Automatic generation control of a two area power system using deep reinforcement learning

S. Reynolds (BMa., BFin.)

This interim report submitted as part of the assessment schedule for:

BACHELOR OF ENGINEERING HONOURS (ELECTRICAL)

Supervisors:
Dr. C. Yeo & Dr. S. Klaric

Charles Darwin University
College of Engineering

May 3, 2020

Declaration

I, *S. Reynolds (BMa., BFin.)*, declare that this interim report is submitted in partial fulfilment of the requirements for the conferral of the degree *BACHELOR OF ENGINEERING HONOURS (ELECTRICAL)*, from Charles Darwin University, is wholly my own work unless otherwise referenced or acknowledged. This document has not been submitted for qualifications at any other academic institution.

S. Reynolds (BMa., BFin.)

May 3, 2020

Abstract

Acknowledgments

Contents

List of Figures

List of Tables

List of Abbreviations

List of Symbols

Chapter 1

Introduction

In 2018, approximately 261TW h of power was generated in the Australian electricity sector. Renewables contributed 19% of the total generation, an increase from 15% in 2017. The Department of Industry, Science, Energy and Resources have observed an increase in renewable energy generation year-on-year in the electricity generation market since 2008, as shown in Figure 1.1 [1].

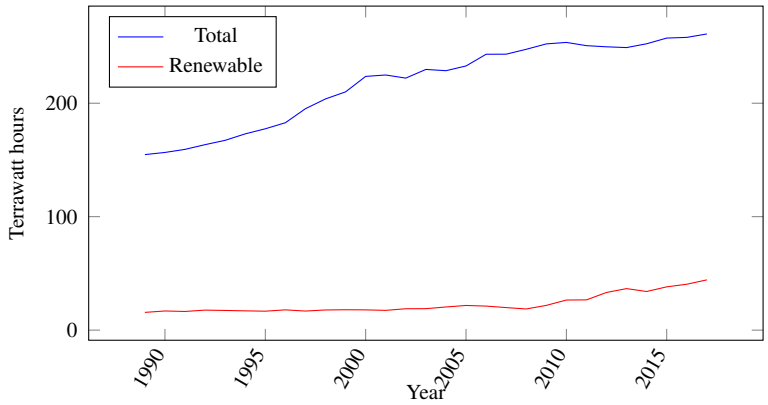

Figure 1.1: Power generation from renewable sources (light blue line), and total power generation (dark blue line) in Australia from 1977 to 2018.

One of the benefits of transitioning from thermal sources of power generation to renewable sources is reduced greenhouse gas emissions [2]; however, this transition is not without its drawbacks. With an increased reliance on renewable power generation sources posing challenges for power system stability owing to load management. A recent example is the system failure in Alice Springs, caused by an event cascade that was triggered by cloud cover shadowing a solar array. The system failure left residents in Alice Springs without power for approximately eight hours [3]. An independent investigation highlighted that poor control policies were one of the factors that contributed to the blackout. In this instance, the generator provisioned to ramp up in the event of cloud cover was unable to be controlled. Moreover, generators that were still under the control regime were

issued operating set points above their rated capacity, that resulted in thermal overload and subsequent tripping from the protection system [4].

Current control methods use classical feedback loop techniques. These methods can be brittle when faced with system changes, or scenarios which they were not designed for. An improved controller would be one that can learn and adapt its controller to an unseen system or event, given some broad control objective. This research proposes a deep reinforcement learning (DRL) agent for controlling the frequency of a power system consisting of multiple generators, and multiple load demands with individual stochastic profiles.

1.1 Research Aim

The principle aim of this research is to compare the performance of known, optimised feedback loop controller architectures against a DRL based control system when tasked with performing load following ancillary services with regulating generators under AGC for a two area power system. This research will be undertaken in order to understand the feasibility of using DRL agents for two area power system management.

1.2 Structure of Thesis

Chapter 2

Background

2.1 Power Systems and Frequency

Interconnected power systems are comprised of power generating units and energy storage systems connected to transmission and distribution networks. Generated power is used to service load demand. A single line diagram of a power network can be seen in Figure 2.1. The diagram shows how thermal generation units (left-hand side), such as coal and nuclear, in addition to renewable sources of generation, like wind and solar provide a power generation mix that is transmitted by a network for the consumers of generated energy: industry and households (right-hand side).

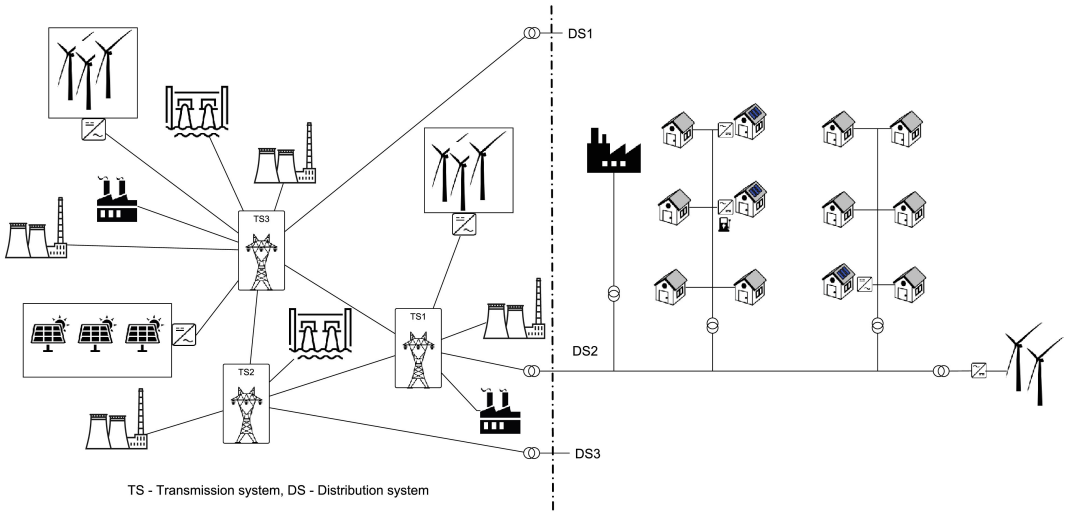

Figure 2.1: A single line diagram of a typical power system. The image shows points of generation from thermal and renewable sources, and the subsequent supply of generated energy to meet load demand through the transmission and distribution network [5].

One of the key elements to successful operation of interconnected power systems is ensuring total load demand is matched with total generation while taking into account power losses involved with generation, transmission, and distribution [6]. To understand

why it is important to match generation with load demand consider the basic operation of a single thermal generator.

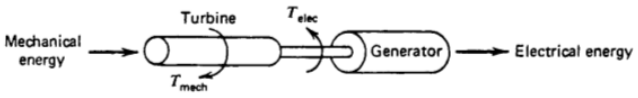

Figure 2.2: A thermal generation unit consisting of a prime mover (turbine), and a synchronous machine [6].

The essential elements of a thermal generator are a prime mover (such as a gas turbine) and a synchronous machine, as depicted in Figure 2.2. The prime mover provides mechanical torque, T_{mech}, which drives the synchronous machine producing electrical energy. In response, the synchronous machine creates an opposing torque that depends on the size of the load demand. This opposing torque is referred to as electrical torque and is denoted as T_{elec}. If α represents angular acceleration of the generator rotating mass, and I is its moment of inertia, then by Newton's second law:

$$T_{mech} - T_{elec} = I\alpha \tag{2.1}$$

When T_{mech} equals T_{elec} the system will be in a steady state of zero angular acceleration with a constant angular velocity, ω. Now, if $T_{mech} > T_{elec}$, then the system has an angular acceleration causing the angular velocity, ω, to increase. This results in a frequency increase in the system. Conversely, if $T_{mech} < T_{elec}$ then the angular velocity ω will decrease, resulting in a frequency decrease. It is important to note that, at any point in time, the total electrical load demand will fluctuate as businesses and households switch grid connected appliances or motors on and off. The result is that an uncontrolled system will have a continually changing frequency. Australia's electricity network is designed to operate at a frequency of 50Hz. In the majority of network scenarios, the Australian Energy Market Operator (AEMO) has a desired operating range for frequency which lies between 49.85 and 50.15Hz [7]. Similarly, the Power and Water Corporation (PWC) network technical code for the Northern Territory states that under normal operating conditions frequency should be maintained in the range of 49.80 to 50.20Hz [8]. Network operation outside of the specified range can cause damage to electrical equipment such as transformers or motors, which are designed to operate at specific frequencies [9]. Network designers engineer protection schemes so that sustained frequency excursions outside of the allowed range will cause equipment to trip from the network [10].

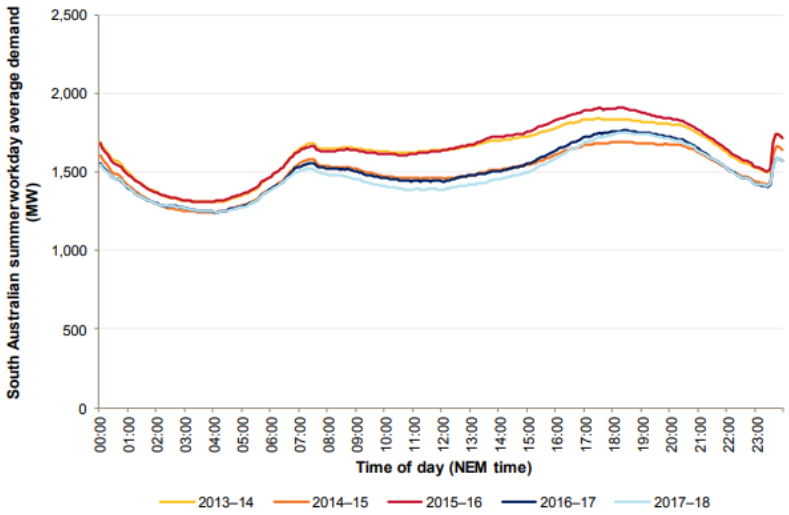

Figure 2.3: Weekday energy demand profile in South Australia during summer [11].

Protection schemes tripping equipment from the network is undesirable as this can leave households and industry without power, resulting in economic loss. Further, if disconnections are uncontrolled the system stability is reduced [10]. System controllers, such as the AEMO and PWC, are interested in being able to control the system to follow changes in load demand so that system frequency is maintained in the allowable range. Additionally, they are interested in control mechanisms to restore frequency excursions as a result of unexpected disturbances. System controllers can use historical data, like that shown in Figure 2.3, to forecast daily demand profiles with some reliability. This type of forecasting does not help when trying to predict the occurrence of random disturbances; however, it does provide a starting point for estimating required generation needed to meet demand. It is important to note that forecasting is not perfect. Inevitably mismatches in supply and demand will occur causing small imbalances between T_{mech} and T_{elec}, resulting in a change to angular velocity ω and the network frequency [12]. To perfectly match supply and demand, system controllers use generators referred to as regulating units, placed under Automatic Generation Control (AGC) [13]. A regulating unit is a generator that has the capacity to increase or decrease mechanical torque T_{mech}, and AGC is the name used for a system providing control over the mechanical torque output of regulating generators. If the system controller has a sufficient number of regulating units under AGC it can perform two functions: load following, and restoring the system to stable operating conditions in the event of a disturbance [14]. Using a regulating unit under AGC control to load follow is referred to, by AEMO, as load following ancillary services [15].

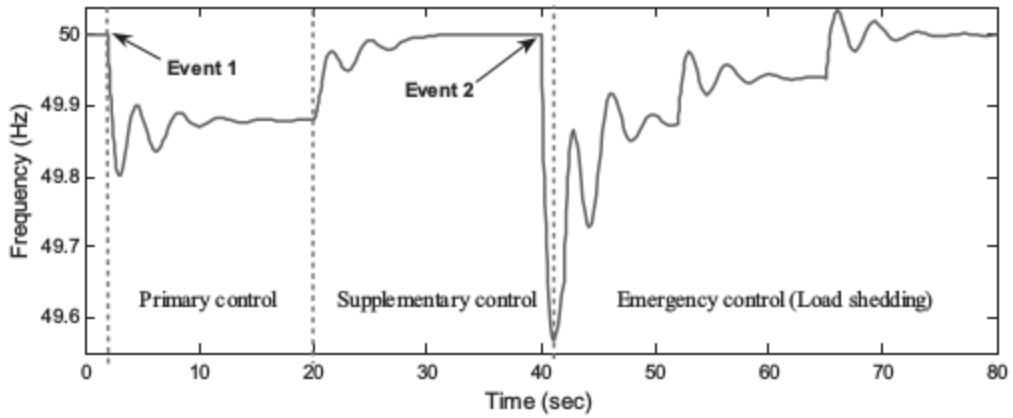

Figure 2.4: A minor frequency disturbance occurs at the 2 sec mark and primary control systems (governors) arrest the frequency drop. System frequency is adjusted to desired 50Hz operating level using AGC control of regulating units. This referred to as supplementary (or secondary) control in the literature. AEMO refers to this as load following ancilliary services. At the 40 sec mark the network experiences a major frequency disturbance which is arrested by emergency control measures such as under-frequency load shedding (UFLS). System restoration is aided using AGC control of regulating units, which AEMO refers to as spinning reserve ancillary services [16].

Load following control adjusts regulating units in order to match supply with a demand load profile, and maintain frequency in a normal operating range a shown in the first 40 seconds of Figure 2.4. Using a regulator under AGC control to restore the system after a major disturbance is referred to, by AEMO, as providing spinning reserve ancillary services. [15]. When used in either fashion it is important to note that the regulating unit is not responsible for arresting frequency excursions, rather, it is used to restore the system back to the allowable frequency operating range after the frequency excursion has been arrested. An example of a frequency excursion, arrest, and subsequent restoration for minor and major disturbances can be seen in Figure 2.4. AEMO and PWC do not require all generators on the network to act as regulating units since adequate frequency control can be achieved using a subset of the total available generators.

2.1.1 Modelling a Single Area System

2.1.2 Frequency Control for a Single Area System

The power system model shown in Figure 2.1 depicts total generation coming from many generation assets — this is complex to model. Researchers often find it useful to divide generation assets into sub-groups referred to as control areas [13]. A control area is defined as a subset of generators that are in close proximity to each other and constitute a coherent group that speed up and slow down together, maintaining their relative power angles [13]. Therefore, the total network is comprised of many interconnected control

areas. An example of this can be seen in Figure 2.5. Notice that for each area there is only one load and one generator.

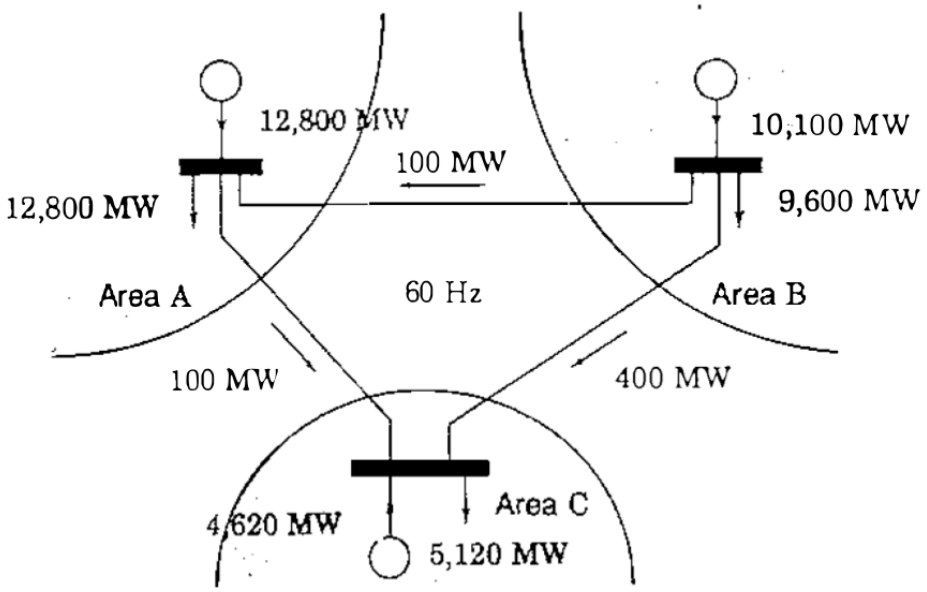

Figure 2.5: An example of three interconnected control areas in a 60Hz power system. The interconnections allow power to flow from one area to another, allowing generators to service loads from different areas. Each control area consists of several generators and loads, but are modelled with a single generator and single load for simplicity [14].

Typically, for each control area, researchers will aggregate many loads into a single load, and many generators into a single generator. This simplifies the model further [14]. The simplest power system to control is one that consists of a single control area. A single control area power system has no interconnections to any other control area. It is comprised of a consumer load demand, and a set of generators, some of which are acting as regulating units. As previously mentioned, for modelling simplicity, loads are aggregated to a single load, and generators can be aggregated to a single generator. This simple system is well understood. It is generally acknowledged that a speed droop governor feedback control regime will perform primary frequency control, and an AGC feedback loop is used to perform secondary frequency control when restoring a minor frequency excursion [6], [13], [14], [17]. A particularly well laid out approach to developing linear models for the turbine, generator, load, and governor was presented by Kundur [17]. The full model is shown in Figure 2.6. This particular model provides a model for a single regulating generator supplying a load. The governor block is a first-order linear model of the governor. The turbine block is a first-order linear model of the turbine. The final block is the generator-load, which is also a first-order linear model. The AGC feedback loop uses a proportional integral controller.

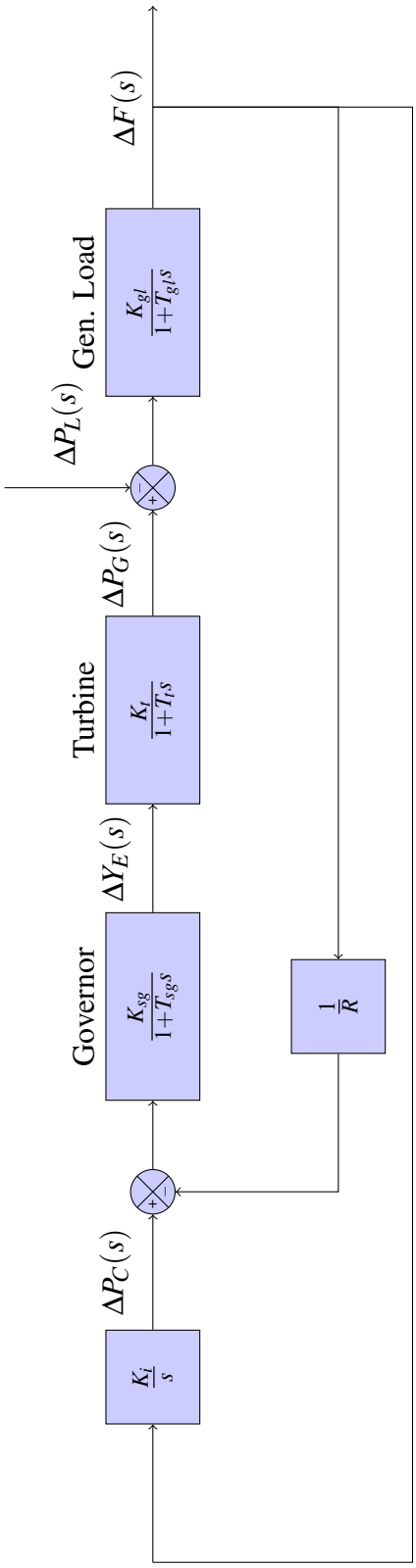

Figure 2.6: A classical feedback control approach for a single control area power system. The system is comprised of a first order models for both turbines, and generators. The governor controllers are also first order models. AGC is implemented using an integral control block in a feedback loop [17].

2.1.3 Modelling a Two Area System

2.1.4 Frequency Control for Two Area System

The single area system presented in Section 2.1.2 is useful to help understand the role of governors and AGC in controlling power system frequency. In reality, power systems are comprised of many control areas connected by transmission lines (referred to in the literature as tie lines). Often it is the case that there is some net power transfer over the tie lines, enforceable by economic contract. Single area models do not provide for this additional complexity.

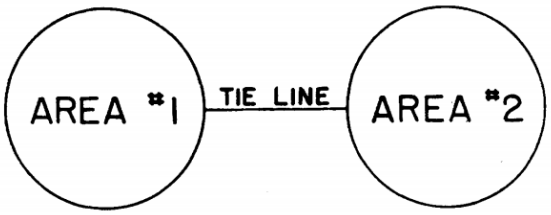

Figure 2.7: A two area power system comprised of generators and load connected via a tie line. Power flows from one area to the other depending on the power demands.

Distinct control areas are typically thought of as different participants in the generation market, or simply as different regions in which generation assets are based [13].

The simplest model that includes tie lines is the two area power system, shown in Figure 2.7. The control objective with this system is to maintain the inter-area power transfer, whilst regulating the frequency of each area. An AGC integral feedback loop on regulating units, like that shown in Figure 2.6, would ensure that power system frequency is maintained, however, would not guarantee inter-area power transfer agreements are observed. Violation of power transfer contracts due to control issues does not allow for a stable market in which energy can be reliably traded. Fortunately, multi control area power systems are well understood. Linear models have been developed to simulate these systems, and classical control approaches have been successfully implemented to meet the new control objectives. In order to achieve this, a metric called Area Control Error (ACE) is used in the AGC feedback loop for each control area. ACE is a linear combination of the frequency deviations and the . The implementation of this control system is shown in Figure 2.8.

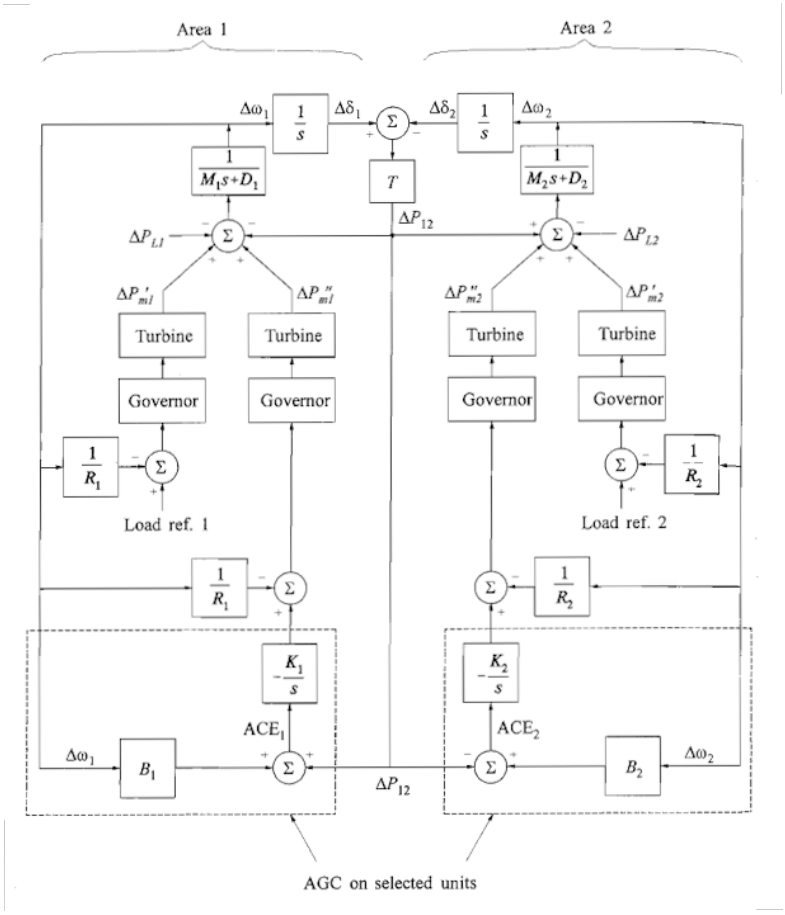

Figure 2.8: A classical feedback control approach to a two area power system [17].

2.2 Reinforcement Learning

According to Sutton and Barto's seminal text, reinforcement learning (RL) is a branch of machine learning, based on trial-and-error, that is concerned with sequential decision making [18]. An RL agent exists in an environment. Within the environment it can act, and it can make observations of it's state and receive rewards. These two discrete steps, action and observation, are repeated indefinitely with the agent's goal being to make decisions so as to maximise its long term reward — this scenario is represented diagramatically in Figure 2.9.

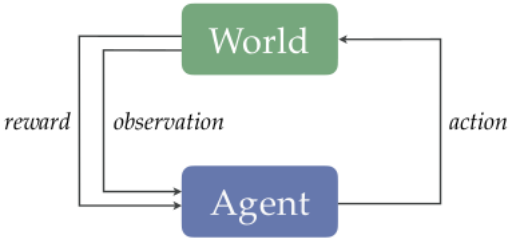

Figure 2.9: text

One of the most common ways to represent the RL problem is to model the environment as a set of discrete probabilistic transitions between states, for a set of possible actions that can be selected by the agent. A state transition presents the agent with a reward signal that informs the agent whether an action taken was good or bad. This environmental architecture is referred to as a Markov Decision Process (MDP). It is the agent's objective to maximise the reward it will receive in the future. An agent can achieve this by learning an optimal policy which maps environment states to actions. Learning such a policy is key idea in RL, and the agent achieves this by experimentation.

2.2.1 Markov Decision Process

Bellman's pioneering work on the Markov Decision Process (MPD) provided the necessary architecture to develop RL algorithms [19]. His work considered an agent that exists in some environment described by a set of discrete states S. At any discrete point in time the agent can take an action from the set of possible actions A. When the agent takes an action in a given state, the agent receives some reward that is assigned according to a reward function $R : S \times A \times S \to [R_{min}, R_{max}]$. Fundamental to Bellman's MDPs were the state transition dynamics which were defined by probabilities: if an agent is in a given state, $s \in S$, and takes action, $a \in A$, this will transition the agent to a new state, $s' \in S$, and yield reward, $r \in R$, with some given probability. This set of probabilities are assigned by a state transition function $P : S \times A \to S$. Generally, the a single reward is bound to a state transition so function P can be thought to assign a state and reward. The function P, and

it's simpler notation p, is typically written as

$$P(S_{t+1} = s',\ R_{t+1} = r \mid S_t = s,\ A_t = a) = p(s', r \mid s, a). \tag{2.2}$$

The set of parameters outlined above, and expression 2.2, make up a framework referred to as an MDP.

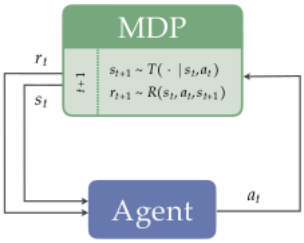

Figure 2.10: text

2.2.2 Returns, Episodes, and Policy

In addition to developing the MDP framework, Bellman was also responsible for key developments in a field of research called dynamic programming (DP) [20]. Assuming that the agent has complete knowledge of the state transition probabilities of an environment, DP algorithms can be used to determine analytical solutions for the problem of how an agent should behave to maximise it's cumulative reward [20], [21]. This idea was originally thought to be distinct from RL. The main difference is that DP provides the agent with complete knowledge of it's environment, whereas RL agents have no knowledge of environment dynamics and must learn them as well as how to maximise their cumulative reward [18]. Many researchers made links between DP and RL [22]–[24], but it wasn't until 1989 that Watkins presented the first formal treatment of RL in an MDP framework. Watkin's work showed that DP algorithms could be modified for use with RL problems [25]. Central ideas used in DP algorithms include episodes, returns, and policies [18].

The duration of time that an agent will spend taking actions and transitioning states before encountering a terminal state is defined as an episode. It is the agent's goal to take actions such that it maximises the sum of all the rewards as it concludes an episode. The cumulative sum of rewards is called the return. Consider an agent taking an action at each discrete time step, t, and receiving reward, r_t, after each action. If there are N discrete time steps before the agent reaches a terminal state, Bellman defines the return as

$$G_t = \sum_{k=0}^{N-1} r_{t+k}. \tag{2.3}$$

Rewards received in the future are often perceived as less valuable than rewards received in the present. To account for this Bellman used a discount factor applied to each

reward in the sequence. Letting $\gamma \in [0,1]$ then 2.3 becomes

$$G_t = \sum_{k=0}^{N-1} \gamma^k r_{t+k}. \tag{2.4}$$

Finally, in order for the agent to take actions it must have a belief of what action it should take, given it's current state. This belief is called a policy and denoted as π [18]. Sutton and Barto define a policy as the mapping of states to actions i.e. a rule that determines what actions the agent should take for a given state. A policy can be deterministic, and depend only on the state, $\pi(s)$, or stochastic, $\pi(a|s)$, such that it defines a probability distribution over the actions, for a given state. An optimal policy, denoted $\pi*$, is a policy which will maximise the return an agent receives over an episode.

2.2.3 Value Function and the Bellman Equations

The basic principal of dynamic programming is to assign a value to each state that informs an agent how useful a state is to achieving a high cumulative reward. Watkins refers to the creation of systems to assign values to states as the credit assignment problem [25]. Bellman's approach to solving credit assignment was to develop mathematical functions to assign values to states [20]. Bellman's *value function*, $V_\pi(s)$, is defined as the expected sum of the discounted return, G_t, that the agent will receive while following policy π from a particular state s. Mathematically, this is expressed as

$$V_\pi(s) = \mathbb{E}_\pi\big(G_t \mid s_t = s\big) = \mathbb{E}_\pi\bigg(\sum_{k=0}^{\infty} \gamma^k r_{t+k} \mid s_t = s\bigg). \tag{2.5}$$

The state value function was useful to Bellman because it provided a way to check if one policy was better than another policy. It is clear an agent would prefer policy π over some other policy π' if the expected return from using policy π is greater than the expected return from using policy π' for all $s \in S$. Since the state value function is defined by the expected return, Bellman expressed this idea in state value function terms i.e. if policy π is preferred to π' then $V_\pi(s) \geq V_{\pi'}(s)$ for all $s \in S$. The optimal policy, π^*, yields the best state value function, $V^*(s)$, referred to as the optimal state value function and defined as:

$$V^*(s) = \max_\pi V_\pi(s),\ \forall\, s \in S. \tag{2.6}$$

Although the state value function provides a way to compare one policy against another policy, it can not be used to specify the policy it evaluates. Bellman developed a variation of equation 2.5 called the *state-action value function* that can evaluate and specify a policy. The state-action value function, $Q_\pi(s,a)$, is defined as the expected sum of the discounted return, G_t, that the agent will receive if it takes action a in state s, and then

follows policy π thereafter. Mathematically, this is expressed as:

$$Q_\pi(s,a) = \mathbb{E}_\pi\big(G_t \mid s_t = s,\ a_t = a\big) = \mathbb{E}_\pi\bigg(\sum_{k=0}^{\infty} \gamma^k r_{t+k} \mid s_t = s,\ a_t = a\bigg). \tag{2.7}$$

Similarly to the state value function, the state-action value function's optimal form, $Q^*(s,a)$, can be defined as:

$$Q^*(s,a) = \max_\pi Q_\pi(s,a),\ \forall\ s \in S,\ a \in A. \tag{2.8}$$

To extract the policy π from a state-action value function Q_π the action corresponding to the largest state-action value is chosen for each given state. This is called the greedy policy and is defined, for each $s \in S$, as:

$$\pi(a|s) = \begin{cases} 1 \ \text{ if } \ a = \arg\max_{a'} Q(s,a') \\ 0 \ \text{ if } \ a \neq \arg\max_{a'} Q(s,a') \end{cases} \tag{2.9}$$

Bellman used the value functions presented in 2.5 and 2.7 to formulate recursive expressions which could then be used to solve the DP problem [19]. These are known as the *Bellman equations*. Letting $A(s)$ be the set of actions available in state s, if the agent is operating under the optimal policy π^* then it is true that

$$V^*(s) = \max_{a \in A(s)} Q_{\pi*}(s,a). \tag{2.10}$$

Using the notation shown in equation 2.2, equation 2.10 can be rewritten using equation 2.7

$$V^*(s) = \max_a \sum_{s',r} p(s',r \mid s,a)[r + \gamma V^*(s')]. \tag{2.11}$$

Equation 2.11 is referred to as the Bellman optimality equation for $V^*(s)$. The Bellman optimality equation for $Q^*(s,a)$ is

$$Q^*(s,a) = \sum_{s'} p(s',r \mid s,a)[r + \gamma \max_{a'} Q^*(s',a')]. \tag{2.12}$$

If the transition probabilities and rewards are known to the agent then the Bellman opitmality equations can be solved iteratively, which is known as dynamic programming [19]. Algorithms which assume known transition probabilities and rewards are collectively referred to as *model-based* algorithms. Most RL problems assume state transition probabilities are unknown. The collection of algorithms that provide solutions to these problems are called *model-free* algorithms.

2.2.4 Value Function Based Methods

Model free methods can be applied to any RL problem since they do not require a model of the environment. Figure 2.11 provides an overview of the different families of algorithms used for solving RL problems. Algorithms that try to solve the RL problem by estimating the optimal state-action value function, Q^*, and inferring the optimal policy from it are referred to as value function based methods. The most common value function based methods are Monte Carlo and temporal difference approaches.

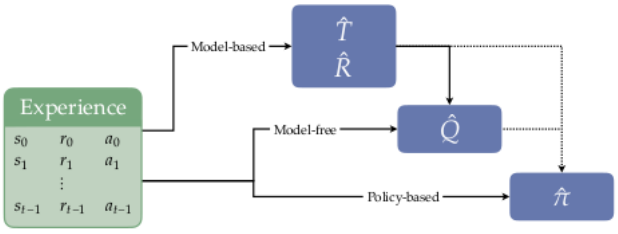

Figure 2.11: text

Monte Carlo Methods

Sutton and Barto were the first to introduce a version of the Monte Carlo (MC) control algorithm for estimating state-action value functions [18]. The MC control algorithm is based on an iterative approach that takes a sample of episodic sequences consisting of states, actions, and rewards. Sequences are obtained from the agent interacting with the environment, using some policy π. This is called the *policy evaluation* step. Once an episode is completed, the state-action value function, $Q_\pi(s,a)$, is updated for each discrete state-action pair visited. This second step is called the *policy improvement* step.

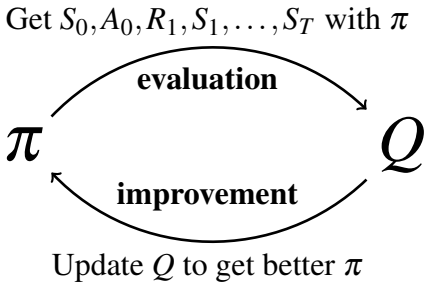

Figure 2.12: text

Sutton and Barto found that during the policy evaluation step, if the agent was allowed to select the action with a greedy policy (2.9) from the current iteration of the state-action value function, then MC control would not always converge to the optimal state-action value function. The problem with using the greedy policy is that it can cause the agent to over commit to locally promising but globally poor solutions. This is due to the agent not

sufficiently exploring the state-action value space, and is especially problematic during the early stages of the algorithm where the agent has little knowledge about the environment.

The most common method of overcoming this problem is to use an ε-greedy policy instead of the greedy policy. The basic idea behind the ε-greedy policy is to select the greedy action most of the time, and select non-greedy actions the other times. This is achieved by setting a parameter, $\varepsilon \in [0,1]$, which allows the agent to select non-greedy actions with a non-zero probability. The ε-greedy policy is defined as:

$$\pi(a|s) = \begin{cases} 1-\varepsilon & \text{if } a = \arg\max_{a'} Q(s,a') \\ \frac{\varepsilon}{|A|-1} & \text{if } a \neq \arg\max_{a'} Q(s,a') \end{cases} \tag{2.13}$$

Algorithm 1 Constant α Monte Carlo Control

Input: `num_episodes`, α, ε_i
Output: Q ($\approx Q^*$ if `num_episodes` is large enough)
Initialise Q such that $Q(s,a) = 0$ for all $s \in A$ and $a \in A$
for $i \leftarrow 1$: `num_episodes` **do**
    $\varepsilon \leftarrow \varepsilon_i$
    $\pi \leftarrow \varepsilon$ − greedy
    Generate and episode $S_0, A_0, R_1, \ldots, S_T$ using π
    for $t \leftarrow 0 : (T-1)$ **do**
        if (S_t, A_t) is a first visit **then**
            $Q(S_t,A_t) \leftarrow Q(S_t,A_t) + \alpha(G_t - Q(S_t,A_t))$
        end if
    end for
end for
return Q

Write a note about how the optimal policy can be selected from the optimal Q function.

Temporal Difference Methods

Talk about the fact that Monte Carlo methods collect an entire episode prior to making an update - temporal difference methods use an idea of bootstrapping to update the Q after each time step. This has low bias but high variance (check this idea)?

Watkins is credited with the most influential integration of RL with MDPs, and DP. His work on an RL algorithm called Q-learning highlighted the importance of another type of value function called the action-value function. The action-value function is defined as the expected sum of rewards that the agent will receive while taking action a in state s and, thereafter, following policy π.

$$Q(s,a) := Q(s,a) + \alpha[r + \max_{a'} Q(s',a') - Q(s,a)] \tag{2.14}$$

Algorithm 2 Q-learning

Input: `num_episodes`, α, ε_i
Output: value function Q ($\approx Q^*$ if `num_episodes` is large enough)
Initialise Q such that $Q(s,a) = 0$ for all $s \in S$ and $a \in A$
for $i \leftarrow 1$: `num_episodes` **do**
    $\varepsilon \leftarrow \varepsilon_i$
    Observe initial state S_0
    $t \leftarrow 0$
    repeat
        Select an action A_t using policy derived from Q (e.g. ε-greedy)
        Carry out action A_t
        Observe reward R_{t+1} and new state S_{t+1}
        $Q(S_t,A_t) \leftarrow Q(S_t,A_t) + \alpha[R_{t+1} + \gamma \max_a Q(S_{t+1},a) - Q(S_t,A_t)]$
        $t \leftarrow t+1$
    until S_t is terminal
end for
return Q

2.2.5 Policy Search Methods

2.2.6 Actor Critic Methods

2.3 Deep Neural Networks

Deep neural networks are the technology responsible for some of the most recent state-of-the-art technological breakthroughs in fields such as audio to text speech recognition systems [26], image classification systems [27]–[30], text to text machine translation (REFERENCE), and robotics [31]–[34].

Deep neural networks are able to adapt to the different needs of diverse research fields due to their unique computational architecture. A deep neural network (DNN) is comprised of many nodes, called perceptrons, each of which are equipped with an activation function. Using the activation function each node can signal when it recognises an input or not. The DNN architecture connects many nodes together, using weighted edges, to form a network. Modifying the edge weights is referred to as *training the network*, and allows the DNN to change its behaviour. Given this flexibility, a DNN is often thought of as a tool for universal function approximation.

2.3.1 Perceptron Model

Rosenblatt is credited with developing the perceptron model ubiquitous to neural network architectures [35]. (WHAT WAS THE MOTIVATION OF HIS WORK). Rosenblatt's perceptron consisted of a single node, or neuron, used for the classification of patterns that are linearly separable. The neuron takes a vector of inputs and applies a weight,

multiplicatively, to each vector element. The multiplied elements are then summed along with a bias term. Letting input vector elements be x_i, weight terms be w_i, and the bias term be b, the summation operation can be expressed as:

$$\sum_i x_i w_i + b \tag{2.15}$$

The summation is then passed through an activation function, f, to produce the neuron output. Using equation 2.15 and letting the neuron output be y, the neuron model can be expressed as:

$$y = f\left(\sum_i x_i w_i + b\right) \tag{2.16}$$

Figure 2.13 provides an shows the computational model of a neuron, expressing 2.16.

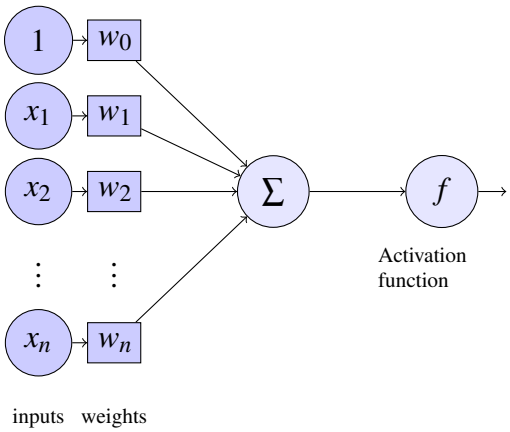

Figure 2.13: text

2.3.2 Activation Functions

sigmoid - who developed these relu - who developed these tanh - who developed these

2.3.3 Feedforward Networks

A typical fully connected feed-forward ANN consists of an input layer, one or more hidden layers, and an output layer, as shown in Figure 4. Hidden layers are made up of multiple nodes. The nodes themselves contain a non-linear activation function, such as a sigmoid or ReLU, and receive weighted input from the previous layers in the model.

2.3.4 Training the Network

Changing the weights in a neuron changes the neurons's contribution to the model, which in turn affects the overall model output. Weight changes occur during model training,

which uses large volumes of labelled data to adjust the weights. Hidden layers are important because they allow highly non-linear models to be constructed, providing an approach for estimating complex phenomena which may be difficult to model with classical approaches, or computationally intractable. Generally, the more hidden layers, the more non-linear the model. Network architectures with multiple hidden layers have become so wide spread that the term Deep Neural Network (DNN) was coined to describe feed-forward ANNs which use two or more hidden layers. It must be noted that whilst increased non-linearity may allow us to model more complex phenomenon, making the ANN deeper does not guarantee increased model performance. This is mainly due to the fact that deeper models may over-fit the data during training, resulting in a failure to generalise on test and validation data sets.

werbos backpropagation

2.3.5 Regularisation

dropout

2.4 Deep Reinforcement Learning

2.4.1 Deep Q-Learning

2.4.2 Deep Deterministic Policy Gradient

Chapter 3

Literature Review

Power systems are non-linear; however, traditional control structures for maintaining power system frequency such as load frequency control (LFC) or automatic generation control (AGC), are designed under the assumption that plant modelled using linear ordinary differential equations can capture the dynamic behaviour of existing power systems. If frequency deviations are small, then this assumption is reasonable for the amount of non-linearity present in existing power systems. If frequency deviations are large, or additional non-linearity were introduced to the system, then it is no longer reasonable to assume linear ODE system models.

Owing to an increase in the proportion of photovoltaic power generation, along with an increase in the use of high voltage direct current (HVDC) transmission lines in Australia's power network, power system dynamics are becoming more non-linear (REFERENCE). This is creating a need to explore novel control architectures for AGC in order to improve control performance for the non-linear system. One such control architecture being investigated is deep reinforcement Learning (DRL). To fully grasp this problem two key areas of understanding are necessary. Firstly, it is important to know what LFC is and the control architectures that have previously been explored to address this problem. Secondly, knowledge of DRL and its historical applications to control problems is required to understand strengths, limitations, and underlying assumptions of the architecture.

3.1 Automatic Generation Control

Since Thomas Edison's first commercial power station, commissioned in 1882 at 255-257 Pearl Street New York, controlling power frequency has been a key factor in power generation [36]. One of the first attempts to control the frequency for a single generator and load system was with a device called a turbine governor (REFERENCE). The turbine governor was designed with an in-built proportional feedback control loop (REFERENCE). DESCRIBE HOW THIS ACTUALLY WORKS. The literature often refers

to this as the primary control loop [16]. Operators often found that primary loop governor control would require constant fine tuning to ensure that the power system was operating at the scheduled frequency (REFERENCE). XXXX undertook a mathematical analysis using first order linear models for the governor, turbine, and generation load control (REFERENCE). The analysis showed that primary control with a governor was successful in arresting frequency deviations from the desired set point, but persistent offset errors from the set point prevented the uptake of the technology [37]. Later research concluded that a secondary control loop to the governor was required to provide sufficient frequency control [38]. The secondary control loop provided integral control (REFERENCE). Mathematical analysis, using first order linear models for plant showed that atuned proportional-integral (PI) controller was able to arrest frequency deviations and subsequently restore the system frequency to it's scheduled value. The PI control scheme constitutes the classical approach to the solving the LFC problem (REFERENCE).

Cohn [39] and Aggarwal *et al.* [40], [41] undertook pioneering work to develop classical control approaches to work with power systems comprised of two or more control areas. A system made up of more than one power system area required frequency control, but also the control of the power flow over the transmission infrastructure (tie lines) which connected the areas (REFERENCE). Cohn's paper used a first order linear system to model the tie line dynamics. The development of a tie line model allowed Cohn to use PI control architectures for each power system area, albeit with modified input signals (REFERENCE). One of the most important things to come out of his work was development of the feedback signal called area control error (ACE). Cohn used ACE to ensure power systems were restored to the scheduled frequency, given a frequency deviations, and that unscheduled tie line power flows were minimised between neighbouring control areas [42]. Classical power system frequency controllers can be designed using Bode and Nyquist diagrams to obtain desired gain and phase margins. Root locus plots can also be used [43]. While these approaches are simple, well known, and easy for practical implementation, investigations using these approaches have resulted in control schemes that exhibit poor dynamic performance. This is especially true in the presence of parameter variations and nonlinearities [17], [38], [44].

Frequency controller analysis and design assumes plant models have linear dynamics; however, studies have shown that modern power systems display complex non-linear dynamics [45]–[48]. Modern power systems are large-scale and comprised of multiple power generation sources such as thermal, hydro, and photovoltaic power — some of the more commonly researched generator non-linearity include governor dead band (GDB) [45] and generator ramp constraint (GRC) [46], [47]. Moreover, modern power systems use high voltage direct current (HVDC) lines to export power over long distances, and they also feature energy storage systems such as pumped hydro or batteries [12], [13], [16], [17]. Both of these features display highly non-linear characteristics

(REFERENCE).

Linear ODE power system models capture underlying plant characteristics; however, these models are only valid within certain operating ranges. Non-linear plant characteristics mean that different linear ODE models are required as plant operating conditions change. Governor dead band is observed as a change in generator angular velocity for which there is no change in the governor valve position. GDB is generally attributed to backlash in the governor mechanism, and degrades LFC performance (REFERENCE). GRC is a physical limitation of the turbine that imposes upper and lower boundaries on the rate of change in generating power from the turbine [48]. In recent years, frequency control methods using fuzzy logic, genetic algorithms (GAs), and artificial neural networks (ANNs), have attempted way to address the problems that arise due to non-linearity.

3.1.1 Fuzzy Logic Control

Fuzzy logic control schemes are developed directly from power system domain experts or operators who control plant manually. Researchers have shown that a fuzzy gain scheduling PI controller can perform as well as a fixed gain controller, for frequency control of two and multi-area power systems. Moreover, it was found fuzzy controllers are simpler to implement [49], [50]. Yesil et al. [51] proposed a self-tuning fuzzy PID controller for a two area power system and noted improvements in controller transient performance when compared to a fuzzy gain scheduling PI controller.

3.1.2 Genetic Algorithms

Genetric algorithms are stochastic global search algorithms based on natural selection. In the context of power system control, GAs operate on a population of individuals. An individual is a set of control system parameters which are initially drawn at random and without knowledge of the task domain. Successive generations of individuals are developed using genetic operations such as recombination or mutation. An individuals chance of being selected for used in an genetic operation is based on an objective measure of fitness — strong individuals are retained and weak individuals are discarded [52].

Chang et al. [53] investigated using GA to determine fuzzy PI controller gains, which resulted in a control scheme which performed favourably when compared to a fixed-gain controller. Rekpreedapong et al. [54] took this one step further by optimally tuning PI controller gains with GA while using linear matrix inequalities (LMI) constraints from a higher order controller. This research, performed on a three area control system, was motivated by the belief higher order controllers are not practical for industry. Rekpreedapong et al. concluded that the GA tuned PI controller, under LMI constraints, performed almost as well a higher order control system. Research undertaken by Ghosal [55] concluded

that PID control with gains optimised by GA provided better transient performance than PI control with gains optimised in the same way.

3.1.3 Artificial Neural Networks

Artificial neural networks are systems that take input signals and, using many simple processing elements, produce output signals. The processing elements, or neurons, each have a number of internal parameters referred to as weights. Changing a weight will change the behaviour of a neuron. If many weights are changed, the behaviour of the ANN can be changed. The goal is to choose weights of the network in order to achieve the desired input/output relationship — this is called training the network [56].

Beaufays et al. [57] demonstrated it was possible to used a neural network for frequency control in one and two-area power systems. The ANN replaced the integral controller in the classical structure; however, employed a state variable vector input containing frequency deviation and tie-line power measurements instead of a single value ACE signal seen with classical controllers. The network was trained using a back propagation through time algorithm, and resulted in better transient performance when compared with a classical PI controller. Using these results, Demiroren et al. [58] went further by including non-linearity in the plant models. Specifically, governor deadband, reheater effects, and generating rate constraints are included and it was shown that the results obtained using the ANN controller outperformed the results of a standard PI classical control model for a two-area power system. Research undertaken a year later confirmed these results for a larger four-area power system with thermal and hydro generation sources [59].

3.2 Deep Reinforcement Learning

Chapter 4

Approach

4.1 Required Data Sources and Data Management

Training a DRL agent to change regulating generator set points in order to maintain system frequency and tie line requirements while load following will require realistic demand profiles. Similarly, performing system restoration after a disturbance will require realistic disturbance scenarios. Ideally this data would come from a major utility provider, such as PWC, in the form of a time series dataset with a large number of features, and high sample rate. To this end, data acquisition will be one of the principle objectives in the early stages of research. Should the acquisition of data from PWC or TGEN be viable, a data management plan will need to be developed which addresses concerns around the sensitivity and security of the data. The data management plan will outline where data the will be stored, and how the data will be treated (or disposed of) once the research is concluded.

In the event data cannot be acquired from a utility provider, a simulated data set may need to be used. This would be achieved by understanding key statistical parameters of a typical load demand profile, and using these to create a process which emulates the load demand signal. This could also be done for other system variables; however, care would need to be taken ensuring correlations are preserved between multiple variables in the simulated time series dataset.

4.2 Theoretical Approach

In order to establish the most effective way to approach this research problem, a clear understanding of the benefits and limitations of existing AGC approaches is needed. Determining justifications for practical AGC design choices will help to uncover important performance aspects the research should focus on. Equally important is exploring alternative approaches to AGC that researchers have investigated historically. This should have

a particular focus on the use of Neural Networks and DRL agents for AGC. A literature review will be the main avenue for achieving this.

As discussed in §4.1, securing load demand profile datasets from a major utility provider, or developing simulated load profile datasets based on local load profile characteristics is an important aspect of this research and is a priority. Similarly, investigating suitable software packages to develop the power system simulation model, and investigating suitable programming languages to implement a DRL agent are necessary. Exploring the field literature and finding published examples will be a key stepping stone. It will be important to understand how other researchers integrated the DRL agent with the simulation environment.

A simulated model of the two area power system will be developed. The decision to use a linear or non-linear model will be informed by the literature review. It may be interesting to explore DRL agent performance on both linear and non-linear models since one of their advantages is that they have a demonstrated capacity for controlling non-linear systems. Classical engineering system modelling techniques will be employed for power system model development [43]. An area of interest is how sensitive a DRL agent control regime is to changes in key plant parameters — for a given set of parameter changes both DRL agent and classical control architecture performance could be compared to see which controller is more brittle.

A feedback loop controller will be developed for the two area power system using models presented in the literature. A DRL agent will be developed using an architecture that takes continuous input signals and provides continuous output signals. There are a number of established DRL models presented in texts like Sutton and Barto that will be explored to determine the most ideal approach [18]. Time permitting, a DRL model that uses discrete input and output signals will be developed. Discrete models offer lower performance due to errors introduced in the discretisation process, but can be computationally less expensive than continuous models. DRL models will be trained using data previously acquired either from a utility provider, or from simulation. Metrics will be selected to measure the performance of both controllers. Choice of metrics will be informed by earlier research (mentioned above). Control models differences in performance will be compared — this will be one of the major focuses of the research.

The full list of tasks for the research design are as follows:

1. Enquire with power utility provider to secure data.
2. Investigate ways to simulate data.
3. Develop data management plan.
4. Literature review centred on three central themes:
    (a) AGC

(b) DRL

(c) Applications of DRL to AGC.

5. Investigate suitable software package to conduct simulation.
6. Investigate suitable programming language to implement DRL agent and integrate with simulation.
7. Develop and test simulation of two area power system.
8. Develop feedback loop controller for two area power system.
9. Test classical controller.
10. Develop DRL model.
11. Train and test DRL model.
12. Execute control trials on both models for an unseen sequences of load demand data.
13. Compare controller performance on AGC task.

It is anticipated that there may be some issues in carrying out the aforementioned research design. The biggest risk would be the inability to successfully build the control models for both the classical engineering controller, and the DRL controller. For the classical engineering controller, the issue would be the inability to find the appropriate parameter settings to deliver stable control. With the DRL controller, the problem is selection of an appropriately sized neural network, and training hyper-parameters.

Chapter 5

Experiments

Chapter 6

Analysis and Discussion of Results

Chapter 7

Conclusion

Chapter 8

Future Work

Bibliography

[1] (2020). Electricity generation, Department of industry science energy and resources, [Online]. Available: https://www.energy.gov.au/data/electricity-generation.

[2] "Special report on renewable energy sources and climate change mitigation," Intergovernmental Panel on Climate Change, Tech. Rep., 2012. [Online]. Available: https://www.ipcc.ch/site/assets/uploads/2018/03/SRREN_FD_SPM_final-1.pdf.

[3] "Independent investigation of alice springs system black incident on 13 october 2019," Utilities commission of the northen territory, Tech. Rep., 2019. [Online]. Available: https://utilicom.nt.gov.au/__data/assets/pdf_file/0011/767783/Independent-Investigation-of-Alice-Springs-System-Black-Incident-on-13-October-2019-Report.pdf.

[4] D. Wilkey, "Alice springs system black 13 october 2019," Entura, Tech. Rep., 2019. [Online]. Available: https://utilicom.nt.gov.au/__data/assets/pdf_file/0012/767784/Advice-Entura-Alice-Springs-System-Black-13-October-2019-Report.pdf.

[5] M. Glavic, "Deep reinforcement learning for electric power system control and related problems: A short review and perspectives," *Annual reviews in control*, 2019.

[6] A. J. Wood, B. F. Wollenberg, and G. B. Shelbe, *Power generation, operation, and control*, 3rd Edition. Wiley, 2013.

[7] "Power system frequency and time deviation monitoring report - reference guide," Australian Energy Market Operator, Tech. Rep., Jul. 2012. [Online]. Available: https://aemo.com.au/-/media/files/electricity/nem/security_and_reliability/ancillary_services/frequency-and-time-error-reports/frequency_report_reference_guide_v2_0.pdf.

[8] *Network technical code and network planning criteria*, Power and Water Corporation, 2013. [Online]. Available: https://www.powerwater.com.au/__data/assets/pdf_file/0022/5962/Power-and-Water-Corporation-Network-Technical-Code-and-Network-Planning-Criteria.pdf.

[9] P. C. Sen, *Principles of Electric Machines and Power Electronics*, 3rd Edition. Wiley, 2014.

[10] "Power system frequency risk review - final report," Australian Energy Market Operator, Tech. Rep., Apr. 2018. [Online]. Available: https://aemo.com.au/-/media/files/electricity/nem/planning_and_forecasting/psfrr/2018_power_system_frequency_risk_review-final_report.pdf?la=en&hash=1684259023A1FA274D7F3B8CE855D0BA.

[11] "South australian electricity report," Australian Energy Market Operator, Tech. Rep., Nov. 2019. [Online]. Available: https://www.aemo.com.au/-/media/Files/Electricity/NEM/Planning_and_Forecasting/SA_Advisory/2019/2019-South-Australian-Electricity-Report.pdf.

[12] J. D. Glover, S. S. Mulukutla, and T. J. Overbye, *Power system analysis and design*, 5th Edition. Cengage Learning, 2012.

[13] D. P. Kothari and I. J. Nagrath, *Modern Power System Analysis*, 4th Edition. McGraw Hill India, 2011.

[14] J. J. Grainger and W. D. Stevenson, *Power System Analysis*. McGraw Hill, 1994.

[15] (2020). Ancilliary services, Australian Energy Market Operator, [Online]. Available: https://aemo.com.au/en/energy-systems/electricity/wholesale-electricity-market-wem/system-operations/ancillary-services.

[16] H. Bevrani and T. Hiyama, *Intelligent Automatic Generation Control*. CRC Press, 2011.

[17] P. Kundur, *Power System Stability and Control*. McGraw-Hill Inc., 1994.

[18] R. S. Sutton and A. G. Barto, *Reinforcement Learning*, M. Press, Ed. 2018.

[19] R. Bellman, "A markovian decision process," *Journal of Mathematics and Mechanics*, vol. 6, no. 5, pp. 679–684, 1957.

[20] ——, "The theory of dynamic programming," *Bulletin of the American Mathematical Society*, vol. 60, no. 6, pp. 503–515, 1954. [Online]. Available: https://projecteuclid.org/download/pdf_1/euclid.bams/1183519147.

[21] R. A. Howard, *Dynamic Programming and Markov Processes*. Cambridge, MA: MIT Press, 1960.

[22] R. Bellman and S. E. Dreyfus, "Functional approximations and dynamic programming," *Mathematical Tables and Other Aids to Computation*, 1959.

[23] I. H. Witten, "An adaptive optimal controller for discrete-time markov environments," *Information and Control*, vol. 34, no. 4, pp. 286–295, 1977, ISSN: 0019-9958. DOI: https://doi.org/10.1016/S0019-9958(77)90354-0. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0019995877903540.

[24] P. J. Werbos, "Building and understanding adaptive systems: A statistical/numerical approach to factory automation and brain research," *IEEE Transactions on Systems, Man, and Cybernetics*, vol. 17, no. 1, pp. 7–20, 1987.

[25] C. Watkins, "Learning from delayed rewards," PhD thesis, King's College, Cambridge, UK, May 1989. [Online]. Available: http://www.cs.rhul.ac.uk/~chrisw/new_thesis.pdf.

[26] G. Hinton, L. Deng, D. Yu, G. E. Dahl, A. Mohamed, N. Jaitly, A. Senior, V. Vanhoucke, P. Nguyen, T. N. Sainath, and B. Kingsbury, "Deep neural networks for acoustic modeling in speech recognition: The shared views of four research groups," *IEEE Signal Processing Magazine*, vol. 29, no. 6, pp. 82–97, 2012.

[27] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," in *Advances in Neural Information Processing Systems 25*, F. Pereira, C. J. C. Burges, L. Bottou, and K. Q. Weinberger, Eds., Curran Associates, Inc., 2012, pp. 1097–1105. [Online]. Available: http://papers.nips.cc/paper/4824-imagenet-classification-with-deep-convolutional-neural-networks.pdf.

[28] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," *arXiv 1409.1556*, Sep. 2014.

[29] C. Szegedy, Wei Liu, Yangqing Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich, "Going deeper with convolutions," in *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2015, pp. 1–9.

[30] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016, pp. 770–778.

[31] V. Mnih, K. Kavukcuoglu, D. Silver, A. A. Rusu, J. Veness, M. G. Bellemare, A. Graves, M. Riedmiller, A. K. Fidjeland, G. Ostrovski, S. Petersen, C. Beattie, A. Sadik, I. Antonoglou, H. King, D. Kumaran, D. Wierstra, S. Legg, and D. Hassabis, "Human-level control through deep reinforcement learning," *Nature*, vol. 518, no. 7540, pp. 529–533, Feb. 2015, ISSN: 1476-4687. DOI: 10.1038/nature14236. [Online]. Available: https://doi.org/10.1038/nature14236.

[32] T. P. Lillicrap, J. J. Hunt, A. Pritzel, N. Heess, T. Erez, Y. Tassa, D. Silver, and D. Wierstra, *Continuous control with deep reinforcement learning*, 2015. arXiv: 1509.02971 [cs.LG].

[33] J. Schulman, S. Levine, P. Abbeel, M. Jordan, and P. Moritz, "Trust region policy optimization," in *Proceedings of the 32nd International Conference on Machine Learning*, F. Bach and D. Blei, Eds., ser. Proceedings of Machine Learning Research, vol. 37, Lille, France: PMLR, Jul. 2015, pp. 1889–1897. [Online]. Available: http://proceedings.mlr.press/v37/schulman15.html.

[34] J. Schulman, P. Moritz, S. Levine, M. Jordan, and P. Abbeel, *High-dimensional continuous control using generalized advantage estimation*, 2015. arXiv: 1506.02438 [cs.LG].

[35] F. Rosenblatt, "The perceptron: A probabilistic model for information storage and organization in the brain.," *Psychological Review*, vol. 65, no. 6, pp. 386–408, 1958. DOI: 10.1037/h0042519. [Online]. Available: https://doi.org/10.1037/h0042519.

[36] N. Cohn, "The evolution of real time control applications to power systems," *IFAC Proceedings Volumes*, vol. 16, no. 1, pp. 1–17, 1983.

[37] H. Saadat, *Power System Analysis*, 3rd. PSA Publishing LLC, 2011.

[38] O. I. Elgerd and C. E. Fosha, "Optimum megawatt-frequency control of multiarea electric energy systems," *IEEE Transactions on Power Apparatus and Systems*, no. 4, pp. 556–563, 1970.

[39] N. Cohn, "Techniques for improving the control of bulk power transfers on interconnected systems," *IEEE Transactions on Power Apparatus and Systems*, vol. PAS-90, pp. 2409–2419, 6 Nov. 1971, ISSN: 0018-9510. DOI: 10.1109/TPAS.1971.292851.

[40] R. P. Aggarwal and F. R. Bergseth, "Large signal dynamics of load-frequency control systems and their optimization using nonlinear programming: I," *IEEE Transactions on Power Apparatus and Systems*, vol. PAS-87, pp. 527–532, 2 Feb. 1968, ISSN: 0018-9510. DOI: 10.1109/TPAS.1968.292049.

[41] ——, "Large signal dynamics of load-frequency control systems and their optimization using nonlinear programming: Ii," *IEEE Transactions on Power Apparatus and Systems*, vol. PAS-87, pp. 532–538, 2 Feb. 1968, ISSN: 0018-9510. DOI: 10.1109/TPAS.1968.292050.

[42] N. Cohn, "Some aspects of tie-line bias control on interconnected power systems," *Transactions of the American Institute of Electrical Engineers. Part III: Power Apparatus and Systems*, vol. 75, pp. 1415–1436, 3 Jan. 1956, ISSN: 2379-6766. DOI: 10.1109/AIEEPAS.1956.4499454.

[43] K. Ogata, *Modern Control Engineering*, 5th ed. Pearson, 2010.

[44] T. E. Bechert and N. Chen, "Area automatic generation control by multi-pass dynamic programming," *IEEE Transactions on Power Apparatus and Systems*, vol. 96, pp. 1460–1469, 5 Sep. 1977, ISSN: 0018-9510. DOI: 10.1109/T-PAS.1977.32474.

[45] C. Concordia, L. K. Kirchmayer, and E. A. Szymanski, "Effect of speed-governor dead band on tie-line power and frequency control performance," *Transactions of the American Institute of Electrical Engineers. Part III: Power Apparatus and Systems*, vol. 76, no. 3, pp. 429–434, Apr. 1957, ISSN: 2379-6766. DOI: 10.1109/AIEEPAS.1957.4499581.

[46] H. G. Kwatny, K. C. Kalnitsky, and A. Bhatt, "An optimal tracking approach to load-frequency control," *IEEE Transactions on Power Apparatus and Systems*, vol. 94, no. 5, pp. 1635–1643, Sep. 1975, ISSN: 0018-9510. DOI: 10.1109/T-PAS.1975.32006. [Online]. Available: https://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=1601608.

[47] O. Elgerd, *Electric energy systems theory: an introduction*, 2nd ed. McGraw-Hill, 1994.

[48] J. Morsali, K. Zare, and M. Tarafdar Hagh, "Appropriate generation rate constraint (grc) modeling method for reheat thermal units to obtain optimal load frequency controller (lfc)," in *2014 5th Conference on Thermal Power Plants (CTPP)*, Jun. 2014, pp. 29–34. DOI: 10.1109/CTPP.2014.7040611.

[49] C. S. Chang and W. Fu, "Area load frequency control using fuzzy gain scheduling of pi controllers," *Electric Power Systems Research*, vol. 42, no. 2, pp. 145–152, Aug. 1997. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0378779696011996.

[50] E. Cam and I. Kocaarslan, "Load frequency control in two area power system using fuzzy logic controller," *Energy Conversion and Management*, vol. 46, no. 2, pp. 233–243, Jan. 2005. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0196890404000779.

[51] E. Yesil, M. Guzelkaya, and I. Eksin, "Self tuning pid type load and frequency controller," *Energy Conversion*, vol. 45, no. 3, pp. 377–390, Feb. 2004. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0196890403001493.

[52] P. J. Fleming and C. M. Fonseca, "Genetic algorithms in control systems engineering," *IFAC Proceedings Volumes*, vol. 26, no. 2, Part 2, pp. 605–612, 1993, 12th Triennal Wold Congress of the International Federation of Automatic control. Volume 2 Robust Control, Design and Software, Sydney, Australia, 18-23 July, ISSN:

1474-6670. DOI: https://doi.org/10.1016/S1474-6670(17)49015-X. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S147466701749015X.

[53] C. S. Chang, W. Fu, and F. Wen, "Load frequency control using genetic algorithm based fuzzy gain scheduling of pi controllers," *Electric Machines and Power Systems*, vol. 26, no. 1, 1998. [Online]. Available: https://www.tandfonline.com/doi/abs/10.1080/07313569808955806.

[54] D. Rerkpreedapong, A. Hasanovic, and A. Feliachi, "Robust load frequency control using genetic algorithms and linear matrix inequalities," *IEEE Transactions on Power Systems*, vol. 18, pp. 855–861, 2 May 2003, ISSN: 1558-0679. DOI: 10.1109/TPWRS.2003.811005.

[55] S. P. Ghoshal, "Application of ga/ga-sa based fuzzy automatic generation control of a multi-area thermal generating system," *Electric Power Systems Research*, vol. 70, no. 2, pp. 115–127, 2004, ISSN: 0378-7796. DOI: https://doi.org/10.1016/j.epsr.2003.11.013. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0378779603002980.

[56] D. H. Nguyen and B. Widrow, "Neural networks for self-learning control systems," *IEEE Control Systems Magazine*, vol. 10, no. 3, pp. 18–23, Apr. 1990, ISSN: 2374-9385. DOI: 10.1109/37.55119.

[57] F. Beaufays, Y. Abdel-Magid, and B. Widrow, "Application of neural networks to load-frequency control in power systems," *Neural Networks*, vol. 7, no. 1, pp. 183–194, 1994, ISSN: 0893-6080. DOI: https://doi.org/10.1016/0893-6080(94)90067-1. [Online]. Available: http://www.sciencedirect.com/science/article/pii/0893608094900671.

[58] A. Demiroren, N. S. Sengor, and H. L. Zeynelgil, "Automatic generation control by using ann technique," *Electric Power Components and Systems*, vol. 29, no. 10, pp. 883–896, 2001. DOI: 10.1080/15325000152646505. eprint: https://doi.org/10.1080/15325000152646505. [Online]. Available: https://doi.org/10.1080/15325000152646505.

[59] H. L. Zeynelgil, A. Demiroren, and N. S. Sengor, "The application of ann technique to automatic generation control for multi-area power system," *International Journal of Electrical Power and Energy Systems*, vol. 24, no. 5, pp. 345–354, 2002, ISSN: 0142-0615. DOI: https://doi.org/10.1016/S0142-0615(01)00049-7. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0142061501000497.

Appendix A

Model Derivation for a Single Area Power System

When looking at a single area power system, there are three main components that the literature has a tendency to focus on:

1. **Governor:** used for controlling the angular velocity (and frequency) of the system;
2. **Turbine:** this is the steam turbine which provides the mechanical torque to drive the generator; and
3. **Generator load:** describes the electrical power that is produced and the electrical torque from connected loads.

A.1 Governor Model

The most important part of a speed governor are the two large masses (the pair of balls) which spin around a central axis. These masses are mechanically coupled to the the turbine drive shaft, so their angular velocity is a function of the turbine speed. Elgerd and Fosha's text [38] provides a really great schematic representation of the governing system for a steam turbine, shown in Figure A.1. This schematic is used to derive the plant model for the governor.

If we let A on the in the schematic be moved downward a little bit, Δy_A, the turbine power output will change by a directly proportional amount. Letting ΔP_C be the power increase, this can be expressed as:

$$\Delta y_A = k_C \Delta P_C \tag{A.1}$$

An increase in ΔP_C will cause the pilot valve to move up, and high pressure oil will flow onto the top of the main piston forcing it downwards. As the steam valve opens,

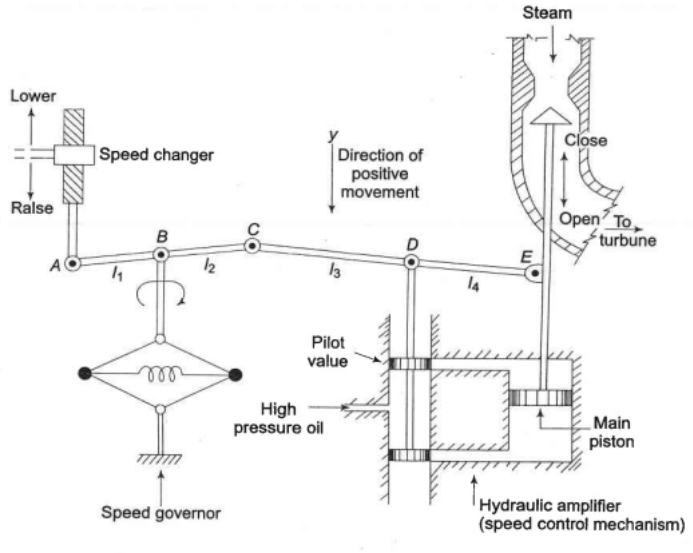

Figure A.1: A schematic of a steam governor

more steam will drive the turbine faster and causing the flyball governor to lower point B. Mathematically, the movement of C can be expressed as the result of two separate inputs:

1. Assuming that Δy_A is small, using similar triangles, it can be written that:

$$\Delta y_C = -\frac{l_2}{l_1}\Delta y_A \tag{A.2}$$

2. Given a frequency increase Δf point B will move downward so, assuming A is fixed, then using similar triangles it is clear that:

$$\Delta y_C = \frac{l_1 + l_2}{l_1}\Delta y_B \tag{A.3}$$

Letting $k_1 = \frac{l_2}{l_1}$, $k_2 = \frac{l_1+l_2}{l_1}k_2'$, and using equation A.1, the total movement in point C can be expressed as:

$$\Delta y_C = -k_1 k_C \Delta P_C + k_2 \Delta f \tag{A.4}$$

A similar analysis, considering movement of point C and E, can be undertaken to mathematically express the movement of point D. The analysis also makes use of similar triangles and results in the expression:

$$\Delta y_D = \frac{l_4}{l3 + l_4}\Delta y_C + \frac{l3}{l_3 + l_4}\Delta y_E \tag{A.5}$$

Letting $k_3 = \frac{l_4}{l_3+l_4}$ and $k_4 = \frac{l3}{l_3+l_4}$, this can be re-expressed as:

$$\Delta y_D = k_3 \Delta y_C + k_4 \Delta y_E \tag{A.6}$$

When there is some movement, Δy_D, of point D the ports of the pilot valve will open and high pressure oil will plow onto the cylinder causing some movement Δy_E. If point D moves up, high pressure oil will move point E down, and conversely if point D moves down, high pressure oil will move point E upwards. To simplify the dynamics of this scenario, the following assumptions are made:

1. Inertial reaction forces of the main piston and steam valve are negligible compared to the forces exerted on the piston by high pressure oil
2. Due to the first assumption, the rate of oil admitted to to the cylinder is proportional to the port opening Δy_D.

The volume of oil admitted to the cylinder is thus proportional to the time integral of Δy_D. Dividing the oil volume by the cross-sectional area of the piston:

$$\Delta y_E = k_5 \int (-\Delta y_D) dt \tag{A.7}$$

Taking the Laplace transform of equations A.4, A.6, and A.7 gives the following:

$$\Delta Y_C(s) = -k_1 k_C \Delta P_C(s) + k_2 \Delta F(s) \tag{A.8}$$

$$\Delta Y_D(s) = k_3 \Delta Y_C(s) + k_4 \Delta Y_E(s) \tag{A.9}$$

$$\Delta Y_E(s) = -k_5 \frac{1}{s} \Delta Y_D(s) \tag{A.10}$$

Algebraically manipulating A.8, A.9, and A.10 eliminates $\Delta Y_C(s)$ and $\Delta Y_D(s)$ and results in the following equation:

$$\Delta Y_E(s) = \frac{k_1 k_3 k_C \Delta P_C(s) - k_2 k_3 \Delta F(s)}{k_4 + \frac{s}{k_5}} \tag{A.11}$$

Equation A.11 can be re-expressed as:

$$\Delta Y_E(s) = \left[\Delta P_C(s) - \frac{1}{R} \Delta F(s) \right] \times \left(\frac{K_{sg}}{1 + T_{sg} s} \right) \tag{A.12}$$

where

$$R = \frac{k_1 k_C}{k_2} \tag{A.13}$$

$$K_{sg} = \frac{k_1 k_3 k_C}{k_4} \tag{A.14}$$

$$T_{sg} = \frac{1}{T_{sg}} \tag{A.15}$$

Equation A.12 is the model of the governor in the frequency domain. The parameter R is referred to as the speed regulation of the governor; the parameter K_{sg} is referred to as

the gain of the speed governor; and the parameter T_{sg} is referred to as the time constant of the speed governor.

The complete block diagram of governor model can be seen in Figure A.2 below.

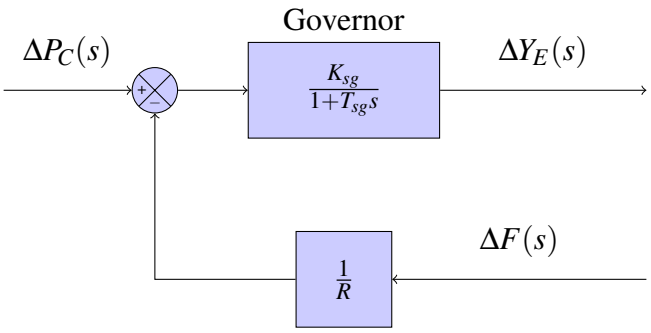

Figure A.2: Block diagram of the steam governor model in the frequency domain

A.2 Turbine Model

A.3 Generator Load Model

Appendix B

Model Derivation for a Two Area Power System

B.1 The title of the first section