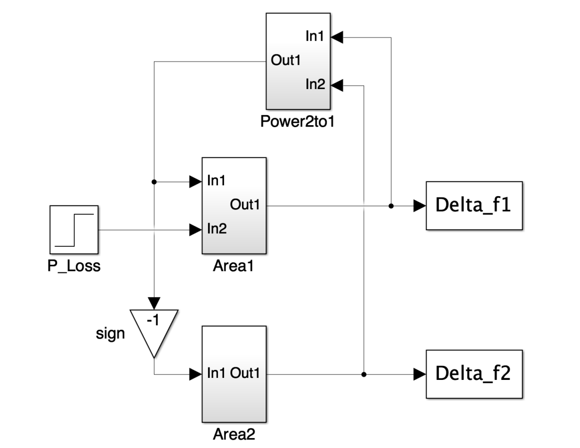

Figure 5: Simulink model for two area system

The model above is utilised by Yiapatis, Nikolas [12] to find a regression model that best fit a two area system. The Poly32 model had a goodness of fit measured by R^2 of 0.9774, which was slightly greater than the linear (0.9554) and quadratic (0.9789) and it was concluded that by increasing the order of the polynomial used, the model's accuracy increases. The relationship between the variables was investigated using a 3D plot by keeping H1 at a constant level, however it did not provide a rule that could be used in the optimisation problem, but allowed the relationship to be seen. This is because in the final model, all variables will be changing and cannot be set to a constant level.

5 Requirement Capture

The overall aim of the project is to create frequency constraint to aid a real-world power system optimisation problem. This chapter will focus on describing the necessary and desirable features that the final deliverable should include and will be used to later on to evaluate whether the constraint produced is successful.

5.1 Purpose of the Study

The aim of this project is to use data-driven techniques to derive a frequency constraint that will be able to classify a safe and unsafe area for a nadir frequency given the system inertia and primary frequency response. The frequency constraint should be able to used with an optimisation problem and have high accuracy in determining whether a point is safe or unsafe. The constraint will not take into account other variables such as the Power Loss, Dampening etc, which could impact the nadir frequency. For the scope of this project, these variables will be fixed, however in the future, the impact of these variables can be explored.

5.2 Deliverables

The project will be divided into 3 stages, with each section requiring a different deliverable. As discussed in the Interim Report, the first stage will involve obtaining a dataset for both areas. This is the most crucial part of the project as the data points will provide the basis on which the frequency constraint will classify any other data points. The simulation provided is implemented in Simulink which is a toolkit that is part of MATLAB. The process of obtaining data points was repeated again as it was found that a higher resolution dataset was needed. The results obtained for this can be found in section 8.2.

The second stage of the project will be exploring the various models that could be used to run a simulation for a constraint and unconstrained fit. In section 5, the various models that were considered are described in detail and why it was decided a regression would be best suited for the problem. This stage was relatively quick due to the nature of a linear programming problem.

The final stage of the project was dedicated to improving and refining the model. This required using random data points with the regression line and inputting the predicted points into simulation to check if the hypothesised safe region was correct. Based on how well the model performed changes were made such as returning to the first stage to improve the resolution of the data.

During the course of the project a limitation of hardware meant producing a improved constrained fit line was not possible. Given access to a system with more RAM, it would have been possible to produce a better constrained fit. However, the hardware that could be accessed using Amazon Web Services allowed a comprise to be made, where the full resolution of the data was not used and a constrained fit line using a larger dataset was obtained.

6 Analysis and Design

This chapter will focus on giving a high level overview of the decision process involved when solving the problem. It will briefly describe the other methods which were considered but not pursued to produce the frequency constraint.

6.1 Regression Techniques

Regression was chosen approach for this problem due to its simplicity and compatibility with the linear programming problem. The regression technique uses a method of least squares to obtain a line that as close as possible the provided data points.

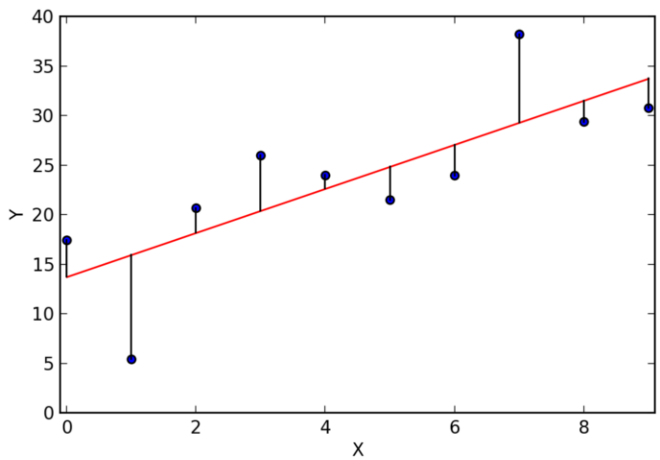

Figure 6: Diagram showing in a 2D the residual that is minimised

The line that is produced aims to minimises the distance from each point, which can be plotted as a vertical line towards the regression line. The overall fit function below shows how the value of H2 will considered as the dependent variable which is altered by the independent variables H1, PFR1 and PFR2. Inertia is chosen as the independent variable arbitrarily as one variable must be fixed in order to find the constraint. There will be two fit functions as the two areas have different nadir frequencies. The f and g function refer to the fit function for each area.

$$H2 = f(H1, PFR1, PFR2)$$
$$H2 = g(H1, PFR1, PFR2)$$

The regression technique will use data point that are in the desired region. This is beneficial as the line produced does not need a significantly large set of unsafe, which is difficult to obtain, to predict whether a point is safe. The relationship produced can easily be implemented in a linear programming problem, which is discussed in the next part. It was proposed that the regression line would have higher orders, however increasing the order of the regression had diminishing returns as the complexity of the problem increases and the significance of the each extra order decreases.

6.2 Investigating regression techniques

There are two types of regression techniques that are implemented in the project known as constrained and unconstrained. The following will summarise the difference between them and the reason why both are relevant.

6.2.1 Unconstrained Regression

H2 is a function of the other three variables, H1, PFR1 and PFR2, therefore the linear regression model can be characterised by the following function:

$$f(x, y, z) = p0 + p1 * x + p2 * y + p3 * z \tag{3}$$

where the variables H1, PFR1 and PFR2 and be defined as x, y and z respectively. [12] The unconstrained regression looks derive the coefficients of this formula using ordinary least square regression characterised by the following linear algebra problem.

$$\text{minimise } 0.5 * ||Ax - b|| * *2$$

The matrix A in the linear algebra problem will contain coefficients of this equation in the form of a Vandermonde matrix. The matrix b will be the the respective dependent variable. For a quadratic regression, the following function is observed:

$$f(x, y, z) = p0+p1*x+p2*y+p3*z+p4*x*x+p5*x*y+p6*x*z+p7*y*y+p8*y*z+p9*z*z \tag{4}$$

The goal of the using unconstrained regression is to find a line which is as close as possible to the analytical constraint as possible. This will ensure that the frequencies which are chosen based on the constraint will provide the most efficient primary frequency response for the given system inertia. The diagram below shows an example of this in a one area system, where the red line is a unconstrained regression line and the green line is the analytical solution.

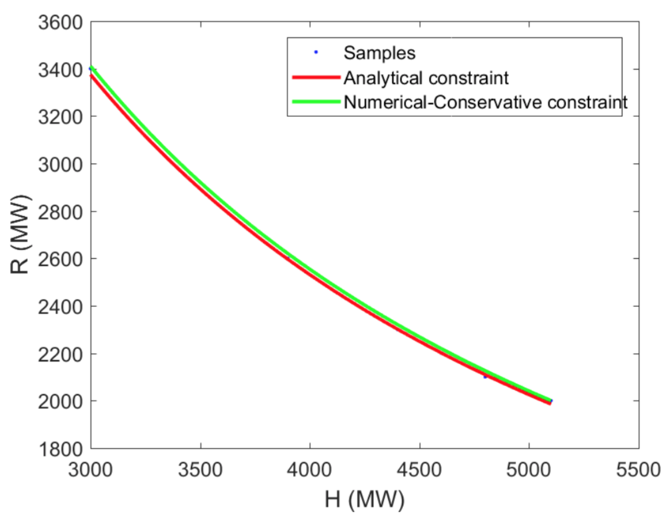

Figure 7: Unconstrained regression

6.2.2 Constrained Linear Regression

In order to obtain the constrained linear regression, the following ordinary least squares linear problem was solved:

$$\text{minimise } 0.5 * ||Ax - b|| * *2$$

$$\text{subject to } Ax >= b$$

Similar to unconstrained regression, the matrix A is a Vandermonde matrix and matrix b contains the independent variables respectively. However, the problem is subject to a constraint which ensures that the least squares fit line is above all points include worst case scenarios. This means the line is fit in such as way that there is a compromise made between minimising the the distance between the points and meeting the criteria of classifying the worst case points.

The diagram below shows how for a one area system the points for a constraint can be above the analytical constraint. Constrained regression can be used in this project to the regression line is above all the data points, therefore ensure that the frequency security works for all points. Although there are some samples within the region between the constrained regression line and analytical line, the constraint will exclude these as unsafe points.

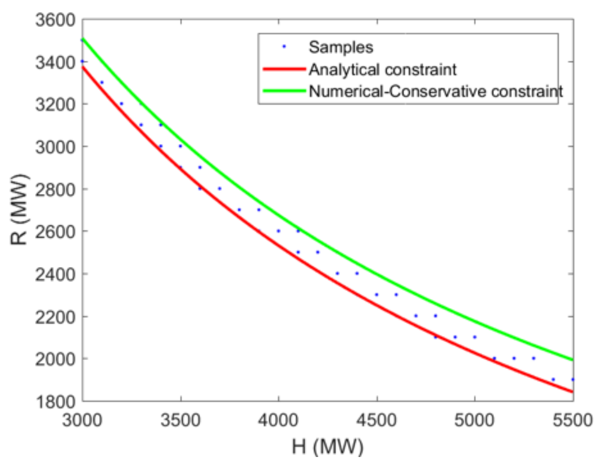

Figure 8: Constrained regression with samples between analytical and numerical conservative constraint

6.3 Decision Trees

A different supervised learning method that was considered was decision trees. As the overall goal is providing a method to classify whether a point is safe or unsafe, a decision tree could have been used to solve the problem. The algorithm would combine a statistical test known as Information Gain with an ID3 algorithm. [16] The statistical test looks at the entropy of the dataset for a given split point. The algorithm is based on the idea of decreasing entropy therefore, the target that returns the highest entropy which is used as a splitting point. The algorithm starts from the root node and works down until a leaf node is reached where it terminates.

Equation 5 is used for the statistical test:

$$G(q) = H(dataset) - (\frac{|subsetA|}{|dataset|} H(subsetA) + \frac{|subsetB|}{|dataset|} H(subsetB)) \tag{5}$$

$$|dataset| = |subsetA| + |subsetB| \tag{6}$$

The steps that are followed to produce a decision tree are as follows:

1. **Finding a split point-** using a statistical test for each attribute, the performance of how well each attribute classifies the training examples is considered
2. **Splitting the dataset-** the dataset is split into two subset using this chosen split point
3. **Iterate until a leaf is reached-** the first and second step are repeated until a leaf node is reached

However, this method was not used in the final implementation as it would require a large dataset to work. This is because the statistical test would need to separate unsafe and safe data points, meaning a complete dataset with data points from both regions is required. Providing such a large dataset is not a viable or efficient method of producing a frequency constraint.

6.4 Neural Networks

Neural networks are a popular method used to classify data. The neural network in this context would use supervised learning, by splitting the dataset into test and training data to learn which points are safe and unsafe. The training data would be chosen using random samples and the test data would provide a accuracy test to evaluate the performance of the network.

If this method were to be used, the neural network can be made using a high level library in python known as *Keras*, which would form a neural network based on the hyperparameters given to optimise the performance of the network. The model would be tested and modified until a working production model obtained. Due to the number of samples that can be obtained, it would use a method known as parameter tuning to improve the accuracy for the model.

This approach was not implemented as the neural network cannot be converted into a form that is compatible with a linear programming problem. Whilst it achieve the objective of being able to classify the data point, it does not provide a relationship between the inertia and primary frequency response that can be used by the final problem.

7 Implementation Plan

This chapter will cover in detail how regression was used to solve the problem. In particular, the method used to obtain the operating points, how the testing stage was completed in MATLAB, and how the the fit functions were obtained. In this project, the constrained model was obtained using MATLAB function called *lsqlinear*, while the unconstrained fit function was calculated using the Python library *sklearn*.

7.1 Obtaining operating points

7.1.1 Algorithm

The first stage of this project involved simulating the electrical system using a MATLAB tool called Simulink. It is designed to aid the design and simulation of systems before implementation. In this context, the Simulink model produces the frequency evolution of for Area 1 and 2 based on the given input variables. The *min* function is used on the variable *Delta_f1* and *Delta_f2* seen in Figure 9, in order obtain the nadir frequency for each of the areas. The main objectives when obtaining the operating points were to ensure that the points were precise, within the desired nadir frequency region between -0.795 and -0.800, and required minimal computational to be obtained, so that large numbers could be produced to improve the accuracy of the model.

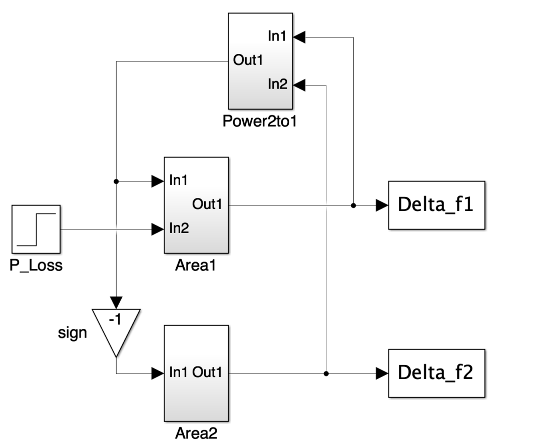

Figure 9: Simulink model for two area system

Initially, double nested for loops were used where the ranges for all four values and were changed in a step size of 500 and to checked to see if the point lied within the desired region.

```
xvals= 500:500:5000; %array for the H1 values from 500-5000 in step size of 500
yvals= 500:500:5000; %array for the H2 values
PFR1vals= 1000:500:5000;%array for the PFR1 values
PFR2vals= 1000:500:5000;%array for the PFR2 values
```

Listing 1: Setting the range of values to vary variables

Based on this initial test, it was clear it would be difficult to obtain a significant number of point using step size of 500 as the nadir frequency varied significantly, however this provided an idea of the ranges for the values of H1, PFR1 and PFR2 that would provide an viable sample. Therefore, the second iteration of the method used a range of 500-5000 for H1 values, 1000-5000 for PFR1 and PFR2 variables respectively.

```
for PFR2index=1:len_PFR2
    PFR2=PFR2vals(PFR2index);
    for PFR1index= 1:len_PFR1
        PFR1=PFR1vals(PFR1index);
            for xindex=1:len_xvals
                H1=xvals(xindex);
                H2=100;
                sim('TwoArea_fromWorkspace_MaxRamp')  %runs the simulink model
                k=min(Delta_f1);
                if k<-0.8
                    while k<-0.8
                        H2=H2+50;
                     sim('TwoArea_fromWorkspace_MaxRamp')
                        k=min(Delta_f1);
                    end
                else
                    p=p+1;
                    r=(p/1000);
                    waitbar(r,h, p);
                    break
                end
                p=p+1;
                row=[H1, H2, PFR1, PFR2, k];
                j=i-1;
               M(i,:)=row;
...
```

Listing 2: Iterator to find the optimum H2 value

Furthermore, it was decided that one of the variables should vary in smaller step sizes using a while loop to check if the limit had been reached. H1, PFR1 and PFR2 values would be set to a set value and the while loop would gradually increase the system inertia for area 2 (H2) until a nadir frequency close to -0.8 was found. Listing 2 shows how the while loop was implemented with an if statement to check if the nadir frequency was already in range. The H2 value started from a low value to ensure that nadir frequency was lower than -0.8 and was increased, until nadir frequency was in this range.

```
if k<-0.8
    while k<-0.8
        if k<-0.87
            H2=H2+200; %large step sizes are taken
        elseif k<-0.81
            H2=H2+20; %smaller steps are taken as the nadir frequency is approached
        else
            H2=H2+5; %the H2 values are incremented in much smaller steps
        end
    sim('TwoArea_fromWorkspace_MaxRamp')
    k=min(Delta_f1);
    end
else
```

Listing 3: Optimisation to improve speed of algorithm

To improve the speed, a optimisation was made where the current nadir frequency was checked for closeness to the desired -0.8 nadir frequency, at which point smaller increments of 20 and then 5 would be taken. This allowed the accuracy of the H2 value to be increased allowing for a better fit due to more precise points and the speed of convergence to increase.

7.1.2 Datsets

The first dataset obtained using this method had a step size of 50 for H1, with a step size of 250 being taken for PFR1 and PFR2. However, having obtained the regression line, it was evaluated that these step size were too large as the constrained fit did not produce a high accuracy value. As a result, the program was run again with smaller step size of 50 for H1, PFR1 and PFR2. This increased the computational time taken by 25 times, meaning it would take over 4-5 days to complete given the right space to store the dataset. Therefore, a cluster was created on AWS and the each machine was tasked to compile a specified range of the full set, which were concatenated after all machines finished. In the first dataset with a large step size, around 6,000 to 7,000 data points were observed, however with the smaller step size of 50, this was increased to 125,000 to 130,000 data points for both areas.

Furthermore, as results were taken, it was observed that some values for the nadir frequency were not with the desired region of -0.795 and -0.8. Therefore, these results were discarded using excel. This occurred as the while loop simply increase the H2 value based on the previously observed nadir frequency. It was chosen using trial and error, -0.87 and -0.81 would be threshold to change the step size. Therefore, naturally some values decreased at a faster rate therefore were not in the region needed.

7.2 Regression

7.2.1 Unconstrained Regression

To obtain the fit function for an unconstrained regression, the values obtained in *Matlab* were saved in CSV format and imported into python. In order to assess the performance of the model in the testing stage, only 90% of the data is used for training.

```python
import pandas as pd
import numpy as np
from sklearn import linear_model
from sklearn.preprocessing import PolynomialFeatures
from sklearn.linear_model import LinearRegression
from sklearn.metrics import accuracy_score
from sklearn.metrics import r2_score

```

Listing 4: Libraries used to produce the unconstrained regression

The libraries used for the unconstrained regression are as follows:

1. *sklearn* library was used to implement a linear regression using ordinary least squares
2. *Polynomial Features* function was used for preprocessing to transform the data into format that included a column of 1s as in the Vandermonde matrix to solve linear algebra problems
3. *score* function is used to obtain the R2 score of the fit produced and the coefficients and intercept of the function can be obtained

```
linearreg_2.fit(X_Poly,Y)

```

Listing 5: Fit function

The dataset was split into labels and features and the fit function from the linear model was used to produce a linear regression based on the matrix created by Polynomial features function.

```
poly_reg=PolynomialFeatures(degree=1)
X_Poly=poly_reg.fit_transform(X)

```

Listing 6: Function used to obtain the Vandermonde matrix

The *Polynomial Features* function has an argument for the degree, which can be used to produce a matrix to obtain the quadratic linear regression. Using the *predict* function the first part of the testing phase is completed, where the predicted value of H2 is obtained for given test values.

7.2.2 Constrained Linear Regression

In order to obtain the constrained linear regression, the MATLAB function *lsqlin* was used. The *lsqlin* function takes arguments A , b and C and d and can solve an equation for the following equation:

$$\text{minimise } 0.5 * ||Ax - b|| * *2$$

$$\text{subject to } Cx <= d$$

Therefore, the code implementation required the variables C and d to be negative, meaning the inequality sign for the constraint would be greater than rather than less than d.

```
d = M(:,1);
C = M(:,2:5);
b = d.*-1;
A = C.*-1;
x = lsqlin(C,d,A,b);
```

Listing 7: Constrained regression code

The solution provided by the function were the coefficients of the function which were used with the testing set to predict the value of H2.

8 Testing

The testing phase is crucial to ensuring that the constraint produced meets the criteria of being reliable frequency constraint. The fit function provides a value based on three independent variables, however to check how close the line is to the true frequency constraint, only points that provide a nadir value in the desired region should be used. Initially it was assumed that an array of any value for H1, PFR1 and PFR2 could be used to obtain the value for H2, which would represent a point in the nadir frequency range between -0.795 and -0.800. However, a significant number of H2 were returned as negative values showing that the equation was providing the lowest value which would allow the set of values to be within in the desired nadir region.

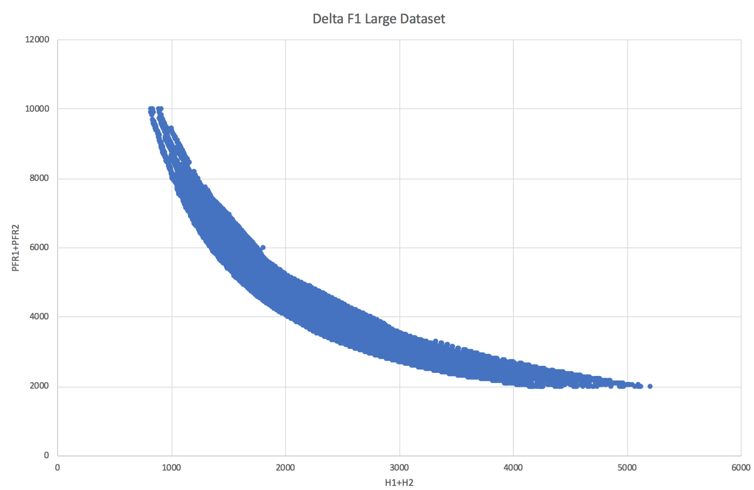

Figure 10: Delta F1 large dataset with PFR1+PFR2 with respect to H1+H2

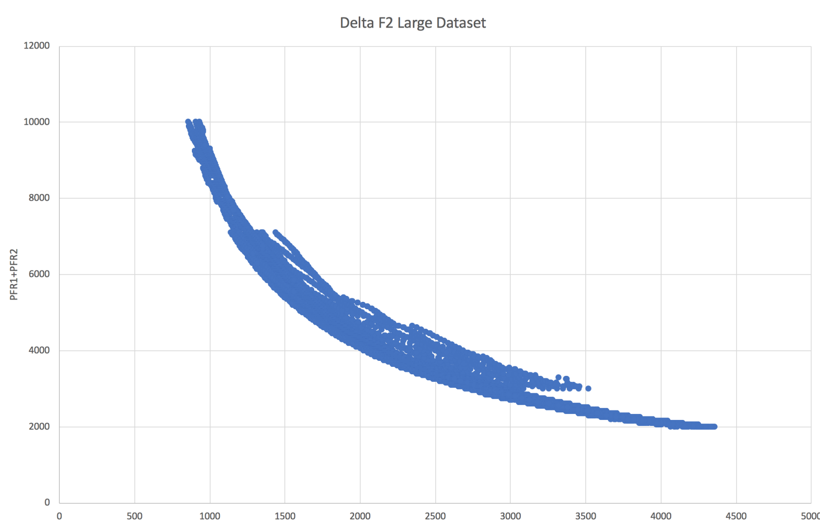

Figure 11: Delta F2 large dataset with PFR1+PFR2 with respect to H1+H2

The Figures 10 and 11 illustrate how the combined H1 and H2 value vary with respect to the combined PFR1 and PFR2 value. As it can be seen, there is a distinct region where these points lie, therefore, only points in the nadir region calculated by the algorithm can be used. This is in contrast to one area system, where any random variable should return a value on the line, the complex dependency between the 4 variables, mean it is not as simple as using random variables. This explains why many values were negative, given three large dependent variables, H2 will need to be negative to lie within the area where most points reside.

Therefore, before the model is trained, as previously mentioned, 10% of the variables are saved to be used for the training set. This dataset was chosen randomly using Excel and were used to predict the H2 value. The accuracy of the fit function was tested by running the four variables, the predicted H2 variable with independent H1,PFR1 and PFR2 variables, in Simulink Matlab model to obtain the nadir frequency.

```
while k<10001
    k=k+1;
    try
        H1=fulltestresults(i,2);
        H2=fulltestresults(i,1);
        PFR1=fulltestresults(i,3);
        PFR2=fulltestresults(i,4);
    catch
        break
    end
    i=i+1
    sim('TwoArea_fromWorkspace_MaxRamp')  %runs the simulink model
    k=min(Delta_f1)
    row=[H1, H2, PFR1, PFR2, k];
    M(i,:)=row;
end
```

Listing 8: Running each value and catching errors

This test would definitively check whether the value predicted was in the safe region as required. For some given random variables, the H2 value predicted was negative. These values were discarded as at negative number, there is no control for the derivative of the equation below. This means the system is infinitely volatile and will not produce a nadir frequency will be safe. The next section will discuss the exact metric which were obtained and how they were used.

9 Results

This chapter will focus on covering the results obtained by running random variable test and the other metrics used to test the frequency constraint.

9.1 Metrics

There are three types of metrics that can be used to test the success of a regression model. The first metric involves the R2 score which determines the goodness of fit for a model. The goodness of fit will identify the variation of the dependent variable (H2) from the independent variables (H1, PFR1 and PFR2). Whilst this metric is useful, it does not provide us a full indication of how well the frequency constraint can identify the safe and unsafe region. For example, for an unconstrained regression line, the points might be skewed generally lower than -0.8 but still obtain a low R2 score.

Another metric that be used to check the accuracy is testing the predicted dependent variable and using the values for H1, H2, PFR1 and PFR2 to obtain the nadir frequency. This will in theory provide a real world simulation of how well the constraint works. The test set mentioned in the previous section can be used to obtain this value. The nadir frequency can be checked for safeness and closeness to the nadir frequency of -0.8, providing an indication of how well the line works.

The final method of determining how well the the plan fits is using 3D plots of the data-points to provide a clear picture of the impact of changing the H1 on the levels of H2,PFR1 and PFR2 needed. The visual representation will provide an indication of which values the plane is working for and which fails the plane is failing. To check conservative and safe the lines are, the spread of the of the nadir frequencies can be visualised using a histogram.

9.2 Datasets

Initially the unconstrained linear regression was completed on a smaller dataset. The figure below provides a summary of the difference between the spread of H1,PFR1 and PFR2 values with the smaller and larger dataset.

Dataset	Area	H1	Step Size	PFR1	Step Size	PFR2	Step Size	Number obtained
Small	Delta F1	100-3000	50	1000-5000	250	1000-5000	250	6876
	Delta F2	100-3900	50	1000-5000	250	1000-5000	250	6646
Large	Delta F1	100-4700	50	1000-5000	50	1000-5000	50	112808
	Delta F2	100-4000	50	1000-5000	50	1000-5000	50	117411

The number of values in the dataset were increased in order to improve the accuracy as when the random test values were run on the smaller dataset, a significant number of values for the constrained linear regression did not have a value larger than 0, meaning it had already failed. Furthermore, in an ideal situation, the constrained regression would have all values in the safe region. To illustrate visually how the impact of increase H2 with respect to H1, PFR1 and PFR2, the value of independent variable were fixed while H2 was increased by arbitrary step size 500 for both areas. The larger dataset were used to provide the most precise image of the change.

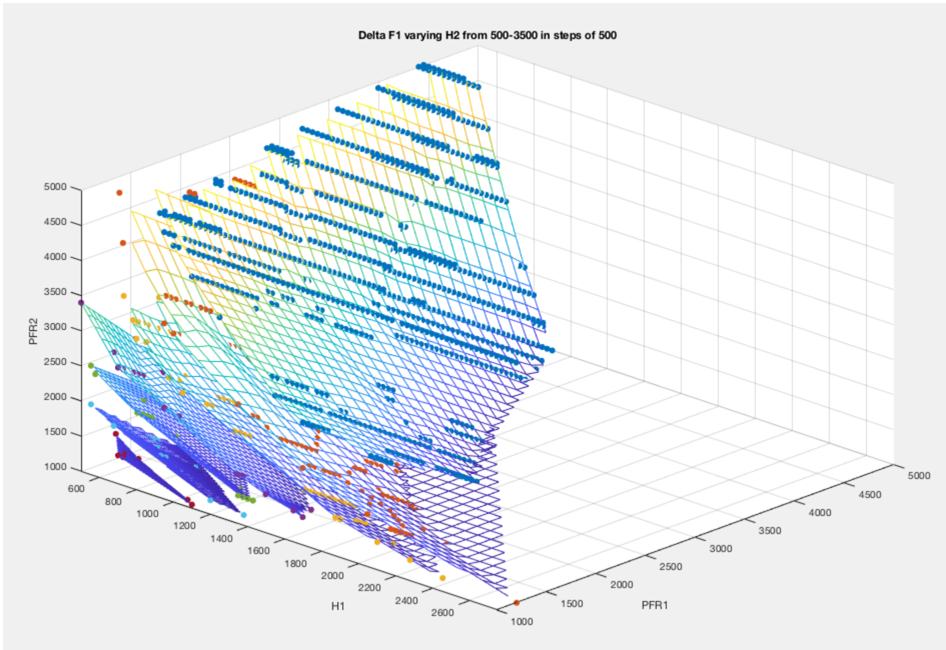

Figure 12: Delta F1 large dataset: H2 is fixed at different values from 500-3500, with 500 being furthest right and 3500 furthest left

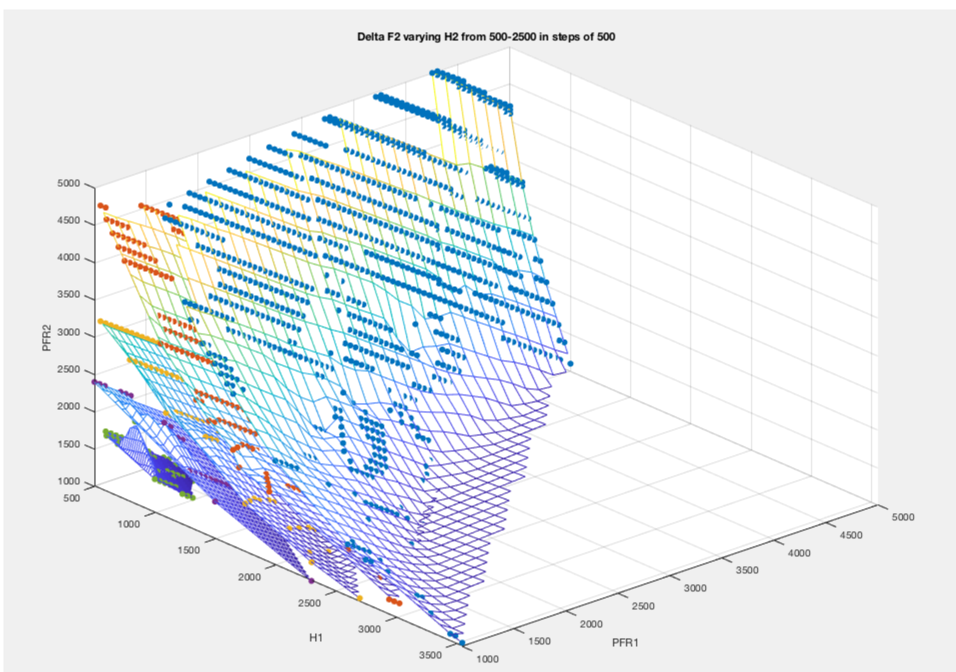

Figure 13: Delta F2 large dataset: H2 is fixed at different values from 500-3500, with 500 being furthest right and 2500 furthest left

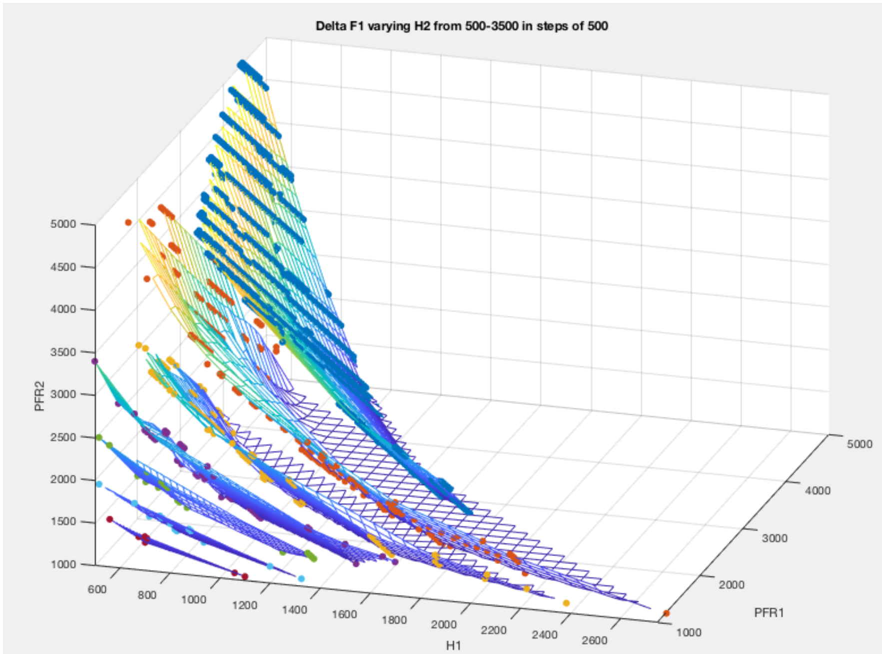

Figure 14: Delta F1 large dataset: H2 is fixed at different values from 500-3500, with 500 being furthest right and 3500 furthest left

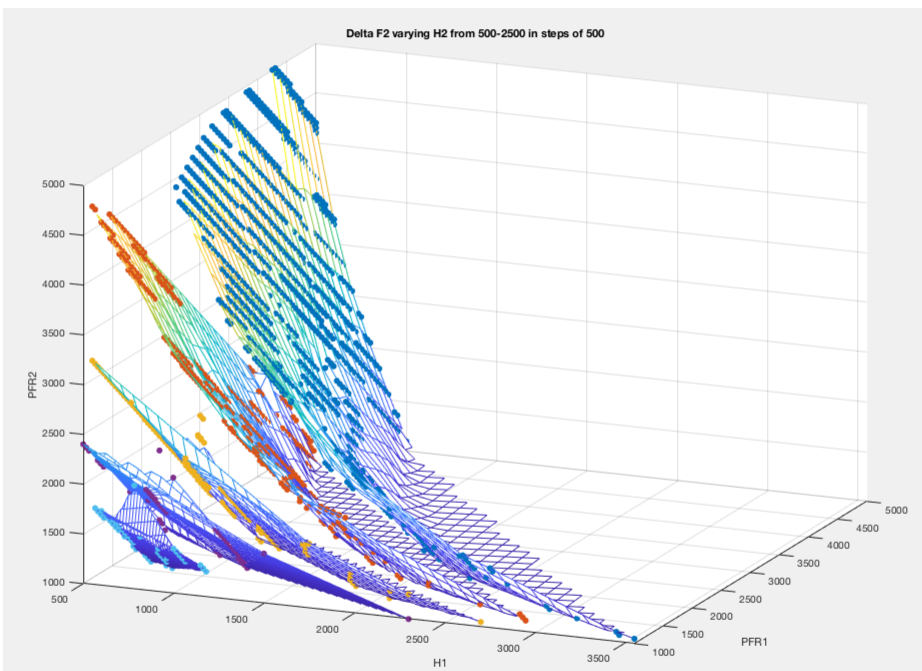

Figure 15: Delta F1 large dataset: H2 is fixed at different values from 500-3500, with 500 being furthest right and 2500 furthest left

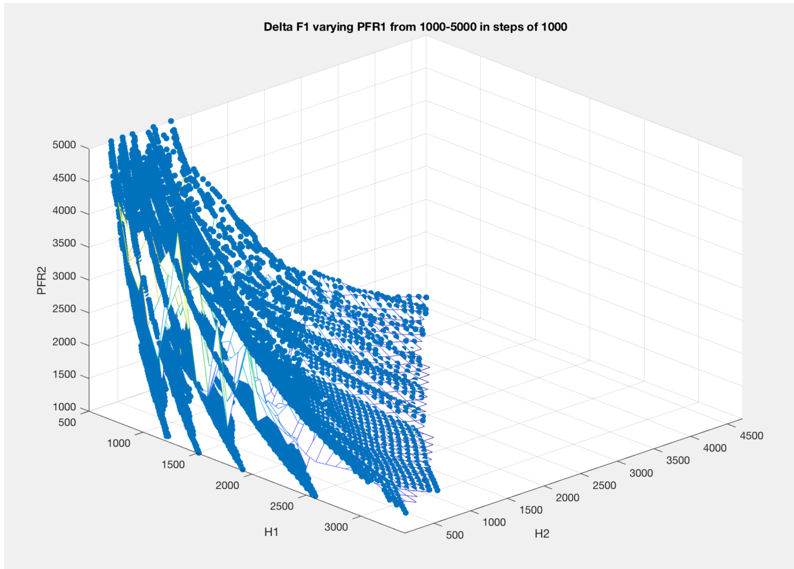

Figure 16: Delta F1 large dataset: PFR1 is fixed at different values from 1000-5000 in steps of 1000, with 1000 being furthest right and 5000 furthest left

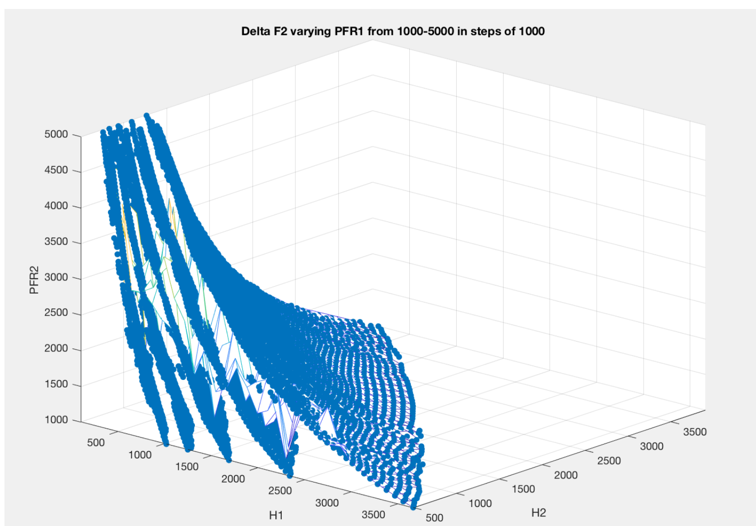

Figure 17: Delta F2 large dataset: PFR1 is fixed at different values from 1000-5000 in steps of 1000, with 1000 being furthest right and 5000 furthest left

The figures 12-15 above show that as H2 increases the values of primary frequency response of both areas and the system inertia of area 1 decreases. The relationship is not completely linear as with H2 at 500, both areas have a concave surface. This indicates as one variable tends to 0, the other variables must compensate more compared to when there is a equal balance of all three variables. The model is seen to be more volatile as the value of H2 is larger as less data points are available to observe the characteristics of the relationship. In particular, when H2 equals 500 in area 2, there is a large peak which is also seen the the surface for H2 equal 1000 for fixed values of PFR1. As PFR1 is decreased, greater values for H1,H2 and PFR2 are needed, therefore the plane moves outward showing more resources being used.

9.3 Unconstrained Regression

Unconstrained regression was run on for both areas using linear and quadratic regression, with a large and small dataset. The coefficients for these models are seen in the following tables.

Delta F1	**Small**	**Large**
Coefficient	Value	Value
P0	4646.1501	4485.7846
P1	-1.4107	-1.2538
P2	-0.4330	-0.4594
P3	-0.4144	-0.4315

Delta F2	**Small**	**Large**
Coefficient	Value	Value
P0	3611.3461	3684.3892
P1	-0.8656	-0.9160
P2	-0.3523	-0.3635
P3	-0.3569	-0.3706

Delta F1	**Small**	**Large**
Coefficient	Value	Value
P000	6617.3233	7499.1317
P100	-1.4467	-2.1807
P010	-1.1863	-1.4689
P001	1.1502	-1.3870
P200	1.34E-04	2.36E-04
P110	-8.73E-05	4.48E-05
P101	-9.61E-05	2.38E-05
P020	7.78E-05	9.79E-05
P011	1.33E-04	1.61E-04
P002	7.45E-05	8.83E-05

Delta F2	**Small**	**Large**
Coefficient	Value	Value
P000	5283.5559	5125.0903
P100	-0.5479	-0.5410
P010	-1.0580	-0.9707
P001	-1.0461	-0.9683
P200	-6.77E-05	-4.76E-05
P110	-6.18E-05	-9.51E-05
P101	-7.05E-05	1.05E-04
P020	7.42E-05	6.87E-05
P011	1.22E-04	1.15E-04
P002	7.22E-05	6.86E-05

The coefficient labels for linear and quadratic refer to equation method in Section 5. The linear equation is as follows:

$$f(x, y, z) = p1 + p2 * x + p3 * y + p4 * z \tag{7}$$

For the quadratic regression, the coefficient refer to multiplication ratio of each variable.

$$f(x, y, z) = p000+p100*x+p010*y+p001*z+p110*x*y+p011*y*z+p101*x*z+p200*x*x+p020*y*y+p002*z*z \tag{8}$$

In these equations, x refers to the variable H1, y refers to the variable PFR1 and z refers to PFR2. The coefficients P1, P2 and P3 for the linear regression in both areas are negative, apart from the intercept, indicating how there is negative relationship between the dependent and independent variables. This suggests that the greater the independent variables, the smaller the value of H2 will be. This is the same relationship seen the 3D plots of with fixed values of H2.

Unconstrained		**Linear**	**Quadratic**	**Closeness Test**	
Delta f1	**Small**	0.6020	0.5248	-0.0740	-0.0394
	Large	0.6110	0.5376	-0.0725	-0.0413
Delta f2	**Small**	0.4450	0.3721	-0.1007	-0.0673
	Large	0.5647	0.5563	-0.0844	-0.0460

The accuracy test above shows that utilising a larger dataset, which contained approximately 115,000 sample points in each area, compared smaller dataset of 6,000 variables led to an improvement in the accuracy. This is especially true for Area 2 where the both the closeness and accuracy improved. The closeness test looked at the deviation from the desired -0.8 nadir frequency. Using a higher order quadratic equation decreased the accuracy of the line however, the points that were in the region were closer to the line as deviation is seen to halve for all model. The R2 scores support this as with quadratic equation, for both of the large dataset, a improvement can be seen in the score with quadratic equation.

R2		**Linear**	**Quadratic**	**Difference**
Delta f1	Small	0.8313	0.8261	-0.0052
	Large	0.9753	0.9794	0.0040
Delta f2	Small	0.8279	0.8264	-0.0015
	Large	0.9686	0.9705	0.0019

The histogram below compare the spread for the points of Delta F1 and Delta F2 in linear and quadratic regression using the large dataset. On the y axis, there are bins which denote the nadir frequency for each of sample sets. The smaller dataset is not as important, as the larger dataset includes the smaller dataset values and the accuracy is shown to better with the larger dataset. Figure 16 reiterates the results seen in accuracy and closeness test, as almost 50% of the data is below the require -0.8 nadir frequency, despite having a significant number of values above the region. This is the same for Figure 18, which shows an almost equal distribution across all bins, showing there is a compromise between accuracy and closeness for both area when using a unconstrained linear regression. Figure 17 is the quadratic regression which has a lower number values above -0.8, however most of these values are within -0.25 deviation. This is significant as it shows that the model will provide values close to the required -0.8 value. However, the compromise in accuracy for both curve makes it difficult to use as constraint that will reliably predict if a point is safe or not.

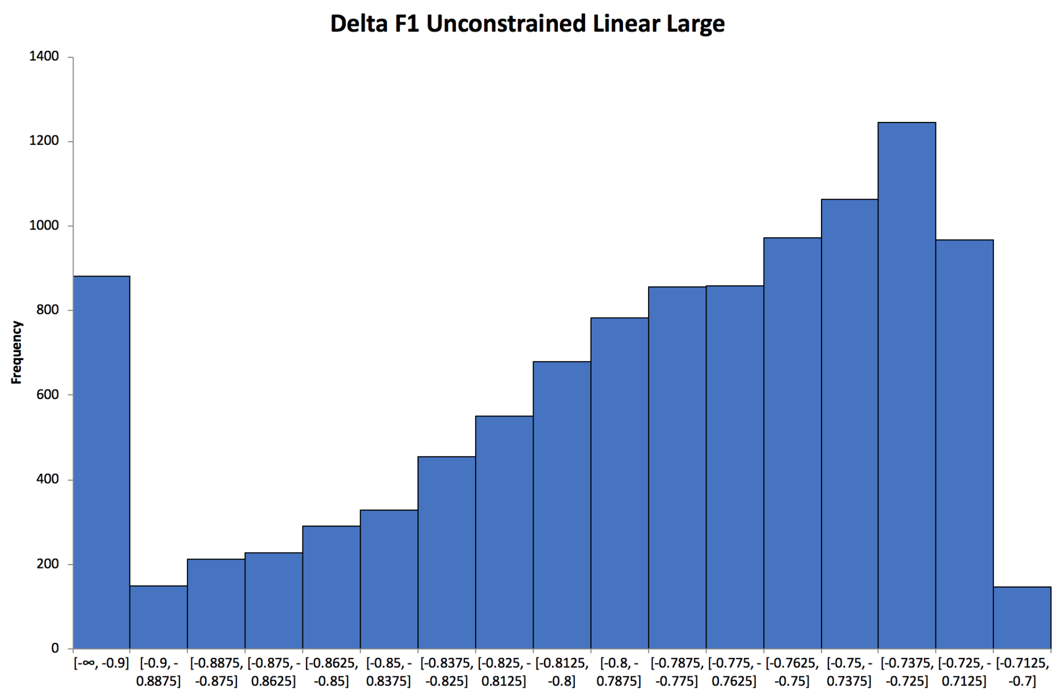

Figure 18: Histogram: nadir frequency of an unconstrained linear regression in Delta F1

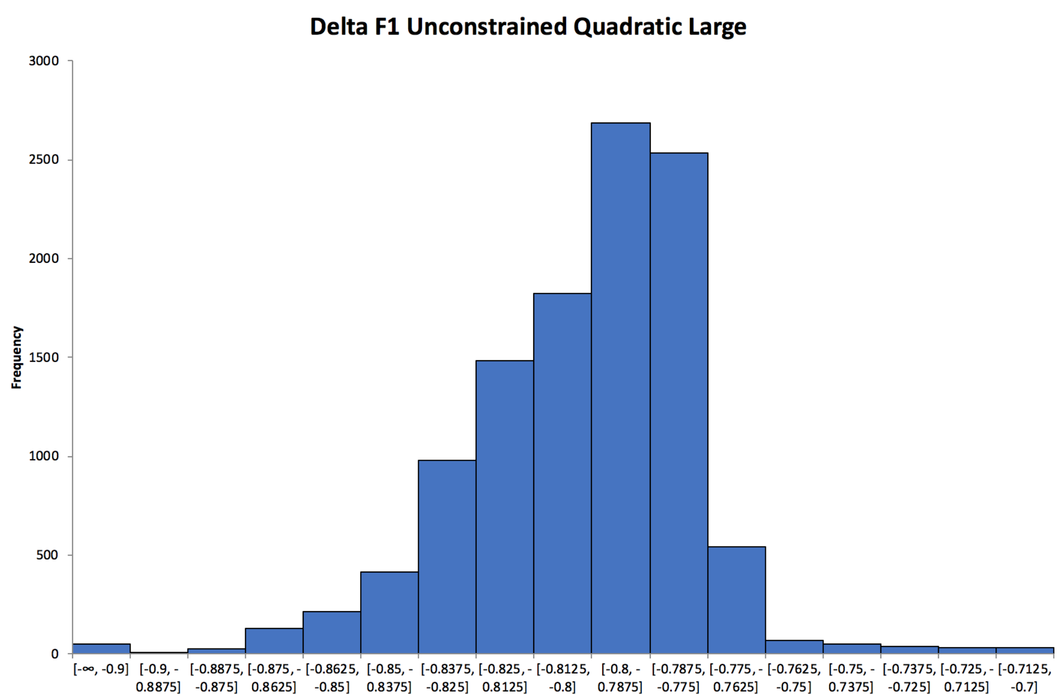

Figure 19: Histogram: nadir frequency of an unconstrained quadratic regression in Delta F1

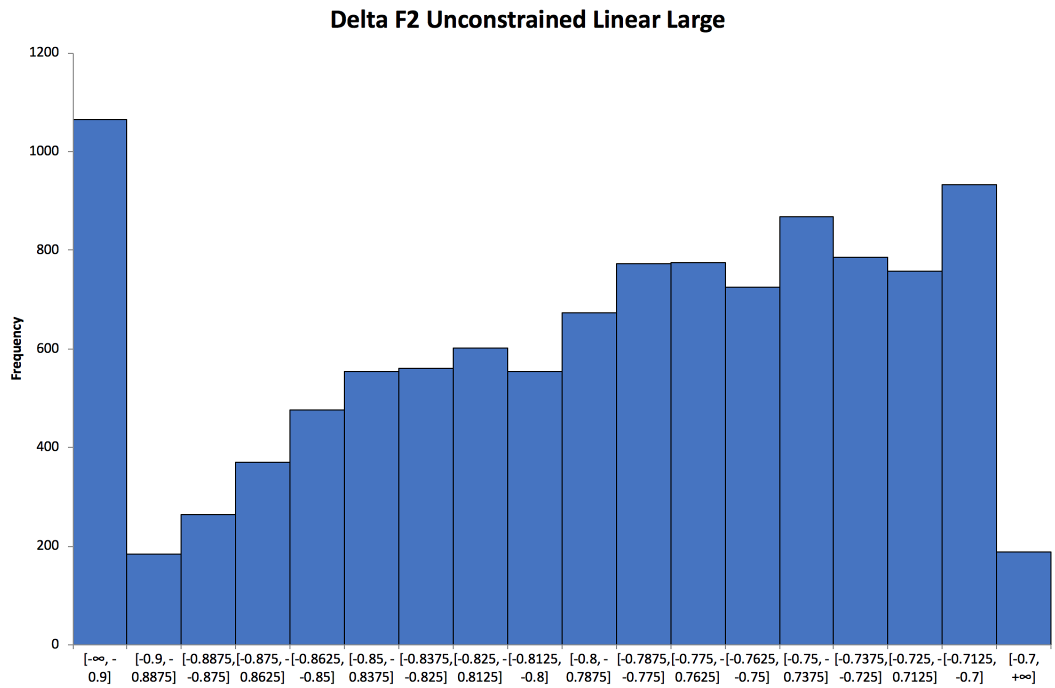

Figure 20: Histogram: nadir frequency of an unconstrained linear regression in Delta F2

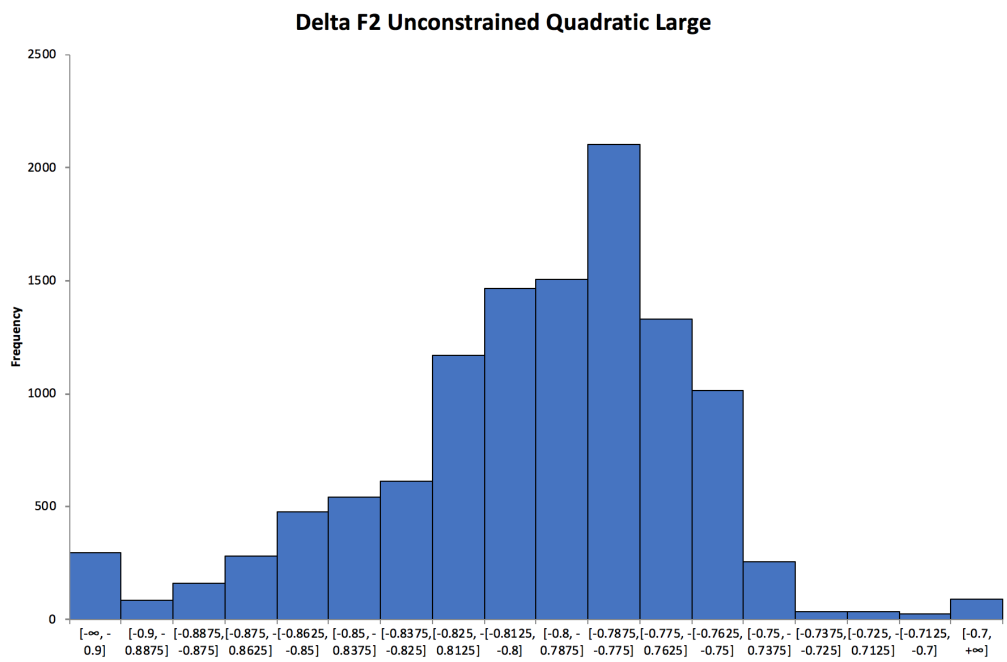

Figure 21: Histogram: nadir frequency of an unconstrained quadratic regression in Delta F2

The figures below shows Delta F1 fitting of using a quadratic regression and large dataset. The H2 value is varied from 500-2000 in step sizes of 500 including 100. As it can be seen, the plane follows a similar pattern to the larger dataset, justifying it's high R2 value. Delta F1 quadratic large model is chosen to demonstrate this as it has the largest R2 score. The values chosen for the fixed variables have are within 5 units within the actual number.

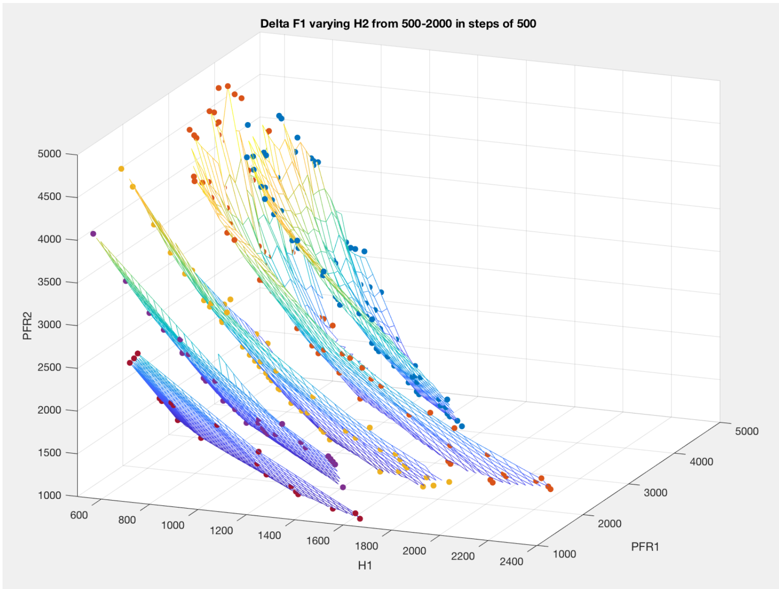

Figure 22: Far Left is H2=500, Far right is H2=2000

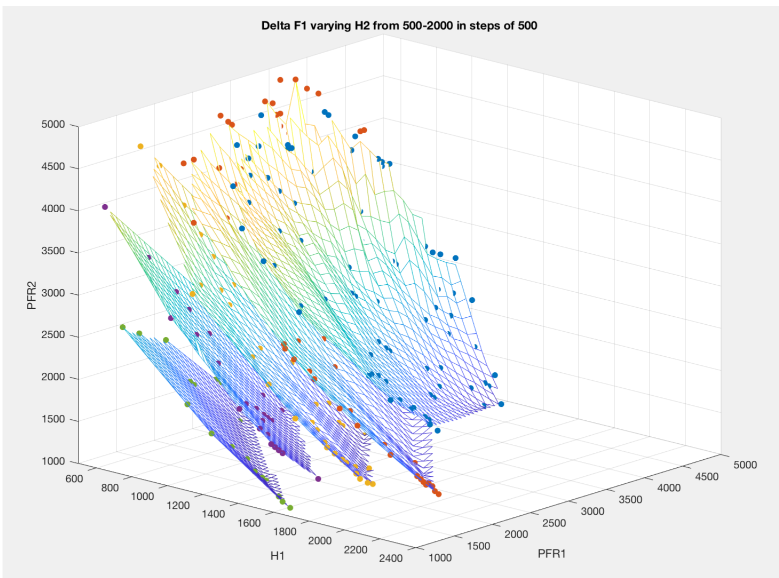

Figure 23: Far Left is H2=500, Far right is H2=2000

9.4 Constrained Regression

Similar to the previous regression, the coefficients follow equations 7 and 8. The constrained regression shows a similar negative relationship for the three independent variables, with a higher intercept to ensure that it is in the safe region.

Delta F1	**Small**	**Large**
Coefficient	Value	Value
P0	6336.9500	6483.2250
P1	-1.4220	-1.4220
P2	-0.5123	-0.6569
P3	-0.4137	-0.4154

Delta F2	**Small**	**Large**
Coefficient	Value	Value
P0	4941.9318	5104.3750
P1	-0.8545	-0.9500
P2	-0.4099	-0.4187
P3	-0.4006	-0.4207

Delta F1	**Small**	**Large**
Coefficient	Value	Value
P000	8479.5146	8837.8692
P100	-1.6787	-1.7711
P010	-1.6992	-2.0557
P001	-1.6244	-1.7231
P200	9.80E-05	9.48E-05
P110	-3.95E-05	5.53E-05
P101	-5.71E-05	-4.51E-05
P020	1.13E-04	1.59E-04
P011	2.17E-04	2.23E-04
P002	1.04E-04	1.33E-04

Delta F2	**Small**	**Large**
Coefficient	Value	Value
P000	6503.6065	7167.7455
P100	-0.5926	-1.1112
P010	-1.4190	-1.4994
P001	-1.4350	-1.4018
P200	-7.47E-05	5.15E-05
P110	-4.16E-05	-4.59E-05
P101	-5.23E-05	-9.91E-05
P020	9.69E-05	1.03E-04
P011	1.96E-04	1.98E-04
P002	9.99E-05	9.56E-05

The large dataset used for the constrained regression consisted of around 30,000 samples for both areas. Due to hardware limitation, a constrained regression using the full large dataset could not be completed, which is discussed further in section 9. The accuracy for all the test were above 0.97 except for Delta F2 small dataset with a quadratic constrained regression. This value could have been an anomaly since the small dataset used a random selection of variables and the test set was relatively small, consisting of 600 values. This value was not significant as the analysis will be completed on the large dataset which has over 0.97 accuracy.

Constrained		**Linear**	**Quadratic**	**Closeness Test**	
Delta f1	Small	0.9956	0.9971	-0.2454	-0.0730
	Large	0.9947	0.9898	-0.2524	-0.1107
Delta f2	Small	0.9596	0.6320	-0.1848	-0.0875
	Large	0.9776	0.9711	-0.2315	-0.0914

A similar pattern is seen with results of the unconstrained regression as the quadratic regression reduces the accuracy but improves the closeness test as the deviation is smaller. The decrease in accuracy is much less significant now as the constrained regression objective is to ensure that all points are above all points. Figure 24 and Figure 25 show a clear improvement in terms of insurance when using a quadratic regression opposed to a linear regression. The closeness is shown to decrease by around -0.04 which is seen in the distribution of values in the figure. The spread of the values are similar with linear regression being positively skewed, while the quadratic regression is negatively skewed. The skew is beneficial as it ensures the points are close to the required nadir frequency. Delta F2 shows that the linear distribution is far more conservative as there is a greater deviation of about -0.1. The constrained quadratic regression looks similar to the constrained regression of Delta F1 quadratic regression, with most points above the required nadir frequency.

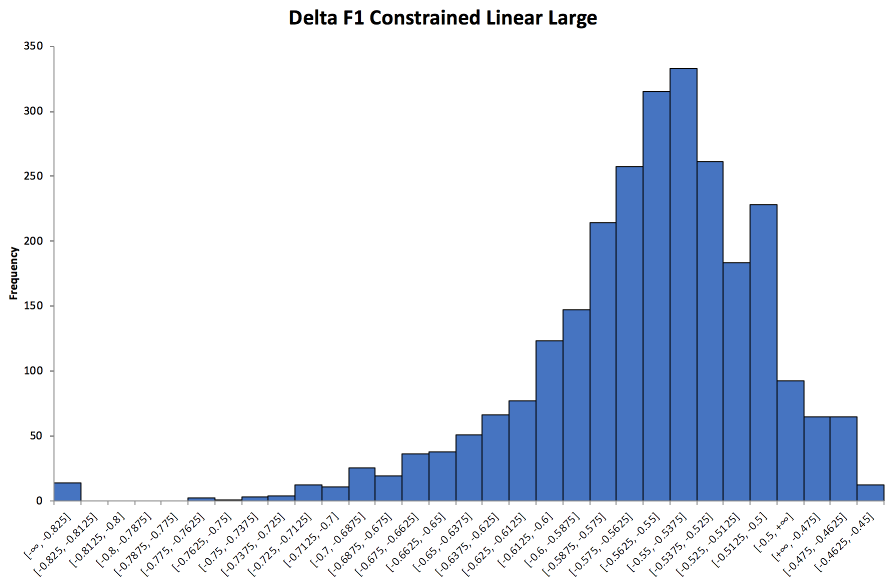

Figure 24: Histogram of nadir frequency for linear regression for Delta F1

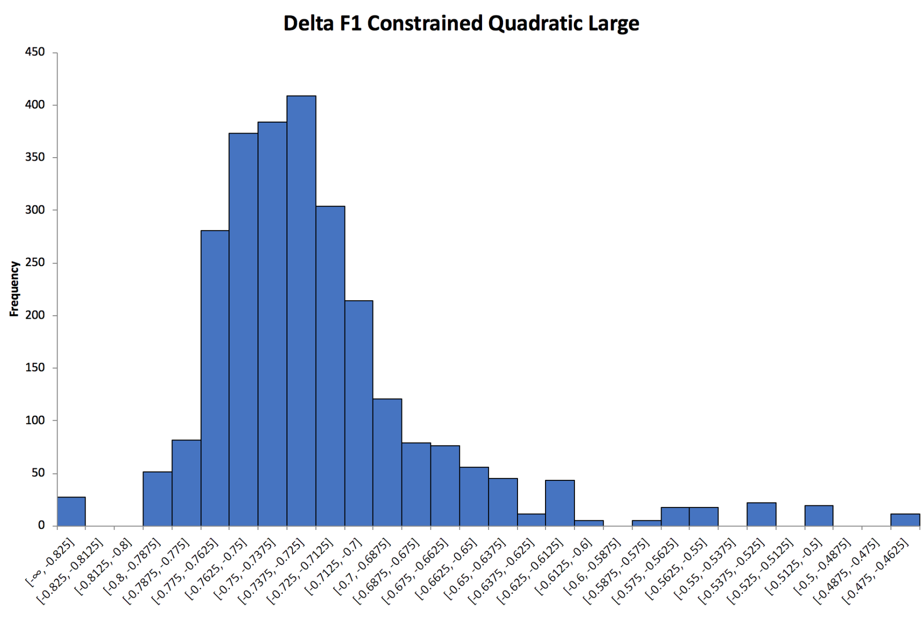

Figure 25: Histogram: nadir frequency of an constrained quadratic regression in Delta F1

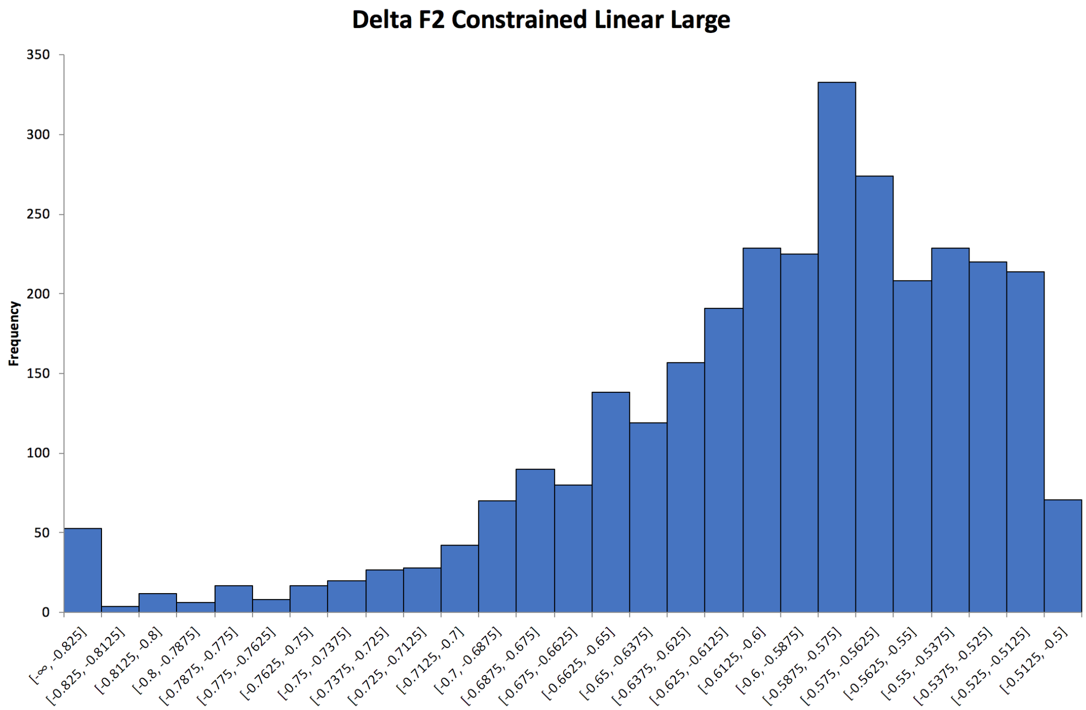

Figure 26: Histogram: nadir frequency of an constrained linear regression in Delta F2

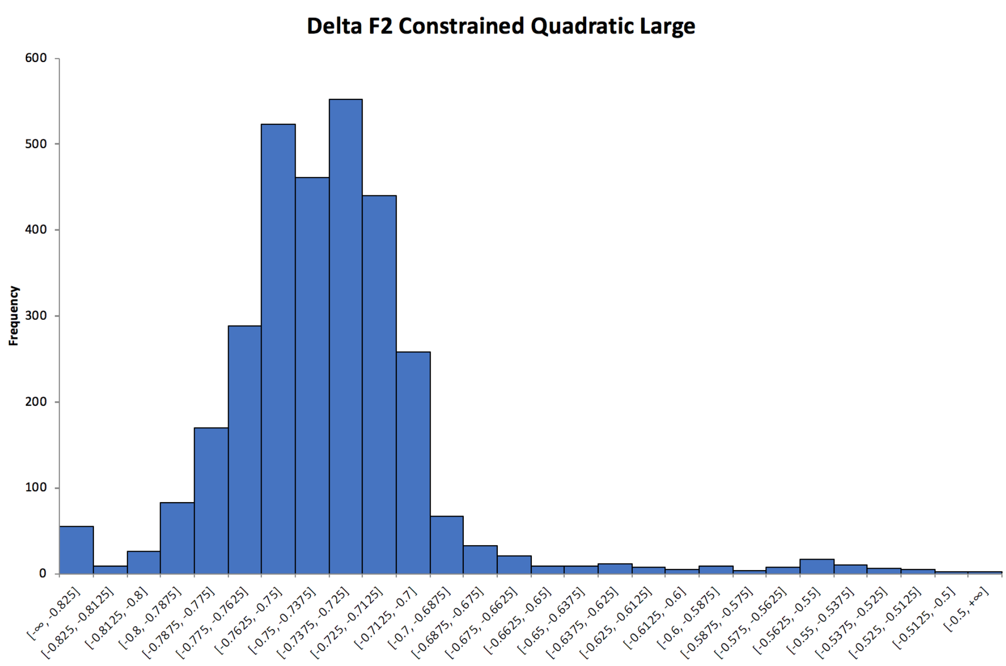

Figure 27: Histogram: nadir frequency of an constrained quadratic regression in Delta F2

10 Evaluation

The results show that the original objective of the project has been achieved, as the model can reliably test whether a frequency is within the nadir safe region. The quadratic constrained regression for Area 1 and 2 would be two constraints that would be used as they have almost 100% accuracy and the average closeness to -0.8 is around -0.1 for each area. This is quite low and provides a reasonable insurance however, the model can be improved. The deviation of average -0.1 means that there are some points that will be labelled unsafe despite being safe. Therefore, the constraint could definitely be closer to the true analytical constraint. The project focused on producing the nadir frequency constraint and it was successful.

The previous research into multi area systems by Nikolas Yiapatis, tested the model using R2 score however, as proven in section 8.3, the R2 score does not provide a reliable indication on whether a frequency constraint was successful or not. For example, in quadratic regression for Delta F1 using the large dataset the R2 score was 0.9794, however nearly 50% of the points were below the required nadir frequency. [12]. Therefore, the project has provided an improved method to test the model and delivered a reliable frequency constraint compared to previous work.

The approach taken to testing this model could have been improved. A better method to find the accuracy would be using cross validation for the smaller dataset. This would have involved randomising the data and creating 5 sets of test data, which would consist of 20% of the data. The model will be trained with the respective 80% of the left over dataset and the model the achieves highest accuracy would have been used as the final equation. The reason this was not implemented was due to the efficiency of the system used to produce and process the data was low. This is because, several applications were used from creating the data point in Excel, to converting it into a *csv* format to import into python and running the test in *Matlab*. In ideal system, it would be best that a more automated method was used.

Furthermore, the constrained regression did not use the full large dataset obtained in the first stage as the Matlab program was limited to 100gb. A larger computer with 192gb was used however, the program did not have enough memory to function. Optimisation were made to save space, but the conclusion was made that another software that is able to do constrained regression, similar to Matlab's lsqlinear function should be used. The issue mainly arrised due to the number of points that were being used by the problem. Given more time, the constrained regression would have been done in a different software.

It was mentioned in the interim report that as an extension, more areas could be added to see if frequency constraint would be possible. Whilst this was not the main objective of this project, this extension was not completed as it took a significant amount of time obtain the data points to do the regression with. It could be better to use *Matlab* with parallel computing toolbox to speed up the process of obtaining points.

11 Conclusions and Future Work

In conclusion, the project was successful in creating a frequency constraint that can be used with very high accuracy. The project has been successful in developing a system to obtain operating points, processing them to produce a frequency constraint and testing them to check if they meet the criteria of being the nadir frequency of -0.8.

This project could be taken further if a more efficient method is found to obtain the operating points. To produce a better regression, small steps must be taken with the system inertia and primary frequency response variable. However, due to the limited time frame, the points obtained by running batches of work on multiple machines were not an efficient method to obtain these points. If the workload was optimised and used parallel computing, a better sample set with more accurate samples could be achieved. This was not explored in this project and it could have taken longer to implement than the time frame provided. This area could be explored further in future work.

A more complex regression technique could have been explored, as the frequency constraint is fairly conservative meaning when used in an optimisation problem, the solution provided will not be optimal. This choice to not explore was made as finding an appropriate method to test the model had taken longer than expected meaning there was limited to spend on trying other methods. The testing stage was difficult, as it had been theorised that using random variables would have been suitable, but exploring the relationship between the combined system inertia and primary frequency response showed that only known points in the desired nadir region of -0.795 and -0.800 should be used.

Furthermore, exploring regression techniques provided an better insight into how they work. In particular, using the Vandermonde matrix which characterises the objective of obtaining the least square fit line into a linear algebra problem that can be solved using numerical methods. The process of using different tools such as Python *numpy* and *sklearn* library provided a better idea of how to manage large dataset to obtain a solution.

References

[1] PSNH. Final Report on the August 14, 2003 Blackout in the United States and Canada: Causes and Recommentations, April 2004, U.S.-Canada Power System Outage Task Force. Technical report, 2003.

[2] Zhixin Miao, Lingling Fan, Dale Osborn, and Subbaraya Yuvarajan. Wind Farms With HVdc Delivery in Inertial Response and Primary Frequency Control. *IEEE Transactions on Energy Conversion*, 25(4):1171–1178, dec 2010.

[3] Electricity Ten Year Statement 2015 UK electricity transmission. Technical report, National Grid.

[4] Fei Teng, Vincenzo Trovato, and Goran Strbac. Stochastic Scheduling With Inertia-Dependent Fast Frequency Response Requirements. *IEEE Transactions on Power Systems*, 31(2):1557–1566, mar 2016.

[5] National Audit Office. Rolling out smart meters A picture of the National Audit Office logo. Technical report.

[6] V. Prakash, R. Bhakar, H.P. Tiwari, H.W. Pain, Y. Bian, and F. Li. Primary frequency response in future low carbon scenario: Opportunities & challenges. In *2016 International Conference on Recent Advances and Innovations in Engineering (ICRAIE)*, pages 1–6. IEEE, dec 2016.

[7] P. (Prabha) Kundur, Neal J. Balu, and Mark G. Lauby. *Power system stability and control*. McGraw-Hill, 1994.

[8] Cynthia S Bogorad and Latif M Nurani. Primary Frequency Response Reliability, Regulatory, and Jurisdictional Challenges Posed by Our Changing Resource Mix. Technical report, 2016.

[9] Arthur R. Bergen and Vijay. Vittal. *Power systems analysis*. Pearson, 2006.

[10] Ernest Orlando, Lawrence Berkeley, Joseph H Eto, John Undrill, Peter Mackin, Ron Daschmans, Ben Williams, Brian Haney, Randall Hunt, Jeff Ellis, Howard Illian, and Martinez. LBNL-4142E Use of Frequency Response Metrics to Assess the Planning and Operating Requirements for Reliable Integration of Variable Renewable Generation. Technical report, 2010.

[11] Electricity Ten Year Statement 2015 UK electricity transmission. Technical report, National Grid.

[12] Nikolas Yiapatis Cid and Fei Teng. Frequency Security Constraints for Multi − Area Power Systems. (September):96, 2018.

[13] Andreas Ulbig, Theodor S. Borsche, and Göran Andersson. Impact of Low Rotational Inertia on Power System Stability and Operation. *IFAC Proceedings Volumes*, 47(3):7290–7297, jan 2014.

[14] Luis Badesa, Fei Teng, and Goran Strbac. Simultaneous Scheduling of Multiple Frequency Services in Stochastic Unit Commitment. *IEEE Transactions on Power Systems*, pages 1–1, 2019.

[15] MathWorks. Simulink Documentation - MathWorks.

[16] An Introduction to Decision Tree Learning: ID3 Algorithm.

12 Appendices

12.1 A code *OperatingPointsChecker.c*

```
clear
close all
clc

%%
% Comments:

% The aim of this code is search for operating points within the region of
% -0.795 and -0.8 iteratively using a Simulink model. This model was used
% in the first stage of the project.


% The name of the parameters in the simulink model has to be the same as
% the name of the variables that are being modified in the loop.

%Use the block "to workspace" in Simulink for saving the output of a
%signal. The value will be saved in a variable with the same name as the
%one given to block "to workspace in the Simulink model. For example, in
%this case it's "Delta_f1" and "Delta_f2"

%%
number_areas = 2;
%H = [1/2*4900 1/2*4900]; % Units: MW*s^2
D = [1e-2 1e-2]; % Units 1/Hz
P_D = 35e3*[1/2 1/2]; % Units: MWs
%R = [0.85e3 0.85e3]; % Units: MW
Td = 10; % Units: s
P_loss = 1.8e3; % Units: MW. The model considers the power outage in area 1
X = 25; % Units: ohms. Line reactance
V = 345; % Units: kV (voltage of the transmission line)

%total_time = 300; % Units: s. Total time of the simulation

%%

%array= minimum_value: step_size: maximum_value
xvals= 500:50:4000;
yvals= 500:50:5000;
PFR1vals= 1000:50:5000;
PFR2vals= 1000:50:3000;

len_xvals=length(xvals);
len_yvals=length(yvals);
len_PFR1=length(PFR1vals);
len_PFR2=length(PFR2vals);
M=zeros(60000,5);
i=2;
k=0;
H2=200;
p=0;
h = waitbar(0,'Initializing waitbar...');
for PFR2index=1:len_PFR2
    PFR2=PFR2vals(PFR2index);
    for PFR1index= 1:len_PFR1
        PFR1=PFR1vals(PFR1index);
            for xindex=1:len_xvals
                H1=xvals(xindex);
                H2=100;
```

```
                    sim('TwoArea_fromWorkspace_MaxRamp')  %runs the simulink model
                    k=min(Delta_f1);
                    if k<-0.8
                        while k<-0.8
                            if k<-0.87
                                H2=H2+200;
                            elseif k<-0.81
                                H2=H2+20;
                            else
                                H2=H2+5;
                            end
                            sim('TwoArea_fromWorkspace_MaxRamp')
                            k=min(Delta_f1);
                        end
                    else
                        p=p+1;
                        r=(p/1000);
                        waitbar(r,h, p);
                        break
                    end
                    p=p+1;
                    row=[H1, H2, PFR1, PFR2, k];
                    j=i-1;
                    M(i,:)=row;
                    if M(i,5)<=-0.8 && M(j,5)<=-50
                        break
                    end
                    i=i+1;
                    r=(p/1000);
                    waitbar(r,h, p);
                end
    end
end
```

12.2 B code *UnconstrainedRegression.py*

```

import pandas as pd
import numpy as np
from sklearn import linear_model
import matplotlib.pyplot as plt
from sklearn.preprocessing import PolynomialFeatures
from sklearn.linear_model import LinearRegression
from sklearn.metrics import accuracy_score
from sklearn.metrics import r2_score

#Choosing file to perform regression on
df=pd.read_csv("DeltaF1large.csv")
nameoffile="DeltaF1largeresults.csv"
nameoftester="Deltaf1largetest.csv"

electric=pd.DataFrame(df)
electric.columns=['H2','H1','PFR1','PFR2']

# Seperating features and labels
X=electric.drop('H2',axis=1)
Y=df.iloc[:,0].values

poly_reg=PolynomialFeatures(degree=1)
X_Poly=poly_reg.fit_transform(X)
linearreg_2=LinearRegression()
linearreg_2.fit(X_Poly,Y)
linearreg_2.predict(X_Poly)
```

```

y_true=Y
y_pred=linearreg_2.predict(X_Poly)
linearreg_2.score(X_Poly,Y)

linearreg_2.intercept_
#To output the intercept for recording purposes

linearreg_2.coef_
#To output the coefficents of the fit function for recording purposes

pd.DataFrame(zip(electric.columns, linearreg_2.coef_), columns= ['features', '
    estimatedcoefficients']);

poly_test=poly_reg.fit_transform(np.array([[675, 4916, 2187]]))
linearreg_2.predict(poly_test);
test=pd.read_csv(nameoftester)
test1=np.array(test)
test2=pd.DataFrame(test1)

poly_testvar=poly_reg.fit_transform(test1)
z=linearreg_2.predict(poly_testvar)
test4=pd.DataFrame(z)
result = pd.concat([test4, test2], axis=1, sort=False)
#Producing test results to be inputted into Matlab

np.savetxt(nameoffile, result, delimiter=",")
#Saving the file in CSV format
```

12.3 C code *ConstrainedRegression.m*

```

import pandas as pd
import numpy as np
from sklearn import linear_model
import matplotlib.pyplot as plt
from sklearn.preprocessing import PolynomialFeatures
from sklearn.linear_model import LinearRegression
from sklearn.metrics import accuracy_score
from sklearn.metrics import r2_score

#Choosing file to perform regression on
df=pd.read_csv("DeltaF1large.csv")
nameoffile="DeltaF1largeresults.csv"
nameoftester="Deltaf1largetest.csv"

electric=pd.DataFrame(df)
electric.columns=['H2','H1','PFR1','PFR2']

# Seperating features and labels
X=electric.drop('H2',axis=1)
Y=df.iloc[:,0].values

poly_reg=PolynomialFeatures(degree=1)
X_Poly=poly_reg.fit_transform(X)
linearreg_2=LinearRegression()
linearreg_2.fit(X_Poly,Y)
linearreg_2.predict(X_Poly)

y_true=Y
y_pred=linearreg_2.predict(X_Poly)
linearreg_2.score(X_Poly,Y)

```

```
linearreg_2.intercept_
#To output the intercept for recording purposes

linearreg_2.coef_
#To output the coefficents of the fit function for recording purposes

pd.DataFrame(zip(electric.columns, linearreg_2.coef_), columns= ['features', '
    estimatedcoefficients']);

poly_test=poly_reg.fit_transform(np.array([[675, 4916, 2187]]))
linearreg_2.predict(poly_test);
test=pd.read_csv(nameoftester)
test1=np.array(test)
test2=pd.DataFrame(test1)

poly_testvar=poly_reg.fit_transform(test1)
z=linearreg_2.predict(poly_testvar)
test4=pd.DataFrame(z)
result = pd.concat([test4, test2], axis=1, sort=False)
#Producing test results to be inputted into Matlab

np.savetxt(nameoffile, result, delimiter=",")
#Saving the file in CSV format
```

12.4 D code *TestingNadirFrequency.m*

```
clear
close all
clc
%%
% Comments:

%The name of the parameters in the simulink model has to be the same as
% the name of the variables that are being modified in the loop.

%Use the block "to workspace" in Simulink for saving the output of a
%signal. The value will be saved in a variable with the same name as the
%one given to block "to workspace in the Simulink model. For example, in
%this case it's "Delta_f1" and "Delta_f2"

%%
number_areas = 2;
%H = [1/2*4900 1/2*4900]; % Units: MW*s^2
D = [1e-2 1e-2]; % Units 1/Hz
P_D = 35e3*[1/2 1/2]; % Units: MWs
%R = [0.85e3 0.85e3]; % Units: MW
Td = 10; % Units: s
P_loss = 1.8e3; % Units: MW. The model considers the power outage in area 1
X = 25; % Units: ohms. Line reactance
V = 345; % Units: kV (voltage of the transmission line)

%total_time = 300; % Units: s. Total time of the simulation

filename = 'deltaf1smalltest.xlsx';
fulltestresults = xlsread(filename,'Sheet1');
k=1;
i=1;
M=zeros(10001,5);
j=0;
while k<10001
    k=k+1;
    try
        H1=fulltestresults(i,2);
```

```
        H2=fulltestresults(i,1);
        PFR1=fulltestresults(i,3);
        PFR2=fulltestresults(i,4);
    catch
        break
    end
    i=i+1;
    sim('TwoArea_fromWorkspace_MaxRamp');  %runs the simulink model
    k=min(Delta_f1)
    row=[H1, H2, PFR1, PFR2, k];
    M(i,:)=row;
end

%plot(Delta_f1)
%hold on
%plot(Delta_f2)
```

12.5 E Test Results with Datasets obtained used

The dataset used and the test results were too large to be included in the report Appendices therefore, they have been uploaded to Github. The repository can be found by following this link:

`https://github.com/ImperialCollegeLondon/FinalYearProject`