



THE UNIVERSITY OF
WAIKATO
Te Whare Wānanga o Waikato

Research Commons

<http://researchcommons.waikato.ac.nz/>

Research Commons at the University of Waikato

Copyright Statement:

The digital copy of this thesis is protected by the Copyright Act 1994 (New Zealand).

The thesis may be consulted by you, provided you comply with the provisions of the Act and the following conditions of use:

- Any use you make of these documents or images must be for research or private study purposes only, and you may not make them available to any other person.
- Authors control the copyright of their thesis. You will recognise the author's right to be identified as the author of the thesis, and due acknowledgement will be made to the author where appropriate.
- You will obtain the author's permission before publishing any material from the thesis.

**Grid Edge System Simulation and Evaluation Tool (GESSO):
Development of a Tool for the Modelling and Design of Distributed
Cooperating Microgrids**

A thesis

submitted **in fulfilment**

of the requirements for the degree

of

Master of Science in Computer Science

at

The University of Waikato

by

NATHAN BRYAN KELLY



THE UNIVERSITY OF
WAIKATO
Te Whare Wānanga o Waikato

2019

Abstract

As advancements are made in the availability of small renewable generation power storage technologies, nations are starting to see a trend towards more distributed power and generation. As times change, changes must also be made to the ways in which we consume and distribute power. Distributed microgrids represent the ability for communities to cooperate in the distribution of power within neighborhoods.

This thesis explores the structure of neighborhood scale distributed microgrids, defining the features and components needed to provide for accurate simulation. Techniques are developed for the integration of non-homogeneous microgrid systems to allow for smart grid-edge trading between units and microgrids. Exploration is conducted on the features required for an interactive system allowing the design and modelling of individual microgrid components and of neighborhood-scale microgrids, including the design of per-unit smart control schemes, and a proof of concept implementation is created allowing for simulation of non-homogeneous neighborhood scale fractal microgrids with arbitrary complexity. Sample cases are presented in order to demonstrate the effectiveness of the tool. The presented sample cases serve to demonstrate the effectiveness of small non-homogeneous microgrids, including those involving third party storage leasing services, and analysis is performed on the expected economic impacts of these types of systems.

Contents

1	Introduction	1
1.1	Definition of a Microgrid	1
1.2	Derived Microgrid Definition	2
1.3	Intentions	3
1.4	Summary of Thesis	3
2	Background	5
2.1	Technologies	6
2.2	Existing Tools and Techniques	8
2.2.1	Suricatta	10
2.3	Conclusion	10
3	Design	11
3.1	Techniques for Simulation and Control	11
3.2	Simulation of Solar Radiation	12
3.3	Defining behavior of individual units	15
3.3.1	Smart Usage Policies	16
3.3.2	Policies for Sale of Power	18
3.4	Simulation of single units	19
3.4.1	Simulation of Solar Generation	20
3.5	Defining Interactions Between Multiple Units Within a Single Microgrid	20
3.5.1	Features of Non-Composite Units	22
3.5.1.1	Consumption	22
3.5.1.2	Storage	23
3.5.1.3	Generation	23
3.5.2	Features of Composite Units	24
3.5.2.1	Consumption and Generation	24

3.5.2.2	Consumption and Storage	24
3.5.2.3	Generation and Storage	24
3.5.2.4	Consumption, Generation and Storage	25
3.6	Simulation of Interconnected Units	26
3.6.1	Descriptions of simulation of individual interconnected components	27
3.6.1.1	Simulation of Unit Coordinator	27
3.6.1.2	Simulation of Generation Controller	28
3.6.1.3	Simulation of Storage Controller	28
3.6.1.4	Simulation of Usage Controller	29
3.6.1.5	Simulation of Unit Comprising Usage and Generation	29
3.6.1.6	Simulation of Unit Comprising Storage and Generation	29
3.6.1.7	Simulation of Unit Comprising Usage and Storage	30
3.6.1.8	Simulation of Unit Comprising Usage, Storage and Generation	30
3.6.1.9	Simulation of Macrogrid	31
3.7	Technical limitations	32
4	Implementation and Application	33
4.1	Technologies Used	33
4.1.1	Sencha GXT + GWT	33
4.1.2	Apache Tomcat 9	34
4.1.3	Apache Derby	34
4.1.4	JBCrypt	34
4.2	Design of a Smart Usage Policy	34
4.3	Design and Evaluation of a Single Unit	37
4.3.1	Consumption	38
4.3.2	Generation	41

4.3.3	Storage	43
4.3.4	Simulation	44
4.3.4.1	Weather	45
4.3.4.2	Simulation Parameters	45
4.3.4.3	Simulation Results	47
4.4	Structure of a System	52
4.5	Calculations for Simulation of a Single Unit	53
4.6	Calculation for the Simulation of Solar Radiation	59
4.7	Calculations for Simulation of a System of Units	59
4.7.1	Implementation of Unit Coordinator	66
5	Results	74
5.1	Addition of Symbiotic Storage to a Microgrid	78
5.1.1	Analysis of a Simple Microgrid	79
5.1.2	Addition of Symbiotic Storage to a Simple Microgrid	81
5.1.3	Diminishing Returns with Symbiotic Storage in a Simple Microgrid	82
5.2	Impact of co-operation in homogeneous units	84
5.2.1	Experimental Results	85
6	Conclusions	90
	References	92
	Appendix A Single unit simulation graphs	95

List of Figures

1	NZ Solar Power Output Tool	9
2	Plot of usage requested from the grid with respect to power stored	18
3	Plot of generation offloaded to the grid with respect to power stored	19
4	Work flow for simulation of single grid connected unit	21
5	Textual Representation of a policy within UoW GREET	36
6	Adjusted usage policy	37
7	Policy sale of power graph	38
8	Adjusted usage policy	38
9	Power consumption design interface	39
10	Default energy distribution	40
11	Adjusted energy distribution	40
12	Power generation design interface	42
13	Power generation design summary	42
14	Interface for the configuration of unit storage.	44
15	Weather Selection	44
16	Single Unit Simulation	46
17	Generation/Consumption by Month	49
18	Grid Interaction by Month	49
19	Load Balance by Period	50
20	Grid Interaction by Month	50
21	Load balances for simple microgrid	80
22	Load balances for microgrid with symbiotic storage	82
23	Load balances for microgrid with symbiotic storage	85
24	Generation/Consumption by Day	96
25	Grid Interaction by Day	96
26	Battery State by Day	97
27	Generation/Consumption by Hour	97
28	Battery State by Hour	98

29	Grid Interaction by Hour	98
----	------------------------------------	----

List of Tables

1	Table of test units	75
2	Usage Policy X (Sell Aggressively)	77
3	Usage Policy Y (Neutral Strategy)	77
4	Usage Policy Z (Moderate Buyer)	77
5	Configurations for Testing Symbiotic Storage	78
6	Sale prices for Symbiotic Storage System	79
7	Simple Microgrid Annual Summary	79
8	Simple Microgrid Grid Impact	80
9	Simple Microgrid Economic Assessment	80
10	Symbiotic Storage Annual Summary	81
11	Symbiotic Storage Grid Impact	81
12	Symbiotic storage economic impact	82
13	More symbiotic storage annual summary	83
14	More symbiotic storage grid impact	83
15	More symbiotic storage economic impact	84
16	Generic homogeneous microgrid tests	85
17	Annual data for a large microgrid	86
18	Reduction in grid interaction for a large microgrid	86
19	Annual data for a medium sized microgrid	87
20	Reduction in grid interaction for a medium sized microgrid	88
21	Annual data for a small microgrid	88
22	Reduction in grid interaction for a small microgrid	89

1 Introduction

As advancements are made in the technology to harness the power of the sun and the ability to efficiently store power, New Zealand is starting to see a trend towards more distributed power and generation. Transpower NZ estimates that as power demand increases, renewable energy and storage alone wont be enough to efficiently meet the demands of most consumers[1]. Changes must be made to the ways in which we consume and distribute power. Distributed microgrids represent the ability for communities to cooperate in the distribution of power within a neighborhood. Through direct cooperation, neighborhoods may be able to reduce reliance on the state grid, while also allowing for equitable trade within the neighborhood.

This thesis explores the design and development of a software for designing and evaluating the structure of neighborhood scale distributed microgrids. The features and components needed to provide for the accurate simulation of distributed microgrids are outlined and defined, and techniques are developed for the integration of non-homogeneous microgrid systems to allow for the exploration of energy balance at the individual micro-grid level. Exploration is conducted on the features required for an interactive system allowing the design and modelling of neighborhood-scale microgrids, including the design of per-unit smart control schemes, and a proof of concept implementation is created allowing for simulation of non-homogeneous neighborhood scale microgrids. Further, the tool developed is intended to allow for interconnection of cooperating microgrids and the exploration of per-unit behaviours for grid edge trading.

1.1 Definition of a Microgrid

The Lawrence Berkeley National Laboratory (Berkeley Lab) defines a microgrid as a 'localized group of electricity sources and sinks (loads) that typically operates connected to and synchronous with the traditional centralized grid, but can

disconnect and maintain operation autonomously as physical and/or economic conditions dictate.’[2]

The *United States Department of Energy* (U.S. DoE) defines a microgrid as ‘a group of interconnected loads and distributed energy resources within clearly defined electrical boundaries that acts as a single controllable entity with respect to the grid’ [3], further stating the requirement that ‘A microgrid can connect and disconnect from the grid to enable it to operate in both grid-connected or island-mode’.[3]

Some of the advantages of microgrid systems include primary grid load shifting, additional points of redundancy in the case of grid failure, and typically the inclusion of renewable energy sources into standard power usage. Distribution of smart microgrids also allows for the reduction of energy loss over distance in powering homes and businesses when used as a primary energy source[4].

1.2 Derived Microgrid Definition

Based on the definitions given above, the concept of a microgrid can reasonably be applied to any scope. A standard house with energy generation and storage capacity, such as an array of solar panels and 25 kWh battery, is able to conform to both the provided definitions. The usage of the occupants and potential draw to fill the battery would serve as loads, the battery and solar panel can both be considered energy sources, and the house would typically operate connected to the grid. From the perspective of the grid, the house would be a single controllable entity. The storage and generation capacity supplies the ability for the house to disconnect from the state grid and operate at least semi-autonomously in ‘island-mode¹.’

A lower band can be given on the scope of a microgrid as ‘a single unit, consisting of at least one source of generation or storage, which can be disconnected from the main grid.’

The composition of microgrids can be considered again as a microgrid, provid-

¹weather permitting

ing that each of the composed microgrids shares a single controlled connection, from which the collection may be isolated or connected to the next-level macrogrid as need arises.

The combination of these rules allows for the construction of a new definition for a microgrid: "A localized collection of energy sources and sinks and/or interior microgrids, which share a single central connection to the next level macrogrid, which has the capacity of operating in island-mode." The use of this definition allows for the consideration of microgrids of arbitrary size and complexity, including renewable power substations, single houses, or even entire neighborhoods (so long as these neighborhoods meet the shared connection requirement) as individual microgrids.

1.3 Intentions

The intention of this project was to design and implement a tool which would facilitate further research projects within the field of smart microgrids. There are many subsets of research into renewable energy which deal with future technology and future concepts; for most of these concepts, there is a lack of real hard data and tools to be used in research. Within New Zealand, and many other parts of the world, the trade of power across residential property lines is currently illegal. This effectively limits any practical demonstration of many microgrid systems. In order for the electricity to shift further toward a cooperative prosumer market, there needs to exist adequate literature suggesting that this would be a positive direction for individual consumers and for power companies as a whole. Further research within this field may involve calculating optimal concentrations of generation and storage within neighborhoods, or optimal strategies for trading.

1.4 Summary of Thesis

If entire neighborhoods can become functional microgrid systems with workable integration into the state grid, or higher level microgrids, this allows for the abil-

ity create city infrastructure with high levels of sustainability, where each block nearly or almost nearly meets its own energy requirements (Net-ZEB), in an incremental fashion[4]. The integration of neighborhood scale microgrids into the general housing network would allow for neighborhood scale load shifting and grid outage resilience. The integration of multiple neighborhoods and smart microgrids into larger microgrids networks would allow for the sharing and equitable trade of run-off or over-generated power, storage capacity, and stored power between microgrids, meaning that individual networks with weak generation capacity may still be able to function within larger microgrids while the microgrid as a whole is disconnected from the state grid.

Within this thesis, a model is presented for the aggregation of multiple microgrids into larger microgrids (Chapter 1.2), an overview is given of the current technologies involved (Chapter 2.1), as well as an overview of several existing methods of simulation relating to microgrids (Chapter 2.2). A outline is given for the requirements and techniques needed to simulate a single microgrid component (Chapters 3.1 - 3.4). A classification of several different classes of microgrid components is presented (Chapter 3.5), and a methodology is presented for the design and simulation of neighborhood scale microgrid systems (Chapter 3.6). Furthermore, a sample implementation is given in which microgrids can be designed as individual unit components, and tested using real world usage and weather data (Chapter 4). Based on the sample implementation, test cases are generated and evaluated to demonstrate the effectiveness of certain types of mixed and homogeneous microgrids (Chapter 5), and some commentary is made on the implications of the results (Chapter 6).

2 Background

In Section 1.2, a definition for a microgrid was given as a 'localized collection of energy sources and sinks and/or interior microgrids, which share a single connection to the next-level macrogrid, and has the capacity to operate (at least temporarily) in Island-mode.' Considering the structure of a microgrid to be a collection of energy sources and energy sinks, immediate benefit can be seen on the conversion of small communities or farms into microgrid systems. Energy sinks within this system would be local demand (the power a house uses), local storage, and adjacent demand. Energy sources include generation through solar or wind power, local storage, or power offloaded by adjacent units. Additionally, the next level macrogrid can also be viewed as a source and sink of power when it is connected. A small community microgrid may enable the sharing of power from one source (Solar, Wind) to other consumers within the system when appropriate. A household generating more power than it can reasonably use or store may be able to offload some or all of that excess power to other agents within the neighborhood, potentially even at a price equitable for each of the agents involved. During power outages, some of these agents may be able to share stored and generated energy, potentially even meeting the demands of the entire microgrid during short or conveniently timed outages. Such a system could be considered an example of a smart microgrid system.

The aims of a smart microgrid system include a reduced grid dependency, increased robustness, reduced annual costs, and increased efficiency of distribution. As mentioned in [5], increased usage of smart microgrid technology may also allow for 'the grid and the electricity provider to plan their systems with a greater confidence.' As a greater number of remote communities and individuals, such as farmers and wine-growers, become equipped with the technology to become more self-sufficient, or in some cases, cooperatively sufficient, the burden on power companies to provide infrastructure on-par with centrally planned infrastructure may be reduced. As the electric infrastructure in New Zealand ages and

requires renewal and replacement, companies responsible for the management of infrastructure will need to evaluate the changing power needs as individuals shift away from centralized grid reliance. Though a 30 year plan has recently been produced by the New Zealand Nation Infrastructure Unit[6], changes in technology within the next few decades may requires significant re-evaluation of that plan. It may soon be more efficient or desirable for a power company to service the needs of a remote community by building a renewable energy power station, rather the connecting infrastructure to a larger populace. When such infrastructure is developed, it will be more optimal still that each member of the community is capable of contributing to the management of such infrastructure, or in some cases, comprising it entirely as a distributed neighborhood microgrid.

In essence, a distributed neighborhood microgrid consists of three qualities: the technologies of which it is composed, the techniques used in the management of the microgrid, and the structure in which the microgrid is created. Each of these features is described below.

2.1 Technologies

Photovoltaic solar panels are packaged collections of photovoltaic cells, designed for the conversion of solar radiation to electrical energy. Each market module is rated by its DC power output under "Standard Test Conditions (STC)," and typically ranges between 100-365 Watts[7][8]. Typically, a photovoltaic system may also include features such as an inverter, battery pack, supporting framework, and a solar tracking system. Conceptually, the state grid can be used as a low-efficiency arbitrary capacity power storage system - power is 'stored' in the grid by the sale of excess energy, and 'retrieved' via purchase. This is one feature that is sometimes used in the design of *Net-Zero Energy Balance* (Net-ZEB) and *Grid-Lite* systems[9].

Power storage is currently one of the most expensive components of an off-grid system. According to Sara Matasci, writing for Energy Sage, in January 2019 "the average national (United States) solar panel cost is \$3.05(USD)/Watt[10]." As of

July, the given cost is \$2.98/watt.[10] According to the United States Energy Information Administration (US EIA), the cost per kilowatt-hour of large-scale long-term battery storage is \$500 USD, while the cost per kilowatt of throughput capacity for large-scale long-term storage is \$2500 USD[11]. In this case, "long-term" refers to duration of two hours or greater. Though none of the devices the typical residential customer would use could be considered large-scale, this still provides an interesting metric for comparison. The Tesla Powerwall 2, being under the US EIA averages, costs \$5900 USD + installation fee[12]. The Powerwall 2 boasts 13.8 kWh of storage, and a 5 kW discharge rate. Using the US EIA numbers for large scale systems, the estimated price would be \$19400 - an almost four-fold increase in cost over the presented market price. EnergySage prices residential solar batteries at \$400 to \$750 USD per kWh of storage, ignoring the throughput as a factor of price[13]. Though the Tesla Powerwall series boasts an impressive storage capacity, it is worth considering just how much storage is practical and how much is needed. As mentioned in [5], batteries should primarily be considered short-term storage, providing power during night-hours, or during weather-related disruptions to supply. Batteries have little capacity to deal with changes in supply due to seasonal variation, and in many cases it is instead more cost effective to increase the the amount of power generated.

Using the prices for solar panels and for storage given by EnergySage, a 500W solar powered unit with a 3kWh battery would cost roughly \$2725 - \$3625 USD, with \$1525 of that cost being assigned to the solar generation, and the remainder being assigned to the storage[10][13]. Similarly, a 750W solar powered unit with 8kWh of storage would be costed between \$5487 - \$8287 USD, with \$2287 of the cost being assigned to the solar generation.

A solar installation will typically require the purchase of an inverter or series of micro-inverters, in order to transform the DC power generated by the solar panels into the AC power used by the host system. Some commercial battery systems, such as the Tesla Powerwall series, include the inverter needed to make use of solar power generation.

The throughput of energy from the solar panels to the battery, the solar panels to the house, or the battery to the house, will be restricted based on the capacity of the lines and the capacity of the installed inverter, as well as the rated throughput of the battery. In a 750W solar system, it would not be abnormal to see generation exceeding 1kW in good weather. If the supplied inverters are only rated for a throughput of 750W, then any excess power would be wasted. If the battery is only capable of charging at a rate of 500W, then there would be additional wastage between the solar panels and the battery. Ideally, this extra wastage would be used to immediately counter the load of the system, or sold to the state grid. These options may not always be available.

2.2 Existing Tools and Techniques

Several tools exist for the simulation of solar generation. Auckland University provides the 'Auckland rooftop solar potential' online tool to 'give an economic estimate of the solar panel potential for most Auckland homes'[14]. The tool makes use of LiDAR data provided by the Auckland council to create digital topography models for Auckland city, using this data to estimate the slope, surface area, shading and azimuth of roofed areas at a single meter resolution. Using this data, the tool makes a calculation for the economic potential of a particular household.

NIWA provides the tool 'SolarView' to calculate the amount of sunlight a house will receive over the course of a year[15]. Solarview makes use of topographical data and NIWA's own weather station data to give month-by-month estimations of the amount of solar energy that will strike a rooftop.

NZ SPOT (New Zealand Solar Panel Output Tool) is an online tool which allows simulates solar panel output at different locations and orientations[16]. The tool takes into account real radiation, dry-bulb temperature and wind speed values from the New Zealand national climate database. A user is able to select a quantity of panels arranged in series and/or parallel, an inverter, optional azimuth and slope tracking, and optimism about the *clearness* of the region in order to

receive estimates for the solar generation profile of the average day for each month. An example of the output given by NZSPOT can be seen in Figure 1.

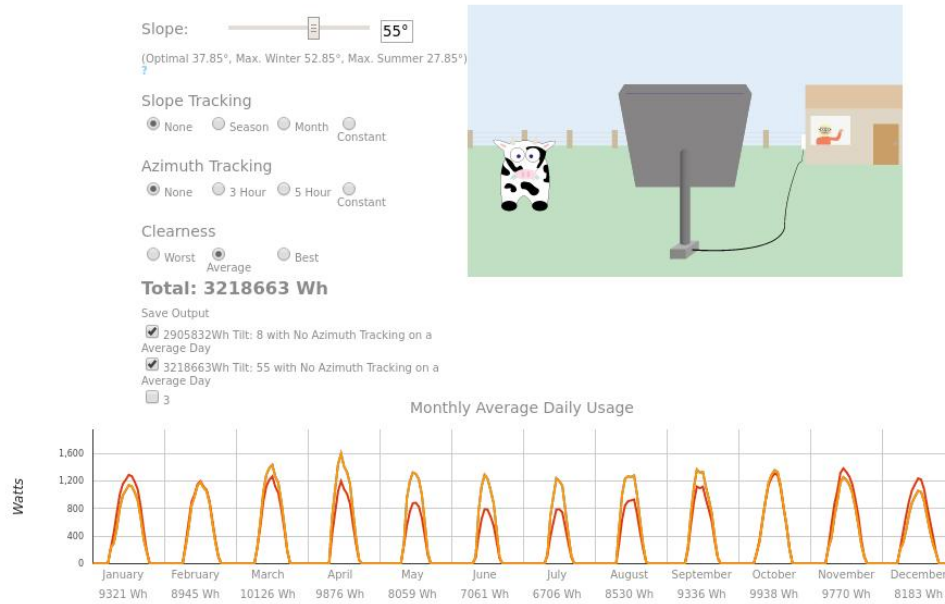


Figure 1: Output from the NZ SPOT tool.

The calculations within the NZ SPOT tool are based on those provided by the Sandia National Laboratories PV Performance Modelling Collaborative (PVPVC for short)[17]. The Sandia Labs PVPVC is a group focused on 'improving the accuracy and technical rigor of PV performance models and analyses'. To this end, the PVPVC provides a host of useful calculations for accurately simulating PV performance based on a range of factors. In general, the steps are as follows:

1. Weather and Design
2. Array Characteristics
3. DC to AC conversion
4. AC System Output

The Sandia Labs PVPVC also provides several useful tools to assist in the simulation of PV and grid performance, such as the PVLib Python and Matlab frameworks. A full description and listing of the calculations used is given in Section 3.2.

2.2.1 Suricatta

Suricatta is a platform produced to enable generation of load profiles for households to be modelled based on various input parameters (such as house occupancy, appliance ownership, and activity patterns)[18]. Generation is performed on a single-day basis, and take as input base voltage and base apparent power, as well as location and time of year. It allows for the generation of granular data (including on a minute-by-minute basis) using acceptance-rejection sampling.

Suricatta provides an interesting platform for the generation and modelling of power demand for small networks at the individual appliance level, and has the potential to allow for the creation of a tool which does not rely on pre-recorded real-world data for household consumption. Although it was not used in the iteration of the tool presented within this thesis, it presents a potential avenue for improvement in future dates.

2.3 Conclusion

With reference to a microgrid defined as 'localized collection of energy sources and sinks and/or interior microgrids, which share a single connection to the next-level macrogrid, and has the capacity to operate (at least temporarily) in Island-mode,' an overview has been given of several existing publications and features. In particular, the NZSPOT tool is able to simulate single solar panels, making use of weather data from NIWA and equations from Sandia Labs[16]. The Suricatta tool is able to generate load profiles for dwellings based on given input data[18]. The Auckland rooftop solar potential tool is able to make estimations for solar potential based on apparent roof size and angle in satellite photography[14] - a similar function to the NIWA SolarView tool, which uses landscape pictures and location to estimate the amount of solar energy a panel will collect over the course of the year[15]. Alone, each of these tools provides a useful function - but none of these tools alone provides enough information to properly assess a microgrid.

3 Design

The UoW GESSO tool was intended to be a tool allowing for the design, simulation, and evaluation of arbitrary microgrids. Design of microgrids should allow for non-homogeneous simulations, in which different types of microgrid components can interact together. The simulation of microgrids should make use of some of the tools or concepts discussed in Section 2.2. In particular, the ability to accurately simulate solar generation as shown in the NZ SPOT and Sandia Labs tools provides a good basis for all of the non-economic simulations within the designed microgrids. Evaluation of the microgrids should be able to take two forms: raw data, to be processed by other programs, and graphs and figures generated by the UoW GESSO tool itself.

In order to simulate a microgrid system, as defined in section 1.2, a series of conditions must be fulfilled:

1. The techniques to be used must be defined
2. All the components of the system must be defined
3. The behavior of each component must be defined
4. The behavior of the 'world' must be defined

Additionally, once the behavior is formed for all units, it is convenient for implementation that the system can be represented as a mathematical model or workflow.

3.1 Techniques for Simulation and Control

An important part of performing any simulation is to define the techniques and strategies that will be used. For the creation of the UoW GESSO tool, the following features were decided on:

- Input data (Usage, Components, Weather) is to be user selected, and static throughout multiple simulations

- Simulation would take place in discrete hourly intervals²
- The input for the simulation of any hourly interval would be the components and the outcome of the previous interval
- For the initial interval, default values are given as a 'simulated outcome' with respect to the features of the components used
- Optional features of the simulation may introduce randomness³

The ability to introduce randomness as a feature was decided upon as a way to measure or verify the robustness of a user designed system. One of the potential uses for for microgrid systems, or localized energy storage in general, is to reduce downtime losses due to conditions outside of the control of the network:

3.2 Simulation of Solar Radiation

The following series of calculations was used to provide solar power generation that scaled appropriately with time of day and relative angles of the solar array and sun in the sky. All of the equations are either taken directly from the Sandia Labs PV Performance group, or derived from the equations provided by the Sandia Labs PV Performance group, as mentioned in Section 2.2[17].

1. Local Standard Time Meridian (*LSTM*)

The Local Standard Time Meridian is the reference time meridian used. For New Zealand, which uses Greenwich Meridian Time(*GMT*) + 12, the *LSTM* is 180°.

$$LSTM = 15 \cdot timeZone; \tag{1}$$

2. Equation of Time ((EqT))

The Equation of Time is the expected difference between the solar time as shown by clocks and the solar time as shown by sundials, with reference to the local of the observer and the time of year. the The equation of time can

²This matches the resolution of the weather data used.

³Simulated power outages are an example.

be approximated with a series of equations based on the time of year.

$$EqT = \begin{cases} day < 107, & -14.2\sin(\pi\frac{day+7}{111}) \\ day < 167, & 4.0\sin(\pi\frac{day-106}{59}) \\ day < 246, & -6.5\sin(\pi\frac{day-166}{80}) \\ day < 367, & 16.4\sin(\pi\frac{day-247}{113}) \end{cases} \quad (2)$$

3. Time Correction Factor (*TC*)

The Time Correction factor is an accounting of the variation from solar time at a certain time zone an solar time at a certain meridian.

$$TC = (LSTM - LONG)/15 \quad (3)$$

Where *LONG* is the longitude of the observer.

4. Solar Time (*TS*)

Solar Time is the observed position of the sun in the sky from a given observation point. The solar time can be given as a synthesis of the Local Time, Equation of Time and Time Correction factor. The values *TS* and *T_{local}* are expressed in terms of hours.

$$TS = T_{local} + \frac{EqT}{60} + TC \quad (4)$$

5. Declination Angle of the Sun

The earth fluctuates on it's axis between -23.45° and +23.45° with relation to the sun. The plane normal to the geographic poles of the earth consequently fluctuates in relation to the sun. This is the declination angle of the sun in relation to the earth (or of the earth in relation to the sun). This can be approximated with the following equation:

$$\theta_d = \frac{23.45\pi}{180} \sin(2\pi\frac{284 + day}{365}) \quad (5)$$

6. Hour Angle of the Sun

The hour angle of the sun is the observed angle of the sun in the sky throughout the day. By convention, solar noon is defined as 0° , with the angle increasing past noon. Since the earth spins roughly 15° an hour, this can merely be approximated as a function of the solar time.

$$\theta_{hr} = \pi \frac{12 - T_{solar}}{12} \quad (6)$$

7. Elevation Angle (EA)

The elevation is the angular height of the sun with respect to the horizon of an observer. At sunrise and sunset, the angle is 0° . During the summer, the peak elevation angle is higher than it is in winter.

$$EA = \sin^{-1}[\sin(\theta_d)\sin(LAT) + \cos(\theta_d)\cos(LAT)\cos(\theta_{hr})] \quad (7)$$

where LAT is the latitude of the observer.

8. Zenith Angle (ZA)

The zenith angle is the angular height of the sun in the sky with respect to the zenith of the sky, rather than the horizon. The calculation given is merely a transformation on the elevation angle.

$$ZA = 90 - EA \quad (8)$$

9. Azimuth Angle (AA)

The azimuth angle of the sun is the compass direction of the sun in the sky. This is calculated from the declination of the sun, the latitude, and the hour angle.

$$AA = \cos^{-1} \frac{\sin(\theta_d)\cos(LAT) - \cos(\theta_d)\sin(LAT)\cos(\theta_{hr})}{\cos(EA)} \quad (9)$$

10. Angle of Incidence

The information generated so far is enough to provide the angle of incidence between a given solar array and the sun at a given time of year and location.

The angle of incidence is the relative angle between the mean solar array surface and the sun in the sky.

$$AOI = \cos^{-1}[\cos(ZA)\cos(PT) + \sin(ZA)\sin(PT)\cos(AA - PA)] \quad (10)$$

where PT is the mean tilt of the solar array, and PA is the mean azimuth of the solar array.

Once the angle of incidence has been calculated, the real data taken local weather stations can be used to generate estimates of the amount of radiation a given solar panel would receive given orientation and time of day. It is suggested that the received solar radiation by a solar array is estimated as a proportion of the plane of array beam and ambient solar radiation using the following equation.

$$mJ = \frac{9 \cdot \text{MAX}(\text{RAD} \cdot \cos(AOI), 0)}{10} + \frac{\text{RAD}}{4} \quad (11)$$

where mJ is the observed radiation in mega-joules per meter squared, and RAD is the radiation value taken from the local weather station, in mega-joules per meter squared.

For the purposes of this thesis, generation received by solar panels is assumed to correlate linearly with radiation received, using the power generated at standard test conditions as a baseline value.

3.3 Defining behavior of individual units

Behavior is defined for individual grid-connected units on a case-by-case basis. Units that do not have storage have no autonomy over their actions, so all excess generation must be dumped and all demand must be supplied, otherwise wastage or shortfall occur. There are no recovery mechanisms for these units.

Addition of storage allows for autonomy of action. The features of storage are total capacity, charge rate, discharge rate, and available power. A typical household unit, having storage and usage, must make a synthesis between these

qualities and the power consumption to make decisions about how much power to draw from the grid. The addition of generation allows for the composite value of *net consumption*, which may be negative in the case that generation exceeds consumption. A unit having generation, storage, and consumption must then make choices not only on how much power to draw from the grid, but also how much power to offload to the grid.

Although potentially practical as a component of a microgrid, a unit comprising only storage would be unpractical in use as a lone grid-connected agent. The price of sale to the grid will likely never exceed the price of purchase from the grid - in reality, prices may be presented as roughly 29-33¢ per kWh for the purchase of power, and 10¢ per kWh for the buy-back of power. These prices are taken from mercury energy NZ - August 8th, 2019. Purchase price excludes GST and line charges, and sale price is subject to taxation and additional lines levies.

Scenarios were designed for the simulation of single-unit grid connected systems. These scenarios were primarily intended to act as a precursor to full neighborhood simulation, and hence only tackled the most complicated type of unit: one comprising of storage, generation, and usage. The simulation scenario allowed for the use of user-defined reactive control schemes for management of stored power, as well as management of financial transactions between the unit and the grid. The scenarios also allowed for the simulation of power outages and restriction of grid throughput capabilities, testing robustness of the system in the event of partial or total grid power failure.

3.3.1 Smart Usage Policies

A system for managing user preference in power management was designed for units which had both storage and consumption. A series of "policies" are defined by the user. Each policy takes into account the current battery state and power consumption, and give as an output a target grid draw. In a real-world system, the execution and management of policies would likely be operated by a device stored within or connected to the storage component of the unit.

A completely dumb system might define only a single policy, such that:

- If Battery State $\geq 00\%$, then Target Draw = Usage * 1.0;

This policy would always attempt to draw as much power from the grid as is used. The calculations take into account battery efficiency behind the scenes, meaning that the phrase "Target Draw = Usage * 1.0;" really means "Target Draw = Usage * 1.0 / Efficiency", where the value Efficiency is a number between 0 and 1, typically 0.87 to 0.97.

The above system would attempt to always maintain battery state at a constant level by drawing exactly as much power as is used. A system like this might have some survival potential in the case of a power outage, but no provisions are made for the replenishment of power between power outages. One potential feature to consider, which was not included in the produced prototype, might be a memory of usage over the course of a power outage, and the attempt to smoothly recover battery capacity afterwards.

A user hoping to maintain a somewhat smarter system might instead have a set of policies like the following⁴:

- If Battery State $\geq 00\%$, then Target Draw = Usage * 3.0
- If Battery State $\geq 25\%$, then Target Draw = Usage * 1.5
- If Battery State $\geq 75\%$, then Target Draw = Usage * 0.8
- If Battery State $\geq 90\%$, then Target Draw = Usage * 0.3

This set of policies would attempt to maintain battery state between 75% and 90% during regular usage, aggressively demanding power when the battery is low - this may serve to aid recovery during situations where there is spurious power availability, quickly replenishing power between micro-outages. In essence, this policy system can act as a load-shifting device under conditions where power availability may be spurious.

⁴Note that rules are evaluated selecting for **most significant match** - In the example presented, if the battery is 80% full then the third rule would be evaluated.

3.3.2 Policies for Sale of Power

The policy control model presented above can easily be extended to account for units that produce as well as consume power. First, considerations must be made for the *net generation/net usage* values obtained through the composition of generation and consumption. A second clause is added to the right-hand side of individual policies, accounting for excesses in generation of power. Selection is made between the two clauses based on whether the *net generation* or *net usage* values are greater than 0. In the case where both values are 0, no action is taken.

An example policy set is presented:

- Battery \geq 00%, then Target Draw = Usage * 3.0, Offload = Gen * 0.0;
- Battery \geq 25%, then Target Draw = Usage * 1.5, Offload = Gen * 0.5;
- Battery \geq 75%, then Target Draw = Usage * 0.8, Offload = Gen * 1.5;
- Battery \geq 90%, then Target Draw = Usage * 0.3, Offload = Gen * 2.5;

Plotting the thresholds and values provided in the policy provides for an easier to understand picture of what the policy does. Figure 2 shows the plotted draw of both policies, in relation to the power consumed and capacity of storage used. Figure 3 shows the plotted offload of the above policy, in relation to the power generated and the capacity of storage used.

Note that the values *Usage* and *Gen* refer to the raw generation and usage,

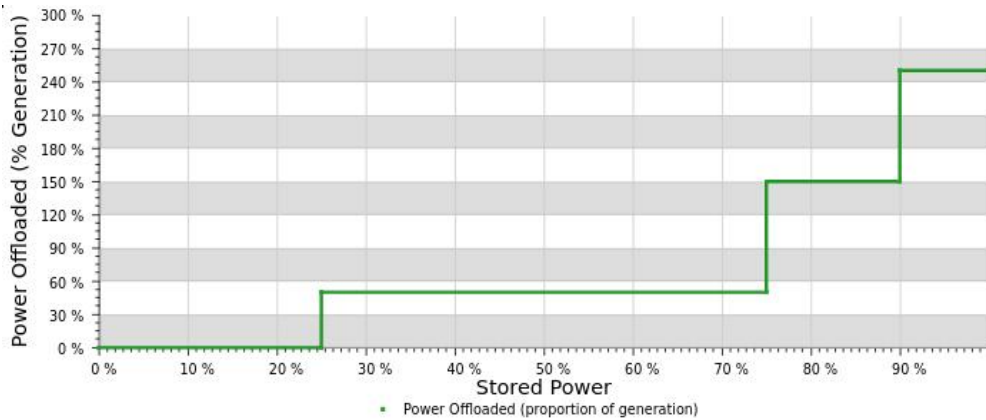


Figure 2: Plot of usage requested from the grid with respect to power stored

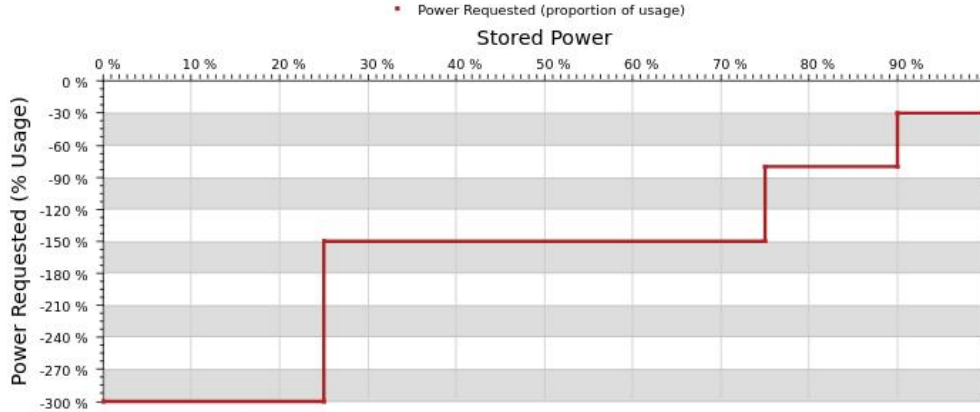


Figure 3: Plot of generation offloaded to the grid with respect to power stored

whereas the selection of the clauses is determined by the *Net Usage/Net Generation*. When generation exceeds usage, and hence the *net generation* is positive, the 'offload' branch is executed.

When a unit using the above policies generates more power than can be reasonably stored, it tries to clear space and offload excess power. When that same unit is low on power, it will either generate higher than it uses (eliminating the need to draw), or it will aggressively request power from the attached macro-grid, if able. In this way, the unit attempts to maintain storage at 70-80% capacity. This allows for emergency power and uninterrupted power through power blackouts, or in situations where the grid connection is spurious.

3.4 Simulation of single units

The simulation of single units (in this case, units that consume and store power, and optionally generate power) requires two features:

- **The inputs are defined.**

This includes the features of the unit (consumption, generation, storage) and the features of the environment (location, weather data, grid capacity, expected outage duration and frequency).

- **The behavior of the unit is defined.**

The unit has a set of policies defining how it acts under typical circum-

stances.

Simulation of the system takes place in single-hour increments, using the diagram presented in Figure 4 as a template for implementation. A starting battery state is defined (80% effective capacity), and then the flow chart can be followed almost exactly. For each period of observation, which in the case of this simulation is one hour, the generated output can be used as the input for the next observable period. Worth noting is that Section B is only evaluated if the policy control option is enabled: otherwise, the simulation only requires execution of Section A.

3.4.1 Simulation of Solar Generation

One important feature of the simulation is the accurate modelling of solar power generation.

The procedure for simulation of power is to take the solar radiation data (GHI) for the current period from the weather data file, and to apply the transform as is shown following step 10 in Section 3.2, scaling the value by the number of solar panels and the area of the solar array.

Specifically, if radiation is generated with the unit mJ/m^2 , or milliJoules per meter-squared, it needs to be multiplied by the surface area of the solar array ($panel_area \cdot panel_quantity$), and the standard generation of the solar panel. Each of these quantities can be obtained from the solar panel spec sheet.

3.5 Defining Interactions Between Multiple Units Within a Single Microgrid

Within the simulation, each unit in the microgrid is made up of three primary features.

1. The capacity to consume power
2. The capacity to generate power
3. The capacity to store power

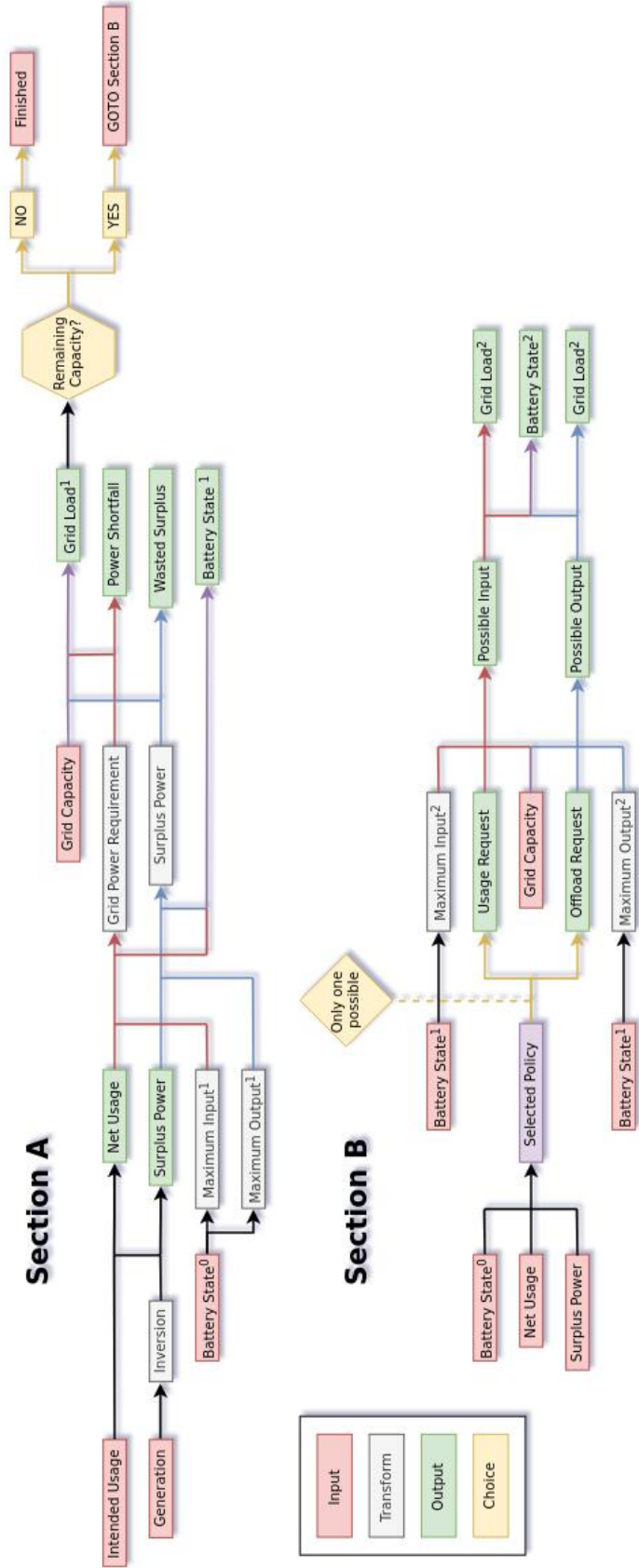


Figure 4: Work flow for simulation of single grid connected unit

Any (or all) of these features may be 0. In the case that all of the features are 0, then the unit can have no interaction with the grid, and can be ignored. In all other cases, behavior must be defined.

3.5.1 Features of Non-Composite Units

Non-composite units are units which only have one of the three feature: generation, consumption, storage. Although these units have lower levels of agency than units with multiple features, the application of units containing only storage or only generation within a microgrid provides clear benefits over the same type of unit being attached to the state grid.

3.5.1.1 Consumption

A unit which only consumes power, and has no capacity to generate or store it, is almost entirely without agency in how it chooses to consume power. While the grid is able to supply the power needed, these units must draw power from the grid to meet demand. Limits to the amount of power this type of unit is able to draw from the grid at any one time may be set. For the standard household in New Zealand, this would be limited directly by what the circuit breakers are able to take, rather than the grid itself. The limit may be artificially low, or lowered due to the demands of the location, hardware or infrastructure, such as in a remote community or research station.

When consumption exceeds supply, then the unit experiences a *shortfall* of power. In many scenarios, minor shortfall may only result in a dimming of the lights, or computer and automation equipment being slightly under-volted - with the deficit being spread proportionally among components of the unit. Alternatively, severe limitations to the intended usage may cause a forced shifting of load. Not having the capacity to perform some energy intensive task, such as running a washing machine, that task must instead be performed later.

Without data which relates to the usage of specific appliances, reactive load-shifting isn't an applicable method that can be applied during simulation. A

model could potentially make use of frameworks such as Suricatta to produce demand based models, to which load shifting could be applied - in the case of this thesis, this has not been done.

3.5.1.2 Storage

A unit which only has the capability for storage of power has no option for making a profit in a regular grid-tied system. The price of selling power to the grid will never (currently) exceed the price of purchasing power from the grid.

Within a smart microgrid system, units capable only of generation are capable of performing work for the system by effectively leasing storage capacity. A unit may declare that it is buying power at \$0.10/kWh, and selling power at \$0.18/kWh. During times of excessive generation, such as the middle of a sunny day, the unit may provide a more attractive sale candidate than the state grid or other units within the microgrid. Then, during the course of the evening and night when solar generation stops being sufficient to power the other units within the microgrid, the unit would provide a more attractive purchasing candidate than the state grid, or other units within the system. Given a 97% battery efficiency, and an expected total loss through transmission of 5%, the profit margin for this unit would be \$0.065/kWh. Though not a particularly high number, it presents a method for a unit within the microgrid to generate money while providing a useful service.

3.5.1.3 Generation

A unit which only generates power and is grid-tied only has one available option, which is to sell off power at the price that the grid defines (or otherwise waste it). Within a neighborhood scale microgrid, the unit may instead attempt to define prices of sale, and offer power to other units at that price. Furthermore, if no units are willing to buy at that price, it may attempt to match prices with the highest priced buyer available rather than wasting the generated power.

3.5.2 Features of Composite Units

Composite units are units which have more than one of the three primary features: Consumption, Generation, Storage. The composition of features generally allows for the unit to express a high degree of agency within a microgrid.

3.5.2.1 Consumption and Generation

A unit which only consumes and generates power still has very limited agency within the system. Any surplus generation must be sold or wasted. Alternatively, surplus power could be used with smart home automation and load shifting. Surplus power could even be used for cryptocurrency mining⁵.

3.5.2.2 Consumption and Storage

A unit containing both consumption and storage will still primarily rely on the local or state grid to source its energy demands. The inclusion of storage allows for the unit to engage in opportunistic trading with other members of the microgrid. The unit will have the ability to buy power at low prices in excess of what is needed, to define prices above which power will not be purchased, and to maintain a stockpile of power in order to last through periods of restricted supply.

A threshold can be set, being a price above which power will not be purchased. These units may make use of the 'smart usage policies', as mentioned in Section 3.3.1, to maintain a stockpile of power within the battery. Power will always be consumed directly from the grid or microgrid, if possible, to minimize energy losses due to conversion.

3.5.2.3 Generation and Storage

A unit comprising storage and generation can afford to be more flexible with how power and trade are managed. There is no need for the unit to rely on purchasing

⁵Japanese power company Kumamoto-Energy uses excess generation to power an affiliate cryptocurrency mining business. See [19].

power in order to sell it, but the purchase of power at low prices can still provide an economic benefit by allowing it's sale at a later time.

A threshold can be set, being some proportion of the battery state, above which power will not be purchased. Then, prices can be set for the purchase and sale of power. Further, there is a potential for this type of unit to end up with *too much* stored power - for example, during a particularly sunny day, if every other unit in the neighborhood is drawing from cheap solar power. In these cases, it may be beneficial to allow for price matching of generated power when above a certain threshold, for example 90% battery capacity. If there is truly nobody willing to buy the power, and there is no longer capacity for storage, then it must either be given away or wasted.

3.5.2.4 Consumption, Generation and Storage

A unit composed of Consumption, Generation and Storage has the most agency over how it manages power. In bad weather, the unit may rely on its stockpile of stored power, or locally distribution, in order to match the energy consumed.

The unit is able to engage in opportunistic trading within the neighborhood: if an abundance of power is being generated, it may be sold within the neighborhood or stored locally as conditions demand. Power will also be purchased from the grid or local distributors to assist in meeting demand when the quantity of stored power is low.

A price can be set, above which power will not be purchased, and a price can be set at which power will be sold. Additionally, the unit will make use of 'smart usage policies', as mentioned in Section 3.3.1, in order to manage the purchase and sale of power. In this way, the programmer or owner of the unit can configure it for a range of behaviors. For example, the unit may:

1. Buy and sell power only at opportunistic prices, and rely on generation otherwise
2. Buy power only when the need is desperate, but otherwise sell power at opportunistic prices

3. Always try to maintain $X\%$ battery capacity, and otherwise buy and sell power so long as the prices are reasonable
4. Some combination of the above

3.6 Simulation of Interconnected Units

In order for the simulation of the system to work, there must be some form of co-coordinator: a prime mover, through which all other actions are permitted to take place. In a real system, this may take the place of a single controlling agent which acts as a "transaction authority" for the system. Alternatively, it may take the place of a distributed computer, verifying all transactions using block-chain technology. For the purposes of this simulation, a single "prime mover" type agent was used.

The agent has the role of facilitating transactions, requests, offers and power transfer between units within the microgrid, or between the microgrid and the state grid. For each component within the system, the unit coordinator must correctly balance transactions in a way that does not favor individuals any further than economic circumstances already permit: ensuring that if multiple units are each willing to pay the same price for quantities of power, that none of the units receives any special preference over the other, and likewise for the sale of power. This raises an interesting philosophical question about fairness and equality:

Alice requires 4 kWh of power

Bobby offers to sell 9 kWh of power

Cindy offers to sell 3 kWh of power

Bobby and Cindy are selling power at the same price

In which proportions should the power be sold? If power is sold equally, then Bobby and Cindy both sell 2 kWh of power each. Assuming no other power is sold or used, then Bobby has wasted 7 kWh of power and Cindy was wasted 1 kWh of power. Should Bobby, who was invested more into the supply of the neighborhood, receive less of a cut than smaller suppliers?

If power is sold strictly proportionally, then Cindy would sell 1 kWh of power, and Bobby would sell 3 kWh of power. At first glance this may seem to be more 'fair,' but should Cindy be penalized on her ability to sell power because of the raw yield of Bobby, despite having the same prices? There is clearly no easy answer to this question. One may attempt to use some combination of the two metrics, and the choice of or balance between these metrics may be left to implementation.

3.6.1 Descriptions of simulation of individual interconnected components

For the purposes of this thesis, power will always be sold equally when prices match. Likewise, when multiple agents want to buy power at the same price, it will always be sold in equal portions between purchasing units.

For each type of unit that will be represented within the system, the unit coordinator itself, and the grid itself, methods are developed for the simulation of that unit.

3.6.1.1 Simulation of Unit Coordinator

The unit coordinator keeps track of all of the transactions within the microgrid. The coordinator directly negotiates with the state grid (or next level or grid, when applicable), and keeps track of the price of trade between the local microgrid and the next level of grid. For the purposes of this paper, prices for offers and requests are with respect to the coordinator price, rather than given as discrete prices.

1. Start with summation of all offers over the period, sorted and *binned*⁶ by price. Include in this the grid itself as a source.
2. Receive all requests for power from the network.
3. Take the lowest price source on the network. Divide it equally between every requesting unit, capped to the maximum input that unit can take. Units that reach their request are removed, and this process is repeated so long as there is leftover power.

⁶equal price = equal priority

4. The billing information of each unit is updated to reflect what was taken or given.
5. If there are still units remaining that have not met their usage threshold, switch to the next-lowest binned source, and repeat the above step.
6. If every unit has exhausted supply, and there is still demand, then there is a shortfall.
7. If every unit has exhausted demand, but there is still supply, then the supplying units will first try to:
 - (a) Store excess power in battery, if possible.
 - (b) If there is still excess power, offer it for free to any unit on the network that will take it.
 - (c) If there are no units that will take it, and there is nowhere to store it, then it is wasted. ⁷.

3.6.1.2 Simulation of Generation Controller

A strategy is given for units comprising only generation:

1. All generated power is offered at a factor of the coordinator price
2. Power is sold while demand exists, and if no demand exists, is offered freely
3. Any remaining power is wasted

3.6.1.3 Simulation of Storage Controller

A strategy is given for units comprising only storage:

1. A purchase price is defined for power, below which purchases are made
2. A sale price is defined for power, at which sales are made
3. Power is offered to the coordinator at the sale price, so long as the battery is above a certain charge threshold

⁷A real unit might use alternate means to store or use the power; such as super-heating water tanks, or mining crypto currencies

4. Purchase requests are made to the coordinator under the sale price, so long as the battery is under a certain charge threshold
5. Will accept freely given power when possible
6. Optionally, thresholds are defined constraining the trade of power - power is only purchased when below a certain threshold, and only sold when above a certain threshold ⁸

3.6.1.4 Simulation of Usage Controller

A strategy is given for units comprising only usage:

1. A unit having only usage has no choice but to buy power to meet demand in real time
2. A unit having only usage may set a maximum purchase price for power, above which power will not be purchased

3.6.1.5 Simulation of Unit Comprising Usage and Generation

A strategy is given for units comprising usage and generation:

1. Generation is first used to complement usage, and then the net surplus or deficit of power must be acquired or disposed with
2. A unit having only usage and generation may set a maximum purchase price for power
3. A unit having only usage and generation may set a standard sale price as a factor of the coordinator price
4. Optionally, a maximum quantity for power offloading can be defined
5. Optionally, a maximum grid load can be defined

3.6.1.6 Simulation of Unit Comprising Storage and Generation

A strategy is given for units comprising storage and generation:

⁸Example: buy when power is below 40% and the price is below 0.4 standard - offer when power is above 20% at 0.8 standard

1. A price is defined for the sale of power
2. Generation is first sold
3. Generation that cannot be sold is stored
4. Generation that cannot be stored is given away
5. A threshold is defined above which stored power will be sold at the defined price
6. A price is defined for the purchase of power
7. A threshold is defined below which power will be purchased at or beneath the defined purchase price

3.6.1.7 Simulation of Unit Comprising Usage and Storage

A strategy is given for units comprising usage and storage:

1. A maximum price is defined for the purchase of power
2. Power will not be purchased above this price
3. A policy is defined (Reactive) which is capable of reading the battery state, reading the system usage, and using a stabilizing algorithm to request the purchase of power
4. A policy is defined (Restrictive) which is capable of reading the time of day, and filters the output of the reactive policy, tailoring the requests to specified maximums.
5. The reactive policy defines the threshold below which is it preferable to use power from the network rather than stored power
6. Optionally, Restrictive policies can be seasonally defined
7. If possible, power will be purchased to match the request given by the current policy set

3.6.1.8 Simulation of Unit Comprising Usage, Storage and Generation

A strategy is given for units comprising usage, storage and generation:

1. A maximum price is defined for the purchase of power

2. A minimum price is defined for the sale of power
3. A policy is defined (Reactive) which is capable of reading the battery state, reading the *net* system usage, and using a stabilizing algorithm to request the purchase of power or offer up power for sale
4. A policy is defined (Restrictive) which is capable of reading the time of day, and filters the output of the reactive policy, tailoring the requests and offers to specified maximums.
5. The reactive policy defines a threshold below which it is preferable to use power from the network rather than stored power
6. The reactive policy defines a threshold above which it is preferable to sell power to the network rather than storing it
7. Optionally, Restrictive policies can be seasonally defined
8. If possible, power will be purchased to match the request given by the current policy set
9. If possible, power will be sold to match the offer given by the current policy set
10. Power that can be neither sold or stored will be given away freely

3.6.1.9 Simulation of Macrogrid

Having gone through all the strategies below, there still remains the issue of creating a macrogrid interface. As an option, the state grid or macrogrid may be considered a component of the microgrid itself, in which case the unit coordinator can be simplified further.

1. The macrogrid is conceptually treated as a single unit within the microgrid.
2. The macrogrid is given a fixed (or dynamic) amount supply for each period.
3. The macrogrid is given a fixed (or dynamic) amount of demand for each period.
4. A price is defined for the sale of power from the macrogrid
5. A price is defined for the purchase of power by the macrogrid

6. The macrogrid may both purchase and sell power at the same time. ⁹
7. If power is being offered freely, and if the macrogrid has remaining demand, it may instead be sold to the macrogrid at the macrogrid defined price (if possible)
8. Any supply or demand the grid has at the end of the period is ignored
9. During an outage, the macrogrid cannot interact with any other units¹⁰

3.7 Technical limitations

There are many differences between the simulation and operation of a neighborhood scale microgrid. These are all features which must be addressed, or simplified out of the equation, in the design of the simulation software.

All power transmitted must be transmitted in roughly real-time. Units may make use of small batteries, flywheels, or short-term storage to turn the real time demand into discrete windows of demand - for example, operating in five minute windows of time.

Transmission of power across a fractal network provides for uncertain voltage loss between consumers. If the sale of power between Bobby and Alice must be routed through Cindy, or perhaps through a handful of other units also, then significant losses may occur in transmission. Additionally, if many units need to route power over the same set of lines, the aggregated load may be higher than those lines can handle, despite all of the units using relatively small amounts of power. The real-world design of any distributed microgrid system would then have to account for potential power bottlenecks.

⁹This is effectively load balancing within the microgrid

¹⁰Conceptually, both supply and demand become 0

4 Implementation and Application

The implementation of the smart microgrid simulation tool, UoW GESSO, takes the form of a dynamic web application. The web application allows for a user to:

- Create an account and log in
- Design individual units, and save those designs for later use
- Design policies for units, and save those designs for later use
- Design systems of units, using the saves units and policies
- Evaluate the operation of individual units under various conditions
- Evaluate the operation of systems of units under various conditions
- Collect the raw data from system-wide evaluations

4.1 Technologies Used

A handful of technologies were used in the construction of the prototype. The most relevant of these are what was used to host the web applet, what was used to design the web app, what was used to secure the web app, and what was used to manage data within the web app.

4.1.1 Sencha GXT + GWT

The Sencha GXT framework was used for the design of the web application. GXT is a web-application framework built on-top of the Google Web Toolkit (GWT). These frameworks allow for the rapid prototyping of web applications using cross-compiled Java code, while also allowing for finer manipulation of values using native javascript where appropriate. GWT allows for the development of both server and client side code using shared libraries, facilitating easy implementation of server-client communication using plain java objects. Additionally, extensions to the toolkit were used to allow for easy implementation of websocket-

ets communication¹¹. The output of GWT, and by extension, GXT is a compiled web-application that can immediately be hosted on a server. This software is available under GNU General Public License version 3 (GPLv3), and GWT is released under the Apache License, v. 2.0, but as noted on GWT website, the product includes tools released under other licences[20].

4.1.2 Apache Tomcat 9

The Apache Tomcat suite was used to develop a separate web-application which managed communication via websocket. From the Apache website, "The Apache Tomcat® software is an open source implementation of the Java Servlet, Java Server Pages, Java Expression Language and Java WebSocket technologies"[21]. Tomcat is released under the Apache License, v. 2.0.

4.1.3 Apache Derby

Apache Derby is an open source relational database implemented in java, and available under the Apache License, V. 2.0[22]. Derby was used in embedded mode on the server to handle user account management and persistence of designs across sessions.

4.1.4 JBCrypt

jBCrypt is a Java implementation of OpenBSD's Blowfish password hashing code, as is described in 'A Future-Adaptable Password Scheme'[23]. jBCrypt was used as an alternative to storing user credentials in plain text or sending them in plain text over the network, in order to provide a bare minimum of security.

4.2 Design of a Smart Usage Policy

Smart Usage Policies, as defined in Section 3.3.1, are an array of conditional arguments which take as input the current state of the battery, B , and give as

¹¹<https://github.com/nmorel/gwt-jackson> (Jan 2019)

output a pair of numeric values, P : The proportion of power to purchase, based on the current net usage, if any, and S : The proportion of power to sell, based on the current net generation, if any. There are two options for execution of a policy at run time:

1. Pick the highest B policy, where B does not exceed the battery capacity
2. Interpolate between the highest B policy, where B does not exceed the battery capacity, and the lowest B policy, where B does exceed the battery capacity

For the purposes of this thesis, the first option is chosen. The reader may note that the first option may actually fail to receive a result when rules exist. This can occur when every value of B exceeds the current battery state. There are two options to negate this, the first of which is adding an additional condition to the execution:

- Pick the highest B policy, where B does not exceed the battery capacity. If none exists, then pick the Lowest B policy, where B does exceed the battery capacity.

Alternatively, a 'virtual' condition can exist, with the value $B = 0$, and the values of P and S being equal to the P and S values of the next lowest B unit. For the purposes of this thesis, virtual values exist at $B = 0$ and $B = 100$, as shown below.

An example of a policy is given.

Though a policy should be designed with the details of the unit in mind, the design of a policy is roughly applicable to any other valid unit. Supposing the unit does not generate power, and relies only on the purchase of power to maintain a reserve. One may note the following from the Figure 5:

1. If virtual rules exist at $B = 0$, $B = 100$, then the first threshold $B = 20$ may as well read $B = 0$.
2. When the stored power is below 50%, but above 0%, the power purchased will always be 110% of what is needed for operation, if available

Stored Power Threshold	Power Requested	Power Offered
Stored Power \geq 20%	Usage * 1.10	Surplus * 0.20
Stored Power \geq 50%	Usage * 0.50	Surplus * 0.50
Stored Power \geq 75%	Usage * 0.20	Surplus * 1.30
Stored Power \geq 85%	Usage * 0.10	Surplus * 1.90
Stored Power \geq 95%	Usage * 0.00	Surplus * 2.90

Figure 5: Textual Representation of a policy within UoW GREET

3. When the stored power is above 50%, the amount purchase will always be half of what was used, or less.
4. Assuming power is always available for purchase, one can expect the reserve battery capacity to hover around 50%. Considering also that the unit spends time both refilling and discharging stored power, the unit then wastes power as it cycles between 45% and 55% power stored

Consider instead a unit that relies on solar generation to provide a majority of consumed power. During good stretches of weather, the unit may hover around 70-75% stored power. If power cannot be sold (all other units in the neighborhood are generation, for example), then it may be stockpiled even further, and thus given even higher priority to be sold when the opportunity is available.

During stretches of bad weather, power will drop until it encounters the same loop that the pure-consumer unit above would experience. This loop would be maintained until the unit is able to generate enough power to account for more than 50% of the power it consumes, or network instability causes the purchase of power to be infeasible. A more appropriate policy may include a window of space where the power requested is always at 100%, for example:

For a unit which does not generate power, this policy will do the following:

1. If the stored power is above 75%, it will be used with minor supplementation from the grid, if possible
2. If the stored power is at or above 60%, then all power used will come directly from the grid, if possible

Stored Power Threshold	Power Requested	Power Offered
Stored Power >= 20%	Usage * 1.10	Surplus * 0.20
Stored Power >= 40%	Usage * 1.10	Surplus * 0.40
Stored Power >= 60%	Usage * 1.00	Surplus * 0.40
Stored Power >= 75%	Usage * 0.20	Surplus * 1.30
Stored Power >= 85%	Usage * 0.10	Surplus * 1.90
Stored Power >= 95%	Usage * 0.00	Surplus * 2.90

Figure 6: Adjusted usage policy. Note that from 60% to 75% battery capacity, power requested will always match the usage of the unit.

3. If the stored power is lower than 60%, then all power used will come directly from the grid, if possible. Minor additional power will be stored, if possible.

During periods in which the grid is able to supply power, battery capacity will approach 60% to 75%, and stabilize when that region has been reached. When the supply of power is restricted or unavailable, power will then be used from the battery to attempt to smooth over the shortage. An example of this policy can be seen in Figure 6.

For a unit which does generate power, attempts will be made to aggressively sell surplus power when the stored power is above 85%, and will otherwise attempt to hover between 60% and 75% power stored when possible.

The implemented tool, UoW GESSO, generates graphs based on the currently active policy: Figure 7 and 8 show proportional requests and offers made.

4.3 Design and Evaluation of a Single Unit

Each and every microgrid is made up of component units: the individual agents which conduct trade, purchase, sell and store power. The UoW GESSO tool provides a rich interface for the design of individual components to be included within a microgrid.

Each unit is made up of three distinct and separate features: consumption,

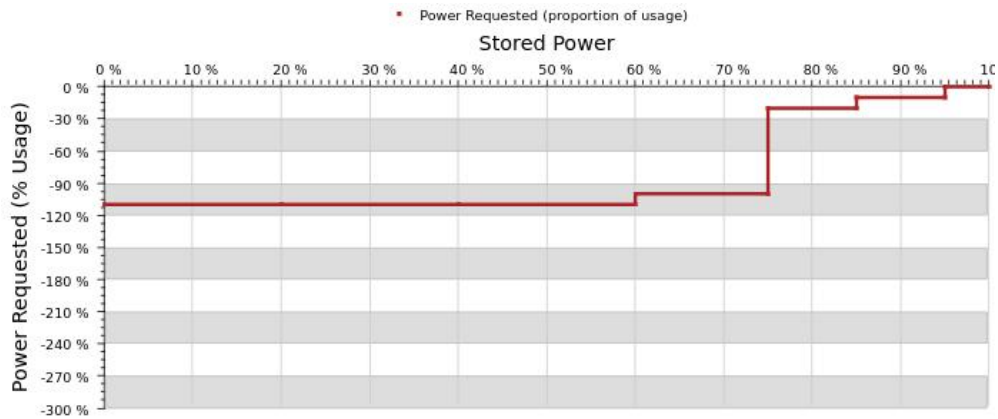


Figure 7: Adjusted usage policy. Note that from 60% to 75% battery capacity, power requested will always match the usage of the unit.

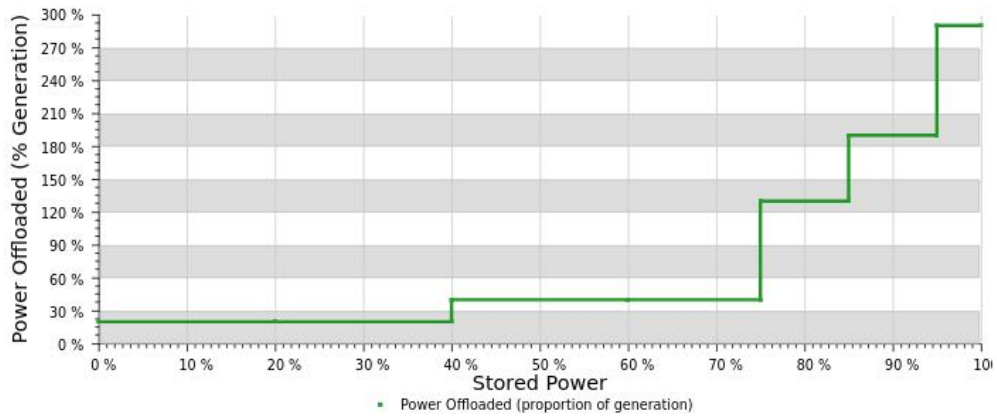


Figure 8: Adjusted usage policy. Note that from 60% to 75% battery capacity, power requested will always match the usage of the unit.

generation, and storage. Separate interfaces are provided for the design of each feature.

4.3.1 Consumption

The consumption for a unit is defined in terms of three variables: The selected profile, the static usage multiplier, and the adjusted usage distribution. An interface is presented in Figure 9. Note the separation between interactive components - active components, being those that can be interacted with, are always in the outermost layer of the design. Interior components, being those that can't be

Define Power Consumption Profile

Enable Consumption Disable Consumption

Select Profile: 14339: [29.84 wH/day] 2 occupants

Consumption Details

House ID: 14339

Heat Pump Installed: Heatpump Installed

Cooking Source: Electricity

Water Heating: Electricity

Space Heating: Electricity & Mains Gas & Fossil Fuels

Children: 0

Adults: 2

Income: 70k or more

Location: City

Yearly Usage: 10921.84 kWh

Daily Usage: 29.84 kWh

Adjust Usage: 1.0

Adjust daily usage profile

Consumption Summary

Adj. Yearly Usage: 10921.84 kWh

Adj. Daily Usage: 29.84 kWh

Figure 9: Interface for the design of power consumption.

Note that detail in the collapsible boxes is non-interactive. A user is able to enable/disable consumption, select usage profiles, multiply the raw usage from the profile, and adjust the distribution of usage daily.

interacted with, are placed within collapsible boxes. This is a convention that is adopted for a large majority of the design of the UoW GESSO tool.

The usage profiles in the 'Select Profile' drop down are processed versions of the household usage profiles available from the University of Waikato Energy Informatics group (UoW EI)[24]. Each profile represents the recorded annual usage of a single household - with average daily figures being provided in the selection dialog. The vertical box below provides information about the selected household, such as:

1. Is a heat pump in use
2. How is space and water heated, and by what means is food cooked
3. The number of children and adults present
4. The income and location type of the household
5. The daily average and yearly usage totals

The dialog titled 'Adjust Usage' allows for a static adjustment to all power

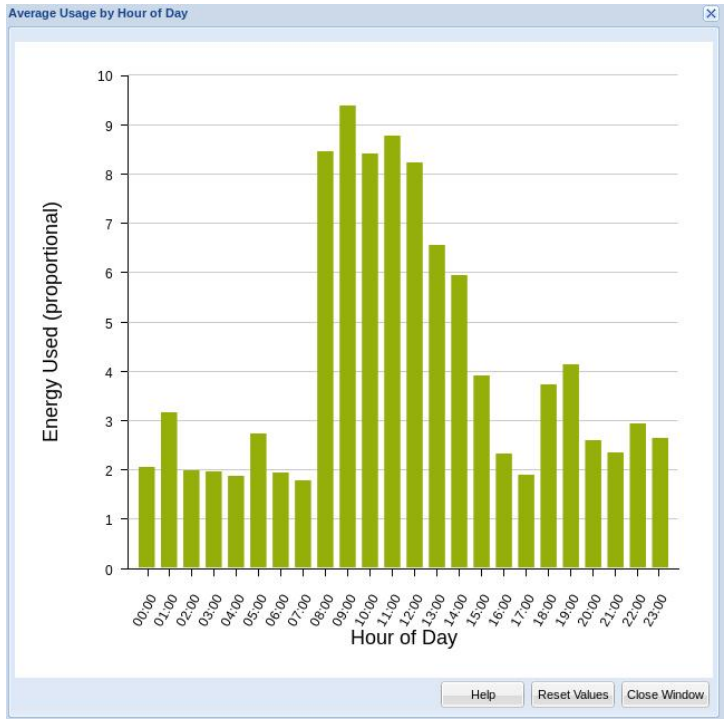


Figure 10: Default distribution of energy on a daily basis for a given usage profile.

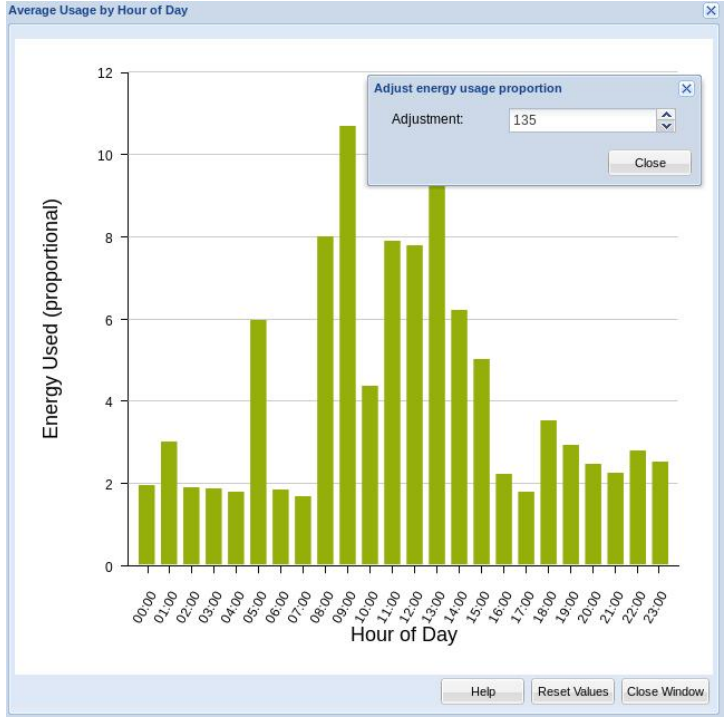


Figure 11: An adjustment made to the default distribution of energy given in Figure

consumed by the household, which may be useful in reaching specific numbers for annual or average daily consumption.

There is a button titled 'Adjust daily usage profile'. Interaction with this button opens up another dialog, showing the distribution of energy usage on average per day in the form of a bar graph. The user may then interact with individual bars in the graph to shape the distribution in the desired way. An example is given in Figure 10, showing the default consumption profile of the household usage data, and Figure 11, showing an adjusted consumption profile.

4.3.2 Generation

Generation within the UoW GESSO tool is restricted to solar generation. For the design of the generation around a unit, the following features may be defined:

Solar Panel The commercial solar panel to be used. The panel is chosen from a restricted version of the data-set provided by The GoSolar California! campaign[8]. Details about the chosen solar panel are shown in a dialog below the selection box (Figure 12).

Quantity The raw number of solar panels used in the array. This is used to calculate the output power, size, and estimated cost of the array.

Azimuth The compass direction of the solar panel array. Ideally, the azimuth should be aligned so that the solar array catches the maximum amount of sunlight each day.

Tilt The angle of the solar array in relation to the ground. A 0 deg tilt would have the panels facing directly upwards, whereas a 90 deg tilt would have the panels facing directly horizontal to the ground.

Inverter The inverter used by the solar panel array. This will be used to convert power from DC to AC to be used directly by the house. A summary of the inverter details is given in Figure 13

The optimal angles for Azimuth and Tilt vary based on the geographic location of the unit. Generally, the azimuth of a non-tracking solar array would be facing

Enable Power Generation Disable Power Generation

Selected Panel:

▲ Solar Panel Details

Manufacturer:

Model Number:

Description:

Output Power:

Long Side:

Short Side:

Panel Quantity:

Azimuth:

Tilt:

Inverter:

Figure 12: Interface for configuration of solar generation for a unit.

▲ Inverter Details

Manufacturer:

Model Number:

Output Power:

Voltage:

Efficiency:

▲ System Summary

Maximum Input Power:

Maximum Output Power:

Total Standard Generation:

Total Area:

Estimated Cost:

Figure 13: Summary of solar generation system.

true north when below the equator, and true south when above the equator - with the tilt of the solar panel being equal to the latitude of the installation. An array directly on the equator would receive near-optimal performance facing directly upwards, whereas an array at latitude -41° would see good performance with the array tilted 41° , and an azimuth facing true north[25].

Based on the parameters given, a summary is created for the generation component of the grid. This includes the maximum input and output power of the chosen inverter, the generation of the solar panels under STC, the total surface area of the solar panel array (not including framing/spacing), and the estimated cost of the system¹². The cost is estimated based on the factored costs of each individual component. For solar panels, from which all data was taken from the GoSolar California data file, no definitive costs were given[8]. To remedy this, estimates were made based on the average cost per kWh of solar panels as was given by EnergySage[10].

4.3.3 Storage

Storage within the UoW tool is restricted to a subset of commercially available batteries only. A user is able to select from a representative sample of batteries that are or have been commercially available in New Zealand. A user is able to pick a model of battery and a quantity of that battery. Most commercially available battery systems include a built in inverter; the component used to convert the stored power to the AC power that the unit consumes.

When the battery does not include a built in inverter system, the user is able to choose a throughput and efficiency value for a custom inverter. The interface can be seen in Figure 14

¹²All costs within UoW GESSO are given in New Zealand Dollars, based on July 2018 exchange rates, where prices in NZD aren't already available. Import duties are not taken into consideration, but a General Sales Tax of 15% is applied.

Enable Storage Disable Storage

Battery:

Battery Details

Manufacturer:	<input type="text" value="Tesla"/>
Model Number:	<input type="text" value="Tesla Powerwall 2 [14 kWh]"/>
Name:	<input type="text" value="Tesla Powerwall 2"/>
Available:	<input type="text" value="Available for retail purchase"/>
Battery Type:	<input type="text" value="Lithium-Ion"/>
Capacity:	<input type="text" value="13.5 kWh"/>
Output Power:	<input type="text" value="5 kW"/>
Price (NZD):	<input type="text" value="\$18000"/>
Inverter:	<input type="text" value="Integrated with system"/>
Efficiency:	<input type="text" value="90%"/>

Battery Quantity:

[%] Inverter Efficiency:

[kW] Inverter Throughput:

Figure 14: Interface for the configuration of unit storage.

Weather Data:

Policy (Type A):

Figure 15: Interface for selection of weather and policy.

4.3.4 Simulation

Once a user has set the storage, generation and consumption parameters for a unit, they are ready to perform their first simulation. The steps to performing a simulation are as follows:

1. Select weather data
2. Select a smart usage policy for the unit (optional)
3. click the 'Simulate Unit' button to bring up the 'Simulate Unit' dialog
4. Set the parameters to be used for the simulation of the unit
5. Click the 'Simulate' button

4.3.4.1 Weather

A curated selection of weather data is available, based on data taken from New Zealand's National Climate Database¹³. A user is able to select the weather data used through the weather selection interface, as seen in Figure 15. The data has been processed to remove gaps from the data - a large selection of the weather stations that can be accessed via the NIWA system experience outages, either through equipment malfunction, software/hardware upgrades, or reliance on local power grids and internet connections for operation. The process for cleaning of the data was as follows:

1. For a given year, at least 90% of entries must be present
2. For any given month, at least 90% of the entries must be present
3. For single instances of missing values, the average of the previous and subsequent values is used
4. For months in which there were strings of missing values, The average radiation values for each valid hour of the day within that month were calculated and used as fill ins for missing values.

4.3.4.2 Simulation Parameters

The simulation dialog, as shown in Figure 16, presents a number of parameters to the user. These are as follows:

Simulate Outages Outages may be simulated year round, February onwards, or not at all. The February onwards option exists to allow for a degree of stabilization for the unit before outages start to occur. The expected frequency and duration of each outage can also be set.

Corrected Angles This option allows for the simulation of solar generation to take into account the relative angles of the solar panel and position of the sun in the sky at any time. If this option is unchecked, the unit is treated

¹³This is accessed via the cliflo system. <https://cliflo.niwa.co.nz/>

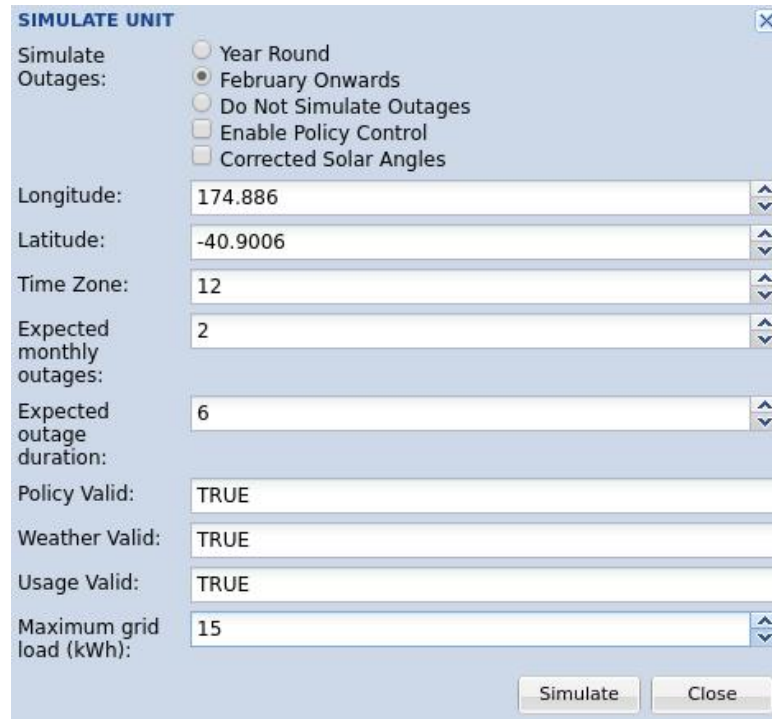


Figure 16: Interface and parameter selection for simulation of individual unit.

as if the solar panels have perfect tracking. This option relies on the user setting the correct longitude, latitude and time zone options.

Maximum Grid Load The maximum load the grid is able to support (in kWh per hour¹⁴). If this option is set to 0, the grid is assumed to have unlimited capacity to store or distribute power.

Policy Control Toggle The ability of the unit to utilize smart usage policies is optional. This may be enabled or disabled within the simulation interface.

Validity Checks Fields are in place as checks to ensure that the selected policy (if any) is valid, the selected weather is valid, and the selected unit consumes power.

¹⁴Considering the resolution of simulation is in single hour intervals, the usage of kWh/h as an alternative simply to kWh would seem appropriate

4.3.4.3 Simulation Results

Once all options have been chosen, the user is able to perform the simulation. The length of the simulation accounts an entire year¹⁵, and the following sets of data are presented to the user:

Usage and Generation Values for the usage and generation of power. This includes the raw generation, the usable generation, the raw intended usage, the net surplus power, and the net intended usage. Each of these categories is described:

1. **Raw generation:** the raw generation performed by the solar array. This is before losses due to inversion and storage.
2. **Usable generation:** The generation usable by by the unit - this is the generation after conversion and storage losses, if any, have been taken into account.
3. **Raw intended usage:** The target consumption of the unit. This is the the consumption value taken from the household usage data, after accounting for multipliers and any modifications made to the distribution profiles.
4. **Net surplus power:** The net surplus power is the positive component of the function *usable generation - raw intended usage*. This value is never negative.
5. **Net intended usage:** The net intended usage is the positive component of the function *raw intended usage - usable generation*. This value is never negative.

One may expect that, with *net surplus power* and *net intended usage* being essentially the positive and negative components of the same function, that either value being positive implies the other value has negative value (clamped to 0). This is true on the basis of individual intervals of observation (one hour), but as aggregated intervals (per day, per 28 days) take

¹⁵Actually 13 billing cycles of 28 days

averages of the next-lowest-level of data, and the values are clamped to zero rather than being negative, it is possible for there to be both average net surplus and average net usage for a period.

This data is available by 28-day period, by individual day, and by hour. See Figure 17, 24 (Appendix) and 27 (Appendix) for examples.

Grid Interaction Values for the interaction of the unit with the macrogrid. This includes the power requested from the grid, the total draw of the unit upon the grid, the power offered to the grid, the power pushed to the grid, the power wasted, and the power in deficit. Each of these categories is described:

1. **Power requested:** The power that the unit has requested from the macrogrid. This is based on the currently active policy, if any.
2. **Grid draw:** The power the unit has actually drawn from the grid. This will not match the power requested in the case that the request oversteps the capacity the grid has to supply power.
3. **Power in deficit:** The amount of power required to make up the target usage, in cases where the supply of the grid and the available power through generation and storage cannot meet the demand.
4. **Power offered:** The amount of stored or generated power offered from the unit to the grid. This is based on the currently active policy, if any.
5. **Power pushed:** The amount of power that has been pushed to the grid from this unit.
6. **Power wasted:** The amount of power which was generated in excess of the ability to be used, which cannot be stored, and cannot be sold to the grid.

In practice, provided the grid is able to meet all supply and demand, the power requested will be equal to the grid draw, and the power offered will be equal to the power pushed. If no policy is enabled, then all generation will be used first for the current demand, then for storage, and then for sale. Meanwhile, power will first be used from generation, then from stored power, and finally from purchase of power from the grid.

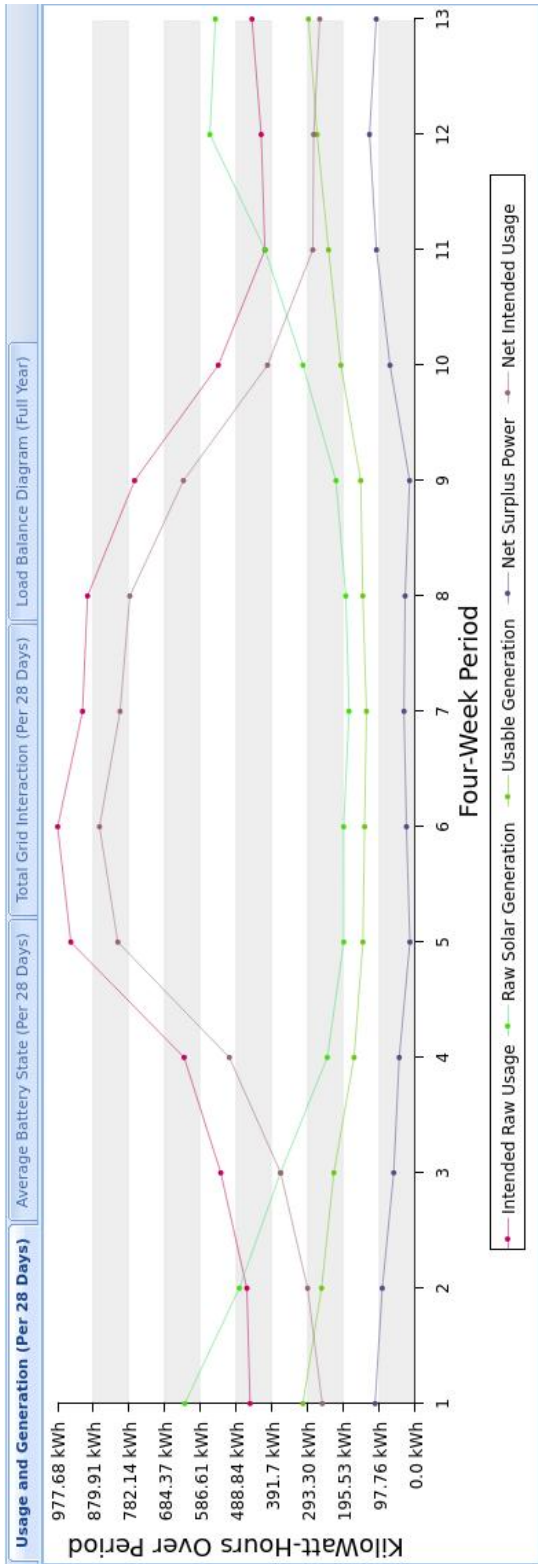


Figure 17: Generation and Consumption values averaged by 28-day period.

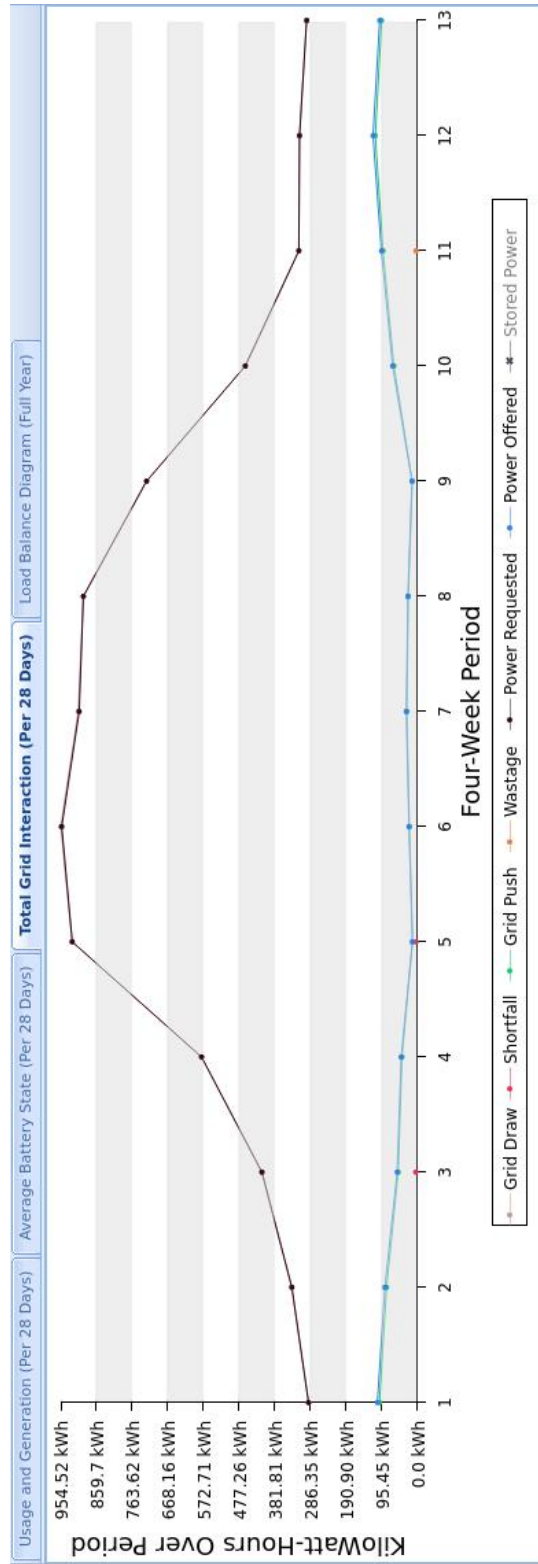


Figure 18: Grid Interaction values averaged by 28-day period.

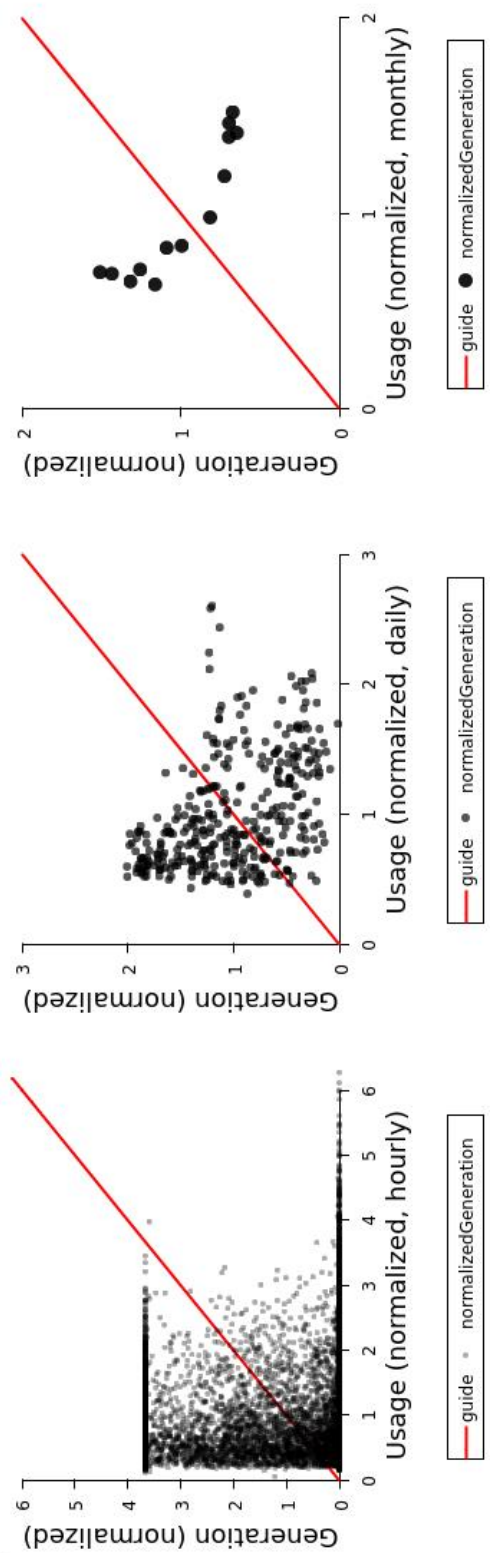


Figure 19: Load balance diagrams given by 1 hour, 24 hour, and 28 day periods.

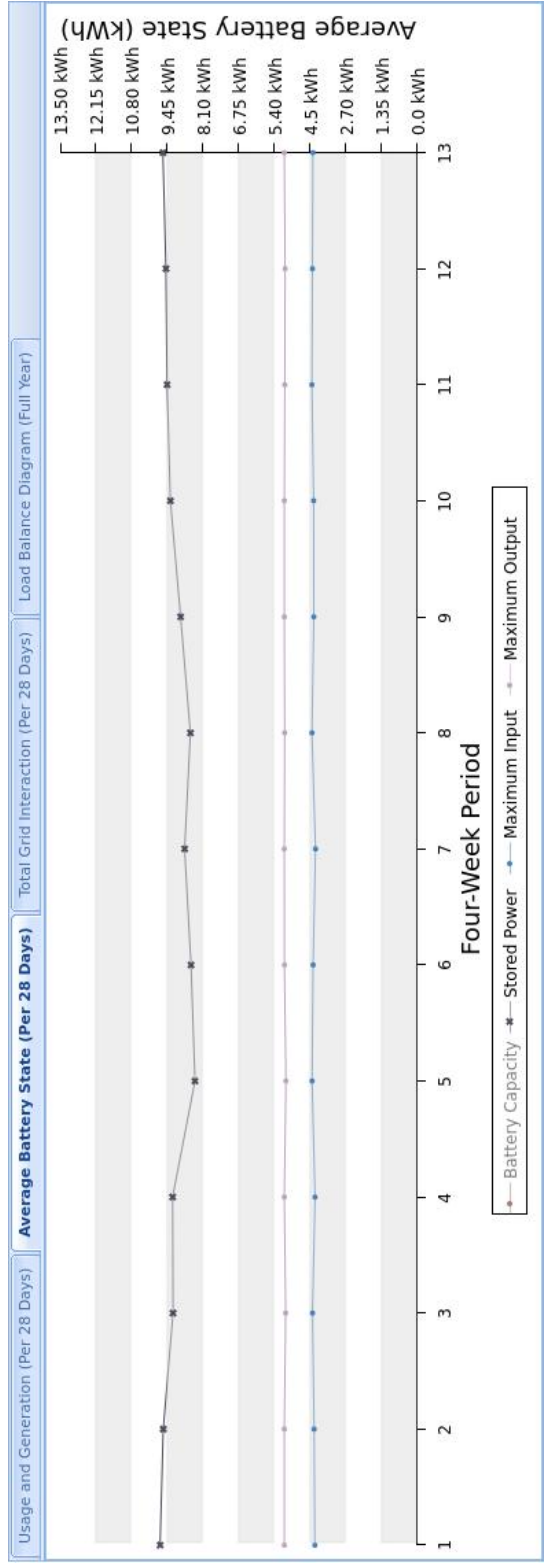


Figure 20: Grid Interaction values averaged by 28-day period.

This data is available by 28-day period, by individual day, and by hour. See Figures 18, 25 (Appendix) and 29 (Appendix) for examples.

Battery State Values are given for the average battery state over the period if the data is aggregated, or the battery state at the beginning of the period if the values are at the lowest resolution. This includes:

1. **Stored power:** The amount of charge the battery has (in kWh).
2. **Maximum Input:** The maximum charge the battery is capable of accepting at the time.
3. **Maximum output:** The maximum amount of power the battery is capable of discharging at the time.

This data is available by 28-day period, by day, and by hour. See Figures 20, 26 (Appendix) and 28 (Appendix) for examples.

Load Balance A series of interval energy balance plots are given, comprising the single hour, 24 hour and 28 day periods of observation. These are scatter-plots showing all instances of generation vs. consumption. For a Net-ZEB system, the expectation would be that the generation and consumption match for the year. See Figure 19.

These graphs are grouped by time period, and a user is able to interact with one graph to expose the data in a more granular fashion. In this way, a user is able to expose the root of any anomalous data: in a month with particularly high shortfall, for example, the user is able to look at the individual days and note whether the anomaly is a general trend for the month, or restricted to a particular set or stretch of days. From then they're able to look at each noteworthy day, and determine from then what the root issue may be - terrible weather, inconvenient power outages, extraordinarily high usage: all things that cannot easily be determined from looking at the high level data.

4.4 Structure of a System

In the context of the UoW GESSO tool, a system is a collection of units comprising one microgrid. As defined in Section 3.6, there are 7 basic types of system that the UoW GESSO tool will manage: These include units with only a single feature (Generation, Consumption, Storage), units with two of the features, and units with all three of the features.

The given definition of a system can be expanded slightly further if an intermediate step is included: *modules*. Each module is a collection of one or more entirely homogeneous units and configurations. A system is then made up of modules, which are made up of units and configurations¹⁶.

Each module is composed of three things:

1. A template unit
2. A quantity of that unit
3. A configuration for that unit

It is worth noting that the configurations apply to each unit individually, rather than to a module as whole. The necessary configurations are as follows:

Consumption: Maximum draw, maximum purchase price.

Generation: Maximum power offloaded, sale price.

Storage: Purchase threshold, purchase price, sale threshold, sale price, where:

1. The purchase threshold is the proportion of stored power below which power will be purchased
2. The sale threshold: the proportion of stored power above which power will be sold

Consumption + storage: Maximum price, smart usage policy

Storage + generation: Purchase threshold, purchase price, sale threshold, sale price

¹⁶The term *configuration* is used to refer to the supplementary information associated with each unit: policies and other details, as mentioned in Section 3.6

Consumption + generation: Maximum draw, maximum purchase price, maximum power offloaded, sale price.

Consumption + storage + generation: Maximum draw, maximum purchase price, sale price, maximum offload, smart usage policy

As was noted in section 3.6.1.9, the macrogrid to which the neighborhood is attached can conceptually be treated as an individual unit. In this case, the features available to the macrogrid are as follows:

1. Maximum load
2. Sale price
3. Purchase price

In situations where there is a power outage, the maximum load will be zero, and the grid will be unable to interact with the system.

4.5 Calculations for Simulation of a Single Unit

Within this section, all the calculations used to perform generation on single units will be presented. The exact Java code used will not be presented here, but instead a condensed and at times abbreviated version - this is done for clarity, and the full code is available online¹⁷.

The rough guideline for a single unit was presented in Figure 4, and to an extent the actual simulation follows.

When calculating the performance of a unit, the variables being used must first be defined:

hour, day, month: Integer values - hour of day, day of month, month of year.

In this circumstance, a month represents exactly 28 days, rather than a calendar month.

generationConstant: The 'generation constant' is the information relating to generation which should remain constant over all calculations - the efficiency of the panel, surface area, and number of panels are all taken into account

¹⁷<https://github.com/NBKelly/UoW-GESSO> (Sep 2019)

totalAdjustment, hourAdjustment[]: The scaling factors applied to the user-selected usage profiles. `totalAdjustment` represents the total scaling applied, and `hourAdjustment` is an array of adjustments made to each hour of the day, applied to shape the average load profile to the desired shape.

In the process of performing calculations for an individual time-period (in this case, one hour), the following is performed:

1. First, basic information about the unit is established. If the unit generates power, the following values are configured:

```
1 double generationConstant = PV.getOutputPower()  
2                             * numberOfPanels;  
3 double maxWatts = inverter.getOutputPower();  
4 double generationInversion = inverter.getEfficiency();
```

2. The condition of the macrogrid is established:

```
1 boolean isPowerOut = (isScheduledPowerOutage(hour, day, month);  
2 double gridCapacityRemaining = (isPowerOut ? 0 : gridCapacity);
```

3. The usage is determined, taking into account the printed usage from the selected usage profile, the distributed adjustment, and total adjustment, as set by the user in Section 4.3.1.

```
1 double whUsed = getUsage(hour, day, month)  
2                 * hourAdjustment[hour]  
3                 * totalAdjustment;
```

4. If the unit generates power, then the generation is performed. Horizontal irradiance is assumed to be a set proportion of the global irradiance. In this case, the value of 25% is used.

```
1 //get global radiation from weather data  
2 double station_global = getWeather(hour, day, month);  
3 /* The calculate_solar function calculates the power  
4    * that is directly striking the solar panel (plane of array).
```

```

5  */
6  double solarValue = calculate_solar(station_global,
7                                     panelTilt, panelAzimuth);
8  /* The calculate_solar function doesn't deal with
9   * cases where the sun is behind the panel, so
10  * that is done here
11  */
12  solarValue = Math.max(solarValue, 0);
13  //an estimation for the DHI is used
14  solarValue += 0.25 * station_global;
15
16  //The radiation is used to make a (very rough)
17  // estimate of the power generated
18  double generatedWH = solarValue * generationConstant;
19  double usableWH = generatedWH * generationInversion;
20  usableWH = Math.min(usableWH, maxWatts);

```

5. The state of the battery is determined, using the previous state of the system and collated information on battery specifications

```

1  //these values are in watt-hours
2  int maxOutputPower = battery.getMaxOutput();
3  int maxInputPower = battery.getMaxInput();
4  //we can simplify I/O efficiency loss by applying
5  // both on the input of power. Current stored power
6  // is in relation to this value
7  double storageLoss = battery.getEfficiency()
8                       * battery.getEfficiency();
9  int maxStorageSpace = (int)(battery.getCapacity()
10                          * storageLoss);
11  int storageSpaceAvailable = maxStorageSpace
12                              - powerStored;
13  //effective max I/O is calculated
14  maxInputPower = Math.min(storageSpaceAvailable, maxInputPower);
15  maxOutputPower = Math.min(maxOutputPower, powerStored);

```

6. The usage and generation values are collated to determine *net_usage* and *net_surplus*. Negatives numbers are clamped to 0, in order to allow for more meaningful information read-outs.


```

1  int net_usage    = Math.max(whUsed - usableWH, 0);
2  int net_surplus = Math.max(0, usableWH - whUsed);

```

7. If there is a surplus of power, and no policy is enabled, the unit attempts to store and then sell excess power

```

1  if(net_surplus > 0 && !policyEnabled) {
2      if(net_surplus * storageLoss <= maxInputPower) {
3          storedPower += net_surplus*storageLoss;
4          maxInputPower -= net_surplus*storageLoss;
5      }
6      else {
7          remaining_surplus = net_surplus -
8                               (maxInputPower / storageLoss);
9          storedPower += maxInputPower;
10         intended_sale = remaining_surplus;
11         maxInputPower = 0;
12     }
13 }

```

8. If there is a deficit of power, and no policy is enabled, the unit attempts to use stored power and then purchase power.

```

1  if(net_usage > 0 && !policyEnabled) {
2      if(net_usage <= maxOutputPower) {
3          storedPower -= net_usage;
4          maxOutputPower -= net_usage;
5      }
6      else {
7          remaining_usage = net_usage - maxOutputPower;
8          storedPower -= maxOutputPower;
9          intended_purchase = remaining_usage;
10         maxOutputPower = 0;
11     }
12 }

```

9. If there is a deficit of power, and a policy is enabled, the unit attempts to purchase power according to that policy.

```

1  if(net_usage > 0 && policyEnabled) {
2      int request = net_usage;
3      request *= policy.getRequest(powerStored, maxStorageSpace);
4      request = Math.min(request,
5                          maxInputPower /storageLoss + net_usage);
6      remaining_usage = net_usage;
7      intended_purchase = request;
8  }

```

10. If there is a surplus of power, and a policy enabled, the unit attempts to sell power according to that policy.

```

1  if(net_surplus > 0 && policyEnabled) {
2      int offer = net_surplus;
3      offer *= policy.getOffer(powerStored, maxStorageSpace);
4      offer = Math.min(offer, maxOutputPower + net_surplus);
5      remaining_surplus = net_surplus;
6      intended_sale = offer;
7  }

```

11. The unit has determined how much power it needs to buy from the grid or sell to the grid, and the state of the grid. The only remaining component of the simulation is that the unit is able to perform a transaction with the macrogrid, if possible.

```

1  if (intended_sale > 0) {
2      if (gridCapacityRemaining >= intended_sale)
3          power_sold = sellPower(intended_sale);
4      else if (gridCapacityRemaining > 0)
5          power_sold = sellPower(gridCapacityRemaining);
6
7      //if there is remaining power, we try to store it
8      int remaining_power = intended_sale - power_sold;
9      if(remaining_power > 0 && maxInputPower > 0) {
10         if(remaining_power * storageLoss <= maxInputPower) {
11             powerStored += remaining_power * storageLoss;
12             remaining_power = 0;
13         }
14     else {

```

```

15         remaining_power -= maxInputPower / storageLoss;
16         powerStored += maxInputPower / storageLoss;
17     }
18 }
19 power_wasted = remaining_power;
20 }
21
22 if(intended_purchase > 0) {
23     if (gridCapacityRemaining >= intended_purchase)
24         power_brought = buyPower(intended_purchase);
25     else if (gridCapacityRemaining > 0)
26         power_brought = buyPower(gridCapacityRemaining);
27
28
29     //if there is remaining deficit, we try to use the battery
30     int remaining_power = intended_purchase - power_brought;
31     if(remaining_power > 0 && maxOutputPower > 0) {
32         if(remaining_power <= maxOutputPower) {
33             powerStored -= remainingPower;
34             remaining_power = 0;
35         }
36         else {
37             remaining_power -= maxOutputPower;
38             powerStored -= maxOutputPower;
39         }
40     }
41     power_deficit = remaining_power
42 }

```

Note that the required inputs for each time-period are either predetermined (features of the unit), determined by the time-period itself (Usage, Weather), or are also outputs of the simulation (battery state). It then follows that being able to simulate a single time-period, it is possible to simulate multiple linear time-periods by feeding the output of each simulation into the input of the next. For the initial simulation, the chained value (battery state) still requires initialization. For the purposes of this thesis, it was decided that starting the simulation with the battery at 80% capacity was appropriate.

4.6 Calculation for the Simulation of Solar Radiation

In Section 4.5, calculations are made to transform global solar radiation into radiation directly striking a solar panel. These calculations are all direct implementations of the ones given in 3.2. To increase efficiency, several shortcuts are taken:

1. The Local Standard Time Meridian is calculated only once
2. The values for the Time Correction factor, equation of time, declination of the sun, solar time, hour angle of the sun, elevation angle of the sun, zenith angle of the sun and azimuth angle of the sun are calculated once per time period, and re-used for each individual calculation performed.
3. The angle of incidence calculations are batched for units which share array orientation specifications

As mentioned in 3.2, actual generation performed by solar arrays is considered to be directly linear in relation to radiation received. The power received by these panels at standard test conditions is used as the baseline for this calculation. Furthermore, since standard test conditions specify radiation received per meter squared, the values are size-agnostic. This means that the only factors needing to be considered once actual radiation per square meter has been calculated are power generated at standard test conditions, number of panels in array and maximum inverter throughput.

4.7 Calculations for Simulation of a System of Units

Because systems of units are non-homogeneous, a more modular approach must be taken to simulation. It is still true that the output of each time-period can be used as the input of the next, but given that each unit may have entirely different behaviour, and each unit needs to be able to interact with each other unit, a modular approach to simulation must be defined.

First, a sequence of actions is defined which allows for all behaviors belonging to each unit. This is based on the workflow given in Section 3.6. The workflow

for the simulation of a unit coordinator can be modified with the intermediate requirements other units have, in order to define a suitable workflow for simulation.

1. The static component of the solar radiation calculation is performed
2. Generation over the period is simulated for each unit that generates power
3. The state of the macrogrid is determined
4. Usage over the period is determined for each unit that consumes power
5. Every unit which has the capacity to sell power decides upon a quantity and price it wishes to sell at
6. Every unit which has the capacity to purchase power decides upon a quantity and price it wishes to buy at
7. All offers over the period are sorted and binned by price
8. Starting with the lowest priced set of offers on the network, all units which are willing to purchase at that price buy power equally, up to the amount that they are willing to purchase. This step is repeated, so long as there is at least one unit willing to purchase power from one other unit.
9. If there is no more demand at the prices units intend to sell power, but there is still supply, then the following steps are performed:
 - (a) Each unit that still has supply must decide if it wishes to store power (if possible), or attempt to sell at a lower price
 - (b) Units which wish to sell at a lower price price-match the highest priced requests, so long as there still exist request to match
 - (c) When this process is completed, any remaining power must be stored, given away freely, or wasted.
10. All statistics must be updated based on transactions that took place, or were unable to take place, during the current time period

Given this set of requirements, a generic class might be set up in place of any unit (or module), with template implementations of all the things *any* given unit is capable of doing. One feature, not mentioned above, is also worth noting: unlike with the simulation of a single unit, there can be no reliance on all the data

having been preloaded from the server. In particular, the given usage data for each unit is loaded at run-time. Hence each component may need to make use of the unit database to set up all of the appropriate variables needed to operate. A unit may also need to finalize transactions and values after all trading is complete for a period of time.

```

1  abstract class GridComponent {
2      protected UnitData unit;
3      private int quantity;    //allowance for homogenous groups
4
5      public abstract void setup(Database DB);
6      public abstract void performGeneration(double wattHours);
7      public abstract void determineUsage(int period);
8
9      public abstract Offer makeOffer();
10     public abstract Request makeRequest();
11
12     public abstract void buy(double quantity, int price);
13     public abstract void sell(double quantity, int price);
14
15     //try store remaining unsold power. Most units will not implement this
16     public void tryStoreRemainer() { }
17
18     //finalize all transactions for the period
19     public void finalize() {}
20
21     //is the unit able to price match with excess unsold power?
22     public boolean willPriceMatchExcess() { return false; }
23     //is the unit able to give away any quantity of free power?
24     public boolean willGiveFreePower() { return false; }
25 }

```

A selection of partial unit implementations is presented. Only the relevant details are presented, and some sections are omitted for clarity.

The C type unit provides examples of how an individual unit is constructed, how the quantity of homogeneous units is used in adjusting data, how a request might be made, how information is recorded (within the *buy* function), and some

usage from the *finalize* function.

```
1 public class GridComponent_C extends GridComponent {
2     public GridComponent_C(int maxdraw, int maxPrice, int quantity){
3         setQuantity(quantity);
4         setMaxDraw(maxDraw) * getQuantity();
5         setMaxPrice(maxPrice);
6     }
7
8     private double periodUsage;
9     private double usageAdjust;
10    private double[] hourlyAdjust;
11
12    @Override void setup(Database DB) {
13        loadUsageData(DB);
14        usageAdjust = unit.getConsumptionAdjustment()
15            * getQuantity();
16        hourlyAdjust = unit.getHourAdjustment();
17    }
18
19    @Override public void determineUsage(int period) {
20        periodUsage = getUsage(period) * usageAdjust
21            * (hourlyAdjust[period % 24]);
22    }
23
24    @Override public Request makeRequest() {
25        //if maxDraw is 0, that means no restriction on usage
26        double requestValue = Math.min(maxDraw <= 0 ? periodUsage
27            : maxDraw, periodUsage);
28        return new Request(requestValue, maxPrice, this);
29    }
30
31    @Override public void buy(double quantity, int price) {
32        //make use quantity, price, recorded_quantity, recorded_price
33        // to determine new average values
34        double averagePrice = quantity*price;
35        averagePrice += getData().getAverageBuyPrice()
36            * getData().getWhPurchased();
37        averagePrice /= (double)(quantity
38            + getData().getWhPurchased());
```

```

39     getData().setAverageBuyPrice(averagePrice);
40     getData().setWhPurchased(getData().getWhPurchased()
41         + quantity);
42     periodUsage -= quantity;
43 }
44
45 @Override public void finalize() {
46     //shortfall defaults to Double.NaN, which allows use to
47     // more easily have a graph with gaps in it
48     if(periodUsage > 0)
49         this.getData().setShortFall(periodUsage);
50 }
51 }

```

The sale of power, and offers to sell power by individual units are implemented in a similar fashion to requests to buy and the purchase of power. When applicable, generation will precede these requests and offers.

```

1 public class GridComponent_G extends GridComponent {
2     ...
3     @Override public void performGeneration(double wattHours,
4         boolean solarAngles) {
5         wattHours = Math.max(solar_calc.calculate(wattHours, panelTilt,
6             panelAzimuth) * -1.0, 0) + wattHours*0.25;
7         generatedWH = wattHours * generationConstant;
8         usableWH = Math.min(generatedWH*generationInversion, maxWatts);
9
10        if(usableWH > maxLoad && maxLoad > 0) {
11            double waste = usableWH - maxLoad;
12            getData().setWasted(waste);
13            usableWH = maxLoad;
14        }
15    }
16    ...
17    @Override public Offer makeOffer() {
18        return usableWH > 0 ? new Offer(usableWH,salePrice,this):null;
19    }
20
21    @Override public void sell(double quantity, int price) {

```



```

22     double averagePrice = quantity * price;
23     ...
24     usableWH -= quantity;
25 }
26     ...

```

If a unit generates more power than it can use or sell regularly, it may attempt to give away excess power, or sell it at a lower price. The unit coordinator handles the ordering of these functions, and the individual unit only needs to return a specific true/false value based on capability. In the case of an individual unit which only generates and sells power, power can be given away or sold at lower prices any time in cannot be regularly sold.

```

27     ...
28     @Override public boolean willPriceMatchExcess(){return usableWH>0;}
29     @Override public boolean willGiveFreePower(){ return usableWH>0;}
30 }

```

The 'Offer' and 'Request' objects act as signals containing the details of a desired transaction - The amount of power offered or requested, the expected price of the transaction (in relation to the coordinator price), and the source component offering or requesting the transaction. In accordance with step 7 of the workflow presented for the unit coordinator, offers and requests must be able to be sorted and binned. Since the Request class is virtually identical to the Offer class, only the Offer class will be shown.

```

1  class Offer implements Comparable<Offer>{
2      private double powerInWH;
3      private int sellPricePerKWH;
4      private GridComponent source;
5
6      public Offer(double powAmount, int price, GridComponent source) {
7          this.setPowAmount(powAmount);
8          this.setSellprice(price);
9          this.source = source;
10     }

```

```

11
12     @Override public int compareTo(Offer arg) {
13         return sellPricePerKWH - arg.sellPricePerKWH;
14     }
15
16     public void sell(double quantity, int price) {
17         source.sell(quantity, price);
18         powerInWH -= quantity;
19     }
20 }

```

A class titled 'OfferGroup' is responsible for the management of binned offers. Power may be sold from a distribution of offers at once, with weight to the number of units represented by the offer. As mentioned previously within this thesis, the distributed sale of power is to be divided equally among all units which would sell at the same price, if possible. The 'OfferGroup' class is responsible for ensuring that this is the case. If an individual wished to change this behaviour, the change would most easily be made within the 'OfferGroup' class.

```

1 class OfferGroup {
2     private ArrayList<Offer> offers;
3     private double powerRemaining;
4     private int sellPrice;
5     private double shares;
6
7     public OfferGroup(ArrayList<Offer> offers, int price) {
8         this.setOffers(offers);
9         this.setPrice(price);
10
11         //sum up all the power offered by this group,
12         // and count the amount of distinct individual units contained
13         for(Offer offer : offers) {
14             powerRemaining += offer.getPowAmount();
15             shares += offer.getSource().getQuantity();
16         }
17     }
18
19     //this will try to sell in even quantities per each individual unit

```

```

20     public void sell(double amount, int price) {
21         powerRemaining -= amount;
22         while(amount > 0 && getOffers().size() > 0) {
23             //determine a share that each unit can sell, if able
24             double share = amount / shares;
25             for(int module = 0; module < getOffers.size(); module++) {
26                 Offer offer = getOffers.get(module);
27                 //determine how much the module has to sell
28                 double draw = offer.getPowAmount();
29                 //ensure that no more than the share * quantity is sold
30                 draw = Math.min(draw,
31                                 share*offer.getSource().getQuantity());
32                 offer.sell(draw, price);
33                 if(offer.getPowAmount() == 0) {
34                     //if an offer is completely exhausted,
35                     // remove it from this bin
36                     shares -= offer.getSource().getQuantity();
37                     getOffers().remove(module);
38                 }
39                 amount -= draw;
40             }
41         }
42         powerRemaining += amount;
43     }
44 }

```

4.7.1 Implementation of Unit Coordinator

With the definitions provided for the majority of functions that the usage coordinator takes advantage of, the function of the unit coordinator can be meaningfully defined.

The coordinator takes note of which component is operating on behalf of the macrogrid, and performs the preliminary setup on each component.

```

1     public ArrayList<Data> calculate2() {
2         GridComponent_Grid = getGrid();
3         for(GridComponent g : gridComponents)

```

```
4      g.setup(DB);
```

The preliminary setup is out of the way, and the simulation itself can begin. For each time period, the state of the grid is determined and the preliminary (static) solar calculations can be performed.

```
5      for(int period = 0; period < duration; period++) {
6          ArrayList<Offer> offers = new ArrayList<Offer>();
7          ArrayList<Request> requests = new ArrayList<Request>();
8
9          boolean isPowerOut = getPowerOut();
10         solar_calc.Precalculate();
```

Each unit creates a new data-log for the current time period, then resolves and generation and consumption before making any offers or requests, if possible.

```
11         for(GridComponent g : gridComponents) {
12             g.newData(); //start new data log
13             g.performGeneration(getMJFromWeather());
14             g.determineUsage(period);
15
16             if(g == grid && isPowerOut)
17                 grid.setIsPowerOut(isPowerOut);
18
19             Offer offer = g.makeOffer();
20             if(offer != null)
21                 offers.add(offer);
22
23             Request req = g.makeRequest();
24             if(req != null)
25                 requests.add(req);
26         }
```

All offers are sorted by price, and then binned using 'OfferGroups'.

```

27     Collections.sort(offers);
28     ArrayList<Offer> _offers = new ArrayList<>(offers);
29
30     int __price = 0;
31     ArrayList<Offer> group = new ArrayList<>();
32     ArrayList<OfferGroup> groupings = new ArrayList<>();
33
34     while(offers.size() > 0) {
35         Offer offer = offers.remove(0);
36
37         //if the current group is empty, or the prices match,
38         // then we add the current offer to the group
39         if(group.isEmpty() || offer.getSellPrice() == __price) {
40             group.add(offer);
41             __price = offer.getSellPrice();
42         } else {
43             //otherwise, start a new group and cache the current one
44             groupings.add(new OfferGroup(group, __price));
45             group = new ArrayList<>();
46             group.add(offer);
47             __price = offer.getSellPrice();
48         }
49     }
50
51     //add the final group to the list of groups, if it exists
52     if(group.size() > 0)
53         groupings.add(new OfferGroup(group, __price));

```

For each OfferGroup, from lowest price to highest price, power is sold. The first step of the trading phase is to ensure that each request is valid.

```

54     int iterations = 10;
55     //make a backup of our current set of requests
56     ArrayList<Request> _requests = new ArrayList<>(requests);
57     while(groupings.size() > 0) {
58         OfferGroup offer = groupings.get(0);
59
60         //remove any requests without remaining draw
61         for(int index = 0; index < requests.size(); index++)

```

```

62         if(requests.get(index).getPowNum() <= 0)
63             requests.remove(index);
64
65         int currentPrice = offer.getSellPrice();
66         Collections.sort(requests);
67
68         //remove any requests priced below the sale price
69         while(requests.size() > 0) {
70             Request req = requests.get(0);
71             if(req.getMaxPricePerKWH() < currentPrice)
72                 requests.remove(req);
73             else
74                 break;
75         }
76
77         //if all requests are exhausted for this set of offers,
78         // we must remove it
79         if(requests.isEmpty()) {
80             groupings.remove(0);
81             continue;
82         }

```

At this point, every request should be valid for the lowest priced group of offers. Power must then be fairly purchased between all modules, with respect to the number of units within each module. If an individual wished to change the way power is distributed to units requesting it, the most convenient place for that change would be here.

```

83         //get a count of all units which request power
84         double filtered_quantity = 0;
85         for(Request r: requests)
86             filtered_quantity += r.getSource().getQuantity();
87
88         //divide amount offered by number of units requesting power
89         double quantity = offer.getPowerRemaing()
90             / filtered_quantity;
91         double rem = offer.getPowerRemaining();
92         boolean sold = false;

```

```

93
94     for(Request req : requests) {
95         double draw = req.getPowNum();
96         //clamp the draw to the maximum amount allowed
97         // per each unit
98         draw = Math.min(quantity*req.getSource().getQuantity(),
99                             draw);
100        req.getSource().buy(draw, currentPrice);
101        offer.sell(draw, currentPrice);
102
103        req.setPowNum(req.getPowNum() - draw);
104
105        rem -= draw;
106        sold = true;
107    }
108
109    //if nothing was sold, the current group is removed
110    if(!sold) {
111        groupings.remove(0);
112        iterations = 10;
113        continue;
114    }
115    //it is likely at this point that tracking
116    // of sales is thwarted by floating point rounding accuracy
117    if(iterations-- == 0) {
118        groupings.remove(0);
119        iterations = 10;
120    }
121 }

```

At this stage, all sales at regular prices for the period will have occurred. The next piece of code attempts to force storage of excess power, where applicable.

```

122     offers = new ArrayList<>(_offers);
123
124     for(int module = 0; module < offers.size(); module++) {
125         if(offers.get(module).getPowAmount > 0) {
126             double stored = offers.get(module).getSource()
127                 .tryStoreRemainder();

```

```

128         double newPowAmount = offers.get(module).getPowAmount();
129         newPowAmount -= stored;
130         offers.get(module).getPowAmount(newPowAmount);
131
132         //if all power was stored, then this module is finished
133         if(offers.get(module).getPowAmount() <= 0)
134             offers.remove(module--);
135     }
136     else
137         //the module has nothing to do regardless
138         offers.remove(module--);
139 }

```

The only modules remaining among the offers are which were unable to sell all of the power they had for sale, and were unable to store any remaining power. The excess presented by these modules will be wasted if it is not utilized. The next step of the process is then to try price-matching down to any remaining users, if possible. A large portion of this code is identical to the method of selling regularly, and hence will be truncated at the point no new information is presented.

```

140     requests = new ArrayList<>(_requests);
141     offers = new ArrayList<>(_offers);
142
143     while(requests.size() > 0) {
144         //remove any requests with no required power
145         for(int request = 0; request < requests.size(); request++){
146             if(requests.get(request).getPowNum() <= 0)
147                 requests.remove(request--);
148         }
149
150         //price matching starts at the highest priced request
151         int highestPrice = 0;
152         Collections.sort(requests);
153         highestPrice = requests.get(requests.size() - 1)
154             .getMaxPricePerKWH();
155
156         //get the list of all offers for which price matching
157         // is enabled

```



```

158     ArrayList<Offer> validOffers = new ArrayList<>();
159     for(int o = 0; o < offers.size(); o++) {
160         if(offers.get(o).getPowAmount() > 0 &&
161             offers.get(o).getSource().willPriceMatchExcess())
162             validOffers.add(offers.get(o));
163     }
164
165     //if there is nothing valid, then nothing can be done
166     if(validOffers.isEmpty())
167         break;
168
169     //create a new OfferGroup based on the matched price
170     for(Offer offer : validOffers)
171         offer.setSellPrice(highestPrice);
172     OfferGroup og = new OfferGroup(validOffers, highestPrice);
173
174     ...
175 }

```

Lastly, all information must be finalized within the network.

```

176     for(SystemComponent sc : systemComponents)
177         sc.finalize();
178     }
179 }

```

At this stage, the function which was titled *calculate2* has been fully defined. For a set number of time-periods, the generation, consumption and trading of power is performed for a network of interconnected units within a microgrid. With the duration set to 8760, and the length of a time period being set to one hour, this function will simulate the conditions of a group of units for a full year.

At every stage of the calculation, data is collected about the condition of each unit: the intended consumption, the generation, the net usage/net generation, the power purchased, sold, given away for free and obtained freely, the average purchase and sale price of power over a period, the amount of power wasted, and the amount of power in deficit. Once all calculations are completed, this data is assembled into a hierarchical structure representing individual periods, individual

days, and 28-day periods (billing cycles).

5 Results

The UoW GESSO tool is capable of allowing for the design, simulation and evaluation of arbitrary microgrids. As a demonstration of the tool, minor data is to be collected for a series of microgrids based on the following set of assertions:

1. The balance of a microgrid can be improved through the addition of 'symbiotic' storage units
2. Symbiotic storage units show diminishing returns
3. Arbitrary microgrids are likely to improve energy balance and reduce grid dependence through mutual co-operation

In order to test the above assertions, a sample set of units must be defined, a set of behaviours must be defined, and test sets must be selected based on the available units and behaviours.

Defined in Table 1 is a set of units that were selected for testing. As a convention, all units that are used in testing are labelled with a number. Units 1 to 4 all simulate units designed to produce Net-ZEB over the course of the year. For each unit, storage is assigned, with the efficiency and effectiveness of the storage increasing from unit 1 through to 4. Unit 5 is a single unit of storage, to be used in testing of the symbiotic storage.

For the presented units, the generation values given are based on the Hamilton, Ruakura 2016 data set. Different data sets were used during actual testing, and each test will mention which data set is used.

A series of usage policies was defined, as shown in Tables 2, 3 and 4. The first policy, labelled X, is intended to represent a unit which undertakes minimal hoarding, and attempts to offload power as quick as possible so long as the battery state is not critical. Usage policy Y is intended to represent a unit which buys power only when the battery state is critically low, and sells only when the battery state is critically high. Usage policy Z is intended to represent a unit which buys power in moderation except for when the battery state is critically low, but sells

Label	Feature	Details
1	Consumption	ID: 12877, 6209 kWh yearly
	Generation	5x 285 W, 345°x 41°, 95.5% inverter efficiency, 6054 kWh yearly
	Storage	Panasonic LJ-SK84A [8.4 kWh] - 86% efficiency, 2kW output
2	Consumption	ID:16106 (x1.1), 4879 kWh yearly
	Generation	4x 275W, 340°x 37°, 94% inverter efficiency, 4843 kWh yearly
	Storage	Tesla Powerwall [6.4 kWh] - 92.5% efficiency, 2kW output
3	Consumption	ID: 12817, 3804 kWh yearly
	Generation	3x 305W, 340°x 40°, 95.5% inverter efficiency, 4171 kWh yearly
	Storage	LG Chem RESU6.5 [6.5 kWh] - 95% efficiency, 4.3kWh output
4	Consumption	ID: 13598, 7769 kWh yearly
	Generation	7x 285W, 315°x 41°, 97% inverter efficiency, 8562 kWh yearly
	Storage	Tesla Powerwall 2 [14 kWh] - 90% efficiency, 5kWh output
5	Storage	Tesla Powerwall 2 [14 kWh] - 90% efficiency, 5kWh output

Table 1: Table of units used in testing. Generation values are based on the Hamilton, Ruakura 2016 dataset.

power only when the battery state is critically high.

Additionally, a set of variables must be defined through which the operator seeks to measure the systems tested. For each variable selected, there are two quantities: the individual value, and the local value. The individual value represents the values obtained from each unit acting solely attached to the grid. The local value represents the values obtained when the unit is acting as part of a mutually co-operative microgrid. Unless otherwise specified, all values given will be based on annual totals.

Consumption The sum of power consumed over the course of the year, in kWh.

The local value considers power sold within the microgrid as a source of consumption.

Generation The sum of usable power generated over the course of the year, in kWh. The local value considers power purchased within the microgrid as a source of generation.

Balance The balance between consumption and generation. The formula for

determining this value is given as $\frac{\text{Generation}-\text{Consumption}}{\text{Generation}+\text{Consumption}}$. Positive values represent dominance of generation, and negative values represent dominance of consumption.

Grid Draw The sum of power offloaded to the state grid over the course of the year.

Grid Offload The sum of power offloaded to the state grid over the course of the year.

Grid Balance The balance between power offloaded to and power drawn from the state grid over the course of the year. The formula for determining this value is given as $\frac{\text{Grid Offload}-\text{Grid Draw}}{\text{Grid Offload}+\text{Grid Draw}}$. If this value is positive, then this unit is offloading more power to the state grid than it is consuming from the grid.

Threshold	Request	Offer
15%	Usage * 1.00	Surplus * 0.50
40%	Usage * 0.50	Surplus * 0.50
60%	Usage * 0.00	Surplus * 0.90
80%	Usage * 0.00	Surplus * 2.00

Table 2: Usage Policy X - Attempt to aggressively offload power, and buy only when necessary

Threshold	Request	Offer
00%	Usage * 1.50	Surplus * 0.20
40%	Usage * 1.10	Surplus * 0.40
45%	Usage * 1.00	Surplus * 0.60
60%	Usage * 1.00	Surplus * 1.00
65%	Usage * 0.50	Surplus * 1.50
75%	Usage * 0.00	Surplus * 2.00

Table 3: Usage Policy Y - Attempt to buy and sell power with a reasonable balance

Threshold	Request	Offer
00%	Usage * 1.20	Surplus * 0.00
20%	Usage * 1.00	Surplus * 0.00
30%	Usage * 0.70	Surplus * 0.00
50%	Usage * 0.50	Surplus * 0.00
60%	Usage * 0.30	Surplus * 0.50
70%	Usage * 0.00	Surplus * 0.80
75%	Usage * 0.00	Surplus * 1.10
90%	Usage * 0.00	Surplus * 2.00

Table 4: Usage Policy Z - Attempt to purchase in moderation, and sell when necessary

5.1 Addition of Symbiotic Storage to a Microgrid

The first two assertions to be tested are that the 'balance' of a microgrid can be improved through the introduction of symbiotic storage units, and that the introduction of multiple storage units may lead to diminishing returns.

Three different tests were carried out in the assessment of these assertions, using the units presented in Table 5. For each test, the following information is to be investigated:

1. Shift in load and balance for individual units
2. Shift in load and balance for entire neighborhood
3. Economic impact on a yearly basis

Test	(Unit,Policy)
Test 1	2Z, 3Y
Test 2	2Z, 3Y, 5
Test 3	2Z, 3Y, 5, 5

Table 5: Configurations for Testing Symbiotic Storage

Given that the economic impact is being assessed, each unit within the system needs to specify the price ranges that it is operating on. For the purposes of this demonstration, all prices will be defined in terms of the grid price, as shown in Table 6. It is assumed that the price of power coming from the grid will remain static across the course of the year. Note that units 2 and 3 will attempt to price match sales, whereas the storage controller (Unit 5) will always buy and sell at exactly the prices specified in the table. Three values are given for the grid sell price, and the economic impact of co-operation is assessed at each of these prices. One feature to note is that because the storage controller does not permit price-matching, the given schema makes it incapable of directly interacting with the state grid.

For all tests under this subsection, the Hamilton, Ruakura 2015 weather data was used. It was assumed that the sale price of power for the state grid is

Unit	Buy	Sell
Unit 2	100	90
Unit 3	100	90
Unit 5	50	90
Grid	100	25,35,45

Table 6: The intended sale and maximum purchase prices set for each unit to be tested.

20c/kWh.

5.1.1 Analysis of a Simple Microgrid

Simulation was performed on the simple microgrid given as Test 1 in Table 5. Graphs are presented of the simulation, with Table 7 showing that the benefit of co-operation in this case is almost negligible. Each unit is likely to be generating excess power at the same time as the other unit, as they both have the same general preconditions for over-generation. Because of this, the majority of trading occurs in very limited intervals, with the period of trading being relative to the difference in orientation between the two solar arrays (37° vs 40°).

Unit	Generation (kWh/year)	Consumption (kWh/year)	Balance	Grid Draw (kWh/year)	Grid Offload (kWh/year)	Grid Balance
2Z	5465 (5515)	5004 (5012)	0.0440(0.0477)	2681(2631)	3019(3011)	0.0594 (0.0673)
3Y	4765 (4773)	3264 (3313)	0.1869(0.1804)	2050(2042)	3491(3442)	0.2600 (0.2552)
*	10231 (10288)	8268 (8326)	0.1060(0.1054)	4730(4673)	6510(6452)	0.1583 (0.1599)

Table 7: Annual summary data for a simple microgrid. Note that the benefit for co-operation in this case is almost negligible.

Table 8 shows that the relative reductions in grid interaction for the neighborhood amount to roughly 1% of individualist total.

Assuming the cost of power to be set at 20c/kWh, and the resale price of power to the state grid to be one of 5 (25%), 7 (35%) or 9(45%) cents per kWh, the annual costs are presented in Table 9. This data shows that at most, the economic impact of co-operation amounts to \$8 per year, taking the net upkeep

Unit	kWh		kWh
	Reduction (fromGrid)	Reduction (toGrid)	Reduction (total)
2Z	50(1.8%)	8(0.3%)	58(1.0%)
3Y	8(0.4%)	50(1.4%)	58(1.0%)
*	58(1.2%)	58(0.9%)	115(1.0%)

Table 8: Annual summary data showing relative impact of co-operation on state grid.

of the system from \$622/year to \$614/year - a reduction of 1.3%.

Unit	Paid	Refunded (5c)	Refunded (7c)	Refunded (9c)
2Z	\$536 (\$535)	\$150 (\$151)	\$211 (\$212)	\$271 (\$271)
3Y	\$410 (\$410)	\$174 (\$180)	\$244 (\$250)	\$314 (\$318)
*	\$946 (\$945)	\$324 (\$331)	\$455 (\$462)	\$585 (\$589)

Table 9: Economic assessment of a simple microgrid. Note that the savings are almost negligible in this case.

Figure 21 shows the hourly, daily and monthly load balances for this microgrid. The primary result of interest is the impact of the change from individual (non-cooperative) to distributed (cooperative) load balances. It can be seen on the figures that the changes from simple cooperation have almost zero impact on the monthly or daily balances. On the hourly trend, it can be seen that there is a minor shift of points towards neutral balance. This implies that even simple non-productive cooperation can help to even out the balance of power on an hour-by-hour basis, regardless of whether it changes balance on a monthly or even daily basis.

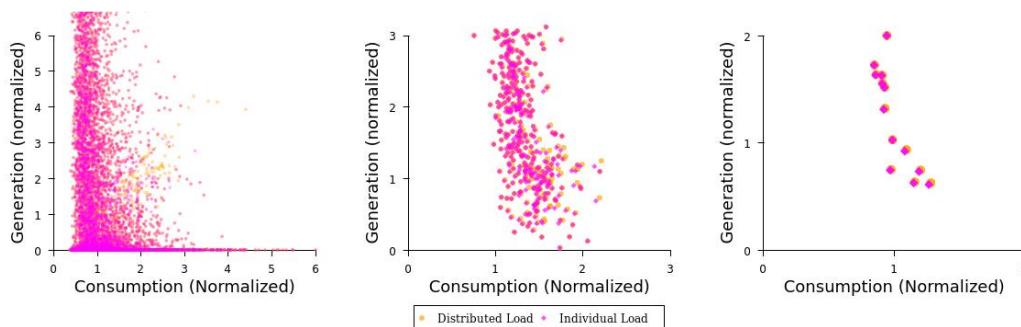


Figure 21: Hourly, daily and monthly load balances for a simple microgrid.

5.1.2 Addition of Symbiotic Storage to a Simple Microgrid

As per Table 5, the simple microgrid was modified with a single unit of symbiotic storage. The unit comprises one Tesla Powerwall 2, which boasts a 14kWh capacity, 90% efficiency, and a 5kW of output power. As can be seen from Table 10, co-operation causes the energy balance of the system to approach zero, and the grid balance of the system to shift further toward production, rather than consumption.

Unit	Generation (kWh/year)	Consumption (kWh/year)	Balance	Grid Draw (kWh/year)	Grid Offload (kWh/year)	Grid Balance
2Z	5465 (6753)	5004 (5969)	0.044(0.0616)	2696(1409)	3070(2105)	0.0646 (0.1979)
3Y	4765 (5872)	3264 (4952)	0.1869(0.085)	2065(958)	3545(1857)	0.2637 (0.3193)
5	0 (2601)	0 (2343)	0(0.0521)	0(0)	0(0)	0(0)
*	10231 (15228)	8268 (13265)	0.1060(0.0688)	4761(2367)	6615(3961)	0.1629 (0.2519)

Table 10: Test of a simple microgrid with symbiotic storage.

It can immediately be seen that the interaction with the grid is decreased significantly for each unit. Total interaction from unit 2Z is reduced by 2.2 mWh (39%), and total interaction for unit 3Y is reduced by 2.8 mWh (50%), giving a total reduction of 5.0 mWh (44%) for the whole system.

Unit	Reduction (fromGrid)	Reduction (toGrid)	Reduction (total)
2Z	1288(47.8%)	965(31.4%)	2253(39.1%)
3Y	1108(53.6%)	1688(47.6%)	2796(49.8%)
*	2396(50.3%)	2653(40.1%)	5049(44.3%)

Table 11: Annual summary data showing relative impact of symbiotic storage unit on state grid.

Annual costs and credits are presented in Table 12, based on energy costs of 20c/kWh, and grid resale prices of 5, 7, and 9 cents per kWh. It can be seen that the addition of the symbiotic storage unit acts to lower the cost paid by the two neighborhood units by a total of \$50 yearly, while providing additional rebates of \$143 (5c), \$100 (7c), or \$38 (9c) per year, for net yearly impacts of \$193, \$150, or \$88 for the system. Including also the value produced by the storage unit itself,

the yearly net impacts become \$355, \$312, or \$250 for the system. This takes the yearly cost of the system down by 52% (5c), 63% (7c), or 69% (9c).

Unit	Paid	Refunded (5c)	Refunded (7c)	Refunded (9c)
2Z	\$536 (\$515)	\$150 (\$202)	\$211 (\$243)	\$271 (\$284)
3Y	\$421 (\$392)	\$174 (\$265)	\$244 (\$302)	\$314 (\$338)
5	\$0 (\$260)	\$0 (\$422)	\$0 (\$422)	\$0 (\$422)
*	\$946 (\$1167)	\$324 (\$889)	\$455 (\$967)	\$585 (\$1044)

Table 12: Economic impact of symbiotic storage at various grid buyback rates.

Figure 22 shows the hourly, daily and monthly load balances for this microgrid. Two sets of diagonal banding on the hourly load balance plot represent the average balances for the two different units of the microgrid. For the hourly and monthly balances, the distributed load is almost entirely separated from the individual load. The general trend is that the addition of local generation/consumption numbers pushing the individual values further into extremity, and the average value of $\frac{\text{generation}}{\text{consumption}}$ closer to 1 (perfect balance).

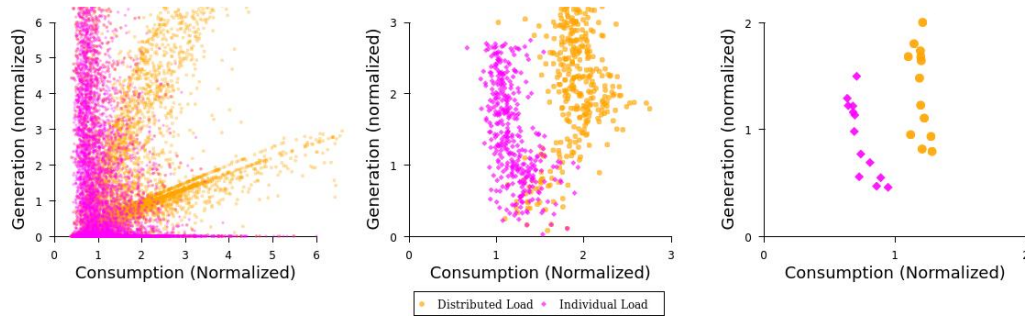


Figure 22: Hourly, daily and monthly load balances for a simple microgrid with a single unit of symbiotic storage

5.1.3 Diminishing Returns with Symbiotic Storage in a Simple Microgrid

For this test, the simple microgrid was modified with the addition of another symbiotic storage unit. The system then contains two Tesla Powerwall 2 storage

units, for a combined throughput of 10kW, and combined storage capacity of 28kWh at 90% efficiency. Within the demonstration, because the two storage units have the exact same specs and prices, they should have the exact same results in every test.

It can be seen that the results for energy balance improve even more than they do with a single unit of symbiotic storage, going from 0.1060 (standard) to 0.0688 (single unit of storage) to 0.0618 (two units of storage), while the grid balance shifts even further from consumer to producer, going from 0.1583 (standard) to 0.2519 (single unit of storage) to 0.3177 (two units of storage) - however, it can be seen that the improvements gained from switching from one unit of storage to two appear to be less significant.

Unit	Generation (kWh/year)	Consumption (kWh/year)	Balance	Grid Draw (kWh/year)	Grid Offload (kWh/year)	Grid Balance
2Z	5465 (7183)	5004 (6447)	0.044(0.0539)	2694(977)	3052(1608)	0.0622 (0.2443)
3Y	4765 (6211)	3264 (5328)	0.1869(0.0764)	2064(617)	3536(1471)	0.2628 (0.4086)
5	0 (1729)	0 (1557)	0(0.0524)	0(0)	0(0)	0(0)
5	0 (1729)	0 (1557)	0(0.0524)	0(0)	0(0)	0(0)
*	10231 (16853)	8268 (14891)	0.106(0.0618)	4756(1594)	6586(3079)	0.1613 (0.3177)

Table 13: Test of a simple microgrid with additional symbiotic storage

In terms of raw interaction with the grid, reductions of 59% can be seen system wide, compared to the 44% system wide reduction that was attained with just a single unit of storage. This means that the first unit of storage reduced interact with the grid by 44% of the total, whereas the second unit of storage reduced the remaining interaction by 25% of the single storage total.

Unit	Reduction (fromGrid)	Reduction (toGrid)	Reduction (total)
2Z	1717(63.7%)	1444(47.3%)	3161(55.0%)
3Y	1446(70.1%)	2065(58.4%)	3511(62.7%)
*	3163(66.5%)	3509(53.2%)	6672(58.8%)

Table 14: Impact of adding second storage unit to simple microgrid on state grid.

Note that the reductions gained from the second unit are much less significant.

Annual costs and credits of the system are presented in Table 15, based on

energy costs of 20c/kWh, and grid resale prices of 5, 7, and 9 cents per kWh. It can be seen that the addition of a second unit of symbiotic storage acts to lower the cost paid by the two neighborhood units by a total of \$56 yearly, while providing additional rebates of \$164 (5c), \$117 (7c), or \$51 (9c), for net yearly impacts of \$220 (5c), \$173 (7c), or \$107 (9c). When the profit gained by the symbiotic storage units is included, the yearly net impacts become \$434, \$387, or \$321 for the system. This takes the yearly cost of the system down by 70% (5c), 79% (7c), or 89% (9c). Within these systems, the yearly value of the storage units amount to \$217 (5c), \$193 (7c), or \$160 (9c) each. At these prices, the repayment period for each battery amounts to between 63 and 88 years. Given a different choice of storage system, such as one with a higher efficiency, or lower price per kWh, these repayment periods could easily be reduced.

Unit	Paid	Refunded (25)	Refunded (35)	Refunded (45)
2Z	\$536 (\$505)	\$150 (\$225)	\$211 (\$259)	\$271 (\$292)
3Y	\$410 (\$385)	\$174 (\$283)	\$244 (\$313)	\$314 (\$344)
5	\$0 (\$173)	\$0 (\$280)	\$0 (\$280)	\$0 (\$280)
5	\$0 (\$173)	\$0 (\$280)	\$0 (\$280)	\$0 (\$280)
*	\$946 (\$1236)	\$324 (\$1069)	\$455 (\$1132)	\$585 (\$1196)

Table 15: Economic impact of adding a second symbiotic storage at various grid buyback rates. Note that the under ideal conditions, the total cost of the system is reduced to \$40/year

Figure 23 shows the hourly, daily and monthly load balances for this microgrid. Compared to Figure 22, it can be seen that the banding for each unit on the hourly scale shifts closer to the diagonal, and though it may not be visually apparent from the daily and monthly graphs, the shift in balance moves further toward parity.

5.2 Impact of co-operation in homogeneous units

One assertion that was marked for testing is that, in the general case, arbitrary microgrids are likely to improve performance in relation to the grid through co-

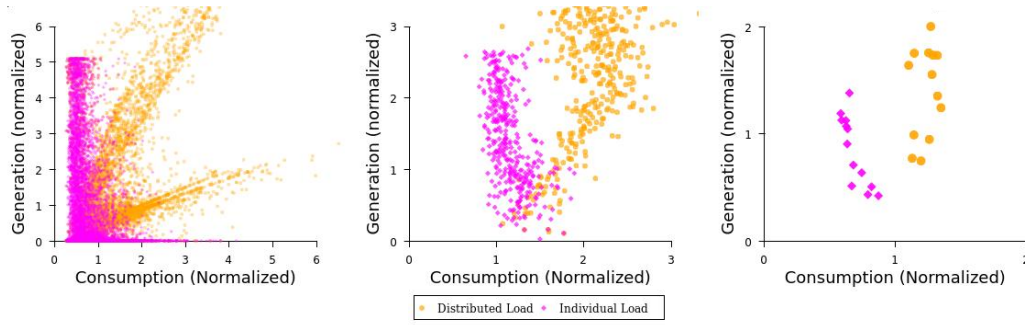


Figure 23: Hourly, daily and monthly load balances for a simple microgrid with a single unit of symbiotic storage

operation. It is expected that in no case will any microgrid be more taxing on the state grid than the collection of its sub-components are individually. The ideal scenario is that for all neighborhoods, the energy balance shifts closer to zero, and the relationship with the grid shifts further towards being a prosumer relationship.

In order to test this assertion, a series of microgrids was generated for testing, as is show in Table 16. One expectation is that larger microgrids have a larger capacity to reduce interaction with the microgrid, but that the individual impact of each unit added will decrease in respect to the size of the system.

Test	(Unit,Policy)
Test 1	1Z, 2X, 3X, 3Y, 4Y, 4X
Test 2	1Y, 1Z, 2Y, 3Y, 4X
Test 3	2X, 3X, 3Y, 4Y

Table 16: Selected homogeneous microgrids to be tested. Each unit within the microgrid has generation, consumption, and storage.

5.2.1 Experimental Results

The first test, shown in Table 17 shows that for the system as a whole, the grid balance improves marginally, whereas the energy balance shifts towards zero. In terms of raw numbers, the total grid interaction of the system is reduced by 2.1%

(1 mWh) over the course of the year. The greatest individual performance was seen by unit 3Y, where consumption of power from the grid was reduced by 5.7%, and consumption of power by the grid was reduced by 3.4%.

Unit	Generation (kWh/year)	Consumption (kWh/year)	Balance	Grid Draw (kWh/year)	Grid Offload (kWh/year)	Grid Balance
1Z	6189 (6233)	5108 (5170)	0.0957(0.0932)	2969(2925)	3825(3762)	0.1259 (0.1252)
2X	4956 (5056)	5004 (5061)	-0.0048(-0.0004)	2971(2871)	2837(2780)	-0.0231 (-0.0161)
3X	4271 (4298)	3264 (3322)	0.1337(0.128)	1529.213(1502)	2462(2404)	0.2337 (0.2307)
3Y	4271 (4390)	3264 (3368)	0.1337(0.1317)	2078(1960)	3034(2930)	0.1869 (0.1983)
4X	8765 (8937)	8726 (8897)	0.0022 (0.0022)	6542 (6370)	6507 (6335)	-0.0026 (-0.0027)
4Y	8765 (8846)	8725 (8814)	0.0022(0.0017)	5242(5161)	5034(4945)	-0.0202 (-0.0213)
*	37220 (37702)	34092 (34574)	0.0438(0.0432)	21391(20909)	23693(23211)	0.051 (0.0521)

Table 17: Annual data for a large microgrid

For every unit involved in the neighborhood, the energy balance shifted closer to zero. For four of the six units, the grid balance as a whole either shifted towards zero, or shifted towards 1. The exceptions to this are units 4X and 4Y, which show marginal decreases in grid balance toward -1. This means that from the perspective of the grid, these units shift further toward a consumer role. Inspecting Table 18, it can be seen that these units have roughly equal or greater reduction in exports to the grid than they do imports from the grid, shifting the calculated balance figure toward the negative end of the scale. This may imply that using only the balances as a metric in the planning of a system may lead to scenarios where beneficial results may be ignored because they produce unsatisfactory ratios.

Unit	Reduction (fromGrid)	Reduction (toGrid)	Reduction (total)
1Z	44(1.5%)	62(1.6%)	106(1.6%)
2X	100(3.4%)	57(2%)	157(2.7%)
3X	27(1.8%)	59(2.4%)	86(2.1%)
3Y	118(5.7%)	104(3.4%)	222(4.3%)
4X	81(1.5%)	89(1.8%)	170(1.7%)
4Y	112(1.7%)	111(1.7%)	223(1.7%)
*	482(2.3%)	482(2%)	964(2.1%)

Table 18: Reduction in grid interaction for a large microgrid

One noticeable feature of this test is that there are two different varieties of unit 3 and unit 4, with use of policies X and Y. For reference, policy X directed the unit to offload power at any point there was surplus generation, and to buy power moderately when the battery capacity was lower than 60%. Policy Y directed the unit be buying power whenever it was needed so long as the capacity of the battery was less than 75%, while selling power in proportion to battery capacity at all stages. During the design, Y was considered to be more balanced, and X to be more aggressive. Interestingly, the Y type policies showed increased sale of power to the grid, as well as increased purchase of power from the grid. One explanation may be that the Y type policy is actually poorly designed, with a large bottleneck around 60% battery capacity where interaction with the grid is guaranteed regardless of weather. By comparison, the sales from the X type policy actually end up being less aggressive, with the battery capacity remaining at lower states on average due to buying power less aggressively.

Test 2, shown in Table 19, shows a slightly smaller system. Similar to the first test, every unit shows energy balance shift towards 1, or towards 0. None of the units shows a shift of energy balance from a negative number towards -1. The neighborhood itself shows the balance shift toward 0. In all five of the units the grid balance either shifts towards zero or shifts towards 1. As a whole, the neighborhood shifted further toward neutral with energy balance, and further towards being a producer under grid balance.

This system also provides the potential to compare the Y type and Z type

Unit	Generation (kWh/year)	Consumption (kWh/year)	Balance	Grid Draw (kWh/year)	Grid Offload (kWh/year)	Grid Balance
1Y	6189 (6249)	5108 (5296)	0.0957(0.0825)	4004(3944)	4960(4772)	0.1065 (0.0948)
1Z	6189 (6212)	5108 (5173)	0.0957(0.0912)	2979(2956)	3812(3747)	0.1227 (0.1179)
2Y	4956 (5177)	5004 (5083)	-0.0048(0.0091)	3519(3297)	3382(3302)	-0.0198 (0.0007)
3Y	4271 (4376)	3264 (3377)	0.1337(0.1287)	2078(1974)	3027(2913)	0.1858 (0.1922)
4X	8765 (8968)	8725 (8890)	0.0022(0.0043)	5244(5040)	5026(4860)	-0.0212 (-0.0181)
*	30373 (30984)	27210 (27822)	0.0549(0.0537)	17824(17212)	20207(19595)	0.0626 (0.0647)

Table 19: Annual data for a medium sized microgrid

policies on Unit 1. It can be seen that the Y type unit, which was previously established as potentially poorly designed, once again showed significantly higher quantities of power purchased and sold. Having a higher representation among the units buying and selling power, it also gained more out of cooperation than the more efficient Z type variant.

Unit	Reduction (fromGrid)	Reduction (toGrid)	Reduction (total)
1Y	60(1.5%)	188(3.8%)	248(2.8%)
1Z	23(0.8%)	65(1.7%)	88(1.3%)
2Y	221(6.3%)	80(2.4%)	301(4.4%)
3Y	104(5.0%)	114(3.8%)	218(4.3%)
4X	203(3.9%)	165(3.3%)	369(3.6%)
*	612(3.4%)	612(3.0%)	1223(3.2%)

Table 20: Reduction in grid interaction for a medium sized microgrid

Test 3 shows the same general trends as the first two tests. First, the energy balance as improved for every involved unit and for the system as a whole, and second the grid balance has improved for most of the participating units, and for the grid as a whole. The notable exception to the trend is that unit 4Y shows a decay in grid balance, from -0.0075 to -0.0082. Examining table 22, it can be seen that 4Y shows higher reductions in power offloaded to the grid than power taken from the grid, and because the unit already had a negative grid balance, this served to further reduce the reported balance.

Additionally, this system serves as another example of the Y type policy being sub-optimal, with the Y variant of unit 3 showing much higher rates or transaction than the X variant, while still gaining a greater benefit from co-operation than

Unit	Generation (kWh/year)	Consumption (kWh/year)	Balance	Grid Draw (kWh/year)	Grid Offload (kWh/year)	Grid Balance
2X	4956 (5054)	5004 (5058)	-0.0048(-0.0004)	2958(2860)	2823(2769)	-0.0233 (-0.0162)
3X	4271 (4295)	3264 (3320)	0.1337(0.128)	1532(1509)	2472(2416)	0.2346 (0.231)
3Y	4271 (4369)	3264 (3367)	0.1337(0.1296)	2079(1981)	3039(2936)	0.1875 (0.1942)
4Y	8765 (8920)	8725 (8886)	0.0022(0.0019)	6596(6441)	6497(6336)	-0.0075 (-0.0082)
*	22265 (22640)	20258 (20632)	0.0472(0.0463)	13166(12791)	14831(14457)	0.0594 (0.0611)

Table 21: Annual data for a small microgrid

the X type variant.

Unit	Reduction (fromGrid)	Reduction (toGrid)	Reduction (total)
2X	98(3.3%)	55(1.9%)	153(2.6%)
3X	23(1.5%)	56(2.3%)	80(2%)
3Y	98(4.7%)	103(3.4%)	201(3.9%)
4Y	155(2.4%)	161(2.5%)	316(2.4%)
*	374(2.8%)	374(2.5%)	749(2.7%)

Table 22: Reduction in grid interaction for a small microgrid

In general, the three systems showed reductions of between 2% and 3.2% in total interaction with the state grid when engaged in a co-operative microgrid. The primary bottlenecks to the systems presented stem from the fact that each unit relies on the same conditions to generate power, and are all likely to be generating power at the same time. This means that when any of the units in these systems are in surplus, it is likely that the other units are also in surplus. This severely limits the potential to trade power under the conditions presented. The second bottleneck occurs in the planning of smart usage policies. The tests above demonstrated that policies X and Z may prove to be reasonable for a system, but that the Y type policy caused more extreme draw and offload of power. Beyond just the planning of these types of policies, it may be that a different type of control strategy may provide a greater benefit. One example may be a strategy which takes into account day/night cycles, and decides to buy or sell power based on those numbers. One potential example may be an attempt to offload stored power prior to sunrise, in anticipation of the storage capacity being exhausted during the following day.

6 Conclusions

In this thesis, a model has been presented for the aggregation of microgrids, a classification of several different types of microgrid components has been given, and a methodology for the design and simulation of various smart microgrid systems has been identified. A sample has been produced, demonstrating the ability to design, simulate, and test a range of microgrid systems. In particular, sample data was presented demonstrating the potential impact of symbiotic storage units on co-operative microgrids.

A derived definition is given for a microgrid, providing scaling with the magnitude of the microgrid. This definition is given as "A localized collection of energy sources and sinks and/or interior microgrids, which share a single central connection to the next level macrogrid, which has the capacity of operating in island-mode." Using this definition, a model is given for a microgrid within the context of a simulation: some combination of storage, generation, consumption, or other microgrids. Logically, since neither transmission loss over distance, or the routing of power through multiple sub-networks, are not features of the simulation, then any arbitrary microgrid can be flattened into a single simple set of units than can be arbitrarily simulated as if they were a single microgrid.

The concept of behavior is defined on a per unit basis, with each individual unit able to specify individual patterns of behavior. The types of behavior definable vary based on the classification of each unit: for example, a unit persisting only of consumption being unable to sell, and only being able to control purchasing habits. As a general rule, the higher the amount of features a unit has, the higher the ability of that unit to determine the actions it can take.

As part of the simulation, the amount of solar radiation generated by each unit (containing solar) is determined based on the quantity and orientation of the solar panels, as well as user selected weather data. The current data processed and available for the tool allows for uniformly accurate generation across most of New Zealand.

A series of tests were performed with the tool, investigating the effectiveness of storage units action in symbiosis to a small microgrid, and the general properties of co-operation that could be attained through mutual cooperation.

It was found that a simple microgrid, containing two units with a shared yearly load of 8683 kWh, and expected generation of 9014 kWh was able to reduce yearly costs by \$250 - \$355 per year (based on sale costs of 20c/kWh, and resale costs of 5/9c per kWh) through the inclusion of a single unit of storage. The addition of a second storage unit led to a reduction in the value per unit of the storage within the neighborhood. It is likely that this type of system, with natural cycles of generation and consumption, has varying limits to saturation - in general, any storage *not* being used is wasted - and the system only has the capacity to regularly use storage equal to the limits of each individual unit. A potential future line of research may focus on determining ideal saturation of storage for particular neighborhoods or microgrids.

Tests of mutual cooperation on homogeneous microgrids which relied on solar generation presented less than ideal results: with all involved units sharing the same expected times of generation, the provided model provides little opportunity for meaningful trades. Notably, certain neighborhood were able to reduce consumption of power by up to 3.6%. As was demonstrated in the section on symbiotic microgrids, it is assumed that the systems which create the most benefit are those of mixed capability; overproducing power should ideally be met by households within the same microgrid which don't generate power at all, or units which lease storage capacity. A potential future line of research could involve generating ideal saturation of prosumer and consumer units for large scale neighborhoods, or potentially ideal ratios of improvements to existing neighborhoods in order to provide the greatest overall return on investment over time.

References

- [1] Transpower NZ, “Transmission tomorrow – our strategy.”
<https://www.transpower.co.nz/sites/default/files/publications/resources/TTourstrategy2018.pdf>, 2018.
- [2] Berkeley Lab Grid Integration Group, “About microgrids.”
<https://building-microgrid.lbl.gov/about-microgrids> (Dec 2018),
archive.is/AR1JE.
- [3] United States Department of Energy, “The U.S. department of energy’s microgrid initiative.”
www.energy.gov/sites/prod/files/2016/06/f32/The%20US%20Department%20of%20Energy%27s%20Microgrid%20Initiative.pdf
(Dec 2018), archive.is/XqcXp.
- [4] H. Fraker, *The Hidden Potential of Sustainable Neighborhoods*.
- [5] Mark Apperley, “Modelling energy balance and storage in the design of smart microgrids,” *ENERGY 2017 : The Seventh International Conference on Smart Grids, Green Communications and IT Energy-aware Technologies*, 2017.
- [6] NZ National Infrastructure Unit, “The thirty year new zealand infrastructure plan 2015,” 2015.
- [7] Lance Ulanoff, “Elon Musk and SolarCity unveil ‘world’s most efficient’ solar panel.” <https://mashable.com/2015/10/02/elon-musk-solarcity-new-solar-panel/#b0gzJiVjLiq2> (Jan 2019), <http://archive.is/Yr4zb>.
- [8] California Energy Commission, “California PV module list.”
https://www.gosolarcalifornia.ca.gov/equipment/pv_modules.php
(Jan 2019), <http://archive.is/OboiN>.

- [9] Mark Apperley, Paul Monigatti and Joris Suppers, “A network integrated semi-autonomous local area electricity system,” *ICGREEN 2015: Proceedings of the Fourth International Conference on Green IT Solutions*, 2015.
- [10] S. Matasci, “How much does a solar panel installation cost?.” <https://news.energysage.com/how-much-does-the-average-solar-panel-installation-cost-in-the-u-s/> (Aug 2019), <http://archive.is/qSWQ5>.
- [11] U.S. Energy Information Administration, “U.S. Battery Storage Market Trends.” https://www.eia.gov/analysis/studies/electricity/batterystorage/pdf/battery_storage.pdf (Aug 2019) <http://archive.is/mNlyg>, 2018.
- [12] F. Lambert, “Tesla powerwall 2 has no competition – comparison with lg resu and sonnen batteries.” <https://electrek.co/2016/10/31/tesla-powerwall-2-comparison-lg-resu-sonnenbatterie/> (Sep 2019), 2016.
- [13] EnergySage, “How much does solar storage cost? understanding solar battery prices.” <https://www.energysage.com/solar/solar-energy-storage/what-do-solar-batteries-cost/> (Sep 2019) <http://archive.is/hyg1H>, 2019.
- [14] B. S. Kiti Suomalainen, Vincent Wang, “Energy centre’s auckland solar power online tool documentation,” 2018.
- [15] NIWA, “Oh solar me-o: Niwa gives you something to sing about!.” <https://www.niwa.co.nz/news/oh-solar-me-o-niwa-gives-you-something-sing-about> (Apr 2019) <http://archive.is/oCSHQ>.

- [16] J. Suppers and M. Apperley, “Interactive solar panel simulation tool-from ghi to pv output,” in *FSEA 2015*, pp. 633–642, University of Bamberg Press, 2017.
- [17] S. Labs, “Sandia Labs PV Performance Modeling Collaborative: Modeling steps.” <https://pvpmc.sandia.gov/modeling-steps/> (Feb 2019), <http://archive.is/DRBE9>, 2019.
- [18] R. Ford, M. McCulloch, T. Helfer, and S. Surrall, “Suricatta: a platform to model smart grid technologies in the distribution system,” in *IEEE PES ISGT Europe 2013*, pp. 1–5, IEEE, 2013.
- [19] Brian Publicover, “Japan: Kumamoto electric pushes pv-powered cryptocurrency unit as potential curtailment solution.” <https://www.pv-magazine.com/2018/03/05/kumamoto-electric-pushes-pv-powered-cryptocurrency-unit-as-potential-curtailment-solution/> (Feb 2019), <http://archive.is/TzGee>.
- [20] Google, “Gwt project.” <http://www.gwtproject.org/terms.html> (Jan 2019), <http://archive.is/fyhxx>, 2018.
- [21] Apache, “Apache tomcat.” <http://tomcat.apache.org/> (Feb 2019), <http://archive.is/jYEmY>.
- [22] Apache, “Apache derby.” <https://db.apache.org/derby/> (Feb 2019), <http://archive.is/2ZFdH>.
- [23] N. Provos and D. Mazieres, “A future-adaptable password scheme.” in *USENIX Annual Technical Conference, FREENIX Track*, pp. 81–91, 1999.
- [24] University of Waikato Department of Computer Science, “Energy informatics.”
- [25] Trevor Berrill, Andrew Blair, *SWH Training Course - Installer User Manual, Ch. 2*. Australian Business Council for Sustainable Energy.

Appendices

A Single unit simulation graphs

Supplementary graphs are presented in Figures 24, 25, 26, 27, 27, 29. These are the daily and hourly versions of figures 17, 18 and 19.

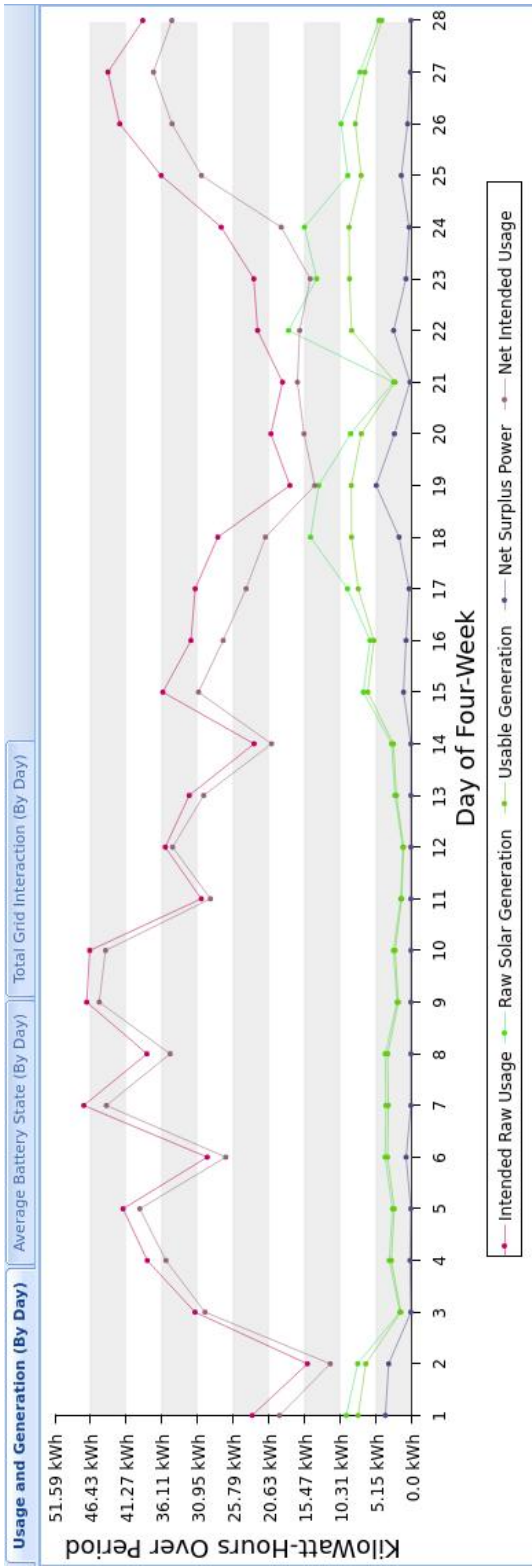


Figure 24: Generation and Consumption values averaged by 24-hour period.

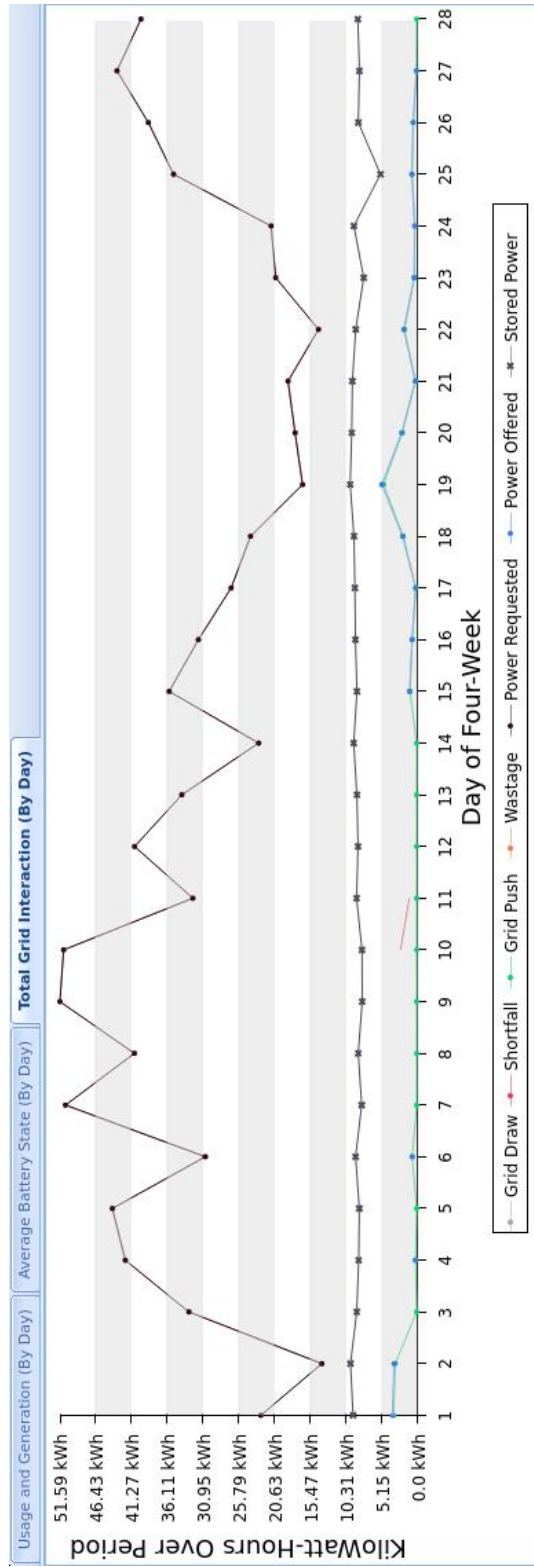


Figure 25: Grid Interaction values averaged by 24-hour period.

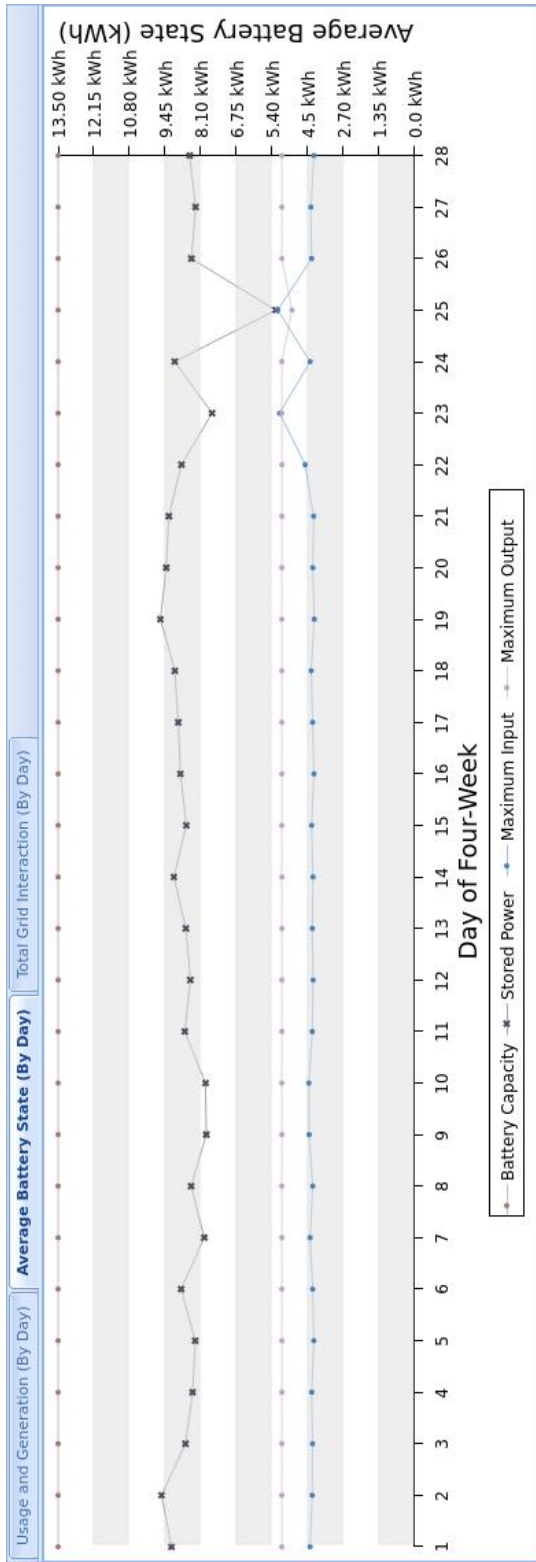


Figure 26: Battery state values averaged by 24-hour period.

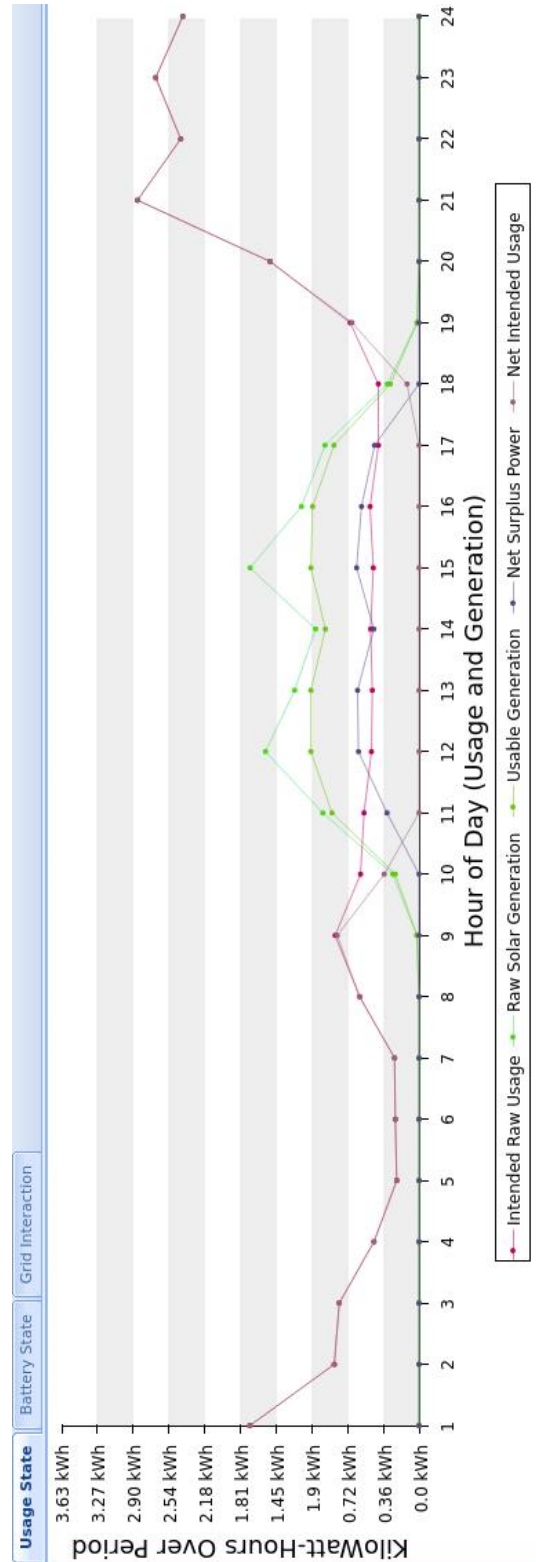


Figure 27: Generation and Consumption values per hour.

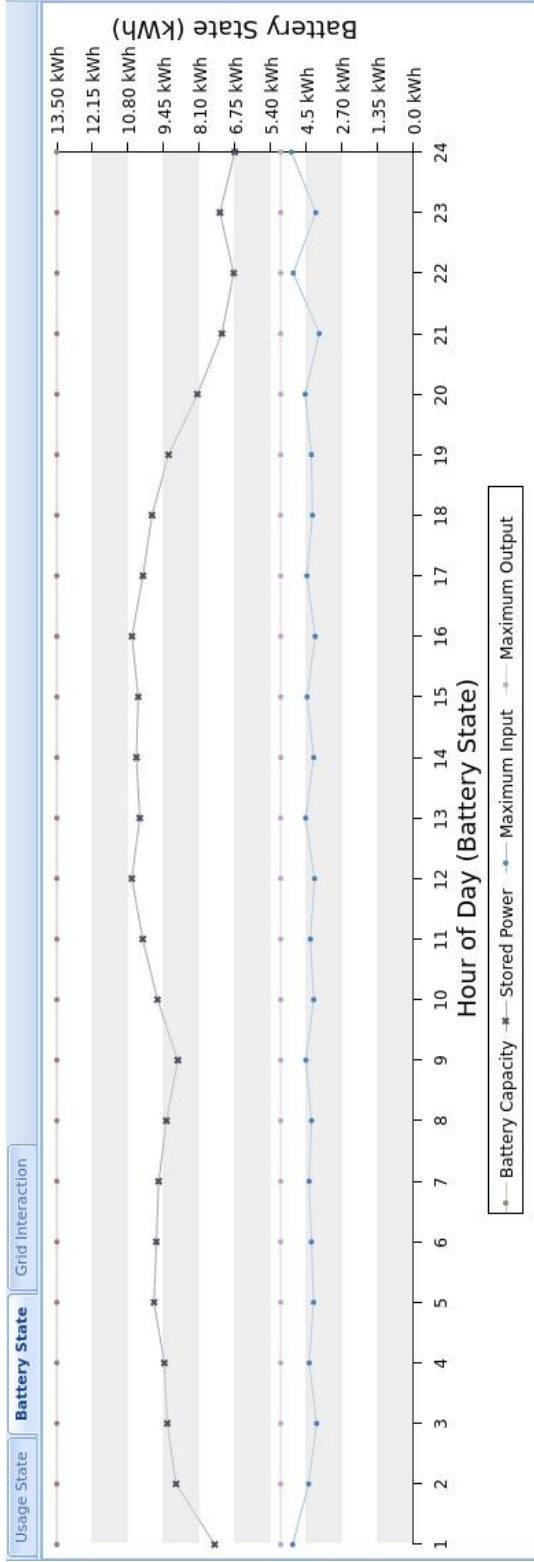


Figure 28: Battery state values by hour.

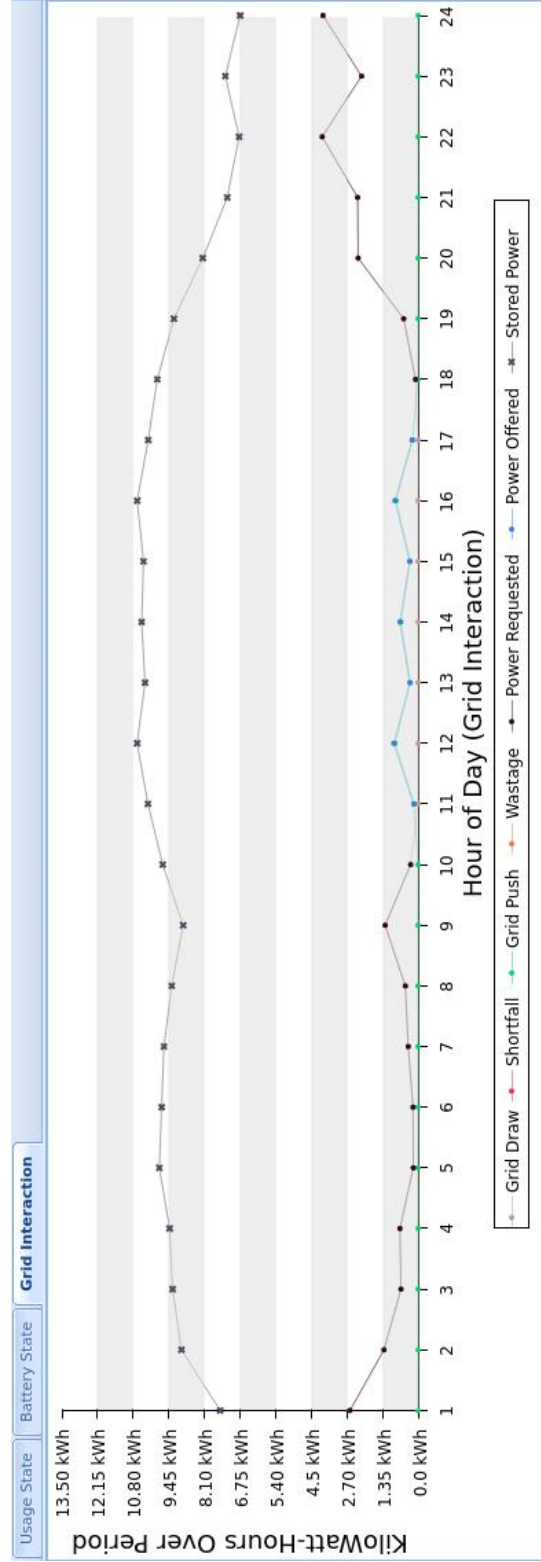


Figure 29: Grid Interaction values by hour.