ENG720: Research Proposal

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

Author: Shane Reynolds

Supervisor: Charles Yeo & Stefanija Klaric

Degree: Bachelor of Engineering (Honours)

Contents

1 Introduction & Background

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].

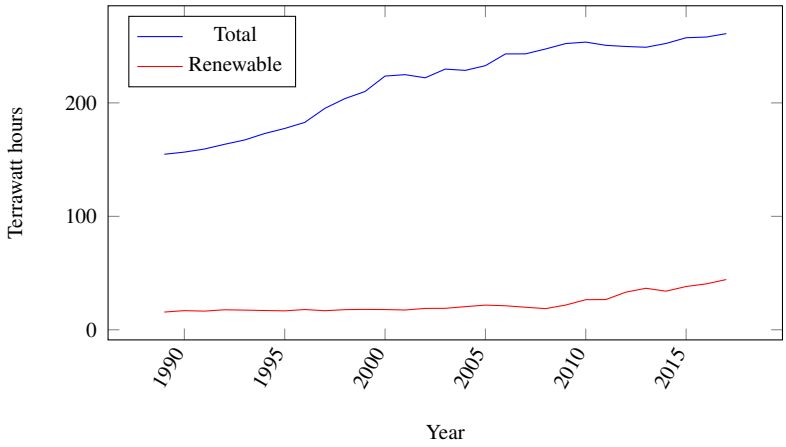


Figure 1: Power generation from renewable sources (light blue line), and total power generation (dark blue line) in Australia from 1977 to 2018.

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 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. 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).

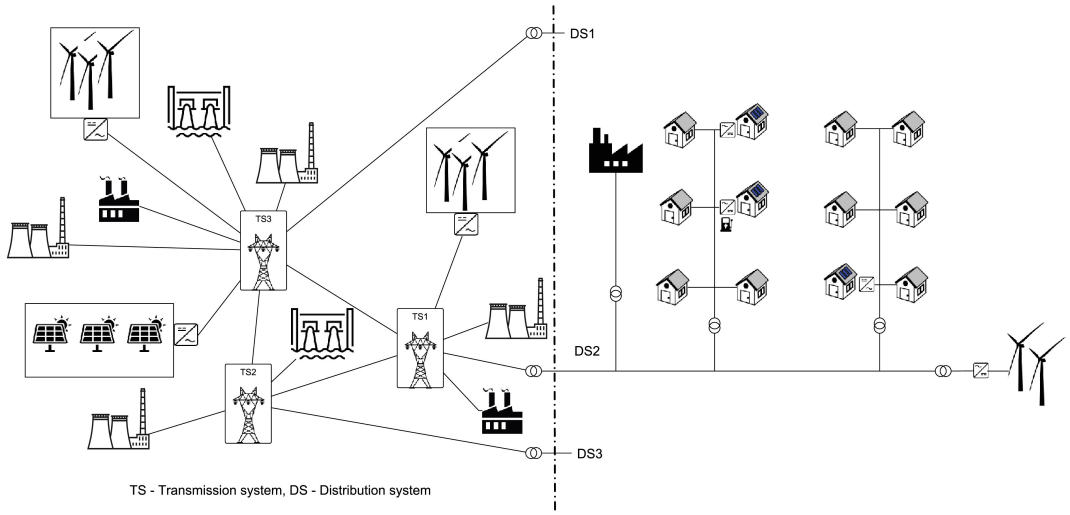


Figure 2: A single line diagram of a typical power system. The image shows points of generation from thermal and renewable sources, and the subsequent supply of generated energy to meet load demand through the transmission and distribution network [5].

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.

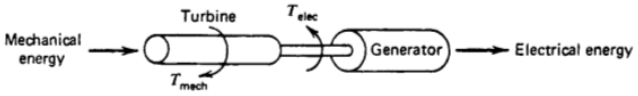


Figure 3: A thermal generation unit consisting of a prime mover (turbine), and a synchronous machine [6].

The essential elements of a thermal generator are a prime mover (such as a gas turbine) and a synchronous machine, as depicted in Figure 3. 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:

$$T_{mech} - T_{elec} = I\alpha \tag{1}$$

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].

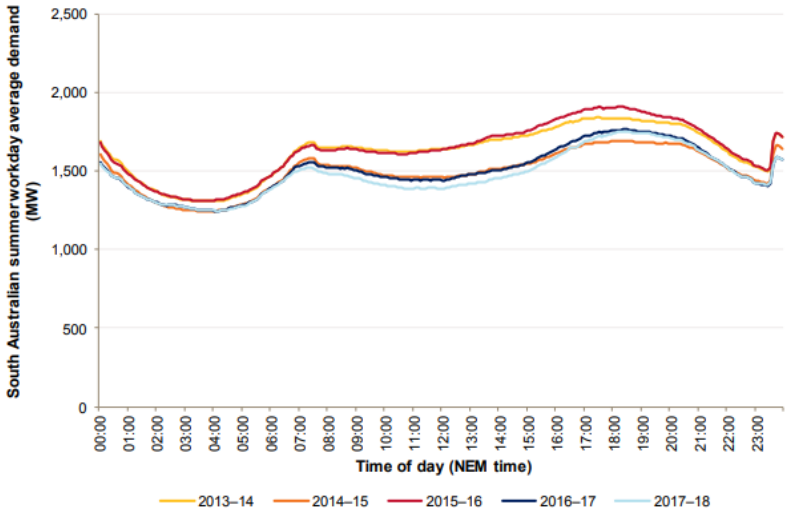


Figure 4: Weekday energy demand profile in South Australia during summer [11].

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 4, 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].

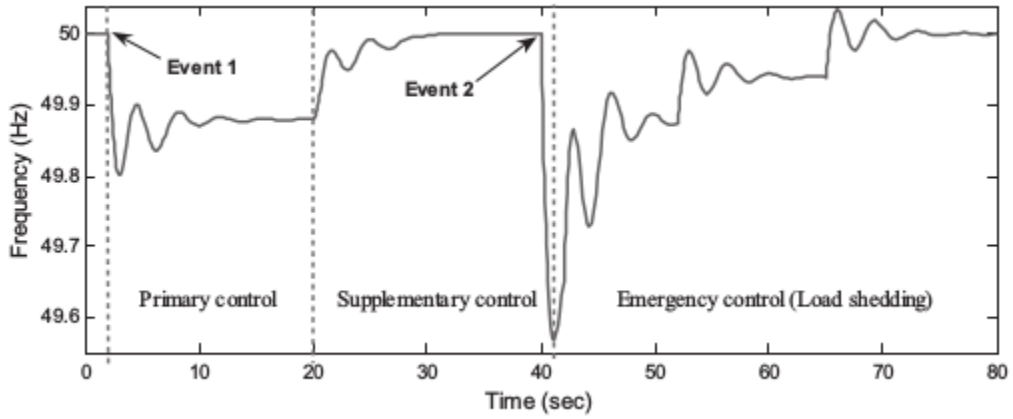


Figure 5: A minor frequency disturbance occurs at the 2 sec mark and primary control systems (governors) arrest the frequency drop. System frequency is adjusted to desired 50Hz operating level using AGC control of regulating units. This referred to as supplementary (or secondary) control in the literature. 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 5. 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 5. 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.

1.2 Frequency Control for a Single Area System

The power system model shown in Figure 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 6. Notice that for each area there is only one load and one generator.

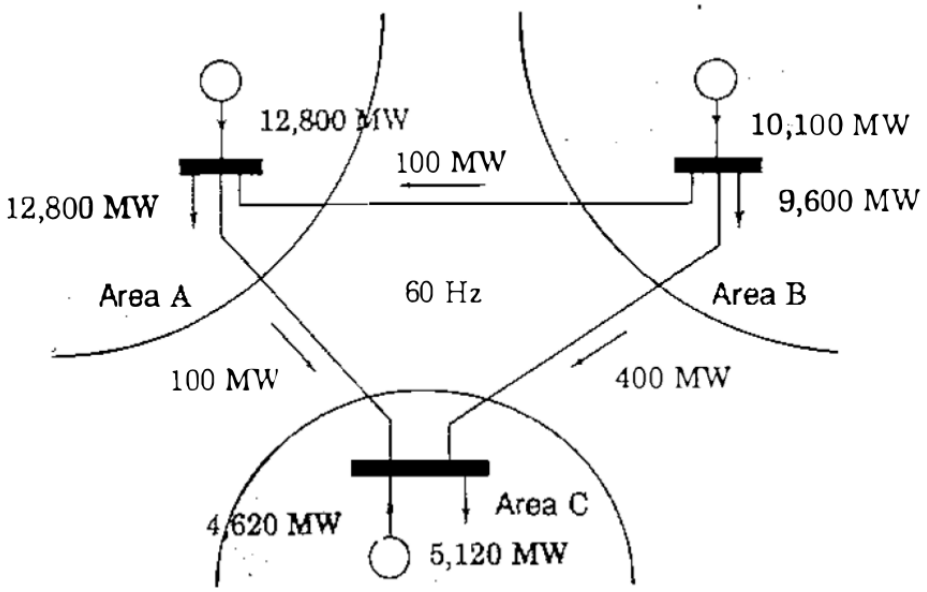


Figure 6: An example of three interconnected control areas in a 60Hz power system. The interconnections allow power to flow from one area to another, allowing generators to service loads from different areas. Each control area consists of several generators and loads, but are modelled with a single generator 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 7. This particular model provides generator models for regulating and non-regulating generators. The governor blocks are first-order linear models of the speed governors. The turbine blocks are first-order models of the turbines. The final block is the generator load, which is also a first order system. The AGC feedback loop uses an integral controller.

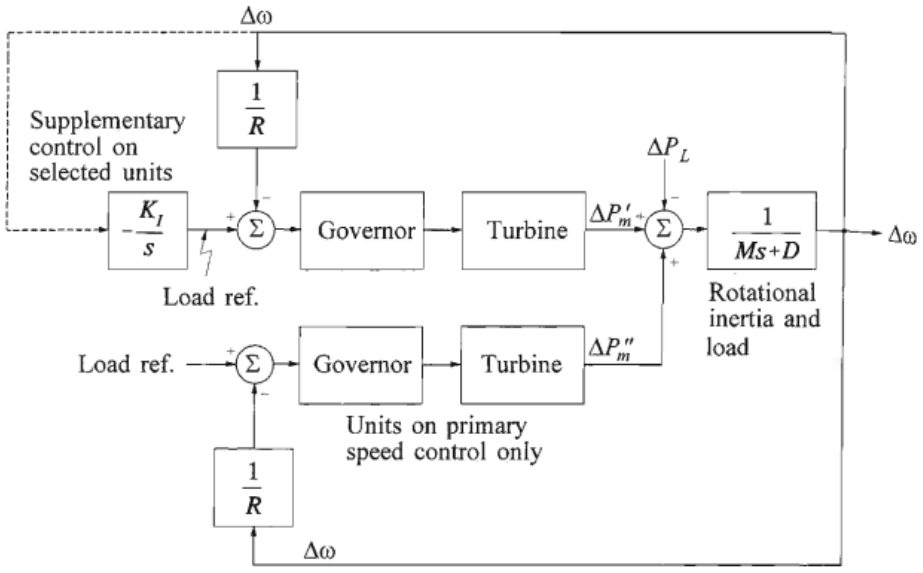


Figure 7: A classical feedback control approach for a single control area power system. The system is comprised of a first order models for both turbines, and generators. The governor controllers are also first order models. AGC is implemented using an integral control block in a feedback loop [17].

1.3 Frequency Control for Two Area System

The single area system presented in Section 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.

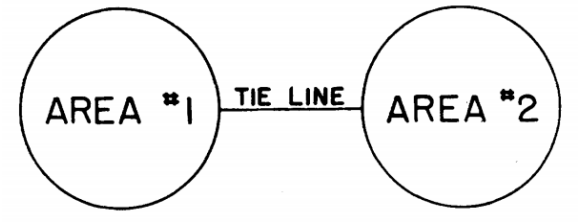


Figure 8: A two area power system comprised of generators and load connected via a tie line. Power flows from one area to the other depending on the power demands.

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].

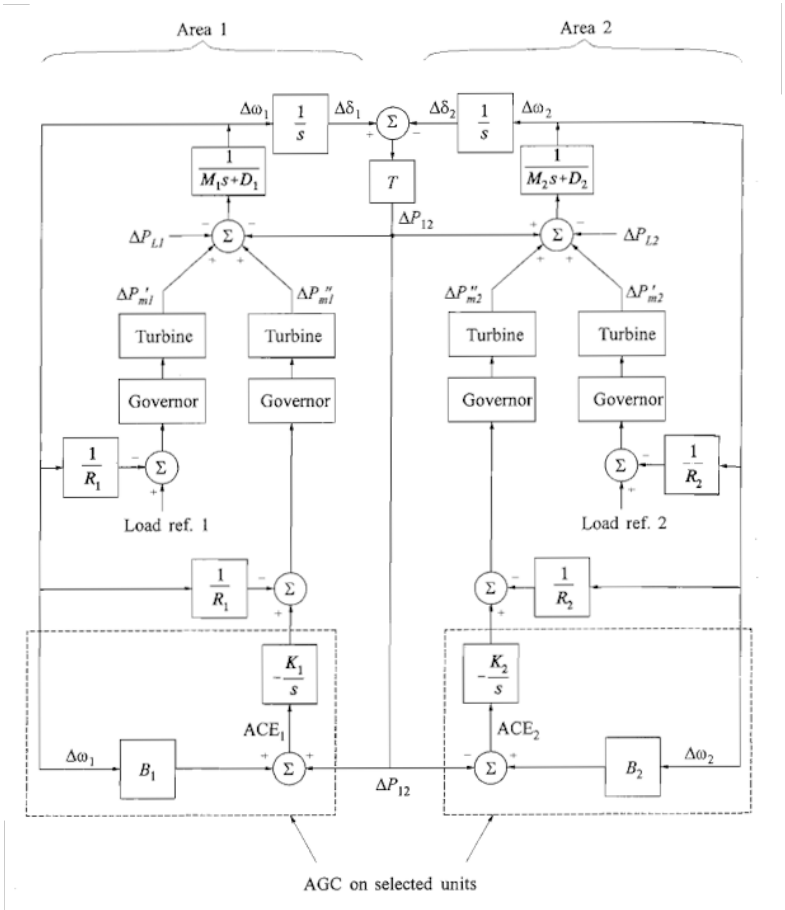


Figure 9: A classical feedback control approach to a two area power system [17].

The simplest model that includes tie lines is the two area power system, shown in Figure 8. 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 7, 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 9.

1.4 Reinforcement Learning

Reinforcement Learning (RL) is a branch of machine learning that is concerned with how agents make sequential decisions to maximise some notion of a cumulative reward. It is a simple idea that allowed Google's DRL agent, AlphaGo, to beat the world's best players in a game of Go [18]. One of the important aspects of RL is the underlying architecture of the model, which allows for generalisation to many different applications, albeit the models do require training with different data sets for each application. A brief overview of key architectural components of RL can be found in §§1.4.1 and 1.4.2. §1.4.3 gives details on how these components are implemented to build an agent that can perform a control activity for some application.

1.4.1 Markov Decision Process

Suppose an agent exists in some environment that is comprised of many discrete states, $s \in S$, such that S denotes the state space. At any discrete point in time the agent can take an action $a \in A$, where A denotes the action space. When the agent takes an action in a given state, the agent receives some reward, denoted with $r \in R$, where R is the reward set. If an agent is in a given state, s, and takes and action, a, this will transition the agent to a new state, s', and yield reward, r, with some given probability — these are referred to as state transition probabilities. Transition probabilities are denoted as follows:

$$P(S_{t+1} = s', R_{t+1} = r \mid S_t = s, A_t = a) \tag{2}$$

The set of parameters outlined above, and equation 2, make up a framework referred to as a Markov Decision Process (MDP) [19]. The MDP framework is important to under-

standing how RL works; however, it must be noted that it is not necessary for the agent to have any information about the state transition dynamics to develop an effective control regime.

1.4.2 Return, Episodes, and Policy

As the agent takes actions at each discrete time step, it receives a reward. The cumulative sum of this reward is referred to as the return [20]. For N discrete time steps the return is denoted as:

$$G_t = r_t + r_{1+1} + r_{t+2} + \ldots + r_{N-1} \tag{3}$$

Often it is convenient to make future rewards less important than more immediate rewards. This is achieved by multiplying each reward in the sequence by a discount factor, $\gamma \in [0,1]$. Equation 3 then becomes:

$$G_t = r_t + \gamma\, r_{1+1} + \gamma^2 r_{t+2} + \ldots + \gamma^{N-1} r_{N-1} = \sum_{k=0}^{N-1} \gamma^k r_{t+k} \tag{4}$$

The duration of time that an agent will cumulate reward is referred to as an episode. An episode is made up of a beginning, a middle, and an end. Typically, this consists of an RL agent beginning in some initial state. As time passes the agent takes actions, undergoes state transitions, and collects rewards. The episode concludes when the agent reaches a terminal state. At the episode conclusion, the agent receives it's cumulative reward [21].

Finally, in order for the robot to act within the environment, it needs to have a policy. A policy, π, is defined as a mapping from states to actions, i.e. a rule which determines what action the robot will take for a given state. A deterministic policy, $\pi(s)$, maps a single action to a single state. A stochastic policy, $\pi(a|s)$, defines a probability distribution over the actions for a given state. An optimal policy, denoted π^*, is a policy which will maximise the cumulative reward that the agent receives over an episode [22]. It can be thought of as the best control policy an agent can have for the given the environment and cumulative reward function.

1.4.3 How Does an RL Agent Learn?

The main objective of RL is to develop an optimal policy. There are many algorithmic approaches to building an optimal policy. One of the most common implementations is called Q-learning. This approach focuses on finding q-values for each state-action pair. A q-value can be thought of as an ordinal value that is discovered and assigned to a state-action pair that tells the agent how important an action is relative to the rest of the actions in a given state. The agent normally starts with a randomised policy meaning that all the q-values are set to zero. This will lead the agent to take actions at random and explore the

state-action space. Higher q-values are then assigned to state-action pairs that the agent classifies as useful for building a high cumulative reward. Similarly, the agent assigns low q-values to state-action pairs that do not lead to high cumulative rewards. This process is akin to the agent modifying its policy. The q-value modification process is iterated over many episodes and eventually the agent policy converges on an optimal policy. Often the q-values are presented in a tabular format that the literature refers to as a Q-table. An example of a Q-table can be seen in Figure 10. Rows represent different states, and columns represent different actions. Values in each cell provide an ordinance on how valuable each action is for a given state. The agent only needs to understand inputs that uniquely define a state, and the actions it can take in order to learn an optimal policy. It is not necessary for the agent to know the state transition dynamics of the system, described by Equation 2. Therefore, an agent can learn to control a system for which is does not have a mathematical model.

Figure 10: The Q-table on the left shows the initialised policy when the agent begins learning. The middle and rightmost Q-tables show the agent developing an understanding of which actions are valuable in which states.

1.5 Deep Reinforcement Learning

For low dimensional state-action spaces RL leads to policy convergence in a reasonable time frame. As state-action space dimensionality increases Q-Learning models experience difficulty. Owing to increasing computing demands, it becomes difficult for the discrete RL algorithm to visit every state-action pair unless the computational power is dramatically increased. The impace of dimensionality on Q-learning has been previously reported. Stopping policy discovery prior to optimal policy convergence is not an op-

tion under problems with high dimensionality as this results in sparse Q-tables (mostly populated with zeros).

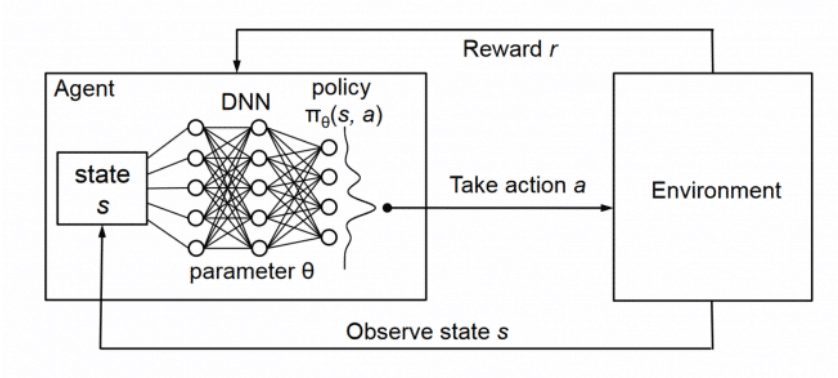


Figure 11: The agent interacts with the environment by taking actions, which affect the state it is in and the reward it receives. The rewards allow agent to adjust the weights in the neural net to build better policies.

The interpretation of sparseness in Q-tables is that the agent does not have a complete knowledge of optimal actions for every given state. This leads to sub-optimal policies. In order to improve policy, for RL problems with high dimensional state spaces, the discrete Q-table is replaced with a function approximator known as a neural network. A high level overview of the architecture can be seen in Figure 11. It is the neural network architecture in which the agent's policy is implemented. The agent learns by adjusting weights in the neural network to change the policy. This approach is significant because neural networks are good at generalising, and hence the agent does not need to visit every state action pair to be able to make good decisions.

2 Research Aims

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.

3 Scope

The proposed research is primarily concerned with the task of load following using DRL and classical control agents. Load following is defined as maintaining system frequency within PWC's allowable region of 49.80 to 50.20Hz for normal operating conditions. Tasks involving frequency restoration following a disturbance event may be considered,

time permitting. The key performance aspects that will assessed are the controller's ability to:

- maintain system frequency to the desired nominal 50Hz value;
- maintain the tie line power flow between control areas at a scheduled value.

The research will focus on two area power systems. Each power area will consist of one regulating generator, and one stochastically fluctuating demand profile. The research will primarily consider the role frequency in the system; however, other system variables may be used as input features under agent training and inference regimes. Comparison of DRL agent performance will be made against theoretical models of classical control architectures. Performance against practical control architectures implemented by PWC (or other utilities) will not be considered. Research will be conducted in a simulated environment. Agent performance on real hardware will not be explored.

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. De-

termining 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 [23]. 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 [24]. 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.

5 Deliverables Specification

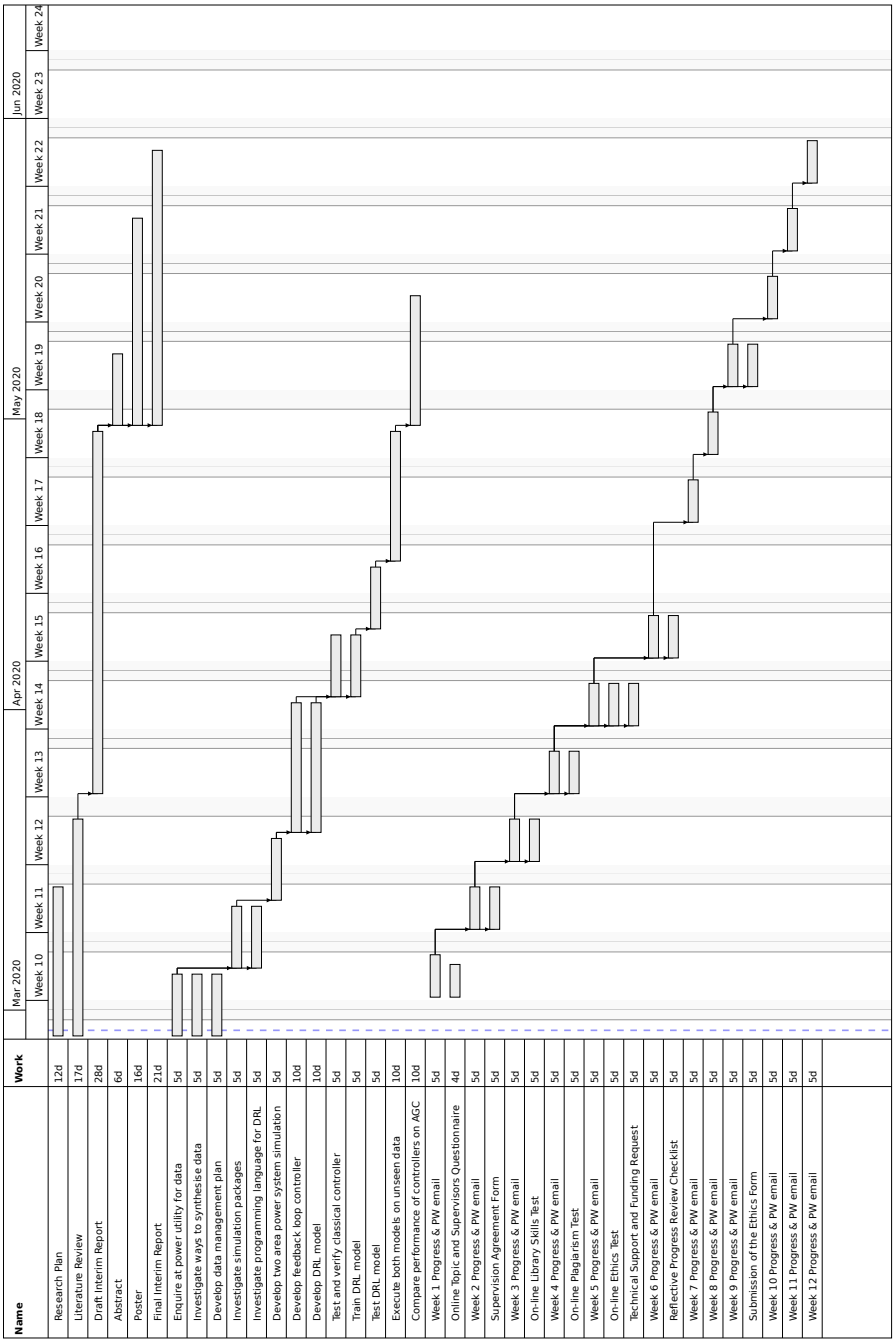
The main outcomes from this research is a conclusion about the feasibility of using DRL agents to provide AGC for a two area power system. The expectation is that the DRL agent will perform at least as well as a classical control approach. This expectation is based on the existing research in this field, which has shown that standard RL agents can perform AGC at least as well as classical control methods [25]. If the research meets expectations, it should provide a pathway for the investigation of novel DRL models that could improve power system control. The ultimate goal would be to have a DRL agent that is always learning, so that it can adapt control strategies to unseen system conditions providing a more flexible and less brittle controller. This would help to avoid problems like that seen in the Alice Springs system blackout event.

A schedule of actual physical deliverable items can be found in Table 1, in chronological order.

Table 1: A table of deliverable items and their respective due dates

Deliverable Item	Due Date
Online Topic and Supervisor Questionnaire	05/03/20
Week 1 Progress report	06/03/20
Research Plan	13/03/20
Week 2 Progress report	13/03/20
Supervision Agreement Form	13/03/20
Literature Review	20/03/20
Week 3 Progress report	20/03/20
On-Line Library Skills Test	20/03/20
Week 4 Progress report	27/03/20
On-line Plagiarism Test	27/03/20
Week 5 Progress report	03/04/20
On-line Ethics Test	03/04/20
Technical Support and Funding Request	03/04/20
Week 6 Progress report	10/04/20
Reflective Progress Review Checklist	10/04/20
Week 7 Progress report	24/04/20
Draft Interim Report	29/04/20
Week 8 Progress report	01/05/20
Abstract	07/05/20
Week 9 Progress report	08/05/20
Submission of Ethics Form	08/05/20
Week 10 Progress report	15/05/20
Poster	21/05/20
Week 11 Progress report	22/05/20
Final Interim Report	28/05/20
Week 12 Progress report	29/05/20

6 Timeline


Name
Work
Mar 2020
Apr 2020
May 2020
Jun 2020
Week 10
Week 11
Week 12
Week 13
Week 14
Week 15
Week 16
Week 17
Week 18
Week 19
Week 20
Week 21
Week 22
Week 23
Week 24
Research Plan 12d
Literature Review 17d
Draft Interim Report 28d
Abstract 6d
Poster 16d
Final Interim Report 21d
Enquire at power utility for data 5d
Investigate ways to synthesise data 5d
Develop data management plan 5d
Investigate simulation packages 5d
Investigate programming language for DRL 5d
Develop two area power system simulation 5d
Develop feedback loop controller 10d
Develop DRL model 10d
Test and verify classical controller 5d
Train DRL model 5d
Test DRL model 5d
Execute both models on unseen data 10d
Compare performance of controllers on AGC 10d
Week 1 Progress & PW email 5d
Online Topic and Supervisors Questionnaire 4d
Week 2 Progress & PW email 5d
Supervision Agreement Form 5d
Week 3 Progress & PW email 5d
On-line Library Skills Test 5d
Week 4 Progress & PW email 5d
On-line Plagiarism Test 5d
Week 5 Progress & PW email 5d
On-line Ethics Test 5d
Technical Support and Funding Request 5d
Week 6 Progress & PW email 5d
Reflective Progress Review Checklist 5d
Week 7 Progress & PW email 5d
Week 8 Progress & PW email 5d
Week 9 Progress & PW email 5d
Submission of the Ethics Form 5d
Week 10 Progress & PW email 5d
Week 11 Progress & PW email 5d
Week 12 Progress & PW email 5d

7 Resources

In order to reach research objectives a computer with a Linux operating system featuring RAM, and a Graphics Processing Unit (GPU) will be required. The exact specifications of these computational resources are not yet known, and will be dependent on the size of the datasets, and the amount of computation required to train the DRL agent. Currently, access has been provided to an Intel Quad Core 3.2GHz i5 CPU machine with 8GB of RAM, Nvidia GTX960 GPU, and Ubuntu operating system. It is expected these resources will be sufficient; however, in the event that greater computational power is needed, access to a virtualised environment, such as Amazon Web Services (AWS) set up to conduct DRL research, may be required. Using such a system can be expensive (1 hour of compute is approximately $20AUD); however, AWS will often supply student research with a monetary credit. Approximately 10 hours of compute would be reasonably expected to achieve desired training outcomes for DRL agent. This allows for errors made in model training parameter specification, and subsequent retraining. Table 2 shows an itemised list of resources needed to complete this research, and their associate costs.

Table 2: Itemised list of resources needed to complete research and associated costs.

Equipment	Details	Cost
Intel Quad Core 3.2GHz i5 CPU	pre-owned hardware	$0
8GB of RAM	pre-owned hardware	$0
Nvidia GTX960 GPU	pre-owned hardware	$0
Ubuntu operating system	open source software	$0
AWS GPU Large Instance	$20 @ 10 hours	$200
Python 3.7 (Open AI Env)	open source software	$0
Matlab	CDU VM licence	$0
Total		$200

References

[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. Wollenberg, and G. 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] D. Silver, A. Huang, C. J. Maddison, A. Guez, L. Sifre, G. van den Driessche, J. Schrittwieser, I. Antonoglou, V. Panneershelvam, M. Lanctot, S. Dieleman, D. Grewe, J. Nham, N. Kalchbrenner, I. Sutskever, T. Lillicrap, M. Leach, K. Kavukcuoglu, T. Graepel, and D. Hassabis, "Mastering the game of go with deep neural networks and tree search," *Nature*, vol. 529, pp. 484–503, 2016. [Online]. Available: http://www.nature.com/nature/journal/v529/n7587/full/nature16961.html.

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

[20] *Open ai spinning up*, https://spinningup.openai.com/en/latest/spinningup/rl_intro.html, 2018. [Online]. Available: https://spinningup.openai.com/en/latest/spinningup/rl_intro.html.

[21] L. P. Kaelbling, M. L. Littman, and A. W. Moore, "Reinforcement learning: A survey," *Journal of Artificial Intelligence Research*, vol. 4, pp. 237–285, 1996. [Online]. Available: https://arxiv.org/pdf/cs/9605103.pdf.

[22] R. Bellman, "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.

[23] K. Ogata, *Modern Control Engineering*, 5th Edition, Test, Ed. Pearson, 2010.

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

[25] T. P. I. Ahamed, N. Rao, and P. S. Sastry, "A reinforcement learning approach to automatic generation control," *Electrical Power Systems Research*, vol. 63, pp. 9–26, Mar. 2002.