



ELEC5870M

Final Report

**Accurate Estimation of Inertia for Power Systems with High Penetration of
Renewable Energy Sources During Loss of Generation Event**

Chaim Rui Hendrik van de Ven

SID: 201386924	Project No. 27
Supervisor: Sadegh Azizi	Assessor: Andrew Kemp



ELEC5870M MEng Individual Project

Declaration of Academic Integrity

Plagiarism in University Assessments and the Presentation of Fraudulent or Fabricated Coursework

Plagiarism is defined as presenting someone else's work as your own. Work means any intellectual output, and typically includes text, data, images, sound or performance.

Fraudulent or fabricated coursework is defined as work, particularly reports of laboratory or practical work that is untrue and/or made up, submitted to satisfy the requirements of a University assessment, in whole or in part.

Declaration:

- I have read the University Regulations on Plagiarism^[1] and state that the work covered by this declaration is my own and does not contain any unacknowledged work from other sources.
- I confirm my consent to the University copying and distributing any part or all of my work in any form and using third parties (who may be based outside the EU/EAA) to monitor breaches of regulations, to verify whether my work contains plagiarised material, and for quality assurance purposes.
- I confirm that details of any mitigating circumstances or other matters which might have affected my performance and which I wish to bring to the attention of the examiners, have been submitted to the Student Support Office.

[1] Available on the School Student Intranet

Student Name: Chaim Rui Hendrik van de Ven

Date: 03/05/2024

Signed: .....

1 Abstract

Historically, a network or power system consisted of synchronous generators, converting mechanical energy into electrical energy [1]. This was done with the use of large rotating masses. These rotating masses, as a byproduct of their use, store energy in the form of Inertia. Inertia, “the tendency of an object in motion to remain in motion” [1], could then be tapped to provide stability to a power system. The large rotating masses across a power system provided a soft buffer in the case of a generator dropping out or other system failure (Loss of Generation or LoG event) [1][2][3][4]. However, power sources, such as Solar energy harvested by Photovoltaic cells, are decoupled from the Alternating Current (AC) networks [2], and do not provide any Inertia to the system. With societal factors pressuring grid operators and energy providers to increase the penetration of Renewable Energy Sources (RES), either to increase supply or to replace existing synchronous sources in power systems, low levels of Inertia make frequency stability and compensation more difficult. [4]

Large power systems’ frequency response to a LoG event can be modelled by the Swing Equation, and a Low-Order System Frequency Response (GSFR) model [4][5] if the per unit value loss of generation is known. Problems arise with power systems because corrective measures, deployed in the case of a LoG event, are dependent on knowing the value of Inertia (and other governing variables). This is due to Inertia’s impact on the rate of change of frequency (RoCoF), the depth of the frequency nadirs, and the size of the transient frequency deviation [6][7]. Estimating Inertia accurately is thus not a novel problem when suggesting new methods of Frequency Containment. This project aims to investigate the applicability of the novel method of Optimal Frequency containment suggested by J. Cortes in [7], OFFC, by developing a better understanding of the impact of inaccuracy of Inertia estimations on the System Frequency Response. This project aims to also define a range or method of finding a range for Inertia, for OFFC. The findings should improve the accuracy of the corrective measures for minimizing the magnitude of the transient frequency and the depth of the frequency nadir, and its damaging effects on the grid.

2 Acknowledgements

- Dr. Sadegh Azizi, IEEE Senior Member, Project Supervisor, for all of his help guiding me through this year and suggesting this project.
- Jesus Cortes, IEEE Graduate Student Member, For providing the basis of this project through his novel approach to Optimal Frequency Containment.
- P. M. Anderson, for his numerous expansive works in the field, including the paper on the Low Order SFR model this project leans on heavily.
- F.D.C. Willard for his useful contributions to the discussion

Contents

1 Abstract	3
2 Acknowledgements	3
3 Key Words/Abbreviations	6
4 Introduction:	7
4.1 Inertia	8
4.2 Methods to address a LoG Event	10
5 Project Objectives:	10
6 Chapter 1: Developing Understanding	11
7 Chapter 2: Investigation	16
7.1 Monte Carlo and the Envelope of Power Injections	16
7.2 Modification to Cortes' Ideal Injection	18
8 Chapter 3: Expanding the current Knowledge Base	21
8.1 Sensitivity Analysis of GSFR	21
8.2 Defining the Window/Range of H	23
8.2.1 Average Case	23
8.2.2 Investigating the Limits with worst case and best-case scenarios	26
8.2.3 Best-Case Results	27
8.2.4 Worst Case Results	27
8.3 Analysis of H Range Results	28
9 Conclusions:	30
9.1 Project Discoveries/Results	30
9.2 Further Work	31
9.3 Reflection	32
A Appendix	35
A.1 Functional Flow Chart of Range of H finder Program	35
A.2 Base M File	35

A.3	Monte Carlo Example Code	36
A.4	Range of H Ideal Injection Code	39
A.5	Range of H Halved Injection Code	41
A.6	Range of H Halved Injection Code	45

List of Figures

1	GSFR Simulink Model.	11
2	Anderson GSFR Output.	12
3	GSFR Simulink Model with Injection SigIn.	12
4	Inverse GSFR Simulink Model.	13
5	Delta f(t).	14
6	Example Injection - Correction Factor 0.8.	14
7	Example GSFR response to Injection - Correction Factor 0.8.	15
8	Monte Carlo Output - 20 Runs.	17
9	Monte Carlo Output - 10k Runs.	17
10	Ideal Injection with potential UFLS overlay.	18
11	Comparison of Proposed Injections - Correction Factor 0.8.	19
12	Ideal Injection - Correction Factor 1.	20
13	Halved Injection with potential UFLS overlay.	20
14	Scatter plot of Governing Variables against GSFR Frequency Nadir and Settling Frequency.	23
15	Tornado Plot showing the Correlation Factor between Governing Variables and minimizing Depth of Frequency Nadir.	23
16	Flow Chart of the Range of H Code.	24
17	Limit Violating H response curve for Ideal Injection.	25
18	Limit Violating H response curve for Halved Injection.	26
19	Limit Violating H response curve for Ideal Injection - worst case.	29
20	Limit Violating H response curve for Ideal Injection - worst case.	29
21	Scatter Plot of the Negative changes of Inertia, per 30 minute period, in the National Grid.	31

List of Tables

1	Values used in GSFR model	24
---	-------------------------------------	----

2	Upper and Lower Bounds of accuracy of H using Anderson Values and Ideal Injection	25
3	Upper and Lower Bounds of accuracy using Anderson Values and Halved Injection	26
4	Values adjusted for the Worst Case and Best-Case Testing	27
5	Upper and Lower Bounds of accuracy of H per value of H, Ideal Injection	27
6	Upper and Lower Bounds of accuracy of H per value of H, Half Injection	27
7	Upper and Lower Bounds of accuracy of H per value of H, Ideal Injection	28
8	Upper and Lower Bounds of accuracy of H per value of H, Half Injection	28
9	Rate of Change in Inertia, National Grid, per 30 minutes.	31

3 Key Words/Abbreviations

- FFR – Fast Frequency Response
- H – Inertia
- CoI – Center of Inertia
- RoCoF – Rate of Change of Frequency
- SFR – System Frequency Response
- LoG – Loss of Generation
- AC – Alternating Current
- RES – Renewable Energy Sources
- PFR – Primary Frequency Response
- GSFR - Generalised System Frequency Response
- OFFC - Optimal fast-acting frequency containment
- SG - Synchronised Generator
- Brown out - when the voltage of the grid drops below its usual value

4 Introduction:

The complexity of modern power networks presents significant challenges in their understanding, management, and managing their behaviour. This complexity arises from a combination of factors that characterize the intricate nature of contemporary electrical grids. These factors include the vast physical scale and interconnectedness of power infrastructure, the diverse array of energy sources and technologies integrated into the grid, and the dynamic, nonlinear behaviour exhibited by network components. Attempts have been made to create models for such modern power networks, including a range of models covering the System Frequency Response (SFR) of a modern power network.

In August of 1990, P. M. Anderson and M. Mirheydar, members of the IEEE, published a paper titled “A Low-Order System Frequency Response Model” [4]. The GSFR model, in specific, the Low-Order System Frequency Response Model, is a model constructed for “estimating the frequency behaviour of a large power system To sudden load disturbances” [4].

The GSFR model attempts to simplify larger, more complex power systems, as a single equivalent generator. To achieve this, a set of assumptions are made. This creates a model that is not entirely accurate, but to quote George Box and Norman Draper in [2]:

“All models are wrong, but some are useful.”

The first of such assumptions is that nonlinearities can be neglected [4][8]. The second, is that all but the largest time constants can also be neglected, leaving the generating unit Inertia and Reheat time constants as the predominant forces in the system average frequency response [4]. The third assumption is that the singular equivalent generator is suitably large enough, and the power disturbance is small enough compared to it, that it can absorb the power disturbance [4]. This results in a model that represents the average system dynamics, with the inter-machine oscillations entirely removed.

The GSFR model in Anderson’s paper, is represented by the following transfer function:

$$\Delta\omega = \left(\frac{\mathbf{R}\omega^2}{DR + K_m} \right) \left(\frac{(1 + T_r s)P_d}{s^2 + 2\zeta\omega + \omega^2} \right) \quad (1)$$

A transfer function is a ratio of the Laplace transform of a system’s input to its output [9]. Borrowing from [4], [7] and [8], the 6 governing variables which can be used to derive all other required variables are:

- P_{step} = A step function representing the power disturbance (per unit)
- H = Inertia constant (seconds)
- F_h = Fraction of power generated
- T_r = Reheat time constant (seconds)
- D = Damping Factor (pu/Hz)
- K_m = Mechanical Power Gain Factor
- R = Governor Droop

P_{step} is the power disturbance, represented as a step function, and mimics a loss of power through a fault in the network or a generator failure. It is also a per unit measure of power loss, such that in the case of a grid with 1000MW of power generation, a generation loss of 100MW is a P_{step} of 0.1. F_h is the fraction of power generated by the singular representative generator. T_r is the reheat time constant, measured in seconds, of the prime mover, which is the representative generator. D is the damping factor, which is the rate at which oscillations decay over time. K_m is the Mechanical Power Gain Factor of the generator and system. R is the governor droop, which is the percent change in speed from no load to full load of a generator [10].

These variables fluctuate throughout the day and year, dependent on the generators and loads connected to the grid. This introduces difficulties with Optimal Frequency Containment. Optimal frequency containment

refers to "a mechanism used by transmission system operators (TSO) to keep the electricity grid stable and reliable" [11]. To solve this, methods to estimate these variables at different points in time, such as throughout the day/year, or after a Loss of Generation event (LoG), have been developed. These include measuring locally at points in the grid, communicating between these points to find the rate of change of frequency (RoCoF), across the grid [6][7]. Another method, proposed by Amin Nassaj and Dr. Azizi, in their paper "Fast Linear State Estimation for Unbalanced Distribution Systems Using Hybrid Measurements", uses a novel "linear DSSE (LDSSE) method" which "can function with purely unsynchronized or hybrid synchronized/unsynchronized measurements" [12]. For this report, the range for the governing variables have been derived from the book "Power System Protection" [13] and [4].

4.1 Inertia

Inertia, "the tendency of an object in motion to remain in motion" [1] is a form of energy. It can be calculated for a rotating mass, using angular velocity and mass. This is relevant for power systems due to how synchronous generators produce power. Historically, power generation is achieved by rotating a turbine, using either steam or water. This turbine has a mass and thus inertia. In the case of a LoG event, the inertial energy of the rotating mass of these turbines can be tapped to slow down the RoCoF of the power system [1]. Inertia, in a power system, resists changes in frequency by temporarily making up for any lost generation of power, providing crucial time for systems to compensate and re-balance the network [1][2][3][4]. Inertia provides a buffer or time to respond.

The effects of Inertia on the RoCoF of a system after a LoG event can be seen in [4]. Changes in Inertia impact the depth of the frequency nadir, the time it takes the system's frequency to drop to the frequency nadir (the point of greatest frequency change), and the magnitude of the transient frequency dip [1][4][6][7]. Inertia, however, has no impact on the frequency the system settles at, in response to the LoG event. [4][7][14][15]

This can be seen in the swing equation. The swing equation quantifies the relationship between Inertia, damping factor, a Function of Frequency and Delta P. The swing equation "is a non-linear second order differential equation that describes the swing of the rotor of synchronous machine" [16]. Different sources have different representations of the swing equation; however they are all similar. For example, Equation 2 comes from [16], and clearly uses a different set of variables than Equation 3, or 4, from [7] and [17].

$$\frac{2H}{\omega_s} \frac{d^2\delta}{dt^2} = P_m - P_e = P_a(pu) \quad (2)$$

The Swing Equation.

$$\Delta P = 2H \frac{df(t)}{dt} + Df(t) \quad (3)$$

Swing equation per paper [7] by Jesus Cortes et al.

$$M \frac{d^2\delta}{dt^2} + P_d \frac{d\delta}{dt} + P_e - P_m = 0 \quad (4)$$

Swing Equation per paper [17] by Colin ADAMSON et al.

These are all slightly different representations of the same equation. For example, M is the Inertia Constant, and is equivalent to $2H$ [18]. It can be easily inferred that P_d is equivalent to the damping factor. The existence of multiple conventional swing equations is suggested in [19]. It is also based on several assumptions, making it another inaccurate but useful model, like the GSFR model. It can be seen from this equation, and all its variations, that Inertia plays a dominant role in determining the depth of the transient frequency dip, or the frequency Nadir.

With the increasing awareness of global warming within society, there are increasing societal pressures on the energy sector to adopt "greener" methods of energy production. These incentives to increase the penetration of Renewable Energy Sources (RES), either to meet new demand or to replace traditional synchronous generators, phasing out pollutants for clean energy, decreases the Inertia of the grid. This steadily decreases the availability of inertia to assist in network frequency stability. As can be seen in [4] and [7], the effect of a decrease in inertia, on the GSFR model during a LoG event is an increase in the transient frequency dip,

a deeper frequency nadir, and a greater RoCoF. Large changes in frequency across a power system can cause cascading faults. Mechanical problems involving torque arise when a generator's turbine spins at frequencies they are not designed for, which can result in damage to the generators, triggering cascading faults, brown outs, and in the worst case, black outs [15].

Calculating inertia is simple, for singular generators. As a function of mass and angular velocity, inertia can be found for simple synchronous power systems by calculating the inertia of each individual generator and finding the sum. Problems arise when the power system expands in size, or generators with variable inertia are introduced. Calculating inertia for the entire power system accurately becomes more difficult. In terms of Optimal Frequency Containment, this relationship between Inertia and the RoCoF, and the difficulties of finding a value for inertia, results in the GSFR model using a value for Inertia that is often an 'expected' value for the system, not an exact one. This leaves a Margin of Error. As the penetration of RES into the grid increases, it becomes even more difficult to maintain an accurate estimation of Inertia. Photovoltaic cells can fluctuate output during the day due to weather conditions, etc. As these generators fluctuate their generation, the ratio of total power to spinning mass in the system changes, hence inertia changes.

As mentioned above, decreasing inertia increases the RoCoF and the transient deviation. This becomes relevant for grid engineers because of the consequences of large changes of frequency in the grid. Our electronics and generators are designed to operate at a specific frequency, of 50 Hz AC. These machines/devices can endure small changes in those values, but not for an extended period. Fluctuations in that frequency for extended periods are problematic for the following reasons:

- Effect on Generators:

Generators are designed to operate at specific synchronous speeds corresponding to the grid frequency (e.g., 50 Hz or 60 Hz). Any deviation in grid frequency from this standard can affect the rotational speed of generators. For example, if the grid frequency decreases, generators connected to the grid may slow down. This can lead to reduced output and efficiency, potentially causing mechanical stress on turbine components and affecting the lifespan of the generator. [20]

- Frequency Sensitivity:

Some types of generators, particularly those with steam turbines or gas turbines, are sensitive to changes in grid frequency. Steam turbines, for instance, rely on precise synchronization with the grid frequency to maintain stable operation. Deviations in frequency can result in turbine overspeed or under speed conditions, triggering protective shutdowns to prevent damage. [21]

- Voltage Regulation:

Changes in grid frequency can also impact voltage regulation. Most synchronous generators are designed to maintain a constant output voltage relative to the grid frequency. If the frequency deviates, it can lead to fluctuations in voltage levels. This can affect the stability and reliability of electrical equipment connected to the grid. [22]

- Electronics and Power Quality:

Many electronic devices and sensitive equipment rely on a stable supply of electrical power with consistent frequency and voltage characteristics. Fluctuations in grid frequency can result in variations in voltage levels and power quality. This can lead to issues like flickering lights, unstable operation of electronic devices, and potential damage to sensitive components. [23]

- Frequency-Sensitive Loads:

Certain types of equipment, such as clocks, motors, and timing devices, are designed to operate based on grid frequency. Changes in frequency can cause these devices to run faster or slower than intended, affecting their accuracy and performance. [24]

- Potential Damage and Malfunctions:

Rapid or significant changes in grid frequency, especially during disturbances or grid events, can pose risks of equipment damage and malfunctions. This is particularly critical for power-sensitive industries, data centres, and critical infrastructure where uninterrupted and stable power supply is essential.[25]

Several of the mentioned effects could cause cascading faults in the grid, as more and more LoG events occur due to mechanical damage or protective shutdowns disable generators. This inevitably would cause brown/black outs which, as mentioned above, could pose as a major risk to critical societal infrastructure, such

as hospitals, schools, data centres, etc. To avoid this, the government has put in place the statutory limits for frequency deviation in the grid. This allows for a 1 percent deviation in the grid's frequency before fines are placed on the grid providers.

4.2 Methods to address a LoG Event

One fundamental approach to Optimal Frequency Containment, primary frequency response, automatically adjusts a synchronous generators output power to counteract sudden imbalances between supply and demand. For instance, when demand increases unexpectedly, generators ramp up their output to match the higher demand, thereby stabilizing the grid frequency. Conversely, if demand decreases abruptly, generators reduce their output to prevent an over-frequency condition. This method relies on a suitable “cushion” provided by inertia, to allow the generators to ramp up before the RoCoF brings the system frequency below statutory limits.

This method has been evolved to include injection from RES, as “controlling the output of power electronic devices is typically much faster than the response time of SGs” [7].

Another method is Under Frequency Load Shedding (UFLS) schemes. UFLS schemes are defined as “emergency mechanisms that are designed to mitigate the risk of power system collapse following multiple Non-Credible Contingency Events.” [26] UFLS is a compromise between linear control and a predefined set of loads to drop in the case of a disparity between demand and supply [27][28]. It acts as a pseudo injection of power to a system, from the perspective of its effect on the frequency response of the system [27][29].

Lastly, J. Sanchez Cortes, alongside Dr. S. Azizi, has proposed a novel approach to compensating the transient section of the GSFR model, for any LoG event. The paper, [7], suggests using a triangular injection of power, to optimally remove the transient dip, either partially or entirely. This also decreases the depth of the frequency nadir. The proposed method, optimal fast-acting frequency containment (OFFC), suggest that “The core aim of OFFC is to reduce or completely remove the transient frequency deviation” [7]. This is a proposed improvement over current methods of compensating for both the Transient Deviation and Steady-state Deviation portions of the GSFR response. Instead, Cortes proposes addressing the entirety of the Transient Deviation, which is the most likely portion to be under the statutory limit and cause the mechanical damage mentioned earlier. This would optimize the use of available energy, over older methods. This project will explore the feasibility of this solution based on an investigation into defining the acceptable inaccuracy range for Inertia, when applying OFFC.

5 Project Objectives:

- To investigate the Probabilistic Envelope of Frequency Responses output from the SFR model.
- To investigate the frequency responses when attempting to compensate for the transient frequency dip.
- To investigate the effects of the Margin of Error specific to each average variable on the SFR model
- Evaluate the effects of the estimation of H on the viability of the application of OFFC
- Define an acceptable range of H from the estimated value of H, in the OFFC use case

6 Chapter 1: Developing Understanding

To develop a better understanding of the GSFR model, and Cortes' OFFC injection, the two corresponding papers, [4] and [7], were remodelled, in both MATLAB and Simulink. These needed to be developed to generate data for future analysis, and to create MATLAB files which can alter the 7 governing variables. The first stage was to model Anderson's "A Low-Order Frequency Response Model" in Simulink. Anderson's Low Order GSFR model reduces a complex power system into an equivalent single generator grid. This was done to simplify the Transfer Function enabling the study of a modern power network. This results in equation 1:

$$\Delta\omega = \left(\frac{\mathbf{R}\omega^2}{DR + K_m} \right) \left(\frac{(1 + T_r s)P_d}{s^2 + 2\zeta\omega + \omega^2} \right)$$

This equation can be converted into a Block diagram, and then modelled in Simulink. This Simulink model can be seen in Figure 1. It is important to notice the use of the constant block P_d . This allows the Simulink model to be loaded in a MATLAB M file, and the variable P_d to be loaded before the model is run. This allows the automation of running many iterations of the Simulink Model. One such output can be seen in Figure 2.

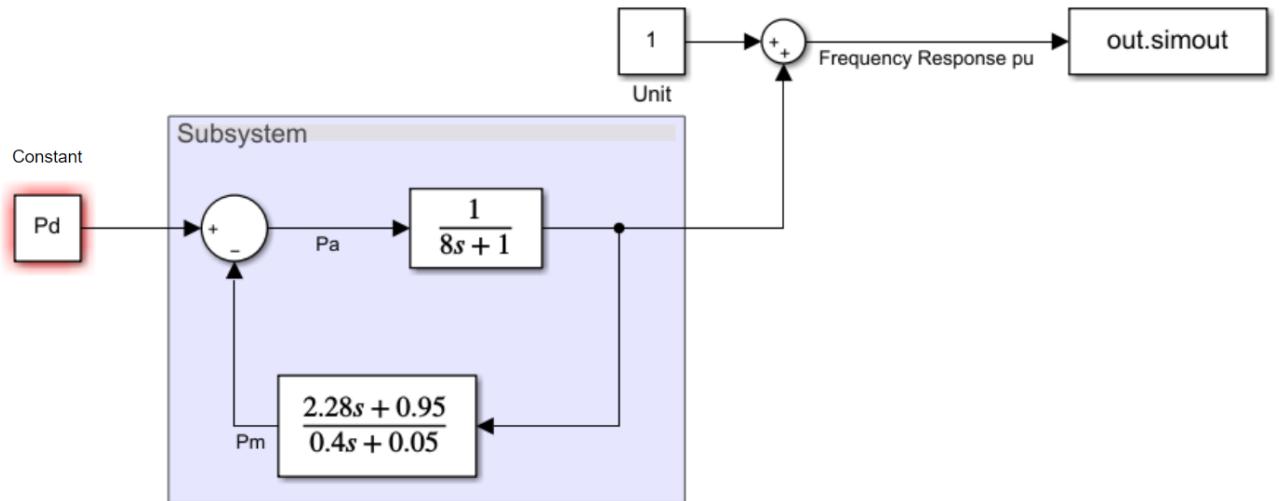


Figure 1: GSFR Simulink Model.

The GSFR graph has a few features of interest for this project. The X axis is time, while the Y axis is the Frequency, per unit. The important part is the per unit result and needs to be acknowledged in any designs or conclusions made based on this GSFR model. The P_d value, which represents the variable P_{step} , -0.3 in the example figure, is also a per unit representation of a LoG event. For example, a loss of generation equivalent to 3W in a 10W system would respond as shown in Figure 2. From the response curve, the following data can be extracted: Depth of the Transient Deviation or the Frequency Nadir, the Settling Frequency, and the Depth of the Steady-State Deviation. The settling frequency can be found, as shown in Figure 2, by taking the value of the response curve at 20 seconds. This isn't perfect in a normal power grid, but as Anderson's GSFR model assumes no major oscillation after the initial transient dip, this is a suitable way to acquire the settling frequency. The Frequency Nadir can be found by taking the lowest point in the entire response curve.

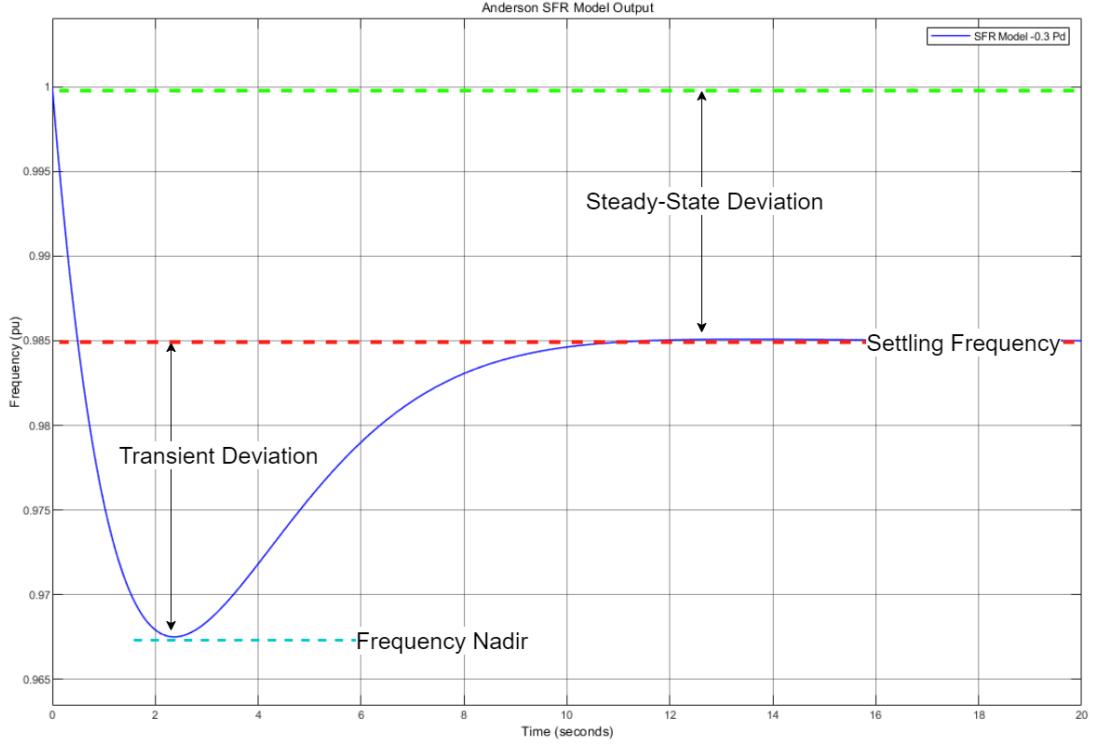


Figure 2: Anderson GSFR Output.

This initial model helped establish the understanding needed to pursue the rest of the project but was incomplete for further use. Thus, a second version of this Simulink model was made, as featured in Figure 3. This version features an Injection Signal, which can be initiated as a time series in a MATLAB m file. The biggest change, however, is the removal of the feedback loop, and the replacement of Anderson's average variables with the governing variables. These two changes enable this model to be used multiple times by a MATLAB m file without repeatedly manually calculating the transfer function, by setting the governing variables in the MATLAB workspace before running the model. The equations now featured in the transfer function are taken from [4], and enable the transfer function to be recalculated for every run.

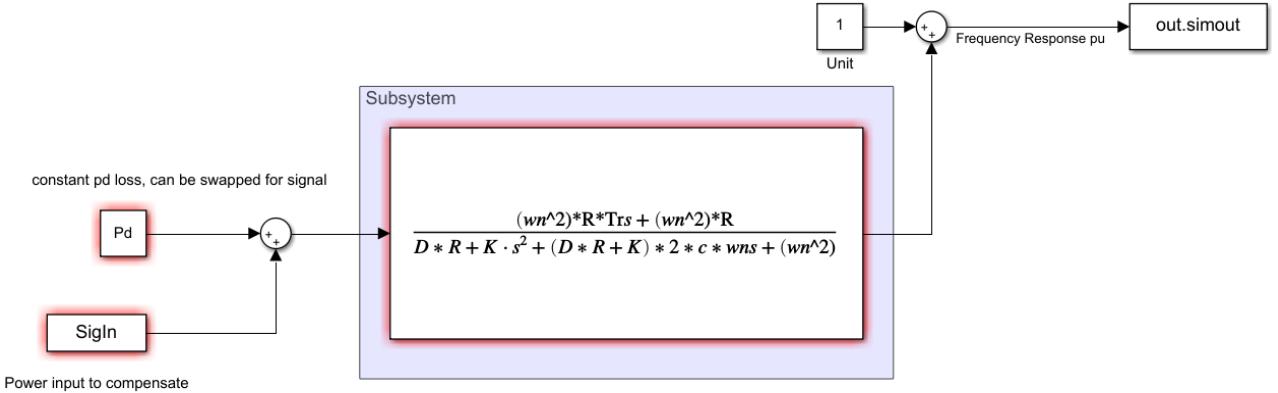


Figure 3: GSFR Simulink Model with Injection SigIn.

The next step was to model/find the injection suggested by Cortes in [7]. This model will become fundamental in the further research of this project. Originally, this was attempted in LiveScript symbolically. This focused on working through [7], recreating the functions and equations that Cortes used to create his Optimal Power Injection findings, developing a deeper understanding. Due to MATLAB's limitations when handling Laplace transforms, efforts were redirected to making the Simulink model seen in Figure 4. The way Cortes' suggestion works is by isolating the transient deviation using 2 values, tau 1 and tau 2. These values can be defined by the amount of compensation or correction of the transient deviation an engineer would like. Tau 1 and tau 2 are then found by looking for the first and second roots of the equation $f(t) - (\beta d_{tr} + f_n) = 0$, where $f(t)$ is the frequency response of the GSFR model, β is the correction factor, d_{tr} is the depth of the

transient deviation, and f_n is the new frequency response which the injection is trying to create [7]. Tau 1 and Tau 2 can then be used in a piece-wise function to isolate the frequency response needed to compensate for the transient deviation and inverse it. The result of which is known as $\Delta f(t)$ and is shown in Figure 5.

$\Delta f(t)$ is then used to find a power injection by putting it through the inverse of the GSFR transfer function. Finding the inverse of the GSFR transfer function required partial fraction expansion algebraically, as the model needs to have individually addressable variables. For example, take equation:

$$G(s) = \frac{xs^2 + as + b}{cs + d} \quad (5)$$

Finding the inverse of this transfer function results in the equation 6, where each term of the equation is a different block in Simulink. The first is the derivative block, $\frac{x}{c}s$. The second is the gain block, with no variable s. The third is the new transfer function block. Using the original GSFR equation, equation 1, the variables x, a, b, etc. can be swapped for their governing variable representations. This was then turned into the Simulink model “Inverse GSFR”, as seen in Figure 4 (the final version of this model, as it required several changes throughout the project’s duration depending on it’s use).

$$G(s)^- = \left[\frac{x}{c}s \right] + \left[\frac{a - \frac{dx}{c}}{c} \right] + \left[b + \frac{(a - \frac{dx}{c})(\frac{d}{c})}{cs + d} \right] \quad (6)$$

To prove this inverse GSFR worked as expected, both models were connected in an M file, and the outputs were compared to Cortes’ findings in [7]. An example of the injection outputted by the inverse GSFR and the output of the GSFR with the injection, can be seen in figures 6 and 7, where the correction factor β has been set to 80 percent.

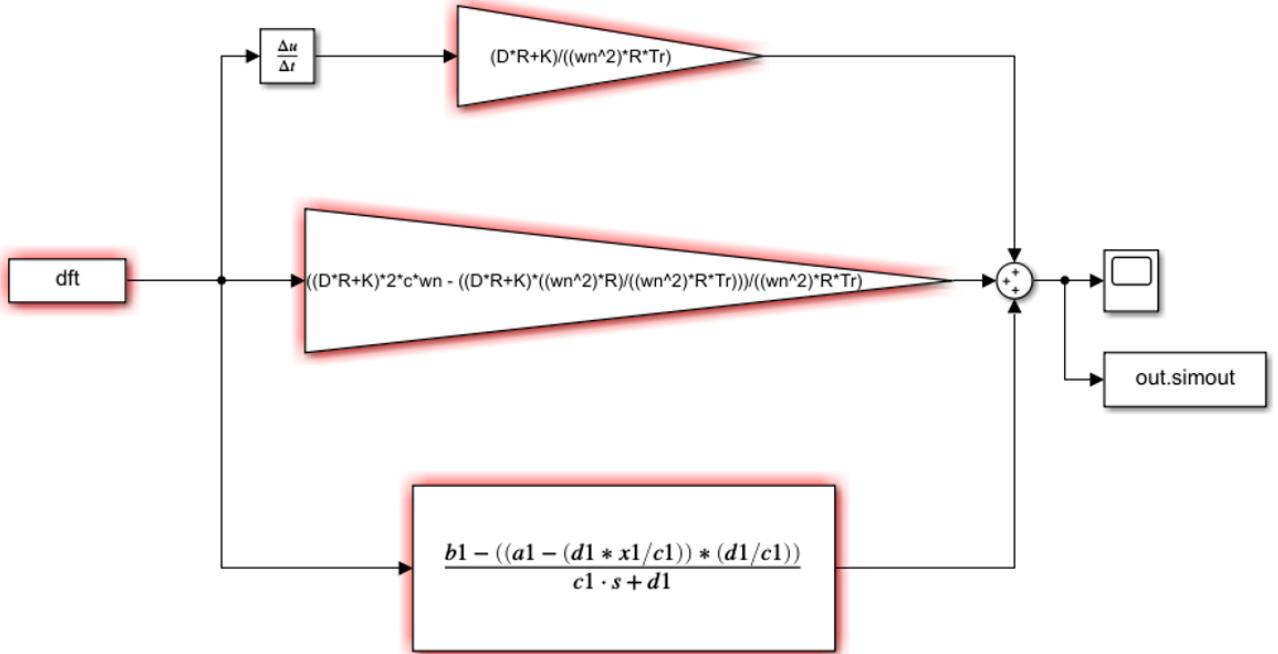


Figure 4: Inverse GSFR Simulink Model.

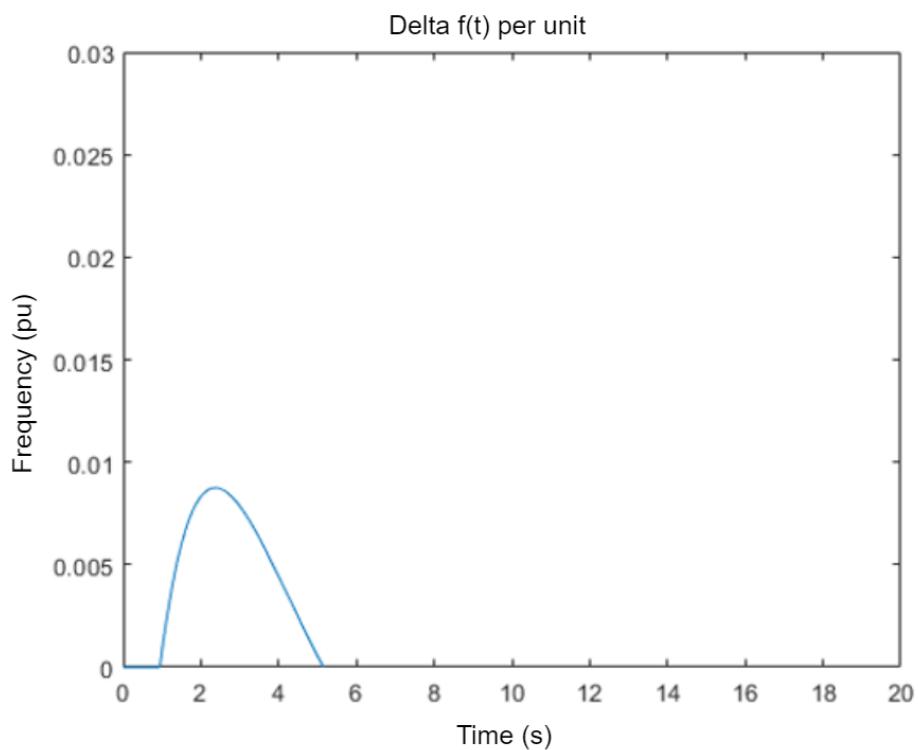


Figure 5: Delta $f(t)$.

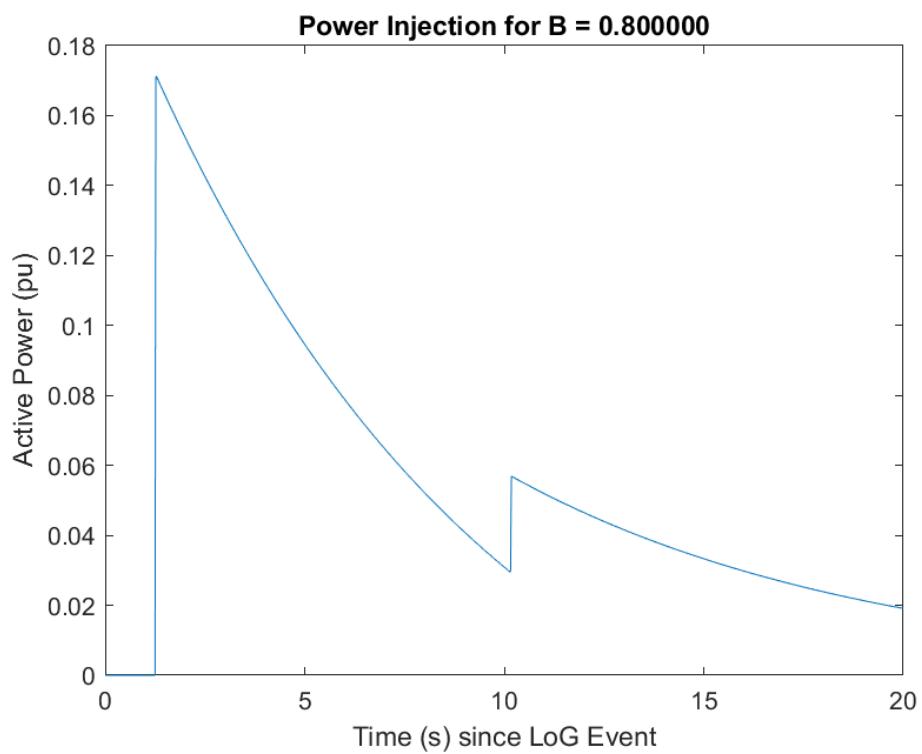


Figure 6: Example Injection - Correction Factor 0.8.

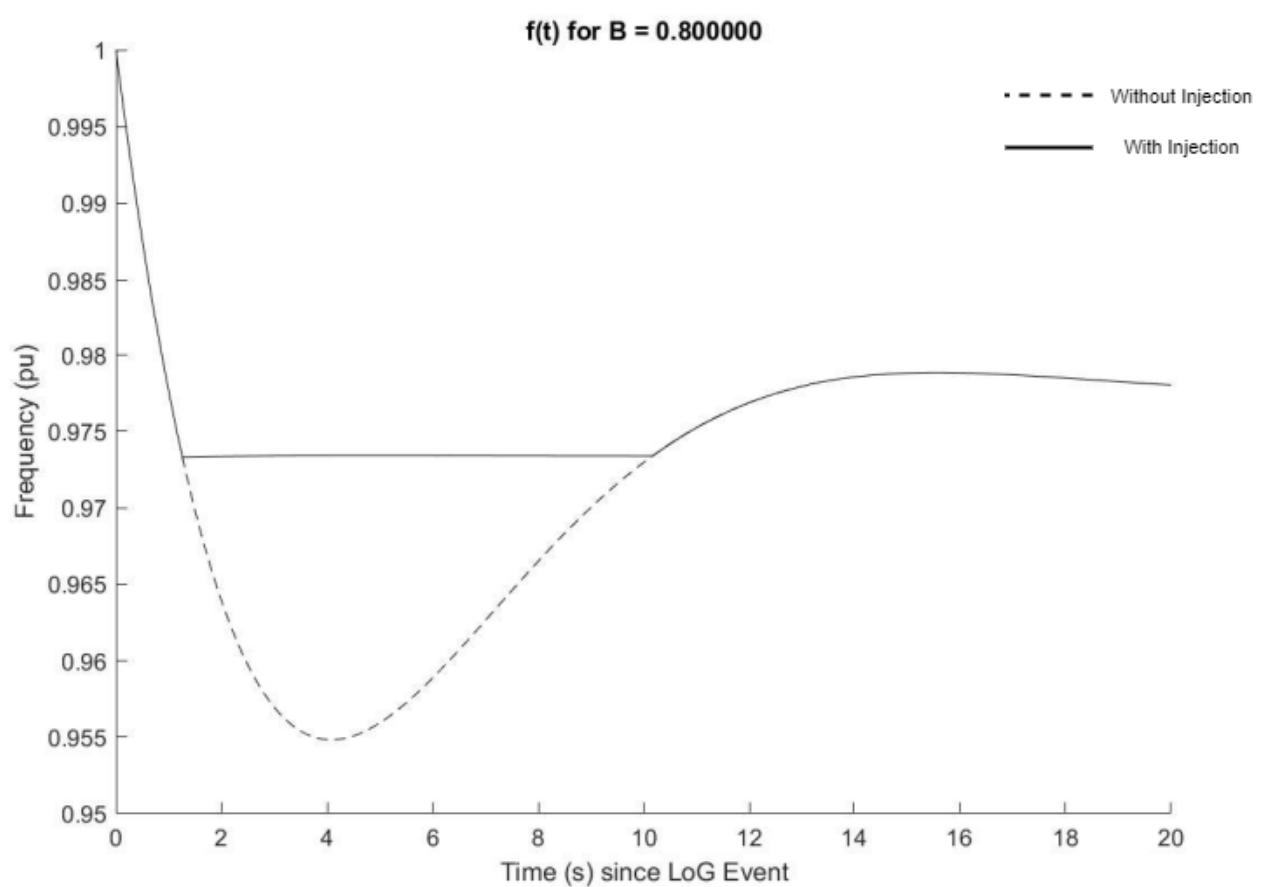


Figure 7: Example GSFR response to Injection - Correction Factor 0.8.

7 Chapter 2: Investigation

7.1 Monte Carlo and the Envelope of Power Injections

Based on the work done in Chapter 1, a MATLAB script was developed to act as the foundation for all further scripts/investigations. This script can be seen in appendix A.2. The first stage of the investigation into Cortes' power injection was to examine the 'Envelope of Power Injections' and the frequency response, with the use of Monte Carlo simulations. The base MATLAB script was thus modified to run N number of simulations, with a normal distribution for each governing variable. The ranges determined for this normal distribution were originally based on the values suggested in [4] with a standard deviation defined by the ranges suggested in [13]. The method chosen to examine the effect of inaccuracy in the values used for the injection, on the frequency response curve after a LoG event, is shown by the following procedure:

1. Establish the mean of the governing variables (Using [4]) as the "estimated" values.
2. Run the GSFR model using the estimated values, without a power injection.
3. Compute tau 1 and tau 2, using $f(t) - (\beta d_{tr} + f_n) = 0$.
4. Process the GSFR output into delta f(t), by isolating the section of the frequency response curve to compensate for based on correction factor β .
5. Feed delta f(t) into the Inverse GSFR model, producing the power injection for the estimated values.
Store the injection.
6. Use a normal distribution rand function to assign randomized values, within 3 standard deviations of the mean values, to the governing variables.
7. Assign the Injection found in step 5 as the power injection to the GSFR model.
8. Run the GSFR model with the power injection and the randomized variable values.
9. Store and plot the output in a .csv file and figure respectively.
10. Loop through steps 6-9 N times, to establish a Monte Carlo simulation of N iterations.

Two examples of the output results for this procedure can be seen in figure 8 or in figure 9. The dashed black line is the original GSFR response, plotted to see the difference the compensating power injection has. The solid black line is the GSFR response, with mean value variables and the compensating power injection. Lastly, the array of dashed coloured lines is the individual Monte Carlo runs.

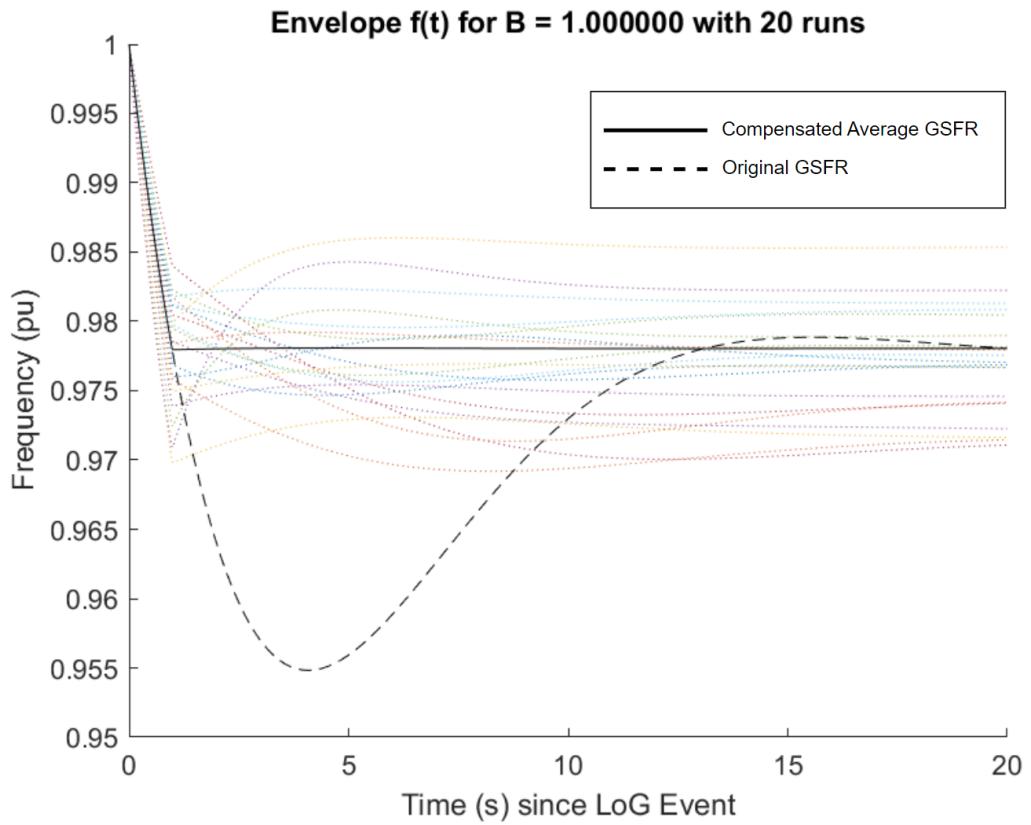


Figure 8: Monte Carlo Output - 20 Runs.

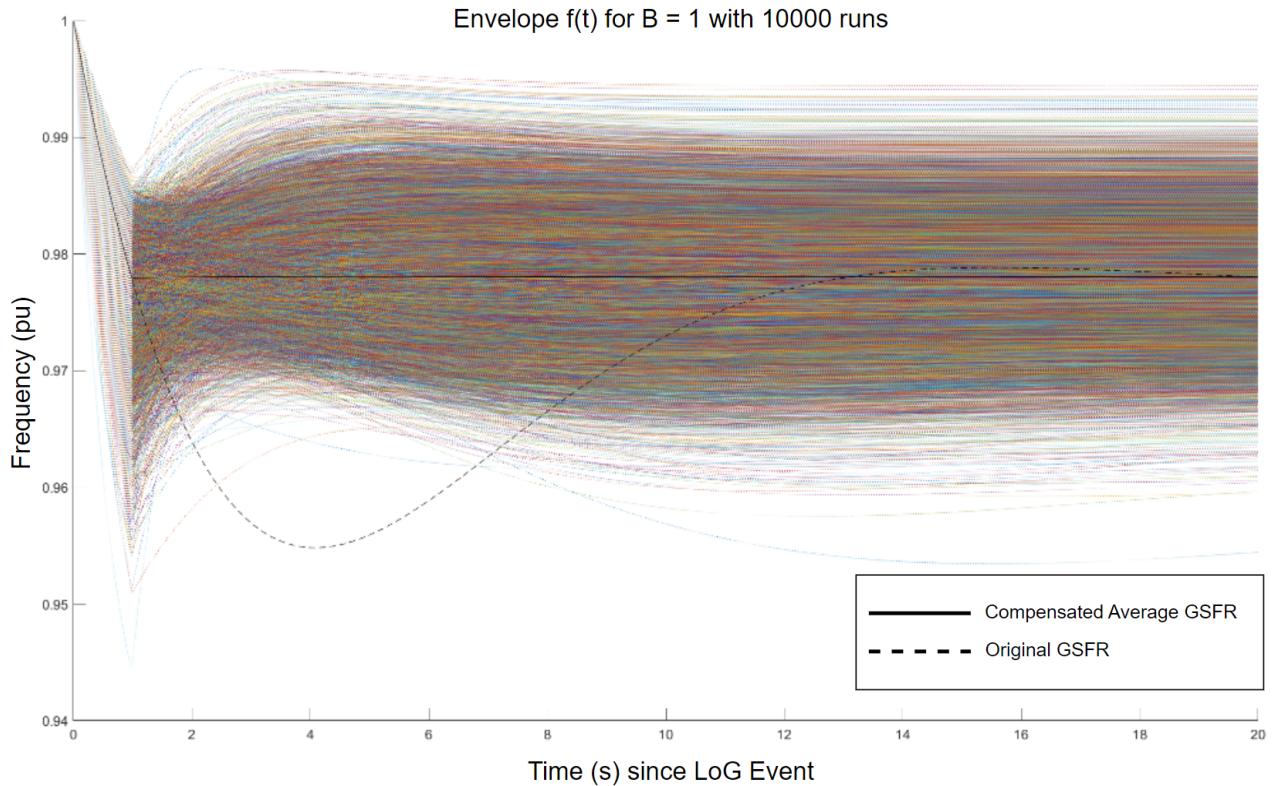


Figure 9: Monte Carlo Output - 10k Runs.

The results of this brief investigation yielded very little of interest, however, it demonstrated the potential major variation between the expected frequency response and the real frequency response with inaccuracy. The need for this project.

7.2 Modification to Cortes' Ideal Injection

During the Monte Carlo investigation, research was done into the feasibility of deploying Cortes' injection, dubbed from this point on as the Ideal Injection (Figure 6). There are two apparent problems with the Ideal Injection, which arise from one suggested implementation, using UFLS. As mentioned earlier, UFLS is a crucial strategy employed in power systems to prevent widespread blackouts during periods of severe imbalance between power supply and demand, such as during a LoG event. This process specifically involves disconnecting or shedding pre-determined blocks of load. This shedding effectively reduces the total power demand, which is akin to injecting power back into the system. By shedding load, the effective power deficit is decreased, allowing the remaining generation to balance the grid and stabilize frequency.

These pre-determined blocks, when shed or reintroduced, change the load of the power system in chunks, not a curving slope as seen in the Ideal Injection. Figure 10 demonstrates what this would look like, on top of the Ideal Injection. After the initial shedding of an aggregation of blocks, consecutively reintroduced blocks would have to be a specific size/load, each smaller than the previously reintroduced block. This would be difficult on its own, but the second triangular injection needed would require a second set of shedding.

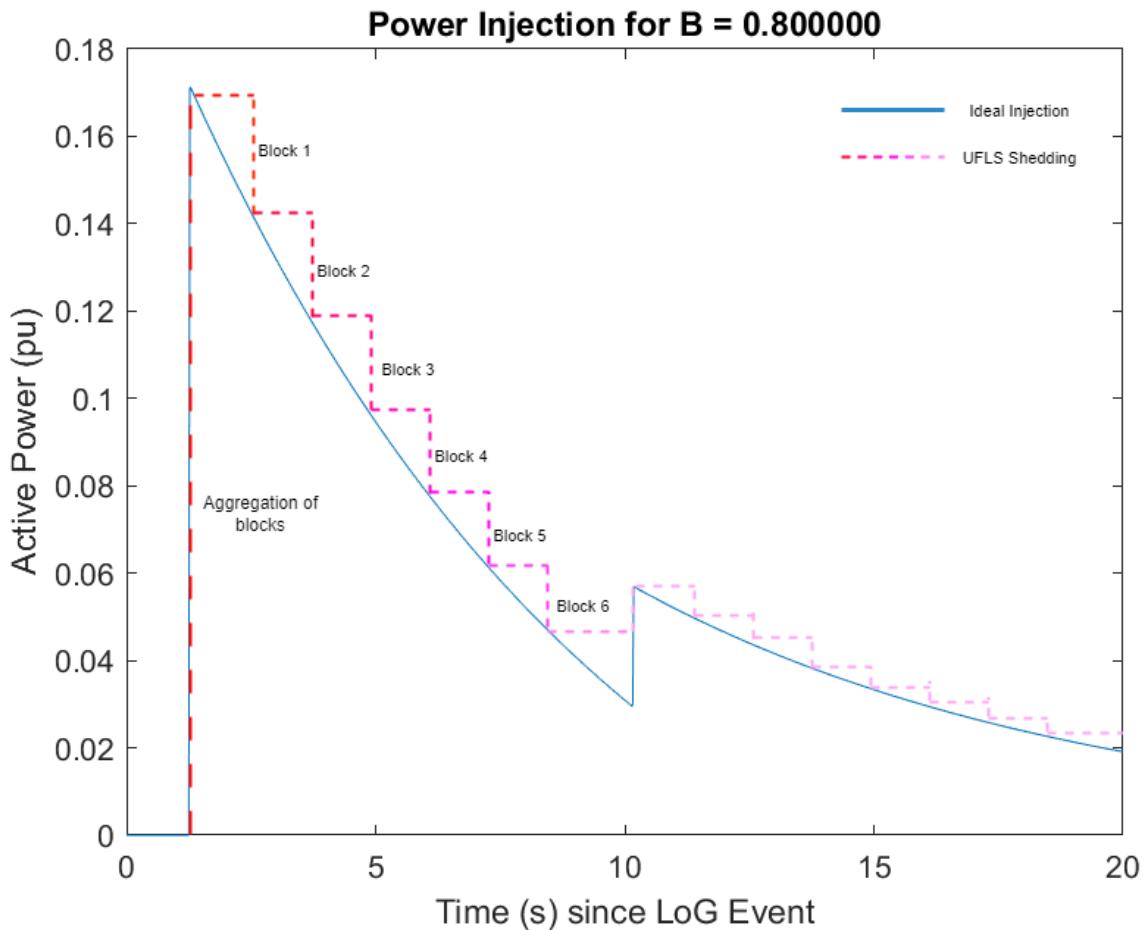


Figure 10: Ideal Injection with potential UFLS overlay.

This led to the suggestion of simplifying Cortes' injection. Simplifying the injection would alter the frequency response behaviour, and this would need to be examined alongside the response of the Ideal Injection. The proposed alternatives are separated into a Simplified Injection and a Halved Injection. These can be seen in figure 11 layered on top of the Ideal Injection. These injections were found using the Ideal injection, thus Cortes' work would still need to be done to establish both the alternative injections, for any instance of a LoG event. The Simplified Injection uses the peak injection value and the final injection amount, to create a constant slope between the two. This results in a greater area under the curve, which translates to a greater amount of energy required, per unit, to create the Simplified Injection. This extra energy requirement is counterproductive to the purpose of Cortes' OFFC, which aims to optimize the use of energy when performing Frequency Containment. The last alternative, the Halved Injection, was developed to address the extra energy

for simplicity trade off. The Halved Injection is based on the idea that the second triangular injection may not be required to adequately address majority of the Transient Deviation. This was established by looking at the difference between a correction factor less than 1 and 1 (see figure 12).

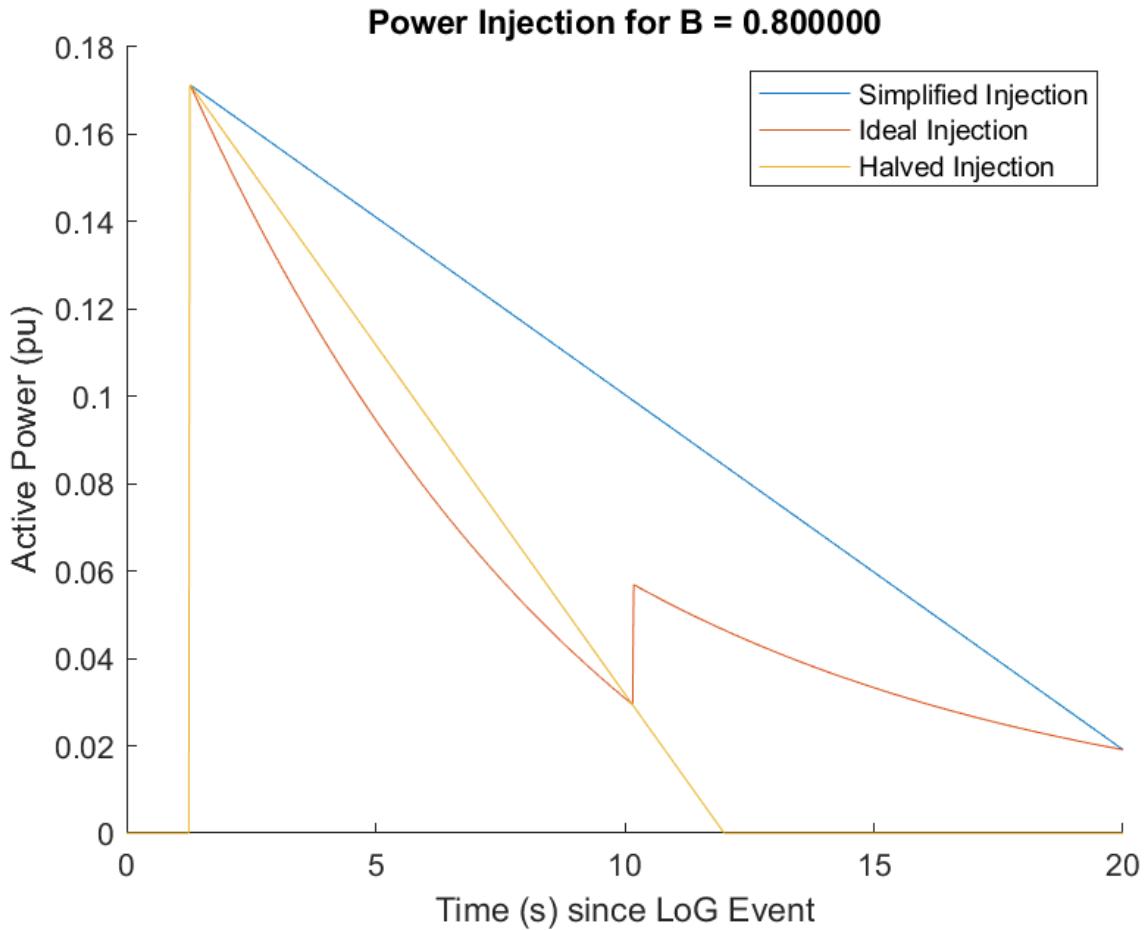


Figure 11: Comparison of Proposed Injections - Correction Factor 0.8.

Generating the Halved Injection is simple. Using the Ideal Injection, search through the data points, comparing the current data point's Y value (active power) to the next data point's Y value, saving the point where the next Y value is greater than the current. A slope can then be established, starting from the peak Active Power, carrying through the point found, till no more power is injected (after the Ideal Injection would have begun the second triangular injection). As can be seen in figure 11, the Halved Injection uses significantly less energy (area under the curve) than the Simplified Injection but shares the same simplified slope. This Halved injection, using UFLS, can be seen in figure 13.

Monte Carlo simulations were then run for the 2 new suggested Injections as well. As a result, the Halved Injection was selected over the Simplified injection for the rest of the project.

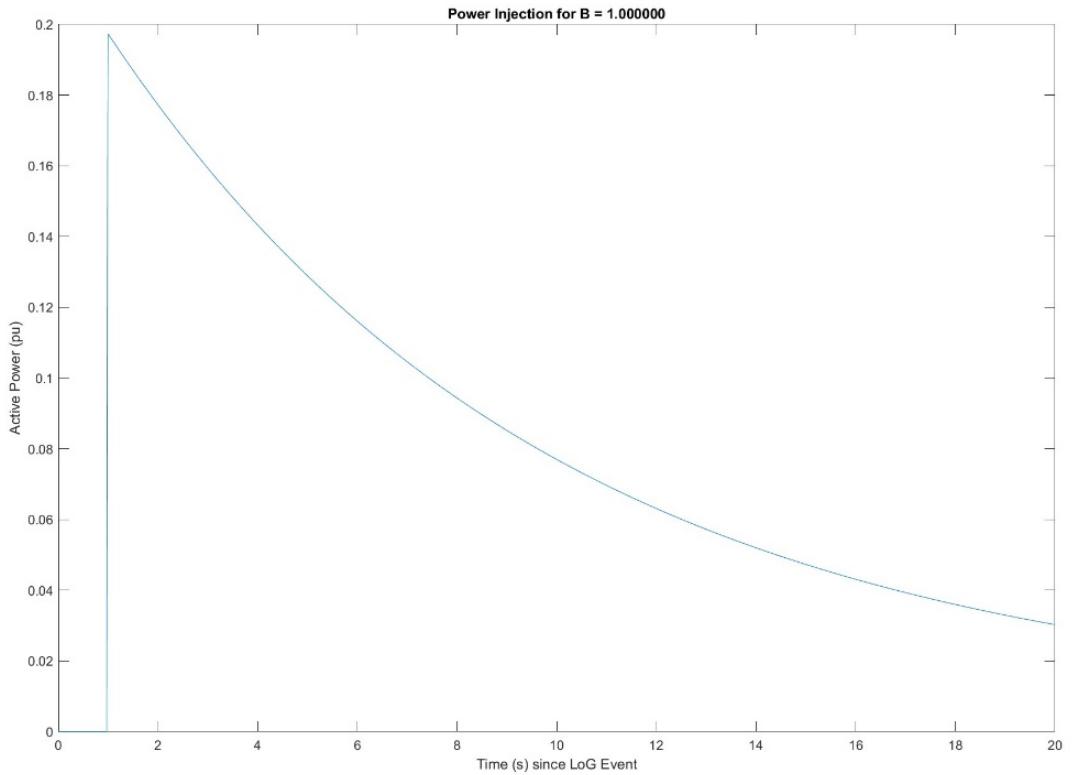


Figure 12: Ideal Injection - Correction Factor 1.

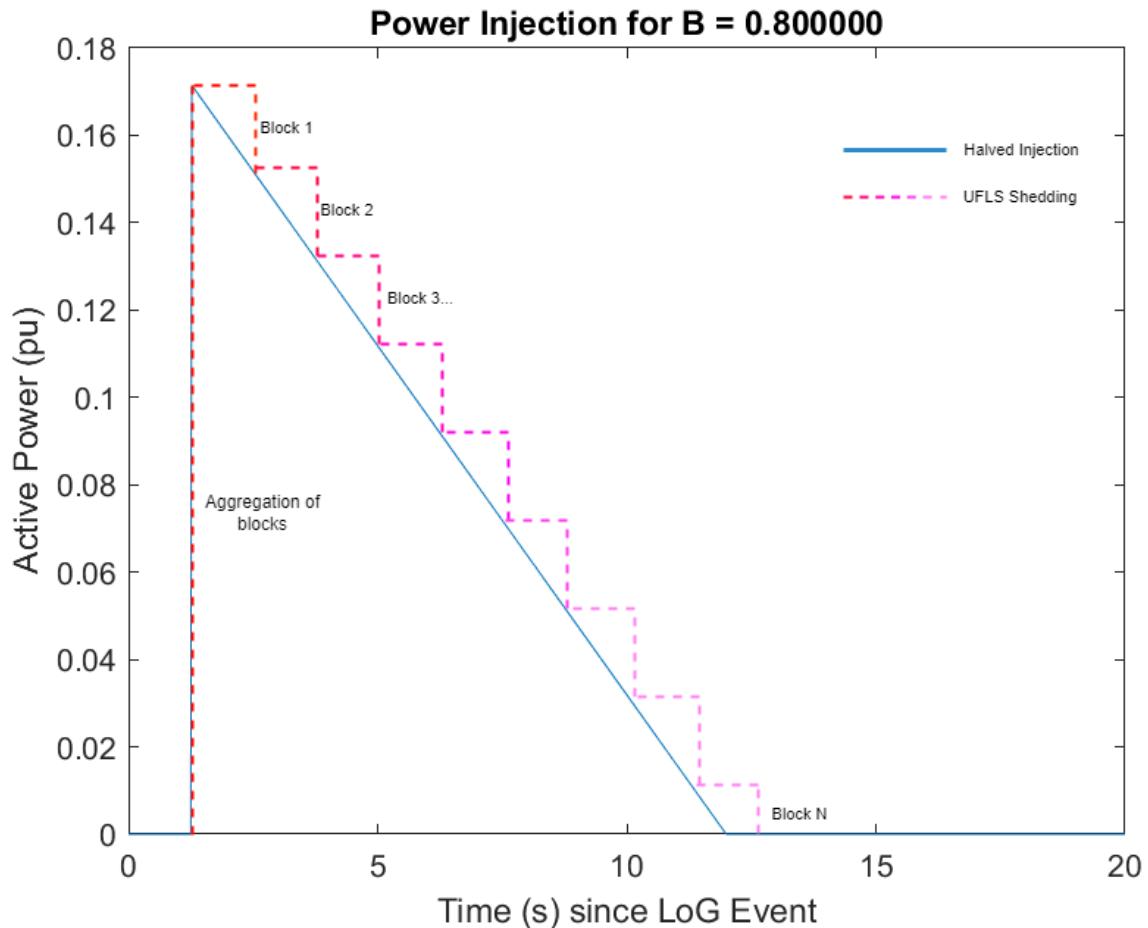


Figure 13: Halved Injection with potential UFLS overlay.

8 Chapter 3: Expanding the current Knowledge Base

8.1 Sensitivity Analysis of GSFR

One of the objectives of this project was to evaluate the effects of the estimation of H on the viability of the application of OFFC. To do this, it must first be established how the GSFR model, and by extension the Inverse GSFR model, responds to uncertainty or variations within the values of the governing variables. This was partially done by the Monte Carlo Simulations, but not with much clarity. For this purpose, each governing variable must be both observed in isolation, and observed with the other factors affecting the model's outcome. More specifically, on the Frequency Nadir, the Time spent below the Statutory limits, and Settling frequency.

To begin, each variable was observed in isolation. Similarly to what was done by Anderson in [4], the characteristics of each variable was identified by running simulations where only that variable changed value. Its impact on the Frequency Nadir and the area of the transient deviation were recorded. This is because the time spent under the Statutory limit, of 1 percent, or 0.99Hz per unit, is what causes cumulative damage to synchronous generators in the grid, and what OFFC is attempting to avoid. The result of this isolated investigation is as follows:

- D: Damping Factor
 - Increases the RoCoF immediately after the LoG event as Damping Factor increases.
 - The depth of the transient deviation, or the frequency Nadir, decreases as Damping Factor increases.
- Fh: Fraction of Power Generated
 - The depth of the transient deviation, or the frequency Nadir, decreases as Fh increases.
 - Slightly Increases the RoCoF immediately after the LoG event as Fh increases.
 - Is bounded between 0 and 1.
- H: Inertia
 - Decreases the RoCoF immediately after the LoG event as Inertia increases.
 - The depth of the transient deviation, or the frequency Nadir, decreases as Inertia increases, but far less than an equivalent percentage change in Fh.
- Km: Mechanical Power Gain Factor
 - Slightly decreases the RoCoF immediately after the LoG event as Km increases.
- R: Governor Droop
 - Increasing values decrease the RoCoF.
 - The depth of the transient deviation, or the frequency Nadir, decreases as R increases.
 - Has a seemingly large effect, but it might be simply due to R being small.
 - values over 0.1 create a response that does not look like the standard GSFR response.
- Tr: Reheat time constant
 - Almost no visible effect of RoCoF
 - The depth of the transient deviation, or the frequency Nadir, decreases as Tr increases.

These observations in isolation cannot give an understanding of which of these variables have a greater effect on the output of the models. This is where observation alongside other variables comes into play. This is done, in this project, by undergoing a Sensitivity analysis. Sensitivity analysis is a technique used to understand how changes in input variables or parameters of a model affect the output or outcomes of that model. It helps to assess the correlation and the impact each variable has on the outcome of the model. [30] [31] [32]

The process of sensitivity analysis typically involves the following steps:

1. Identify All Input Parameters: Determine the input variables of the model, in the case of the GSFR and Inverse GSFR models, the governing variables, that you want to analyse for sensitivity.

2. Define Variable Ranges: Each governing variable must have a specified range that will be used in the analysis. This involves setting a minimum and maximum value and determining their distribution pattern (i.e. Normal Distribution). [32]
3. Select Analysis Method: Several sensitivity analysis methods exist, and the applicability of them depends on the nature of the model being analysed. Common methods include: [30] [31]
 - One-at-a-time (OAT) analysis: Vary one parameter while keeping others constant.
 - Factorial design: Systematically vary multiple parameters across combinations.
 - Response surface methodology (RSM): Fit a mathematical model to understand relationships between inputs and outputs.
 - Monte Carlo simulation: Randomly sample input parameter values to analyse output variability.
4. Execute Analysis.
5. Analyse Results: Examine the output data obtained from the sensitivity analysis.
6. Interpret Findings: Lastly, interpret the sensitivity analysis results to understand which input parameters have the greatest influence on the model's output.

For this project, all governing variables were selected as the input parameters to the GSFR model to be analysed, as these are the only factors influencing the model's output and the frequency response.

Defining the range for each governing variable requires knowing the expected values in a real power grid. For this project's sensitivity analysis, we lean on Anderson's paper, and the mean values were taken from [4]. This was done because the method selected for this project was a Monte Carlo simulation, due to the previous use of Monte Carlo simulation in this project. MATLAB's Sensitivity Analysis Tool was selected for running the analysis. This tool gives control over the distribution pattern, and the number of runs, automating the process in an all-in-one package. Execution was thus handled by MATLAB. The process involved modifying the model under test to accept variables from the model space instead of the workspace, then loading the model onto the Sensitivity Analysis Tool. The variables could then be selected for random generation, the method of generation chosen (Normal Distribution with a standard deviation of 15 percent), and several thousand data points generated. In this project, a sample size of 20000 runs was used.

Analysis of the results was handled using the same Sensitivity Analysis Tool, which generated Tornado plots and Scatter plots during the run time of the Monte Carlo simulations, aiding analysis. These results could be seen through common measurements of sensitivity, such as the correlation factors for each governing variable to the outcome. Correlation factors are values between -1 and 1, where the magnitude of the factor indicates the correlation between change in the parameter and change in the output. Thus, a parameter with a correlation factor of 0 has no effect on the outcome of the model, and a parameter with a correlation factor of -1 or 1 is entirely responsible for the outcome of the model. The polarity of the correlation factor only indicates whether the relationship between the parameter and outcome is direct or inverse.

The results of the Sensitivity Analysis can be seen in figures 14 and 15. As shown, Settling (indicated by Resting) frequency is almost entirely defined by R. R also has a dominant role determining Frequency Nadir, with a correlation value of -0.7. Frequency Nadir is impacted by each variable, but Inertia is the 3rd least important variable for determining Frequency Nadir, with a correlation factor of 0.168. The next least correlated variable is almost 3 times more correlated than Inertia. R, K, and Fh are all 3 more correlated than H, by at least a factor of 3. This indicates that there exists a significant amount of room, or window, for inaccuracy within the estimation of H, as it is relatively speaking, not extremely correlated with determining the Frequency Nadir or area of the Transient Deviation.

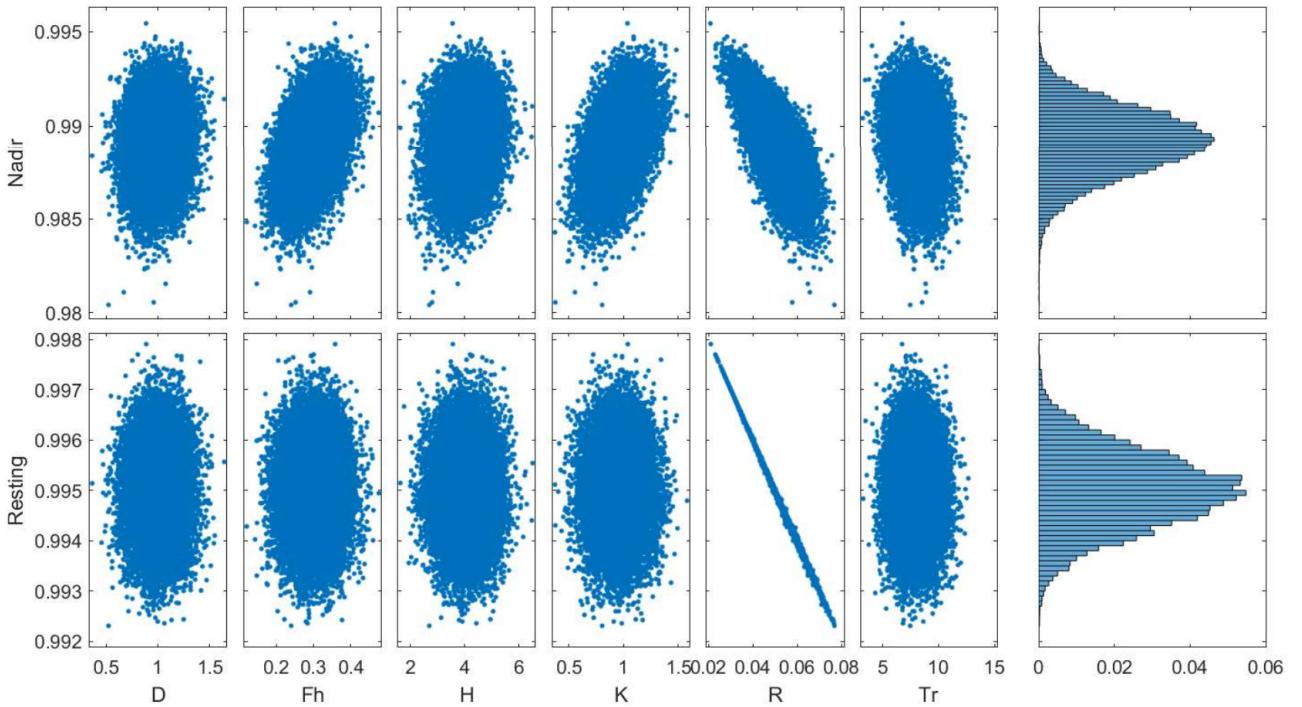


Figure 14: Scatter plot of Governing Variables against GSFR Frequency Nadir and Settling Frequency.

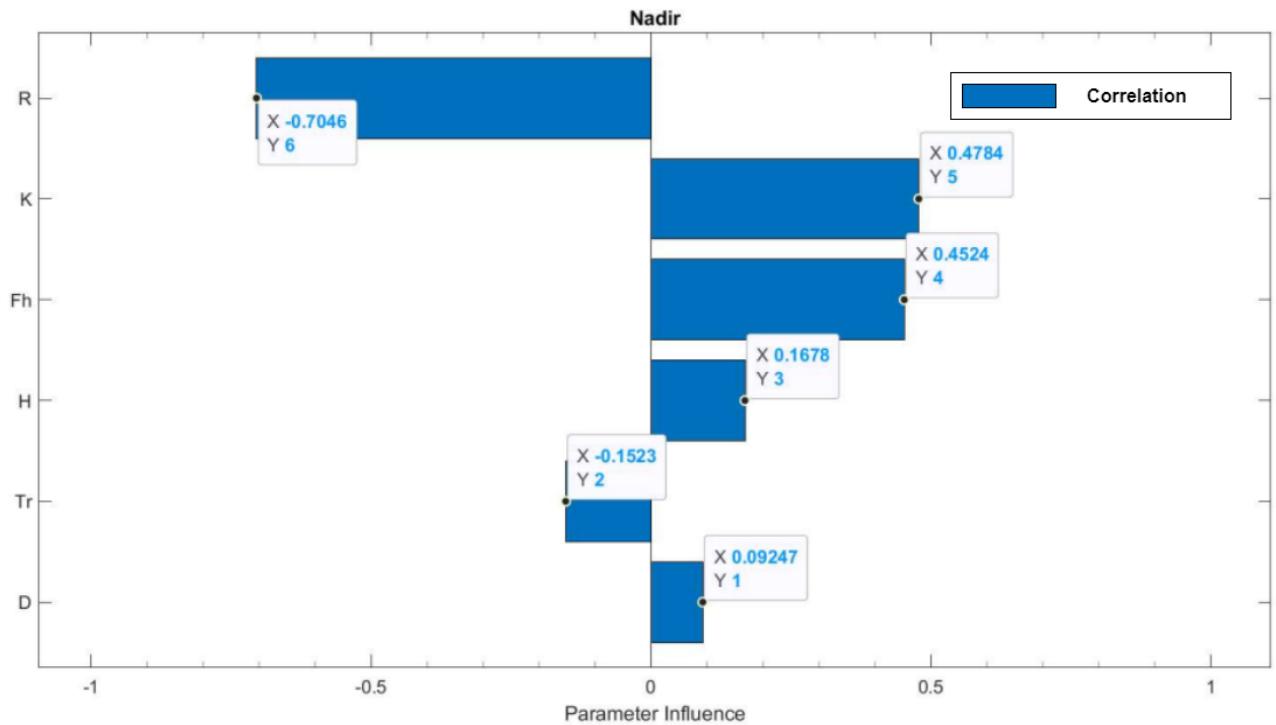


Figure 15: Tornado Plot showing the Correlation Factor between Governing Variables and minimizing Depth of Frequency Nadir.

8.2 Defining the Window/Range of H

8.2.1 Average Case

The Sensitivity Analysis indicates that a variation in H won't have as significant of an effect on the Transient Deviation as feared, and thus not as great an impact on the viability of the application of OFFC. The final objective of this project was to define an acceptable range of H from the estimated value of H, in

Variable	Value	Units
R	0.05	%
K	0.95	(A Ratio)
Fh	0.3	(A Ratio)
Tr	8	seconds
D	1	pu/Hz
P_{step}	0.1	Per Unit loss of Generation

Table 1: Values used in GSFR model

the OFFC use case. What this entails is finding the window of inaccuracy, where the real value of H can still be properly compensated by the estimated value of H. In simple terms, if the real time value of H is within X percent of the estimated value, plus or minus, then the OFFC injection will have the intended effect.

The method suggested and used in this paper to find the window of inaccuracy, was to use a set of fixed values for the governing variables and find when the smallest amount of inaccuracy of H violated the Statutory limits set, of 1 percent or ± 0.01 Hz per unit of frequency. The mean values for the governing variables were taken from [13], including the ranges specified (the mean value being the median of the range). This would define the estimated values GSFR model. A range of H was then be established, borrowing from the same book and modern data, of 1-12 seconds. A FOR loop was then constructed, for each value of this range, which followed the following methodology (also see Appendix A.1 for a verbal description):

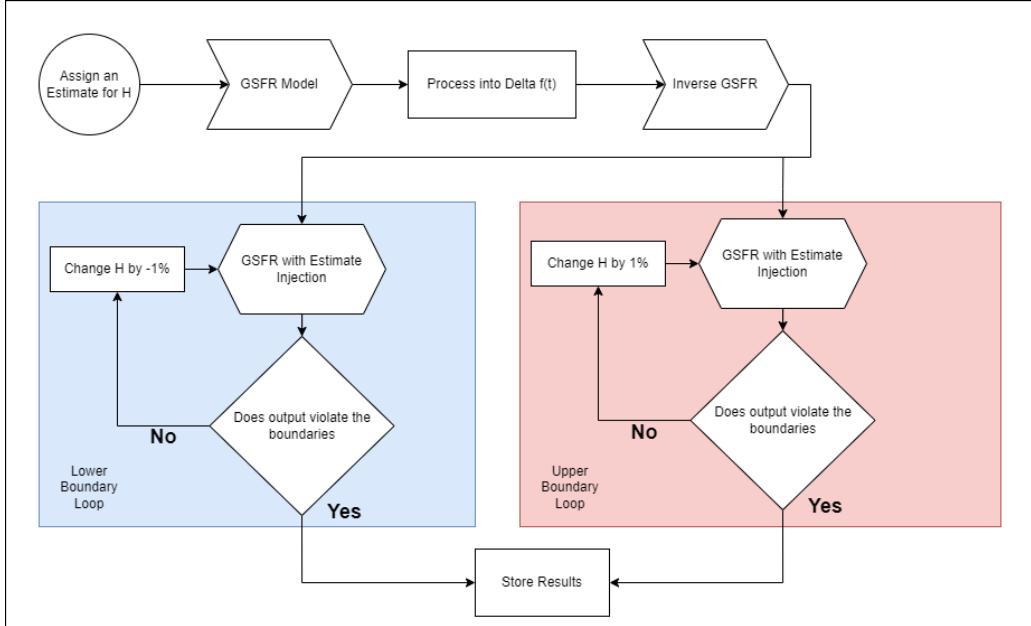


Figure 16: Flow Chart of the Range of H Code.

This approach effectively takes an assumed/estimated value of H, generates the Ideal Injection, and then checks for the greatest deviation, in percent, that the real value of H can be before the statutory limits are violated. A real value of H further away from the estimated value, would then begin incurring damage as the frequency response dips below and stays below those boundary limits. The upper and lower bounds together thus form the Window of Inaccuracy of H.

The initial proof of concept testing used the governing variable values from [4], but as mentioned in the method, this was eventually swapped to values from [13] which were deemed more appropriate.

The results of using only the mean variables from [4], found that there were no upper bounds. I.e. if the estimated value of H was 3s, then the real value of H could be any number greater than 3s, without ever violating the upper limit of the statutory boundaries. This was tested using values of H up to 5000 percent greater than the estimated value of H, where testing stopped as it was unrealistic to assume H could ever exceed a value of 50s. further tests were limited to 99 percent, upper and lower, for practical applications. The method did find lower bounds which violated the statutory limit 0.99 Hz pu for each value of H used in the range 1-12s.

Value of H (in seconds)	Accuracy Lower Bound	Accuracy Upper Bound
1	-63%	5000%
2	-63%	5000%
3	-64%	5000%
4	-64%	5000%
5	-64%	5000%
6	-64%	5000%
7	-65%	5000%
8	-65%	5000%
9	-66%	5000%
10	-66%	5000%
11	-66%	5000%
12	-67%	5000%

Table 2: Upper and Lower Bounds of accuracy of H using Anderson Values and Ideal Injection

As shown in Table 2, results indicate that for the average system, as presented by Anderson in [4], there is a minimum accuracy lower bound of 63 percent, for a Correction factor of $\beta = 1$. This accuracy window expands as the value of H increases, but as we are concerned with the penetration of RES lowering H, the lowest window of accuracy is what will be focused on. This indicates that if the real value of H, is within 63 percent of the Estimated value of H, at the time of calculating the Ideal Injection, Cortes' OFFC injection would prevent the System Frequency Response from going below the statutory limit. It may not completely correct for the transient deviation, as observed in the earliest Monte Carlo simulations, but it would prevent cumulative damage building up on parts of the grid. The Frequency response curves for the first values of H which violate the Statutory Limits, such that they are ± 1 percent greater than the values presented in table 2, can be seen in figure 17. The Statutory limits are indicated by the 2 horizontal black lines.

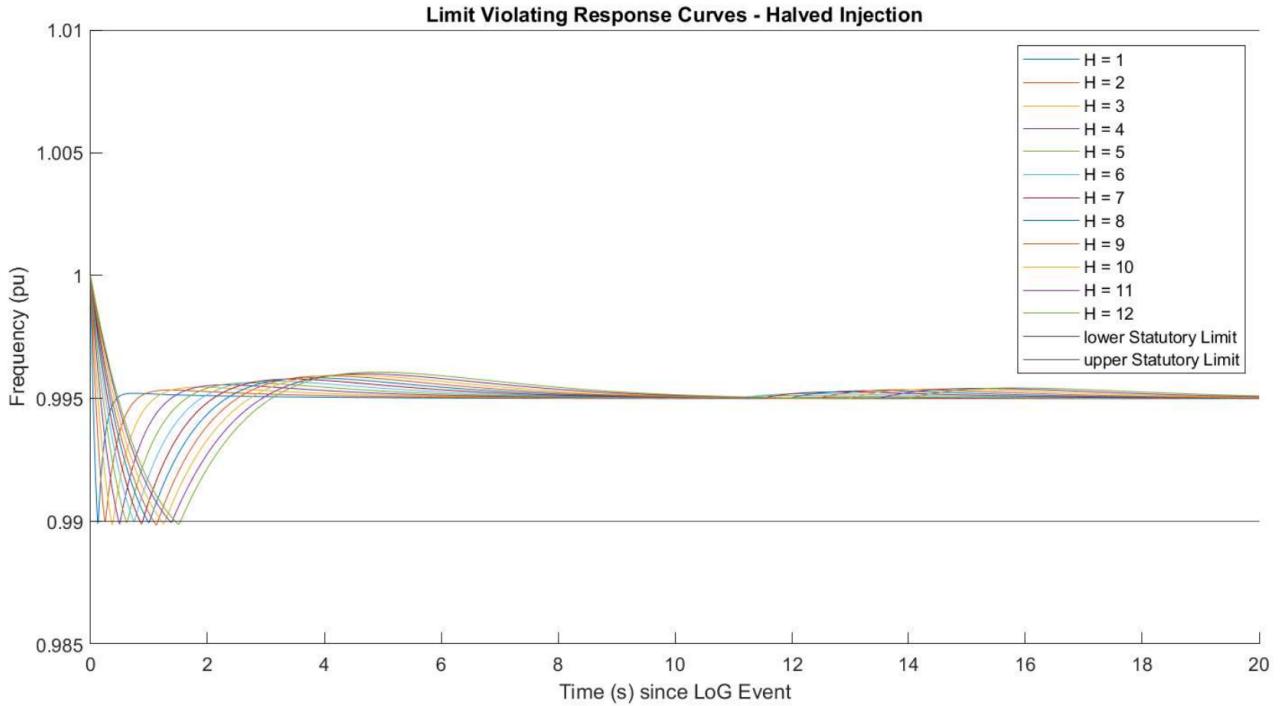


Figure 17: Limit Violating H response curve for Ideal Injection.

This process was then also repeated for the Halved Injection suggested by this paper. As can be seen by table 3, the results for both the Ideal Injection and Halved Injection are similar under the Anderson Mean values. This is important, as mentioned earlier, the Halved Injection is simpler to implement using UFLS, or injection of RES power, and uses less energy overall. The limit violating frequency response curves can be seen in figure 18.

Value of H (in seconds)	Accuracy Lower Bound	Accuracy Upper Bound
1	-63%	100%
2	-63%	100%
3	-64%	100%
4	-64%	100%
5	-64%	100%
6	-64%	100%
7	-65%	100%
8	-65%	100%
9	-66%	100%
10	-66%	100%
11	-66%	100%
12	-67%	100%

Table 3: Upper and Lower Bounds of accuracy using Anderson Values and Halved Injection



Figure 18: Limit Violating H response curve for Halved Injection.

8.2.2 Investigating the Limits with worst case and best-case scenarios

Like the previously mentioned methodology, the same process was undertaken to find the worst- and best-case scenarios, and the effects that would have on the Window of Inaccuracy of H. This would effectively present the practical limits of the inaccuracy window, in real world use. This process requires setting the values of the 3 most influential/correlated variables, identified in the Sensitivity Analysis, to their Worst Case and Best-Case values as per the book “Power System Protection” [13]. Those 3 governing variables were R, K, and Fh, with a minimum of 3 times the correlation than any other variables in the GSFR model. The greater the correlation factor, the greater impact on the resulting frequency response. Table 4 shows the new values of R, K, and Fh, that were used for the worst- and best-case testing. The results of the best-case testing, for both the Ideal and Halved Injections can be seen in tables 5 and 6.

Variable	Worst Case Value	Best-Case Value
R	0.075	0.04
K	0.8	1.0
Fh	0.2	0.4

Table 4: Values adjusted for the Worst Case and Best-Case Testing

8.2.3 Best-Case Results

Value of H (in seconds)	Accuracy Lower Bound	Accuracy Upper Bound
1	-99%	99%
2	-99%	99%
3	-99%	99%
4	-99%	99%
5	-99%	99%
6	-99%	99%
7	-99%	99%
8	-99%	99%
9	-99%	99%
10	-99%	99%
11	-99%	99%
12	-99%	99%

Table 5: Upper and Lower Bounds of accuracy of H per value of H, Ideal Injection

Value of H (in seconds)	Accuracy Lower Bound	Accuracy Upper Bound
1	-99%	99%
2	-99%	99%
3	-99%	99%
4	-99%	99%
5	-99%	99%
6	-99%	99%
7	-99%	99%
8	-99%	99%
9	-99%	99%
10	-99%	99%
11	-99%	99%
12	-99%	99%

Table 6: Upper and Lower Bounds of accuracy of H per value of H, Half Injection

In the best-case scenario, the inaccuracy is limited in code up to 99%, in either direction, lower or upper, as it cannot be 100% inaccurate for obvious reasons (H cannot be equal to 0), and previous findings indicate no upper bound exists.

8.2.4 Worst Case Results

The more interesting results can be found when looking at the worst-case scenario. The worst case defines the lowest possible window of H, that can be used in any scenario, as the window only becomes wider as the variables approach their best-case values. Thus, a worst-case window of X% would be applicable to a best-case system, but not vice versa. The results of the worst-case testing can be seen in tables 7 and 8.

Value of H (in seconds)	Accuracy Lower Bound	Accuracy Upper Bound
1	-15%	99%
2	-16%	99%
3	-16%	99%
4	-16%	99%
5	-17%	99%
6	-17%	99%
7	-17%	99%
8	-17%	99%
9	-17%	99%
10	-17%	99%
11	-17%	99%
12	-18%	99%

Table 7: Upper and Lower Bounds of accuracy of H per value of H, Ideal Injection

Value of H (in seconds)	Accuracy Lower Bound	Accuracy Upper Bound
1	0%	99%
2	0%	99%
3	0%	99%
4	0%	99%
5	0%	99%
6	-17%	99%
7	-17%	99%
8	-17%	99%
9	-17%	99%
10	-17%	99%
11	-17%	99%
12	-18%	99%

Table 8: Upper and Lower Bounds of accuracy of H per value of H, Half Injection

8.3 Analysis of H Range Results

The worst case scenario investigation is most relevant for the applicability of the OFFC method presented in [7]. Looking at the results seen in table 7, it can be seen that the smallest window of inaccuracy for H is between -15% and +99%. As previously explained, this means that as long as the REAL value of H is no more than -15% or +99% from the estimated value, the injection will prevent boundary violation. This range was found in the most extreme, worst case combination, of an Inertia value of 1 second. However, looking at the boundaries of possible combinations of the governing variables, defines the range which would also apply to every combination that is more ideal. Therefore, the results indicate that for the Ideal Injection, the range of H the investigation suggests which avoids violating the statutory boundaries can be defined as $\pm 15\%$.

As discussed earlier, the more practical and lower energy injection is the Halved injection. Looking at the results in table 8, the results are similar to the Ideal injection. The upper bound remains unchanged, and previous discoveries during this project suggest that is because there is no real upper bound.

the Halved Injection results are of interest because of the lower boundary. The results 8 show that the Halved Injection has a lower bound inaccuracy of 0% until H = 6. This is initially disappointing for the use case of the Halved Injection, as this suggests that the Halved Injection is impractical to implement, due to needing a perfectly accurate estimation of H. However, the reason for the difference becomes apparent when graphing the Violating Frequency Response curves, as seen in figures 19 and 20.

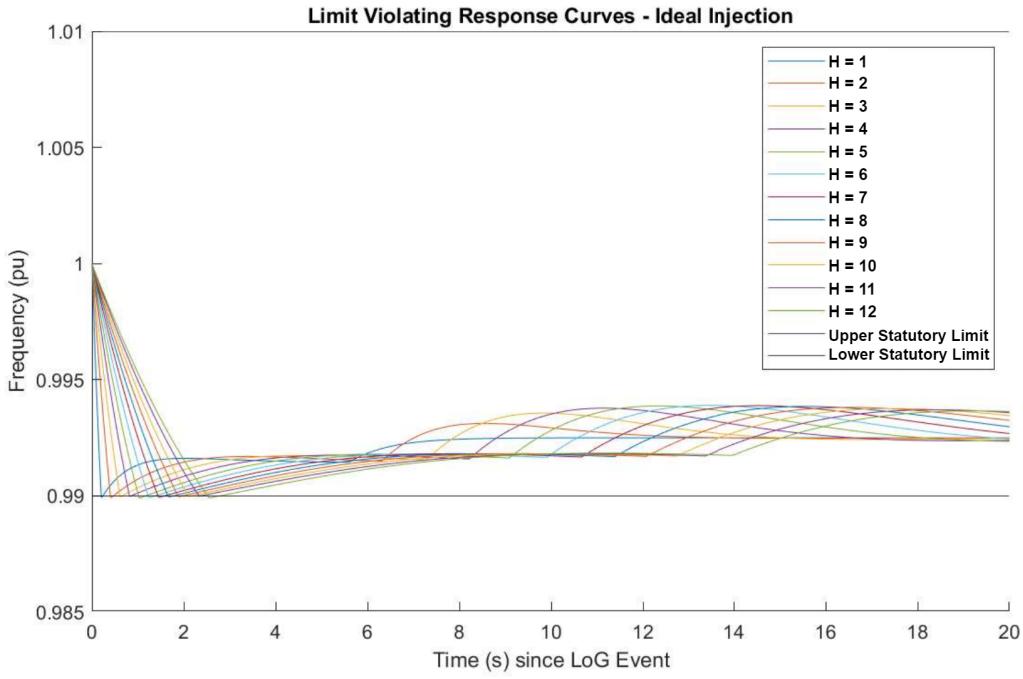


Figure 19: Limit Violating H response curve for Ideal Injection - worst case.

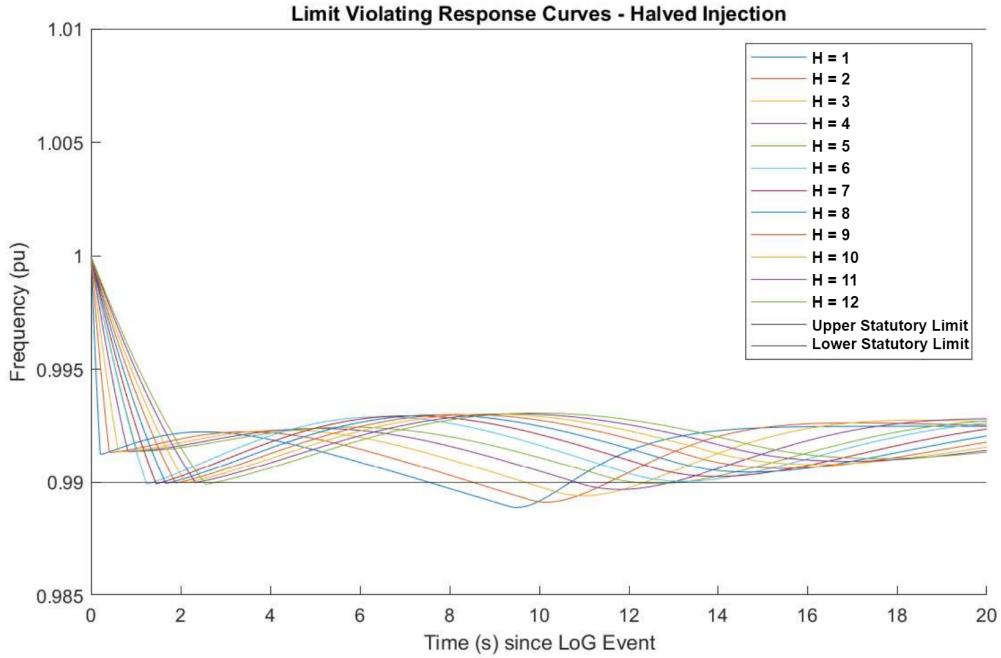


Figure 20: Limit Violating H response curve for Ideal Injection - worst case.

It can be seen that for the values $H = 1$ to $H = 5$, the Frequency Response curve does not violate the boundaries at the initial drop, or the portion of the curve that has the highest RoCoF. Instead, it can be seen that the response curve violates the lower boundary further along the time axis. The frequency response curve seems to 'dip' back down after a length of time. The reason for this can be inferred from figure 11. The Halved Injection does not inject power for the same length of time as the Ideal Injection. It also lacks the secondary triangular injection, presumably present to avoid the 'dipping' seen in figure 20. the LoG event's GSFR transient deviation will initially be compensated for, but once the injection ends, the system will fluctuate back down, to under the statutory limit. This effectively means that for a value of H between 1-5 seconds, the Halved Injection acts as a 'delay', avoiding violating the boundaries, and thus cumulative damages to anything connected to the power system, for a period of time after the LoG event, without entirely removing the transient deviation.

9 Conclusions:

9.1 Project Discoveries/Results

The main objective of this project was to investigate the acceptable range of H from the estimated value of H. This project went beyond that goal, and established a suggested alteration to the proposed OFFC injection, dubbed the Halved Injection. The range of H investigation demonstrated that the window of inaccuracy is identical between the Halved and Ideal injections, in the majority of circumstances. However, the results also revealed that the Halved injection will in practice, for values of Inertia less than 6 seconds, act as a boundary violating delay, instead of a complete correction.

However, this does not entirely remove the value of the Halved Injection. To reiterate, as can be seen in figure 11, the Halved Injection suggested requires less energy overall to correct the transient deviation, compared to the OFFC Ideal Injection. It was also shown that the practicality of deploying a Halved Injection solution, be it with RES injection or UFLS acting as a pseudo power injection, is simpler than the Ideal Injection (see figure 10 vs figure 13).

The main concern is the second dip, or the lack of the second triangular injection in the Halved Injection. The important thing to remember is that the Halved Injection would not be the only protocol in place, to avoid the consequences of a LoG event. The Halved Injection, when looked at as a part of a more complex set of protocols, retains its value proposition, because of the delay. As shown in figure 20 compared to the original frequency response in figure 2, the Halved Injection delays frequency from dipping below the statutory limit by over 7 seconds. This delay gives operators crucial extra time to address the LoG issue with more robust but slower Frequency Containment techniques. The delay provides breathing room.

The combination of the similarity between the Halved and Ideal Injections' ranges of H, and the value proposition of the delaying behaviour of the Halved Injection, this project suggests that the Window of Inaccuracy of H, for the Halved Injection, should be equivalent to the Window of Inaccuracy of H for the Ideal Injection. This projects research, simulations, and results also suggest that the Halved Injection, given the value proposition of the delaying behaviour, has several benefits over the Ideal Injection. This project thus concludes that the simulated lower range of acceptable inaccuracy of H, for which the Halved Injection is an appropriate solution, is -15% of the estimated value of H, with no upper bound.

Applying this to a real grid, and the concerns about communicating the value of H at the CoI of the power system, we can look at actual recordings of grid inertia in the National Grid, Great Britain. Using the data collected by ESO, the electricity system operator for Great Britain's National Grid, a rate of change of Inertia can be loosely established, for a real power system. The data ESO makes public is separated by days, months and years, with individual time periods between recordings set to 30 minutes. They provide 2 numbers for Inertia in the grid, Total Inertia and Market provided Inertia. The total inertia in the system "is the sum of the estimated inertia of each of the generators with a power output > 15 MW or operating to provide inertia and an estimate of the demand side contribution to system inertia" [33]. The Market provided inertia is defined as "the position of the generators recalculated to remove the impact of actions taken by National Grid ESO. These include: 1: Trades: BMU specific trades. Trades to manage constraints, mitigate RoCoF risk and voltage requirements. 2: Bid Offer Acceptance (BOAs): Instructions from National Grid ESO to a participant in the balancing mechanism. Participants must then act to ensure that their BM Units produce the required level of output" [33]. In simpler terms, Market Provided Inertia is the inertia of the grid without any Frequency Containment Methods/Protocols, thus without any added or removed inertia. Both of these are in gigavolt amp seconds (GVAs).

After comparing the data between provided by ESO, from the dates 04/01/2024 to 08/05/2024, the maximum negative change in Inertia, for the Total Inertia was -17.14% over a period of 30 minutes, with an average of -2.45%. The maximum positive change is less relevant due to the appearance of no upper bound in the Window of Inaccuracy. For the Market Provided Inertia, the maximum negative change was -18.87% over a period of 30 minutes, with an average of -2.83%. Given the Window of Inaccuracy this project found was -15% in the worst case, as long as the Inertia value does not decrease by over 15% between measurements of Inertia are made, the Halved Injection is suitable. For this reason, the data suggests that, for the National Grid, a suitable rate of communication and measuring Inertia is just a bit less than once per 30 minutes. As the average change is only around -3%, and the scatter plot of change, figure 21, shows the majority of changes within a 30 minute interval are less than -15%, this paper suggests that the suitable rate of communication and measuring Inertia is somewhere between 25 to 30 minutes, inclusive.

Category	Maximum negative change	Average negative change	Maximum positive change
Total Inertia	-17.14%	-2.45%	19.63%
Market Inertia	-18.87%	-2.83%	21.12%

Table 9: Rate of Change in Inertia, National Grid, per 30 minutes.

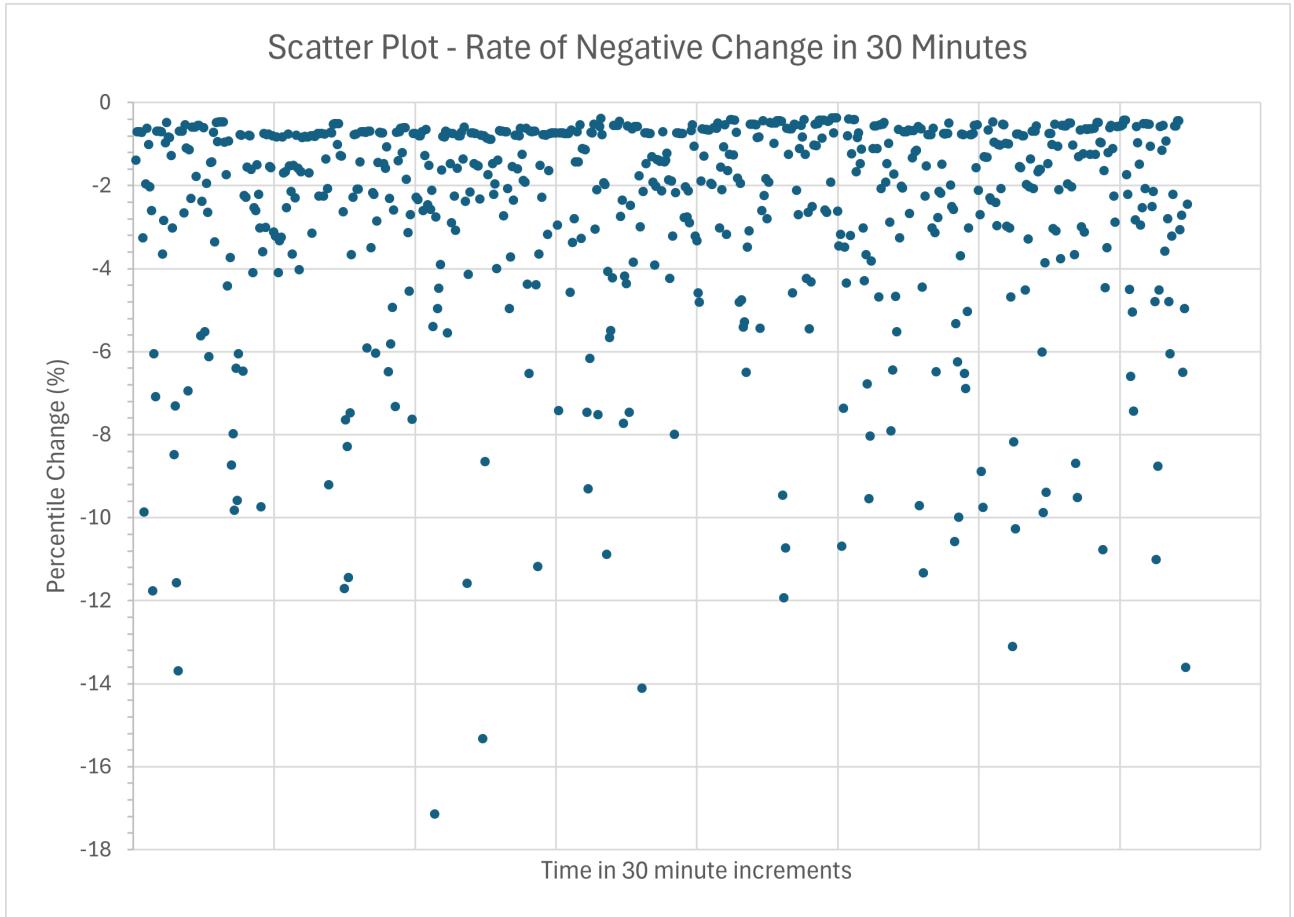


Figure 21: Scatter Plot of the Negative changes of Inertia, per 30 minute period, in the National Grid.

9.2 Further Work

As was noted at the start of this report, "All models are wrong, but some are useful." [2]. All the research and simulations done in this project were done using the GSFR model suggested in [4], in combination with the injection proposed in [7]. These came with assumptions and simplifications, which made them useful, but not completely accurate to a real power system with thousands of synchronous generators, interconnects between regions, etc. For this reason, the results and assertions made in this report can only go so far. They are an indication of what, in theory, is true, but not yet "tried and tested". It is a logical process to then suggest that further work should be done looking into these findings, in a model with fewer assumptions. The logical extension would be to apply this research and theory to a model such as the IEEE 39-bus test system [7].

It would be important to discover if the range were to expand, or constrict, depending on the complexity of the power system. Whether injecting using UFLS to create pseudo power results in the same response. It would also be of value to discover which methods of Frequency Containment would best benefit from the delay introduced by the Half Injection, even at the worst cases.

Lastly, the conclusions drawn from this project would benefit greatly from a set of Inertia data with a smaller time period between recordings. This would enable a firmer time window to be drawn. However, this would need to be collected for every power system the Halved Injection OFFC is used in, as different measuring periods would need to be used on power systems with different rates of change of Inertia.

9.3 Reflection

As mentioned in my Interim report, there was an unexpected need for me to become reacquainted with not having a 9-5 work schedule, as I experienced during my placement year. This originally threw me for a loop, and made progress sporadic. After writing the interim report, and reflecting on the underlying problem, I came to learn how important that sectioning off of time is for my productivity and project management. It is a lot easier to work on a project when you have a different space and time dedicated to it. For that reason, I constructed a schedule, very akin to a 9-5, which dedicated roughly 17.5 hours per work week, in the library, for this MEng project. It also allowed me to better balance the load of this project with the sudden projects or assignments in other courses. I had a buffered time slot which belonged only to this project, and any time needed for other tasks/daily life/coursework or extra work for this MEng project could be allocated by necessity on the weekends.

Giving myself soft deadlines was also beneficial. Having a target to hit makes it much easier to sit down, and crack on with work while ignoring distractions. It also meant that when actual deadlines came, I was not overly stressed or unable to handle the workload as I had already completed sections. I believe my project management skills, when it's a solo project and no one else is overseeing my progress, has improved. My ability to self motivate and my determinism have been aided by that growth in my time management and project management skill set.

References

- [1] P. Denholm et al. “Inertia and the Power Grid: A guide without the spin”. In: *arXiv* (May 2020). DOI: 10.2172/1659820.
- [2] M. Dreidy, H. Mokhlis, and S. Mekhilef. “Inertia response and frequency control techniques for renewable energy sources: A review”. In: *Renewable & Sustainable Energy Reviews* 69 (Mar. 2017), pp. 144–155. DOI: 10.1016/j.rser.2016.11.170.
- [3] L. Mehigan et al. “Renewables in the European power system and the impact on system rotational inertia”. In: *Energy* 203 (July 2020), p. 117776. DOI: 10.1016/j.energy.2020.117776.
- [4] P. M. Anderson and M. Mirheydar. “A low-order system frequency response model”. In: *IEEE Transactions on Power Systems* 5.3 (Jan. 1990), pp. 720–729. DOI: 10.1109/59.65898.
- [5] S. Y. Çalışkan and P. Tabuada. “Uses and abuses of the swing equation model”. In: *IEEE*. Dec. 2015. DOI: 10.1109/cdc.2015.7403268.
- [6] K. Lohana. “Enhanced Fast Frequency Response Strategy for Power Systems with Volatile Inertia”. University of Leeds, 2023.
- [7] Cortes, Jegarluei, and Azizi. “Targeted Fast Frequency Response by Decomposing Frequency into Transient and Steady-State Deviations”. In: *IEEE International Conference on Energy Technologies for Future Grids*. In press. 2023.
- [8] M. Krpan and I. Kuzle. “Introducing low-order system frequency response modelling of a future power system with high penetration of wind power plants with frequency support capabilities”. In: *IET Renewable Power Generation* 12.13 (Apr. 2018), pp. 1453–1461. DOI: 10.1049/iet-rpg.2017.0811.
- [9] Lennart Ljung. “On the estimation of transfer functions”. In: *Automatica* 21.6 (1985), pp. 677–696. DOI: 10.1016/0005-1098(85)90042-1. URL: <https://www.sciencedirect.com/science/article/pii/0005109885900421>.
- [10] MathWorks. *SM Governor with Droop*. <https://uk.mathworks.com/help/sps/ref/smgoovernorwithdroop.html>.
- [11] Frequency Containment Reserve FCR definition. <https://nanoenergies.eu/knowledge-base/frequency-control-reserve-fcr>.
- [12] Amin Nassaj et al. “Fast Linear State Estimation for Unbalanced Distribution Systems Using Hybrid Measurements”. In: *IEEE Transactions on Instrumentation and Measurement* 73 (2024), pp. 1–10. DOI: 10.1109/TIM.2024.3385837.
- [13] P. M. Anderson and J. Wiley. *Power System Protection*. New York, NY; Piscataway, NJ: McGraw-Hill, 1999.
- [14] Y. Guo and T. H. Summers. “A Performance and Stability Analysis of Low-inertia Power Grids with Stochastic System Inertia”. In: *IEEE* (July 2019). DOI: 10.23919/acc.2019.8814402.
- [15] Fernández-Guillamón et al. “Power systems with high renewable energy sources: A review of inertia and frequency control strategies over time”. In: *Renewable and Sustainable Energy Reviews* 115 (Nov. 2019), p. 109369. DOI: 10.1016/j.rser.2019.109369.
- [16] J. Grainger and W. Stevenson Jr. *Power System Analysis*. McGraw-Hill Series in Electric, 1994.
- [17] C. Adamson, L. Barnes, and B. D. Nellist. “The automatic solution of power-system swing-curve equations”. In: *The Proceedings of the Institution of Electrical Engineers* 104.14 (Jan. 1957), p. 143. DOI: 10.1049/pi-a.1957.0036.
- [18] T. Inoue et al. “Estimation of power system inertia constant and capacity of spinning-reserve support generators using measured frequency transients”. In: *IEEE Transactions on Power Systems* 12.1 (Jan. 1997), pp. 136–143. DOI: 10.1109/59.574933.
- [19] J. Zhou and Y. Ohsawa. “Improved swing equation and its properties in synchronous generators”. In: *IEEE Transactions on Circuits and Systems I-regular Papers* 56.1 (Jan. 2009), pp. 200–209. DOI: 10.1109/tcsi.2008.924895.
- [20] S. Rama Mohan. “Impact of Grid Frequency Variations on Power Plant Performance”. In: *International Journal of Engineering Research and Applications* 5.8 (Aug. 2015).
- [21] K. Mehta and S. Sharma. “Impact of Grid Frequency Fluctuations on Power System Stability”. In: *International Journal of Innovative Research in Science, Engineering and Technology* 3.5 (May 2014).
- [22] M. Singh and A. Verma. “Impact of Grid Frequency Variations on Voltage Stability”. In: *International Journal of Emerging Technology and Advanced Engineering* 4.7 (July 2014).
- [23] D. S. Oliveira et al. “Effects of Power Quality Disturbances on Sensitive Electronic Equipment”. In: *IEEE Transactions on Power Delivery* 21.3 (July 2006).
- [24] P. Kumar and R. Gupta. “Impact of Frequency Deviations on Industrial Equipment”. In: *International Journal of Electrical Power & Energy Systems* 78 (May 2016).
- [25] L. Wang et al. “Impact of Grid Frequency Fluctuations on Power System Reliability”. In: *IEEE Transactions on Power Systems* 30.3 (May 2015).
- [26] Under frequency load shedding. <https://aemo.com.au/en/energy-systems/electricity/wholesale-electricity-market-wem/system-operations/under-frequency-load-shedding>.

- [27] R. Sobbouhi and A. Vahedi. "Transient stability prediction of power system; a review on methods, classification and considerations". In: *Electric Power Systems Research* 190 (Jan. 2021), p. 106853. DOI: 10.1016/j.epsr.2020.106853.
- [28] K. Das et al. "Inertia dependent droop based frequency containment process". In: *Energies* 12.9 (Apr. 2019), p. 1648. DOI: 10.3390/en12091648.
- [29] M. Rezkalla et al. "Comparison between synthetic inertia and fast frequency containment control based on single phase EVs in a microgrid". In: *Applied Energy* 210 (Jan. 2018), pp. 764–775. DOI: 10.1016/j.apenergy.2017.06.051.
- [30] H. Christopher Frey and Sumeet R. Patil. "Identification and Review of Sensitivity Analysis Methods". In: *Risk Analysis* 22.3 (2002), pp. 553–578. DOI: <https://doi.org/10.1111/0272-4332.00039>.
- [31] T Lenhart et al. "Comparison of two different approaches of sensitivity analysis". In: *Physics and Chemistry of the Earth, Parts A/B/C* 27.9 (2002), pp. 645–654. DOI: [https://doi.org/10.1016/S1474-7065\(02\)00049-9](https://doi.org/10.1016/S1474-7065(02)00049-9).
- [32] MATLAB. *version (R2019b)*. Natick, Massachusetts: The MathWorks Inc., 2024.
- [33] National Grid Electricity System Operator (ESO). *GB System Inertia - 2024-2025*. <https://www.nationalgrideso.com/data-portal/system-inertia/gb-system-inertia-2024-2025>. [Online; accessed 10-May-2024]. 2024.

A Appendix

A.1 Functional Flow Chart of Range of H finder Program

1. Run the GSFR model with the mean values and the value of H [1-12 in ascending order]. This would be the Estimated GSFR model at the time of a LoG event, mimicking the data grid operators would have. To maintain applicability, a LoG event which has a settling frequency above the statutory limit had to be selected, or else correcting the transient dip would be fruitless as the system remains under the statutory limit anyway. The Pstep selected was 0.1.
2. Process the Estimated GSFR output and run it through the Inverse GSFR model with a correction factor of 1, to find the OFFC Ideal Injection for that value of H, which completely corrects for the transient deviation in the Estimated GSFR model.
3. Run two FOR loops, acting as lower and upper bound finders. These loops change the value of H by 1 percent at a time, or less for finer control, and then reruns the GSFR model with the Ideal Injection and the new H value. The output of which is then compared to the statutory limit of 0.99 Hz and 1.01 Hz per unit. If the lowest or highest point of the GSFR model's output violates the statutory limits, store the previous percentage value as it did not violate the boundaries.

A.2 Base M File

```
1 %Initialization of the experiment
2 %Correction Factor Value:
3 B = 0.9;
4 %Pd or P step loss in the case of an LoG event
5 Pd = -0.3;
6
7 %Part 1: calculate GSFR(s) and the BASE CASE
8 model1 = "GSFR_with_P_sigIn";
9 open_system(model1);
10 %load an empty signal for Power injection (SigIn)
11 SigIn = [0 0];
12
13 %run the Simulink Model for GSFR, store output
14 output = sim(model1);
15 GSFR_output = output.simout;
16 %the Data is hidden in GSFR_output's sub section of "simout" (Plotted for
17 %ease of visibility)
18 %plot(GSFR_output);
19
20
21
22 %Part 2: establishing the signals needed
23 %Need to establish Tau and Tau2, using the value of B (correction factor)
24 %and the Dtr, Settling frequency, and frequency nadir.
25 %can then manually do the unit step function work with a for loop
26 Nadir = min(GSFR_output);
27 settling_Frequency = GSFR_output.Data(end, :);
28 Dtr = settling_Frequency - Nadir;
29 New_Nadir = B*Dtr + Nadir;
30
31 %Finding tau
32 tau = GSFR_output.Time(find(GSFR_output.Data < Dtr*B+Nadir,1,'first'));
33 tau_Index = find(GSFR_output.Data < Dtr*B+Nadir,1,'first');
34 %Finding tau2
35 tau2 = GSFR_output.Time(find(GSFR_output.Data < Dtr*B+Nadir,1,'last'));
36 tau2_Index = find(GSFR_output.Data < Dtr*B+Nadir,1,'last');
```

```

39 %creating the compensating injection frequency
40 dft = GSFR_output;
41 for i = 1 : length(dft.Data)
42     if(i >= tau_Index && i <= tau2_Index)
43         dft.Data(i) = New_Nadir - GSFR_output.Data(i);
44     else
45         dft.Data(i) = 0;
46     end
47 end
48
49
50 %Part 3: running this through a Simulink Model which does the Laplace
51 %transform and Inversve Laplace transform using time series dft and GSFR
52 model2 = "inverse_GSFR";
53 open_system(model2);
54 pt_output = sim(model2);
55
56
57 hold on
58 plot(dft) %Just proving it works and looks right
59 xlim([0 20]);
60 ylim([0 0.2]);
61 plot(pt_output.simout);
62 hold off
63
64
65 % Injection_File = load("Correction_half");
66 % SigIn = Injection_File.SigIn;

```

A.3 Monte Carlo Example Code

```

1 %Part 0: Initialization of the experiment
2 %Initialize Global Variables
3 Percent_Correction = 80;
4 Pd = -0.3; %the LoG event
5 SigIn = [0 0]; %load an empty signal for Power injection (SigIn)
6 Total_Num_Runs = 10;
7 %Initialize Storage Variables:
8 results = cell(2,Total_Num_Runs+2);
9 run_num = 2;
10 %make folder to save output files to
11 folder_title = sprintf('results_B_%d_Runs_%d',Percent_Correction,
12     Total_Num_Runs);
13 mkdir (folder_title);
14
15 %Initialize Simulink Models
16 GSFR_model = "GSFR_Individual_Vars";
17 load_system(GSFR_model); %load the GSFR Model with Workspace Variables
18 inverse_GSFR_model = "inverse_GSFR_Individual_Vars";
19 load_system(inverse_GSFR_model); %load the inverse GSFR Model
20
21 %Progress Bar for the Longer Monte Carlo Sims
22 clf; %Clear Plot
23 progress = waitbar(0, 'running simulation');
24
25 %Part 1: Running The Assumed Values GSFR model
26 %Initialize variables of the ORIGINAL SFR model (assumed network values)
27 R = 0.075;
28 H = 5.5;

```

```

29 K = 0.95;
30 Fh = 0.25;
31 Tr = 9.0;
32 D = 1.0;
33 B = Percent_Correction/100; %Correction Factor Value:
34
35
36 %Part 1.1: calculate GSFR(s) and the BASE CASE
37 %calculate the extra variables
38 wn = sqrt((D*R+K)/(2*H*R*Tr));
39 c = (((2*H*R)+((D*R)+(K*Fh))*Tr))/(2*(D*R+K)))*wn;
40 %polynomial variables
41 a1 = (D*R+K)*2*c*wn;
42 b1 = wn^2;
43 c1 = (wn^2)*R*Tr;
44 d1 = (wn^2)*R;
45 x1 = D*R+K;
46
47 %run the Simulink Model for GSFR, store output
48 output = sim(GSFR_model);
49 GSFR_output = output.simout;
50 results{1,1} = GSFR_output; %store the results for Plotting Later
51 results{2,1} = [R H K Fh Tr D];
52
53
54 %Part 2: establishing the signals needed
55 %Need to establish Tau and Tau2, using the value of B (correction factor)
56 %and the Dtr, Settling frequency, and frequency nadir.
57 %can then manually do the unit step function work with a for loop
58 Nadir = min(GSFR_output);
59 settling_Frequency = 1 + ((R*Pd)/(D*R+K));
60 Dtr = settling_Frequency - Nadir;
61 New_Nadir = B*Dtr + Nadir;
62
63 %Finding tau
64 tau = GSFR_output.Time(find(GSFR_output.Data < Dtr*B+Nadir,1,'first'));
65 tau_Index = find(GSFR_output.Data < Dtr*B+Nadir,1,'first');
66 %Finding tau2
67 tau2 = GSFR_output.Time(find(GSFR_output.Data < Dtr*B+Nadir,1,'last'));
68 tau2_Index = find(GSFR_output.Data < Dtr*B+Nadir,1,'last');
69
70 %creating the compensating injection frequency
71 dft = GSFR_output;
72 for i = 1 : length(dft.Data)
73     if(i >= tau_Index && i <= tau2_Index)
74         dft.Data(i) = New_Nadir - GSFR_output.Data(i);
75     else
76         dft.Data(i) = 0;
77     end
78 end
79
80
81 %Part 2: running Delta f(t) through Inverse GSFR for injection p(t)
82 inverse_output = sim(inverse_GSFR_model);
83 SigIn = inverse_output.simout; %store the results for Monte Carlo
84
85 %Plot and Save Injection for Postarity
86 plot(SigIn);
87 plot_title = sprintf('Power Injection for B = %f',B);
88 title(plot_title);
89 xlabel('Time (s) since LoG Event')
90 ylabel('Active Power (pu)')

```

```

91 file_title1 = sprintf('Corrective_Power_Injection_for_B_%d.png',
92     Percent_Correction,Total_Num_Runs);
93 saveas(gcf,file_title1);
94 movefile(file_title1, folder_title);
95 clf;
96
97 %Run Injection through GSFR to get the Base Case response with Injection
98 output = sim(GSFR_model);
99 GSFR_output = output.simout;
100 results{1,2} = GSFR_output; %store the results for Plotting later
101 results{2,2} = [R H K Fh Tr D];
102
103
104
105
106
107 %Part 3: Monte Carlo Runs
108 %Insert the Power Injection from Part 2 into a GSFR with Random Values
109 %Determined by a Normal Distribution for each Variable or System condition
110 for i = 1 : Total_Num_Runs
111     %Increment Run Number
112     run_num = run_num + 1;
113     %Randomize the Variables by Normal Distribution
114     R = (0.05/3)*randn()+0.075;
115     H = 1.17*randn()+5.5; %mean of 5.5 with dev 1.17
116     K = 0.95;
117     Fh = 0.034*randn()+0.25;
118     Tr = 0.67*randn()+9;
119     D = 1.0;
120     wn = sqrt((D*R+K)/(2*H*R*Tr));
121     c = (((2*H*R)+((D*R)+(K*Fh))*Tr))/(2*(D*R+K)))*wn;
122
123     %Run the GSFR model with new Variable values
124     output = sim(GSFR_model);
125     GSFR_output = output.simout;
126     results{1,run_num} = GSFR_output; %store the results for Plotting later
127     results{2,run_num} = [R H K Fh Tr D];
128
129
130     %display progress
131     waitbar((1/Total_Num_Runs)*(run_num/2),progress);
132 end
133
134
135
136 %Update progress bar
137 waitbar(0.5,progress,'Simulations Complete -> Plotting results');
138 %Plot Results for Saving
139 hold on
140 for i = 3 : length(results) %Plot all the results in one Figure
141     plot(results{1,i},':');
142     waitbar(0.5+(1/Total_Num_Runs)*((i-2)/2),progress);
143 end
144 close(progress);
145 %Plot the Mean result last so it sits on top and is most obvious
146 plot(results{1,1},'k--');
147 plot(results{1,2},'k');
148 %Add appropriate title and axis labels
149 plot_title = sprintf('Envelope f(t) for B = %f with %d runs',B,
150     Total_Num_Runs);
151 title(plot_title);

```

```

151 xlabel('Time (s) since LoG Event')
152 ylabel('Frequency (pu)')
153 hold off
154
155
156
157 %Save results to figure file, PNG, and raw Cell Array
158 file_title = sprintf('B_%d_Runs_%d',Percent_Correction,Total_Num_Runs);
159 saveas(gcf,file_title);
160 movefile([file_title '.fig'], folder_title);
161
162 file_title1 = sprintf('B_%d_Runs_%d.png',Percent_Correction,Total_Num_Runs);
163 saveas(gcf,file_title1);
164 movefile(file_title1, folder_title);
165
166 file_title2 = sprintf('B_%d_Runs_%d.mat',Percent_Correction,Total_Num_Runs);
167 save(file_title2, "results");
168 movefile(file_title2, folder_title);

```

A.4 Range of H Ideal Injection Code

```

1 %-----Initialize the Experiment and Models-----
2 GSFR_model = "GSFR_Individual_Vars";
3 load_system(GSFR_model); %load the GSFR Model with Workspace Variables
4 inverse_GSFR_model = "inverse_GSFR_Individual_Vars";
5 load_system(inverse_GSFR_model); %load the inverse GSFR Model
6 %-----Initialize the Experiment and Models-----
7
8
9
10
11 %-----Initialize Experiment Variables-----
12 Percent_Correction = 100;
13 Pd = -0.1; %the LoG event
14 SigIn = [0 0]; %load an empty signal for Power injection (SigIn)
15
16 %Initialize Storage Variables:
17 results = cell(5,12);
18 %-----Initialize Experiment Variables-----
19
20
21
22
23
24 %-----Initialize Progress Tracker-----
25 clf; %Clear Plot
26 progress = waitbar(0, 'running simulation');
27 %-----Initialize Progress Tracker-----
28
29
30 for j = 1: 12
31
32 %-----Part 1: Estimated GSFR Response-----
33 %Initialize Estimated Variables (See Anderson's Paper)
34 SigIn = [0 0];
35 H = j;
36 R = 0.04;
37 K = 1;
38 Fh = 0.4;
39 Tr = 8.0;

```

```

40 D = 1.0;
41 B = Percent_Correction/100; %Correction Factor Value:
42 %calculate the extra variables
43 wn = sqrt((D*R+K)/(2*H*R*Tr));
44 c = (((2*H*R)+(((D*R)+(K*Fh))*Tr))/(2*(D*R+K)))*wn;
45 %polynomial variables
46 a1 = (D*R+K)*2*c*wn;
47 b1 = wn^2;
48 c1 = (wn^2)*R*Tr;
49 d1 = (wn^2)*R;
50 x1 = D*R+K;
51 %Run the Model
52 output = sim(GSFR_model);
53 GSFR_output = output.simout;
54 results{1,j} = [H R K Fh Tr D];
55 %-----Part 1: Estimated GSFR Response-----
56
57
58
59
60 %-----Part 2: Finding Injection for Estimated Values-----
61 %Find Nadir, Establish Tau and Tau2
62 Nadir = min(GSFR_output);
63 settling_Frequency = 1 + ((R*Pd)/(D*R+K));
64 Dtr = settling_Frequency - Nadir;
65 New_Nadir = B*Dtr + Nadir;
66
67 %Finding tau
68 tau = GSFR_output.Time(find(GSFR_output.Data < Dtr*B+Nadir,1,'first'));
69 tau_Index = find(GSFR_output.Data < Dtr*B+Nadir,1,'first');
70 %Finding tau2
71 tau2 = GSFR_output.Time(find(GSFR_output.Data < Dtr*B+Nadir,1,'last'));
72 tau2_Index = find(GSFR_output.Data < Dtr*B+Nadir,1,'last');
73
74 %Manually applying Unit Step Function
75 dft = GSFR_output;
76 for i = 1 : length(dft.Data)
77 if(i >= tau_Index && i <= tau2_Index)
78 dft.Data(i) = New_Nadir - GSFR_output.Data(i);
79 else
80 dft.Data(i) = 0;
81 end
82 end
83
84 %Run the Delta f(t) through Inverse GSFR for injection p(t)
85 inverse_output = sim(inverse_GSFR_model);
86 SigIn = inverse_output.simout; %store the results for Monte Carlo
87 %-----Part 2: Finding Injection for Estimated Values-----
88
89
90
91
92 %-----Part 3: the Experiment-----
93 %Using Injection, check the lower bounds for H's innaccuracy
94 %store the last value that does not violate Statutory Limits
95 for i = 1 : 99
96 H = j-((i/100)*j);
97 wn = sqrt((D*R+K)/(2*H*R*Tr));
98 c = (((2*H*R)+(((D*R)+(K*Fh))*Tr))/(2*(D*R+K)))*wn;
99
100 %Run the GSFR model with new Variable values
101 output = sim(GSFR_model);

```

```

102         GSFR_output = output.simout;
103
104         if min(GSFR_output) > 0.99
105             results{2,j} = i;
106         else
107             results{4,j} = GSFR_output;
108             break;
109         end
110     end
111
112
113 %Using Injection, check the upper bounds for H's innaccuracy
114 %store the last value that does not violate Statutory Limits
115 for i = 1 : 99
116     H = j+((i/100)*j);
117     wn = sqrt((D*R+K)/(2*H*R*Tr));
118     c = (((2*H*R)+((D*R)+(K*Fh))*Tr))/(2*(D*R+K)))*wn;
119
120     %Run the GSFR model with new Variable values
121     output = sim(GSFR_model);
122     GSFR_output = output.simout;
123
124     if max(GSFR_output) < 1.01
125         results{3,j} = i;
126     else
127         results{5,j} = GSFR_output;
128         break;
129     end
130 end
131 %-----Part 3: the Experiment-----
132 %display progress
133 waitbar((1/12)*(j),progress);
134
135 hold on
136 for i = 1 : length(results) %Plot all the results in one Figure
137     plot(results{4,i});
138     plot(results{5,i});
139 end
140 yline(0.99, 'k', 'DisplayName', 'lower Statutory Limit');
141 yline(1.01, 'k', 'DisplayName', 'upper Statutory Limit');
142 hold off
143
144 %-----Saving Results-----
145 folder_title = sprintf('results_ideal_R_%f_K_%f_Fh_%f',R,K,Fh);
146 mkdir (folder_title);
147
148 %Save results to figure file, PNG, and raw Cell Array
149 file_title2 = sprintf('boundaries.mat');
150 save(file_title2, "results");
151 movefile(file_title2, folder_title);
152 %-----Saving Results-----
153
154 close(progress);

```

A.5 Range of H Halved Injection Code

```

1 %-----Initialize the Experiment and Models-----
2 GSFR_model = "GSFR_Individual_Vars";
3 load_system(GSFR_model); %load the GSFR Model with Workspace Variables

```

```

4 inverse_GSFR_model = "inverse_GSFR_Individual_Vars";
5 load_system(inverse_GSFR_model); %load the inverse GSFR Model
6 %-----Initialize the Experiment and Models-----
7
8
9
10
11 %-----Initialize Experiment Variables-----
12 Percent_Correction = 100;
13 Pd = -0.1; %the LoG event
14 SigIn = [0 0]; %load an empty signal for Power injection (SigIn)
15
16 %Initialize Storage Variables:
17 results = cell(5,12);
18 %-----Initialize Experiment Variables-----
19
20
21
22
23
24 %-----Initialize Progress Tracker-----
25 clf; %Clear Plot
26 progress = waitbar(0, 'running simulation');
27 %-----Initialize Progress Tracker-----
28
29
30
31 for j = 1: 12
32
33 %-----Part 1: Estimated GSFR Response-----
34 %Initialize Estimated Variables (See Anderson's Paper)
35 SigIn = [0 0];
36 H = j;
37 R = 0.075;
38 K = 0.8;
39 Fh = 0.2;
40 Tr = 8.0;
41 D = 1.0;
42 B = Percent_Correction/100; %Correction Factor Value:
43 %calculate the extra variables
44 wn = sqrt((D*R+K)/(2*H*R*Tr));
45 c = (((2*H*R)+((D*R)+(K*Fh))*Tr))/(2*(D*R+K))*wn;
46 %polynomial variables
47 a1 = (D*R+K)*2*c*wn;
48 b1 = wn^2;
49 c1 = (wn^2)*R*Tr;
50 d1 = (wn^2)*R;
51 x1 = D*R+K;
52 %Run the Model
53 output = sim(GSFR_model);
54 GSFR_output = output.simout;
55 results{1,j} = [H R K Fh Tr D];
56 %-----Part 1: Estimated GSFR Response-----
57
58
59
60
61 %-----Part 2: Finding Injection for Estimated Values-----
62 %Find Nadir, Establish Tau and Tau2
63 Nadir = min(GSFR_output);
64 settling_Frequency = 1 + ((R*Pd)/(D*R+K));
65 Dtr = settling_Frequency - Nadir;

```

```

66 New_Nadir = B*Dtr + Nadir;
67
68 %Finding tau
69 tau = GSFR_output.Time(find(GSFR_output.Data < Dtr*B+Nadir,1,'first'));
70 tau_Index = find(GSFR_output.Data < Dtr*B+Nadir,1,'first');
71 %Finding tau2
72 tau2 = GSFR_output.Time(find(GSFR_output.Data < Dtr*B+Nadir,1,'last'));
73 tau2_Index = find(GSFR_output.Data < Dtr*B+Nadir,1,'last');
74
75 %Manually applying Unit Step Function
76 dft = GSFR_output;
77 for i = 1 : length(dft.Data)
78     if(i >= tau_Index && i <= tau2_Index)
79         dft.Data(i) = New_Nadir - GSFR_output.Data(i);
80     else
81         dft.Data(i) = 0;
82     end
83 end
84
85 %Run the Delta f(t) through Inverse GSFR for injection p(t)
86 inverse_output = sim(inverse_GSFR_model);
87 SigIn = inverse_output.simout; %store the results for Monte Carlo
88
89
90 %Manipulate SigIn to use a "Simplified" Triangular Injection
91 injection_Max = max(SigIn);
92 injection_Index = find(SigIn.Data >= injection_Max,1,'first');
93 injection_Final = SigIn.Data(find(SigIn.Data >= 0,1,'last'));
94 final_Index = length(SigIn.Data);
95 %find the "Half" point, to not include second triangle
96 for i = injection_Index : length(SigIn.Data)-1
97     if (SigIn.Data(i)-SigIn.Data(i+1)<0)
98         final_Index = i;
99     if (SigIn.Data(i)>0)
100         injection_Final = SigIn.Data(i);
101     end
102     break
103 end
104 end
105 injection_Steps = (injection_Max - injection_Final) /(final_Index -
106 injection_Index);
107 for i = injection_Index : length(SigIn.Data)
108     value = injection_Max - ((i-injection_Index)*injection_Steps);
109     if(value>=0)
110         SigIn.Data(i) = value;
111     else
112         SigIn.Data(i) = 0;
113     end
114 end
115 %-----Part 2: Finding Injection for Estimated Values-----
116
117
118
119
120 %-----Part 3: the Experiment-----
121 %Using Injection, check the lower bounds for H's innaccuracy
122 %store the last value that does not violate Statutory Limits
123 for i = 0 : 99
124     H = j-((i/100)*j);
125     wn = sqrt((D*R+K)/(2*H*R*Tr));
126     c = (((2*H*R)+((D*R)+(K*Fh))*Tr))/(2*(D*R+K)))*wn;

```

```

127
128      %Run the GSFR model with new Variable values
129      output = sim(GSFR_model);
130      GSFR_output = output.simout;
131
132      if min(GSFR_output) > 0.99
133          results{2,j} = i;
134      else
135          results{4,j} = GSFR_output;
136          break;
137      end
138  end
139
140
141 %Using Injection, check the upper bounds for H's innaccuracy
142 %store the last value that does not violate Statutory Limits
143 for i = 1 : 99
144     H = j+((i/100)*j);
145     wn = sqrt((D*R+K)/(2*H*R*Tr));
146     c = (((2*H*R)+((D*R)+(K*Fh))*Tr))/(2*(D*R+K)))*wn;
147
148     %Run the GSFR model with new Variable values
149     output = sim(GSFR_model);
150     GSFR_output = output.simout;
151
152     if max(GSFR_output) < 1.01
153         results{3,j} = i;
154     else
155         results{5,j} = GSFR_output;
156         break;
157     end
158 end
159 %-----Part 3: the Experiment-----
160 %display progress
161 waitbar((1/12)*(j),progress);
162 end
163
164
165
166 %-----Part #: Plotting the results-----
167 hold on
168 for i = 1 : length(results) %Plot all the results in one Figure
169     plot(results{4,i});
170     plot(results{5,i});
171 end
172 yline(0.99, 'k', 'DisplayName', 'lower Statutory Limit');
173 yline(1.01, 'k', 'DisplayName', 'upper Statutory Limit');
174 plot_title = sprintf('Limit Violating Response Curves');
175 title(plot_title);
176 xlabel('Time (s) since LoG Event')
177 ylabel('Frequency (pu)')
178 legend('show')
179 hold off
180 %-----Part #: Plotting the results-----
181
182
183
184
185
186 %-----Saving Results-----
187 folder_title = sprintf('results_half_R_%f_K_%f_Fh_%f',R,K,Fh');
188 mkdir (folder_title);

```

```
189
190 file_title2 = sprintf('results_half.mat');
191 save(file_title2, "results");
192 movefile(file_title2, folder_title);
193 %-----Saving Results-----
194
195 close(progress);
```

A.6 Range of H Halved Injection Code

The entirety of the code base, models, literature, and figures can be found at the GitHub Repository:

<https://github.com/TheOneRui/IndividualProject>