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 there was approximately 261TWh of power generation in the Australian electricity sector. Renewables contributed to 19% of the total generation, increasing from 15% in 2017. The Department of Industry, Science, Energy and Resources have noted a small trend of increasing renewable energy penetration year on year in the electricity generation market since 2008, as shown in Figure 1 [1].
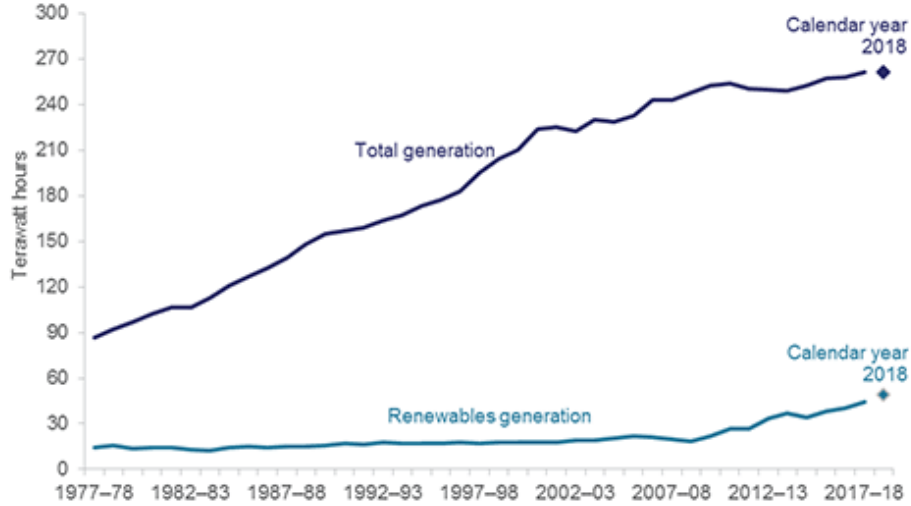

Figure 1: Timeseries of Terrawatt hours of generation for renewable sources of generation, and total sources of generation in Australia from 1977 to 2018.

One of the benefits of transitioning from thermal sources of generation to renewable sources is reduced greenhouse gas emissions [2], however, this transition is not without its drawbacks. Increased reliance on renewable generation sources poses some challenges regarding power system stability. A recent example is the system failure resulting from an event cascade, triggered by cloud cover shadowing a solar array in Alice Springs. The system failure resulted in a blackout occurring in Alice Springs for approximately 8 hours [3]. The Entura report highlighted that poor control policies were one of the many factors contributing to the blackout. In this instance, a generator provisioned to ramp up in the event of cloud cover limiting solar array output was unable to be controlled. Moreover, generators that were still under the control regime were issued operating set points above their rated capacity, which eventually resulted in thermal overload and subsequent tripping from the protection system [4].

Currently employed control methods use classical feedback loop techniques. They can be brittle when faced with system changes, or scenarios which they were not designed for. A better controller would be one that can learn and adapt it's control system to an unseen system or event. to a system, given some broad control objective. This research proposes a Deep Reinforcement Learning agent for controlling the frequency of a power system consisting of multiple generators, and multiple load demands with stochastic profiles.

2

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 left hand side of the diagram shows thermal generation units, such as coal and nuclear, in addition to renewable sources of generation, like wind and solar. The right hand side of the figure shows the distribution network and the consumers of generated energy: industry and households.
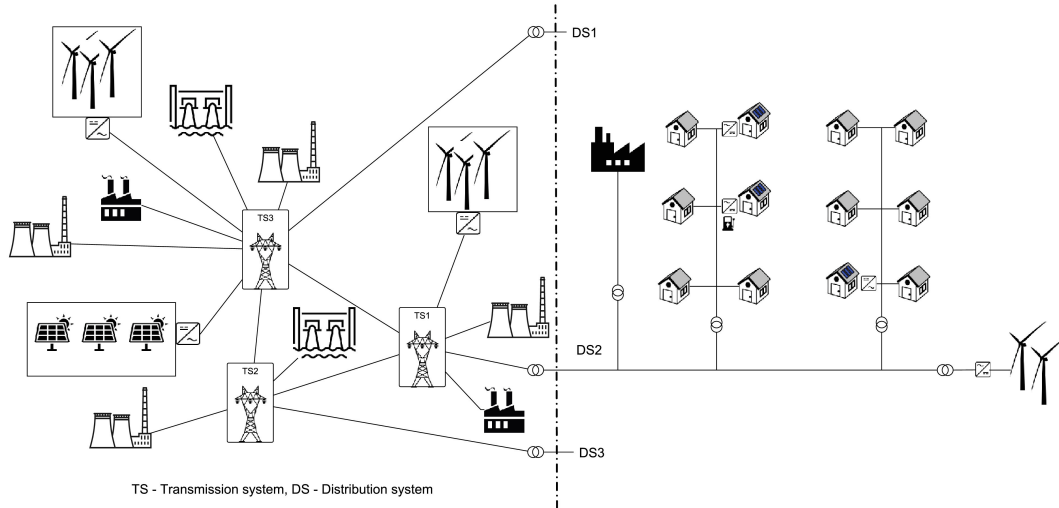

Figure 2: A single line diagram of a typical power system taken from [5]. The image shows points of generation from thermal and renewable sources, and the subsequent supply of generated energy to meet load demand through the transmission and distribution network.

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 it is useful to first consider the basic operation of a single thermal generator.
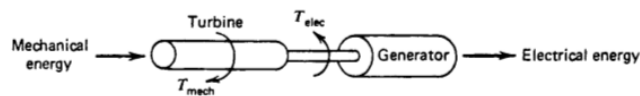
Figure 3: A thermal generation unit consists of a prime mover (turbine), and a synchronous machine. This image was taken from [6].

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 which depends on the size

3

of the load demand from households and industry. 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:

$$\sum T_i = I\alpha \tag{1}$$

Equation 1 shows that 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 some 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. What makes this situation interesting is that, at any point in time, the total electrical load demand will fluctuate stochastically as businesses and households switch grid connected devices or motors on and off. The implication 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 49.85Hz 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 49.80Hz to 50.20Hz [8]. Operation outside of specified ranges 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].
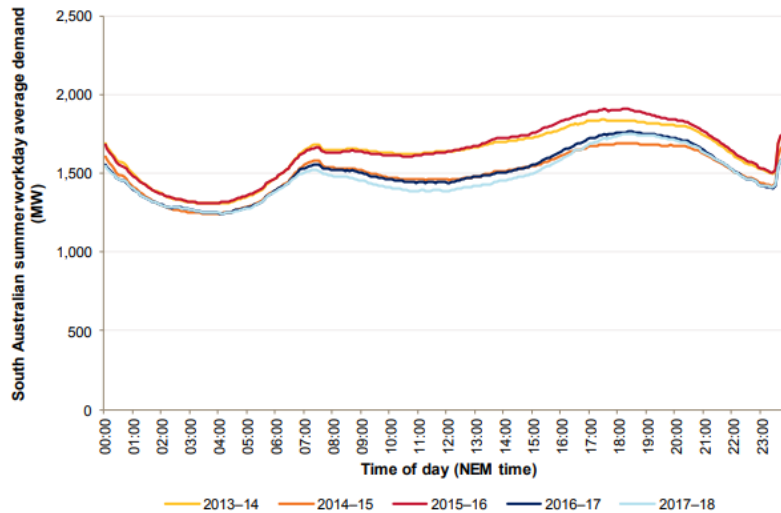

Figure 4: Intraday timeseries of weekday energy demand profile in South Australia during summer [11].

Protection schemes tripping equipment from the network is undesirable since this can leave households and industry without power, resulting in economic loss. Further, if disconnections are uncontrolled then this can reduce system stability further [10]. System controllers, such as the AEMO and PWC, are therefore 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 [13]. A regulating unit is a generator that has capacity to increase or decrease mechanical torque T_{mech}. If the system controller has a sufficient number of regulating units 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 to load follow is referred to as the provision load following ancillary services [15]. Load following control adjusts regulating units slightly to match supply perfectly with a demand load profile, like that shown in Figure 4. Using a regulator to restore the system after a disturbance is referred to as providing spinning reserves [15].
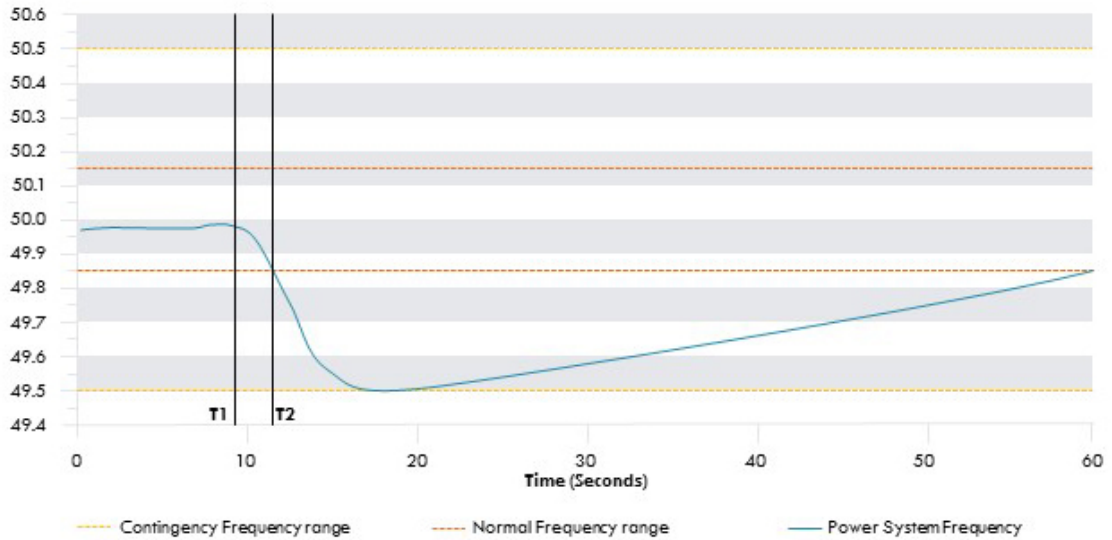

Figure 5: A frequency disturbance occurs just before the 10 second mark, and regulating units ramp up their generation to first arrest the disturbance, and provide the subsequent correction, returning system frequency to 50Hz.

5

When used in this 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 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 which 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]. The total network is therefore comprised of many interconnected control areas. An example of a series of interconnected control areas can be seen in Figure 6. Notice that for each area there is only one load and one generator. 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].
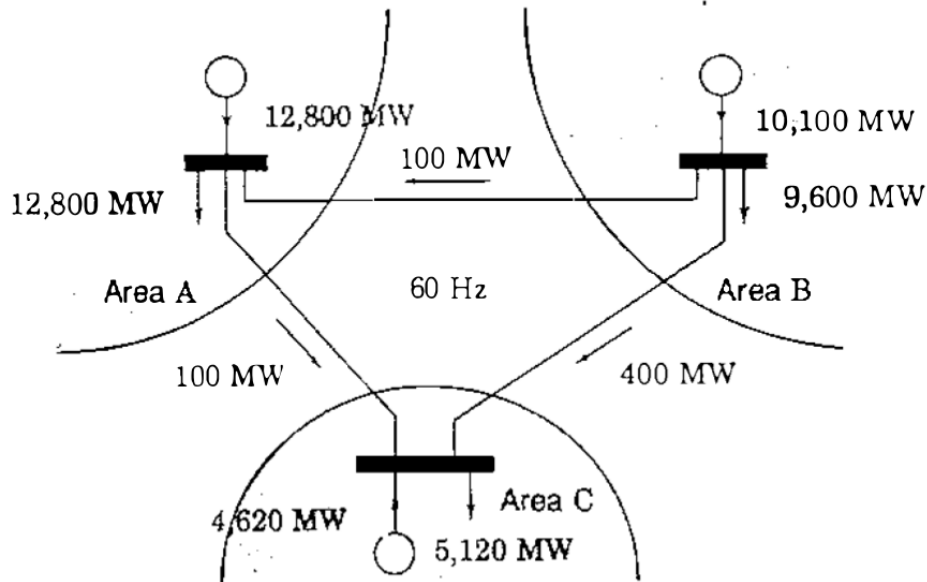
Figure 6: An example of three interconnected control areas in a 60Hz power system. The interconnections allow power to flow from one area to another, allowing generators to service loads from different areas. Each control area is consists of many generators and loads, but are modelled with a single generator and single load, respectively [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

6

comprised of 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 are aggregated to a single generator. This simple system is a classic example and well understood. It is generally acknowledged that a governor feedback control regime can successfully perform frequency control of a single area power system [6], [13], [14]. Most introductory textbooks on power systems cover governor control of this system. A particularly well laid out approach to developing linear models for the turbine, generator load, and governor can be found in [13] - the full model is shown in Figure 7. The leftmost block is a first-order linear model of the speed governor, the second block is a first-order model of the turbine, which the controller directly acts on. The final block is the generator load, which is also a first order system. The over all system model is a second order linear model, with a first order controller.
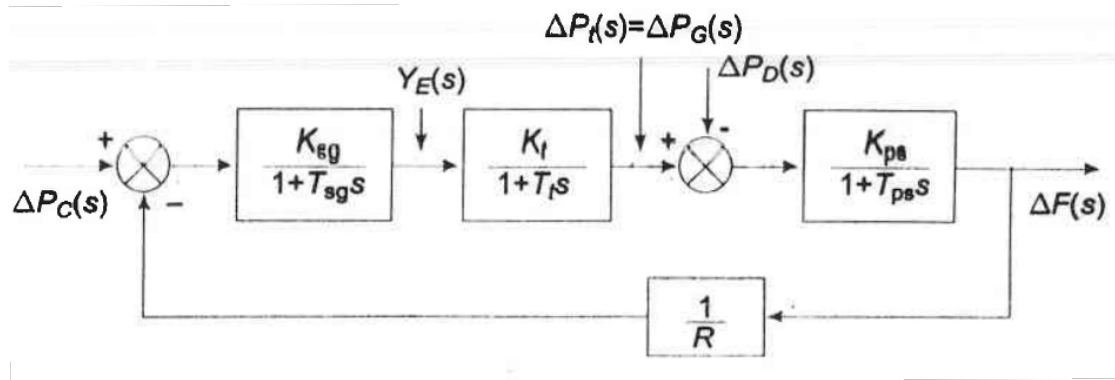

Figure 7: A classical feed back control approach to a second order linear system. The second order system is comprised of a first order model for both the turbine, and generator load. The controller is modelled as a first order system [13].

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 in controlling power system frequency, however, a single area model is too simple. 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 contract.

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 which 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. Simply relying on governor control in each area will not satisfy the control objective. To see this, consider control area A supplying a 50MW load, with a contract to supply control area B with 20MW over the tie line
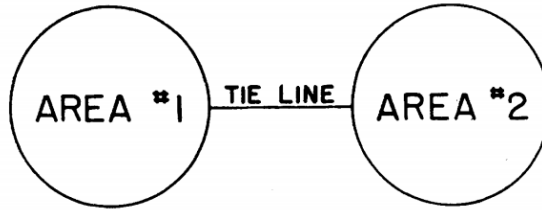
Figure 8: Two area power system is comprised of generators and load connected via a tie line. Power flows from one area to the other depending on economic contracts.

connecting them. If control area B also has a 50MW load, then it is supplying only 30MW to satisfy the demand in this area. Now suppose a 30MW load increase was observed in the demand for area A. Relying on governor control will see generators from both area A and area B speed up in response to this increased load. Ultimately the increased power demands will be met, however, the power transfer over the tie line is likely to be less than the contracted 20MW value, which is problematic. Contract violations due to system instability and control issues do not allow for a stable market in which energy can be reliably traded. Fortunately, two 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. This measures the distance from a target frequency as well as the deviation from tie-line contractual obligations. The implementation of this control system is shown in Figure 9.
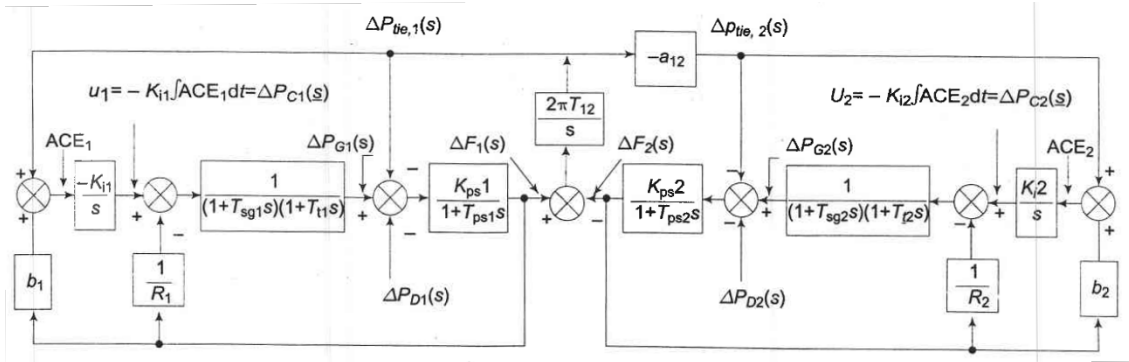

Figure 9: A classical feed back control approach to a two power area system. Image taken from [13].

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 Deep Mind to beat the worlds best players in the game of Go [16]. One of the important aspects of RL is the underlying architecture of the model allows for generalisation to many different applications, albeit the models do require training with different data sets for each application. Subsections 1.4.1 and 1.4.2 provide a brief overview of key architectural components of RL, and subsection 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 which 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, make up a framework referred to as a Markov Decision Process (MDP) [17]. The MDP framework is important to understanding 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 robotic agent takes actions at each discrete time step, it receives a reward. The cumulative sum of this reward is referred to as the return [18]. The return is denoted, for N discrete time steps, 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:

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

The duration of time that an agent will cumulate reward is referred to as an episode. An episode is made up of a beginning, 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 [19].

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, that is, 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 [20]. 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 which tells the agent how important an action is relative to the rest of the actions in a given state. To discover q-values, 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 to explore the state-action space. Higher q-values are assigned to state-action pairs that the agent determines are useful for building a high cumulative reward. Similarly, the agent assigns low q-values to state-action pairs which do not lead to high cumulative rewards. This process is akin to the agent modifying its policy. The q-value modification process is repeated for 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. More concretely, 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.(REFERENCE)

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 models like Q-Learning experience difficulty. The main reason for this is that it becomes difficult for the discrete RL algorithm to visit every state-action pair unless the computational power is dramatically increased. Researchers often refer to this as the curse of dimensionality. Stopping policy discovery early for problems with high dimensionality results in Q-tables that are sparse (mostly populated with zeros).
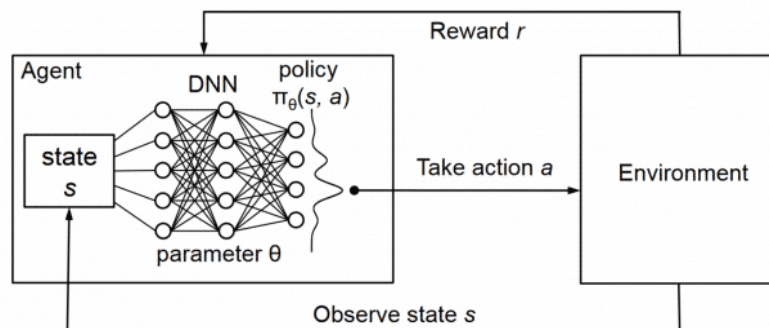

Figure 11: The agent interacts with the environment by taking actions, which affect the state it is in and the reward it receives. The rewards allow agent to adjust the weights in the neural net to build better policies.

The interpretation is that the agent does not have complete knowledge of optimal actions for every given state, leading to the derivation of sub-optimal policies. To get around this problem, 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 policy is implemented. As the agent learns, it adjusts weights in the neural network to change the policy. This is approach is powerful 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 classical engineering control methodologies against a control system based on a DRL agent, when tasked with performing AGC for load following tasks on a two area power system. The ultimate research outcome will be to provide comment on the feasibility of using DRL agents for this purpose.

3 Scope

The proposed research is only 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 49.80Hz to 50.20Hz for normal operating conditions. Task involving frequency restoration following a disturbance event may be considered, time permitting. The key performance aspects that will assessed, for DRL and classical, 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 the scheduled value.

The research will not consider power systems larger than a two area power system. Each power area will consist of one regulating generator, and one stochastically fluctuating demand profile. The research will not consider the control of system variables other than frequency, however, note that 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 an entirely 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 contractual obligations 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, such as PWC, in the form of a time series dataset with a large number of features, and high sample rate. 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 provide advice on where data will be stored, and how data will be treated (or disposed of) once the research is concluded.

In the event data cannot be acquired from a utility, a synthetic data set may need to be derived. This would be achieved by understanding key statistical parameters of a typical load demand profile, and using these to create a stochastic 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 synthetic 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 Section 4.1, securing load demand profile datasets from a major utility, or developing synthetic load profile datasets based on local load profile characteristics is an important aspect of the research and will need to be conducted as early as possible. Similarly, investigating suitable software packages (open source or commercial) to develop the power system simulation model, and investigating suitable programming languages to implement DRL agent should be explored. This information will most likely be found when exploring the field literature. It will be important to understand how other researchers integrated the DRL agent implementation 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 highly non-linear systems. Classical engineering system modelling techniques will be employed for power system model development [21]. An area of interest in this domain 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. This is an application of classical engineering control theory taught in most undergraduate engineering courses. A DRL agent will be developed using an architecture that provides for continuous input signals and continuous output signals. There are a number of well established DRL models presented in texts like Sutton and Barto that will be explored to determine the most appropriate approach. Time permitting, a DRL model that uses discretised 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 the a utility, or by synthesis. Metrics will be selected to measure the performance of both controllers. Choice of metrics will be informed by earlier research (mentioned above). Control models will be compared, and differences in their 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 to secure data

2. Investigate ways to synthesise data

3. Develop data management plan

4. Literature review centred on three central themes:

   (a) Automatic Generation Control

   (b) Deep Reinforcement Learning

   (c) Applications of DRL to Automatic Generation Control

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

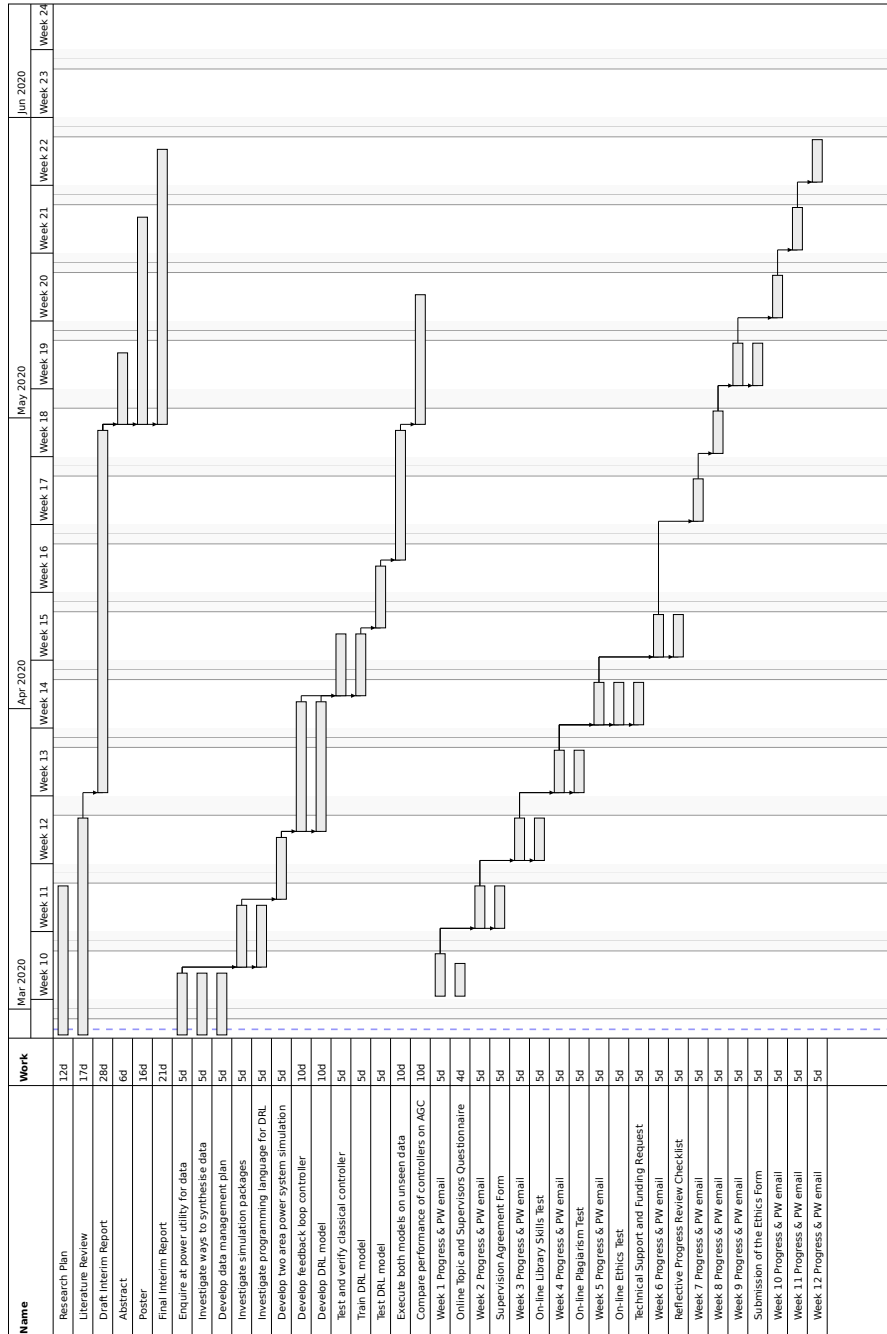8. Develop feedback loop controller for two area power system

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 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 present as 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 deliverable 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 [22]. It is envisaged that should the research meet expectations, this could provide a pathway for investigation of more exotic DRL models to find better power system controllers. The ultimate goal in this field 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 black event.

6 Timeline

The timeline is presented as a Gantt chart. The task list (Name / Work columns) is as follows:

Name	Work
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

The main research objectives, at a minimum, will require a computer with a Linux operating system, RAM, and a Graphics Processing Unit (GPU). 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 optimise the the DRL agent's cost function. 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. Most likely these resources will suffice, however, in the event that greater computational power is needed, access to a high end virtualised Amazon Web Services (AWS) system set up to conduct DRL research will be required. Currently, such an instance has been provisioned on an existing AWS account and is currently deactivated to avoid incurring cost. Using such a system can be expensive (1 hour of compute is approximately $20AUD), however, AWS will often supply student research with monetary credit.

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," Australian Energy Market Operator, Tech. Rep., Jul. 2012. [Online]. Available: `https : / / www . aemo . com . au/ - /media / archive / files / other / reports / frequency / 2011 / august-2011-freq-report.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] 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`.

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

[18] *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`.

[19] 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`.

[20] 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`.

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

[22] 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.