TRINITY COLLEGE DUBLIN

Minimising CO_2 Emissions Produced from Water Heater Usage

by

David Kelly

Supervisor: Siobhán Clarke

A thesis submitted in partial fulfillment for the
degree of Masters of Computer Science

in the
School of Computer Science and Statistics

Submitted to the University of Dublin, Trinity College, May, 2017

Declaration of Authorship

I, David Kelly, declare that the following dissertation, except where otherwise stated, is entirely my own work; that it has not previously been submitted as an exercise for a degree, either in Trinity College Dublin, or in any other University; and that the library may lend or copy it or any part thereof on request.

Signed: _____

Date: _____

Summary

The electricity supply is produced from a mixture of different sources. These sources can be categorised as renewable and non-renewable. Renewable sources, such as wind and solar produce no CO_2 emissions, but they are weather dependent. Non-renewable sources, such as coal and gas, are stable, but they do produce CO_2 emissions. European Union (EU) countries are required to reduce CO_2 emissions in order to comply with EU policy. This is achieved by increasing the use of renewable energy. The weather dependent nature of renewable sources means that the proportion of renewable and non-renewable energy in the electricity supply is constantly changing. This means that the CO_2 emissions produced by the electricity supply will vary over time.

The pattern of electricity consumption needs to dynamically adapt to enable the minimisation of CO_2 emissions. This dissertation focusses on water heaters, which have two sources of heat energy, an electrical element and a gas-fired space heating system. Existing research in this area has examined energy consumption of water heaters which only have a single source of heat energy; an electrical element. A water heater can store energy which gives it a flexible consumption pattern. This means that a water heater can consume energy during periods when a low CO_2 emission source becomes available.

This dissertation focusses on the implementation of a controller which manages the operation of the water heater. The goal of the controller is to provide utility to the end user when required, while minimising the CO_2 emissions produced by the energy it consumes. Two types of controllers are investigated. The first controller is referred to as the expert controller. The expert controller performs actions using a known, good control policy. The second controller employs Q-learning, a reinforcement learning algorithm. The Q-learning controller learns a near optimal control policy by exploring as many different state-action combinations as possible. The Q-learning controller learns its policy through receiving positive rewards for good actions, while bad actions result in negative rewards.

A set of experiments were carried out with the aim of determining whether the Q-learning controller could learn a control policy that resulted in superior performance compared to the expert controller. These experiments were carried out through the use of a simulation framework, GridLAB-D . This dissertation describes the implementation of the two controllers in the GridLAB-D simulation framework.

The results of the experiments show that the Q-learning controller performs similarly to the expert controller. At times, the Q-learning controller produces a lower amount of CO_2 emissions in order to meet its heating requirements. However, this also results in a loss of utility to the end user.

Abstract

Masters in Computer Science

Minimising CO_2 Emissions Produced from Water Heater Usage

by David Kelly

2017

Supervisor: Siobhán Clarke

The electricity supply is produced from renewable and non-renewable energy sources. Renewable sources, such as wind and solar produce no CO_2 emissions, but they are weather dependent. Non-renewable sources, such as coal and gas, are stable, but they do produce CO_2 emissions. European Union (EU) countries are required to reduce CO_2 emissions in order to comply with EU policy. This is achieved by increasing the use of renewable energy. The weather dependent nature of renewable sources means that the proportion of renewable and non-renewable energy in the electricity supply is constantly changing. This means that the CO_2 emissions produced by the electricity supply will vary over time.

The pattern of electricity consumption needs to dynamically adapt to enable the minimisation of CO_2 emissions. This dissertation focusses on water heaters, which have two sources of heat energy, an electrical element and a gas-fired space heating system. Existing research in this area has examined energy consumption of water heaters which only have a single source of heat energy; an electrical element. A water heater can store energy which gives it a flexible consumption pattern. This means that a water heater can consume energy during periods when a low CO_2 emission source becomes available.

This dissertation investigates the implementation of two different controllers which manage the operation of the water heater. The first controller, the expert, implements a known, good control policy. The second controller employs Q-learning. A set of experiments were carried out in order to determine if the Q-learning controller could achieve better performance than the expert controller. The results of the experiments show that the Q-learning controller could produce lower levels of CO_2 emissions. However this resulted in a loss of utility to the end user.

Acknowledgements

I would like to thank my academic supervisor, Dr. Siobhán Clarke for providing me with her support and guidance throughout the course of my dissertation.

I would also like to thank Alan McKenna and Ivor Roddy for their helpful assistance in discussing the idea for this project.

I would like to thank Dr. Adam Taylor for providing very helpful advice, especially with regard to GridLAB-D and the implementation of a Q-learning agent.

To my dear friend Joseph Mercier, thank you for your excellent feedback on this dissertation. It also made for some entertaining reading.

I must also thank my parents for their support throughout this project.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

This goal of this dissertation is to minimise the CO_2 emissions that are produced from water heater usage. The fundamental way in which this is achieved is through shifting the energy consumption pattern of a water heater to periods when low CO_2 energy becomes available.

1.1 Background

The electricity supply is produced from two different categories of energy sources. These two categories are renewable and non-renewable energy. Renewable sources, such as wind and solar produce no CO_2 emissions, but they are weather dependent. Non-renewable sources, such as coal and gas, are stable sources of energy but they do produce CO_2 emissions. The weather dependent nature of renewable sources means that levels of renewable and non-renewable energy generation is constantly changing. This means that the CO_2 emissions produced by electrical supply generation is also constantly changing.

Based on these characteristics of electricity supply generation, it is clear that the pattern of energy consumption needs to coincide with periods of high renewable energy availability. Devices with a flexible electricity consumption pattern (eg. electric vehicles, water heaters, etc.) are ideal for adapting the pattern of energy consumption.

1.2 Motivation

In an effort to reduce CO_2 emissions, the European Union (EU) has proposed a set of reduction targets to reach by 2050. The EU 2050 roadmap (European Commission, 2011) sets a target to achieve at least an 80% reduction in CO_2 emissions by 2050. Increasing the use of renewable energy will contribute to achieving these targets. It is clear that EU member countries will be required to continue lowering their carbon emissions in the coming years.

1.3 Dissertation Objectives

This dissertation investigates two types of controllers which are designed to manage the operation of the water heater. The first controller is referred to as the expert. The expert controller manages the operation of the water heater based on a known, good control policy. The second controller employs Q-learning, a reinforcement learning algorithm in order to learn a near optimal control policy. This dissertation investigates whether the performance of the Q-learning controller can exceed the performance of the expert controller.

1.4 Dissertation Overview

The first chapter, *Introduction*, introduces the goal and background of this dissertation.

The second chapter, *State of the Art*, discusses the existing research related to the topic of this dissertation. It also identifies areas that this dissertation builds on.

The third chapter, *Design*, outlines an overview of the type of water heater that this dissertation focusses on. It also outlines the operation of two different water heater controllers.

The fourth chapter, *Implementation*, describes the implementation of the water heater and its controllers which are outlined in the third chapter. *Note: Code implementations can be found in appendix B*

The fifth chapter, *Experimental Procedure*, outlines the experiments that were carried out in order to evaluate the performance of the two water heater controllers which are described in the third chapter.

The sixth and final chapter, *Conclusion*, discusses the conclusions made based on the results from the fifth chapter.

Chapter 2

State of the Art

This chapter examines work related to this dissertation. Related works have examined shifting the pattern of electricity consumption in order to achieve a particular goal. The shifting of electricity consumption is commonly referred to as *Demand Response* in the electricity industry.

This dissertation examines shifting electrical consumption to periods when the CO_2 emissions produced from the electrical supply are lower. The CO_2 emissions produced from the electrical supply are lower at times when renewable energy forms a large part of the supply. Related work has examined the shifting of electrical consumption in order to achieve other goals such as minimising the total cost of energy consumption or maximising the use of renewable energy.

2.1 Adapting Electrical Consumption Pattern

The shifting of electricity consumption patterns is a topic that is of interest to both academia and industry. Existing research has examined this activity with the aim of achieving a particular goal. This goal may be to minimise the cost of electricity consumption, reduce peak in electrical demand, maximise renewable energy use or respond to changes in electrical frequency. The following section examines various goals of adapting electrical consumption in existing research.

2.1.1 Demand Response

The shifting of electrical consumption in response to a changing price profile is referred to as *Demand Response*. Electrical grid operators have always focussed on modifying the level of electrical supply on the grid. In recent years, managing the level of demand on the grid has become increasingly important for grid operators. Today, grid operators can determine incentives for consumers to adjust their demand upon request. In Albadi and El-Saadany (2007), Demand Response is discussed in detail with a particular focus on the different mechanisms for price incentives.

2.1.2 Minimise Financial Cost

In O'Neill et al. (2010), Ruelens et al. (2016a), the use of reinforcement learning algorithms are employed with the goal of minimising the financial cost of electricity consumption to the end user. In O'Neill et al. (2010), the electricity consumption pattern of general devices was successfully adapted to avoid peaks in electricity pricing. Peaks in electrical demand were avoided by delaying the energy consumption of a device while also minimising the dis-utility to the end user. In Ruelens et al. (2016a), the electrical consumption of a water heater was shifted to periods of lower prices. Two different pricing profiles were examined; day-ahead pricing and imbalance pricing. The proposed system for minimising cost was shown to be successful in both a simulation experiment and a lab experiment.

2.1.3 Reduce Peak Energy Consumption

In Dusparic et al. (2013), the consumption pattern of electric vehicle charging is adapted with the goal of reducing peak energy demand. A collection of 9 residential households with varying base loads, were examined. Each of these households had a reinforcement learning agent controlled electric vehicle charging station. The agent in each household was designed to avoid overloading a local transformer which was connected to all of the households. Each agent was provided with information on current load as well as the predicted day ahead load. The agent successfully learned to charge the electric vehicle during periods of low demand. The agent was also designed so that current load information could be interchanged directly with a pricing signal.

2.1.4 Frequency Response

The electricity grid operator is constantly working to balance the electrical supply with the current demand. Failure to maintain this balance results in negative consequences for equipment connected to the grid. The balance of electrical supply and demand can be observed by measuring the frequency of the electrical signal on the grid. The electrical grid operator in Ireland, Eirgrid, works to maintain the desired frequency of 50Hz Eirgrid (2015a). A rise in frequency indicates that the electrical supply is greater than demand. A decrease in frequency indicates that electrical demand is greater than supply. In order to maintain the desired frequency of 50Hz, electricity consumers can make small changes to their electrical consumption in order to correct fluctuations in electrical frequency.

In Short et al. (2007), the use of dynamic demand control of consumer appliances is investigated as a mechanism for providing frequency stability. Refrigeration and heating devices were identified as suitable appliances for providing frequency response. For example, freezers operate throughout the entire year at all times of the day. Such devices are always available to participate in adjusting electricity demand. Dynamically controlling the demand of a number of refrigeration devices has been shown to successfully smooth out fluctuations in frequency. It also proves effective in smoothing frequency during periods when renewable wind energy supply fluctuates.

2.1.5 Maximise Renewable Energy

The proportion of renewable energy in the electricity supply is constantly changing. This is because of the largely weather dependent nature of renewable energy. In Dusparic et al. (2015), the electricity consumption pattern of a collection of households is adjusted. A distributed learning algorithm was used to control an electric vehicle charger in each household. The control agent was encouraged to avoid causing the charging load to exceed the total available renewable energy. Each control agent was highly rewarded for reaching a state where the electric vehicle battery charge reached the minimum requirement. The control agent is also encouraged to avoid charging the electric vehicle during periods of peak demand.

2.1.6 Analysis

Existing research has examined the shifting of electricity consumption for the following goals:

- Minimising the cost of energy consumption (O'Neill et al., 2010, Ruelens et al., 2016a).
- Reducing peaks in demand (Dusparic et al., 2013).
- Responding to changes in electrical frequency (Short et al., 2007).
- Maximising the use of renewable energy (Dusparic et al., 2015).

This dissertation examines the minimisation of CO_2 emissions through dynamically adapting the pattern of electricity consumption. As of the writing of this dissertation no other work in this area has examined this goal.

2.2 Renewable Energy Use

Existing research has examined shifting electricity consumption to times when renewable energy is available. Renewable energy sources such as wind and solar are weather dependent which means that the amount of available renewable energy varies over time. Existing research has examined directly shifting consumption to times when renewable energy is available. Currently, electrical grid operators are required to limit the amount of renewable energy generation. The following section discusses these topics in more detail.

2.2.1 Renewable Energy Curtailment

Electrical grid operators cannot always accommodate renewable energy to its maximum available level of power generation. The electrical grid operator in Ireland, Eirgrid, is required to limit renewable energy generation once it exceeds a safe proportion of the total energy supply. This limiting of renewable energy is referred to as *Renewable Curtailment*. Renewable energy generation is typically curtailed in order to ensure the

stable operation of the power system. Eirgrid publishes an annual report (Eirgrid, 2015b, 2016, 2017) documenting the level of renewable curtailment which occurs each year. In its report, Eirgrid also documents measures put in place to reduce the amount of renewable energy that is curtailed.

2.2.2 Adapting Energy Consumption Based on Renewable Availability

Existing research has examined directly and indirectly shifting electricity consumption to times when renewable energy is available.

In Dusparic et al. (2015), the shifting of electricity consumption to times when renewable energy is available is examined. A collection of residential households were examined, each with a different base load profile. Each household had an electric vehicle which required charging. The consumption pattern of the electric vehicle charger was managed by a controller which performed the majority of the charging during times when renewable energy was available. The controller was encouraged to consume renewable energy even during times of peak demand. However, when renewable energy was not available, the controller was encouraged to postpone charging to periods of low overall demand.

Dusparic et al. (2013) examined the shifting of electricity consumption to periods of low overall demand. The proposed system was successful in its goal of shifting energy consumption to periods of off-peak demand. This provided the ground work for creating a system which shifted demand to periods of high renewable availability in Dusparic et al. (2015).

In Ruelens et al. (2016a), the electricity consumption of a water heater device is modified according to an external price profile. A system for controlling the water heater is proposed which aims to minimise the cost of energy consumption while providing a high utility to the end user. The energy pricing examined was based on the Belgian day-ahead and imbalance profiles. The Belgian price profile incorporates a component that reflects the renewable energy availability.

2.2.3 Analysis

Existing research has examined directly increasing the consumption of renewable energy by shifting energy consumption to times when it becomes available. Research has also examined the shifting of energy consumption for other goals which can be adapted for the goal of maximising the use of renewable energy.

This dissertation considers electricity consumed during periods of renewable curtailment as having a zero CO_2 emissions cost. The goal of this dissertation is to minimise CO_2 emissions of energy consumption. This can be achieved through maximising the use of electricity at times when renewable energy curtailment occurs.

2.3 Residential Energy Application

Existing Research has examined shifting the energy consumption pattern of electric vehicle charging, water heating and also general applications. The following section outlines some of the related works which examine each of these applications.

2.3.1 Electric Vehicle Charging

The application of electric vehicle charging has been examined in many related works which aim to adapt a pattern of energy consumption (eg. Dusparic et al. (2013, 2015), Shao et al. (2011)). Electric vehicles store energy in a battery. An electric vehicle is typically connected to a power supply when the owner is at home in order to replenish the battery. The main requirement of the electric vehicle is that it be sufficiently charged for its next journey. It is not important when the charging of the electric vehicle occurs as long as it provides utility when required. These properties mean that the application of electric vehicle charging has a flexible consumption pattern.

2.3.2 Water Heating

Water Heating has been examined in many related works as a suitable application for shifting a pattern of energy consumption (eg. Ruelens et al. (2016a, 2014, 2016b), Aljabery et al. (2014), Gholizadeh and Aravinthan (2016)). Water heaters are typically

insulated water storage tanks with one or more sources of energy. Related works have only examined water heaters with electricity as a single source of energy. Similar to the charging of electric vehicles, a water heater can consume and store energy at any time. Although energy can be consumed at any time, it is important that the water heater provide hot water (utility) to the end user when required. These properties make water heating an excellent application for adapting an energy consumption pattern.

2.3.3 General

The shifting of the consumption pattern of general electrical devices has also been examined in existing research (eg. O'Neill et al. (2010), Wen et al. (2015)). Research using the application of general devices has focussed on learning patterns in end user behaviour. General devices may have a flexible or inflexible consumption pattern. This can lead to a level of dis-utility to the end user as the utility of an inflexible device may be delayed.

2.3.4 Analysis

Existing research in the area of shifting the energy consumption pattern of electrical devices has examined various applications which are summarised below:

- Electric Vehicle Charging.
- Electric only water heating.
- General electrical device usage.

This dissertation examines the application of water heating using two sources of energy; electricity from the grid and heat from a gas-fired boiler. This type of water heater inherits the useful properties of a standard electric water heater which has a flexible consumption pattern. Multiple sources of energy provide the water heater with an increased level of flexibility in electrical energy consumption.

2.4 Control Algorithm for Adapting an Energy Consumption Pattern

Existing research has examined various algorithms for controlling an electrical device such that its energy consumption pattern is adapted. Reinforcement learning based algorithms are frequently used in related works. This is largely due to the fact that obtaining an optimal control policy for a device is typically a non-trivial task. It involves managing the operation of a device, shifting its energy consumption pattern and also providing maximum utility to the end user. The Q-learning algorithm is an example of a reinforcement learning algorithm. Existing research has also examined the use of Batch Reinforcement Learning algorithms such as Fitted Q-iteration.

2.4.1 Q-learning

The Q-learning algorithm has been frequently implemented in related works in order to learn an optimal control policy for an electrical device (O'Neill et al., 2010, Wen et al., 2015, Pan and Lee, 2016). O'Neill et al. (2010) chose the Q-learning algorithm because of its common usage and relative ease of understanding. Pan and Lee (2016) chose to use the Q-learning algorithm as it does not require a state transition model of its environment and it can operate in stochastic environments.

2.4.2 Fitted Q-iteration

A number of related works have implemented the Fitted Q-iteration algorithm in order to learn an optimal control policy for an electrical device (Ruelens et al., 2016a, 2014, 2016b). In Ruelens et al. (2014), the Fitted Q-iteration algorithm was implemented to learn a control policy for an electric water heater with the aim of minimising the cost of energy consumption. Fitted Q-iteration was chosen as it is a batch reinforcement learning technique that does not require many interactions with its environment before converging to a reasonably good policy. This is based on how Fitted Q-iteration updates its policy using a combination of off-line learning as well as on-line learning.

2.4.3 Distributed W-learning

A distributed multi-agent reinforcement learning algorithm, Distributed W-learning (DWL) has been examined in existing research (Dusparic et al., 2013, 2015, Taylor et al., 2014). This algorithm is based on Q-learning and W-learning. Each agent in the DWL system passes a message about the current state it is in and the value of that state. Each device learns an optimal policy based on meeting its own objectives and how it affects neighbouring agents Taylor et al. (2014).

2.4.4 Analysis

The control algorithms which have been examined in existing research can be summarised as follows:

- Q-learning (Reinforcement Learning)
- Fitted Q-iteration (Batch Reinforcement Learning)
- Distributed W-learning (DWL)

This dissertation examines the adapting of the energy consumption of a single water heater device which is managed by a single agent. This means that a multi-agent algorithm such as DWL would be inappropriate. This dissertation uses Q-learning as an algorithm for learning an optimal control policy. Q-learning is a reinforcement learning algorithm which learns an unknown transition model and can be used in stochastic environments. This makes it suitable for controlling the energy consumption of a water heater device in a stochastic environment. Furthermore, as mentioned in (O'Neill et al., 2010), Q-learning is commonly used and it is relatively straightforward to understand its operation.

Chapter 3

Design

In this chapter, an overview of the water heater is outlined. The operation of a controller device which chooses the appropriate action the water heater should perform is described. Two separate designs of the controller based on different control policies are outlined. The first controller is called the "expert" controller. The expert controller performs actions based on a known good control policy. A second controller is described which uses a reinforcement learning algorithm to learn an optimal control policy. The next chapter will describe the implementation of the controllers outlined in this chapter.

3.1 Water Heater Overview

The water heater examined in this dissertation uses a combination of two energy sources, electricity from the grid and heat from a gas-fired boiler. The water heater is an insulated water tank which means that it can store thermal energy before hot water is required. These two properties of the water heater, *energy storage* and *multiple energy sources*, result in an application that is highly suitable for a flexible and adaptable energy consumption pattern. The design of the water heater is illustrated in figure 3.1.

3.1.1 Hot Water Requirement

The water heater is required to provide hot water (utility) to the end user on a defined schedule. The end user can choose to deviate from this schedule if necessary. It is

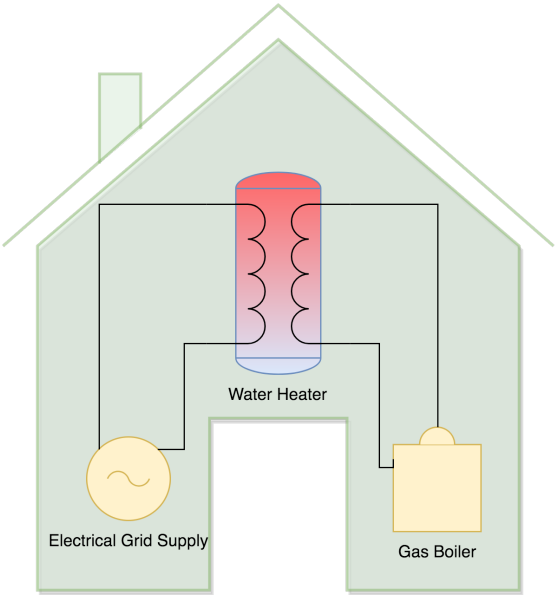

FIGURE 3.1: Water Heater

assumed that the end user will only consume hot water during the defined requirement schedule.

3.1.2 Electrical Energy

The water heater has an internal electrical element which can be powered on or off in order to increase the temperature of the water.

3.1.3 Heat Energy from a Gas-fired Boiler

The water heater has an internal heating coil which is heated by a gas-fired boiler. The gas-fired boiler heats water which is circulated inside the coil. The coil then transfers this heat energy into the water heater causing the temperature of the water to rise. The gas-fired boiler is assumed to provide heat energy for both water heating and space heating.

3.1.4 Controller

The water heater is managed by a controller device. The controller device reads a set of input data at a finite time step interval. The controller uses this input data to update its internal state. Using this state information, the controller makes a decision about

the next action that should be performed. The controller then carries out the chosen action before waiting for the next time step. Section 3.1.4.1 outlines the set of input data that is read by the controller. Section 3.1.4.2 outlines the set of available actions that the controller can take based on the current state.

Note: The controller device is considered to be an embedded device which is network connected.

3.1.4.1 Controller Input Data

Data Item	**Description**
Temperature	This indicates the temperature of the water leaving the tank.
Height	This indicates the height of the hot water column inside the water heater.
Water Demand	This indicates the rate at which hot water is being consumed.
Electric Element On/Off	The on/off state of the electrical element.
Gas Boiler On/Off	The on/off state of the gas-fired boiler.
Electricity CO_2 Emissions	The current amount of CO_2 per kWh of electricity produced by the grid.
Time Until Hot Water Required	The length of time before hot water is required by the end user.

Table 3.1: Input Data Provided to the Controller

3.1.4.2 Controller Actions

The controller can select one of five different actions after each state update. However not all actions can be chosen from all states.

1. Switch the electrical element on.
2. Switch the electrical element off.
3. Switch the gas-fired boiler on.

4. Switch the gas-fired boiler off.

5. Do nothing.

The following rules determine which actions can be performed in a particular state. Not all actions are available to be chosen in any given state.

- In states where the gas fired boiler is on, the agent can only select from actions: 1, 4 or 5.
- In states where the electrical element is on, the agent can only select from actions: 2, 3 or 5.
- In states where the water heater is overheating and the electrical element is on, the agent can only select action 2.
- In states where the water heater is overheating and the gas-fired boiler is on, the agent can only select action 4.
- In states where the water heater is overheating and the electrical element is off and the gas-fired boiler is off, the agent can only select action 5.

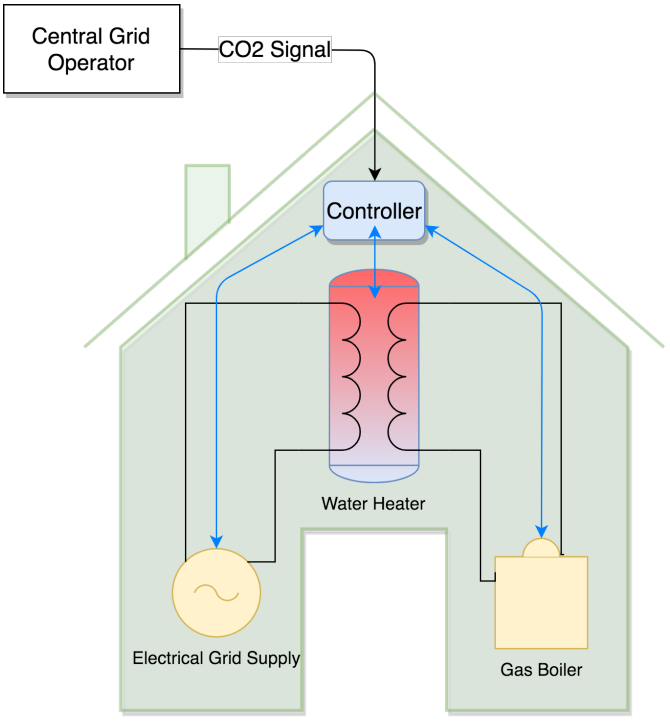

FIGURE 3.2: Water Heater with CO_2 Signal

3.2 CO_2 Signal

The controller receives a signal over a network connection (see figure 3.2) which represents the current level of CO_2 emissions produced by the electricity grid. This signal is sent from a central authority which manages the electricity grid. The signal is a broadcast message which informs the controller of the current CO_2 cost of consuming a unit of electricity (gCO_2 /kWh).

This dissertation considers times when renewable energy approaches a level where it begins to be curtailed as periods of *high renewable availability*. Consuming electricity during these periods results in less renewable energy being curtailed. In other words, consuming electricity during periods of high renewable availability, means that available renewable energy is consumed which would otherwise be wasted.

This dissertation considers the CO_2 emissions produced from the electricity grid during periods of high renewable availability to be zero. This is because additional electricity consumed during these periods maximises the use of available renewable energy.

The CO_2 signal which is broadcast to the controller, can also be interchanged with a price signal. For example, if the electricity grid operator implements a financial incentive to consume energy at times when CO_2 emissions are low, the price will also be low. Similarly, consuming energy at times when CO_2 emissions are high, the price will also be high.

3.3 Expert Controller

The expert controller was designed based on a known good policy. This policy was derived from a set of *expert* rules. The expert rules are shown in table 3.2.

Rule Number	**Description**	**Action**
1	Tank is overheating	Switch off
2	Hot water is not required until much later and electricity with low CO_2 emissions is available and the tank is below the target temperature	Switch on
3	Hot water is soon required and the tank is below the target temperature	Switch on
4	Hot water is soon required and the tank is less than half full of hot water	Switch on
5	Hot water is soon required and the tank is within the target temperature range and the tank is currently heating	Switch on
6	All other states	Switch off

TABLE 3.2: Expert Controller Rules

Note: when the expert chooses to switch on the water heater, the energy source with the lowest CO_2 emissions is chosen.

3.4 Q-learning Controller

The second controller uses Q-learning (Watkins and Dayan, 1992), a reinforcement learning algorithm which learns an optimal control policy. The precepts of the Q-learning agent are the same inputs available to the controller as outlined on page 15. The actions available to the Q-learning agent are the same controller actions as outlined on page 15.

A reinforcement learning agent learns an optimal (or near optimal) control policy without a known state-action transition model. The agent learns a state transition model through trying different actions in each state. As the agent attempts new actions, it receives feedback in the form of a positive or negative reward once it reaches the next state. By trying an action and observing the resulting reward, the agent can begin to learn an optimal policy.

3.4.1 Agent Environment

- The task environment of the agent is *partially observable*. The agent can only observe state at finite time steps. The entire state of the environment is not visible to the agent. For example, the agent will not observe a change in environment state between observation time steps.
- The agent operates as a *single agent*. There are no other agents which an individual agent needs to cooperate with in order to achieve its goal.
- The environment that the agent operates in is *stochastic*. The next state observed by the agent is not completely determined by the current state and the action performed by the agent. For example, the end user of the water heater may leave a tap running by accident which completely drains the tank and prevents the agent from heating the tank.
- The agent operates in a *sequential* task environment. Actions that the agent decides to perform or not perform will influence the future states of the environment.
- The environment that the agent operates in is a *continuous* environment. For example, the temperature, height and demand of hot water are a combination of continuous values (as opposed to discrete values).

3.4.2 Q-learning Agent

(Watkins and Dayan, 1992) developed an off-policy temporal-difference (TD) learning algorithm which is known as Q-learning. It is defined below (Sutton and Barto, 1998):

$$Q(S_t, A_t) \leftarrow Q(S_t, A_t) + \alpha \left[R_{t+1} + \gamma \max_a Q(S_{t+1}, a) - Q(S_t, A_t) \right]$$

FIGURE 3.3: Q-learning Algorithm

The Q function is a learned action-value function which provides the agent with the value for taking a given action in a given state. It directly approximates the optimal action-value function $q*$ regardless of the current policy of the agent. The current policy

will determine how the state-action pairs (S_t, A_t) are updated. Correct convergence to an optimal policy is eventually achieved through continuously updating the state-action pairs.

The diagram below from Sutton and Barto (1998) demonstrates a pseudo-code version of the Q-learning algorithm. For each step of an agent episode, the agent selects an action A based on the current state S using the Q-value function. The agent then performs the selected action and observes the resulting new state S' and the reward R for performing the previous action. The agent then updates its Q-value for (S, A) based on the resulting S' and the received R. The update of the Q-value also incorporates the learning rate (α) and the discount factor (γ).

The *learning rate* (α) is a value set between 0 and 1. A low learning rate means that Q-values are never updated. A high learning rate means that Q-values are updated very frequently and the agent learns very quickly.

The *discount factor* (γ) is a value set between 0 and 1. The value of the discount factor indicates whether the agent is biased towards receiving current rewards or future rewards. When the discount factor is close to zero, the agent considers rewards in the future to be less important than current rewards. When the discount factor is 1, the agent considers rewards of the infinite horizon to be more important than current rewards.

Q-learning: An off-policy TD control algorithm

Initialize $Q(s, a), \forall s \in \mathcal{S}, a \in \mathcal{A}(s)$, arbitrarily, and $Q(\textit{terminal-state}, \cdot) = 0$
Repeat (for each episode):
  Initialize S
  Repeat (for each step of episode):
    Choose A from S using policy derived from Q (e.g., ϵ-greedy)
    Take action A, observe R, S'
    $Q(S, A) \leftarrow Q(S, A) + \alpha\big[R + \gamma \max_a Q(S', a) - Q(S, A)\big]$
    $S \leftarrow S'$
  until S is terminal

FIGURE 3.4: Q-learning Algorithm Pseudo-code (Sutton and Barto, 1998)

3.4.3 Action Selection

The agent must update its Q-values by visiting as many state-action (S, A) pairs as it can. However, it is not feasible to visit every (S, A) pair and update the appropriate Q-value. This dissertation provides training time for the agent to explore as many random (S, A) combinations as possible. After the exploration phase, the agent then performs actions based purely on its learned Q-values. The latter phase is referred to as the exploitation phase.

3.4.3.1 Exploration

During the exploration phase, the agent chooses from a selection of weighted-random actions. A set of softmax probabilities are calculated based on the current Q-values for each available action ($a \in A$) in the current state S (see figure 3.5).

The temperature parameter (τ) is a value between 0 and infinity which normalises the resulting probabilities. A high temperature value means that the resulting probabilities will be relatively similar. A low temperature value will highlight the higher probabilities and suppress the lower probabilities.

During the exploration phase, the agent uses the set of softmax probabilities as weights for choosing a random action. This means that the agent will tend to choose random actions which have at least been moderately successful during a previous experience. The temperature value used to calculate the set of softmax probabilities is reduced over time as the agent nears an optimal policy.

$$P_t(a) = \frac{\exp(q_t(a)/\tau)}{\sum_{i=1}^{n} \exp(q_t(i)/\tau)}$$

FIGURE 3.5: Softmax Function

3.4.3.2 Exploitation

During the exploitation phase, the agent calculates the softmax probability for each available action ($a \in A$) based on the known Q-value for the current state S. The

temperature used to calculate the set of probabilities is relatively small. This means that only the currently known optimal actions for a given state are chosen.

3.4.4 State Representation

The Q-learning controller requires that the state of the environment be presented to it in a way that adequately encodes the pertinent pieces of information. However, it is also important to consider the size of the state space. A large state space can encode more information about an environment but will require more exploration in order to evaluate state-action pairs. On the other hand, a small state space will require less exploration in order to evaluate state-action pairs but it may not be large enough to adequately inform the agent about the state of the environment.

The following list outlines the various components that form the overall view of environment state for the agent.

Temperature The temperature of the water leaving the water heater is encoded such that it represents whether it is below, at, or above the target temperature. It can also represent when the tank is overheating.

Height The height of the hot water column in the water heater is encoded such that it represents whether it is empty, less than half full, more than half full or full.

Water Required The time until hot water is next required by the end user is encoded in the following values: required, soon required, and required later.

Electric CO_2 The level of CO_2 emissions produced from the electricity grid supply generation.

Electric The state of the electrical element is encoded to indicate whether it is on or off.

Gas The state of the gas boiler is encoded to indicate whether it is on or off.

3.4.5 Rewards

A central component to the operation of a reinforcement learning agent is the reward it receives for carrying out an action in a given state. Receiving a reward for each state-action pair is the mechanism which provides feedback to the agent about how well it has performed. In order for the agent to maximise its long term reward, it must choose optimal actions which result in a positive goal state. A structure of rewards is therefore required to positively reward and reinforce good actions while negatively rewarding (punishing) the agent for taking bad actions.

Chapter 4

Implementation

In this chapter, the implementation of a water heater simulation is outlined. The design of the water heater is outlined in the previous chapter. This chapter focusses on the implementation of the expert controller and the Q-learning agent controller. It also describes the sources of the data used in the simulation. The next chapter will outline the experiments that were run using the simulation and the results that were obtained.

4.1 GridLAB-D Simulation Framework

The GridLAB-D simulation framework was chosen to simulate the water heater. GridLAB-D is designed to simulate power system distribution. It is a framework intended for end users who are investigating the design and operation of power distribution systems (Carlon, 2012). The framework uses a language called GLM in order to describe an electrical grid model. The model can include a number of houses, transformers and power stations which are all connected. A model can specify a number of *players* for loading input data and a number of *recorders* for saving measurements of electrical consumption.

GridLAB-D was chosen as the simulation framework for implementation of this dissertation for a number of reasons.

- GridLAB-D is an *open source* framework which means that the source code can be modified. This was necessary for the implementation of the two water heater controllers.

- The GridLAB-D framework contains sophisticated functions for *modelling the physical characteristics of an electric water heater*. The existing water heater model provides a starting point for implementing the water heater as described on page 13.
- The GridLAB-D framework includes mechanisms for loading *input interval data*.
- GridLAB-D provides mechanisms for saving *output interval data*.
- Related research has used GridLAB-D for simulating electric vehicle charging (Dusparic et al., 2013)

4.2 GridLAB-D Core Operation

The core of GridLAB-D is a collection of *modules* that contain the *class* implementation of different entities. These entities are instantiated as *objects* whose specifications are explicitly defined in a *model*. The operation of each *object* is determined by its implementation which is defined in its *class*. GridLAB-D manages the state of each object and the interactions between each object over a finite time period. There are a number of key terms used to describe GridLAB-D which are outlined below.

Model A model in GridLAB-D describes the overall system that will be simulated. It may specify a collection of houses, transformers and power lines and how they are all connected. This is represented in a GLM file.

Class A class is a c++ source code file. The GridLAB-D project contains a *class* for every item that it models. Within the class are a number of functions which carry out the mathematical modelling of the operation of that class. For example, GridLAB-D has a class file for water heater, capacitor, transformer, dishwasher, air conditioner, etc.

Object Objects refer to instances of a class (similar to the c++ language). Objects are defined and initialised in a model. Each object must specify the values of required parameters in the GLM model file.

Module A module in GridLAB-D contains a collection of related classes. For example, the residential module contains classes for the water heater, air conditioner,

dish washer, electric vehicle charger, etc. A required module must be defined in the model before classes from that module can be instantiated as objects.

The core operation of the GridLAB-D framework is based on discrete time step simulation. Each module is responsible for maintaining the state of its objects during simulation. The module uses a *solver* function which iterates through each object. The solver ensures that all the objects of a module are updated correctly at each time step. Each class implements the logic necessary for updating the state of an object at each time step. This logic is referred to as object synchronisation. This solver function terminates once the state of the all the objects in a module stabilise. An object is said to have reached a stable state after it has not changed state for a pre-determined constant number of time step iterations.

4.2.1 Object Synchronisation

At the core of this implementation is a *sync* function. The sync function accepts two parameters T_0 and T_1. These parameters are passed from the GridLAB-D core. T_0 indicates the time at which the object last performed the sync function. T_1 indicates the time at which the object is currently executing its sync function.

The object must first determine if it has changed in state since its previous update at T_0. If the agent determines that it has changed in state it must calculate the duration of the change $(T_1 - T_0)$. This duration is essential for many calculations that an object may have to perform (eg. calculating the amount of electrical energy consumed since the last update).

The object must finally determine a value to return to GridLAB-D core, T_2. The T_2 return value indicates the time in the future at which the object expects to require its next update. The object can set T_2 to a value `TS_NEVER` which indicates to GridLAB-D core that the object has stabilised. If this occurs, the object can only execute the sync function in future if another object has modified its boundary condition.

4.3 Water Heater

GridLAB-D provides a number of modules including a *residential* module. Within the residential module are a number of related classes including a house and a water heater class. A water heater object in GridLAB-D can only be instantiated in a model as a child of a house object. The house class models the characteristics of a single family home. The house class considers heat gains and losses from: conduction through exterior walls and roof, air infiltration, solar radiation and internal gains from lighting and people. These heat gains and losses influence the ambient temperature of the water heater environment and is therefore important when modelling the operation of the water heater.

The water heater class models the state of the tank as it gains and loses heat energy. The water heater class was originally implemented to model a water heater with only a single source of energy, an electrical element. The water heater required a number of modifications in order to model the different types of controllers which are simulated.

4.3.1 Overriding the GridLAB-D Controller

The standard GridLAB-D controller was bypassed so that a custom controller could control the operation of the water heater. This was achieved by modifying an internal property of the water heater class to indicate the mode of operation of the water heater.

4.3.2 Implementing Gas Heating

The GridLAB-D water heater class was extended in order to model a water heater with two sources of energy; electricity and heat from a gas-fired boiler as described in section 3.1 . This was fundamentally achieved by switching the power input to the water heater to a level that would be supplied by a coil when the gas boiler is running. When the water heater returned to electrical heating, the power input to the water heater was switched to the level supplied by the electrical element in the tank.

4.3.3 Implementing a CO_2 Signal

The water heater class was modified to read the value of a CO_2 signal which is supplied by a GridLAB-D player object. The player object reads a CSV file containing the CO_2 emission value for the current time interval. This source of this CSV file is described in section 4.4.1.2 .

4.4 Data

In order to run the simulation experiments, a collection of data was required. The experiments (see chapter 5) are run over a two year period, 2014 and 2015. It was possible to source data pertaining to the electricity supply from the Irish electrical grid operator, Eirgrid. However, it was not possible to source real data relating to residential hot water usage. In cases where data was not available, it was generated based on the known characteristics of the data.

4.4.1 Eirgrid Data

The electricity grid operator in Ireland, Eirgrid, maintains a data set which contains information about system generation, system demand, renewable energy generation, as well as the CO_2 emissions produced from electricity generation. This data is available upon request to the Eirgrid organisation.

For the purposes of this dissertation, data was requested relating to system generation, system demand, renewable energy generation, and CO_2 emissions produced from electricity generation over a two year period 2014-2015. The data received from Eirgrid was in a time interval format at a 15 minute granularity. *Note: Gaps in data were linearly interpolated.*

4.4.1.1 Renewable Curtailment Events

It was not possible to directly source data regarding when renewable curtailment events have occurred. Eirgrid publishes data about the level of renewable curtailment which

has occurred each year. However, it is not possible to determine this data at a finer granularity.

In this dissertation, renewable curtailment events are considered to occur at times when the amount of renewable energy generation approaches a proportion of the total generation supply which is close to the level when curtailment is required. In 2014, Eirgrid carried out renewable curtailment when renewable energy generation exceeded 50% of total system generation (Eirgrid, 2015b). In 2015, Eirgrid increased the maximum level of renewable generation to 55% of system generation.

Using this information, a collection of renewable curtailment events for 2014 was identified by selecting time periods when the renewable supply exceeded 40% of the total supply. In a similar fashion, a collection of renewable curtailment events for 2015 was identified when the renewable supply exceeded 45% of the total supply. This resulted in a collection of renewable curtailment events which span roughly 16% of each year.

4.4.1.2 Electricity CO_2 Emissions

Data relating to CO_2 emissions produced from electricity generation was sourced from Eirgrid. Gaps and null values occurred in sparse patterns throughout the data and were corrected using linear interpolation. This CO_2 data was in the form of $tCO_2/hour$. This was converted into the form of gCO_2/kWh. The formula for conversion is $\frac{(tCO_2/hr)\times 10^6}{systemgeneration(MW)}$.

This dissertation considers CO_2 emissions produced from electricity generation to be *zero* during periods of renewable curtailment. Therefore, at time intervals which coincide with a renewable curtailment event, the gCO_2/kWh was set to zero.

4.4.2 Generated Data

Unfortunately, data relating to space heating demand, water heating demand and hot water consumption could not be sourced. In Sustainable Energy Authority of Ireland (SEAI) (2013), these sets of data were identified as "Data Gaps". In order to provide a complete set of necessary input data for simulation experiments, some data sets were generated. In an effort to make generated data as representative of real data as possible, it was based on known characteristics and best assumptions.

The generation of the data was implemented using a python script. Space heating demand data, water heating demand data and hot water consumption data was all generated in parallel. This was necessary as certain data such as space heating demand will influence the nature of water heating demand.

4.4.2.1 Space Heating Demand

Space heating demand data was generated based on the requirements of a four person residential household. It modelled a typical requirement pattern of morning and evening heating. The demand pattern was varied according to weekday and weekend requirements as well as seasonality. The generation of space heating demand data also accounted for unexpected increases in demand for space heating.

4.4.2.2 Water Heating Demand

Water heating demand data indicates the times at which hot water is required. It does not indicate the actual level of consumption of hot water. The generation of water heating demand data was heavily influenced by the pattern of space heating demand. Water heating data was generated in such a way that it closely matches the pattern of space heating demand. This was based on the assumption that space heating demand data indicates the level of occupancy of the household.

Water heating demand is encoded as a number which represents the time until hot water is required. This number is based on the number of 30 minute intervals before the requirement of hot water. For example, a value of 0 indicates that hot water is currently required while a value of 2 indicates that hot water is required in one hour and so on. An example sequence is {4, 3, 2, 1, 0}. This method of encoding time before required hot water allows for encoding unexpected increases in demand. For example another sequence, {4, 1, 0}, represents an unexpected advance in demand for hot water.

4.4.2.3 Hot Water Consumption

Typical hot water consumption is estimated to be 15.8 gallons per person per day in a typical household (National Renewable Energy Laboratory, 2011). The hot water

consumption for four people was spread across the period of water heating demand for each day.

4.4.3 Water Heater Properties

The characteristic properties of the water heater such as the volume, heat loss, power output of the electrical element and power output of the coil heated by the gas boiler are based on the values of a real water heater, the Kingspan Tribune XE TXD250 (Kingspan, 2017). The values for each of the properties are shown in the table below.

Property	**Value**
Tank Volume (L)	250
Tank Height (m)	1.8
Coil Power Output (kW)	20.2
Electrical Power Output (kW)	5
Standing Loss (W)	65

TABLE 4.1: Water Heater Properties

4.4.4 Gas Boiler Properties

The characteristic properties of the gas-fired boiler such as the power output, electrical consumption and CO_2 emissions, are based on the values of a real gas boiler, the Baxi Ecoblue 24 Combi boiler (Baxi, 2016). The values for each of the properties are shown in the table below.

Property	**Value**
Power Output (kW)	24
Electrical consumption when firing (kW)	0.85
CO_2 Emissions (gCO_2/kWh)	250

TABLE 4.2: Gas Boiler Properties

4.5 Implementation of the Expert Controller

The expert controller was implemented within the GridLAB-D water heater class. The implementation of the expert controller was based on a known good policy as outlined on page 18. Listing B.1 on page 49 outlines the core implementation of the expert controller based on the rules in table 3.2.

4.6 Implementation of the Q-learning Controller

The Q-learning controller was implemented based on the agent design as outlined on page 19. At the core of the Q-learning controller is the `step` function. The `step` function accepts the current state of the environment as a parameter. It then updates the Q-value for the previous state-action combination based on the new state and the reward received for entering the new state. The learning rate (α) used to update the Q-value is 0.3 . The discount factor (γ) used is 0.7 . Finally the step function returns a new action to be performed. The code implementation for the `step` function can be seen in section B.2.1 on page 50.

4.6.1 Action Selection

The last section of the Q-learning controller `step` function selects the next action to take. Action selection during the exploration phase is performed differently to action selection during the exploitation phase. The `select_action` function selects the appropriate action for the current phase of the agent. An implementation of the `select_action` function can be seen in section B.2.2 on page 51 .

4.6.1.1 Exploration

During the exploration phase, the action selects a weighted-random action as described on page 21. A random number between 0 and 1 is chosen. Each softmax probability is compared against the random number. If the random number is less than the current softmax probability, the action which is associated with that probability is chosen. If the random number is greater than the current softmax probability, the random number

is subtracted by the softmax probability amount. The next softmax probability is then compared. The implementation of this exploration action selection function can be seen in section B.2.2.1 on page 52.

4.6.1.2 Exploitation

During the exploitation phase, the agent chooses what it perceives as the known optimal action. It does this by selecting the action with the maximum softmax probability out of all the softmax probabilities for available actions which is calculated using a relatively small softmax temperature (see 3.4.3.2 on page 21). The maximum probability is selected by searching the entire array of softmax probabilities until the maximum value is found. The action corresponding to the maximum softmax probability is selected. The implementation of the exploitation action selection function can be seen in section B.2.2.2 on page 52.

4.6.2 State Representation

The implementation of the state representation follows the design as outlined on page 22. The agent state was implemented in such a way that it could easily be represented as a number. This was achieved by creating a bit vector representation. This implementation resulted in having each state of the environment represented by 10 bits. This means that there are a total of 1024 agent states. The table 4.3 illustrates the implementation of the bit vector. The bit vector was stored in an `unsigned short` which has a size of 16 bits. The dark grey cells in the table indicate unused bits.

Bit Index	State Component	Value Type
0	RENEWABLE_AVAILABLE	BOOLEAN
1	ELECTRIC_ON	BOOLEAN
2	GAS_ON	BOOLEAN
3	TANK_DRAINING	BOOLEAN
4	TANK_HEIGHT	EMPTY, LESS_THAN_HALF_FULL, MORE_THAN_HALF_FULL, FULL
5		
6	WATER_TEMPERATURE	BELOW_TARGET, AT_TARGET, ABOVE_TARGET, OVERHEATING
7		
8	WATER_REQUIRED	NOW, SOON, LATER, MUCH_LATER
9		
10		
11		
12		
13		
14		
15		

Table 4.3: Bit Vector State Representation

4.7 Water Heater Model

The GridLAB-D model which instantiates the objects necessary for running the simulation experiments was implemented in a GLM file. This GLM file instantiates the following objects:

- A house object which was used as a parent object to the water heater.
- A water heater object which is controlled by the expert or the Q-learning controller.
- A player object for loading CO_2 emission interval data into the simulation.

- A player object for loading the exploration and exploitation schedule for the Q-learning agent.
- A player object for loading the water heating demand interval data.
- A player object for loading the hot water consumption interval data.
- A player object for loading the space heating demand interval data.
- A recorder object for saving the performance of the controller.
- A recorder object for saving the CO_2 emissions produced from water heater usage.
- A recorder object for saving the power consumption data of the house and water heater objects.

Chapter 5

Experimental Procedure

This chapter outlines the experiments that were carried out in order to evaluate the performance of the expert controller and the Q-learning controller. The first experiment compares the performance of the expert controller to the Q-learning controller using regular exploration data and exploitation data. The second experiment compares the performance of the expert controller to the Q-learning controller which mixes the exploration data and exploitation data used in the first experiment. This mixing of the exploration data and the exploitation data is also referred to as a spliced data set.

5.1 Evaluation Period

Each experiment is run over a two year period, 2014 and 2015. The year 2014 is used for the purpose of allowing the Q-learning controller to learn a control policy during its exploration phase. The year 2015 is used for the purpose of comparing the performance of the expert policy to Q-learning agents learned policy.

5.2 Evaluation Episode

An episode is considered to include the time before hot water is required as well as the time when hot water is required. Episodes are used as a time measurement for comparing the performance of the expert controller to the Q-learning controller.

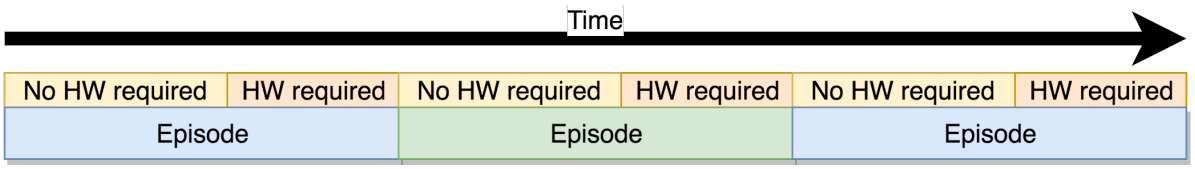

FIGURE 5.1: Length of an Evaluation Episode

5.3 Experiment One

5.3.1 Aim

The aim of this experiment is to directly compare the performance of the expert controller to the Q-learning controller using the default settings and implementation as outlined in the previous chapter. The optimal controller should have the lowest CO_2 emissions without compromising the utility to the end user.

5.3.2 Method

1. A simulation is executed using the expert controller over the 2014 – 2015 data set.
2. A second simulation is executed using the Q-learning controller over the same 2014 – 2015 data set. The Q-learning controller trains on the 2014 data during its exploration phase before exploiting its learned policy during 2015.
3. The results are based on the performance of the controllers during the 2015 period of the simulation.

5.3.3 Results

Metric	Q-learning Agent	Expert
Total CO2 emissions (kg)	1096.25	1124.53
Total energy consumed (kWh)	5018.16	4985.72
Total renewable energy consumed (kWh)	692.81	517.77
Total Utility	0.99	1.0
Mean CO2 emissions per episode (kg)	1.30	1.34
Number episodes without full utility	3.0	0.0
Mean Utility per episode	0.99	1.0
Mean renewable energy consumed (kWh)	0.27	0.20

TABLE 5.1: Results from Experiment One

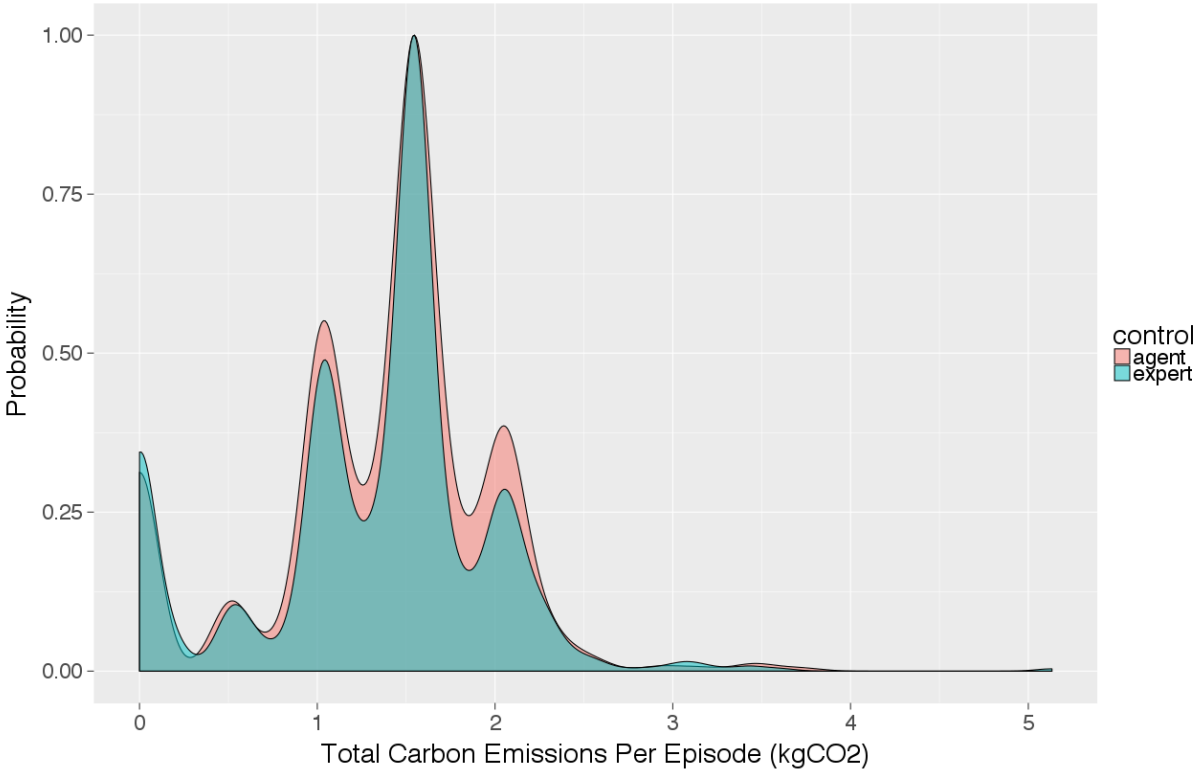

FIGURE 5.2: Probability Density Carbon Emissions ($kgCO_2$) Per Episode

5.3.4 Conclusion

The results of the experiment show a lower level of mean CO_2 emissions per episode for the Q-learning agent at 1.30 compared with the expert at 1.34 . However, the Q-learning

agent mean CO_2 per episode is not a statistically significant lower mean value (p value = 0.07) (statistical significance is when $p < 0.05$).

The mean utility of the Q-learning agent, 0.99 is lower than the mean utility of the expert, 1.0 . The mean utility of the Q-learning agent is a statistically significant lower mean value by a marginal amount (p value = 0.04).

From observing the resulting probability density graph in figure 5.2 it appears that the expert and the Q-learning agent perform at roughly the same level in terms of CO_2 produced per episode. The second experiment uses a different exploration and exploration data set in an attempt to obtain conclusive results.

5.4 Experiment Two

5.4.1 Aim

The aim of this experiment is to directly compare the performance of the expert controller to the Q-learning controller using the implementation as outlined in the previous chapter. However, in this experiment, the CO_2 emission data for 2014 and 2015 are spliced together. This was done by taking the values from every second day in 2015 and using them in 2014 and vice versa.The optimal controller should have the lowest CO_2 emissions without compromising the utility to the end user.

5.4.2 Method

- A simulation is executed using the expert controller over ***spliced*** 2014 – 2015 data set.
- A second simulation is executed using the Q-learning controller over the same ***spliced*** 2014 – 2015 data set. The Q-learning controller trains on the 2014 data during its exploration phase before exploiting its learned policy during 2015.
- The results are based on the performance of the controllers during the ***spliced*** 2015 period of the simulation.

5.4.3 Results

Metric	**Q-learning Agent**	**Expert**
Total CO2 emissions (kg)	1173.84	1221.99
Total energy consumed (kWh)	4962.11	5350.75
Total renewable energy consumed (kWh)	318.34	498.58
Total Utility	0.99	1.0
Mean CO2 emissions per episode (kg)	1.40	1.45
Number episodes without full utility	8.0	0.0
Mean Utility per episode	0.99	1.0
Mean renewable energy consumed (kWh)	0.24	0.20

TABLE 5.2: Results from Experiment Two

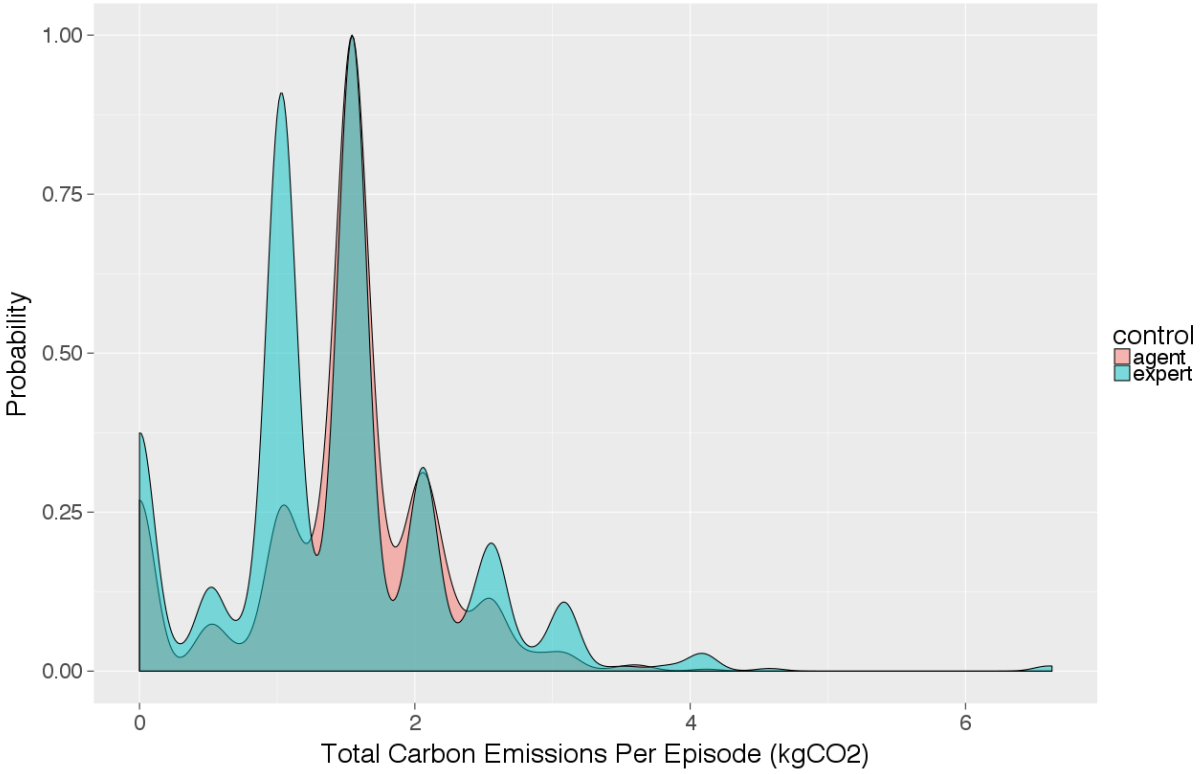

FIGURE 5.3: Probability Density Carbon Emissions ($kgCO_2$) Per Episode

5.4.4 Conclusion

The results from the experiment show that the Q-learning agent achieved a lower mean value of CO_2 emissions per episode (1173.84) compared to the expert (1221.99). The

agent achieved a statistically significant lower mean CO_2 emissions value per episode (p value = 0.02) .

The results also show that the mean utility per episode achieved by the agent (0.99) was lower that the mean utility achieved by the expert (1.0). The Q-learning agent lost utility in 8 out of the total 837 episodes in the simulation. This lower mean utility per episode achieved by the agent was statistically significant (p value 0.008).

Based on these results, it seems that the Q-learning agent can operate and produce lower CO_2 emissions. However, this lower CO_2 emissions level come at a price to the utility of the end user.

Chapter 6

Conclusion

6.1 Controller Comparison

The core objective of this dissertation was to determine whether the Q-learning controller can exceed the performance of the expert controller. The results show that the Q-learning controller performed almost at the same level as the expert controller. This indicates that the agent learned a control policy that is similar to the known, good control policy implemented by the expert. The results also show that the Q-learning controller achieves a lower mean value of CO_2 emissions produced during each episode. However, this also resulted in a slightly lower mean utility per episode.

6.2 Analysis

The following analysis section discusses some reasons why the Q-learning controller may not have performed better than the expert controller.

6.2.1 Lack of Real Data

Unfortunately, the experiments lack the use of *real* space heating demand, hot water demand and hot water consumption data. Running experiments based on real hot water consumption patterns would have most likely provided significantly different results regarding the performance of the two controllers.

If the experiments were run using real data based on the hot water consumption pattern of an individual household, the Q-learning agent would have the opportunity to adapt its control policy to the characteristics of that household. The expert controller would be at a disadvantage under these circumstances as its control policy is static and does not adapt.

The experiments in this dissertation used a set of generated data as described in section 4.4.2 on page 29. Ultimately, this data can not adequately reflect the varying levels of hot water consumption that a real household will exhibit.

6.2.2 CO_2 Emissions of Electricity

In the initial stages of this project, the CO_2 emissions per kWh of electricity generated were assumed to vary quite substantially as a result of the varying levels of renewable energy availability. Once the data regarding CO_2 emissions produced via electricity generation was sourced from Eirgrid, it became apparent that the level of CO_2 per kWh of electricity never drops below the CO_2 per kWh produced by a gas boiler. This dissertation considers the CO_2 produced from the electricity supply to be zero during periods of renewable curtailment. This means that electricity only has lower CO_2 emissions per kWh compared to the gas boiler during periods of renewable curtailment.

This discovery had a large impact on this project as it was originally assumed that a relatively sophisticated control policy would be required to strategically consume electricity outside of renewable curtailment events.

6.3 Future Work

In the future, renewable energy will become a larger source of the electricity grid generation. Renewable energy can be safely accommodated at higher levels on the grid each year. This means that renewable curtailment will become less frequent. It also means that the CO_2 emissions per kWh of electricity generated will eventually drop below the CO_2 emissions per kWh produced by a gas boiler. This will require a more sophisticated control policy in order to minimise the CO_2 produced from water heater usage in the future.

Water heaters are continually progressing to accommodate more sources of energy. Water heaters are even starting to be used for space heating purposes. This type of water heater will require a more sophisticated control policy in order to manage an increasing number of requirements. Under these circumstances, a learning agent which learns an optimal control policy may prove more successful.

Further work is required in the area of gathering data about residential hot water consumption patterns and heating demand patterns. This data is important for carrying out simulations which are representative of the environment they are modelling.

Appendix A

Security Considerations

In the following chapter, security considerations relating to the implementation and deployment of the system as outlined in chapter 3 are discussed. This chapter will focus on the interaction between the learning agent (client) and the central authority (server). The main purpose of the server is to provide data to the client about the state of the environment (i.e. the current CO_2 emissions produced from electricity generation).

In a real world deployment, the client will most likely be running on a low power, embedded device connected to the internet. The central server will actively listen for requests from any client. The client will make requests at regular intervals in order to maintain an up-to-date state of the agent environment. An example of the client requesting data from the server can be seen in figure A.1 below.

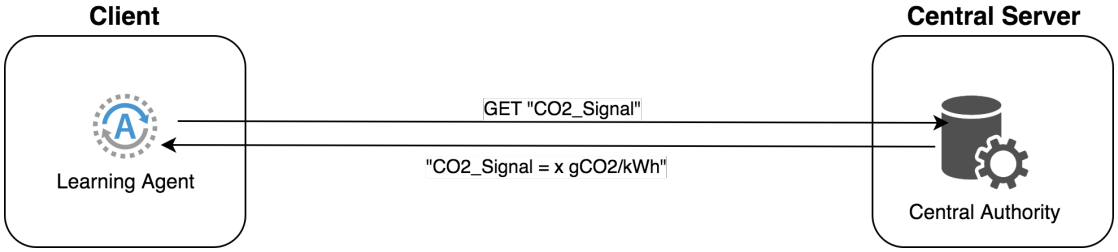

FIGURE A.1: Overview of Client Request to Central Server

The communication between the client and server introduces some potential security issues which could allow an attacker to tamper with the system. The following sections discuss how to prevent these attacks from occurring.

A.1 Central Server Authentication and Message Integrity

In this system, the client must be able to verify the authenticity of the server that it is communicating with. If the client cannot verify the identity of the server, a rogue server could be servicing the request of the client.

In order to provide server authentication, the server must have a certificate. A digitally signed certificate is issued from a certificate authority (CA). The certificate of the issuing CA may in turn be signed by a superior CA. The certificate combines the identity of the central server authority and the server public key. The private key which is linked to the certificate public key must be kept secret. The client must trust the certificate of the server itself or of the issuing CA or a CA superior to the issuing CA.

Before any communication between the client and server can occur, both parties must first negotiate a Transport Layer Security (TLS)(Dierks and Rescorla, 2008) handshake. As part of this handshake, the server will send it's certificate. The client will then encrypt its own secret key using the server public key. If the client trusts the server certificate and the server can decrypt the client secret, then the client has successfully verified the authenticity of the server.

If a successful TLS handshake has been negotiated between the client and server, a secure Transmission Control Protocol (TCP) connection will be established. The client and server will both have formed a symmetric key which is used to encrypt the data payload of each TCP packet. The data payload consists of the user data to be transmitted as well as a Message Authentication Code (MAC). The MAC ensures the integrity of the data so that the message sent by the server is the same message that the client receives.

A.2 Message Insertion, Deletion, Modification or Replay

The connection between the client and server is secured using TLS. This means that each packet that is transmitted contains an encrypted payload. The decrypted payload includes the data and the corresponding MAC. The TLS standard specifies that the message, data length, sequence number, MAC key and two fixed character strings are all components of the computed MAC.

The sequence number component makes it possible to detect when messages are inserted or deleted. The MAC itself protects against message modification as this can only be computed by the sender. The sequence number also protects against replay attacks.

A.3 Man-In-The-Middle (MITM)

An attacker can launch a MITM attack by hijacking a TCP connection between a client and server. If successful, the attacker can intercept and retransmit the data transferred between the client and server.

This type of attack can be protected against by using TLS to secure the TCP connection as described in section A.1. This provides a means of authentication so that the client can verify the legitimacy of the server. For the purposes of this system, only one side of the communication needs to be authenticated; the server.

A.4 Eavesdropping

The data being transmitted from the server to the client is not particularly sensitive information. This is because the data is regarding the state of the public electrical grid system and is not personal data. However, communication that occurs over a TLS connection as outlined in section A.1 will also ensure data confidentiality. This is because the data transferred between the client and server is encrypted and can only be decrypted by the client and server.

A.5 Client System Security

The physical security of the client is considered to be of low risk due to its location within a residential household. A much greater risk to the security of the client comes from its connection to the internet. This potentially exposes the device to an attacker located anywhere in the world. The client should be configured to not listen for incoming connections. The client should be configured to only open connections to the trusted central server.

A.6 Client Fail-safe

In cases where the client cannot communicate with the server due to some sort of failure (eg. server offline due to DOS attack, TLS handshake failure, etc.), the client should resort to a "safe" mode of operation. This means that the client should update its state so that it can operate without severe negative consequences. For example, if the client cannot obtain the state of renewable energy availability from a trusted source, it should assume that renewable energy is not available.

Appendix B

Code Implementation

B.1 Expert Controller

```
if(overheating){
   // Rule # 1
   switch_on = false;
} // Hot water is not required until much later
else if(gda_st_heating_distance > 1){
   // Rule # 2
   if(gda_st_renewable_available && less_than_target){
      switch_on = true;
   } else switch_on = false;
} // Hot water is soon required
else if(gda_st_heating_distance <= 1){
   if(less_than_target){
      // Rule # 3
      switch_on = true;
   } else if(ai->get_tank_height(h) == ql_state::LESS_THAN_HALF ){
      // Rule # 4
      switch_on = true;
   } else if(currently_inside_target_band && currently_heating){
      // Rule # 5
      switch_on = true;
   } else{
      // Rule # 6
      switch_on = false;
```

```
24      }
25   }
```

B.2 Q-learning Controller

B.2.1 Q-Learning Step Function

```
1  ql_action::QL_ACTION qlearning::step(ql_state new_state){
2     ql_action::QL_ACTION action;
3     double alpha = props->getLearningRate();
4     double gamma = props->getDiscountRate();
5
6     /* Manage the episode state */
7
8     // check if the last_state was the end of an episode
9     if(isGoalState(last_state) && !isGoalState(new_state)){
10       episode_state = END_OF_EPISODE;
11    }
12
13    if(episode_state == END_OF_EPISODE){
14       // do the end of episode stuff here
15       number_episodes++;
16       // calculate the softmax temp
17       double discount = props->getSoftmaxTemperatureDiscount() * (↩
     number_episodes / 10);
18       current_softmax_temperature = props->getSoftmaxTemperatureIntial() ↩
     - discount;
19       if(current_softmax_temperature < props->↩
     getSoftmaxTemperatureMinimum()){
20          current_softmax_temperature = props->↩
     getSoftmaxTemperatureMinimum();
21       }
22       // transition to not in episode
23       episode_state = NOT_IN_EPISODE;
24    }
25
26    if(episode_state == NOT_IN_EPISODE){
27       // the agent is starting for first time
```

```
28        episode_state = NEW_EPISODE;
29    }
30
31    else if(episode_state == NEW_EPISODE){
32        // transition to in episode state if already in new episode
33        episode_state = IN_EPISODE;
34    }
35
36    /* Now perform action selection and update q values */
37
38    if(episode_state != NEW_EPISODE && episode_state != NOT_IN_EPISODE){
39        // get the current reward
40        char current_reward = r_values[new_state.get_state()];
41        // update q value for the last state, action combination
42        // based on the current reward
43        double old_q_value = q_values[last_state][last_action];
44        double max_following_reward = max_reward(new_state.get_state());
45        double new_q_value = alpha * (current_reward + ((gamma * ↩
     max_following_reward) - old_q_value));
46        q_values[last_state][last_action] = new_q_value;
47    }
48    if(episode_state != NOT_IN_EPISODE){
49        // select a new action
50        action = select_action(new_state);
51    }
52
53    last_state = new_state.get_state();
54    last_action = action;
55    return action;
56 }
```

B.2.2 Action Selection

```
1 /*
2  * Manages action selection and exploration/exploitation phase.
3  * Always makes the appropriate action selection operation.
4  * Returns the action the agent should take based on the current state.
5  */
6 ql_action::QL_ACTION qlearning::select_action(ql_state st){
```

```
7     ql_action::QL_ACTION action;
8     if(exploit){
9        action = exploit_action(st);
10    }
11    else{
12       action = explore_action(st);
13    }
14    return action;
15 }
```

B.2.2.1 Exploration

```
1  ql_action::QL_ACTION qlearning::explore_action(ql_state st){
2     vector<double> probs = softmax_probs(st, current_softmax_temperature);
3     // get random number between 0 and 1
4     double random_num = (double)rand()/(double)RAND_MAX;
5     int selected_action = -1;
6
7     for(int i = 0; i < probs.size() && selected_action == -1; i++){
8        if(probs[i] >= 0.0){
9           if(random_num < probs[i]){
10             selected_action = i;
11          }else{
12             random_num -= probs[i];
13          }
14       }
15    }
16
17    assert(selected_action != -1);
18    return ql_action::QL_ACTION(selected_action);
19 }
```

B.2.2.2 Exploitation

```
1  ql_action::QL_ACTION qlearning::exploit_action(ql_state st){
```

```
2    vector<double> probs = softmax_probs(st, props->↩
     getSoftmaxTemperatureMinimum());
3    int max_index = 0;
4    for(int i = 0; i < probs.size(); i++){
5       if(probs[i] > probs[max_index]){
6          max_index = i;
7       }
8    }
9    return (ql_action::QL_ACTION)max_index;
10 }
```

Bibliography

Richard S. Sutton and Andrew G. Barto. *Introduction to Reinforcement Learning*. MIT Press, Cambridge, MA, USA, 1st edition, 1998. ISBN 0262193981.

European Commission. A roadmap for moving to a competitive low carbon economy in 2050. 2011.

M. H. Albadi and E. F. El-Saadany. Demand response in electricity markets: An overview. In *2007 IEEE Power Engineering Society General Meeting*, pages 1–5, June 2007. doi: 10.1109/PES.2007.385728.

D. O'Neill, M. Levorato, A. Goldsmith, and U. Mitra. Residential demand response using reinforcement learning. In *Smart Grid Communications (SmartGridComm), 2010 First IEEE International Conference on*, pages 409–414, Oct 2010. doi: 10.1109/SMARTGRID.2010.5622078.

F. Ruelens, B. Claessens, S. Quaiyum, B. De Schutter, R. Babuska, and R. Belmans. Reinforcement learning applied to an electric water heater: From theory to practice. *IEEE Transactions on Smart Grid*, PP(99):1–1, 2016a. ISSN 1949-3053. doi: 10.1109/TSG.2016.2640184.

Ivana Dusparic, Colin Harris, Andrei Marinescu, Vinny Cahill, and Siobhán Clarke. Multi-agent residential demand response based on load forecasting. In *Technologies for Sustainability (SusTech), 2013 1st IEEE Conference on*, pages 90–96. IEEE, 2013.

Eirgrid. Ds3: Frequency control workstream 2015. 2015a.

J. A. Short, D. G. Infield, and L. L. Freris. Stabilization of grid frequency through dynamic demand control. *IEEE Transactions on Power Systems*, 22(3):1284–1293, Aug 2007. ISSN 0885-8950. doi: 10.1109/TPWRS.2007.901489.

Ivana Dusparic, Adam Taylor, Andrei Marinescu, Vinny Cahill, and Siobhan Clarke. Maximizing renewable energy use with decentralized residential demand response. *2015 IEEE First International Smart Cities Conference (ISC2)*, page 1, 2015. ISSN 9781467365529. URL `http://elib.tcd.ie/login?url=http://search.ebscohost.com/login.aspx?direct=true&db=edb&AN=112657562&site=eds-live&scope=site`.

Eirgrid. Annual renewable energy constraint and curtailment report 2014. 2015b.

Eirgrid. Annual renewable energy constraint and curtailment report 2015. 2016.

Eirgrid. Annual renewable energy constraint and curtailment report 2016. 2017.

S. Shao, M. Pipattanasomporn, and S. Rahman. Demand response as a load shaping tool in an intelligent grid with electric vehicles. *IEEE Transactions on Smart Grid*, 2 (4):624–631, Dec 2011. ISSN 1949-3053. doi: 10.1109/TSG.2011.2164583.

F. Ruelens, B. J. Claessens, S. Vandael, S. Iacovella, P. Vingerhoets, and R. Belmans. Demand response of a heterogeneous cluster of electric water heaters using batch reinforcement learning. In *2014 Power Systems Computation Conference*, pages 1–7, Aug 2014. doi: 10.1109/PSCC.2014.7038106.

F. Ruelens, B. J. Claessens, S. Vandael, B. De Schutter, R. Babu?ka, and R. Belmans. Residential demand response of thermostatically controlled loads using batch reinforcement learning. *IEEE Transactions on Smart Grid*, PP(99):1–11, 2016b. ISSN 1949-3053. doi: 10.1109/TSG.2016.2517211.

K. Al-jabery, D. C. Wunsch, J. Xiong, and Y. Shi. A novel grid load management technique using electric water heaters and q-learning. In *2014 IEEE International Conference on Smart Grid Communications (SmartGridComm)*, pages 776–781, Nov 2014. doi: 10.1109/SmartGridComm.2014.7007742.

A. Gholizadeh and V. Aravinthan. Benefit assessment of water-heater management on residential demand response: An event driven approach. In *2016 North American Power Symposium (NAPS)*, pages 1–6, Sept 2016. doi: 10.1109/NAPS.2016.7747831.

Z. Wen, D. ONeill, and H. Maei. Optimal demand response using device-based reinforcement learning. *IEEE Transactions on Smart Grid*, 6(5):2312–2324, Sept 2015. ISSN 1949-3053. doi: 10.1109/TSG.2015.2396993.

X. Pan and B. Lee. An approach of reinforcement learning based lighting control for demand response. In *PCIM Europe 2016; International Exhibition and Conference for Power Electronics, Intelligent Motion, Renewable Energy and Energy Management*, pages 1–8, May 2016.

A. Taylor, I. Dusparic, C. Harris, A. Marinescu, E. Galvn-Lpez, F. Golpayegani, S. Clarke, and V. Cahill. Self-organising algorithms for residential demand response. In *2014 IEEE Conference on Technologies for Sustainability (SusTech)*, pages 55–60, July 2014. doi: 10.1109/SusTech.2014.7046218.

Christopher J. C. H. Watkins and Peter Dayan. Q-learning. *Machine Learning*, 8(3): 279–292, 1992. ISSN 1573-0565. doi: 10.1007/BF00992698. URL `http://dx.doi.org/10.1007/BF00992698`.

Teresa Carlon. Gridlab-d, 2012. URL `http://www.gridlabd.org/`.

Sustainable Energy Authority of Ireland (SEAI). Energy in the residential sector 2013 report. September 2013.

(NREL) National Renewable Energy Laboratory. Domestic hot water assessment guidelines, June 2011. URL `http://www.nrel.gov/docs/fy11osti/50118.pdf`.

Kingspan. Tribune xe unvented installation and maintenance instructions, 2017. URL `https://www.kingspanenviro.com/range/tribune-xe-indirect-cylinders`.

Baxi. Ecoblue combi, system and heat only boilers. range guide, 2016. URL `http://www.baxi.co.uk/literature-library.htm`.

T. Dierks and E. Rescorla. The transport layer security (tls) protocol version 1.2. RFC 5246, RFC Editor, August 2008. URL `https://tools.ietf.org/html/rfc5246`.

M. Pipattanasomporn, M. Kuzlu, and S. Rahman. An algorithm for intelligent home energy management and demand response analysis. *IEEE Transactions on Smart Grid*, 3(4):2166–2173, Dec 2012. ISSN 1949-3053. doi: 10.1109/TSG.2012.2201182.

J. K. Kok, C. J. Warmer, and I. G. Kamphuis. Powermatcher: Multiagent control in the electricity infrastructure. In *Proceedings of the Fourth International Joint Conference on Autonomous Agents and Multiagent Systems*, AAMAS '05, pages 75–82, New York, NY, USA, 2005. ACM. ISBN 1-59593-093-0. doi: 10.1145/1082473.1082807. URL `http://doi.acm.org/10.1145/1082473.1082807`.

Sustainable Energy Authority of Ireland (SEAI). Renewable electricity in ireland 2015. August 2016.

Eirgrid. Annual report 2013. 2014.

S. D. J. McArthur, E. M. Davidson, V. M. Catterson, A. L. Dimeas, N. D. Hatziargyriou, F. Ponci, and T. Funabashi. Multi-agent systems for power engineering applications #x2014;part i: Concepts, approaches, and technical challenges. *IEEE Transactions on Power Systems*, 22(4):1743–1752, Nov 2007. ISSN 0885-8950. doi: 10.1109/TPWRS.2007.908471.

Sustainable Energy Authority of Ireland (SEAI). Energy in transport 2014 report. 2014.

Eirgrid. Eirgrid smart grid dashboard. URL `http://smartgriddashboard.eirgrid.com/`.

W. Li, T. Logenthiran, W. L. Woo, V. T. Phan, and D. Srinivasan. Implementation of demand side management of a smart home using multi-agent system. In *2016 IEEE Congress on Evolutionary Computation (CEC)*, pages 2028–2035, July 2016. doi: 10.1109/CEC.2016.7744037.

J. L. Hippolyte, S. Howell, B. Yuce, M. Mourshed, H. A. Sleiman, M. Vinyals, and L. Vanhee. Ontology-based demand-side flexibility management in smart grids using a multi-agent system. In *2016 IEEE International Smart Cities Conference (ISC2)*, pages 1–7, Sept 2016. doi: 10.1109/ISC2.2016.7580828.

Danny Weyns, H. Van Dyke Parunak, Fabien Michel, Tom Holvoet, and Jacques Ferber. *Environments for Multiagent Systems State-of-the-Art and Research Challenges*, pages 1–47. Springer Berlin Heidelberg, Berlin, Heidelberg, 2005. ISBN 978-3-540-32259-7. doi: 10.1007/978-3-540-32259-7_1. URL `http://dx.doi.org/10.1007/978-3-540-32259-7_1`.

Y. Lasheng, A. Marin, H. Fei, and L. Jian. Studies on hierarchical reinforcement learning in multi-agent environment. In *2008 IEEE International Conference on Networking, Sensing and Control*, pages 1714–1720, April 2008. doi: 10.1109/ICNSC.2008.4525499.

Toshiyuki Yasuda, Motohiro Wada, and Kazuhiro Ohkura. *Instance-Based Reinforcement Learning Technique with a Meta-learning Mechanism for Robust Multi-Robot Systems*, pages 161–172. Springer Berlin Heidelberg, Berlin, Heidelberg, 2011. ISBN

978-3-642-23232-9. doi: 10.1007/978-3-642-23232-9_15. URL http://dx.doi.org/10.1007/978-3-642-23232-9_15.

A. D. Peacock and E. H. Owens. Assessing the potential of residential demand response systems to assist in the integration of local renewable energy generation. *Energy Efficiency*, 7(3):547–558, 2014. ISSN 1570-6478. doi: 10.1007/s12053-013-9236-4. URL http://dx.doi.org/10.1007/s12053-013-9236-4.

Michael Gleason Robert Preus Eric Lantz, Benjamin Sigrin and Ian Baring-Gould. Assessing the future of distributed wind: Opportunities for behind-the-meter projects. techreport, National Renewable Energy Laboratory, November 2016.

Gregor P. Henze and Jobst Schoenmann. Evaluation of reinforcement learning control for thermal energy storage systems. *HVAC&R Research*, 9(3):259–275, 2003. doi: 10.1080/10789669.2003.10391069.

Sascha Lange, Thomas Gabel, and Martin Riedmiller. *Batch Reinforcement Learning*, pages 45–73. Springer Berlin Heidelberg, Berlin, Heidelberg, 2012. ISBN 978-3-642-27645-3. doi: 10.1007/978-3-642-27645-3_2. URL http://dx.doi.org/10.1007/978-3-642-27645-3_2.

Damien Ernst, Pierre Geurts, and Louis Wehenkel. Tree-based batch mode reinforcement learning. *J. Mach. Learn. Res.*, 6:503–556, December 2005. ISSN 1532-4435. URL http://dl.acm.org/citation.cfm?id=1046920.1088690.

S. Sebastian and V. Margaret. Application of demand response programs for residential loads to minimize energy cost. In *2016 International Conference on Circuit, Power and Computing Technologies (ICCPCT)*, pages 1–4, March 2016. doi: 10.1109/ICCPCT.2016.7530345.

F. Rahimi and A. Ipakchi. Overview of demand response under the smart grid and market paradigms. In *2010 Innovative Smart Grid Technologies (ISGT)*, pages 1–7, Jan 2010. doi: 10.1109/ISGT.2010.5434754.

Stuart J. Russell and Peter Norvig. *Artificial Intelligence: A Modern Approach*. Pearson Education, 2 edition, 2003. ISBN 0137903952.

S Katipamula ZT Taylor, K Gowri. Gridlab-d technical support document: Residential end-use module version 1.0. Technical report, GridLAB-D, 2008.