

Design of a Direct Load Control Program for Residential Electric Water Heaters

by

Mostafa Shad Zolpirani

M.Sc., Sharif University of Technology, 2005

**A Dissertation Submitted in Partial Fulfillment of
the Requirements for the Degree of**

Doctor of Philosophy

In the Graduate Academic Unit of Electrical and Computer Engineering

Supervisor(s):	C. P. Diduch, PhD, Electrical & Computer Engineering M. E. Kaye, MScE, P.Eng., Electrical & Computer Engineering
Examining Board:	L. Chang, PhD, Electrical & Computer Engineering E.C. Guerra, PhD, Electrical & Computer Engineering W. Cook, PhD, Chemical Engineering
External Examiner:	A. Gole, PhD, Electrical & Computer Engineering University of Manitoba

This dissertation is accepted by the

Dean of Graduate Studies

UNIVERSITY OF NEW BRUNSWICK

Sept., 2016

©Mostafa Shad Zolpirani, 2016

Abstract

Domestic electric water heaters (DEWH) hold a large share of residential load in North America. The aggregated load profile of electric water heaters follows a similar pattern to the total household load profile, which means that changing the profile of DEWH load can significantly change the shape of the aggregated load profile.

The research contribution is a new methodology for aggregated load control such that a prescribed load profile is followed by controlling the power consumption of the individual electric water heaters. This is achieved by:

- a novel methodology to model an individual DEWH and estimate its hot water usage based on measurements of its power consumption,
- a forecasting algorithm that considers the payback effect of the control actions and adjusts the load forecasts accordingly,
- a novel predictive controller that provides a forecast on ramp-up or ramp-down reserve capacity to follow a desired load profile.

There are two main benefits of using this controller: i) it provides an estimate of reserve capacity in future, which is important for scheduling ancillary services power load balancing, peak shaving and load shifting, and ii) it is a more green, less expensive alternate method compared to using fossil fuel generators for balancing the power generation and consumption.

One of the main significance of this research is that the methodology for aggregated control was deployed on a pilot project called PowerShift Atlantic (PSA), led by the provincial utility, NB Power. The objective of the PSA was to provide up to 20MW of load following ancillary services to balance the intermittency of renewable generation by directly controlling the power consumption of various load classes, such as electric water heaters, electric thermal storage units and HVACs. The new methodology was deployed as a load aggregator in the PSA project to model and control 300 residential water heaters, and to assess performance in real-time operation.

key words: Direct Load Modeling, Load Forecast, Load Control, Demand-Side Management, Direct Load Control, Smart Grid

Dedication

I dedicate this dissertation to my family, especially ...
to my wife, Pegah, for her supports, patience and understanding
to my parents who taught me the importance of hard work and dedication

Acknowledgments

The author would like to thank his supervisors.

The author has to express his appreciation to the thesis examiners for serving on the exam committee. Your comments and suggestions have greatly improved the thesis.

The amazing faculty and staff in the Department of Electrical and Computer Engineering are appreciated for all the document submission reminders, seminar notices, and continuous help, and special thanks to Shelley Cormier and Ryan Jennings for their priceless help and support.

The University of New Brunswick has provided a supportive and enjoyable environment in which students can study and work. A big thank you to all the faculty, staff and students.

There have been many people who have actively contributed to this project, all their help and support are appreciated.

And special thanks to my wife, Pegah Yazdkhasti, who passionately accompanied and supported me during this journey.

Table of Contents

Abstract	ii
Dedication	iv
Acknowledgement	v
Table of Contents	vi
List of Tables	xi
List of Figures	xii
Abbreviations	xvii
1 Introduction	1
1.1 Problem Statement	4
1.1.1 Load Modeling	7
1.1.2 Load Forecast	8
1.1.3 Load Control	8
1.2 Literature Review and Contributions	9
1.2.1 DEWH Load Modeling	9
1.2.1.1 Literature Review	9
1.2.1.2 Methodology and Contribution	11
1.2.2 DEWH Load Forecast	13
1.2.2.1 Literature Review	13

1.2.2.2	Methodology and Contribution	15
1.2.3	DEWH Load Control	16
1.2.3.1	Literature Review	16
1.2.3.2	Methodology and Contribution	18
1.3	Overview	19
2	Load Modeling	20
2.1	DEWH Model	21
2.2	Parameter Identification	24
2.2.1	Scenario I	25
2.2.2	Scenario II	26
2.3	Water Usage Estimation	27
2.3.1	Scenario I	27
2.3.2	Scenario II	28
2.4	Load Simulator	30
2.4.1	2-Step Simulator	30
2.5	Experimental Results	31
2.5.1	Test-bed Structure	31
2.5.2	Parameter Identification	33
2.5.3	Water Usage Estimation	34
2.5.4	Sensitivity Analysis	36
2.5.5	Simulating Control Signal on Single DEWH	40
2.5.5.1	Case 1: zero water usage, no control action	41
2.5.5.2	Case 2: zero water usage with control action	41
2.5.5.3	Case 3: non-zero water usage, no control action	42
2.5.5.4	Case 4: non-zero water usage with control action	43
2.5.6	Simulating Control Action on Multiple DEWHs	44
2.5.6.1	Case 1: no control action	45

2.5.6.2	Case 2: with control action	46
3	Load Forecast	47
3.1	Introduction	47
3.1.1	Payback Effect	48
3.1.2	Single Load Forecast	48
3.1.3	Aggregate Load Forecast	49
3.2	Single DEWH Load Forecast	49
3.2.1	Load Forecast Using ANN	50
3.2.2	Load Forecast Using Statistic-based Method	50
3.2.3	Payback model	53
3.2.4	Aggregating Single Load Forecast	56
3.3	Experimental Results for Single Forecast	60
3.3.1	Evaluation Criteria	60
3.3.2	Evaluation of the Single Load Forecast	61
3.3.3	Aggregated Forecast Error of Uncontrolled Load	63
3.4	Aggregated Load Forecast	68
3.4.1	Aggregated Forecast Using ANN	68
3.4.2	Enhancing ANN Load Forecast Using Kalman Filter	69
3.4.3	Payback Model	71
3.5	Experimental Results for Aggregated Forecast	76
3.5.1	Evaluation of the Load Forecast	76
3.5.2	Evaluate the Payback Effect	78
4	Load Control Strategies	82
4.1	Definitions	83
4.1.1	Terminology	83
4.1.2	Formulation	86

4.1.2.1	Reserve Sets	86
4.1.2.2	Reserve Capacity	87
4.2	Basic Load Following Strategy	89
4.2.1	Formulation	89
4.2.2	Unmanaged Reserve Capacity	90
4.3	Experimental Results	92
4.3.1	Average Unmanaged Reserve Capacity	92
4.3.2	Control System Performance	95
4.4	Balanced-Reserve Control Strategy	96
4.4.1	Formulation	99
4.4.2	Unmanaged Reserve Capacity	101
4.5	Experimental Results	105
4.5.1	Average Unmanaged Reserve Capacity	105
4.5.2	Control System Performance	107
5	PowerShift Atlantic	111
5.1	Virtual Power Plant	112
5.2	Controllable Load and Communication Infrastructure	113
5.3	Load Aggregator	116
5.4	Experimental Results	120
5.4.1	MECL Pilot	120
5.4.1.1	Load Forecast Evaluation	120
5.4.1.2	Load Controller Performance	122
5.4.2	SJE Pilot	126
5.4.2.1	DEWH Load Extraction	126
5.4.2.2	Communication Delay	127
5.4.2.3	Unmanaged Load Forecast	129
5.4.2.4	Load Controller Performance	130

6	Conclusions	135
6.1	Conclusions	135
6.2	Future Work	138
	References	141
	Appendix A Source Codes	154
A.1	Calculating the physical parameters	154
A.2	Estimating W_D in Scenario I	160
A.3	Estimating W_D in Scenario II	162
A.4	Estimating W_D on the pilot	166
A.5	Statistic-Based Load Forecast	171
A.6	ANN-Based Aggregated Load Forecast	182
A.7	Load Control	188
	Appendix B Forecast Evaluation	197
B.1	Single Load Forecast Evaluation	197
B.2	Sets of <i>ON</i> Loads	199
	Appendix C Time Constraints for Customer Comfort	203
C.1	Worst Case Scenario	203

Vita

List of Tables

2.1	Estimation of physical parameters of the testbed	33
2.2	Different test cases for sensitivity analysis	38
3.1	Performance evaluation of the forecasting methods for a single load. .	62
3.2	A sample of <i>ON</i> -Probability (λ_k) of various houses at various times .	64
3.3	Expected value of relative error of forecasted aggregated load for dif- ferent numbers of loads at 7 : 00am for 15 minutes ahead.	65
3.4	A sample of the expected ratio of the water heaters that are on during consecutive intervals	79
5.1	A summary of the various load classes in PSA	114
5.2	A comparison of the two load classes for UNB	115
B.1	Average performance of the forecasting methods for single load. . . .	197
B.2	Average number of the loads that remains <i>on</i> during consecutive time intervals	199

List of Figures

1.1	Overall structure of the grid	2
1.2	Time scales for ancillary services	2
1.3	Comparison of daily production of various sources [8]	3
1.4	Aggregated household and DEWH power consumption for approxi- mately 200 houses	4
1.5	The desired aggregated load over a time interval in the future	5
1.6	The minimum and maximum achievable load	6
1.7	A DEWH model for double heating elements and multiple zones . . .	10
2.1	Model of a thermostatically controlled DEWH with an external con- trol signal	22
2.2	Two stage model identification for Scenario-I and Scenario-II.	25
2.3	Structure of a single DEWH simulator	31
2.4	The instrumented DEWH as the testbed	32
2.5	Comparison of measured parameters and simulation results	34
2.6	The GUI of the application to analyze the testbed data	35
2.7	Measurements on testbed: water usage, temperature, and power con- sumption	35
2.8	Comparison of the rate of water usages, measured rate; estimated by scenario I, and II	37
2.9	Comparison of the volume of water usage, measured volume, esti- mated by scenario I, and scenario II	37

2.10	Impact of parameter uncertainties, as listed in Table 2.2, on uncontrolled simulation	39
2.11	Impact of parameter uncertainties, as listed in Table 2.2, on controlled simulation	40
2.12	Comparing simulation results with the measured power and temperature when $W_D = 0$ with no control signal (Case 1)	41
2.13	Comparison of simulation results with the measured power and temperature when $W_D = 0$ and 10 minutes off-command (Case 2)	42
2.14	Comparison of the simulation results with the estimated W_D obtained from the proposed methods, with no control signal (Case 3)	43
2.15	Comparison of the simulation results with the estimated W_D obtained from the proposed methods, with a 10-minute turn-off control signal .	44
2.16	Comparison of the aggregated simulation result and the aggregated measured load with no control signal	45
2.17	Comparison of the aggregated simulation result and the aggregated measured load with a 30 minute turn off command	46
3.1	Structure of the neuron and ANN.	51
3.2	At time k_0 , the status of DEWH is unchanged for $t_s^0 = \alpha$ and it is expected to stay unchanged for a total of $\hat{t}_s$	51
3.3	The probability distribution function of the On and Off interval of a single house	53
3.4	Payback effect of a control action on a single DEWH	54
3.5	The success rate of the forecasting methods over a 4-hour ahead forecast in 5 minutes time intervals.	62
3.6	The average probability of a load to be <i>on</i> (λ_k) at various times. . . .	65
3.7	A sample comparison of the relative aggregated error of the two forecasting methods.	67

3.8	The comparison of the aggregation of measured load with the aggregation of forecasting individual loads	67
3.9	The probability distribution function of X	73
3.10	Comparison of the ANN forecast error and Filtered ANN for each sample in the forecast horizon	77
3.11	A snapshot of the one ANN forecast, Filtered ANN forecast and their comparison with the actual measurements	78
3.12	Impact of a control action on the aggregated load	80
4.1	The overall structure of a load following control system.	84
4.2	The average probability that a load would be <i>on</i> at different times of a day	92
4.3	The average measured ($\bar{P}^D$), forecasted ($\widehat{P}^D$) and modeled ($\bar{P}_{theory}^D$) ramp-down reserve capacity (top), and the mean absolute percentage of forecast error (bottom)	93
4.4	The operation of the controller with the basic strategy (top) and its absolute percentage error (bottom)	96
4.5	The mean absolute percentage error of the basic load following strategy	97
4.6	The state transition of an individual DEWH	98
4.7	The state transition of an individual DEWH in the balanced-reserve strategy	100
4.8	The average probability that a load consumes power over two consecutive time intervals.	104
4.9	The average measured, forecasted, and estimated ramp-up and ramp-down reserve capacities.	106
4.10	The mean absolute percentage error in forecasting the reserve capacities.	107
4.11	A sample snapshot of the balanced-reserve strategy trying to follow a target load profile (DAL)	109

4.12	The mean absolute percentage error of the balanced-reserve strategy.	110
5.1	Structure of the PSA Project	111
5.2	The two pilot projects for the University of New Brunswick in PSA .	115
5.3	The structure of the load aggregator for the MECL pilot	117
5.4	The GUI of the main window for the MECL pilot load aggregator. . .	119
5.5	One example of a short term aggregated load forecast for uncontrolled aggregated load.	121
5.6	The mean absolute percentage error of each element in the forecast horizon.	122
5.7	The average unmanaged ramp-down reserve capacity of the basic con- trol strategy over a day (measured vs. forecasted)	123
5.8	The controller changes the uncontrolled load profile to match the de- sired aggregated load.	124
5.9	The mean absolute percentage error of the basic load following strategy	124
5.10	A comparison of the uncontrolled load with controlled load profile to illustrate a peak shaving	125
5.11	Extracting the DEWH load profile from the household load profile in summer (top) and winter (bottom)	128
5.12	The communication delay of reading smart meter of one house	128
5.13	An example of the aggregated load forecast with delayed measurements.	130
5.14	The mean absolute percentage error of each element in the forecast horizon.	130
5.15	Impact of a turn-off command for all water heaters on the aggregated household power consumption.	132
5.16	An example of the aggregated load following controller in the SJE pilot)	133
5.17	The mean absolute percentage error of the load following on SJE pilot	134

C.1	The worst case scenario for water usage (top figure) and its resulting power consumption (middle) and water temperature(bottom) with and without external control signal.	205
-----	---	-----

Abbreviations

Abbreviations

ANN	Artificial Neural Network
DEWH	Domestic Electric Water Heater
DAL	Desired Aggregated Load
DLC	Direct Load Control
DSM	Demand-Side Management
EPS	Electric Power System
FLC	Forecast Load Capability
EUD	End Use Device
GDI	Gross Dispatch Instruction
LAG	Load Aggregator
LCM	Load Control Module
MECL	Maritime Electric Company Limited
PSA	PowerShift Atlabtic
PSU	Performance Status Update
SJE	Saint John Energy
SO	System Operator
TCA	Thermostatically Controlled Appliances
UNB	University of New Brunswick

Nomenclature

C	Heat capacity = $VOL \times \rho \times c_p$, ($Ws/^\circ C$)
c_p	Specific heat of water ($Ws/(kg^\circ C)$)
D_k	The ramp-down reserve set
$E\{X\}$	The expected value of the random variable X
e_R	The relative error of the aggregated load
G	Thermal conductance of the tank = S/R_{th} , ($W/^\circ C$)
$\log(...)$	Napierian logarithm
$n(A)$	Number of members in the set A
N	Total number of the controllable loads
P	Power consumption of a DEWH without external controller (Chapter 2) (W)
P_k	The aggregated power consumptions at time instant k
$\hat{P}(k+j k)$	The forecasted aggregated power consumption at time $k+j$, generated at time k
P_k^D	The ramp-down reserve capacity at time k
P_k^U	The ramp-up reserve capacity at time k
p_k^i	Power consumption of the i^{th} DEWH at time k
P_0	Nominal power rating of the water heater (W)
$Prob\{..\}$	Probability function
Q	Power consumption of a DEWH coupled with an external controller (W)
R_{th}	Thermal resistance of the tank ($^\circ C m^2/W$)
R_E	Relative error of the aggregated power consumption (%)

S	Surface area of the tank (m^2)
S_R	Success rate of forecasting on/off state of a load
$s^i(k)$ or s_k^i	The On/Off status of the i^{th} load at time k
$\hat{s}^i(k k_0)$	The forecasted state of the i^{th} load at time k , generated at time k_0
$\tilde{s}^i(k k_0)$	The forecast error: $s^i(k) - \hat{s}^i(k k_0)$
T	Temperature of the water ($^{\circ}C$)
T_{hi}	Higher threshold level of the thermostat ($^{\circ}C$)
T_{in}	Temperature of incoming water ($^{\circ}C$)
T_{lo}	Lower threshold level of the thermostat ($^{\circ}C$)
t_{on}	Time interval when the heater is on
t_{off}	Time interval when the heater is off
T_{out}	Ambient temperature ($^{\circ}C$)
U_k	The ramp-up reserve set
u_k^i	Control Signal (1: normal internal thermostatically controlled operation, 0:off) for the i^{th} load at time k
VOL	Volume of the tank (m^3)
W_D	Rate of the hot water usage (l/s)
ρ	Density of the water (kg/m^3)
λ_k	Probability that a load is on at time k , which is a function of time: $Prob\{s(k) = 1\}$
η	The expected value of the individual forecast error: $E\{\tilde{s}(k k_0)\}$
γ^2	The expected value of the square error of the individual forecast error: $E\{(\tilde{s}(k k_0))^2\}$

1 Introduction

One of the important constraints in electric power systems is that the generated power should match consumption, which is imposed by physical laws. The generation plants and transmission equipment may shut down if the demand for power exceeds the generation. Thus, at each point in time, the generated power must follow the demand load profile. It is the responsibility of the system operator (SO) to balance generation with demand through scheduling generation and using ancillary services [1]. Figure 1.1 depicts the general structure of the electric power system.

The variation of the consumption may contain fast components in the order of seconds, or slow components in the order of minutes, hours or even days. For each type of variation, some kind of ancillary services exist, as depicted in Figure 1.2.

Traditionally, the system operator adjusts the generation level by including or excluding various generators to track the consumption load profile. Depending on how fast the generation should match the consumption, various ancillary services, such as regulation or non-spinning reserve are defined. Some services can be obtained within a minute, such as spinning reserve, some others may take a longer time to be ready to deliver, such as non-spinning reserve, as illustrated in Figure 1.2. It means that the SO must plan ahead to ramp up generation in order to provide the increasing demand, or ramp down generation when the demand is low.

Moreover, use of sustainable energies, such as wind, is constantly growing around the world. In Canada, the Maritimes has excellent wind resources [2]. The provincial governments have a strategic plan that by 2016 the total installed wind capacity in

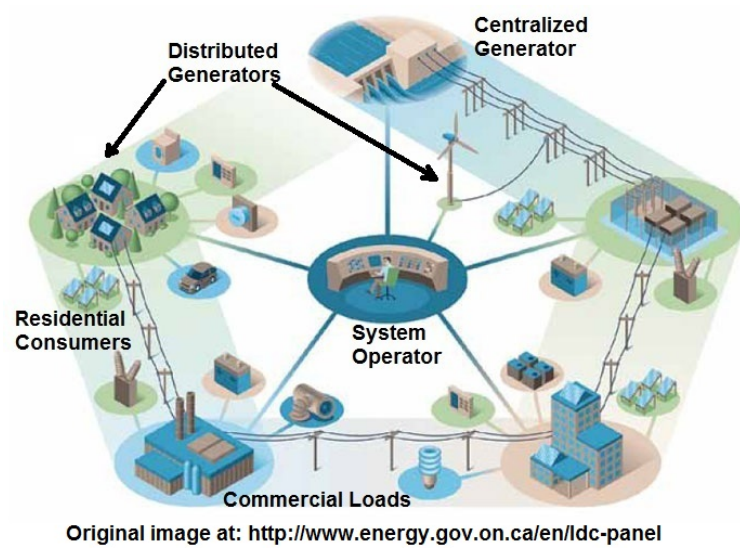


Figure 1.1: Overall structure of the grid

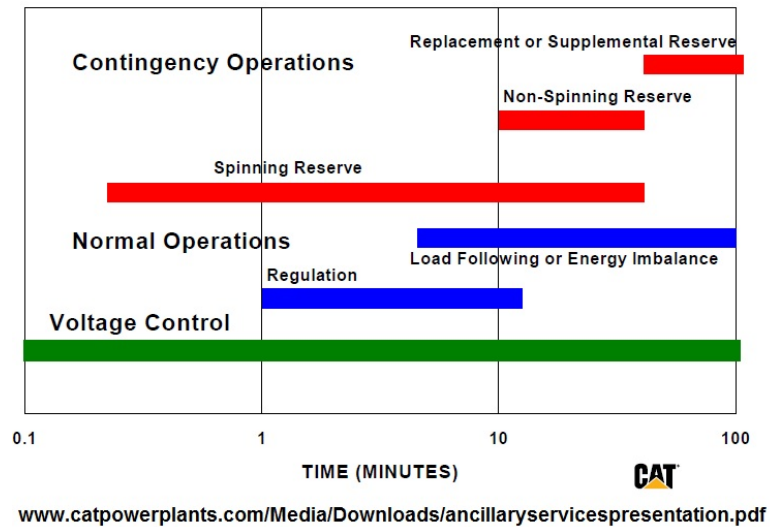


Figure 1.2: Time scales for ancillary services

the Maritimes should reach the range of 16 – 24% of the peak power system load [3] . However, the intermittency of wind energy tends to add more wiggling and uncertainties on the generation side [4, 5, 6, 7], as can be seen in Figure 1.3. This figure shows the variability of the daily power produced by wind and conventional generation [8] during 2014.

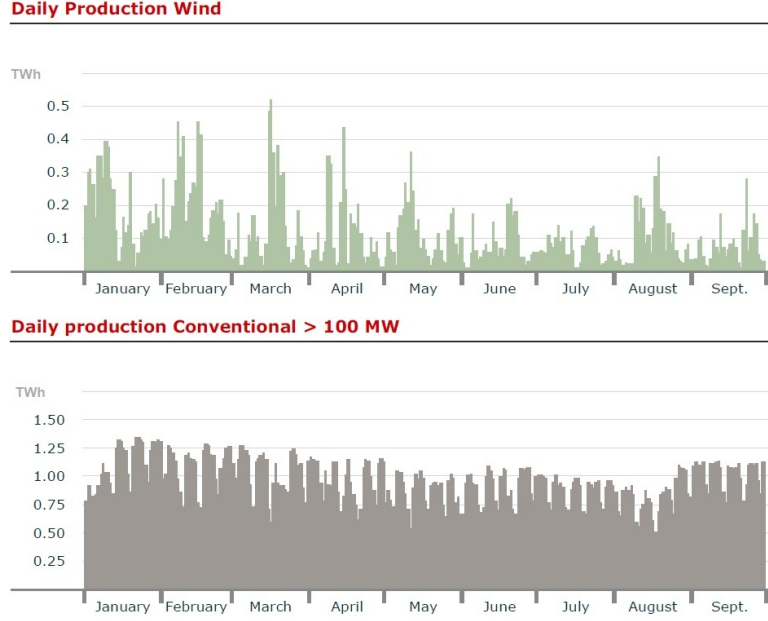


Figure 1.3: Comparison of daily production of various sources [8]

Demand-side management (DSM) is a method to manage the customers load in response to certain conditions. In the traditional electric power systems (EPS), the demand response was the last step to match consumption with generation before forced plant outages [9]. However, in recent years, DSM has been widely used and studied. DSM can be used to reduce energy bills of the customers [10], and can ease power system operation through peak shaving, load following, providing reserve capacity [11, 12] and fast response ancillary services [13].

DSM is a technique to adjust the consumption profile to match generation [13]. The consumption profile can be adjusted indirectly through pricing policies, or directly through controlling specific appliances, with customers' permission, which is called direct load control (DLC) [14]. DLC is the main focus of this dissertation.

Domestic electric water heaters (DEWHs) are feasible candidates for a DLC program because the aggregated load profile of the DEWH i) contributes significantly to the total aggregated residential load [15] and ii) follows similar pattern and peak times [16]. Therefore, changing the shape of the aggregated DEWH load profile can

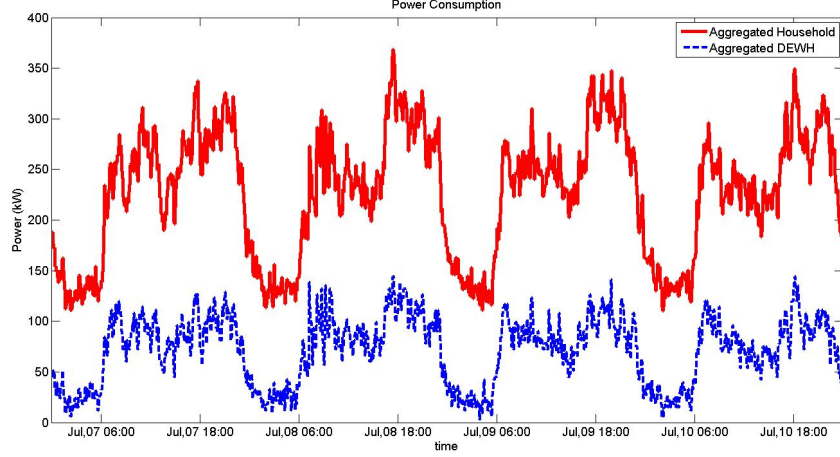


Figure 1.4: Aggregated household and DEWH power consumption for approximately 200 houses

reshape the total load profile. An example of the aggregated household and water heater power consumption for several days is depicted in Figure 1.4. The measurements are obtained from Saint John, NB. in July 2014. It can be observed that the DEWH load holds a large share of the total load with similar pattern and peak time.

1.1 Problem Statement

The problem is to develop a controller for individual water heater loads in a way that the aggregated DEWH load tracks a desired aggregated load (DAL) profile (provided by the system operator), while maintaining customer comfort. The desired aggregated load is a profile of desired power levels, defined over a finite time interval in the future, as depicted in Figure 1.5.

It is assumed that the power consumption of the end-use device (EUD) is available. It is also assumed that the controller can manipulate the demand by strategically controlling when the EUD is turned on or off via an external control signal.

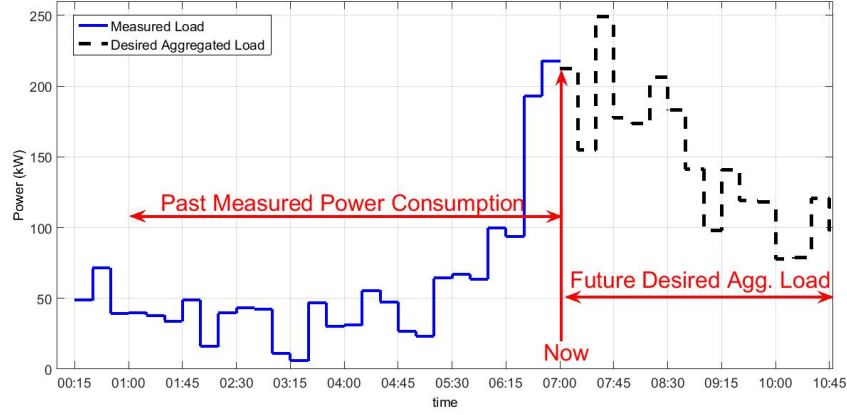


Figure 1.5: The desired aggregated load over a time interval in the future

It should be noted that due to safety constraints, in thermostatically controlled appliances (TCA), the external on-off control does not bypass the inherent interlock safety of the thermostatic control system of the EUD. In other words, it means that when the thermostat turns off the device due to high water temperature, the external controller cannot turn it back on. This constraint limits the maximum achievable power consumption to the number of loads that are thermostatically on at a given time.

Customer comfort is another constraint imposed on the system, which indicates the availability of hot water for the end users. The controller may turn off a power consuming water heater to decrease the aggregated load. This action may result in a drop in the water temperature, and may result in customer discomfort if the controller does not turn the DEWH back on after a certain time. Therefore, it may not be possible to turn off all the loads at a given time, which means that the minimum achievable power consumption is also limited and is a function of the number of the loads that should be left on due to customer comfort.

The minimum and maximum achievable load is limited, as depicted in Figure 1.6. The maximum achievable power consumption is proportional to the number of loads

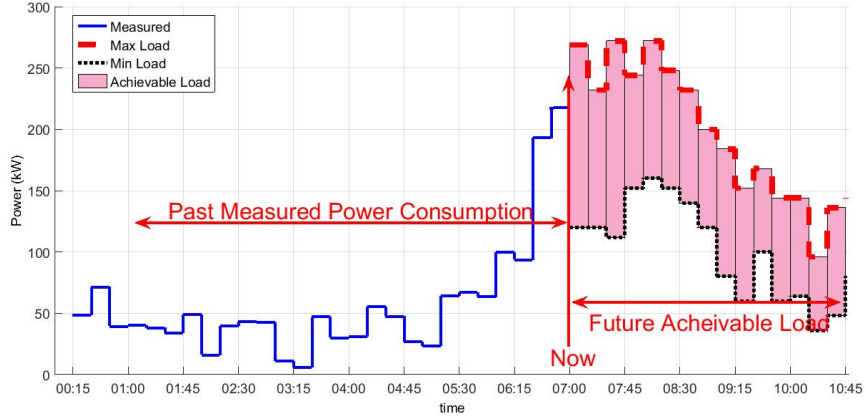


Figure 1.6: The minimum and maximum achievable load

that are on due to its internal thermostatic controller. The minimum achievable power consumption is proportional to the number of loads that cannot be turned off due to customer comfort.

It is assumed that the DAL is generated in a way that the control objective can be met, i.e. the desired aggregated load is feasible. Thus, the DAL should fall within the maximum and minimum limits of the achievable load. This assumption can be fulfilled by forecasting the load and its limits.

The control strategy defines how the load should be scheduled to both follow the DAL and maintain customer comfort. The definitions for the customer comfort and the strategy for scheduling the loads may change the limits of the achievable loads. So, different control strategies may have different behaviour and capacity to follow the DAL.

Finally, before deploying any controller on any pilot, its performance should be evaluated through a simulator. The simulator should mimic the behaviour of the given pilot. It means that the simulated power consumption should be similar to the measured power consumption of that pilot, especially in presence of an external on/off control signal.

A model of the DEWH is required to simulate it. Then, the pilot can be simulated by aggregating the individual DEWH model. The simulated DEWH power consumption will be similar to an actual DEWH power consumption, if the model matches the actual DEWH. Thus it is required to identify the model of each individual DEWH in the given pilot through a systematic method.

So, the problem can be broken into smaller pieces:

1. model the load and identify its parameters
2. forecast the power consumption of the load
3. define an objective function and control actions to optimize it

1.1.1 Load Modeling

Various models have been proposed for the electric water heaters. These models describe a relation between power consumption of a DEWH and its parameters, including water temperature, physical characteristics, water usage, and ambient temperature. By entering these parameters into the model, the on/off status of the simulated DEWH can be obtained.

The goal of simulating a specific water heater is to get a close estimation of its parameters in a way that the simulated power consumption would be close to that of the actual water heater. In practice, these parameters are not known precisely; thus, a methodology is required to identify the system parameters through indirect measurement.

Due to variation of hot water usage and lack of additional sensors, it is a complicated problem to identify the DEWH system parameters by means of its power consumption alone.

1.1.2 Load Forecast

Load forecasting has been widely studied in electric power systems, and many different methods have been proposed to generate accurate load forecasts. However, there are two main issues that makes the load forecast more challenging in a DLC program:

1. Modeling the payback effect: Executing a control action to follow a desired aggregated load will manipulate the load profile; thus, the forecasted load may be different from the measured controlled load. This is known as the payback effect of the control actions. The challenge in providing the load forecast is to model the payback effect and update the forecast regarding the control action.
2. Forecasting uncontrolled load based on controlled history: Most of the forecast methods use previously measured data to estimate future consumption. However, executing the control commands will change the normal behaviour of the load; thus, forecasting the load based on un-natural history may lead to an inaccurate forecast.

1.1.3 Load Control

The control strategy defines how the controller schedules the DEWHs power consumption in order to follow a desired load while maintaining customer comfort. Other goals can also be defined for the controller, such as providing a reserve capacity to increase or decrease the load (e.g. turn off a DEWH before it is turned off by its internal thermostat to provide ramp-up reserve capacity).

1.2 Literature Review and Contributions

1.2.1 DEWH Load Modeling

1.2.1.1 Literature Review

Several studies have focused on modeling the aggregated DEWH load. The goal is to predict the behaviour of the aggregated load and the way it changes in the presence of a control action.

Lee and Wilkins developed a regression-based model to model the aggregated power consumption of the water heaters and the impact of the control action on it [17]. Kobe and Tsoi used a similar model for the aggregated load, but they included the time of the day and day of the week into their model [18]. Lane and Beute [19] proposed a statistical method based on a Gaussian distribution function to estimate the number of water heaters that consume energy at a given time. Through this function, the amount of power consumption was inferred and used for the control purpose.

These models are useful when the information of individual loads are difficult to obtain, or there are too many loads in the control system to be modeled. However, the focus of this research is to model individual loads, and these methods cannot be used to model the individual loads.

Individual DEWH models show how the water temperature in a water heater changes due to various parameters such as water usage, incoming power, or heat loss through its body. A general structure of an electric water heater is depicted in Figure 1.7. There may be one or more heating elements in a water heater. The water temperature may also be assumed to be uniform or to be gradually distributed

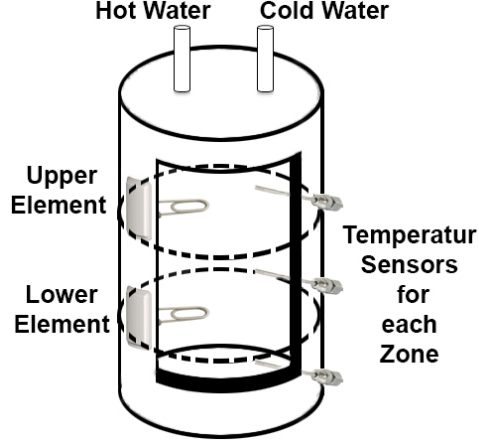


Figure 1.7: A DEWH model for double heating elements and multiple zones

across the tank. Therefore, based on different assumptions, different models can be developed to express the behaviour of a water heater.

Gustafson et al. present a single zone model with a single heating element in [20]. The model is developed based on heat transfer equations. It consists of two parts: 1) convective heat transfer, which is a function of the tank's thermal resistance. 2) heat transfer for replacing water which is a function of water usage. Although this model is valid in terms of energy consumption, the value of temperature and the power consumption cannot be obtained at each instance of time. It only provides the average values over a given time interval.

Tonder and Lane proposed a model [21] based on a histogram of DEWH events. In this method, a simple heat transfer model was used with a fixed parameter for the changes in water temperature. Although this model is useful for detecting the on/off states of the water heaters, it does not include the heat loss parameter and the dynamic behaviour of the water heater at each instance of time.

Fernandez-Seara et al. have proposed a multiple-zone for a single element water heater in [22, 23]. This model expresses the temperature of various levels of the tank in terms of physical parameters, power consumption, and heat transfer between each

layer. However, to identify these parameters, the temperature of various zones is required, which is not practical.

Kondoh et al. presented a dual-zone model for a dual-element water heater [12]. The nature of the size of each zone depends on the rate of water usage. Since the water usage information is not available, the size of zones cannot be formulated. Beside that, the heat transfer equations between each layer is also a function of the heat transfer coefficient of different layers of water with different temperature, where there is no methodology to estimate these coefficients.

A second order differential equation is proposed by Vrettos, Koch, and Andersson [24] to model the temperature of any level in the tank at any given time. Xu et al. used a similar model [25] and presented several numerical methods to solve this equation. Although this model describes a complete behaviour of the tank, the solution process is time consuming and complicated. Besides, the only important temperatures are the top level which is used by the user, and the temperature at thermostat level which affects the on/off state of the water heater.

A first order differential equation is proposed by Dolan et al. in [26] to model a single element water heater. This model assumes that the temperature of the water is uniform across the tank and that changes are based on the thermal characteristics of the water heater. This model is easy to implement, easy to solve. The solution of this model can provide the water temperature at any time.

1.2.1.2 Methodology and Contribution

The model proposed by Dolan et al. in [26] is widely used by other research studies [27, 19, 28, 15, 29, 30] and will be used in this project. However, none of these studies used this model to imitate a real DEWH. In these studies, the water usage

and physical parameters (such as thermal resistance of the tank) are either directly measured or replaced by a general value.

In practice, the thermal and physical parameters for individual water heaters are not known precisely. There is no guarantee that using random values for water usage or physical parameters result in accurate simulation of the pilot. Thus, a methodology is required to estimate physical parameters and water usage of each individual load through indirect measurement.

Paull et al. presented a method [31] to estimate the volume of hot water usage during the off-interval (i.e. the interval at which the water heater is off). This method requires measurements of the DEWH power consumption, the water temperature, and the volume of the tank; however, no method is proposed to estimate the volume of the tank, nor the water usage during the on intervals.

The contribution of this research on model identification is distinguished from other cited work because in this research, a novel methodology is presented to estimate physical parameters (thermal conductivity and heat storage capacity) and water usage based on indirect measurements of a real DEWH. Two scenarios are considered: i) the measurements of power consumption and water temperature are available, ii) only the measurements of power consumption are available.

The estimated parameters and the model are used to build a simulator to study the behaviour of the DEWH in a DLC program. Experimental results show that the output of the simulator mimics the behaviour of individual DEWHs and also mimics the aggregated behaviour of many DEWHs.

1.2.2 DEWH Load Forecast

1.2.2.1 Literature Review

Several scholars have modeled the aggregated power consumption with a time series periodic signal to predict future power consumption using regression-based methods. Rahman and Baba presented a self learning method to estimate the coefficients of the time series signals, considering holidays and abnormal weather conditions [32]. Cho et al. [33] clustered the load into various groups (residential, commercial, office and industry) and developed a time-series based method to model and predict each cluster. Sun and Zhang [34] used a self organizing data mining method to deduce the coefficients of the load forecast model.

Artificial neural networks (ANN) are commonly used for creating load forecasts. The output of the ANN is the forecasted load, and the inputs are a combination of historical power consumptions, meteorological data (including temperature and humidity) along with date and time. According to the Kolmogorol theorem, a three layer back propagation ANN can best simulate any nonlinear equation [35]. Hence most of the studies that focused on ANN for load forecasting, adopt a three-layered ANN (one input, one output, and one hidden layer) for this purpose. The ANN-based forecast methods were compared with time-series based and regression based methods [36, 37], and the results show that the ANN-based methods perform better.

Olofsson and Andersson [38] used the difference between indoor and outdoor temperature with the required energy for heating the house to feed an ANN to produce a household load forecast. Ekonomou used a multilayer perceptron ANN to generate a long-term aggregated load forecast [39]. Pramelakumari et al. [40] also used an ANN to generate a short-term load forecast of a power system. Chen and York [41] used the load dynamics in the immediate past, the variations in load dynamics dur-

ing the course of a day, and the weather factors as the inputs of an ANN to generate a load forecast.

Dong-Xiao et al. [35] presented a method that first finds the similar meteorological features, then uses a varying structure neural network to determine the optimal structure and optimal fitting approximation. This method produces accurate results, despite its slow convergence. To mitigate the slow convergence of the ANN training, Mian et al. [42] presented a genetic neural network based on a Tabu search in order to mix the global optimization and mountain climbing ability of the generic algorithm with the neural network. Xin-hui et al. [43] also used an ANN and showed that the load forecast performance can be improved when combined with a genetic algorithm.

Fuzzy neural networks are also used by various studies [44, 45]. Temperature, humidity, and the measurement dates are fuzzified by assigning a fuzzy set, which describes the quality of that input, for each input. These sets are fed into the ANN to produce a load forecast [45], or a forecast on the peak time of the load [44].

Al-Hamdi and Soliman presented a forecast method [46] that models the load as a linear combination of the weather factors and previous load data with fuzzy coefficients. These coefficients were calculated by Kalman filter techniques in conjunction with expert knowledge stored in a fuzzy rulebase.

Jain et al., [47] used a fuzzy rule base to determine a similar day based on weather conditions and power consumption to produce a load forecast. Osman et al. [48] used a correlation analysis to determine a similar day and used it in an ANN for load forecasting.

Some other approaches were also used to generate a load forecast. Lai et al. [49] used the support vector learning machine as a data mining tool to generate load

forecast of a building. Li et al. [50] presented a state space time varying parameter equation to generate a load forecast.

1.2.2.2 Methodology and Contribution

None of the previously cited research addresses the effect of a control action on the forecast nor the use of an altered historical load (due to a control action) to generate an uncontrolled load forecast. This research is distinguished from others by taking into account the payback effect of the control actions on the forecasting algorithms.

This research has two main contributions on load forecast:

- a statistic-based load forecast method for the individual water heaters along with a mathematical model for the aggregated forecast error based on individual forecast error.
- a model to formulate the payback effect of a control action on a single and aggregated load. This model can be used to adjust the forecasted load when a control action is scheduled, and also can be used to convert a historical profile of the controlled load into an uncontrolled load profile to generate a load forecast.

Experimental results show that the statistic-based load forecast provides up to 10% better prediction of the DEWH on/off state over the first hour ahead than the ANN-based method. Also the payback model can predict the impact of the control actions on the forecasted load with less than 5% error.

1.2.3 DEWH Load Control

1.2.3.1 Literature Review

The DEWH load class has been widely studied since the early 1980s [17] and is still an attractive research area. Kondoh et al. [12] have shown that the water heater loads can be used to provide regulation ancillary service. This study suggests that 33000 water heaters are required to provide $2 - MW$ regulation service 24 hour a day. Kyoho et al. [51] have shown that using direct load control of the water heaters can reduce the additional battery storage capacity required to satisfy the transmission constraints. Pourmousavi et al. [52] have shown that the customers can reduce their electric bills by 45% through controlling the thermostat set point of their water heaters and also can provide reserve capacity for load shifting.

Saele and Grande [53] placed an 'El-Button' on the end-use device (such as dryer or DEWH) in a pilot in Norway to indicate the peak hour to users and remind them not to use them during the peak hour. Hu and Li [54] have developed hardware to implement a real-time price-responsive smart load controller with machine learning algorithms based on the Naive Bayes Classifier and hidden Markov model. Since the individual loads are not aware of the actions of other controllers, the aggregated controlled load may result in an even larger peak [55, 56].

Several other studies have used additional hardware to change the voltage level of the DEWH in order to change its power consumption. Jia et al. [28, 15] have reduced the voltage on the DEWH during peak hours to shave the peak of the aggregated load and shift it to an off-peak period. Ayoub [57] proposed a central controller which receives temperature profiles from all water heaters and then broadcasts the level of the power consumption to all water heaters. Mufaris et al. [58] proposed a similar method where the power consumption level can be adjusted as 0, 25%, 50%, 75%

and 100% of the nominal power. In this project, It is assumed that only an on/off control action can take place in this system; thus, this method is not applicable in this research.

Various studies have used artificial intelligence to control the DEWH load. Yao et al. [59] used an iterative deepening genetic algorithm (IDGA) to minimize the system operation cost. An artificial neural network (ANN) was proposed by Atwa et al. [60] to shift the peaks of average residential DEWH power demand from periods of high demand to off peak periods where the water heaters are grouped according to the distribution feeders that they are connected to.

Many studies have used fuzzy controllers as a tool for direct load control. Nehrir et al. [61] presented a multiple-block fuzzy controller in order to level the distribution feeder profile. Salehfar et al. [62] used a fuzzy controller on electric water heaters and air conditioners to shift the average power demand and improve the utility's load factor. Elgazzar et al. [29] used a centralized fuzzy controller to reshape the aggregated load profile of the DEWH load class. A fuzzy controller is proposed by Wu et al. in [63] to schedule various appliances (such as DEWH, electric vehicle, photovoltaic cells, etc.). Al-jabery et al. [64] used the fuzzified measurements of instantaneous grid load, water consumption and water temperature in a semi-Markov decision process to determine if a DEWH should be on or off. A fuzzy controller for a large water heater population is proposed by Vrettos et al. [24] to track an aggregate power set point while maintaining customer comfort. However, none of these methods provides a forecast on achievable loads.

Ruelens et al. [65] used a model-free batch reinforcement learning (batch RL) to minimize the cost of hourly varying electricity price by scheduling a cluster of the DEWH loads. Guo et al. [66] used a decentralized control algorithm to optimize the energy use plans for various loads including thermostatically controlled loads such

as DEWH and deferrable loads such as dryers. However, this model cannot be used for a load following controller.

All of these methods are using real-time measurements of the load and optimize the consumption at present time. Sossan et al. [67] have used a model predictive control that forecasts the future on/off states of the load and also the photo-voltaic (PV) power generation to maximize the PV-self consumption. Another predictive control is presented by Lajoie et al. [68] to predict the power consumption of the water heaters and use it in a load following control strategy. However, these methods do not consider the payback effect of the control actions on the forecast model.

1.2.3.2 Methodology and Contribution

Despite the variety of the proposed controllers, none of them address the calculations of the forecasted reserve capacity or the impact of control actions on the behaviour of the load. Therefore, a new methodology is required to implement a load control that models the future consumption of the load and considers previous control actions to generate reserve capacity and follow the aggregated set points.

Two control strategies (basic strategy and balanced-reserve strategy) are developed in this dissertation: 1) a basic control strategy that simply combines an uncontrolled load forecast with the payback model to minimize the difference between the forecasted load and the set point, subject to a time constraint for customer comfort, and 2) a balanced reserve provider which performs additional control actions on the loads to balance the ramp-up and ramp-down capacity of the forecasted load by enabling or disabling some water heaters in advance.

A mathematical model for the amount of reserve capacity provided by each control strategy is developed and validated through both simulation and experimental

results on a pilot project with a set of approximately 100 electric water heaters. On a daily basis, the basic strategy provides approximately $57kW$ of ramping down reserve capacity with approximately 8% error in load following. The balanced-reserve provides approximately $42kW$ and $21kW$ of ramp-down and up capacity respectively, while the average error on load following is approximately 20%.

The significance of this controller is that it can be implemented by using a simple 'on/off' electronic switch with no complex additional hardware. The significance of this methodology is that the controller was deployed in a real pilot, PowerShift Atlantic [69], to assess the performance of this controller in operation.

1.3 Overview

The rest of this thesis is organized as follows:

Chapter 2 presents the dynamic model of the water heater. It also addresses a methodology for system and parameter identification. The structure of the simulator and some experimental results are also listed.

The load forecast module is illustrated in Chapter 3. This chapter addresses the methods for single and aggregated load forecast along with the payback model of the control signals. The presented models are compared with experimental results.

Chapter 4 presents the control strategies and their reserve capacity. Simulation and experimental results are also described in this chapter.

The controller and its various components were deployed on a pilot project, PowerShift Atlantic [69]. An overview of this pilot project along with experimental results are presented in Chapter 5. Finally Chapter 6 concludes this dissertation by providing a summary of this project with suggestions for future work.

2 Load Modeling

It is desirable to simulate control systems before deployment in order to establish expectations on performance. If the simulator closely reflects the behaviour of the uncontrolled experimental system, then it can be expected that the performance of the controlled system in simulation would be close to its performance within the experimental system.

This chapter provides detailed information about the characteristics and modeling of a DEWH. Also two methodologies are provided to identify model parameters in two scenarios: I) a combination of DEWH power consumption and water temperature, or II) through the measurements of the DEWH power consumption only. This method can be used to identify the DEWH model parameters of any pilot and build a simulator to mimic the behaviour of that pilot by aggregating the simulation of the effect of the control actions on individual DEWHs.

The organization of this chapter is as follows: The dynamic model of the DEWH is presented in Section 2.1. A methodology is provided in Section 2.2 to estimate the physical parameters of the DEWH model, and Section 2.3 presents a methodology to estimate a hot water consumption profile. Section 2.4 illustrates the structure of the simulator. The experimental results are presented in Section 2.5.

2.1 DEWH Model

DEWH models show how the water temperature in a water heater may change due to water usage, incoming power, and heat loss through the body of the tank. There may be one or two heating elements in a water heater. The water temperature may also be assumed to be uniform or to be gradually distributed across the tank. Therefore, based on different assumptions, different models can be developed to express the behaviour of a water heater.

A single zone model for a single element water heater is selected in this research because 1) it is possible to identify a thermodynamic model using only measurements of the heater power, and 2) the model is easy to solve and scalable to simulate many loads.

The single zone thermodynamic model given by [26]

$$\dot{T}(t) = \frac{1}{C} [Q(t) - G(T(t) - T_{out}) - \rho c_p W_D(t)(T(t) - T_{in})] \quad (2.1)$$

$$P(t) = \begin{cases} P_0 & T(t) \leq T_{lo} \\ 0 & T(t) \geq T_{hi} \\ P(t - \delta t) & otherwise \end{cases} \quad (2.2)$$

$$Q(t) = P(t)u(t) \quad (2.3)$$

Such a model has been widely applied [28, 15, 29, 31, 70] and is characterized by the thermal capacity, C , and thermal resistance, G , the specific heat, c_p , the density of water, ρ , the heater power $Q(t)$, the inlet water temperature, T_{in} , water demand, $W_D(t)$, and ambient room temperature, T_{out} .

In a thermostatically controlled water heater, the incoming power, $P(t)$, is a function of the water temperature as expressed in (2.2). If the temperature of the

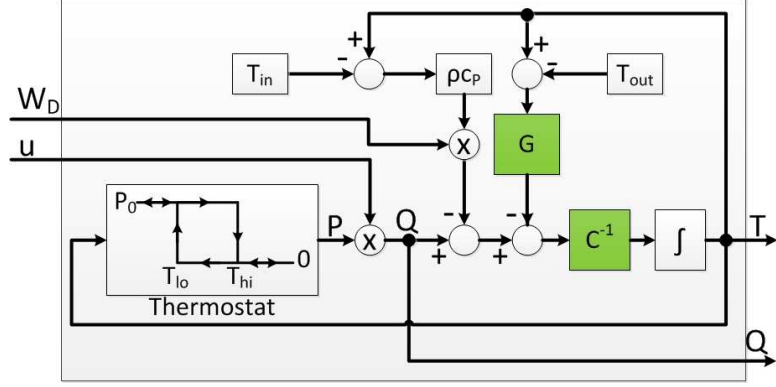


Figure 2.1: Model of a thermostatically controlled DEWH with an external control signal

water is less than the lower threshold value, T_{lo} , then the incoming power will be equal to the nominal power, i.e. it turns on. If the temperature of the water is greater than the higher threshold value, T_{hi} , then the water heater will be turned off; otherwise, it keeps its previous value (where δt denotes a small time interval; thus, $t - \delta t$ in (2.2) indicates the previous time instance in continuous domain).

In this dissertation, it is assumed that an electronic switch is in series with the heating element. Deactivating the switch will turn off the water heater regardless of the state of the water heater thermostat. Activating the switch will enable the internal thermostatic control. This switch can be modeled with a control signal, $u(t)$, as shown in (2.3).

Figure 2.1 illustrates a block diagram of the dynamic model of the DEWH represented by equations (2.1)-(2.3).

Given constant values for the physical parameters C , G , ρ , c_p , T_{in} , and T_{out} and assuming that $W_D(t)$ and $Q(t)$ are piecewise constant over the time interval $t \in [t_0, t_f]$, then the solution of (2.1) is given [31] by Equation 2.4:

$$T(t) = T(t_0)e^{-(t-t_0)/\tau} + A(1 - e^{-(t-t_0)/\tau}) \quad (2.4)$$

where

$$\tau = (C)/(G + \rho c_p W_D), \quad A = (GT_{out} + \rho c_p W_D T_{in} + Q)/(G + \rho c_p W_D)$$

This model contains various parameters that can be categorized into the following groups:

- Constant parameters: these parameters can be assumed to be fixed and constant for all DEWHs during the operation. These parameters are ambient temperature (T_{out}), inlet temperature (T_{in}), thermostat set-points (T_{lo} , T_{hi}), and nominal power rating (P_0).
- Physical parameters: these parameters can be assumed to be fixed during the operation, but may be different from house to house and depend on the physical characteristics of the DEWH. These parameters are heat capacity (C) and thermal conductance (G) of the tank.
- Water usage (W_D): This parameter may change on each usage.

It is assumed that the constant parameters are known and fixed for all end-use devices. But a methodology is required to estimate the physical parameters and water usage through indirect measurements. The available measurements can be either both the measurements of the DEWH power consumption and water temperature (T) sensed by the thermostat, or just the power consumption of the DEWH (Q). Based on the availability of either $\langle Q, T \rangle$ or $\langle Q \rangle$, two scenarios were considered to estimate the model parameters:

1. Scenario I: for the water heaters equipped with a power meter and temperature sensor

2. Scenario II: for the water heaters equipped with only a power meter.

Let the time interval when the DEWH is off ($Q = 0$) be denoted as t_{off} and the time interval when the DEWH is on ($Q = P_0$) be denoted as t_{on} . It can be shown [70] that $\max(t_{off}) = t_{off}^m$ and $\min(t_{on}) = t_{on}^m$ occur when $W_D = 0$. Therefore, it can be assumed that the water usage is zero during the longest *off* interval and the shortest *on* interval of the power consumption profile of a DEWH. Hence, the system identification can be divided into two stages, as illustrated in Figure 2.2:

1. Stage 1: Determine the no-water-usage intervals (t_{off}^m and t_{on}^m) to estimate the physical parameters
2. Stage 2: Use the physical parameters from stage 1 to estimate water usage on the other intervals

Figure 2.2 depicts the 2-stage estimation block for different scenarios where the measurements of $\langle Q, T \rangle$ are available or just $\langle Q \rangle$ is available. Details of each stage will be described in the following sections.

2.2 Parameter Identification

There are two main physical parameters in the dynamic model of (2.1): i) thermal capacity (C), which is a function of the volume of the tank and ii) thermal conductance (G), which is a function of the thermal resistance of the tank and its surface area. These parameters can be identified during the intervals with zero water usage.

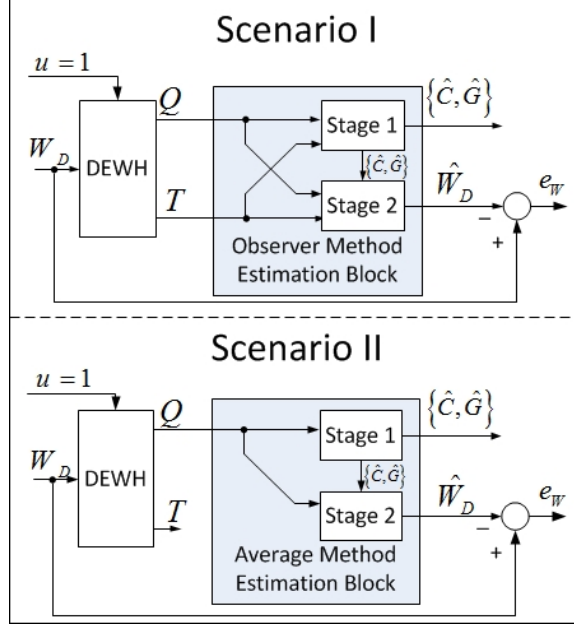


Figure 2.2: Two stage model identification for Scenario-I and Scenario-II.

2.2.1 Scenario I

As discussed in Section 2.1, the first step in identifying the physical parameters is to find the intervals with zero water usage. These are assumed to be the longest *off* interval and shortest *on* interval. For these specific intervals, (2.1) can be rewritten as

$$\frac{d}{dt}(T - T_{out}) = a(T - T_{out}) + bQ(t) \quad (2.5)$$

where

$$a = -\frac{\hat{G}}{\hat{C}}, \quad b = \frac{1}{\hat{C}} \quad (2.6)$$

A linear least square method was used to identify the a and b coefficients using the measurements of power and temperature[71].

2.2.2 Scenario II

In this scenario, only the measurements of $Q(t)$ are available. As mentioned in Section 2.1, it is assumed that $W_D = 0$ during the longest *off* interval and shortest *on* interval. Let us denote the maximum *off* interval by t_{off}^m , and similarly, the minimum *on* interval by t_{on}^m . If $W_D = 0$ then (2.4) may be written as

$$T(t) = T(t_0)e^{-(t-t_0)G/C} + \left(T_{out} + \frac{Q}{G}\right) (1 - e^{-(t-t_0)G/C}) \quad (2.7)$$

During the *off* interval where $Q = 0$, from (2.7), the temperature drops from T_{hi} to T_{lo} as follows,

$$T_{lo} = T_{hi}e^{-t_{off}^m \times G/C} + T_{out} \left(1 - e^{-t_{off}^m \times G/C}\right) \quad (2.8)$$

During the *on* interval where $Q = P_0$, the temperature increases from T_{lo} to T_{hi} as follows

$$T_{hi} = T_{lo}e^{-t_{on}^m \times G/C} + \left(T_{out} + \frac{P_0}{G}\right) (1 - e^{-t_{on}^m \times G/C}) \quad (2.9)$$

P_0 , t_{off}^m and t_{on}^m are readily extracted from measurements of $Q(t)$. Furthermore, assume that the thermostat set-points, T_{hi} and T_{lo} , are at known factory settings. The ratio G/C can be obtained from (2.8),

$$\frac{G}{C} = \frac{\log(T_{hi} - T_{out}) - \log(T_{lo} - T_{out})}{t_{off}^m} \quad (2.10)$$

Now G and C may be obtained using (2.9) and (2.10),

$$G = \frac{P_0(\Delta - 1)}{\Delta(T_{hi} - T_{out}) - (T_{lo} - T_{out})} \quad (2.11)$$

$$C = \frac{P_0(\Delta - 1)}{\Delta(T_{hi} - T_{out}) - (T_{lo} - T_{out})} \times \frac{t_{off}^m}{\log(T_{hi} - T_{out}) - \log(T_{lo} - T_{out})} \quad (2.12)$$

where

$$\Delta = \left(\frac{T_{hi} - T_{out}}{T_{lo} - T_{out}} \right)^{t_{on}^m / t_{off}^m} \quad (2.13)$$

2.3 Water Usage Estimation

The physical parameters obtained from the intervals with zero water usage can be used to estimate water usage profiles in other intervals. In this section, a methodology is proposed for estimating water consumption that best fits the dynamic model of (2.1).

Again two scenarios were considered: Scenario I where the measurements of both power consumption and temperature are available and scenario II where only the measurements of the power consumption is available.

2.3.1 Scenario I

Assuming that the water usage is constant over the sampling interval, the dynamic model of (2.1) can be rewritten as

$$\begin{cases} \dot{T}(t) = f(T) + g(T)W_D(t) \\ \dot{W}_D(t) = 0 \end{cases} \quad (2.14)$$

where

$$\begin{cases} f(T) = \left(Q(t) + \widehat{G}T_{out} - \widehat{G}T(t) \right) / \widehat{C} \\ g(T) = \rho c_p (T_{in} - T(t)) / \widehat{C} \end{cases} \quad (2.15)$$

The water demand, W_D , is estimated using a state observer [72] that is driven by the temperature residual, $\widehat{T}(t) - T(t)$:

$$\begin{cases} \dot{\widehat{T}}(t) = f(\widehat{T}) + g(\widehat{T})\widehat{W}_D(t) - 2\beta(\widehat{T}(t) - T(t)) \\ \dot{\widehat{W}}_D(t) = -\beta^2 \left(\widehat{T}(t) - T(t) \right) / g(\widehat{T}) \end{cases} \quad (2.16)$$

The observer gain, β , governs the convergence properties of the observer error. This observer finds the best estimation of water consumption ($\widehat{W}_D$) that minimizes the error of the estimated temperature ($\widehat{T} - T$).

2.3.2 Scenario II

Although the temperature is unavailable for measurement in this scenario, the temperature is known at the switching instants of the thermostat. At the instant the thermostat turns on, $T(t) = T_{lo}$, and $T(t) = T_{hi}$ at the instant the thermostat turns off.

The time-domain model in (2.4) can be separated 1) to characterize the interval when $Q = P_0$, starting at $t_0 = t_0^{on}$, and 2) to characterize the interval when $Q = 0$,

starting at $t_0 = t_0^{off}$:

$$\begin{cases} T(t) = T_{lo}e^{-(t-t_0^{on})/\tau_{on}} + A_{on}(1 - e^{-(t-t_0^{on})/\tau_{on}}) \\ T(t) = T_{hi}e^{-(t-t_0^{off})/\tau_{off}} + A_{off}(1 - e^{-(t-t_0^{off})/\tau_{off}}) \end{cases} \quad (2.17)$$

where

$$\begin{cases} \tau_{on} = \frac{C}{G + \rho c_p W_D^{on}(t)}, & A_{on} = \frac{GT_{out} + \rho c_p W_D^{on}(t)T_{in} + P_0}{G + \rho c_p W_D^{on}(t)} \\ \tau_{off} = \frac{C}{G + \rho c_p W_D^{off}(t)}, & A_{off} = \frac{GT_{out} + \rho c_p W_D^{off}(t)T_{in}}{G + \rho c_p W_D^{off}(t)} \end{cases} \quad (2.18)$$

Since the temperature measurements during the on/off intervals are not available, an infinite number of solutions for the W_D^{on} and W_D^{off} can be obtained. One of the solutions can be obtained by assuming that the water usage is constant during the entire interval. This method provides the average water rate over each interval and can be obtained by solving the following equation:

$$\begin{cases} T_{lo}e^{-t_{on}/\tau_{on}} + A_{on}(1 - e^{-t_{on}/\tau_{on}}) - T_{hi} = 0 \\ T_{hi}e^{-t_{off}/\tau_{off}} + A_{off}(1 - e^{-t_{off}/\tau_{off}}) - T_{lo} = 0 \end{cases} \quad (2.19)$$

where t_{on} and t_{off} denote the duration of *on* and *off* intervals respectively and

$$\begin{cases} \tau_{on} = \frac{\hat{C}}{\hat{G} + \rho c_p \hat{W}_D^{on}}, & A_{on} = \frac{GT_{out} + \rho c_p \hat{W}_D^{on}T_{in} + P_0}{\hat{G} + \rho c_p \hat{W}_D^{on}} \\ \tau_{off} = \frac{\hat{C}}{\hat{G} + \rho c_p \hat{W}_D^{off}}, & A_{off} = \frac{GT_{out} + \rho c_p \hat{W}_D^{off}T_{in}}{\hat{G} + \rho c_p \hat{W}_D^{off}} \end{cases} \quad (2.20)$$

2.4 Load Simulator

A DEWH load simulator is a useful tool to study the changes in power consumption of a water heater (or aggregated water heaters) in a DLC program. More accurate modeling of the DEWH will result in more reliable simulation. This section describes the methods used to develop a realistic simulator.

2.4.1 2-Step Simulator

If the power consumption of the simulated water heater is close to the power consumption of an actual water heater, then it is expected that the aggregated power consumption of the simulated water heaters will be close to the aggregated power consumption of the actual water heaters.

As described in Section 2.1, the DEWH model requires the water usage profile and certain physical parameters as inputs. The outputs of the model are water temperature and power consumption. The goal is to tune the water usage and physical parameters in a way that the simulated power consumption is close to the actual power consumption of the DEWH.

For this purpose, a 2-step simulator is developed, as illustrated in Figure 2.3. The measurements of power consumption and/or temperature from a DEWH are used in the first step (estimation block) to estimate the physical parameters ($\hat{C}$ and $\hat{G}$) and water usage profile ($\hat{W}_D$). The structure of the estimation block is illustrated in Figure 2.2, and the methods are described in Section 2.2-2.3. The DEWH model is represented in Figure 2.1 and given by (2.1)-(2.3).

The estimated parameters and the water usage are used as the inputs of the DEWH model in the second step, as depicted in Figure 2.3. At this step, an external

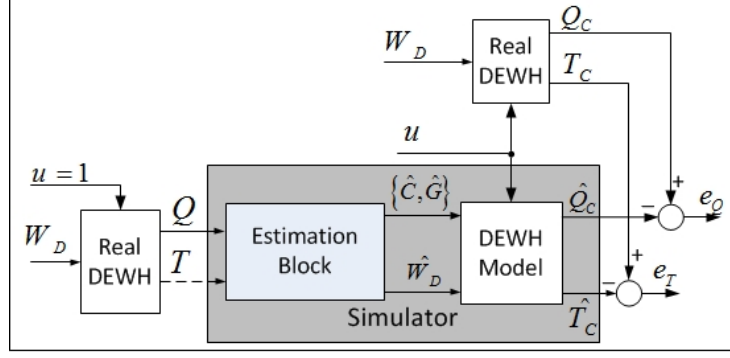


Figure 2.3: Structure of a single DEWH simulator

control signal, u , may be applied to simulate the impact of an external control action on the power consumption of the water heater. In order to validate the simulation results, the same condition will also be applied on a real water heater to compare the simulation results with the actual measurements.

2.5 Experimental Results

Various experiments have been conducted to validate the presented methods in Section 2.2-2.4. Most of these experiments were executed using an instrumented DEWH, called the test-bed. This section will present the structure of the testbed and the experiments for evaluating the parameter identification, water usage estimation, and simulation validation.

2.5.1 Test-bed Structure

The testbed, as shown in Figure 2.4, consists of a 270 liter tank. This DEWH contains two $3kW$ heating elements. In the normal wiring, only one element can turn on at each time and the upper element has higher priority; however, the wiring

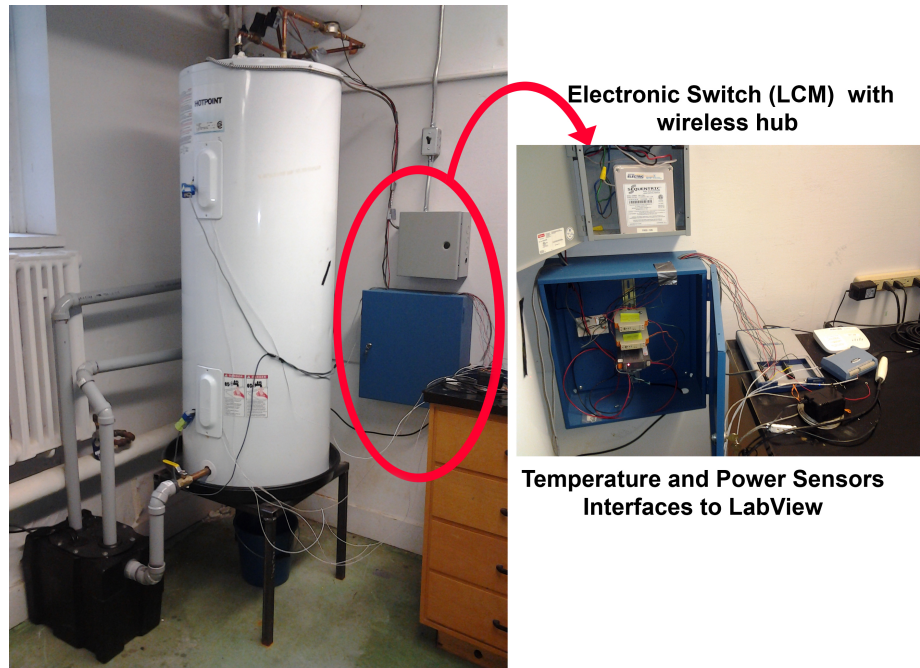


Figure 2.4: The instrumented DEWH as the testbed

was changed to bypass the upper element in order to mimic a DEWH with a single heating element.

The DEWH was equipped with a temperature sensor and power meter to monitor and record its temperature and power consumption using the LABVIEW interface. A temperature sensor was placed on the outlet pipe to measure the temperature of the outgoing water. Another temperature sensor was installed on the inlet pipe to measure the temperature of the incoming water. A temperature sensor measured the ambient temperature.

An electronic valve was installed on the outlet pipe to control the water usage. Also a load control module (LCM) was installed on the water heater to implement the control signal.

2.5.2 Parameter Identification

The physical parameters may be estimated using the intervals with zero water usage. Therefore, without using any hot water from the testbed, the power consumption (Q) and temperature (T) were measured and stored.

For the first scenario, both measurements of the Q and T were used to estimate C and G , but only the measurements of Q were used in the second scenario to estimate C and G . It was observed that for this testbed, the maximum off interval, t_{off}^m , is approximately 710 minutes, and the minimum on interval, t_{on}^m , is roughly 21 minutes. The thermostat set points were between 56 and $60^\circ C$ and the ambient temperature was recorded $T_{out} = 22^\circ C$. Results are listed in Table 2.1. The source codes for estimating C and G are listed in Appendix A.1.

Table 2.1: Estimation of physical parameters of the testbed

Scenario	I (Q and T)	II (only Q)
C	$9.27 \times 10^5 J/^\circ C$	$9.18 \times 10^5 J/^\circ C$
G	$2.60 W/^\circ C$	$2.396 W/^\circ C$

The estimated parameters, C and G , were used in a dynamic simulator to estimate power consumption and temperature. The measured power and temperature are depicted in Figure 2.5 along with the simulation results. As can be seen in this figure, simulation results of both methods are very close to the measured profiles. However, the physical parameters obtained from scenario I result in more accurate simulated power consumption and temperature.

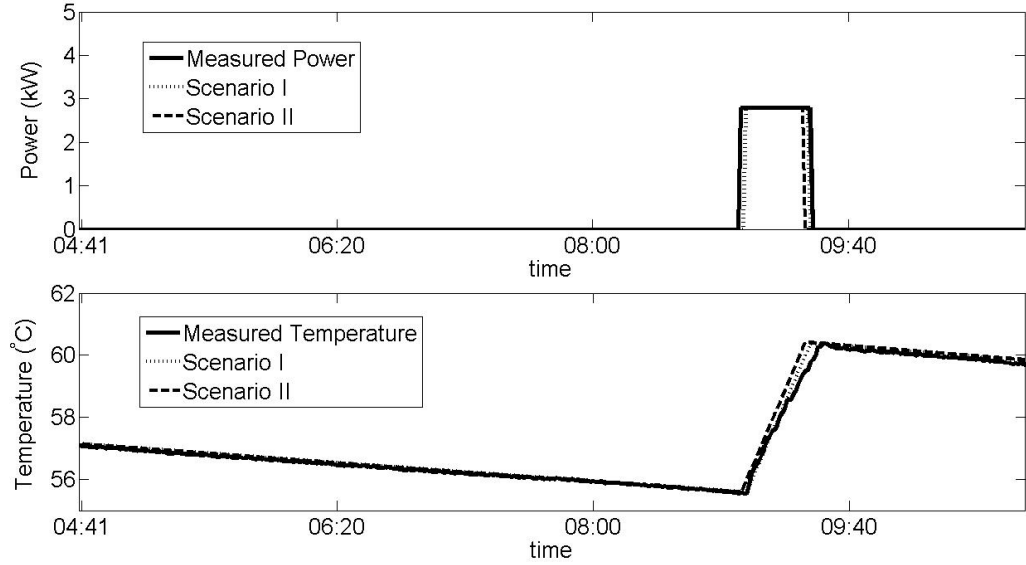


Figure 2.5: Comparison of measured parameters and simulation results

2.5.3 Water Usage Estimation

Various experiments were conducted on the instrumented testbed to validate the proposed water-usage estimation method. Figure 2.6 depicts the graphical user interface of the program that was developed to analyze the measurements obtained from the testbed. The measured data of each experiment can be stored in a database through this software. Then, further analysis, such as parameter identification and water usage estimation under different scenarios, can be performed. The source code to estimate water usage through scenario I is listed in Appendix A.2 and the water usage estimation source code from scenario II is listed in Appendix A.3.

Among many experiments, one of them is illustrated in Figure 2.7. In this experiment, 60 liters of water was drawn from the water heater in 2 minutes. This action caused a drop in the temperature, and the water heater turned on to heat up the water.

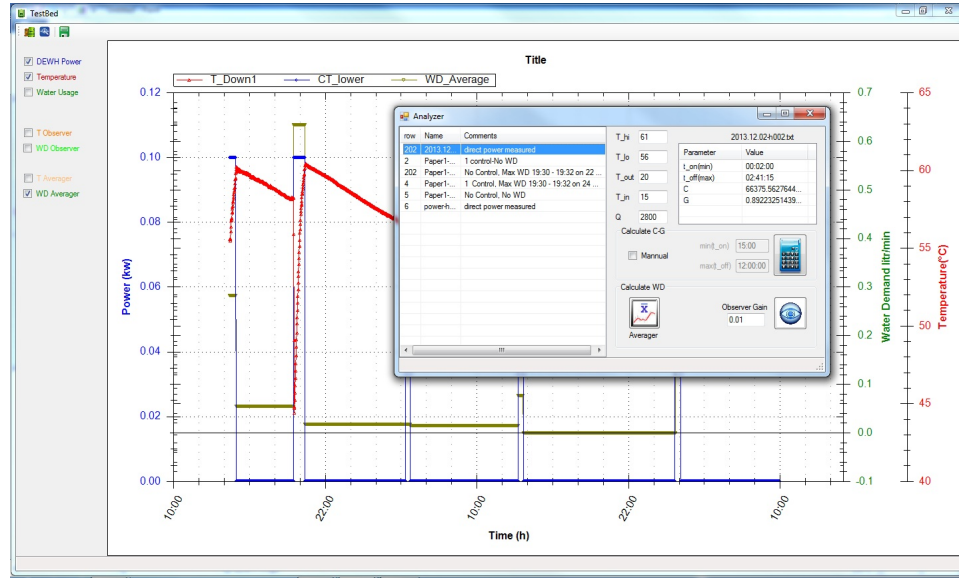


Figure 2.6: The GUI of the application to analyze the testbed data

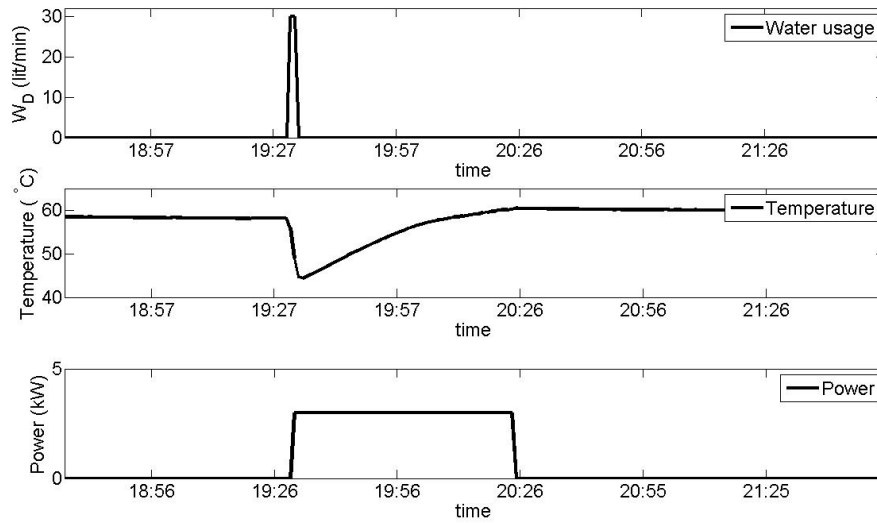


Figure 2.7: Measurements on testbed: water usage, temperature, and power consumption

The water usage was estimated, as described in Section 2.3. For scenario I, the measurements of both power and temperature were used. For the second scenario, only the power profile was used. The obtained water usage is depicted in Figure 2.8.

As can be seen in Figure 2.8, none of the proposed methods produce an exact estimation of the water usage. This estimation error occurs for various reasons such as the assumption of uniform temperature in the tank. However, since the model is based on energy conservation, the amount of consumed energy in the hot water (or equivalently, the volume of the hot water which is equal to the area below the water usage rate) would be estimated correctly. The actual volume of the consumed water and the estimated volumes through each scenario is depicted in Figure 2.9. As can be seen in this figure, the observer method (scenario I) for this experiment estimated 58.2 liters and the average method (scenario II) estimated 56.4 liters while the actual volume was 60 liters.

This experiment also shows that in comparison to the average method, the observer model provides a better estimation of the water usage for both rate and volume because it uses more information (power consumption and water temperature). On the other hand, in many cases, such as this pilot project (PSA), the temperature data is not available, but the power consumption can be easily obtained; therefore, the average method (scenario II) is still useful.

2.5.4 Sensitivity Analysis

Several constant parameters such as ambient temperature and thermostat set points are assumed to be known during the estimation process. However, in a real pilot project such as PSA, these parameters may not be identical and vary from house to house. In this section, effects of the variation in these variables on the modeling process will be studied.

The experiments have 5 steps:

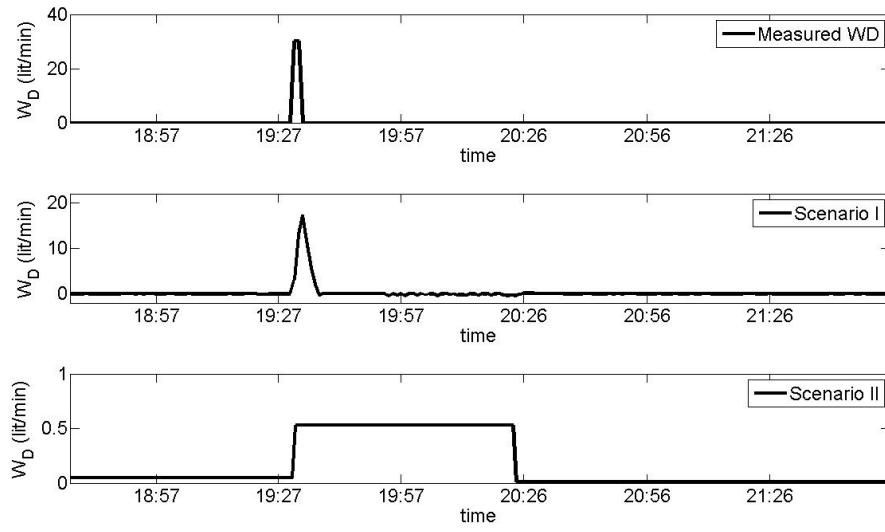


Figure 2.8: Comparison of the rate of water usages, measured rate; estimated by scenario I, and II

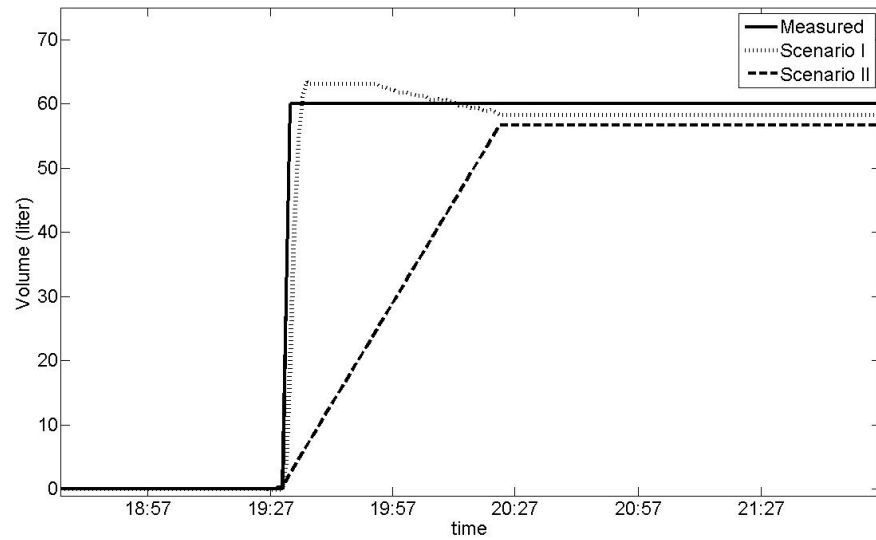


Figure 2.9: Comparison of the volume of water usage, measured volume, estimated by scenario I, and scenario II

1. record the sensor measurements (including ambient temperature, thermostat set-point, DEWH power consumption, and water temperature)
2. use the methods presented in Sections 2.2 and 2.3 to estimate the DEWH model parameters and water demands
3. feed the estimated parameters to the simulator and record the simulated power consumption and temperature
4. change the values of the fixed parameters and repeat steps 2 – 3
5. repeat steps 1 – 4 by applying a turn-off command

Table 2.2 lists the calculated physical parameters for each test. The estimated physical parameters ($\hat{C}$ and $\hat{G}$) are used to estimate the water usage profile. These values (physical parameters and water usage) were used by the simulator to regenerate power consumption and water temperature of a DEWH. The results are depicted in Figure 2.10. The indexes of the curves represent the test cases of the Table 2.2.

Table 2.2: Different test cases for sensitivity analysis

Test cases	$T_{out}(^{\circ}C)$	$\{T_{hi}, T_{lo}\}(^{\circ}C)$	$\hat{C}(J/^{\circ}C)$	$\hat{G}(W/^{\circ}C)$
1 - Actual and Nominal Values	22	$\{60, 56\}$	9.18×10^5	2.39
2 - 10% uncertainty in T_{out}	19	$\{60, 56\}$	9.18×10^5	2.21
3 - 10% uncertainty in setpoints	22	$\{54, 50\}$	9.18×10^5	2.61

As can be seen in Figure 2.10, the uncertainties in ambient temperature have no noticeable impact on the estimated water usage, temperature, or power consumption. But the variations in the thermostat set-point result in larger variations in water demand and temperature estimation; however, the power consumption remains the same.

Figure 2.11 illustrates the same comparison when an external control signal was applied. As can be seen in this figure, the power consumption of all three cases are similar. Since the thermostat set-point in the third test case is lower, the temperature obtained from the simulator is lower than the first two test cases.

These results show that small variations in the physical parameters have no significant impact on the power consumption output of the simulator; thus, it is meaningful to use a fixed value for these parameters for all DEWHs in the pilot project.

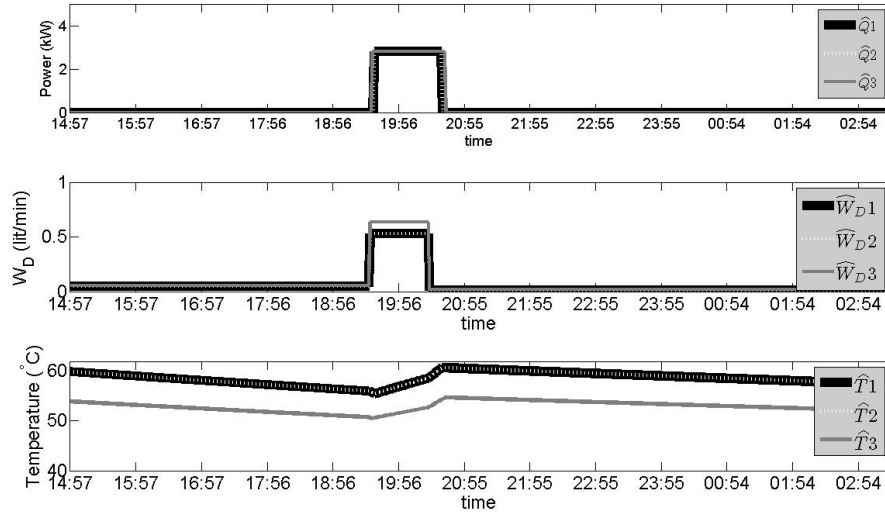


Figure 2.10: Impact of parameter uncertainties, as listed in Table 2.2, on uncontrolled simulation

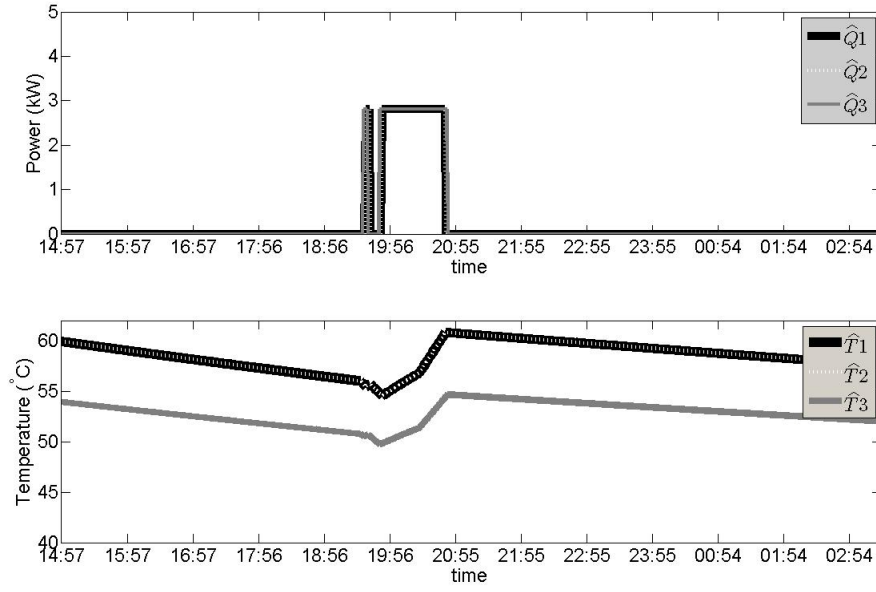


Figure 2.11: Impact of parameter uncertainties, as listed in Table 2.2, on controlled simulation

2.5.5 Simulating Control Signal on Single DEWH

As described in Section 2.4, the simulator uses measured data to estimate the model inputs and parameters and then simulates the model in the presence of a control signal (Figure 2.3).

Two sets of experiments have been designed to validate the simulation results of 1) a single DEWH and 2) aggregated DEWHs. This section will describe the test cases used to validate the simulation of the single DEWH, and Section 2.5.6 describes the test cases for the aggregated DEWHs.

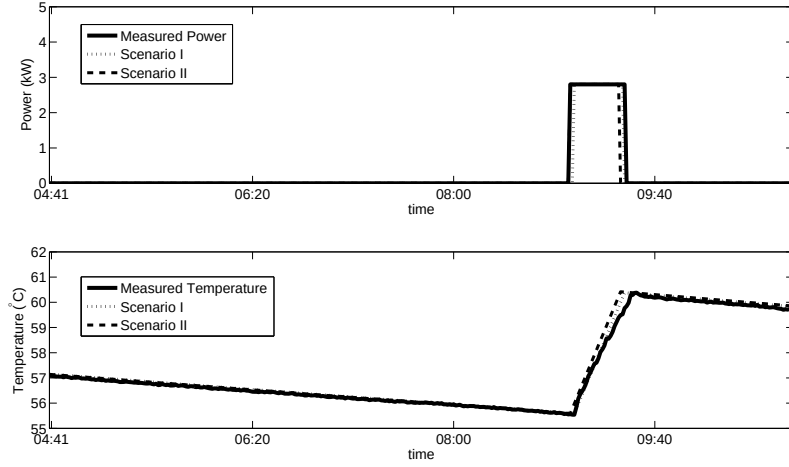


Figure 2.12: Comparing simulation results with the measured power and temperature when $W_D = 0$ with no control signal (Case 1)

2.5.5.1 Case 1: zero water usage, no control action

In this case, zero water was used from the water heater; thus the changes in the temperature were governed by the initial water temperature, ambient temperature, physical parameters and incoming power. Knowing the initial conditions and the physical parameters, the simulator generates a similar pattern for the water temperature and the DEWH load as the actual water heater, as illustrated in Figure 2.12.

2.5.5.2 Case 2: zero water usage with control action

In this case, no water was used from the water heater. A 10-minute turn off signal was sent to the water heater 5 minutes after it was turned on by its internal thermostat, as illustrated in Figure 2.13.

As expected, the temperature started dropping after the heater was turned off by the controller, and the decrease in temperature estimated by the simulator was very close to the measured temperature drop.

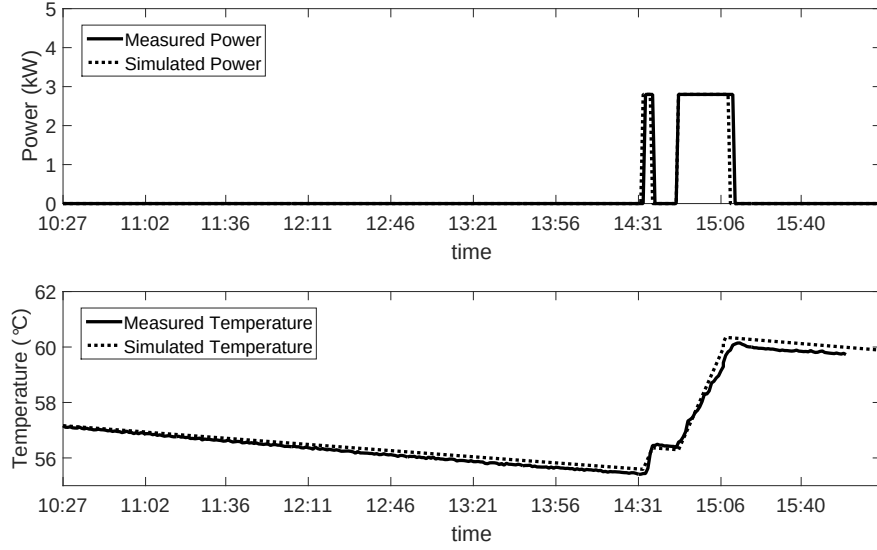


Figure 2.13: Comparison of simulation results with the measured power and temperature when $W_D = 0$ and 10 minutes off-command (Case 2)

2.5.5.3 Case 3: non-zero water usage, no control action

In this case, a specific amount of water (in this experiment, 60 liters in 2 minutes) was drawn from the water heater, and the DEWH temperature and power consumption were used to estimate the water rate. This estimate was used as an input to the simulator for generating the temperature and power consumption. Figure 2.14 shows both simulation methods provide good estimates of the power consumption profile. W_D obtained by the state observer results in a better estimation of the temperature, because more information was used as inputs.

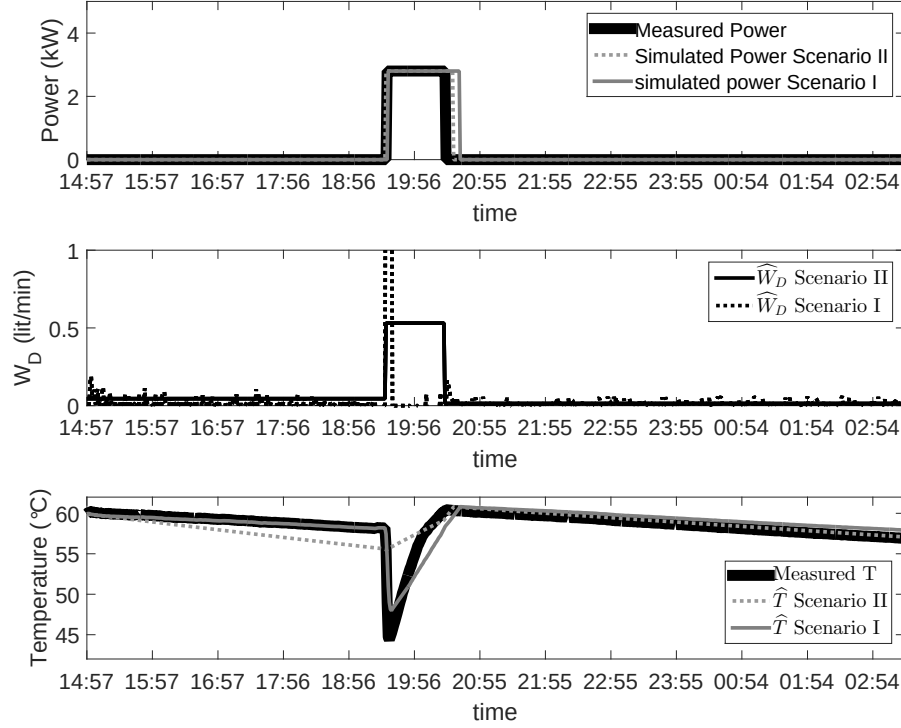


Figure 2.14: Comparison of the simulation results with the estimated W_D obtained from the proposed methods, with no control signal (Case 3)

2.5.5.4 Case 4: non-zero water usage with control action

In this test case, a 10-minute turn off command was sent to the water heater 5 minutes after it turned on. The water demand obtained from the test case 3 was used as the input of the simulator, and the power consumption and water temperature were estimated. The same pattern for water drawn was also used in the experiment in order to have identical conditions. Like the simulation, the DEWH was turned off 5 minutes after it turned on.

The simulation results along with the measured data are depicted in Figure 2.15. As can be seen in this figure, the power consumption obtained by both W_D profiles closely match the actual measured power consumption; however, the error in

temperature estimation is large. The reason for the large error in temperature is the assumption of a fixed water usage rate during the on/off interval. The power consumption profile is the important parameter in this project, which makes this method useful.

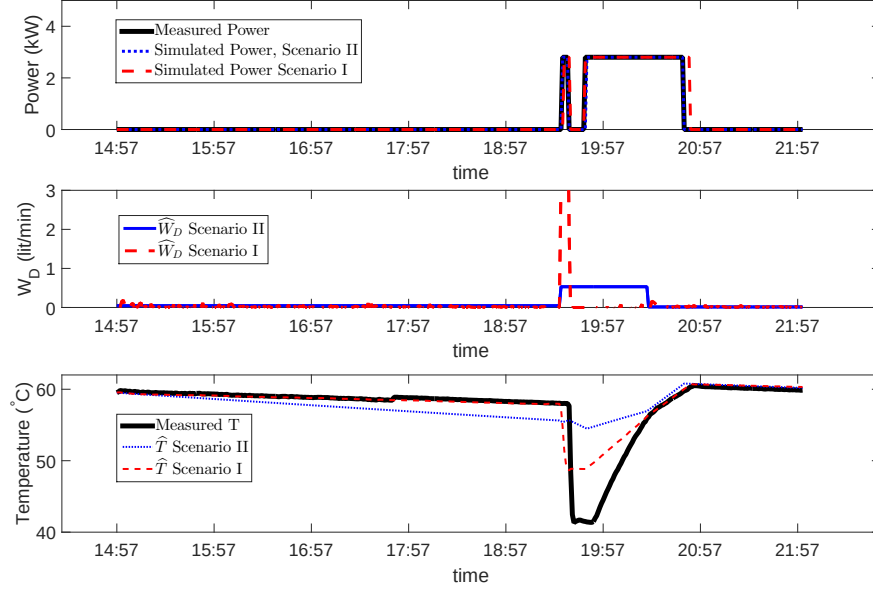


Figure 2.15: Comparison of the simulation results with the estimated W_D obtained from the proposed methods, with a 10-minute turn-off control signal

2.5.6 Simulating Control Action on Multiple DEWHs

Previous experiments showed that the estimated water usage profiles, through the proposed methods, may be used in the simulator to study the impact of control actions on the DEWH power consumption. The power consumption obtained by the simulator, with or without control action, resembles the actual dynamic response of the DEWH.

In this set of experiments, the power consumption of 96 DEWHs was collected and applied as input to the simulator. The goal of these experiments were to compare the aggregated results of the simulator with the actual measurements. In the first

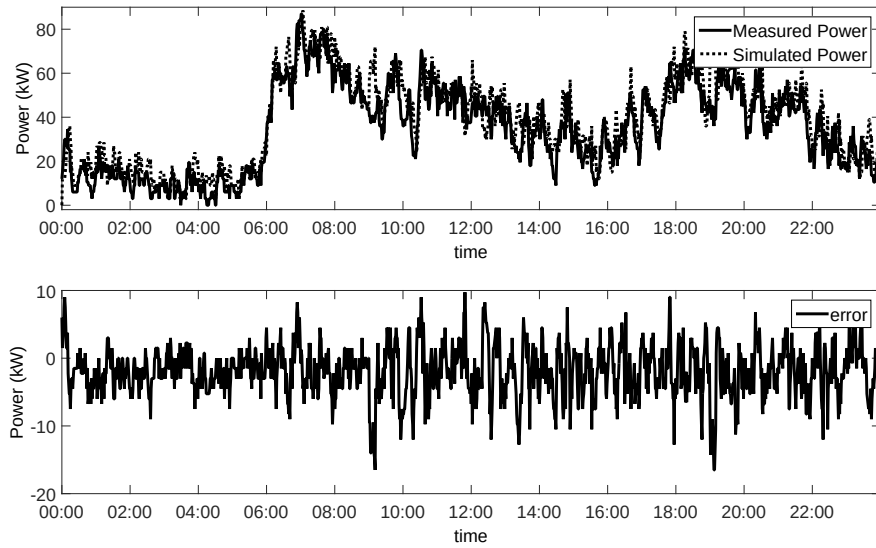


Figure 2.16: Comparison of the aggregated simulation result and the aggregated measured load with no control signal

experiment, the simulation results were compared with the actual measurements without any control action. The impact of the control action on the aggregated load was studied in the second experiment.

2.5.6.1 Case 1: no control action

The power consumption of 96 water heaters on the pilot project was obtained through smart meters. Using the methods described in Section 2.3, a water usage profile for each DEWH was estimated.

Figure 2.16 shows the summation of the actual measured DEWH load with the summation of the individual simulated DEWHs over a day along with the error. The mean absolute error (MAE) for the whole day is $6.5kW$ with an average relative error of 7.4%. As can be seen in this figure, the simulated aggregated load profile follows the same profile as the measured aggregated load.

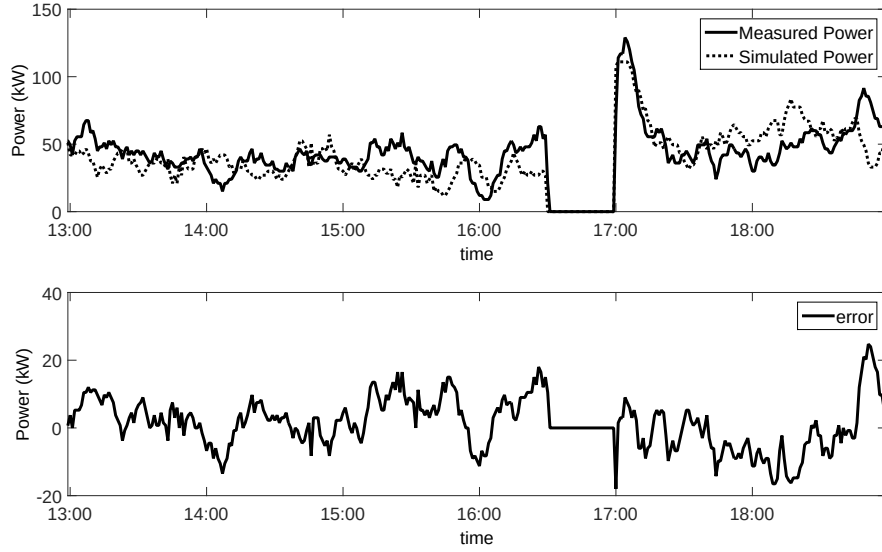


Figure 2.17: Comparison of the aggregated simulation result and the aggregated measured load with a 30 minute turn off command

2.5.6.2 Case 2: with control action

To test the impact of the control action, all water heaters were turned off for 30 minutes at a certain time and the payback effect was observed in the simulator, as illustrated in Figure 2.17.

To evaluate this simulation result, all water heaters of the pilot project were turned off at the same time for 30 minutes. Though the initial conditions and the water usage patterns were not exact, the total aggregated load follows the same pattern. The amount of the peak observed in the experiment agrees with the estimated simulation result. The mean absolute error of this simulation is $9.8kW$ with an average relative error of 9.7%.

3 Load Forecast

Using TCA devices for load following can be problematic due to the internal interlock control mechanism, which limits the load following capacity. Therefore, an estimation of the future status of the load is required for designing a viable load following controller.

An introduction to the load forecast in the load control context is presented in Section 3.1. Section 3.2 shows the methods used for forecasting the power consumption of the individual controllable loads, and the experimental results are presented in Section 3.3. The aggregated load forecast method is described in Section 3.4, followed by the experimental results in Section 3.5.

3.1 Introduction

In a load following DLC, the set points (a profile of the desired aggregated load in the future) should be within an achievable range. This range can be estimated by forecasting the future power consumption of the load. Thus, a reliable load forecast is a key parameter in a direct load control program. However, the load forecast in a load control program is more challenging than a load forecast without direct load control due to the payback effect of the control actions. In this section, the challenges of the problem and approaches for the load forecast methods are presented.

3.1.1 Payback Effect

Any control action in this project changes the power consumption profile of the load. A shed command by the controller reduces the power consumption level, but due to consideration of customers' comfort, the controller should turn the DEWH back on at a later time, before the water temperature falls too low. Therefore, the control action is just a tool to shift the load to a later time. This shift in time, which is a change in the shape of the load profile, is known as the payback effect of the control action [17].

Modeling the payback effect is important because it changes the power consumption profile. The payback model is required for two reasons:

1. to estimate the effect of any control action on the previously forecasted load profile for optimization purposes, and
2. to prepare the controlled historical load profile to be used for an uncontrolled load forecast.

3.1.2 Single Load Forecast

The total aggregated load forecast can be created by adding up the individual load forecasts. Having an estimation of the future power consumption of each individual water heater enables the controller to schedule each DEWH consumption in order to follow the desired load level. However, the error of each individual forecast will add up when aggregating the forecasts, which may result in a large error.

Section 3.2 describes the methods used in this project to generate individual load forecasts. A mathematical model for the aggregated error will also be presented in this section. The experimental results are discussed in Section 3.3.

3.1.3 Aggregate Load Forecast

The aggregated load forecast can also be generated directly from the aggregated power consumption. In this method, the power consumption of the individual water heaters are used to shape an aggregated load profile. This load profile will be used to create an aggregated load forecast. Therefore, in this method, the forecasts of individual consumption are not available.

Section 3.4 presents the methods to generate and enhance the aggregated load forecast. An aggregated payback model is also introduced in this section. Experimental results of these methods are presented in Section 3.5.

3.2 Single DEWH Load Forecast

In order to follow a trajectory of the power consumption, the controller needs to know which loads tend to consume power during that specific interval and, among these loads, which loads can be stopped. Generating a load forecast for each individual load can help the controller in its task.

This section presents two methods (a neural network and a statistical-based method) to generate the individual load forecast, and a simple payback model of the control action on each individual load will be discussed. At the end of this section, an analytical model for the aggregation error will be presented.

3.2.1 Load Forecast Using ANN

Artificial neural networks (ANN) are widely used to generate a load forecast [39]. The author also developed an ANN to forecast the power consumption of an individual load and to compare the performance of the two methods.

The inputs of the neural network constitute previous $N_1 = 96$ samples (past 4 hours of the same day and past 4 hours of the previous day, with 5 minutes sampling time) of the load. The output layer contains $N_2 = 12$ samples (next 1 hour) of the forecasted load. A multi-layer Perceptron (MLP) artificial neural network with 20 neurons in the hidden layer was used and trained using the Levenberg-Marquardt training algorithm [73]. Figure 3.1 illustrates the structure of the neural network and each neuron.

The main advantage of this method is its simplicity, easy implementation, and fast response. Another advantage of this method is its capability to learn the behavioral patterns of each individual end-use device. One of the disadvantages is the neural network's need for online training for enhancing its precision and for adapting to the changes in the system's behaviour. Training is a time-consuming process that may limit the real-time operation. In addition to that, if the training parameters are not chosen carefully, the training may not converge properly.

3.2.2 Load Forecast Using Statistic-based Method

Instead of forecasting the on/off state of the DEWH at each time instance, this method forecasts the duration of the on/off *intervals*. An example is illustrated in Figure 3.2. As can be seen in this figure, a forecast is generated at time k_0 . At this time, the DEWH was off for $t_s^0 = \alpha$ time units. This method estimates that the DEWH will stay off for $\hat{t}_s$ and therefore will turn back on at time $(k_0 - t_s^0) + \hat{t}_s$.

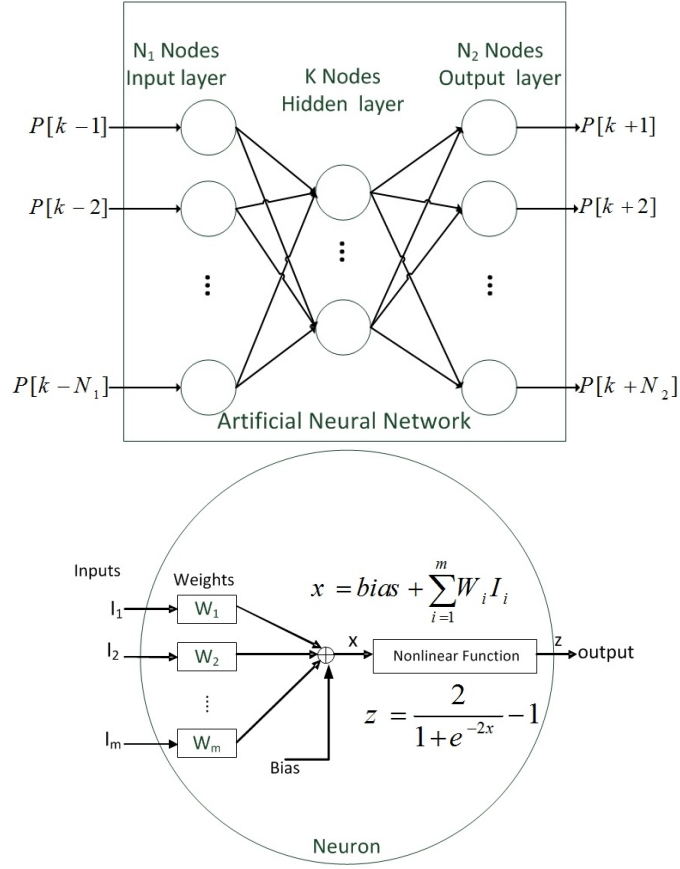


Figure 3.1: Structure of the neuron and ANN.

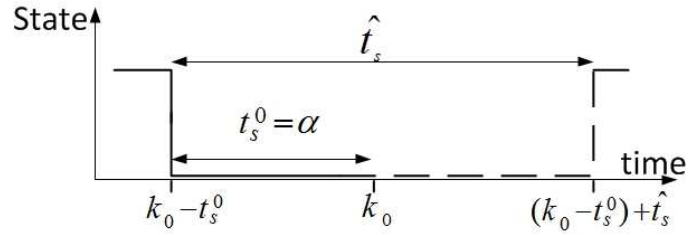


Figure 3.2: At time k_0 , the status of DEWH is unchanged for $t_s^0 = \alpha$ and it is expected to stay unchanged for a total of $\hat{t}_s$.

Let t_s^0 denote the time interval where the state of the DEWH is unchanged at the time of generating the forecast. Also, let $\hat{t}_s$ denote the total time that the state of the DEWH stays unchanged. At time k_0 , the expected value of $\hat{t}_s$, knowing that it was already unchanged for α time units, can be calculated using conditional probability:

$$\begin{aligned}
E\{\hat{t}_s | t_s^0 = \alpha\} &= \sum_{m=0}^{m=t_s^{max}} m P\{\hat{t}_s = m | t_s^0 = \alpha\} \\
&= \sum_{m=\alpha}^{m=t_s^{max}} m \frac{P\{\hat{t}_s = m, t_s^0 = \alpha\}}{P\{t_s^0 = \alpha\}} \\
&= \sum_{m=\alpha}^{m=t_s^{max}} m \frac{P\{\hat{t}_s = m\}}{P\{t_s^0 = \alpha\}}
\end{aligned} \tag{3.1}$$

where t_s^{max} denotes the maximum time that DEWH remains in the given state and $P\{\hat{t}_s = m\}$ is the probability that the DEWH interval is equal to m . This probability function may be different from house to house and can be measured in the pilot project.

Figure 3.3 shows the probability distribution function of the *on* and *off* intervals of a single DEWH in the PSA pilot project. A similar PDF can be obtained for any DEWH in the pilot.

Since it is expected that the on/off state of the DEWH will not change during the interval $\hat{t}_s$, the forecasted state of the DEWH at each sampling time in this interval can be obtained by using (3.2).

$$\hat{s}(k|k_0) = s(k_0) \text{ for } k_0 < k < k_0 - \alpha + \hat{t}_s \tag{3.2}$$

Note that the forecasted load can be updated at each sample by receiving new updates from the DEWH.

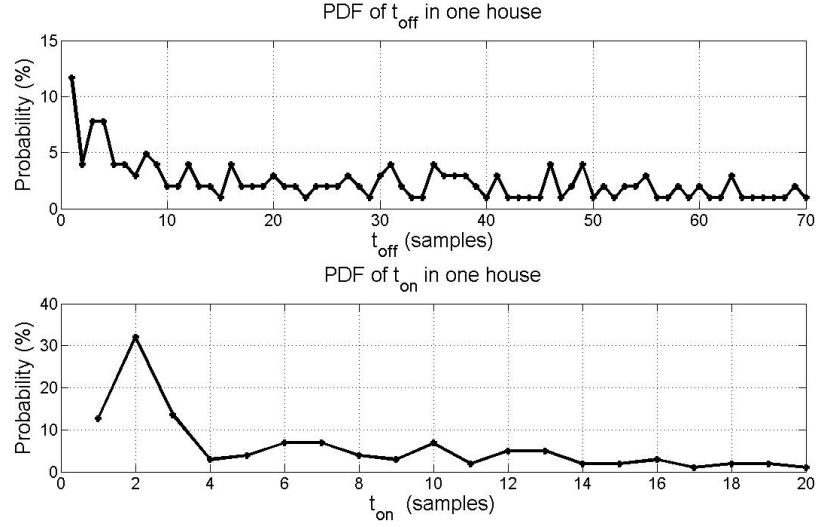


Figure 3.3: The probability distribution function of the On and Off interval of a single house

3.2.3 Payback model

The idea behind the load control program is to shift the time of power consumption to a more *appropriate* moment. This means that the load should be interrupted at the non-appropriate time and then turned back on at a later time while the power disruption would not violate the customer's comfort.

The payback effect is the impact of this control action (turning off and on) on the behaviour of the power consumption of the device. This impact can be formulated either as the change in the amount of energy consumption, or using the time when the energy is consumed.

Figure 3.4 shows the power consumption of a DEWH in two cases: 1) normal operation with no external control action and 2) a disruption of the consumption for a duration of t_d . As depicted in Figure 3.4, before applying the control signal at time t_0 , the uncontrolled load profile and controlled load profile are exactly the same.

Let us define some variables in order to formulate the payback:

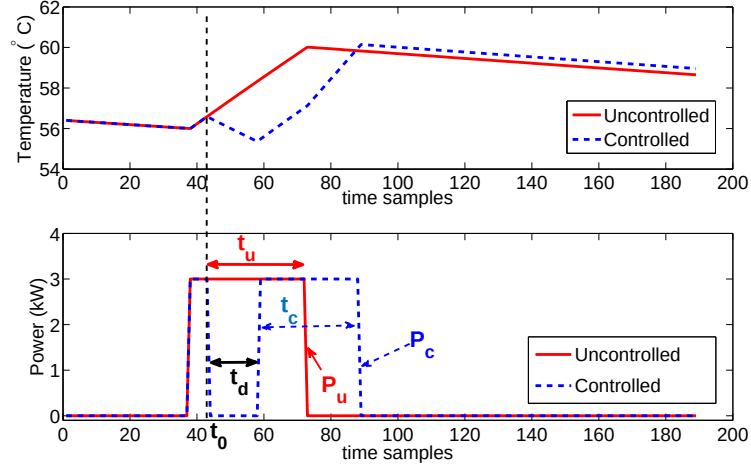


Figure 3.4: Payback effect of a control action on a single DEWH

t_d : the disruption time which is the time interval when the controller turns off the load (s)

P_c : the controlled power consumption profile (W)

P_u : the uncontrolled power consumption profile (W)

E_c : the amount of energy consumption of the load, after the controller turns the load back on (Ws)

E_u : the amount of energy consumption of the load, if there were no control action (Ws)

t_c : the time interval that the load consumes energy after the control action (s)

t_u : the time interval that the load consumes energy with no control action (s)

$$E_{pb} = E_c - E_u \quad (3.3)$$

Since $E_c = P_0 t_c$ and $E_u = P_0 t_u$, then (3.3) can be written as

$$E_{pb} = P_0(t_c - t_u) \quad (3.4)$$

According to (3.4), the impact of a control action may result in more or less energy consumption if $t_c \neq t_u$. Therefore, it is useful to study the effect of a control action on the power consumption of the DEWH.

The power consumption of a DEWH, as described in Chapter 2, is a function of its water temperature. The water temperature is also a function of water usage.

For a specific power profile, as depicted in Figure 3.4, numerous water profiles can be obtained, as discussed in Section 2.3. In order to simplify the model, let us consider a water usage which only draws water before t_0 . Let T_0 denote the water temperature at the start of control action at time t_0 , and T_1 denote the water temperature at the end of control action at time $t_0 + t_d$.

Using the time domain DEWH model presented in Chapter 2, for the uncontrolled load profile, the temperature will reach to T_{hi} from T_0 within t_u as follows:

$$T_{hi} = T_0 e^{-t_u * G/C} + (T_{out} + P_0/G)(1 - e^{-t_u * G/C}) \quad (3.5)$$

The uncontrolled time can be obtained from

$$t_u = \frac{C}{G} \ln\left(\frac{T_{out} + P_0/G - T_0}{T_{out} + P_0/G - T_{hi}}\right) \quad (3.6)$$

Similar equations can be written for the controlled profile. The only difference is the initial temperature, T_1 , which is slightly lower than T_0 due to the leakage of the tank body.

$$t_c = \frac{C}{G} \ln\left(\frac{T_{out} + P_0/G - T_1}{T_{out} + P_0/G - T_{hi}}\right) \quad (3.7)$$

When there is no water usage, it takes 10 to 12 hours for the temperature to drop around $4^\circ C$ from T_{hi} to T_{lo} . However, the disruption time, t_d , is around 15 minutes;

thus, the temperature drops through body loss is negligible during this time interval. It means that $T_0 \cong T_1$; therefore,

$$t_c \cong t_u \tag{3.8}$$

This equation suggests that the effect of a control action is shifting the consumption in time. The amount of energy consumption may not change significantly.

In order to be more precise, a simulation was executed with different water usage profiles and different disruption times. Simulation results show that (3.8) is valid. The curves depicted in Figure 3.4 are obtained from the simulator with the average method to estimate W_D .

Simulation results were also verified with several experiments on the test-bed under controlled initial conditions. All experimental results also show that the control action is simply shifting the load without making any significant change to the energy consumption.

3.2.4 Aggregating Single Load Forecast

As mentioned earlier in 1.1, an aggregated load forecast is required in order to produce a desired aggregated load. Therefore, the load aggregator should also create an aggregated load forecast. In this section, the effect of the forecast error of each individual load on the aggregated error will be studied.

Let $P_t(k)$ denote the total power consumption at time k and $\hat{P}_t(k|k_0)$ denote the forecasted total power at time k generated at time k_0 . These parameters can be

formulated as

$$P_t(k) = \sum_{i=1}^N (s^i(k) P_0^i) \quad (3.9)$$

$$\widehat{P}_t(k|k_0) = \sum_{i=1}^N (\widehat{s}^i(k|k_0) P_0^i) \quad (3.10)$$

where N is the total number of DEWHs, P_0^i is the nominal power rating of the i^{th} DEWH, $s^i(k)$ is the on/off status of the i^{th} DEWH at time k , and $\widehat{s}^i(k|k_0)$ is the forecasted on/off status of the i^{th} DEWH at time k generated at time k_0 .

Let $\tilde{s}^i(k|k_0)$ denote the estimation error of the i^{th} DEWH at time k which is generated at time k_0 . Then, it can be formulated as

$$\tilde{s}^i(k|k_0) = s^i(k) - \widehat{s}^i(k|k_0) \quad (3.11)$$

The relative absolute error of the aggregated load can be defined as

$$e_R = \frac{|P_t(k) - \widehat{P}_t(k|k_0)|}{P_t(k)} = \frac{\sqrt{(P_t(k) - \widehat{P}_t(k|k_0))^2}}{P_t(k)} \quad (3.12)$$

The expected value of relative error (e_R) can be formulated as

$$E\{e_R\} = \frac{\sqrt{E\{(P_t(k) - \widehat{P}_t(k|k_0))^2\}}}{E\{P_t(k)\}} \quad (3.13)$$

To simplify the equations, the following assumptions have been made:

1. $P_0^i = P_0$ for all i . This means that nominal power rating of all DEWHs are equal. In practice, most of the water heaters in a region are similar.

2. $E\{s^i(k)\} = \lambda_k$ for all i . This means that it is expected that water heaters would have similar behaviour.
3. $E\{(\tilde{s}^i(k|k_0))^2\} = \gamma^2$ for all i .
4. $E\{\tilde{s}^i(k|k_0)\} = \eta$ for all i . These means that the estimation error for all houses are similar.
5. The estimation error of DEWHs are uncorrelated, which means that $E\{\tilde{s}^i(k|k_0)\tilde{s}^j(k|k_0)\} = E\{\tilde{s}^i(k|k_0)\}E\{\tilde{s}^j(k|k_0)\}$ for any i, j .

These assumptions indicate that the expected state of the DEWH for all houses are similar. Also, the forecast error for each DEWH is consistent with the other DEWHs. These assumptions were also validated in this project.

Based on these assumptions, the expected value of the total power consumption is:

$$\begin{aligned}
E\{P_t(k)\} &= E\{\sum_{i=1}^N (s^i(k)P_0^i)\} \\
&= E\{P_0 \sum_{i=1}^N s^i(k)\} \\
&= P_0 \sum_{i=1}^N E\{s^i(k)\} \\
&= NP_0\lambda_k
\end{aligned} \tag{3.14}$$

and:

$$\begin{aligned}
P_t(k) - \hat{P}_t(k|k_0) &= \sum_{i=1}^N (s^i(k) - \tilde{s}^i(k|k_0)) P_0^i \\
&= P_0 \sum_{i=1}^N \tilde{s}^i(k|k_0)
\end{aligned} \tag{3.15}$$

Therefore

$$\begin{aligned}
& E\{(P_t(k) - \widehat{P}_t(k|k_0))^2\} \\
&= E\left\{\left(P_t(k) - \widehat{P}_t(k|k_0)\right)\left(P_t(k) - \widehat{P}_t(k|k_0)\right)\right\} \\
&= E\left\{P_0^2 \sum_{i=1}^N \tilde{s}^i(k|k_0) \sum_{j=1}^N \tilde{s}^j(k|k_0)\right\}
\end{aligned}$$

This can be rewritten as

$$\begin{aligned}
& E\{(P_t(k) - \widehat{P}_t(k|k_0))^2\} \\
&= P_0^2 \sum_{i=1}^N E\{(\tilde{s}^i(k|k_0))^2\} \\
&\quad + P_0^2 \sum_{i=1}^N \sum_{j=1, j \neq i}^N E\{\tilde{s}^i(k|k_0) \tilde{s}^j(k|k_0)\} \\
&= P_0^2 \left(\sum_{i=1}^N \gamma^2 + \sum_{i=1}^N \sum_{j=1, j \neq i}^N \eta^2 \right) \\
&\Rightarrow E\{(P_t(k) - \widehat{P}_t(k|k_0))^2\} = P_0^2 (N\gamma^2 + N(N-1)\eta^2) \tag{3.16}
\end{aligned}$$

Now by substituting (3.14) and (3.16) in (3.13), the expected value for the aggregated relative error can be obtained:

$$\begin{aligned}
E\{e_R(k|k_0)\} &= \frac{\sqrt{E\{(P_t(k) - \widehat{P}_t(k|k_0))^2\}}}{E\{P_t(k)\}} \\
&= \frac{\sqrt{P_0^2 (N\gamma^2 + N(N-1)\eta^2)}}{NP_0\lambda_k} \\
&= \sqrt{\frac{P_0^2 N\gamma^2 + P_0^2 N(N-1)\eta^2}{N^2 P_0^2 \lambda_k^2}} \\
&= \sqrt{\frac{\gamma^2 + (N-1)\eta^2}{N\lambda_k^2}} \tag{3.17}
\end{aligned}$$

It can be inferred (3.17) that if N is sufficiently large, then

$$\lim_{N \rightarrow \infty} E\{e_R(k|k_0)\} = \frac{|\eta|}{\lambda_k} \quad (3.18)$$

3.3 Experimental Results for Single Forecast

The load forecast methods presented in Section 3.2 were implemented in a pilot project with approximately 100 water heaters. The power consumption of each individual load could be obtained through a smart meter installed in that house. This section presents the performance of each of these forecasting methods along with validating the model of the relative error in the aggregation of forecasts.

3.3.1 Evaluation Criteria

At any given time, k_0 , the forecast algorithms generate an estimation of the next L samples of the load profile, which can be modeled as a binary value (1 for on state, 0 for off).

Let $\widehat{s}(k_0 + j|k_0)$ denote the j^{th} forecasted status of the load, generated at time k_0 . Thus, at j sample time after k_0 , the real state of the load can be measured. Let $s(k_0 + j)$ denote the measured state of the load. Thus, the error in the j^{th} forecasted interval at time k_0 can be calculated by comparing the forecasted value with the j^{th} measured value after its generation.

$$\tilde{s}(j) = s(k_0 + j) - \widehat{s}(k_0 + j|k_0) \quad (3.19)$$

Running the algorithm at different times produces various forecast profiles. The number of successful estimations for each sample in the forecast horizon over the

total number of times running the algorithm gives the probability of successful state estimation. The success rate can be formulated as:

$$S_R(j) = Prob\{s(k_0 + j) = \widehat{s}(k_0 + j|k_0)\} \quad \text{for all } k_0 \text{ and } j \in [1..L] \quad (3.20)$$

3.3.2 Evaluation of the Single Load Forecast

The two forecasting methods were applied in a pilot containing approximately 100 electric water heaters. Every 5 minutes, each algorithm creates a load forecast for each individual load for the next 4 hours in 5 minutes intervals. So, each forecast contains 48 samples.

The forecasted loads were analyzed for each individual load over a month. The success rate of each interval in the forecast horizon was calculated for each load. Also, the average error ($E\{\tilde{s}\}$) and the average square error ($E\{(\tilde{s})^2\}$) were also calculated for each forecast interval (See Appendix B.1 for the complete data). These values may be used to estimate the relative error of the aggregated forecast. Table 3.1 lists the performance of each method over the first hour. Figure 3.5 illustrates the success rates of the forecasting methods over the 4 hour forecast horizon.

The success rate indicates the confidence in the forecasted value of each interval. For example, consider the 30 minutes ahead (6^{th} samples) forecast: according to Table 3.1, it is 76% expected that the forecasted value by the ANN-based method be correct, while the probability that the estimated value by the statistic-based method be correct is 84%.

As can be seen in Figure 3.5, for very short-term load forecasts (e.g. less than an hour), the statistic-based method has a better performance than the ANN-based

Table 3.1: Performance evaluation of the forecasting methods for a single load.

index	Statistic-Based			ANN-Based		
	$S_R(j)$	$E\{\tilde{s}\} = \eta$	$E\{(\tilde{s})^2\} = \gamma^2$	$S_R(j)$	$E\{\tilde{s}\} = \eta$	$E\{(\tilde{s})^2\} = \gamma^2$
1	0.97	0.02	0.03	0.78	0.19	0.21
2	0.93	0.04	0.06	0.78	0.19	0.22
3	0.9	0.08	0.1	0.77	0.2	0.22
4	0.9	0.08	0.1	0.76	0.2	0.21
5	0.88	0.09	0.1	0.76	0.2	0.22
6	0.84	0.12	0.12	0.76	0.21	0.24
7	0.82	0.16	0.19	0.75	0.22	0.24
8	0.8	0.18	0.2	0.76	0.22	0.24
9	0.8	0.21	0.22	0.75	0.21	0.24
10	0.78	0.19	0.23	0.74	0.22	0.24
11	0.78	0.2	0.22	0.74	0.25	0.25
12	0.76	0.21	0.22	0.75	0.23	0.25

forecast. However, as the forecasting horizon expands, the success rate of the static-based method drops, and is comparable to the ANN method up to $j \cong 30$.

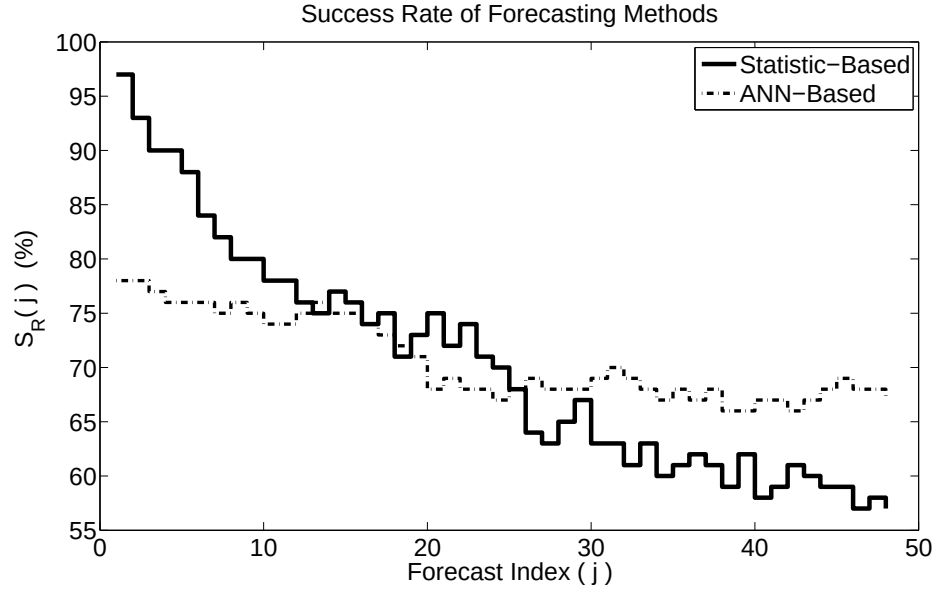


Figure 3.5: The success rate of the forecasting methods over a 4-hour ahead forecast in 5 minutes time intervals.

It can also be observed that the success rate of the ANN-based method is more consistent along the forecasting horizon due to the learning algorithm of the ANN which tries to minimize the error of all outputs.

3.3.3 Aggregated Forecast Error of Uncontrolled Load

To generate an aggregated load forecast, the individual forecasted load profiles may be added together. Then, by comparing the aggregated measured load profile with the aggregated forecasted load profile, the relative error can be calculated, as listed in (3.12).

In order to use Equations (3.17) and (3.18) as an estimate for the relative error in aggregating load forecasts, the following parameters should be computed first:

1. average forecast error for each time interval: $E\{\tilde{s}(k|k_0)\} = \eta$
2. average square error of the individual forecasted load for each time interval:

$$E\{(\tilde{s})^2(k|k_0)\} = \gamma^2$$
3. expected on/off state of the load for each time interval: $E\{s(k)\} = \lambda_k$

The forecast error and the square error of the forecast for each interval in the forecasting horizon may be calculated directly through analysis of the single load forecasts, as listed in Table 3.1. On the other hand, λ_k can be measured by analyzing the power consumption of each individual load and calculating the probability that the load is 'on' over each time interval ($Prob\{s(k) = 1\}$).

$$\begin{aligned} E\{s(k)\} &= \lambda_k = 1 \times Prob\{s(k) = 1\} + 0 \times Prob\{s(k) = 0\} \\ &= P\{s(k) = 1\} \end{aligned} \tag{3.21}$$

Table 3.2: A sample of *ON*-Probability (λ_k) of various houses at various times

time	average (100 loads)	DEWH#1	DEWH#2	DEWH#3	DEWH#4	DEWH#5
06:30:00	0.201	0.219	0.243	0.230	0.228	0.247
06:35:00	0.213	0.217	0.216	0.240	0.252	0.260
06:40:00	0.224	0.232	0.264	0.240	0.250	0.232
06:45:00	0.234	0.257	0.238	0.245	0.280	0.242
06:50:00	0.241	0.246	0.289	0.241	0.280	0.282
06:55:00	0.250	0.272	0.296	0.259	0.263	0.257
07:00:00	0.258	0.301	0.289	0.276	0.284	0.278
07:05:00	0.266	0.287	0.268	0.311	0.313	0.291
07:10:00	0.273	0.312	0.292	0.285	0.293	0.278
07:15:00	0.283	0.295	0.301	0.324	0.284	0.285
07:20:00	0.288	0.315	0.303	0.325	0.297	0.322
07:25:00	0.287	0.333	0.326	0.311	0.309	0.309
07:30:00	0.287	0.319	0.306	0.328	0.314	0.305
07:35:00	0.282	0.292	0.297	0.306	0.294	0.324
07:40:00	0.288	0.304	0.334	0.310	0.297	0.333
07:45:00	0.299	0.329	0.312	0.329	0.335	0.310
07:50:00	0.289	0.293	0.302	0.329	0.290	0.335
07:55:00	0.289	0.337	0.316	0.315	0.301	0.313
08:00:00	0.297	0.299	0.341	0.343	0.337	0.302

These probabilities were calculated for each individual load over the experimental data obtained from the pilot project. Some of the results are listed in Table 3.2 for individual loads. This table contains the λ_k^i for various loads at various times. As can be seen, loads have similar expected state at each interval, which confirms the assumption of using one λ_k for all houses. Figure 3.6 shows the average λ_k for each 5-minutes interval over a day.

As an example, let us focus on aggregating forecast profiles of individual forecasts that were generated at 7 : 00 am. These forecasts were generated for the same experimental pilot which contains approximately 100 DEWHs.

Table 3.3 shows the measured relative error of the power consumptions and the 3rd sample of the load forecasts (7 : 15 am) by aggregating various numbers of houses

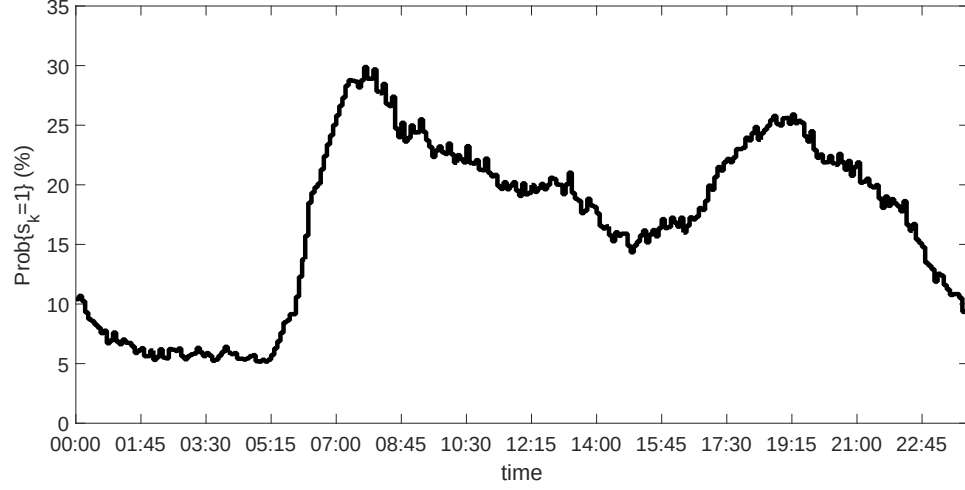


Figure 3.6: The average probability of a load to be *on* (λ_k) at various times.

Table 3.3: Expected value of relative error of forecasted aggregated load for different numbers of loads at 7 : 00am for 15 minutes ahead.

	N	λ_k	η	γ^2	measured	Eq. 3.17	Eq. 3.18
Statistic	10	0.26	0.08	0.10	0.45	0.49	0.31
	20	0.26	0.08	0.10	0.42	0.41	0.31
	50	0.26	0.08	0.10	0.38	0.35	0.31
	100	0.26	0.08	0.10	0.34	0.33	0.31
	500	0.26	0.08	0.10	-	0.31	0.31
ANN	10	0.26	0.20	0.22	0.87	0.93	0.77
	20	0.26	0.20	0.22	0.81	0.86	0.77
	50	0.26	0.20	0.22	0.79	0.81	0.77
	100	0.26	0.20	0.22	0.78	0.79	0.77
	500	0.26	0.20	0.22	-	0.78	0.77

for each forecasting method. The values for λ_k , η , and γ^2 are obtained from Tables 3.1 and 3.2.

As can be seen in Table 3.3, as the number of loads increases, the expected value of the relative error generated by (3.17) and (3.18) gets closer. It also shows that the measured relative error is also close to these estimated values, which confirms the mathematical model for the relative error of the aggregated forecasts.

The mathematical model (3.18) shows that a 5% error in individual state estimations may result in a 20% error in the aggregated load, and as the time horizon extends beyond an hour, the individual estimation error may increase to more than 40%, which results in more than 200% error in the aggregated forecast.

The expected values of the relative error of the aggregated profiles for the other samples in the forecast horizon can be obtained using (3.18), as depicted in Figure 3.7. These curves are obtained using the average errors listed in Table 3.1 and the λ_k from Table 3.2. It can be observed that the aggregation of individual forecast profiles may end up in significant aggregated error, as depicted in Figure 3.8.

Figure 3.8 illustrates the aggregation of individual load forecasts (created at 7 : 00 am). As can be seen in this figure, the statistic-based method produces more accurate results over the first hour.

These experiments show that for long-term load forecasts, aggregating the individual forecasts may not be a practical way to create an aggregated load forecast due to significant error. However, for short-term forecasts, this method gives a good estimation about each individual load that can be used by the controller to schedule each individual load in order to follow the desired aggregated load profile.

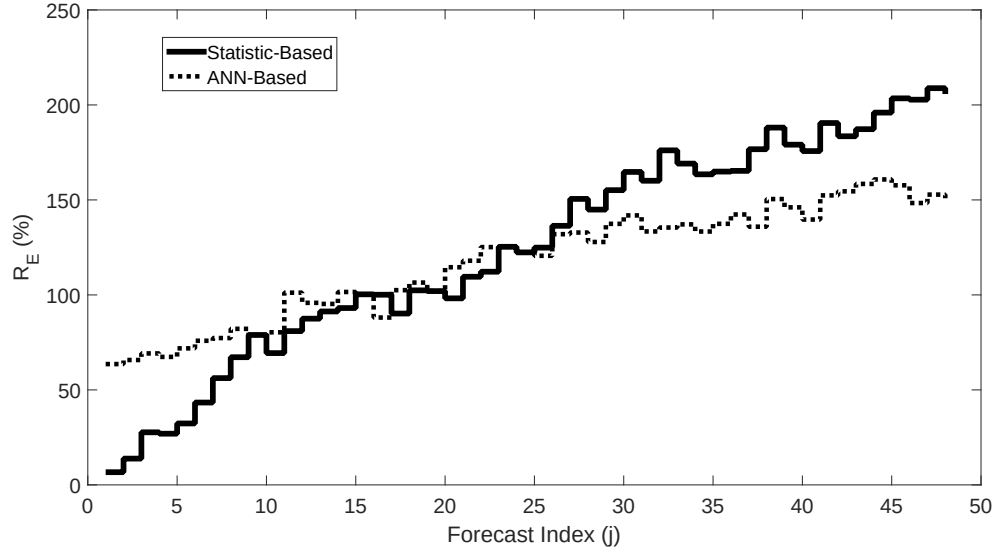


Figure 3.7: A sample comparison of the relative aggregated error of the two forecasting methods.

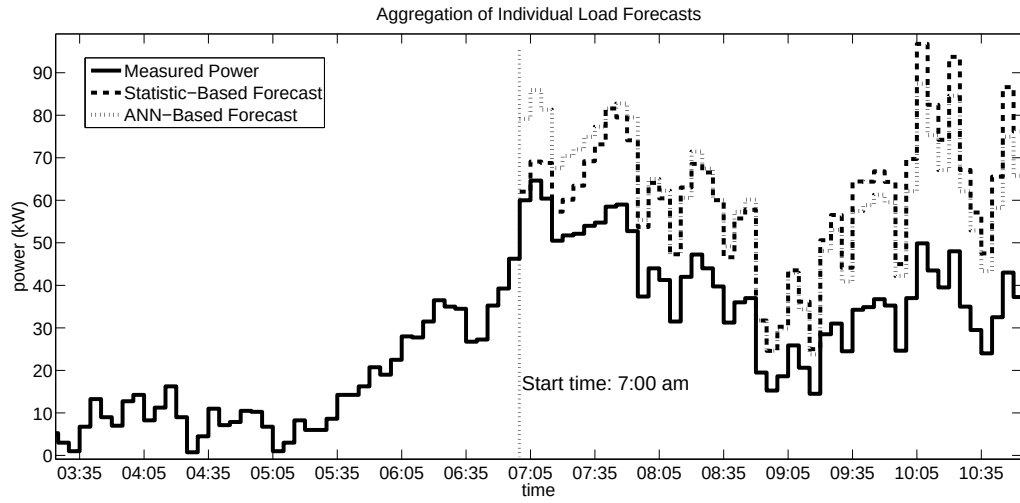


Figure 3.8: The comparison of the aggregation of measured load with the aggregation of forecasting individual loads

3.4 Aggregated Load Forecast

As described in Section 3.3.3, adding up individual load forecasts may result in a significant aggregated forecast error for long-term forecasts. Therefore, another method will be used to create long-term aggregated load forecasts. This section presents a neural network aggregated load forecast with a Kalman filter to reduce its forecast error. Since this method does not generate individual load forecasts, a new aggregated payback model is also developed and presented.

3.4.1 Aggregated Forecast Using ANN

An artificial neural network (ANN) is developed to generate an aggregated load forecast of the controllable load. In this pilot project, information such as weather conditions or ambient temperature are not available; therefore, the input of the proposed ANN is only the previously measured aggregated load profile.

The structure of the ANN is exactly the same as the ANN used in Section 3.2.1, as depicted in Figure 3.1. As can be seen in this figure, the input of the neural network constitutes previous $N_1 = 48$ samples of the aggregated load before k_0 , and its output is $N_2 = 16$ samples of forecasted load. A multi-layer Perceptron (MLP) artificial neural network is used with 20 nodes in the hidden layer.

Figure 3.1 illustrates the structure of a neuron and its internal function. As can be seen in this figure, if the inputs of the neuron are denoted with $I_1, \dots, I_m$, then the output of the neuron is

$$z = \frac{2}{1 + e^{-2x}} - 1 \quad (3.22)$$

where

$$x = bias + \sum_{i=1}^m W_i I_i \quad (3.23)$$

This ANN was trained using the scaled conjugate gradient algorithm (SCG) developed by Moller [74] . It is a very fast training algorithm compared to some other training algorithms such as Levenberg-Marquardt [73]. This feature makes it possible to run the training algorithm more frequently and adapt the neural network to the dynamic behaviour of the customers.

3.4.2 Enhancing ANN Load Forecast Using Kalman Filter

At time k_0 , a short-term forecast packet is generated for the next 4 hours. This packet contains 16 samples in 15-minutes sampling intervals. Let the $e(k_0 + i|k_0)$ denote the forecast error of the i^{th} sample that was created at time k_0 . This error can actually be measured at time $k_0 + i$ by

$$e(k_0 + i|k_0) = P(k_0 + i) - \hat{P}(k_0 + i|k_0) \quad (3.24)$$

If this error is known at generation time, then the actual load at that time can be obtained:

$$P(k_0 + i) = \hat{P}(k_0 + i|k_0) + e(k_0 + i|k_0) \quad (3.25)$$

This value cannot be calculated until the actual measurement at the given time is obtained; however, using Kalman filter [75], an estimation of this error can be calculated. Kalman Filter is named after Rudolph E. Kalman, who published a recursive solution to the discrete-data linear filtering problem in 1960. This filter is essentially a set of mathematical equations that implements a predictor-corrector type estimator that minimizes the estimated error covariance, which is described below.

Let $\hat{e}(k_0 + i|k_0)$ denote the i^{th} sample's estimation error, which is generated at time k_0 . The forecasted load at this time can be modified as

$$\hat{P}_m(k_0 + i) = \hat{P}(k_0 + i|k_0) + \hat{e}(k_0 + i|k_0) \quad (3.26)$$

The observation is that the error of the i_{th} sample can be assumed to be constant with a small variation, which means

$$e(k_0 + i|k_0) = e(k_0|k_0 - i) + w, \quad w \sim N(0, R) \quad (3.27)$$

where w is a normal distributed noise with 0 mean and variance R . Therefore, to estimate the forecast error of the i^{th} sample at time k_0 , the i^{th} sample of the i^{th} forecast ago can be used.

At time k_0 , when the actual measurement for $P(k_0)$ can be obtained, the filter updates its previous measurements using:

$$z = e(k_0|k_0 - i) - \hat{e}^-(k_0|k_0 - i) \quad (3.28)$$

$$G_k = \frac{M^-}{M^- + R} \quad (3.29)$$

$$\hat{e}(k_0|k_0 - i) = \hat{e}^-(k_0|k_0 - i) + G_k z \quad (3.30)$$

$$M = (1 - G_k)M^- \quad (3.31)$$

where M is the matrix of state error, G_k is the Kalman gain, and $\hat{e}^-$ is the pre-measurement estimation. The prediction of the next forecast error is given by:

$$\hat{e}^-(k_0 + i|k_0) = \hat{e}(k_0|k_0 - i) \quad (3.32)$$

$$M^- = M + R \quad (3.33)$$

The modified forecast then will be:

$$\hat{P}_m(k_0 + i|k_0) = \hat{P}(k_0 + i|k_0) + \hat{e}^-(k_0 + i|k_0) \quad (3.34)$$

3.4.3 Payback Model

The forecast method provides an aggregated load forecast with no information about the individual states. Therefore, a new model is required to estimate the impact of a control action on the load forecasts. In Section 3.2.3, the payback effect of a single DEWH was modeled. In this section, the payback effect of a group command on the aggregated forecast will be studied.

To start formulating the system, let us define the following variables:

- N : total number of controllable loads.
- P_0 : nominal power rating of the controllable load.
- s_k^i : the on/off status of the i^{th} controllable load at time k .
- u_k^i : the control signal for the i^{th} controllable load at time k , $u = 1$: internally controlled, $u = 0$: off.
- A_k : a set of the controllable loads that are on at time k , $(\{n \in [1, \dots, N] | s_k^n = 1\})$.
- C_k : a set of the controllable load that are turned off by the controller at time k , $(\{n \in [1, \dots, N] | u_k^n = 0\})$.
- $n(\cdot)$: total number of members in the given set.
- P_k^u : the uncontrolled aggregated power consumption at time k .
- P_k^c : the controlled aggregated power consumption at time k .

I_j^k : denotes the impact of a control action at time k on a future time j ,
 $(j > k)$.

Under no control condition, at time k , several water heaters are consuming energy due to their internal thermostat control. So, it is expected that

$$P_k^u = n(A_k)P_0 \quad (3.35)$$

Let us assume that due to a large communication delay with the AMI devices, the current status of the water heaters is not known. However, the load forecast algorithm provides a reliable estimation of the current level of the load, and so the number of water heaters that are consuming energy can be obtained from (3.35).

If a random set of water heaters (denoted by C_k) is selected to be turned off by the controller, then the amount of reduction on the total aggregated load will be

$$P_k^u - P_k^c = n(A_k, C_k)P_0 \quad (3.36)$$

where $n(A_k, C_k)$ indicates the number of water heaters that were on at time k and are selected to be turned off by the controller. This value may change from 0 to $n(A_k)$.

Let X be a random variable indicating the $n(A_k, C_k)$. Therefore, $P(X = x)$ denotes the probability that exactly x number of selected water heaters are *on*. If $a = n(A_k)$ and $c = n(C_k)$, then it can be shown that the probability of having exactly x water heaters from the selected water heaters to be on is:

$$Prob(X = x) = \frac{\binom{a}{x} \binom{N-a}{c-x}}{\binom{N}{c}} \quad (3.37)$$

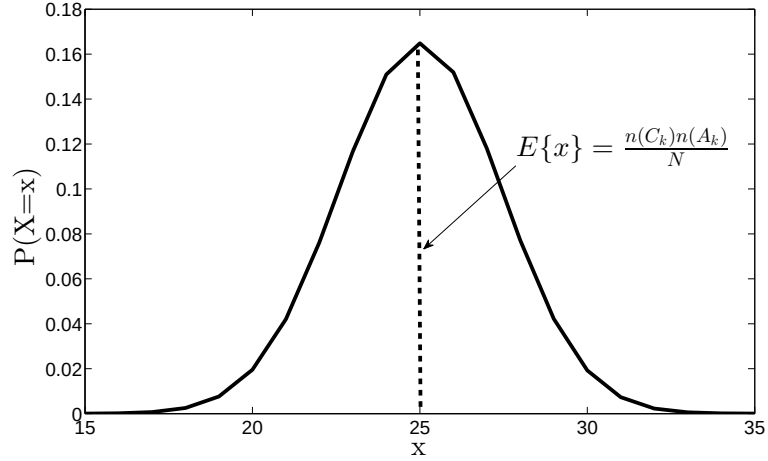


Figure 3.9: The probability distribution function of X

Figure 3.9 illustrates the distribution function of the equation (3.37). As can be seen in this figure, the distribution looks like a normal distribution and it can be shown that the expected value of this random variable is:

$$E\{X\} = E\{n(A_k, C_k)\} = \frac{n(C_k)n(A_k)}{N} \quad (3.38)$$

This equation means that at time k , if $n(C_k)$ water heaters are selected to be turned off, then it is expected that $\frac{n(C_k)n(A_k)}{N}$ number of water heaters are on at this time.

Therefore, the expected value of the amount of reduction on the aggregated load is:

$$E\{P_k^u - P_k^c\} = \frac{n(C_k)}{N}n(A_k)P_0 = \frac{n(C_k)}{N}P_k^u \quad (3.39)$$

For example, let us assume that at a certain time, 20% of the water heaters are on (that means $n(A_k)/N = 0.2$); then, by selecting 50 random water heaters, it is expected that 20% of these random water heaters will be on, which is 10 water

heaters. So it is expected that by turning off these random water heaters, the load could be reduced by $10P_0$.

In practice, especially in this pilot project, the P_k^c is given to the controller as the set point. The controller should determine how to achieve this set point from the forecasted P_k^u . The number of houses that should be controlled can be calculated using:

$$n(C_k) = \frac{NE\{P_k^u - P_k^c\}}{P_k^u} \quad (3.40)$$

The payback effect of this control action on the next time interval expresses the effect of turning on the selected water heater in the next sample interval.

Let us assume that the controller turns the water heaters back on at time j . Therefore, the reduced energy at time k will be shifted to the time j . However, some of these water heaters were already consuming energy at time j ; therefore, they do not participate in the payback effect. Only those *controlled* water heaters that were *on* at time k and *off* at time j take part in the payback effect. These loads can be obtained by excluding the water heaters in set A_k, C_k, A_j from the water heaters in set A_k, C_k . So the payback can be formulated as:

$$I_j^k = \left(n(A_k, C_k) - n(A_k, C_k, A_j) \right) P_0 \quad (3.41)$$

The expected value for the impact of a control action on the next interval is

$$\begin{aligned} E\{I_j^k\} &= n(C_k) \frac{n(A_k) - n(A_k, A_j)}{N} P_0 \\ &= n(C_k) \frac{1 - n(A_k, A_j)/n(A_k)}{N} n(A_k) P_0 \\ &= n(C_k) \frac{1 - n(A_k, A_j)/n(A_k)}{N} P_k^u \end{aligned} \quad (3.42)$$

This equation can be rewritten as:

$$E\{I_j^k\} = \frac{n(C_k)}{N} \left(1 - \frac{n(A_k, A_j)}{n(A_k)}\right) P_k^u \quad (3.43)$$

where $n(C_k)$ is the number of controlled water heaters and P_k^u is the uncontrolled power consumption at time k . The final parameter, $n(A_k, A_j)/n(A_k)$, is the expected ratio of the number of water heaters that are on during two successive intervals. This value can be measured with the statistical analysis of the uncontrolled behaviour of the individual water heaters.

The impact of this control action at time k may still have some impact on the following intervals as well. The impact is due to those water heaters that were on during k and j and turned off during $j + 1$; therefore, the control during k may impact $j + 1$.

$$I_{j+1}^k = \left(n(A_k, A_j, C_k) - n(A_k, A_j, A_{j+1}, C_k)\right) P_0$$

Similar to the previous solution, the expected value for the impact will be:

$$E\{I_{j+1}^k\} = \frac{n(C_k)}{N} \left(\frac{n(A_k, A_j)}{n(A_k)} - \frac{n(A_k, A_j, A_{j+1})}{n(A_k)}\right) P_k^u \quad (3.44)$$

In general, the impact can be formulated as:

$$E\{I_{j+m}^k\} = \frac{n(C_k)}{N} \left(\frac{n(A_k, A_j, \dots, A_{j+m-1})}{n(A_k)} - \frac{n(A_k, A_j, \dots, A_{j+m})}{n(A_k)}\right) P_k^u \quad (3.45)$$

And the aggregated power can be adjusted with

$$P_{j+m}^c = P_{j+m}^u + E\{I_{j+m}^k\} \quad (3.46)$$

3.5 Experimental Results for Aggregated Forecast

The forecast method presented in Section 3.4 was implemented in a pilot project that contains approximately 100 electric water heaters. The power consumption of each load could be obtained through a smart meter in that house.

This section presents the performance evaluation of the forecasting methods along with the experimental results to validate the payback model.

3.5.1 Evaluation of the Load Forecast

The forecasting algorithm generates a 4-hours ahead load profile in 15 minute time intervals. The algorithm runs every 15 minutes to produce a new forecasted load profile. Thus, each forecasted load profile contains 16 samples.

Let $\hat{P}(k_0 + j|k_0)$ denote the forecasted power consumption at time $k_0 + j$ which is created at time k_0 , where $j \in [1..16]$. The relative error of the j^{th} forecast can be calculated at time $k_0 + j$ when the actual measurements can be obtained. Let $R_E(j|k_0)$ denote the relative error of the j^{th} sample, created at time k_0 .

$$R_E(j|k_0) = \frac{P(k_0 + j) - \hat{P}(k_0 + j|k_0)}{P(k_0 + j)} \quad (3.47)$$

The average relative error for each forecasting interval then can be calculated by averaging $R_E(j|k_0)$ for $\mathcal{L}$ various k_0 s.

$$\bar{R}_E(j) = \frac{\sum_{\text{all } K_0} R_E(j|k_0)}{\mathcal{L}} \quad (3.48)$$

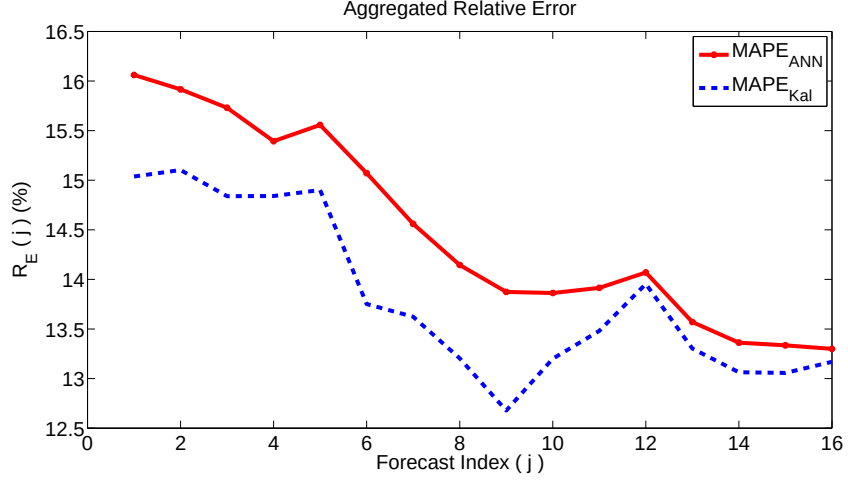


Figure 3.10: Comparison of the ANN forecast error and Filtered ANN for each sample in the forecast horizon

The ANN-based method combined with the Kalman filter was used to generate a 4-hours ahead aggregated load forecast over approximately 100 loads. These algorithms were running for several months and the relative error for each sample in the forecasting horizon was calculated and depicted in Figure 3.10.

Figure 3.10 shows that the Kalman filter can reduce the relative error by approximately 1% for the first 2 hours. Comparing Figure 3.10 with Figure 3.7 indicates that the performance of the aggregated load forecast is significantly better than the aggregation of individual load forecasts.

Figure 3.11 illustrates a sample snapshot of an aggregated forecast profile (the outputs of the ANN and the Kalman filter) at 7 : 00 am along with the measured aggregated load.

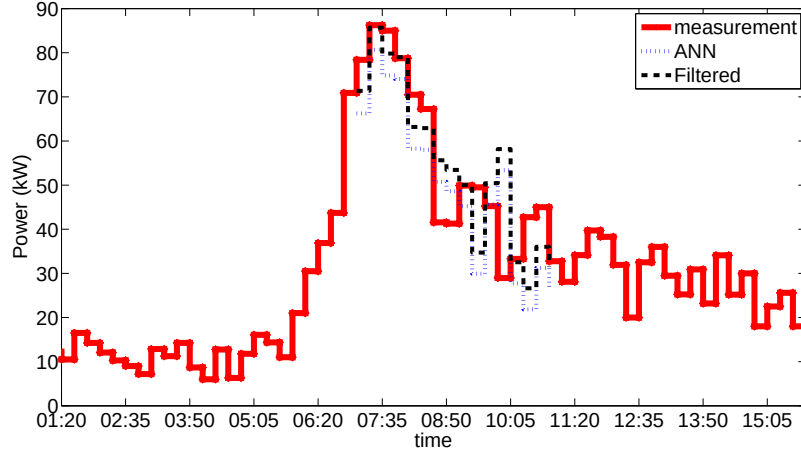


Figure 3.11: A snapshot of the one ANN forecast, Filtered ANN forecast and their comparison with the actual measurements

3.5.2 Evaluate the Payback Effect

To build the proposed model of the payback, the first step is to calculate the expected ratio of the water heaters that are consuming energy over consecutive intervals. These values have been calculated for different times of the day. Table 3.4 lists some average values for these parameters for the interval of 5 : 00 am to 9 : 00 am. The full table is listed in Appendix B.2.

Each row of this table shows the ratios of the number of water heaters that remain on during the consecutive intervals to the number of water heaters that are on during the first interval. For example, the first row where $k = [5 : 00 - 5 : 15]$. This row indicates that on average, 39% of the water heaters that are on during $[5 : 00 - 5 : 15]$ remain on for the next interval ($[5 : 15 - 5 : 30]$), and 16% of them still remain on during $[5 : 30 - 5 : 45]$ and so on.

Table 3.4: A sample of the expected ratio of the water heaters that are on during consecutive intervals

k	$\frac{n(A_k, A_{k+1})}{n(A_k)}$	$\frac{n(A_k, \dots, A_{k+2})}{n(A_k)}$	$\frac{n(A_k, \dots, A_{k+3})}{n(A_k)}$
05:00-05:15	0.39	0.16	0.14
05:15-05:30	0.55	0.21	0.18
05:30-05:45	0.53	0.31	0.25
05:45-06:00	0.60	0.39	0.33
06:00-06:15	0.68	0.48	0.38
06:15-06:30	0.71	0.47	0.35
06:30-06:45	0.71	0.48	0.37
06:45-07:00	0.69	0.47	0.38
07:00-07:15	0.71	0.52	0.44
07:15-07:30	0.72	0.53	0.42
07:30-07:45	0.71	0.50	0.34
07:45-08:00	0.70	0.44	0.31
08:00-08:15	0.68	0.41	0.29
08:15-08:30	0.63	0.39	0.29
08:30-08:45	0.63	0.40	0.30
08:45-09:00	0.64	0.42	0.31

Note that these values have been calculated for the water heaters in this pilot project while no control actions were applied to them. The complete table contains 8 ratio columns (which show the ratio for 2 hours after each interval) and 96 rows for each 15-minute interval of the day.

To validate the proposed payback model, various experiments have been conducted on the pilot project with $N = 100$ participating water heaters (with nominal power $P_0 = 3kw$). One of the experiments is depicted in Figure 3.12.

In this experiment, a command is executed to keep the controllable load at $30kw$ during $7 : 00 - 7 : 30$. The most recent forecast, as illustrated in Figure 3.12, indicates that the power level will be around $81kw$, which means:

$$n(A_k) = P_k^u / P_0 = 27$$

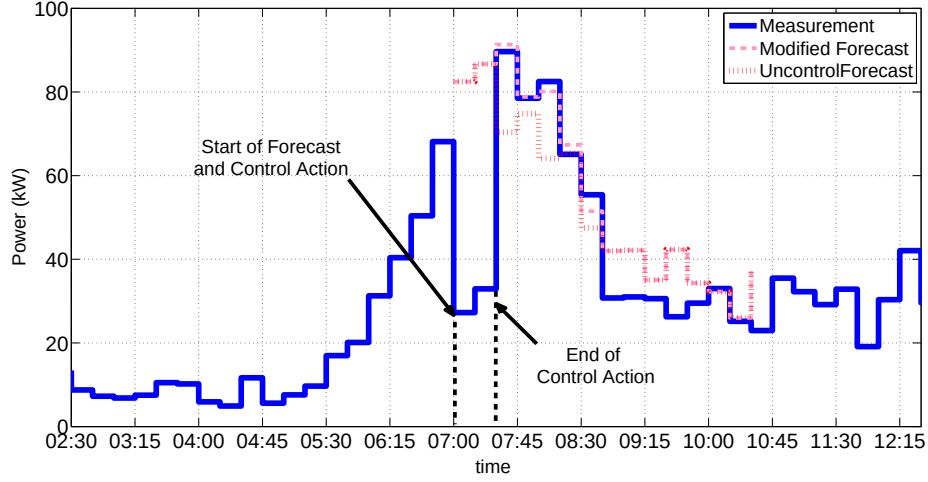


Figure 3.12: Impact of a control action on the aggregated load

In order to keep the load at 30, the controller should reduce 50kW of the load. The number of random water heaters that should be turned off can be obtained using (3.40):

$$n(C_k) = \frac{50 \times 100}{27 \times 3} \simeq 62 \quad (3.49)$$

So the controller randomly sends a turn-off command to 62 water heaters, and as predicted, the total power level drops to around 30kW. Because of the forecast error and the statistical nature of the selected house, the power level cannot be exactly 30kW and some deviations from the set-point can be observed.

The selected water heaters will be kept off for 2 sample times. Therefore, the impact of this control action must be observed after $k + 2$. Table 3.4 suggests that 52% of the water heaters that are on during $[7 : 00 - 7 : 15]$ will remain on 2 samples later. Consequently, the payback model of (3.45) predicts that the impact will be

$$E\{I_{k+2}^k\} = \frac{62}{100}(1 - 0.52)81 = 24.1kW$$

The uncontrolled forecasted load for this interval, as illustrated in Figure 3.12, is $70kw$; thus, the power level after this control action is predicted to be:

$$P_{k+2}^c = 70 + 24.1 = 94.1kw$$

As can be seen in Figure 3.12, the actual measured power after the control action is $90kw$, which is very close to the value that the model predicted. As before, the error could be due to the forecast error and the nature of the statistical model.

The impact of this control action can also be estimated on the next interval using (3.44). Using the data in Table 3.4 suggests that the impact on the second interval will be:

$$I_{k+3}^k = \frac{62}{100}(0.52 - 0.44)81 = 4kw$$

Since the uncontrolled load forecast for $[7 : 45 - 8 : 00]$ is around $75kw$, the control action will increase this level by $4kw$:

$$P_{k+2}^c = 75 + 4 = 79kw$$

which agrees with the measured value.

4 Load Control Strategies

It is assumed that the power consumptions of individual controllable loads are available, and each load is equipped with an electronic switch in series with its internal thermostat system which allows centralized control for connecting/disconnecting power to the device. With these assumptions, it can be concluded that TCA devices such as DEWHs may not be controllable at all instants in time:

1. when the internal thermostat turns the DEWH off due to high water temperature, the controller cannot turn it back on.
2. when the water temperature falls below a certain threshold, customer comfort will be compromised and it becomes necessary for the DEWH to remain *on* to heat up the water.

These constraints reduce the load following capacity of a controller. In order to produce feasible set points for load following, an estimation of the system capacity is required. The control strategy defines how the load should be scheduled to both follow a given set point, and maintain customer comfort. The definitions for the customer comfort and the strategy for scheduling the loads may change the limits of the achievable power level. So, different control strategies may have different capacities to follow a desired load.

Section 4.1 presents the terminology and definitions of various concepts used by the controller. A basic load following strategy along with a model of its capacity is presented in Section 4.2, followed by the experimental results in Section 4.3. Another

control strategy is presented in Section 4.4 which tries to make a balance in its reserve capacity. The experimental results are presented in Section 4.5.

4.1 Definitions

Figure 4.1 depicts a general overview of the direct load control system. In a load following program, the power consumptions of the individual loads are manipulated by the controller in a way that the *aggregated load* follows a *desired aggregated load (DAL)* profile, given by the system operator. In order to generate a feasible DAL, a forecast of power consumption along with the *reserve capacities* are required, as illustrated in this figure.

Various terms, such as aggregated load, which are used frequently in this chapter are required to be defined carefully. This section presents the definitions of these terms and how they are formulated.

4.1.1 Terminology

Aggregated Load: Let us assume that there are N end-use devices (DEWHs) in the system, and the power consumptions of the i^{th} device at time k is denoted by p_k^i . The aggregated load, denoted by $P_k = \sum_{i=1}^N p_k^i$, is the sum of the measured power of each individual load at time k . The forecasted aggregated load, denoted by $\hat{P}_{k+j|k}$ is the forecasted aggregated power consumptions of the loads that is generated at time k for a time horizon after k , where $j \geq 1$. Similarly, the forecasted aggregated power for time k that is generated one sample earlier is denoted by $\hat{P}_{k|k-1}$. For simplicity, this will be written as $\hat{P}_k$.

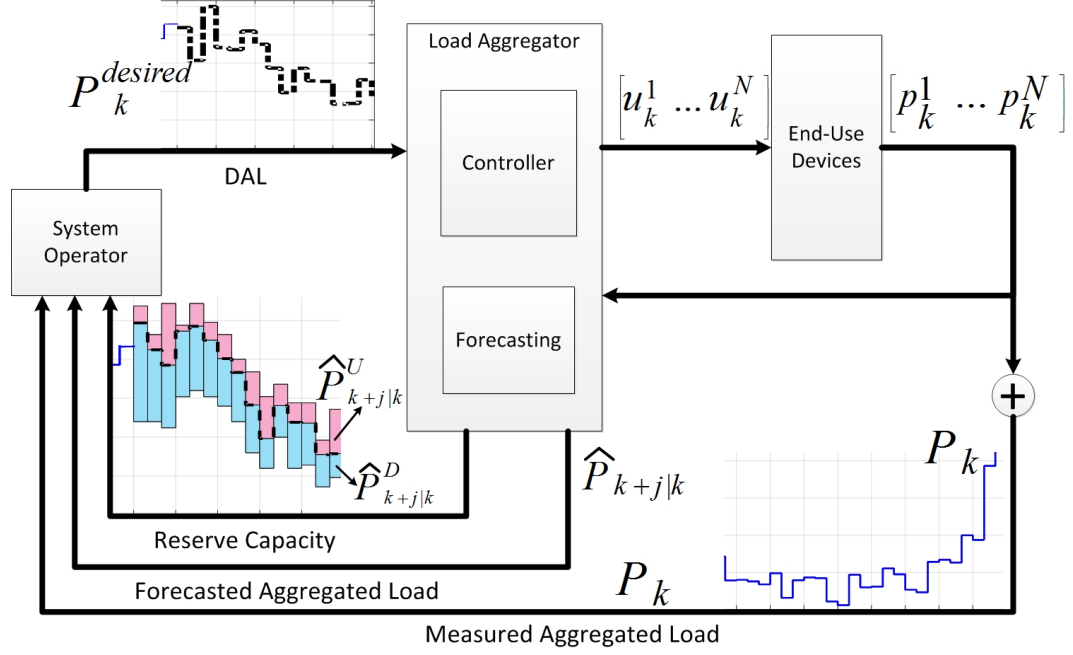


Figure 4.1: The overall structure of a load following control system.

Desired Aggregated Load (DAL): is a load profile of the aggregated power consumption that is generated by the system operator for the controller to follow in a future time horizon.

Set Point: The value of the DAL at each point in time is called a set point, denoted by $P_k^{desired}$. It is the desired aggregated power to be achieved by controlling the power consumptions of individual loads at the given time k in the future.

Controlled/Uncontrolled Load Profile: If the controller performs any action on the load and changes its normal power consumptions, then the power consumption load profile will be referred to as a *Controlled Load Profile*. In the absence of centralized control, the power consumption load profile is called an *Uncontrolled Load Profile*.

Managed/Unmanaged Load Profile: In a special case when the set point is equal to the forecasted load ($P_k^{desired} = \hat{P}_k$), then the resulting load profile will be referred to as an *Unmanaged Load Profile*, regardless of whether the controller is

sending a control signal or not. For all other choices of $P_k^{desired}$, the resulting load profile will be referred to as a *Managed Load Profile*

Customer Comfort may be measured by the average water temperature of the controlled DEWH. To ensure customer comfort, the controller should prevent the average water temperature from dropping below a certain threshold value. There are circumstances when customer comfort is compromised without any control action, for example, when a large amount of water is used in a short time, the temperature may drop quickly. Under these circumstances, control will not alleviate this effect.

Ramp-Up Reserve Set (U_k): This is a set of DEWHs whose internal thermostats are active at the given time k , but the controller has disabled power to the heater. The DEWH will consume power at the instant the controller enables power to the DEWH.

Ramp-Down Reserve Set (D_k): This set contains the DEWHs that are consuming power at the given time k , and may be turned off by the controller without compromising the customers.

Ramp-Up Reserve Capacity (P_k^U): This is the power that can be added to the unmanaged load profile by turning on the controllable loads. A forecast of the ramp-up reserve capacity for the j^{th} sample ahead at time k is denoted by $\hat{P}_{k+j|k}^U$, where $j \geq 1$. For simplicity, the forecasted value of the ramp-up reserve capacity at time k will be denoted by $\hat{P}_k^U$.

Ramp-Down Reserve Capacity (P_k^D): This is the power that can be reduced from the unmanaged load profile by turning off the controllable loads. A forecast of the ramp-down reserve capacity for the j^{th} sample ahead at time k is denoted by $\hat{P}_{k+j|k}^D$, where $j \geq 1$. For simplicity, the forecasted value of the ramp-down reserve capacity at time k will be denoted by $\hat{P}_k^D$.

4.1.2 Formulation

4.1.2.1 Reserve Sets

Reserve capacity is the amount of power that can be added to or deducted from an unmanaged load under centralized control while maintaining customer comfort. Let us assume that the measurements of the power consumption of each end-use device are available. Therefore, the controller can cluster the loads into various groups that can be used as reserve sets.

Let N denote the number of controllable loads (in this project, electric water heaters) in the system. If s_k^i denotes the on(= 1)/off(= 0) internal state of the i^{th} controllable load at time k , and T_k^i denotes the water temperature in the i^{th} tank at time k , then the forecasted value for each of these parameters will be denoted by $\hat{s}_k^i$ and $\hat{T}_k^i$ respectively. The forecasted ramp-down reserve set at time k can be formulated as:

$$\hat{D}_k = \{i \in 1..N | u_k^i = 1, \hat{s}_k^i = 1, \hat{T}_k^i > T_s\} \quad (4.1)$$

where T_s is a threshold temperature for customer comfort.

However, the temperature data may not be available for use by the controller to ensure customer comfort. Another method to guarantee customers comfort is to impose a constraint on the control action such that the load is not interrupted more than l samples within a window of L samples. The ramp-down reserve set with this constraint can be formulated as:

$$\hat{D}_k = \{i \in 1..N | u_k^i = 1, \hat{s}_k^i = 1, \sum_{m=1}^L (1 - u_{k-m}^i) < l\} \quad (4.2)$$

where u_{k-m}^i denotes the control signal for the i^{th} load at time $k-m$. In this equation, $u = 0$ indicates a turn-off command.

The values for l and L can be chosen by a simulator to ensure that the controller will not compromise the customer comfort, as explained in Appendix C.1. The values of $l = 4$ and $L = 8$ were selected and used in two pilot projects with approximately 300 DEWHs in total where no complaints were received regarding customer comfort.

For the ramp-up reserve set, a similar approach can be used. Ramp-up reserve set contains the controllable loads that are turned off by the controller, and if enabled, they can start consuming power. This means that a load is in the ramp-up reserve set at time k , if it is turned off by the controller at time $k-1$, and its internal thermostatic state is on and ready to consume power. This set can be formulated as:

$$\hat{U}_k = \{i \in 1..N | \hat{s}_k^i = 1, u_k^i = 0, \sum_{m=1}^L (1 - u_{k-m}^i) < l\} \quad (4.3)$$

4.1.2.2 Reserve Capacity

Let P_0^i denote the nominal power rating of the i^{th} controllable load. Then the forecasted ramp-down capacity ($\hat{P}_k^D$) at time k can be obtained from:

$$\hat{P}_k^D = \sum_{i \in \hat{D}_k} P_0^i \quad (4.4)$$

where $\hat{D}_k$ denotes the forecasted ramp-down reserve set at time k .

Similarly, the forecasted ramp-up reserve capacity ($\hat{P}_k^U$) at time k can be obtained from:

$$\hat{P}_k^U = \sum_{i \in \hat{U}_k} P_0^i \quad (4.5)$$

where $\hat{U}_k$ denotes the forecasted ramp-up reserve set at time k .

Assuming that all controllable loads have identical nominal power rating (P_0), then (4.4) and (4.5) can be rewritten as:

$$\hat{P}_k^D = P_0 n(\hat{D}_k) \quad (4.6)$$

$$\hat{P}_k^U = P_0 n(\hat{U}_k) \quad (4.7)$$

where $n(..)$ denotes the number of members in that set.

The ramp-up or down reserve capacity can be calculated at each instant of time using (4.6) and (4.7). The average reserve capacity can be calculated using the following equations:

$$\bar{P}_k^D = P_0 E\{n(D_k)\} \quad (4.8)$$

$$\bar{P}_k^U = P_0 E\{n(U_k)\} \quad (4.9)$$

where $\bar{x}$ denotes the average value of the parameter x and $E\{x\}$ is the expected value of the parameter x .

4.2 Basic Load Following Strategy

4.2.1 Formulation

The main objective of the basic load following controller is to minimize the difference between the forecasted load and the desired load.

The desired load at time k , denoted by $P_k^{desired}$, may be provided by the system operator. Let $\hat{s}_k^i$ denote the forecasted on/off status of the i^{th} load at time k . Thus the aggregated forecasted power will be:

$$\hat{P}_k = \sum_{i=1}^N u_k^i P_0 \hat{s}_k^i \quad (4.10)$$

where u_k^i is the control signal for the i^{th} load at time k .

The controller generates the control signals to minimize the objective function in (4.11) as a function of $\mathbf{u}_k$ which is a vector of control signals for each individual load ($\mathbf{u}_k = [u_k^1, \dots, u_k^N]$):

$$J_k(\mathbf{u}_k) = \left(P_k^{desired} - \sum_{i=1}^N u_k^i P_0 \hat{s}_k^i \right)^2 \quad \text{s.t. (4.2) and (4.3)} \quad (4.11)$$

To decrease the forecasted load in order to match the desired set point, the controller sends ‘turn off’ commands ($u_k^i = 0$) to those loads in the D_k set. In order to increase the forecasted load, the ‘turn on’ commands ($u_k^i = 1$) will be sent to the loads in the U_k set.

4.2.2 Unmanaged Reserve Capacity

Reserve capacity not only depends on the time, but also depends on the previous commands to a specific load due to customer comfort in (4.2).

To determine the average reserve capacity of the controller, let us assume that the desired load is equal to the forecasted load (unmanaged load). In this case, the controller does not need to change the power consumption of any load; so, $u^i(k) = 1$ for all loads at all times. Since no load was turned off at any time, no load will be placed in U_k at any time, which means that the ramp-up capacity will be equal to zero.

$$n(U_k) = 0 \Rightarrow P_k^u = 0 \quad \text{for all } k \quad (4.12)$$

For the ramp-down capacity, according to (4.8), the average reserve capacity is proportional to the average number of loads in the ramp-down reserve set ($n(D_k)$). The expected value of $n(D_k)$ can be obtained by adding the probability of each load may be in the ramp-down reserve set:

$$E\{n(D_k)\} = \sum_{i=1}^N Prob\{i \in D_k\} \quad (4.13)$$

where $Prob\{i \in D_k\}$ is the probability that the i^{th} load is in the ramp-down reserve set at given time k . Based on the definition of the ramp-down reserve set in (4.2), This probability can be formulated as:

$$Prob\{i \in D_k\} = Prob\{s_k^i = 1, \sum_{m=1}^L (1 - u_{k-m}^i) < l\} \quad (4.14)$$

Since $u_k = 1$ during the unmanaged time interval, $\sum_{m=1}^L (1 - u_{k-m}^i) = 0$, thus

$$Prob\{i \in D_k\} = Prob\{s_k^i = 1\} \quad (4.15)$$

Substituting (4.15) in (4.13) and assuming that all loads have similar statistical behaviour, the expected number of loads in the ramp-down reserve set will be:

$$E\{n(D_k)\} = N Prob\{s_k = 1\} \quad (4.16)$$

Therefore, at a given time k , the average value for the ramp-down reserve capacity will be

$$\bar{P}_{theory}^D(k) = NP_0 Prob\{s_k = 1\} \quad (4.17)$$

where $Prob\{s_k = 1\}$ is the probability that a load is consuming power at time k . This probability can be computed by dividing the number of water heaters that are on at a time instant k over the total number of water heaters.

Figure 4.2 depicts the average probability that a load would be on at each time interval in a day. This figure was obtained through the direct measurements of approximately 100 DEWHs over 2 months.

For example, at 7 : 00 am, the probability that a water heater is on, is about 25%, thus using (4.17), it is expected that the ramp-down reserve capacity for a group of 100 DEWHs with $3kW$ nominal power rating would be $75kW$.

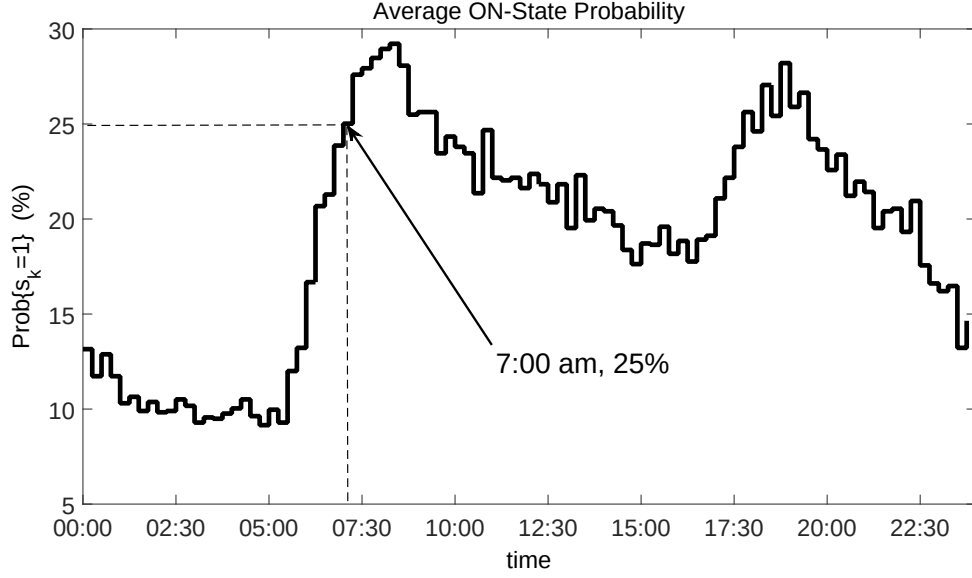


Figure 4.2: The average probability that a load would be *on* at different times of a day

4.3 Experimental Results

The proposed controller was deployed on a pilot project containing approximately 100 loads and executed for several months. The reserve capacity of this controller and its performance were analyzed.

4.3.1 Average Unmanaged Reserve Capacity

In order to measure the average unmanaged reserve capacity of this control strategy, the controller was executed with the set points equal to the forecasted load, in order to preserve the capacity. The reserve capacity for each time was forecasted by forecasting the state of each load ($\hat{P}_k^D$). Then, by measuring the actual state of the loads and forming the reserve sets using (4.2) and (4.3), the actual reserve capacity at that time can be measured (P_k^D).

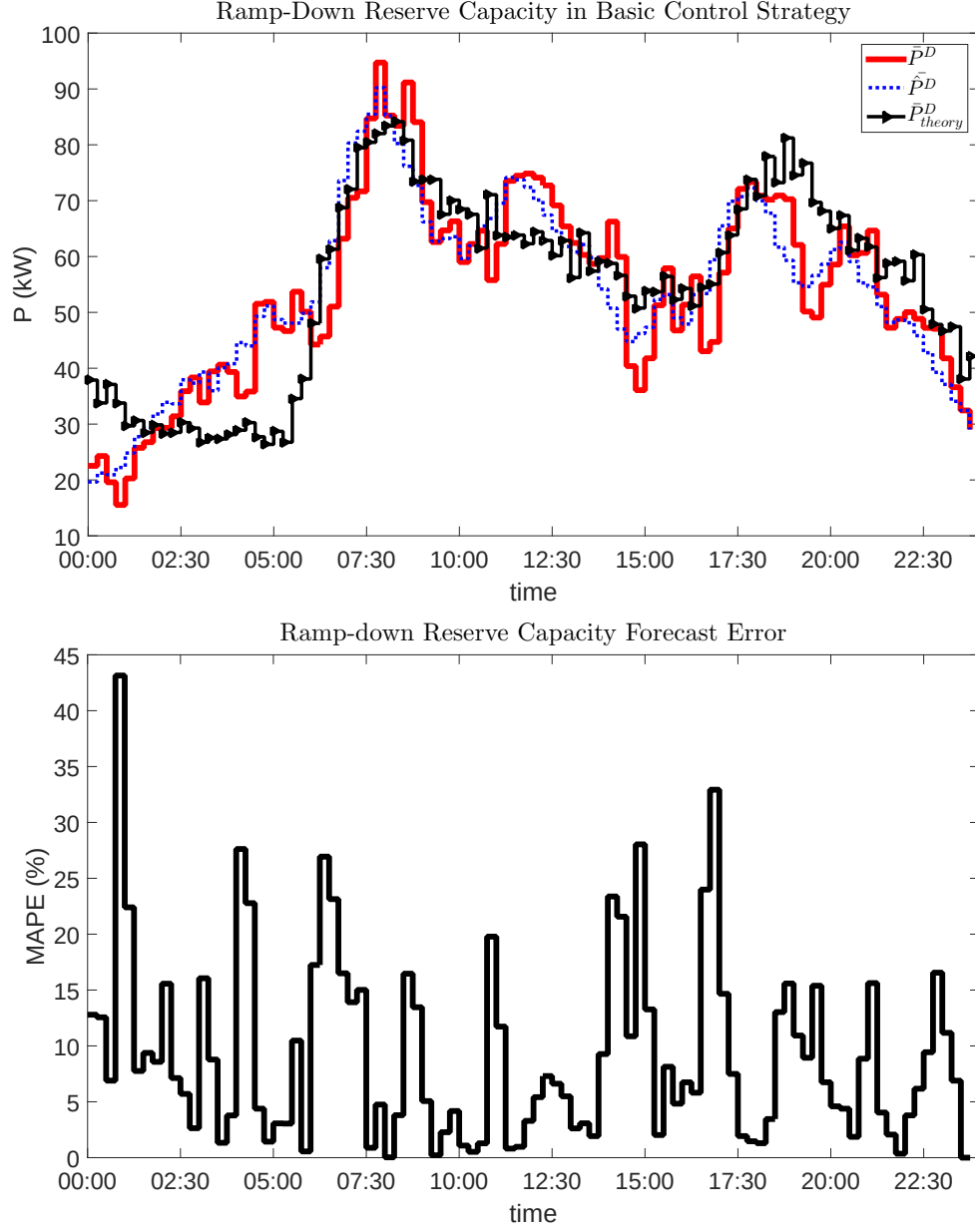


Figure 4.3: The average measured ($\bar{P}^D$), forecasted ($\hat{\bar{P}}^D$) and modeled ($\bar{P}_{theory}^D$) ramp-down reserve capacity (top), and the mean absolute percentage of forecast error (bottom)

Moreover, (4.17) provides a theoretical estimation on the average ramp-down reserve capacity, $\bar{P}_{theory}^D(k)$. The actual average ramp-down reserve capacity, $\bar{P}_k^D$, is obtained by calculating the mean value of the reserve capacity at a specific time of day over the three previous months. The same method was applied to calculate

the average forecasted reserve capacity, $\bar{P}_k^D$. These average values along with the theoretical value for the ramp-down capacity are depicted in Figure 4.3. The average ramp-down capacity is obtained by averaging the ramp-down reserve capacity over a month.

As can be seen in this figure, the reserve capacity changes with the time. In the morning, when the probability of the water heater being energized is higher, the reserve capacity is also higher.

The mean absolute percentage error (MAPE) used in forecasting the reserve capacity, which is depicted in Figure 4.3 (bottom), is calculated by:

$$MAPE_k = 100 \times \frac{|\hat{P}_k^D - P_k^D|}{\hat{P}_k^D} \quad (4.18)$$

From the definition of the relative error in (4.18), when the reserve capacity is small, a small error in forecast may result in large relative error. For example at 1 : 00am, when the measured reserve capacity is 21kW while the forecasted reserve is 15kW, the 6kW error is due to the error in forecasting the status of 2 water heaters out of 96, but the relative error will be approximately 40%. However, at 7 : 00am when the measured capacity is 72kW and the forecasted capacity is 81kW (forecast error in the state of 3 water heaters out of 96), the relative error is approximately 11%.

It should be noted that the forecasted reserve capacity is obtained by aggregating the forecasted on/off status of the individual loads. As described in Chapter 3, it is expected to observe a 20% aggregated error if there is a 5% error in individual forecasts, which agrees with the results depicted in this figure.

4.3.2 Control System Performance

The difference between the measured aggregated load and the desired aggregated load (DAL) shows how much the controller is successful in following the desired load profile. The basic load following control strategy was deployed on a pilot containing approximately 100 DEWHs. A system operator produced the set points using the forecasted reserve capacity, load profile, and wind generation capacity from various sources.

Figure 4.4 depicts an example of the load following program over a day, along with the absolute relative error in following the desired load. This figure shows the forecasted load ($\hat{P}_k$) along with the set points ($P_k^{desired}$) and the measured power consumptions (P_k) of the DEWHs. As can be seen in this figure, the set points are given in a way to shave the forecasted peak in the morning and evening. The controller tries to follow this target profile by controlling the individual loads.

The absolute relative error in following the desired load is calculated by:

$$e_R = 100 \times \frac{|P_k^{desired} - P_k|}{P_{desired}} \quad (4.19)$$

This controller was operating over several months and the load following absolute error ($e_k = |P_k^{desired} - P_k|$) and the absolute relative error (4.19) were calculated for each set point. The average absolute relative error of this controller over a month at different times of a day is depicted in Figure 4.5. On average, the absolute error of this control strategy in following the DAL is $4.6kW$ and the average relative error is approximately 8%.

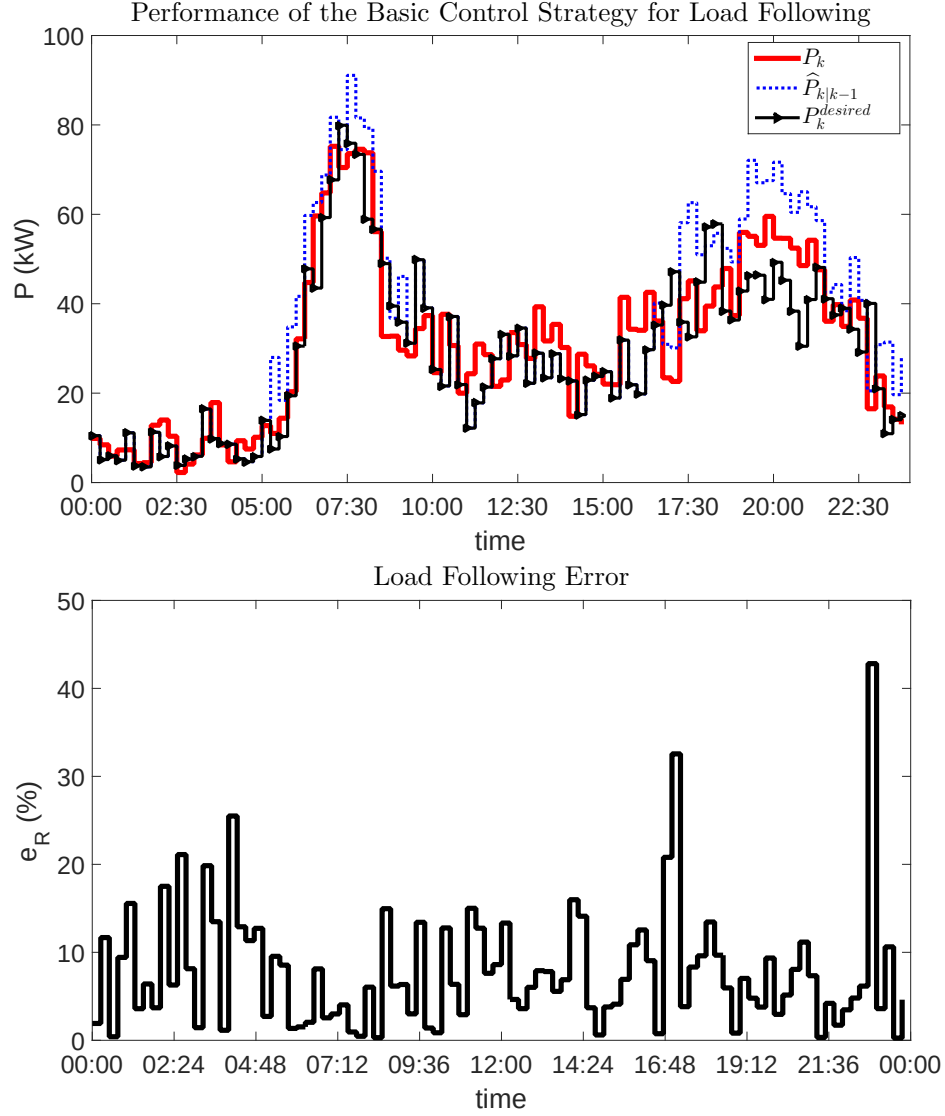


Figure 4.4: The operation of the controller with the basic strategy (top) and its absolute percentage error (bottom)

4.4 Balanced-Reserve Control Strategy

In this dissertation, it is assumed that if a DEWH is turned off by its internal thermostat, it cannot be turned back on unless its thermostat turns it on due to low water temperature, as described in Chapter 2. When the DEWH is thermostatically turned on, it may, or may not be eligible to be used for control purposes due to customer comfort. According to (4.2), if the DEWH is not interrupted more than

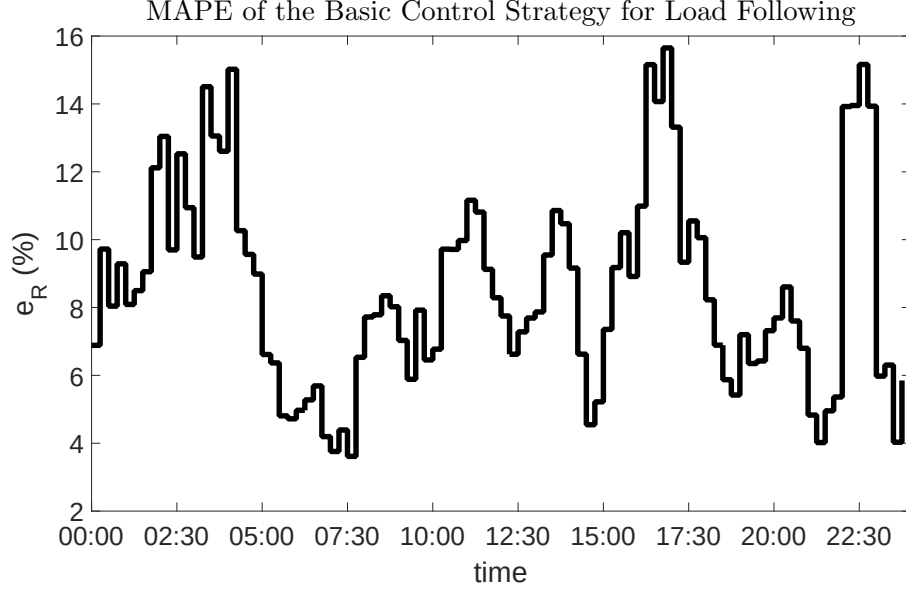


Figure 4.5: The mean absolute percentage error of the basic load following strategy

l samples in the previous L sample window, and it is thermostatically on, then it will be placed in the ramp-down reserve set. Then, the controller may select it to be interrupted, and move it to the ramp-up set by turning it off. Figure 4.6 depicts the transitions between possible states that a water heater could take. The various states and conditions can be summarized as:

- s_k : denotes the on/off status of the internal thermostat of the DEWH.
- R_k : denotes a condition for customer comfort, as implied in (4.2). In this dissertation, the condition for a load to be used in ramp-down reserve set is that it is not interrupted more than l samples during past L samples:

$$R_k = \begin{cases} 1 & \text{if } \sum_{m=1}^L (1 - u_{k-m}) < l \\ 0 & \text{otherwise} \end{cases} \quad (4.20)$$

- *Off*: The DEWH is turned off by its internal thermostat (i.e. $s_k = 0$).

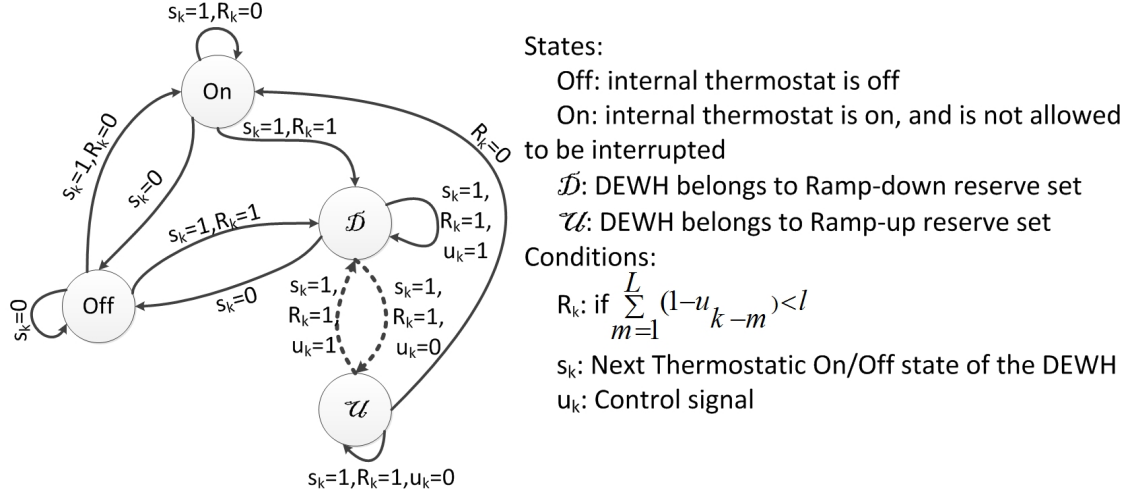


Figure 4.6: The state transition of an individual DEWH

- *On*: The DEWH is on (i.e. $s_k = 1$), and the controller is not allowed to turn it off due to customer comfort (it means that the load was interrupted at least l times over past L samples, i.e. $R_k = 0$).
- $\mathcal{D}$: The DEWH is on, and it can be interrupted without compromising customer comfort. This load belongs to the ramp-down reserve set, D_k .
- $\mathcal{U}$: The internal thermostat is on, however, the external controller has interrupted the load (i.e. $u_k = 0$). This load belongs to the ramp-up reserve set, U_k .

As can be seen in this figure, the ramp-up reserve set can only be formed when a DEWH is turned off by the controller. Thus, the controller can manipulate the reserve sets by turning on and off the loads in the ramp-down reserve set, as depicted in Figure 4.6.

The balanced-reserve control strategy aims to create a balance in the number of loads that belong to U_k and D_k by intentionally causing the load to transition from ramp-down reserve set to ramp-up reserve set.

4.4.1 Formulation

At each time instant, k , the ramp-down reserve set, D_k , is a set of the loads that can be used for reducing the aggregated power consumption, and the ramp-up reserve set, U_k , is a set of the loads that can be used to increase the aggregated power consumption.

The controller can adjust the number of loads that belong to U_k by intentionally turning off a DEWH that initially belongs to the set D_k . However, this operation may reduce the capability of a load to be used in D_k in future due to the customer comfort constraint imposed in (4.2). Therefore, an additional constraint is considered that only allows the controller to transition loads from D_k to U_k if the controller has not turned it off in the previous sample, i.e. $u_{k-1} = 1$. Let A_k denote the set of the loads that are eligible and available for this operation:

$$\hat{A}_k = \{i \in \hat{D}_k | u_{k-1} = 1\} \quad (4.21)$$

A new water heater state, $\mathcal{A}$, as shown in Figure 4.7, identifies when a load satisfies the condition in (4.21). At any time instant, k , some water heaters are ‘off’ by their internal thermostat controls, and some water heaters are ‘on’. Within the ‘on’ set, only those water heaters that satisfy (4.2), i.e. $(R_k = 1)$, can be used as the ramp-down reserve set, D_k . Again, among these water heaters, only those that satisfy (4.21), and therefore belong to the set A_k , can be turned off to transition to the ramp-up reserve set U_k . The process of selecting the loads that belong to $\hat{A}_k$ and intentionally transitioning loads to the set U_k is formulated by the control strategy.

To follow the desired aggregated load and balance the number of loads belonging to the reserve sets, the controller minimizes two cost functions J_k^1 and J_k^2 , as a

4.4.2 Unmanaged Reserve Capacity

The reserve capacity depends on the control strategy and the given set points to follow. In the unmanaged load control scenario, the given set point is equal to the forecasted load, and therefore does not effect the reserve capacity. So, the unmanaged reserve capacity reflects the maximum reserve capacity available for various control strategies.

At each time instant k , the controller intentionally changes the state of some water heaters from $\mathcal{A}$ to $\mathcal{U}$ in order to create a ramp-up reserve capacity during the interval $[k, k + 1]$; thus, the reserve sets will be different at a time instant k and slightly before k , denoted by k^- , changing from an *uncontrolled* reserve set to a *controlled* reserve set.

Let us define $D_{k^-}^u$, where the superscript u specifies the *uncontrolled* ramp-down reserve set before the sampling time k . This set contains the loads that can be turned off without disturbing the customers. However, due to (4.21), the controller can only turn off the loads that are not turned off in the previous sampling interval. This means that only a subset of the $D_{k^-}^u$, denoted by A_{k^-} , is available to be used for this control strategy.

To estimate the reserve capacity, let us assume that the control actions start at k_0 . This means that the controller has not issued any turn-off commands earlier. Therefore,

$$A_{k_0^-} = D_{k_0^-}^u \quad (4.23)$$

To minimize the $J_{k_0}^2$, the controller sends turn-off commands to one half of the loads belonging to the uncontrolled ramp-down reserve set ($D_{k_0^-}^u$), which result in a change in the ramp-down and ramp-up reserve sets. These new sets are denoted by $D_{k_0}^c$ and $U_{k_0}^c$ respectively where the superscript c denotes the *controlled* sets.

$$n(D_{k_0}^c) = n(U_{k_0}^c) = \frac{1}{2}n(A_{k_0}^-) = \frac{1}{2}n(D_{k_0}^u) \quad (4.24)$$

Let k_1 denote the next sampling time. Thus, the ramp-down reserve set will be $D_{k_1}^u \cup U_{k_0}^c$. The available load for the control will be:

$$A_{k_1}^- = D_{k_1}^u - (D_{k_1}^u, U_{k_0}^c) \quad (4.25)$$

where the minus sign denotes an exclusive operation on the sets.

Now by turning off one half of the available loads at k_1 , the ramp-up reserve set will contain $n(U_{k_1}^c)$ DEWHs, and the other half of the eligible loads plus the controlled load from previous interval will form the controlled ramp-down reserve set $n(U_{k_1}^c)$.

$$\begin{cases} n(U_{k_1}^c) = \frac{1}{2}n(A_{k_1}^-) = \frac{1}{2}n(D_{k_1}^u) - \frac{1}{2}n(D_{k_1}^u, U_{k_0}^c) \\ n(D_{k_1}^c) = \frac{1}{2}n(A_{k_1}^-) + n(U_{k_0}^c) \end{cases} \quad (4.26)$$

According to (4.24), since the controlled ramp-up reserve set contains half of the loads in the uncontrolled ramp-down reserve set, therefore

$$n(U_{k_0}^c) = \frac{1}{2}n(D_{k_0}^u) \quad (4.27)$$

$$\Rightarrow E\{n(D_{k_1}^u, U_{k_0}^c)\} = \frac{1}{2}E\{n(D_{k_1}^u, D_{k_0}^u)\} \quad (4.28)$$

$$\Rightarrow E\{n(U_{k_1}^c)\} = \frac{1}{2}E\{n(D_{k_1}^u)\} - \frac{1}{4}E\{n(D_{k_1}^u, D_{k_0}^u)\} \quad (4.29)$$

The controlled ramp-down reserve set would be the other half of the eligible load in addition to the controlled ramp-up set from the previous interval:

$$n(D_{k_1}^c) = n(U_{k_0}^c) + \frac{1}{2}n(A_{k_1}^-) \quad (4.30)$$

In general, it can be concluded that

$$\begin{cases} E\{n(U_k^c)\} = \frac{1}{2}E\{n(D_{k-}^u)\} - \frac{1}{4}E\{n(D_{k-}^u, D_{k-1-}^u)\} \\ E\{n(D_k^c)\} = E\{n(U_k^c)\} + E\{n(U_{k-1}^c)\} \end{cases} \quad (4.31)$$

According to (4.9) and (4.8), the average ramp-up and ramp-down reserve capacity is:

$$\begin{cases} \bar{P}_k^U = P_0 E\{n(U_k^c)\} \\ \bar{P}_k^D = E\{n(D_k^c)\} \end{cases} \quad (4.32)$$

Substituting (4.16) in (4.31) and (4.32), the average controlled reserve capacities can be obtained from (4.33).

$$\begin{cases} \bar{P}_{theory}^U(k) = \frac{NP_0}{2} \left(Prob\{s_k = 1\} - \frac{1}{2} Prob\{s_k = 1, s_{k-1} = 1\} \right) \\ \bar{P}_{theory}^D(k) = \bar{P}_{theory}^U(k-1) + \bar{P}_{theory}^U(k) \end{cases} \quad (4.33)$$

where at the initial time k_0 :

$$\begin{cases} \bar{P}_{theory}^U(k_0) = \frac{NP_0}{2} Prob\{s_{k_0} = 1\} \\ \bar{P}_{theory}^D(k_0) = \frac{NP_0}{2} Prob\{s_{k_0} = 1\} \end{cases} \quad (4.34)$$

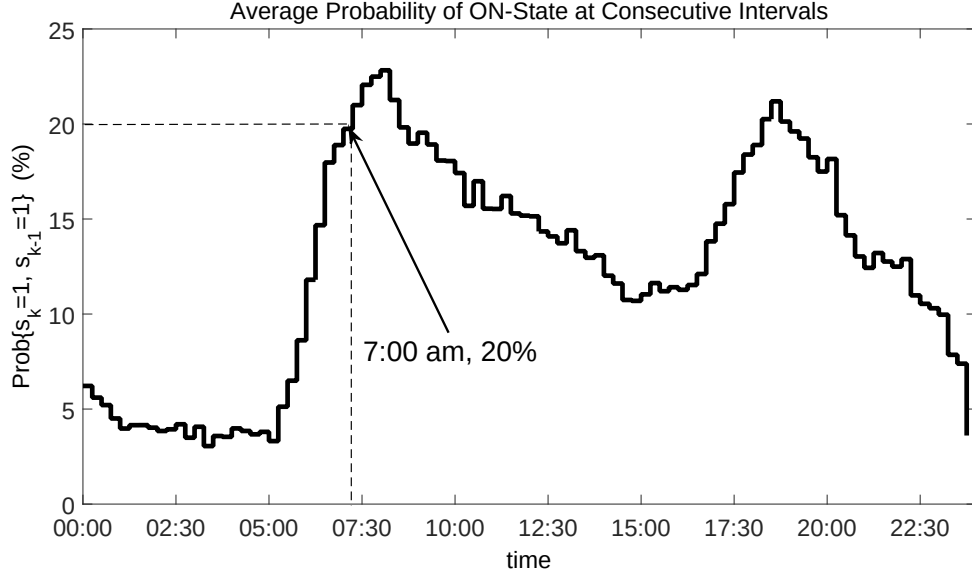


Figure 4.8: The average probability that a load consumes power over two consecutive time intervals.

It should be noted that (4.33) and (4.34) are based on uncontrolled load state probabilities. It means that these probabilities can be calculated with direct measurements.

The probability that a load is on at time instant k , $Prob\{s_k = 1\}$, can be computed by dividing the number of water heaters that are on at that time over the total number of water heaters, as described for (4.17) and depicted in Figure 4.2.

The probability that a load is on at two consecutive time instants $k - 1$ and k , $Prob\{s_k = 1, s_{k-1} = 1\}$, also can be computed similarly by dividing the number of water heaters that are on over that two consecutive samples over the total number of water heaters, as depicted in Figure 4.8.

As an example, at 7 : 00am the probability that a water heater is on, is about 25%, as depicted in Figure 4.2. And at the same time, $Prob\{s_k = 1, s_{k-1} = 1\} = 20\%$, as depicted in Figure 4.8. Thus, at this time, it is expected that the ramp-up and

down reserve capacities for a group of 100 DEWHs with a $3 - kW$ nominal power rating would be $25kW$ and $50kW$ respectively.

4.5 Experimental Results

The proposed controller was deployed on a pilot project with approximately 100 DEWHs. In this pilot, the power consumption of each individual load could be obtained through a smart meter and the load could be controlled through a load control module, installed at each house.

4.5.1 Average Unmanaged Reserve Capacity

The controller receives the forecasted load for the next control interval to shape the uncontrolled ramp-down set (D_{k-}^u). It should be noted that the forecast algorithm has already included the controlled house for this interval into the *on* loads over the next interval.

The controller then selects the available loads (A_{k-}) regarding (4.21). This set contains the loads that were not selected to be turned off in the previous control interval. Half of this set will be selected randomly to be off in order to provide ramp-up capacity for the interval after this interval ($k + 1$).

By turning off the loads at each time interval, the controller produces ramp-up (P^U) and ramp-down (P^D) reserve capacities. These capacities can be forecasted by forecasting the on/off status of the individual loads, and then can be measured using (4.8) and (4.9).

The average ramp-up ($\bar{P}^U$) and ramp-down ($\bar{P}^D$) reserve capacities of this controller were measured over 3 weeks of unmanaged load profiles. The average fore-

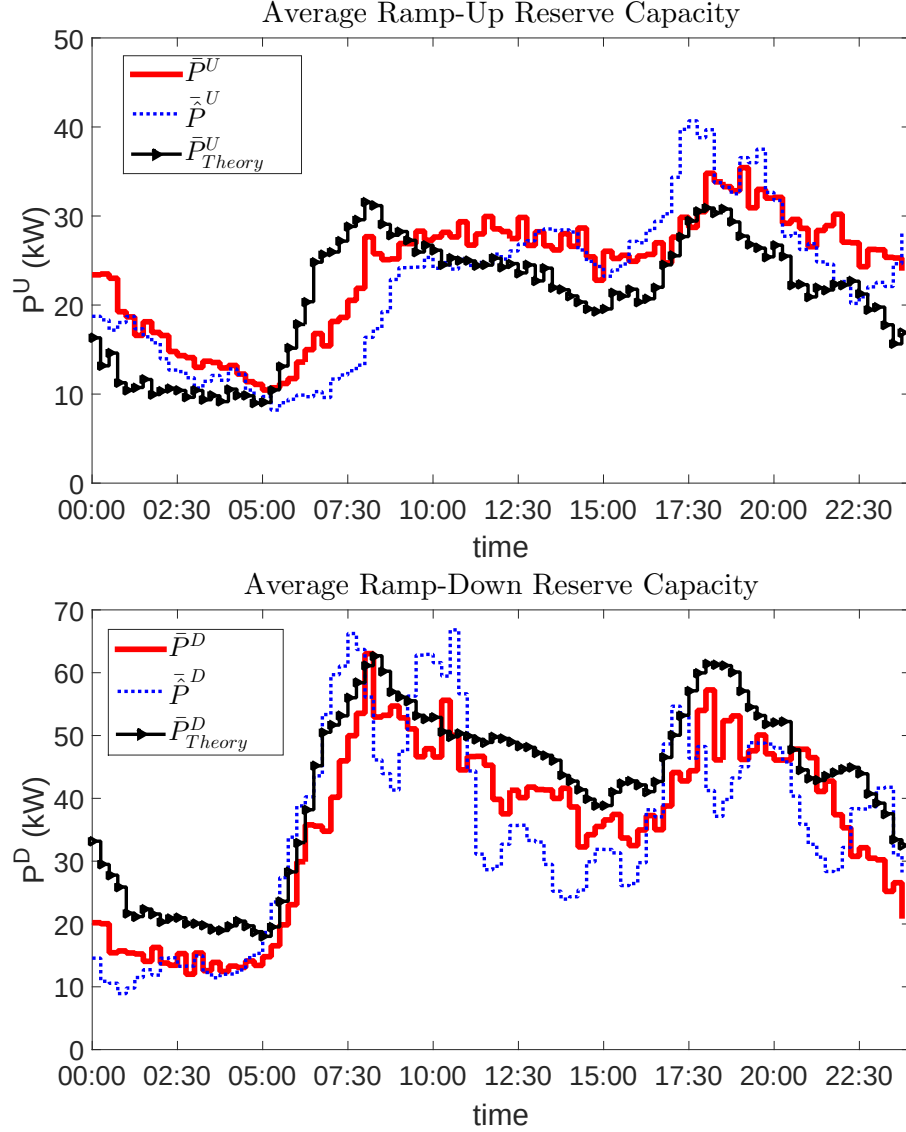


Figure 4.9: The average measured, forecasted, and estimated ramp-up and ramp-down reserve capacities.

casted ($\hat{\bar{P}}^U$ and $\hat{\bar{P}}^D$) and measured reserve capacities during each interval in a day are depicted in Figure 4.9, along with the values predicted by the mathematical model ($\bar{P}_{theory}^U$ and $\bar{P}_{theory}^D$) of the unmanaged reserve capacity by (4.33). As can be seen in this figure, the mathematical model provides a good estimation of the unmanaged reserve capacities.

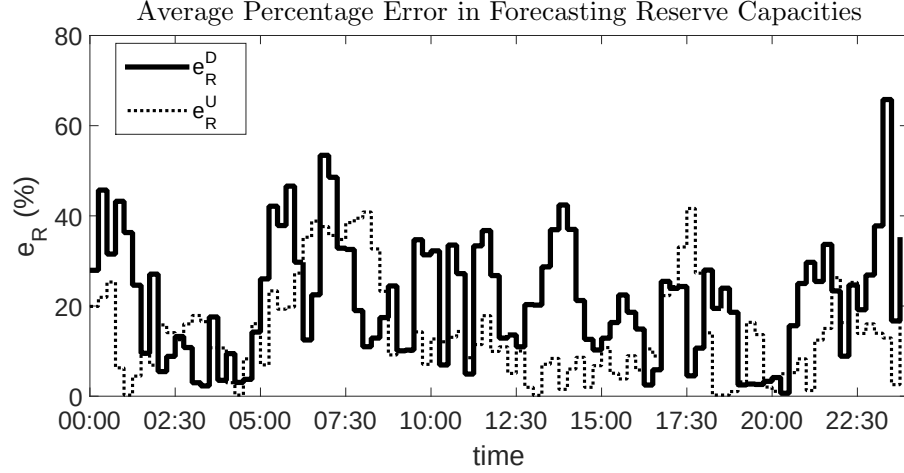


Figure 4.10: The mean absolute percentage error in forecasting the reserve capacities.

The forecasting absolute percentage error of the reserve capacities are calculated by:

$$\begin{aligned} e_R^D &= 100 \times \frac{|\hat{P}_k^D - P_k^D|}{\hat{P}_k^D} \\ e_R^U &= 100 \times \frac{|\hat{P}_k^U - P_k^D|}{\hat{P}_k^U} \end{aligned} \quad (4.35)$$

The average of the absolute percentage error in forecasting the reserve capacities is depicted in Figure 4.10.

4.5.2 Control System Performance

To evaluate the performance of the load following capability of the balance-reserved control strategy, a set-point load profile was given to the controller for each time interval, called desired aggregated load (DAL).

The objectives of the controller, as described in (4.22), are to control the loads in order to follow the DAL set points (J_k^1) and to provide both ramp-up and ramp-down reserve capacity by intentionally turning off the available loads (J_k^2). A forecast of

the individual thermostat status ($\hat{s}_k^i$) of each individual water heater is required in order to minimize the controller function (4.22) and form the ramp-up and ramp-down reserve sets. These forecasts can be computed using the methods introduced in Section 3.2. It should be noted that due to the definitions of the reserve capacities, the customer comfort is guaranteed.

Figure 4.11 illustrates a snapshot of this operation. In this figure, the target set points (DAL) and the measured load profile (the controlled load profile) are depicted on the top and the absolute percentage error is depicted in the bottom. The absolute percentage error is calculated by:

$$e_R = 100 \times \frac{|P_k^{desired} - P_k|}{P_k^{desired}} \quad (4.36)$$

This controller was operating over 2 weeks. The mean absolute percentage error, $\bar{e}_R$, as depicted in Figure 4.12, was calculated across each of 14 days .

It can be observed that the average error over a day is approximately $8kW$ and the average MAPE is approximately 20%. Compared to the basic strategy, the balanced-reserve strategy produces larger errors in following the DAL. One main reason is due to use of the ramp-down reserve capacity in order to provide ramp-up capacity. This operation reduces the capability to decrease the load in future. However, it is expected that with large number of controllable loads, the amount of reserve capacity increases, which provides better load following capacities.

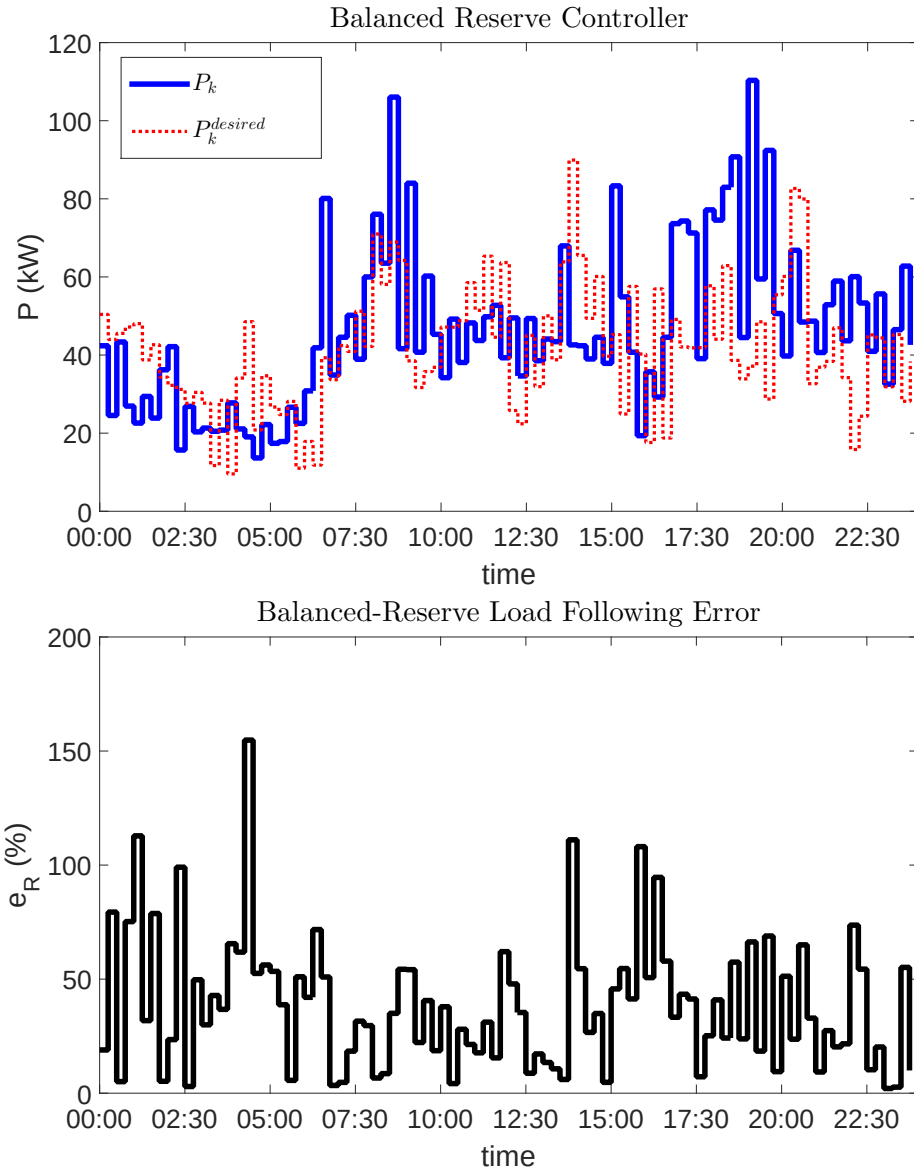


Figure 4.11: A sample snapshot of the balanced-reserve strategy trying to follow a target load profile (DAL)

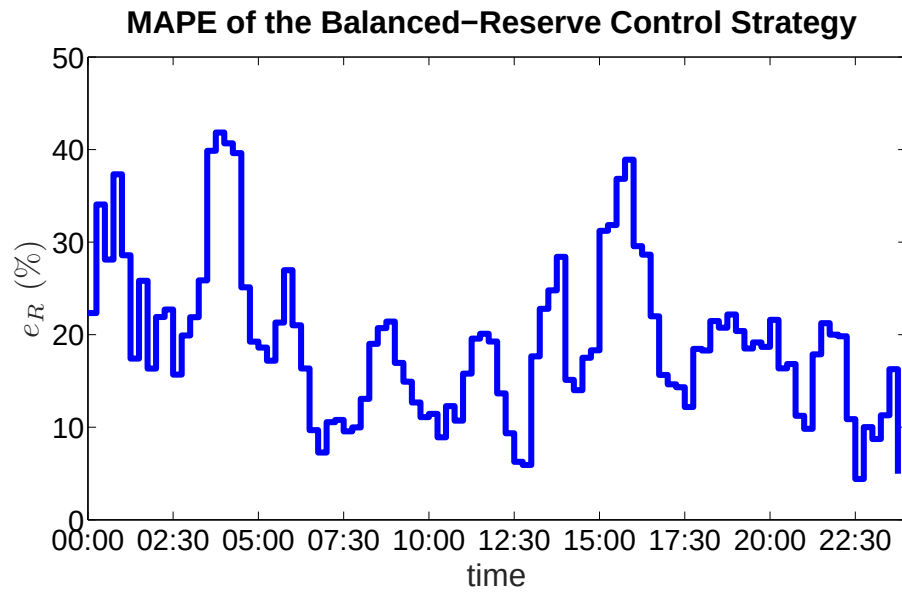


Figure 4.12: The mean absolute percentage error of the balanced-reserve strategy.

5 PowerShift Atlantic

A pilot project, called PowerShift Atlantic (PSA) [69], was started in 2009 and aimed to provide up to 20MW of ancillary service to balance the intermittency of renewable generation. The PSA project provided an experimental platform for testing and verifying the aggregated control system presented in Chapter 4.

The overall structure of the PSA project is illustrated in Figure 5.1. As can be seen in this figure, the PSA project has a hierarchical structure that makes it easily scalable by integrating more load aggregators with different load classes or different communication infrastructures.

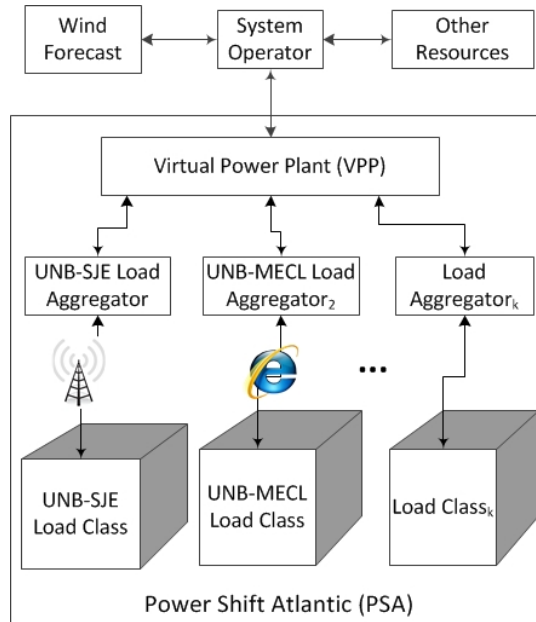


Figure 5.1: Structure of the PSA Project

There are various modules in the PSA architecture. The virtual power plant (VPP) is the system manager. It communicates with the system operator like a power

plant, using the same protocol as others. The VPP also communicates with various load aggregators. Each load aggregator controls a load class, such as residential water heaters or thermos-storage units, through a communication infrastructure. The designed controller in this dissertation was implemented as one of the load aggregators in Figure 5.1 that controls a water heater load class.

The rest of this chapter is organized as follows. Section 5.1 describes the structure of the VPP. Various load classes and communication infrastructures are introduced in Section 5.2. Section 5.3 illustrates the structure of the UNB Load aggregators, and the experimental results are presented in Section 5.4.

5.1 Virtual Power Plant

The concept of the virtual power plant (VPP) is based on the future use of distributed power generation for small energy production units [76]. A VPP may contain various generators, combine different types of renewable and non-renewable power generators and/or storage devices, and deliver them to the electricity market [77].

VPPs are one of the new strategies of integrating distributed energy resources into the power grid [78, 79]. The main characteristics of a VPP are its software-dependant nature, and its focus on energy-market participation [80].

The VPP plays a key role in the architecture of the PSA. It is responsible for managing the load aggregators and communicating with the system operator (SO). From the SO point of view, the VPP is a unique entity like a power plant that is capable of providing ancillary services, such as 10-minutes spinning reserve. The VPP regularly sends a load control plan and status update to the SO and receives forecast shapes and real-time dispatch instructions from the SO.

The VPP performs two key functions for the system operator:

1. It simplifies the system operators job of balancing the grid by instructing the aggregators how to redistribute load throughout the day, within feasibility constraints, in a manner that takes into account the variation in load and wind generation forecast provided by the system operator. This function (balancing the grid) will occur continuously throughout the day. The optimization required to determine the optimal load values for the aggregators will be conducted over a rolling 24 hour planning horizon.
2. It fulfills the equivalent of a 10-minute spinning reserve activation instruction by increasing or decreasing the total load, upon a real time request from the system operator. In this manner the VPP will provide a resource to the system operator that can be used to balance load with generation through manipulating the load. To the system operator the VPP will appear to provide 60 minutes of Generation-Up or Generation-Down, within 10 minutes of the receipt of the dispatch.

5.2 Controllable Load and Communication Infrastructure

The controllable load should have special characteristics to be used in a DSM program. It should be interruptible without influencing customers' comfort and should hold a large share of the total load in order to change the shape of the aggregated load.

There are three main load classes in the PSA: i) domestic electric water heaters (DEWHs), ii) electric thermal storage units (ETS), and iii) heating, ventilation and

Table 5.1: A summary of the various load classes in PSA

Class Name	Type	# of sites	# of devices
UNB-MECL	DEWH	135	97
UNB-SJE	DEWH	204	204
LAG1-LC1	DEWH	411	411
LAG2-LC11	DEWH	74	74
LAG2-LC22	DEWH	47	47
LAG2-LC33	DEWH	23	23
LAG2-LC44	ETS	27	86
LAG2-LC55	ETS	88	161
LAG3-LC1	ETS	138	353
LAG4-LC11	HVAC	22	22
LAG4-LC22	HVAC	4	4

air conditioning units (HVAC). Each load class (LC) has several groups of controllable end-use devices, and is assigned to various load aggregators (LAG). A summary of various load classes is listed in Table 5.1.

Among the various load classes listed in Table 5.1, two classes were assigned to the University of New Brunswick (UNB) to deploy the proposed load controller. These two classes are *UNB-MECL* and *UNB-SJE*. The first pilot was developed in Prince Edward Island by Maritime Electric Corporation Ltd (MECL), and the second pilot was developed in Saint John, NB, by Saint John Energy (SJE). The overall structure of the pilot projects for UNB is depicted in Figure 5.2. Each pilot has its own specifications, limitations, and infrastructure, as listed in Table 5.2. Thus, each one needs a unique load aggregator.

MECL Pilot:

MECL uses Sequentrics products to equip electric water heaters in Prince Edward Island with smart meters and load control modules (LCMs). These devices connect to a gateway through a dedicated wireless connection, and the gateway uses the Internet to send telemetry data and the LCM state to a server (called Sequentrics

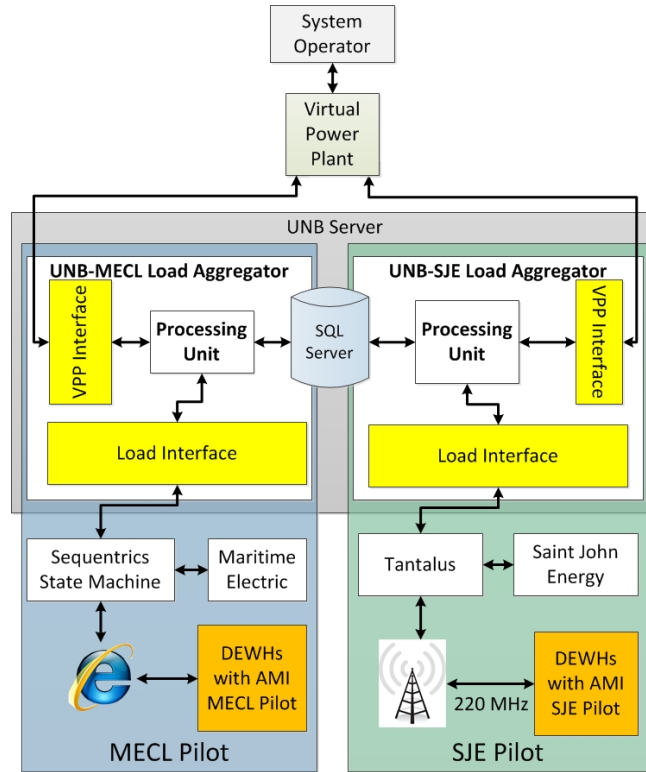


Figure 5.2: The two pilot projects for the University of New Brunswick in PSA (State Machine) at the utility. Other third party applications, such as UNB-MECL Load Aggregator, can communicate with each end-use device (DEWH) through a connection to this state machine, as illustrated in Figure 5.2.

In this pilot, the on/off status of each individual DEWH is available. The sampling time can be varied from 10 seconds to a month. In this project, a sampling time of one sample per minute was used. The communication delay for receiving the

Table 5.2: A comparison of the two load classes for UNB

	MECL	SJE
# of loads	96	195
Communication	Internet	Dedicated Wireless
Measured Parameter	DEWH On/Off State	Household Energy
Sampling Interval	1 minute	5 minutes
Average Measurement Delay	less than 1 second	45minutes
Average Control Delay	1 minute	10 minutes

measurements was less than a second. Due to the architecture of their system, the commands to the load could only be sent when a new measurement was coming. This limits the control speed to the sampling time (1 command per minute).

SJE Pilot:

Saint John Energy has equipped the houses with a smart meter and an LCM on the water heater. The LCM communicates with the smart meter and the smart meter uses a $220MHz$ wireless link to connect to the Tantalus server, as depicted in Figure 5.2. The hardware and infrastructure were developed by Tantalus.

Smart meters measure and send the energy consumption of each house at specific times. In this pilot, smart meters are set to sample every 5 minutes. The LCMs contain a controllable switch that can only be used to connect/disconnect the power line to the DEWH with no information transmitted about the on/off state of the DEWH. It was observed that the average delay in receiving smart meter data was around 45 minutes (due to the limitation on transmitting power, each smart meter can only send the measured data once every 90 minutes). However, the control signals took about 10 minutes to get to the load and be executed. These difficulties make the controller design more challenging. Without real-time measurements, the controller error may grow large.

5.3 Load Aggregator

The load aggregators are in direct communication with the VPP and the end-use devices (EUD) to control the EUD consumptions so that it follows the desired aggregated load from the VPP and maintains customer comfort.

The load aggregator has a modular design, as depicted in Figure 5.3. This figure shows the block diagram of the load aggregator for the MECL pilot. The main blocks of this system are:

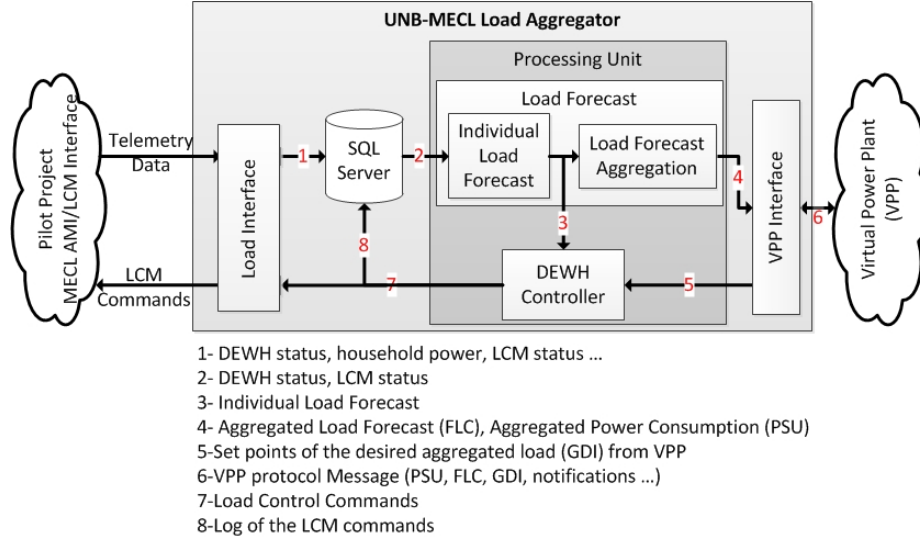


Figure 5.3: The structure of the load aggregator for the MECL pilot

1. **Load interface:** This module implements the communication protocol with the load. For each pilot, a different block can be designed. This module stores the telemetry data into a local database and also sends the control commands (turn on/off) to the loads.
2. **VPP interface:** This module implements a specific protocol to communicate with the VPP. According to this protocol, the load aggregator should send a load forecast (called forecast load capability, FLC) to the VPP every hour and receive the desired aggregated load (called gross dispatch instruction, GDI) every minute. The VPP should also send a status report on the aggregated power consumptions of the EUDs (called performance status updated, PSU). The PSUs and FLCs are generated by the load forecast module and shipped to this module to be transmitted to the VPP according to the communication protocol.

3. Processing Unit: This module implements the main parts of the load aggregator. It contains two main modules:

- Load forecast: This module generates a forecast of the load to be used by the controller and the VPP interface. It uses the telemetry data received by the load interface through a database. Details on the forecast methods are discussed in Chapter 3.
- DEWH controller: This module implements the actual load following controller that tracks the desired aggregated load, received from the VPP interface. The on/off commands will be sent to the loads through the load interface. Details of the control strategy is presented in Chapter 4

Figure 5.4 illustrates a snapshot of the GUI of the main window. It contains a section for the load forecast settings, a section for the controller setting, and a canvas to show the aggregated measured power (PSU), forecasts (FLC) and set points (GDI).

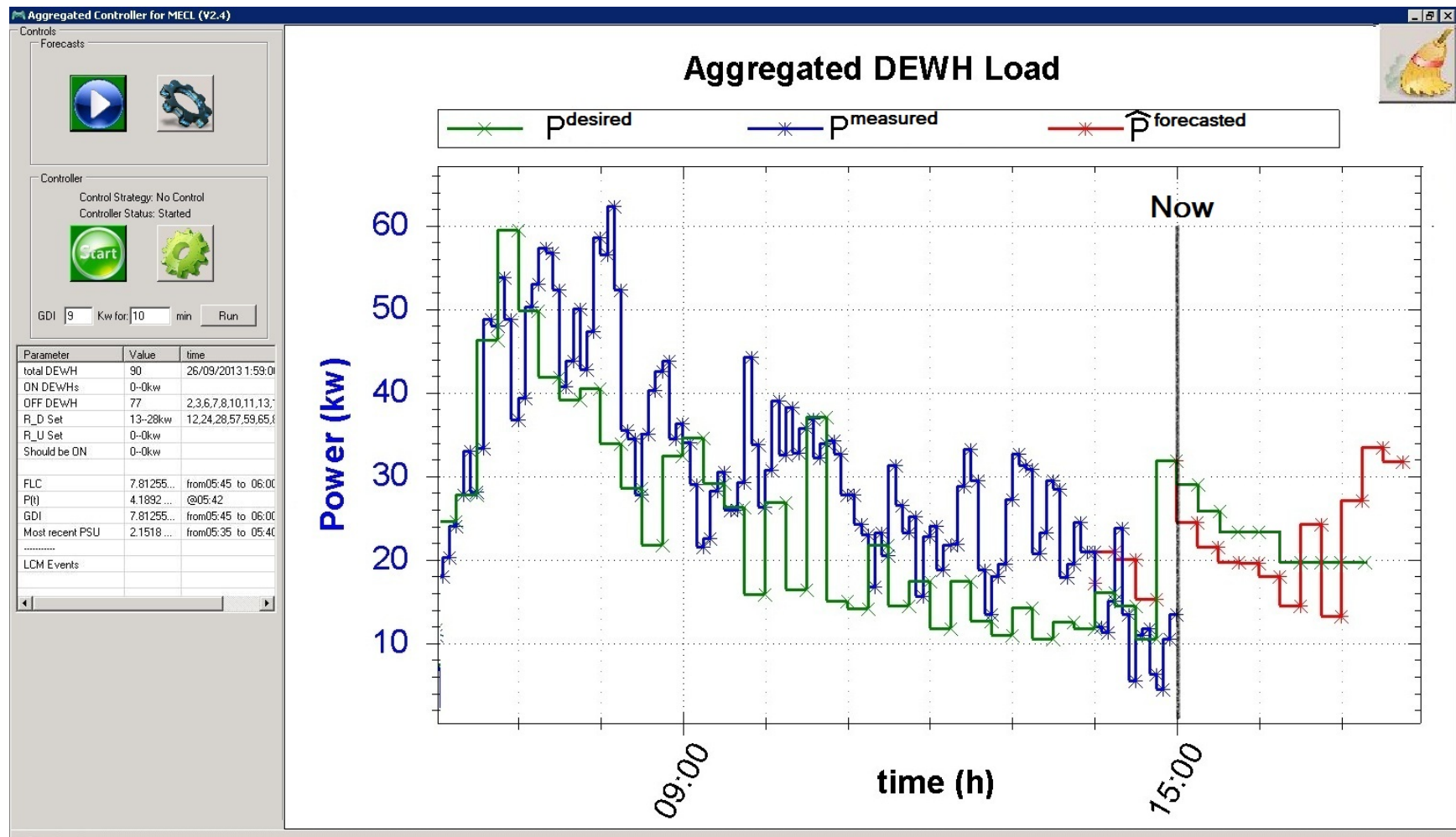


Figure 5.4: The GUI of the main window for the MECL pilot load aggregator.

5.4 Experimental Results

As mentioned in Section 5.2, two classes of the electric water heaters were provided for UNB to deploy the load aggregators. The experimental results of each pilot are explained in this section.

5.4.1 MECL Pilot

The structure of the load aggregator for the MECL pilot is depicted in Figure 5.3. The on/off status of each individual DEWH in this pilot is received by the load interface and stored in a local database. A load forecast module produces individual and aggregated load forecasts (as described in Chapter 3). The VPP interface sends the forecasted power consumptions and reserve capacities to the VPP and receives the set points in the desired aggregated load. The controller uses the set points and individual load forecasts to control the on/off status of each individual loads, as described in Chapter 4.

5.4.1.1 Load Forecast Evaluation

An unmanaged load profile, as described in Chapter 4, is more appropriate than a managed load to evaluate a forecasted load because in the managed load, the controller will change the power consumptions of the loads in order to follow a desired trajectory.

Figure 5.5 depicts one example of an unmanaged load profile. This figure shows the aggregated measured power consumption of the loads in the MECL pilot, overlaid with a forecasted aggregated power which was created at 6 : 00 am for a window of 4 hours.

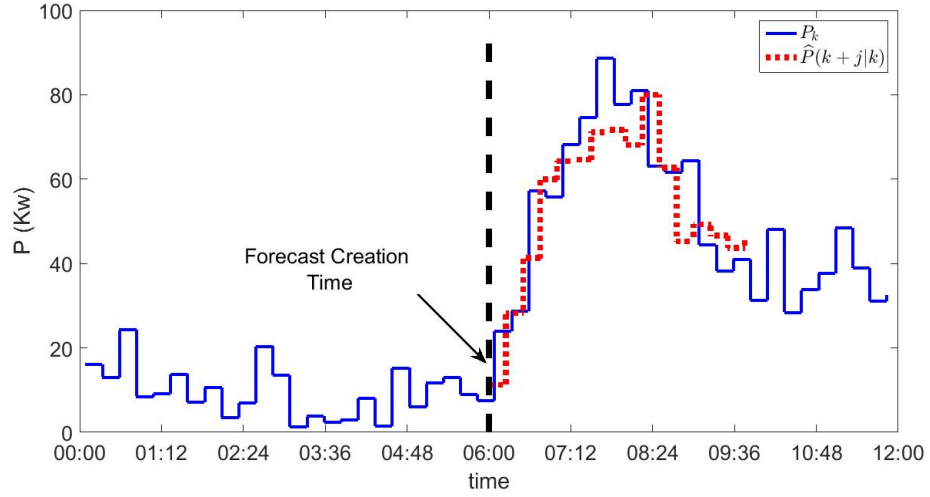


Figure 5.5: One example of a short term aggregated load forecast for uncontrolled aggregated load.

Let the k denote the time instant. The forecasted power at time $k + j$ that is created at time k is denoted by $\hat{P}(k + j|k)$, where j indicates the sample index in the forecast horizon: $1 \leq j \leq 16$ in this study. With this definition, the absolute percentage relative error in forecasting the j^{th} sample in the forecasting horizon can be obtained by:

$$e_R(j|k) = 100 \times \frac{|P(k + j) - \hat{P}(k + j|k)|}{P(k + j)} \quad 1 \leq j \leq 16 \quad (5.1)$$

where $P(k + j)$ denotes the measured power consumption at time $k + j$.

The mean absolute percentage error for each sample in the forecast horizon can be obtained by averaging the values of $e_R(j|k)$ for all values of k . Figure 5.6 depicts the mean absolute percentage error of each sample in the forecast horizon. This figure shows that the forecasting error varies between 12 – 15% over a 4-hour forecast horizon.

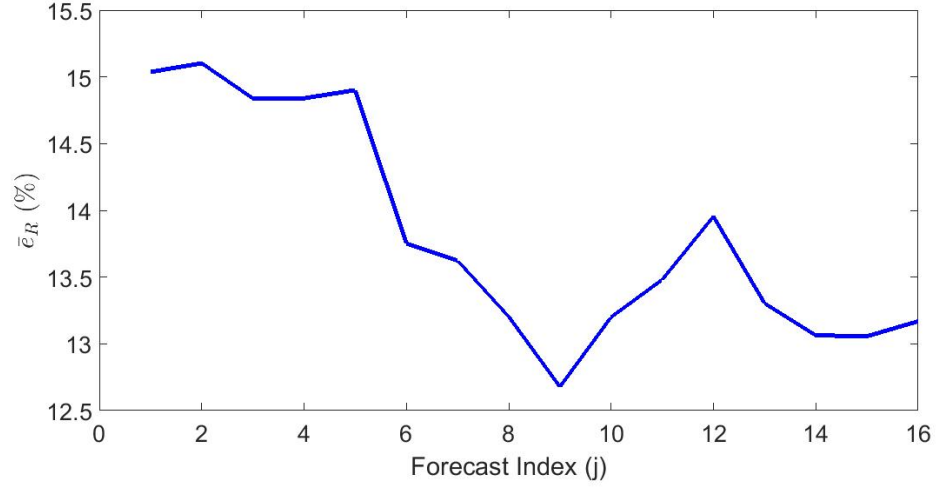


Figure 5.6: The mean absolute percentage error of each element in the forecast horizon.

5.4.1.2 Load Controller Performance

The basic load following algorithm, described in Chapter 4, was deployed in the MECL pilot for more than 8 months. During this time, the unmanaged reserve capacity of the controller was measured and compared with the forecasted reserve capacity, as depicted in Figure 5.7.

During the managed load profile, as described in Chapter 4, the controller will follow the desired aggregated load by strategically turning on and off the individual water heaters. Figure 5.8 depicts a snapshot of the operation of the controller.

At each time instant k , the controller uses the most recent forecast for this time, $\hat{P}(k|k-1)$, to determine the control signals for each house, u_k^i , in order to follow the given set point, $P_k^{desired}$.

The absolute percentage error of the load following can be calculated by

$$e_R(k) = 100 \times \frac{|P_k^{desired} - P_k|}{P_k^{desired}} \quad (5.2)$$

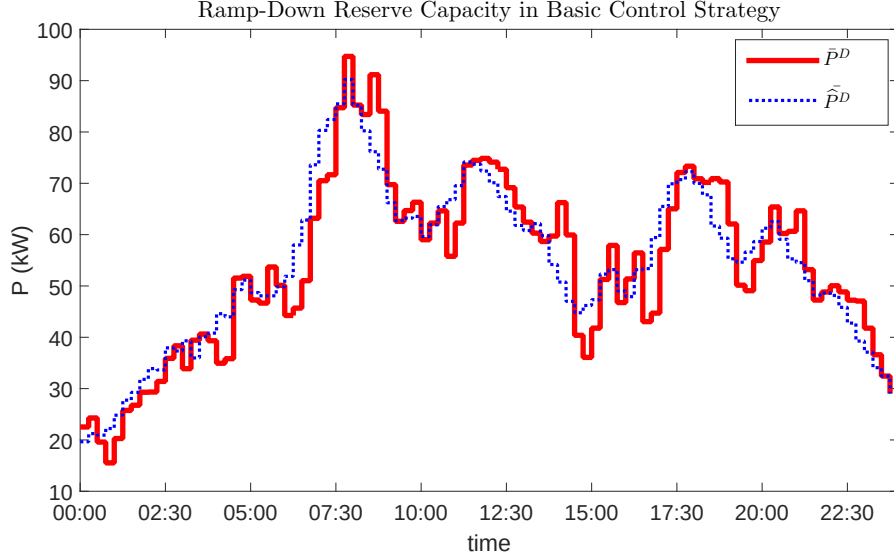


Figure 5.7: The average unmanaged ramp-down reserve capacity of the basic control strategy over a day (measured vs. forecasted)

And the mean absolute percentage error of the controller at each time in a day, can be calculated by computing (5.2) at that time in different days. Figure 5.9 illustrates the mean absolute percentage error of the controller over a day. As can be seen in this figure, the average load following error may vary within 3 – 15% with an average of 8%.

Remarks: It can be observed that the average error in following a desired aggregated load on the MECL pilot is around 8%, which is better than the forecast error (12-15%) because the controller is adjusting the power consumption of the loads.

It is also observed that the load following error may become large (upward approximately 30% or more, as depicted in Figure 4.4) at some time intervals. It is a consequence of the lack of reserve capacity to control. This is the payback effect of the control actions on the reserve capacity that still need to be studied and formulated.

It should also be noted that although the objective of the VPP is to shape the aggregated load of all load classes in a way to best utilize the renewable energies, it is

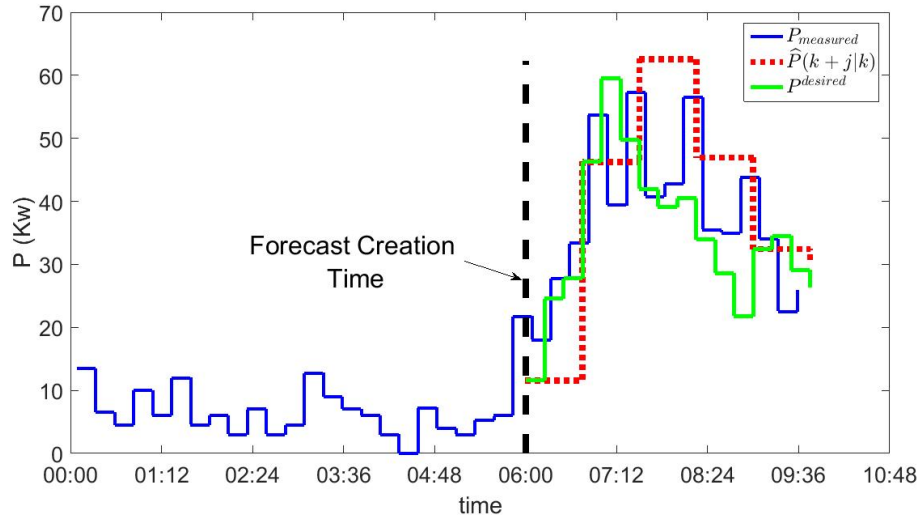


Figure 5.8: The controller changes the uncontrolled load profile to match the desired aggregated load.

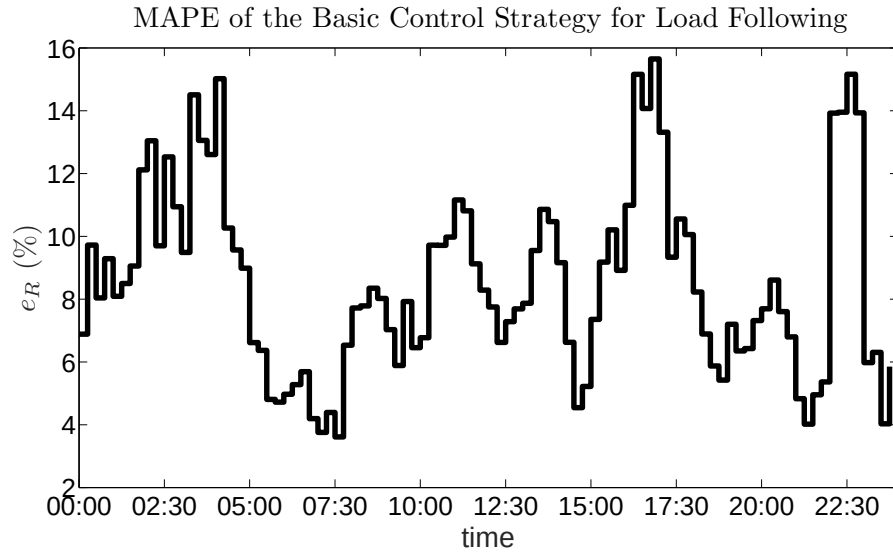


Figure 5.9: The mean absolute percentage error of the basic load following strategy

possible that this objective could result in a peak shaving over a day, as illustrated in Figure 5.10. This figure shows an uncontrolled aggregated load profile super-imposed with the controlled load profile during another similar day (with a similar daily peak). As can be seen in this figure, the morning peak is reduced by approximately $14kW$ and shifted to the off-peak time intervals.

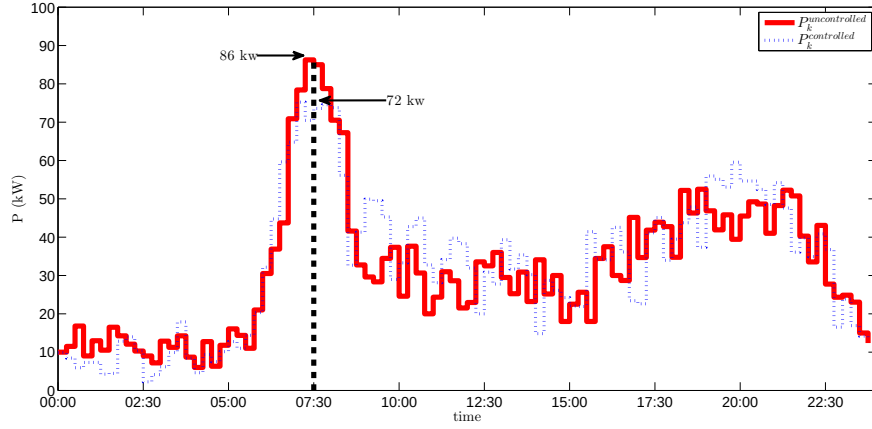


Figure 5.10: A comparison of the uncontrolled load with controlled load profile to illustrate a peak shaving

Another important remark is about the required computational power. As discussed in Chapter 3 and 4, two types of load forecasts are required in this system:

- Individual load forecast: where the forecast is created based on a look-up table of the probability distribution of the uncontrolled on-off time intervals, using (3.1)
- Aggregated load forecast: which is basically a trained neural network, so the computational time is just sum of a multiplication, as listed in (3.22-3.23)

Once the forecasts are created, the reserve sets are formed based on the previous control signals and forecasted individual loads, and since all data required is already stored in the RAM for these calculations, this operation will be fast.

The bottleneck of the operation is to load individual data from the database. As the number of loads increases, the size of the database grows, which increases the response time to the queries. For 100 loads, it takes up to 250 ms to load and store data from the database and calculate the individual forecasts and reserve capacities. Since the sampling time for creating forecasts is 5 minutes, with the

existing data structure and architecture of the software, the maximum possible loads will be 120,000.

5.4.2 SJE Pilot

The structure of the load aggregator for the SJE pilot is different from the MECL pilot due to the nature of measurement devices and communication infrastructure, as will be described in the following sections.

5.4.2.1 DEWH Load Extraction

In the SJE pilot, the AMI measures the energy consumption of the household. This value consists of the DEWH energy consumptions and all other appliances in the house.

Various methods, including ANN, were used to extract the DEWH load from the household load. The input layer of the ANN took the past 4 hours samples of the household power consumptions (48 input neurons) to estimate the current on/off status of the water heater (1 output neuron) with 12 neurons in the hidden layer. To obtain the actual DEWH on/off status for training the ANN, 4 houses were equipped with additional smart meters to measure the power consumption of their water heaters. Figure 5.11 depicts an example of the household power consumption ($P_{household}$), measured DEWH power consumption (P_{DEWH}), and extracted DEWH power consumption ($\hat{P}_{DEWH}$) in summer (top) and winter (bottom).

Let s_k denote the on/off status of a DEWH at time instant k , then, the power consumption of that water heater at time k will be $P_0 s_k$, where P_0 is the nominal power rating of that DEWH. Thus, estimating the power consumption of a water heater is equivalent to estimating the on/off status of the DEWH. Hence, the success

rate of a load extraction method can be defined as the number of successful on/off state estimations, which is the probability that the estimated state is equal to the actual state:

$$S_R = 100 \times Prob\{\hat{s}_k = s_k\} \quad (5.3)$$

It was observed that during summer, the DEWH load can be extracted from the household load with approximately 95% success rate. However, during winter, the success rate falls below 50% because the power consumption patterns of heating devices (such as baseboard heaters) are similar to the water heaters. This similarity makes it difficult to distinguish power consumption of a DEWH from other appliances in winter time.

5.4.2.2 Communication Delay

The smart meters continuously measure the energy consumption of the house. In this project, the devices were programmed to store the samples every 5 minutes. However, due to the limitation on the wireless communication, each smart meter was allowed to send the data once every 90 minutes, and different smart meters transmit data at different times.

Figure 5.12 depicts the delay of reading various measurements of one smart meter. As can be seen in this figure, the smart meter is reporting the measured values at approximately every 90 minutes. This means that on average, the last received measurement at any give time is approximately 45 minutes old.

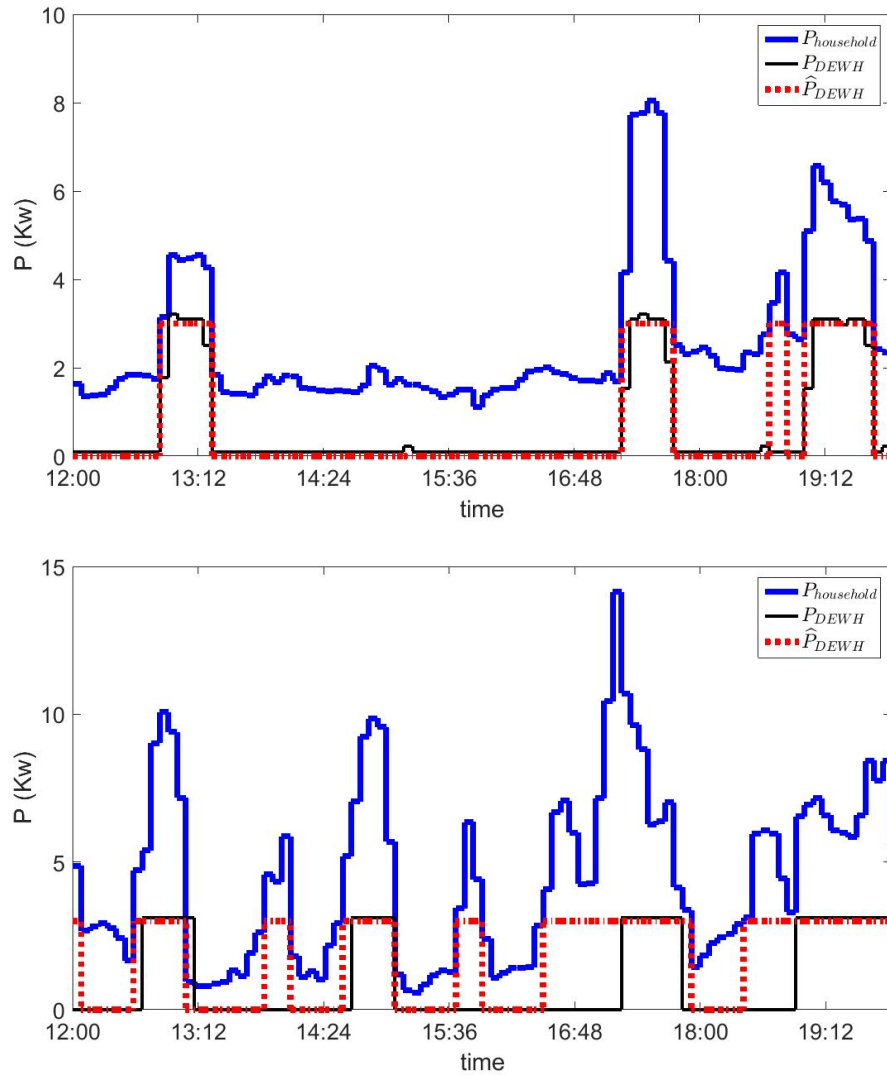


Figure 5.11: Extracting the DEWH load profile from the household load profile in summer (top) and winter (bottom)

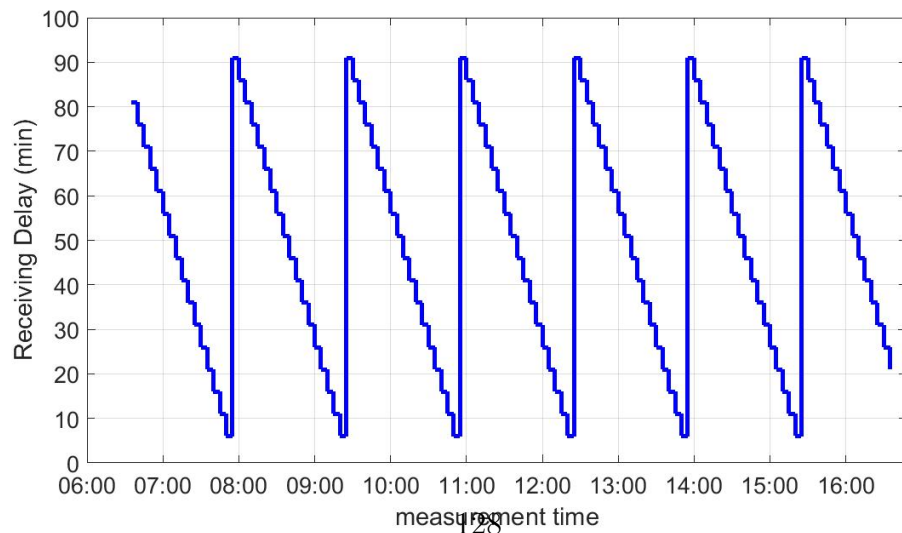


Figure 5.12: The communication delay of reading smart meter of one house

5.4.2.3 Unmanaged Load Forecast

Since the on/off state of the individual loads may not be available due to the load extraction difficulties, and also the telemetry data of the individual loads may not be available in real time due to communication delay, it is difficult to forecast the individual loads; thus, only an aggregated load forecast is created in this pilot, as described in 3.4.

Figure 5.13 depicts an example of an aggregated load forecast ($\hat{P}(k + j|k)$) overlaid with the actual aggregated measurements of household power consumption ($P_{measured}$). As can be seen in this figure, at the time instant when the forecast is created, the measurements of all loads may not be received due to the communication delay. This means that only the aggregated load profile of 90 – 100 minutes before the creation time can be used to create an aggregated load forecast. Therefore, at each time instant k , instead of having $\hat{P}(k + j|k)$, the forecasts are $\hat{P}(k + j|k - m)$, where $k - m$ is the last time when the telemetry data of all houses are received. However, for simplicity, the same notation ($\hat{P}(k + j|k)$) is used for the forecast that is generated at time k , knowing that it was using the data of approximately 2 hours earlier.

The absolute percentage error of each sample in the forecast horizon can be calculated by

$$e_R(j|k) = 100 \times \frac{|\hat{P}(k + j|k) - P(k + j)|}{\hat{P}(k + j|k)} \quad (5.4)$$

where k is a time index, and j belongs to $[1..16]$, showing a window of 4 hours with a 15-minute time interval.

The mean absolute percentage error of each sample in the forecast horizon, $\bar{e}_R(j)$ can be calculated by averaging $e_R(j|k)$ at different time instants, k . Figure 5.14 depicts the mean absolute percentage error of each forecast intervals in the forecast

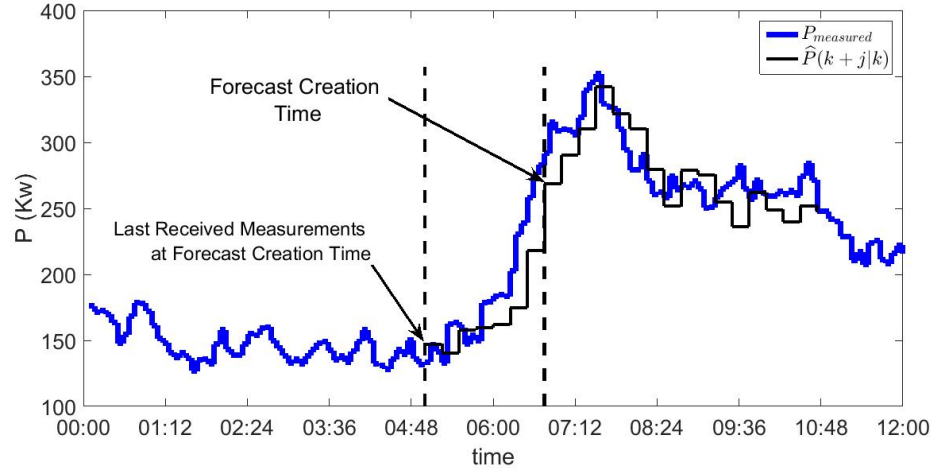


Figure 5.13: An example of the aggregated load forecast with delayed measurements.

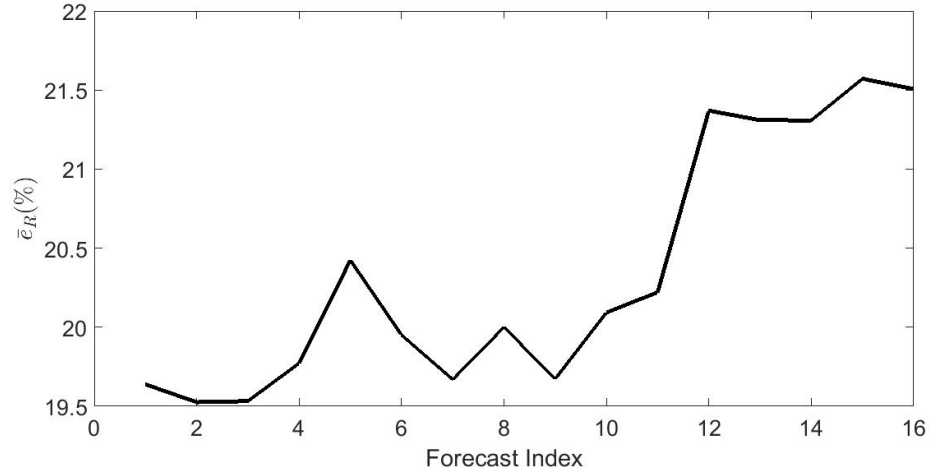


Figure 5.14: The mean absolute percentage error of each element in the forecast horizon.

horizon. As can be seen in this figure, the average relative forecast error may vary within 19 – 21% over a 4-hour forecast horizon.

5.4.2.4 Load Controller Performance

In the SJE pilot, load extraction produces unacceptable estimation errors. Even if the load extraction was acceptable, the large latency in receiving the measured data, from a feedback control perspective is unacceptable. The control algorithm, defined

in Chapter 4, requires an estimation of individual loads which cannot be created in the SJE pilot. Thus, an alternate aggregated load control has been implemented for the SJE pilot.

However, as described in Section 3.4.3, knowing the statistical characteristics of the water heaters can help to estimate the expected value of the power reduction obtained by a turn-off command to a group of loads and its payback effect after that.

Assuming that the water heaters on the SJE pilot have a similar statistical characteristics to the water heaters on the MECL pilot, it can be inferred that the $prob\{s_k = 1\}$ would be similar for both pilots, following the same pattern as shown in Figure 4.2.

Let $\hat{P}_k$ and $P_k^{desired}$ denote the forecasted power and the desired power at time k , respectively. Assuming the all DEWHs have similar nominal power rating (denoted by P_0), the number of loads that need to be controlled can be obtained by:

$$\mathcal{N}_k = \frac{\hat{P}_k - P_k^{desired}}{P_0} \quad (5.5)$$

Although the on/off states of the loads at time k are not available, the percentage of the total loads that are on, i.e. $Prob\{s_k = 1\}$, is known. This means that at time k , in a group of n_k DEWHs, there are $n_k \times Prob\{s_k = 1\}$ DEWHs that are on. Therefore, in order to have $\mathcal{N}_k$ 'turned on' water heaters in the control set, a group with n_k random loads can be selected where:

$$n_k = \frac{\mathcal{N}_k}{Prob\{s_k = 1\}} \quad (5.6)$$

$$\Rightarrow n_k = \frac{\hat{P}_k - P_k^{desired}}{P_0 \times Prob\{s_k = 1\}} \quad (5.7)$$

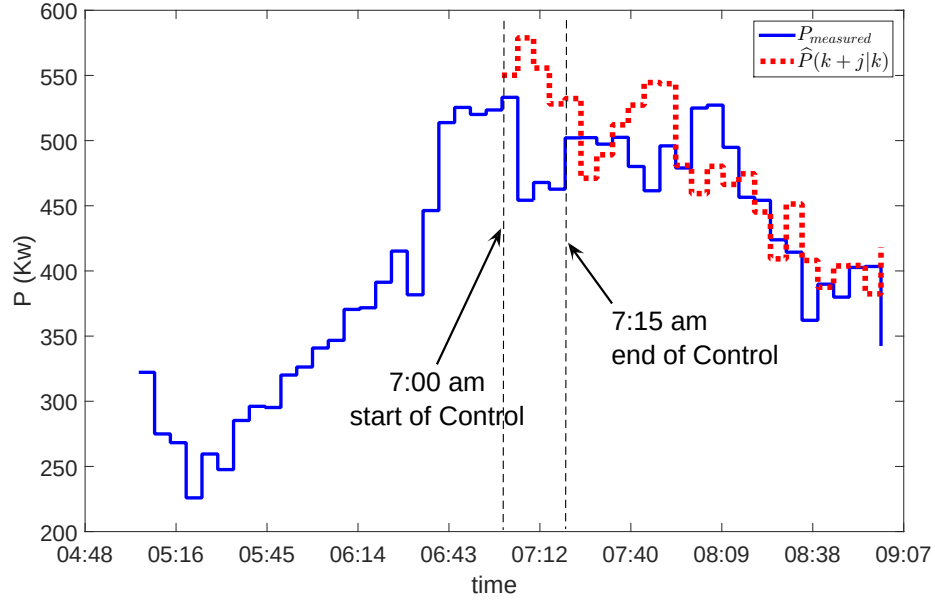


Figure 5.15: Impact of a turn-off command for all water heaters on the aggregated household power consumption.

As an example, based on Figure 4.2, at 7 : 00 am, approximately 25% of the water heaters are on; thus, in order to reduce $150kW$ from the forecasted load (i.e. $\hat{P}_k - P_k^{desired} = 150kW$), according to (5.7), 200 loads should be selected to be turned off. Figure 5.15 depicts an experiment in which 200 water heaters were interrupted at 7 : 00 am, overlayed with an uncontrolled aggregated load forecast (using the methods explained in Section 3.4), generated before the control action. As can be seen in this figure, the aggregated household loads have dropped approximately $100kW$ to $140kW$ during this interval, which validates the assumption of similar characteristics of the water heaters.

Using the statistical characteristics of the water heaters on the MECL pilot and adapting it to the SJE pilot and using the payback model introduced in Section 3.4.3, the controller can determine how many water heaters should be selected to be turned on or off in order to follow a desired aggregated load. Figure 5.16 depicts a snapshot of the operation of this controller.

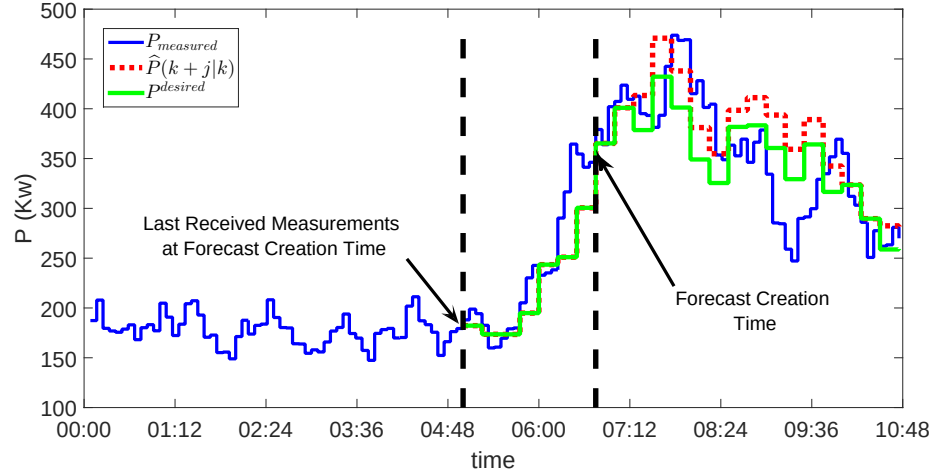


Figure 5.16: An example of the aggregated load following controller in the SJE pilot)

The absolute percentage error in following a desired aggregated load at each time instant, k , can be calculated using:

$$e_R(k) = 100 \times \frac{|P_k^{desired} - P_k|}{P_k^{desired}} \quad (5.8)$$

The mean absolute percentage error, $\bar{e}_R$, is calculated by averaging the e_R at each time instant in a day across several days. Figure 5.17 depicts the MAPE of the controller in the SJE pilot. This figure shows that on average, the load following error is approximately 20%.

Remarks: Experimental results show that the average error in following the desired aggregated load in the SJE pilot is larger than MECL pilot due to various reasons:

- The large communication delay expands the forecast horizon, which increases the error in the forecasted load and reserve capacities.
- Due to lack of individual on/off states in this pilot, the controller sends the control signals to a larger group of loads according to (5.7). This action limits

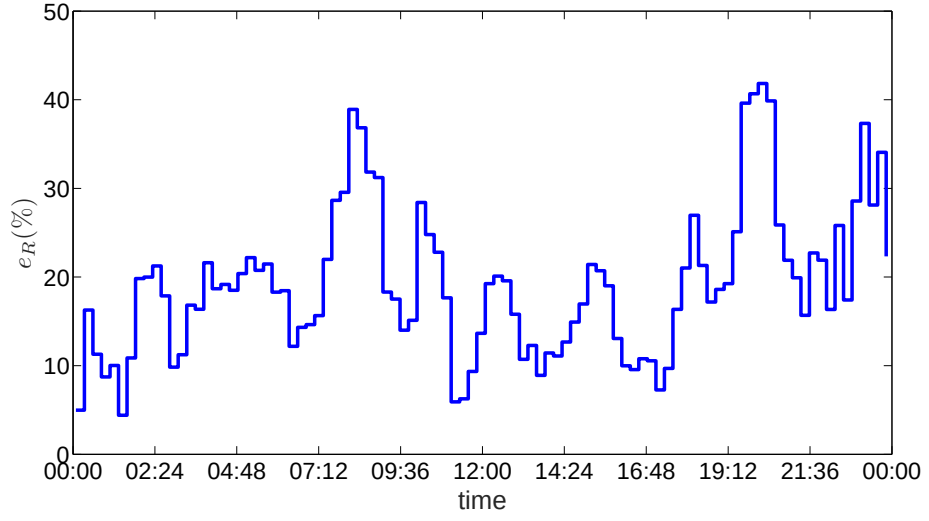


Figure 5.17: The mean absolute percentage error of the load following on SJE pilot

the possibility of using those loads in the future intervals as a reserve set, due to the customer comfort constraints introduced in (4.2) and (4.3). Thus, the controller may not be able to have as much capacity as predicted which results in a larger error in following the desired load.

Despite the limitations imposed by the infrastructure to the SJE pilot, it was a successful demonstration using direct load control but with 20% error in forecasting the load and tracking the desired load. Although the tracking errors are large, there is still value with using such a control method when, for example, peak shaving is an objective.

6 Conclusions

6.1 Conclusions

Demand-side management (DSM) is a technique to adjust the consumption profile to match generation. Studies show that DSM can be used to reduce energy bills of the customers [10], and can ease power system operation through peak shaving, load following, providing reserve capacity [11, 12] and fast response ancillary services [13].

Direct load control (DLC) is a method to manage the demand side. Domestic electric water heaters (DEWHs) are feasible candidates for a DLC program because the aggregated load profile of the DEWH i) contributes significantly to the total aggregated residential load, and ii) follows similar pattern and peak times. Therefore, changing the shape of the aggregated DEWH load profile can reshape the total load profile and follow a desired aggregated load.

However, it should be noted that due to the internal thermostatic control of the DEWH as well as the requirement to maintain level of customer comfort, a DEWH may not always be available for control, which reduces the capacity of the controller to increase or decrease the total power consumption.

To demonstrate the engineering merits of the research, the presented methodology was deployed on a pilot project called PowerShift Atlantic (PSA), led by the provincial utilities, NB Power, Saint John Energy, Maritime Electric, and NS Power. The objective of the PSA was to provide up to 20MW of load following ancillary services to balance the intermittency of renewable generation through directly controlling

the power consumption of various load classes, such as electric water heaters and electric thermal storage units. The presented methodology was deployed as a load aggregator in the PSA project to model and control 300 residential water heaters, and to assess performance in real-time operation.

The contributions of this research can be summarized as follow:

1. Load Modeling and Simulator: Despite vast studies on modeling the electric water heaters [27, 19, 28, 15, 29, 30], none of them presented a methodology to identify the model parameters and water usage of an electric water heater in a pilot in order to simulate and mimic its behaviour. In this dissertation, a methodology is presented to estimate the DEWH model parameters and its water usage pattern through the measurements of the DEWH power consumption, and if available, its water temperature. Through this methodology, a model for each individual DEWH in any given pilot can be identified. Then, this model can be used to build a simulator to study the performance of the controller, before deploying it on the real loads. Experimental results show that the simulator can mimic the behaviour of the individual loads in the presence of the external control signals. The average relative error of the simulator is less than 10%.
2. Load Forecast: The main challenge of load forecasting in a DLC is how the control actions changes the forecasts. None of the previously cited research [33, 34, 36, 37, 39, 40] addresses the effect of a control action on the forecast nor the use of an altered historical load (due to a control action) to generate an uncontrolled load forecast. This research is distinguished from others by taking into account the payback effect of the control actions on the forecasting algorithms. In this research, a payback model is presented and validated

through experimental results for both individual load forecasts and aggregated forecasts.

A statistic-based method is presented for individual load forecasts, which produces a higher success rate (on average, 15% higher) over the short-term forecast horizon (4-hour ahead) compared to an ANN-based load forecast. Also, a mathematical model is presented to estimate the relative error of aggregating individual forecasts, which shows that the relative error can grow large, compared to forecasting the aggregated load (from 10% to 60% for the short term aggregated load forecast).

A Kalman-Filtered ANN-based method is developed to forecast the aggregated load, where the forecast error is within 12 – 15%. The aggregated payback model can also predict the change in the forecasted load profile with an error of 4 – 8%.

3. Load Control: Despite the variety of the proposed controllers [12, 51, 52, 54, 64, 63], none of them address the calculations of the forecasted reserve capacity or the impact of control actions on the behaviour of the load. Therefore, a new methodology is required to implement a load control that models the future consumption of the load and considers previous control actions to generate reserve capacity and follow the aggregated set points.

A basic strategy for following the desired aggregated load is presented which can only produce ramp-down reserve capacity. The model predicts that on a daily basis, it can produce approximately 30 – 90kW of ramp-down reserve capacity at different times of a day by controlling a set of 100 DEWHs. This predication is validated through experimental results. The experimental results also show that this controller can follow the desired aggregated load within 8 – 15% error.

This control strategy was modified to be applied on a second pilot with approximately 200 DEWHs with large communication delay and lack of measurements of DEWH power consumptions. Experimental results show that despite these limitations, the controller can follow the desired load with approximately 20 – 30% relative error.

Another control strategy, called balanced-reserve, is also formulated to produce both ramp-up and ramp-down capacities. The model predicts that on a daily basis, for a set of 100 DEWHs, this control strategy can produce approximately 20 – 60kW and 10 – 30kW for ramp-down and ramp-up reserves, respectively, which is validated by the experimental results. It was also observed that this controller can track the desired load within 5 – 40%.

6.2 Future Work

Although this research resulted in deployment of a load controller on a pilot project, there is still room for improving the process. In addition to that, during this research, new challenges evolved that require further investigation. Some of these challenges can be summarized as follow:

1. In order to maintain customers' comfort, the number of disruptions of a water heater was limited to 4 interruptions in a 2-hour window, as mentioned in 4.1.2. Deployment of this controller in the pilot caused no complaints from the customers. This means that for the next implementation, the imposed limit can be pushed to have more interrupts which may produce more reserve capacity.
2. The scalability of the presented system depends on the data processing sections, which are 1) storing metering data of each house, 2) retrieving historical data

to produce a load forecast (either individual or aggregated). As the number of participants increases, the complexity of the data storage grows bigger, and new techniques for handling big data will be required.

3. Enhancing the load forecast algorithms: Despite the vast studies on load forecast algorithms and methods, it is still an open problem. The performance of the load forecast can always be improved either by using better algorithms or finding better patterns in the historical data.
4. Increase the control horizon: Currently, the cost functions of the controller are just considering one time interval. Increasing the control horizon may enable the controller to balance its reserve capacity for longer time, which may result in better load following capabilities.
5. Payback effect on the reserve capacity: This research only focused on the impact of the control actions on the forecasted load and how the shape of the power consumption may change due to various control actions. However, the control actions also consume the reserve capacities, which will reduce the forecasted reserved capacities for the next intervals. Modeling the reserve capacity based on the control actions can produce better estimation on the strength of a control strategy.
6. Peak shaving algorithm: Although through the load forecast, the peak of the load can be determined and shaved by setting an appropriate desired aggregated load, it might be useful that the controller itself, without any other set point could shave the peak of demand and shift it to an off-peak time interval.
7. Load extraction: If the on/off state of the controllable load is not available, then the controller should estimate it from the measurement of the household energy consumption. This is a challenging task, especially when there are

other devices with similar pattern of power consumption in the house. For example, during winter, the power consumption of the baseboard heaters are similar to the power consumption of the DEWH. This makes it very difficult to disaggregate household power consumption into various load classes.

References

- [1] T. Janssen and Y. Rebours, “A standard framework to analyze electricity market couplings and system operator coordination,” in *Proc. of 9th International Conference on the European Energy Market (EEM)*, Florence, Italy, May 2012, pp. 1–7.
- [2] Y. Han and L. Chang, “A study of the reduction of the regional aggregated wind power forecast error by spatial smoothing effects in the maritimes Canada,” in *Electric Power and Energy Conference (EPEC), 2010 IEEE*, Aug 2010, pp. 1–6.
- [3] New Brunswick System Operator (NBSO). (2007) Maritime area wind integration study. Accessed: 2014-10-30. [Online]. Available: <http://goo.gl/OkCbGe>
- [4] C. Chompoo-inwai, W.-J. Lee, P. Fuangfoo, M. Williams, and J. R. Liao, “System impact study for the interconnection of wind generation and utility system,” *IEEE Transactions on Industry Applications*, vol. 41, no. 1, pp. 163–168, Jan 2005.
- [5] A. O. B. L. H. Holttinen, Peter Meibom, “Impacts of large amounts of wind power on design and operation of power systems, results of IEA collaboration,” *Wind Energy*, vol. 14, pp. 179–192, 2011.

- [6] P. S. Moura and A. T. de Almeida, “Multi-objective optimization of a mixed renewable system with demand-side management,” *Renewable and Sustainable Energy Reviews*, vol. 14, pp. 1461–1468, 2010.
- [7] C. Schneller, “Integrating wind power into the power supply system,” in *Intelligent Policy Option*, Germany, 20200415, pp. 1–8.
- [8] B. Burger, “Electricity production from solar and wind in Germany in 2014,” in *Fraunhofer ISE*, Freiburg, Germany, Dec 2014, pp. 1–8.
- [9] D. He, J. Mei, R. Harley, and T. Habeter, “Utilizing building-level demand response in frequency regulation of actual microgrids,” in *Industrial Electronics Society, IECON 2013 - 39th Annual Conference of the IEEE*, Nov 2013, pp. 2205–2210.
- [10] N. Zhang, L. Ochoa, and D. Kirschen, “Investigating the impact of demand side management on residential customers,” in *Innovative Smart Grid Technologies (ISGT Europe), 2011 2nd IEEE PES International Conference and Exhibition on*, Dec 2011, pp. 1–6.
- [11] K. Schisler, T. Sick, and K. Brief, “The role of demand response in ancillary services markets,” in *Transmission and Distribution Conference and Exposition, IEEE/PES*, April 2008, pp. 1–3.
- [12] J. Kondoh, N. Lu, and D. J. Hammerstrom, “An evaluation of the water heater load potential for providing regulation service,” *IEEE Transactions on Power Systems*, vol. 26, no. 3, pp. 1309–1316, Aug 2011.
- [13] I. Koutsopoulos and L. Tassiulas, “Challenges in demand load control for the smart grid,” *IEEE Network*, vol. 25, no. 5, pp. 16–21, September 2011.
- [14] J. Kondoh, “Direct load control for wind power integration,” in *2011 IEEE Power and Energy Society General Meeting*, July 2011, pp. 1–8.

- [15] M. H. Nehrir, R. Jia, D. A. Pierre, and D. J. Hammerstrom, "Power management of aggregate electric water heater loads by voltage control," in *Power Engineering Society General Meeting, 2007. IEEE*, June 2007, pp. 1–6.
- [16] M. H. Nehrir, B. J. LaMeres, and V. Gerez, "A customer-interactive electric water heater demand-side management strategy using fuzzy logic," in *Power Engineering Society 1999 Winter Meeting, IEEE*, vol. 1, Jan 1999, pp. 433–436 vol.1.
- [17] S. H. Lee and C. L. Wilkins, "A practical approach to appliance load control analysis: A water heater case study," *IEEE Transactions on Power Apparatus and Systems*, vol. PAS-102, no. 4, pp. 1007–1013, April 1983.
- [18] M. Kobe and A. Tsoi, "Modelling of domestic hot-water heater load from on-line operating records and some applications," *Generation, Transmission and Distribution, IEE Proceedings C*, vol. 133, no. 6, pp. 336–345, September 1986.
- [19] I. Lane and N. Beute, "A model of the domestic hot water load," *Power Systems, IEEE Transactions on*, vol. 11, no. 4, pp. 1850–1855, Nov 1996.
- [20] M. Gustafson, J. Baylor, and G. Epstein, "Direct water heater load control-estimating program effectiveness using an engineering model," *Power Systems, IEEE Transactions on*, vol. 8, no. 1, pp. 137–143, Feb 1993.
- [21] J. van Tonder and I. Lane, "A load model to support demand management decisions on domestic storage water heater control strategy," *Power Systems, IEEE Transactions on*, vol. 11, no. 4, pp. 1844–1849, Nov 1996.
- [22] J. Fernandez-Seara, F. Uhia, and J. Sieres, "Experimental analysis of a domestic electric hot water storage tank. Part I: Static mode of operation," *Applied Thermal Engineering*, vol. 27, no. 1, pp. 129–136, 2007.

- [23] J. Fernandez—Seara, F. Uhia, and J. Sieres, “Experimental analysis of a domestic electric hot water storage tank. Part II: dynamic mode of operation,” *Applied Thermal Engineering*, vol. 27, no. 1, pp. 137–144, 2007.
- [24] E. Vrettos, S. Koch, and G. Andersson, “Load frequency control by aggregations of thermally stratified electric water heaters,” in *Innovative Smart Grid Technologies (ISGT Europe), 2012 3rd IEEE PES International Conference and Exhibition on*, Oct 2012, pp. 1–8.
- [25] Z. Xu, R. Diao, S. Lu, J. Lian, and Y. Zhang, “Modeling of electric water heaters for demand response: A baseline PDE model,” *Smart Grid, IEEE Transactions on*, vol. 5, no. 5, pp. 2203–2210, Sept 2014.
- [26] P. Dolan, M. Nehrir, and V. Gerez, “Development of a Monte Carlo based aggregate model for residential electric water heater loads,” *Electric Power Systems Research*, vol. 36, no. 1, pp. 29 – 35, 1996.
- [27] J. Laurent and R. Malhame, “A physically-based computer model of aggregate electric water heating loads,” *Power Systems, IEEE Transactions on*, vol. 9, no. 3, pp. 1209–1217, Aug 1994.
- [28] R. Jia, M. Nehrir, and D. Pierre, “Voltage control of aggregate electric water heater load for distribution system peak load shaving using field data,” in *In Proc. of 39th North American Power Symposium*, 2007, pp. 492 –497.
- [29] K. Elgazzar, H. Li, and L. Chang, “A centralized fuzzy controller for aggregated control of domestic water heaters,” in *Electrical and Computer Engineering, 2009. CCECE '09. Canadian Conference on*, May 2009, pp. 1141–1146.
- [30] R. Diao, S. Lu, M. Elizondo, E. Mayhorn, Y. Zhang, and N. Samaan, “Electric water heater modeling and control strategies for demand response,” in *Power and Energy Society General Meeting, 2012 IEEE*, July 2012, pp. 1–8.

- [31] L. Paull, H. Li, and L. Chang, “A novel domestic electric water heater model for a multi-objective demand side management program,” *Electric Power Systems Research*, vol. 80, no. 12, pp. 1446 – 1451, 2010.
- [32] S. Rahman and M. Baba, “An integrated load forecasting-load management simulator: its design and performance,” *Power Systems, IEEE Transactions on*, vol. 4, no. 1, pp. 184–189, Feb 1989.
- [33] M. Cho, J. Hwang, and C. Chen, “Customer short term load forecasting by using ARIMA transfer function model,” in *Energy Management and Power Delivery, 1995. Proceedings of EMPD '95., 1995 International Conference on*, vol. 1, Nov 1995, pp. 317–322 vol.1.
- [34] W. Sun and X. Zhang, “Application of self-organizing combination forecasting method in power load forecast,” in *Wavelet Analysis and Pattern Recognition, 2007. ICWAPR '07. International Conference on*, vol. 2, Nov 2007, pp. 613–617.
- [35] N. Dong-Xiao, G. Zhi-Hong, X. Mian, and W. Hui-Qing, “Approach to daily load forecast of VSNN based on data mining,” in *Industrial Electronics and Applications, 2007. ICIEA 2007. 2nd IEEE Conference on*, May 2007, pp. 363–368.
- [36] G. Tso and K. Yau, “Predicting electricity energy consumption: A comparison of regression analysis, decision tree and neural networks,” *Energy*, vol. 32, no. 1, pp. 1761–1768, 2007.
- [37] H. Daneshi and A. Daneshi, “Real time load forecast in power system,” in *Electric Utility Deregulation and Restructuring and Power Technologies, 2008. DRPT 2008. Third International Conference on*, April 2008, pp. 689–695.

- [38] T. Olofsson and S. Andersson, “Long-term energy demand predictions based on short-term measured data,” *Energy and Buildings*, vol. 33, no. 2, pp. 85–91, 2001.
- [39] L. Ekonomou, “Greek long-term energy consumption prediction using artificial neural networks,” *Energy*, vol. 35, no. 2, pp. 512–517, 2010.
- [40] K. Pramelakumari, S. Anand, V. Raj, and E. Jasmin, “Short-term load forecast of a low load factor power system for optimization of merit order dispatch using adaptive learning algorithm,” in *Power, Signals, Controls and Computation (EPSCICON), 2012 International Conference on*, Jan 2012, pp. 1–7.
- [41] D. Chen and M. York, “Neural network based very short term load prediction,” in *Transmission and Distribution Conference and Exposition, IEEE/PES*, April 2008, pp. 1–9.
- [42] X. Mian, W. Shu-ling, G. Zhi-Hong, and L. Zhen-tao, “Power load forecast model of genetic neural network based on Tabu search,” in *Industrial Electronics and Applications, 2007. ICIEA 2007. 2nd IEEE Conference on*, May 2007, pp. 1903–1906.
- [43] D. Xin-hui, T. Feng, and T. Shao-qiong, “Study of power system short-term load forecast based on artificial neural network and genetic algorithm,” in *Computational Aspects of Social Networks (CASON), 2010 International Conference on*, Sept 2010, pp. 725–728.
- [44] F. Han, Q. Zhang, X. Zhang, and T. Li, “Application of fuzzy neural network to power system short-term load forecast,” in *Computer Design and Applications (ICCD), 2010 International Conference on*, vol. 2, June 2010, pp. V2–496–V2–499.

- [45] Y. Hu, H. Ji, and X. Song, "To forecast short-term load in electric power system based on FNN," in *Fuzzy Systems and Knowledge Discovery, 2009. FSKD '09. Sixth International Conference on*, vol. 6, Aug 2009, pp. 474–478.
- [46] H. Al-Hamadi and S. Soliman, "Fuzzy short-term electric load forecasting using kalman filter," *Generation, Transmission and Distribution, IEE Proceedings*-, vol. 153, no. 2, pp. 217–227, March 2006.
- [47] A. Jain, E. Srinivas, and R. Rauta, "Short term load forecasting using fuzzy adaptive inference and similarity," in *Nature Biologically Inspired Computing, 2009. NaBIC 2009. World Congress on*, Dec 2009, pp. 1743–1748.
- [48] Z. Osman, M. Awad, and T. Mahmoud, "Neural network based approach for short-term load forecasting," in *Power Systems Conference and Exposition, 2009. PSCE '09. IEEE/PES*, March 2009, pp. 1–8.
- [49] F. Lai, F. Magoules, and F. Lherminier, "Vapniks learning theory applied to energy consumption forecasts in residential buildings," *International Journal of Computer Mathematics*, vol. 85, no. 10, p. 15631588, 2008.
- [50] X. Li, H. Chen, and S. Gao, "Electric power system load forecast model based on state space time-varying parameter theory," in *Power System Technology (POWERCON), 2010 International Conference on*, Oct 2010, pp. 1–4.
- [51] R. Kyoho, T. Goya, W. Mengyan, T. Senjyu, A. Yona, T. Funabashi, and C.-H. Kim, "Thermal units commitment with demand response to optimize battery storage capacity," in *Power Electronics and Drive Systems (PEDS), 2013 IEEE 10th International Conference on*, April 2013, pp. 1207–1212.
- [52] S. Pourmousavi, S. Patrick, and M. Nehrir, "Real-time demand response through aggregate electric water heaters for load shifting and balancing wind

- generation,” *Smart Grid, IEEE Transactions on*, vol. 5, no. 2, pp. 769–778, March 2014.
- [53] H. Saele and O. S. Grande, “Demand response from household customers: Experiences from a pilot study in norway,” *IEEE Transactions on Smart Grid*, vol. 2, no. 1, pp. 102–109, March 2011.
- [54] Q. Hu and F. Li, “Hardware design of smart home energy management system with dynamic price response,” *IEEE Transactions on Smart Grid*, vol. 4, no. 4, pp. 1878–1887, Dec 2013.
- [55] K.-Y. Huang and Y.-C. Huang, “Integrating direct load control with interruptible load management to provide instantaneous reserves for ancillary services,” *IEEE Transactions on Power Systems*, vol. 19, no. 3, pp. 1626–1634, Aug 2004.
- [56] N. Saker, M. Petit, and J. Coullon, “Demand side management of electrical water heaters and evaluation of the cold load pick-up characteristics (clpu),” in *PowerTech, 2011 IEEE Trondheim*, June 2011, pp. 1–8.
- [57] S. Ayoub, “Electric water heaters control strategy for providing regulation services and load leveling in electric power systems,” in *Electrical Power Energy Conference (EPEC), 2013 IEEE*, Aug 2013, pp. 1–6.
- [58] A. Mufaris, S. Kawachi, and J. Baba, “Voltage control using coordinated control of heat pump water heaters with large penetration of photovoltaic systems,” in *Electric Power and Energy Conversion Systems (EPECS), 2013 3rd International Conference on*, Oct 2013, pp. 1–6.
- [59] L. Yao, W.-C. Chang, and R.-L. Yen, “An iterative deepening genetic algorithm for scheduling of direct load control,” *IEEE Transactions on Power Systems*, vol. 20, no. 3, pp. 1414–1421, Aug 2005.

- [60] Y. Atwa, E. El-Saadany, and M. Salama, “DSM approach for water heater control strategy utilizing elman neural network,” in *Proceedings of IEEE Canada Electrical Power Conference*, 2007, pp. 1–5.
- [61] M. Nehrir and B. LaMeres, “Multiple-block fuzzy logic-based electric water heater demand-side management strategy for leveling distribution feeder demand profile,” *Electric Power System Research*, vol. 56, no. 3, pp. 225–230, 2000.
- [62] H. Salehfar, P. J. Noll, B. J. LaMeres, M. H. Nehrir, and V. Gerez, “Fuzzy logic-based direct load control of residential electric water heaters and air conditioners recognizing customer preferences in a deregulated environment,” in *Power Engineering Society Summer Meeting, 1999. IEEE*, vol. 2, 1999, pp. 1055–1060 vol.2.
- [63] Z. Wu, S. Zhou, J. Li, and X.-P. Zhang, “Real-time scheduling of residential appliances via conditional risk-at-value,” *Smart Grid, IEEE Transactions on*, vol. 5, no. 3, pp. 1282–1291, May 2014.
- [64] K. Al-jabery, D. Wunsch, J. Xiong, and Y. Shi, “A novel grid load management technique using electric water heaters and Q-learning,” in *Smart Grid Communications (SmartGridComm), 2014 IEEE International Conference on*, Nov 2014, pp. 776–781.
- [65] F. Ruelens, B. 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 *Power Systems Computation Conference (PSCC), 2014*, Aug 2014, pp. 1–7.

- [66] D. Guo, W. Zhang, G. Yan, Z. Lin, and M. Fu, “Decentralized control of aggregated loads for demand response,” in *American Control Conference (ACC)*, 2013, June 2013, pp. 6601–6606.
- [67] F. Sossan, A. Kosek, S. Martinenas, M. Marinelli, and H. Bindner, “Scheduling of domestic water heater power demand for maximizing PV self-consumption using model predictive control,” in *Innovative Smart Grid Technologies Europe (ISGT EUROPE)*, 2013 4th IEEE/PES, Oct 2013, pp. 1–5.
- [68] K. Lajoie, D. Halamay, and T. Brekken, “Residential water heaters as a grid-scale energy storage solution using model predictive control,” in *Technologies for Sustainability (SusTech)*, 2013 1st IEEE Conference on, Aug 2013, pp. 62–69.
- [69] PSA. (2009) Powershift atlantic an energy research project. Accessed: 2014-10-30. [Online]. Available: <http://www.powershiftatlantic.com/index.html>
- [70] M. Shaad, A. Momeni, C. P. Diduch, M. Kaye, and L. Chang, “Parameter identification of thermal models for domestic electric water heaters in a direct load control program,” in *Electrical Computer Engineering (CCECE)*, 2012 25th IEEE Canadian Conference on, April 2012, pp. 1–5.
- [71] M. Verhaegen and V. Verdult, *Filtering and System Identification*. New York: Cambridge University Press, 2007.
- [72] F. Liu, M. Faraz, and M. M’Saad, “Unknown inputs estimation for a class of nonlinear systems,” *Int. Journal of Modeling, Identification and Control*, vol. 1, no. 1, pp. 33–48, 2007.
- [73] D. Marquardt, “An algorithm for least-squares estimation of nonlinear parameters,” *SIAM Journal on Applied Mathematics*, vol. 11, no. 2, pp. 431 – 441, 1963.

- [74] N. Mohamad, F. Zaini, A. Johari, I. Yassin, and A. Zabidi, “Comparison between Levenberg-Marquardt and scaled conjugate gradient training algorithms for breast cancer diagnosis using mlp,” in *Proc. of 6th International Colloquium on Signal Processing and Its Applications (CSPA)*, Florence, Italy, May 2010, pp. 1–7.
- [75] R. E. Kalman, “A new approach to linear filtering and prediction problems,” *Journal of Basic Engineering*, vol. 1, no. 1, pp. 35–45, 1960.
- [76] P. Andersen, B. Poulsen, M. Decker, C. Traeholt, and J. Ostergaard, “Evaluation of a generic virtual power plant framework using service oriented architecture,” in *Power and Energy Conference, 2008. PEGCon 2008. IEEE 2nd International*, Dec 2008, pp. 1212–1217.
- [77] I. Kuzle, M. Zdrilic, and H. Pandzic, “Virtual power plant dispatch optimization using linear programming,” in *Environment and Electrical Engineering (EEEIC), 2011 10th International Conference on*, May 2011, pp. 1–4.
- [78] B. Abegaz and S. Mahajan, “Optimal real-time integration control of a virtual power plant,” in *North American Power Symposium (NAPS), 2014*, Sept 2014, pp. 1–6.
- [79] L. Nikonowicz and J. Milewski, “Virtual power plants general review: Structure, application and optimization,” *Journal of Power Technology*, vol. 92, no. 3, pp. 135–149, 2012.
- [80] P. Asmus, “Microgrids, virtual power plants and our distributed energy future,” *The Electricity Journal*, vol. 23, no. 10, pp. 72–82, 2010.
- [81] P. Hawrylak, W. Nichols, X. Li, T. Johannes, R. DeCook, K. Mongold, C. Cornell, L. Flint, and A. Singh, “Hydrosense: A self-powered wireless device for

monitoring water usage in hotel showers,” in *Global Humanitarian Technology Conference (GHTC), 2015 IEEE*, Oct 2015, pp. 314–320.

Appendices

A Source Codes

A.1 Calculating the physical parameters

```
private void threadCalcParam()
{
    ListView.SelectedListViewItemCollection sel =
        lstExperiments.SelectedItems;
    string expName = sel[0].SubItems[1].Text;
    //calculate all t_on and t_off
    if (chkManual.Checked == false)
        CalcOnOffTimes(expName);
    else
    {
        string[] time = txtMinOn.Text.Split(':');
        if (time.Length != 2) { MessageBox.Show("please_enter_mm:ss_for_min(t_on)"); return; }
        int min = 0;
        if (int.TryParse(time[0], out min) == false) { MessageBox
            .Show("please_enter_an_integer_for_min(t_on)_minutes")
            ;
            return; }
        if (int.TryParse(time[1], out min) == false) { MessageBox
            .Show("please_enter_an_integer_for_min(t_on)_seconds")
            ;
            return; }
```

```

    TimeSpan delta_t = new TimeSpan(0, int.Parse(time[0]),
        int.Parse(time[1]));
    lstParam.Items[0].SubItems[1].Text = delta_t.ToString();

    time = txtMaxOff.Text.Split(':');
    if (time.Length != 3) { MessageBox.Show("please_enter_hh:
        mm:ss_for_max(t_off)"); return; }
    if (int.TryParse(time[0], out min) == false) { MessageBox
        .Show("please_enter_an_integer_for_max(t_off)_hours");
    return; }
    if (int.TryParse(time[1], out min) == false) { MessageBox
        .Show("please_enter_an_integer_for_max(t_off)
        minutes"); return; }
    if (int.TryParse(time[2], out min) == false) { MessageBox
        .Show("please_enter_an_integer_for_max(t_off)
        seconds"); return; }

    delta_t = new TimeSpan(int.Parse(time[0]), int.Parse(time
        [1]), int.Parse(time[2]));
    lstParam.Items[1].SubItems[1].Text = delta_t.ToString();
}

//Now calc C & G based on t_on(min) and t_off(max)
CalcCG();

btnCalcParam.Enabled = true;
stsProgress.Value = 0;
stsProgress.Visible = false;
}

```

```

private void CalcOnOffTimes(string expName)
{
    //Load power for this experiment(expName)
    SqlConnection objSqlCnt = new SqlConnection(stConnection);
    SqlCommand objSqlCmd = new SqlCommand();
    SqlDataAdapter objSqlDataAdp = new SqlDataAdapter();
    DataView objDataView = new DataView();

    objSqlDataAdp.SelectCommand = new SqlCommand();
    objSqlDataAdp.SelectCommand.CommandText = "select date,
        value from Measurements where experiment=@exp and
        parameter=@param order by date";
    objSqlDataAdp.SelectCommand.Parameters.AddWithValue("exp",
        expName);
    objSqlDataAdp.SelectCommand.Parameters.AddWithValue("param",
        "power");
    objSqlDataAdp.SelectCommand.CommandType = CommandType.Text;
    objSqlDataAdp.SelectCommand.Connection = objSqlCnt;
    DataSet list = new DataSet();
    try { objSqlDataAdp.Fill(list, "Power"); }
    catch (Exception ee) { MessageBox.Show(ee.Message); return;
        }
    objDataView = new DataView(list.Tables[0]);

    //Initialize arraylists
    ArrayList Ons = new ArrayList();
    ArrayList Offs = new ArrayList();

    //Now get the times when the power rate changes

```

```

double power = double.Parse(objDataView[0]["value"].
    ToString());
DateTime start_time = DateTime.Parse(objDataView[0]["date"]
    ].ToString());
DateTime first_time = start_time;
DateTime last_time = DateTime.Parse(objDataView[objDataView
    .Count - 1]["date"].ToString());

while (objDataView.Count > 0)
{
    objDataView.RowFilter = "value_<>" + power + "_and_date>"
        + "'" + start_time + "'";
    if (objDataView.Count == 0) break;
    stsProgress.Value = (int)((((start_time - first_time).
        Days * 24 + (start_time - first_time).Hours) * 60 +
        (start_time - first_time).Minutes) * 100.0 / (((last_time
        - first_time).Days * 24 + (last_time -
        first_time).Hours) * 60 + (last_time - first_time).
        Minutes));

    DateTime end_time = DateTime.Parse(objDataView[0]["date"]
        ].ToString());

    if (power == 0) //add an off-time
        Offs.Add(end_time - start_time);
    else //add an On-time
        Ons.Add(end_time - start_time);
    start_time = end_time;
    power = double.Parse(objDataView[0]["value"].ToString());
}

```

```

}

//load min_on and max_off timespans
TimeSpan min_on = new TimeSpan(10, 0, 0);
TimeSpan max_off = new TimeSpan();

for (int i = 0; i < Ons.Count; i++) if ((TimeSpan)Ons[i] <
    min_on) min_on = (TimeSpan)Ons[i];
for (int i = 0; i < Offs.Count; i++) if ((TimeSpan)Offs[i]
    > max_off) max_off = (TimeSpan)Offs[i];

lstParam.Items[0].SubItems[1].Text = min_on.ToString();
lstParam.Items[1].SubItems[1].Text = max_off.ToString();
}

private void CalcCG()
{
    //get T_hi and T_lo and T_out and Q
    double T_hi = 0;
    double T_lo = 0;
    double T_out = 0;
    double Q = 0;

    try { T_hi = double.Parse(txtThi.Text); }
    catch { MessageBox.Show("Please_input_a_valid_number_for_
        T_hi"); return; }
    try { T_lo = double.Parse(txtTlo.Text); }

```

```

catch { MessageBox.Show("Please_input_a_valid_number_for_
    T_lo"); return; }
try { T_out = double.Parse(txtTout.Text); }
catch { MessageBox.Show("Please_input_a_valid_number_for_
    T_out"); return; }
try { Q = double.Parse(txtQ.Text); }
catch { MessageBox.Show("Please_input_a_valid_number_for_Q"
    ); return; }

try
{
    TimeSpan t_on_min = TimeSpan.Parse(lstParam.Items[0].
        SubItems[1].Text);
    TimeSpan t_off_max = TimeSpan.Parse(lstParam.Items[1].
        SubItems[1].Text);

    double t_on = (t_on_min.Hours * 60 + t_on_min.Minutes) *
        60.0 + t_on_min.Seconds;
    double t_off = (t_off_max.Hours * 60 + t_off_max.Minutes)
        * 60.0 + t_off_max.Seconds;
    double alpha = Math.Pow((T_hi - T_out) / (T_lo - T_out),
        t_on / t_off);

    double G = (Q * (alpha - 1)) / (alpha * (T_hi - T_out) -
        (T_lo - T_out));
    double C = (G * t_off) / Math.Log((T_hi - T_out) * 1.0 /
        (T_lo - T_out));
    lstParam.Items[2].SubItems[1].Text = C.ToString();
    lstParam.Items[3].SubItems[1].Text = G.ToString();
}

```

```

    }
    catch (Exception ee) { MessageBox.Show(ee.Message); }
}

```

A.2 Estimating W_D in Scenario I

```

private void ThreadStateObserver()
{
    double c_p = 4181.8;
    double ro = 998.2;

    ListView.SelectedListViewItemCollection sel = lstExperiments.
        SelectedItems;

    //Load the measured Temperature and Power
    DataView Temperature = LoadExpData(sel[0].SubItems[1].Text, "
        Temperature");
    DataView Power = LoadExpData(sel[0].SubItems[1].Text, "Power");

    double[] T_hat = new double[Temperature.Count];
    double[] Wd_hat = new double[Temperature.Count];
    DateTime[] time = new DateTime[Temperature.Count];
    time[0] = DateTime.Parse(Temperature[0]["date"].ToString());
    T_hat[0] = double.Parse(Temperature[0]["value"].ToString());
    Wd_hat[0] = 0;

    double k = .01;    //observer gain
    try { k = double.Parse(txtObsGain.Text); }
    catch { }
}

```

```

int power_counter = 0;
for (int i = 1; i < Temperature.Count; i++)
{
    stsProgress.Value = (int)(i * 100 / Temperature.Count);
    double T = double.Parse(Temperature[i]["value"].ToString());
    time[i] = DateTime.Parse(Temperature[i]["date"].ToString());
    double delta_t = (time[i] - time[i - 1]).Minutes * 60 + (time
        [i] - time[i - 1]).Seconds;
    //get the power state at this time
    //Power.RowFilter = "date>" + time[i] + " ";
    while (DateTime.Parse(Power[power_counter++]["date"].ToString
        ()) < time[i]) ;
    double Q = double.Parse(Power[power_counter]["value"].
        ToString()) * 1000;    //convert it to watt

    T_hat[i] = T_hat[i - 1] + (-G * T_hat[i - 1] / C + Q / C + (
        T_in - T_hat[i - 1]) * ro * c_p * Wd_hat[i - 1] / C + G *
        T_out / C - 2 * k * (T_hat[i - 1] - T)) * delta_t;
    Wd_hat[i] = Wd_hat[i - 1] + (-k * k * C * (T_hat[i - 1] - T)
        / (ro * c_p * (T_in - T_hat[i - 1]))) * delta_t; //unit:m
        ^3/s
}
stsProgress.Value = 0;
//Now store T_hat and WD_hat in DB
StoreObserver(T_hat, Wd_hat, time, sel[0].SubItems[1].Text);

btnObserver.Enabled = true;
stsProgress.Visible = false;
stsProgress.Value = 0;

```

```
}
```

A.3 Estimating W_D in Scenario II

```
private void ThreadAveragingMethod()
{
    //This method calculate average WD during each on/off intervals
    , and stores it in database

    //First clear database from previous calculations
    SqlConnection objSqlCnt = new SqlConnection(stConnection);
    SqlCommand objSqlCmd = new SqlCommand();

    for (int i = 0; i < lstExperiments.SelectedItems.Count; i++)
    {
        //First clear previous values
        objSqlCmd = new SqlCommand("delete Average where experiment=
            @exp");
        objSqlCmd.Connection = objSqlCnt;
        objSqlCmd.Parameters.AddWithValue("@exp", lstExperiments.
            SelectedItems[i].SubItems[1].Text);
        try { objSqlCnt.Open(); objSqlCmd.ExecuteNonQuery();
            objSqlCnt.Close(); }
        catch (Exception ee) { MessageBox.Show("Clearing previous
            Average for " + lstExperiments.SelectedItems[i].SubItems
            [1].Text + "\n" + ee.Message); }

        ListView.SelectedListViewItemCollection sel = lstExperiments.
            SelectedItems;
```

```

DataView Power = LoadExpData(sel[i].SubItems[1].Text, "Power"
    );

//Now get the times when the power rate changes
double Q = double.Parse(Power[0]["value"].ToString());
DateTime start_time = DateTime.Parse(Power[0]["date"].
    ToString());
DateTime first_time = start_time;
DateTime last_time = DateTime.Parse(Power[Power.Count - 1]["
    date"].ToString());

while (Power.Count > 0)
{
    Power.RowFilter = "value_<>" + Q + "_and_date>" + "'" +
        start_time + "'";
    if (Power.Count == 0) break;
    stsProgress.Value = (int)((((start_time - first_time).Days
        * 24 + (start_time - first_time).Hours) * 60 + (
        start_time - first_time).Minutes) * 100.0 / (((last_time
        - first_time).Days * 24 + (last_time - first_time).
        Hours) * 60 + (last_time - first_time).Minutes));

    DateTime end_time = DateTime.Parse(Power[0]["date"].
        ToString());
    double delta_t = (((end_time - start_time).Days * 24 + (
        end_time - start_time).Hours) * 60 + (end_time -
        start_time).Minutes) * 60 + (end_time - start_time).
        Seconds; //in seconds

```

```

    if (Q == 0) //Calculate WD for off interval
        StoreAveWdT(CalcWD(0, delta_t), start_time, end_time, 0,
            T_hi);
    else //add an On_time
        StoreAveWdT(CalcWD(nominal_Q, delta_t), start_time,
            end_time, nominal_Q, T_lo);

    //move the start_time to the end of this interval
    start_time = end_time;
    //store DEWH status for next iteration
    Q = double.Parse(Power[0]["value"].ToString());
}
}

//hide progress bar
stsProgress.Value = 0;
stsProgress.Visible = false;
btnAverage.Enabled = true;
}

private double CalcWD(double Q, double delta_t)
{
    //This function calculates WD assuming that the load of DEWH is
    Q (in watt) for delta_t seconds

    //if Q==0, means that water heater was off, so the temperatuer
    starts from T_hi and ends at T_lo
    //if Q>0, means that DEWH was on, so the temperatuer changes
    from T_lo to T_hi

```

```

double T_start = (Q == 0) ? T_hi : T_lo;
double T_end = (Q == 0) ? T_lo : T_hi;

double Wd1 = -.1 / 60000.0;
double Wd2 = 60 / 60000.0;
int counter = 0;
while (++counter < 100)
{
    double R1 = 1 / (ro * c_p * Wd1 + G);
    double R2 = 1 / (ro * c_p * Wd2 + G);

    double y1 = Temperature_function(R1, Q, T_start, delta_t, G,
        C, T_out, T_in) - T_end;
    double y2 = Temperature_function(R2, Q, T_start, delta_t, G,
        C, T_out, T_in) - T_end;
    if (y1 * y2 > 0) return 0;
    if (Math.Abs(y1) < .1) return Wd1;
    if (Math.Abs(y2) < .1) return Wd2;
    double Wd3 = (Wd1 + Wd2) / 2;
    double R3 = 1 / (ro * c_p * Wd3 + G);
    double y3 = Temperature_function(R3, Q, T_start, delta_t, G,
        C, T_out, T_in) - T_end;
    if (y1 * y3 < 0) Wd2 = Wd3;
    else if (y2 * y3 < 0) Wd1 = Wd3;
    else MessageBox.Show("Could not find root");
}
return 0;

```

```
}
```

```
private double Temperature_function(double R, double Q, double
    T_start, double delta_t, double G, double C, double T_out,
    double T_in)
{
    //This function returns T equation for different values of R
    //Q is in watt
    //delta_t is in second

    double result = T_start * Math.Exp(-delta_t / (R * C)) + (G * R
        * T_out + (1 - G * R) * T_in + Q * R) * (1 - Math.Exp(-
        delta_t / (R * C)));

    return result;
}
```

A.4 Estimating W_D on the pilot

```
private void thread_clacWD_ave()
{
    DateTime start_time = new DateTime(StartCalendar.Value.Year
        , StartCalendar.Value.Month, StartCalendar.Value.Day);
    DateTime end_time = new DateTime(EndCalendar.Value.Year,
        EndCalendar.Value.Month, EndCalendar.Value.Day);

    //for (DateTime start = new DateTime(2012, 07, 1); start <
        new DateTime(2012, 07, 14); start = start.AddDays(7))
```

```

DateTime start = start_time;
{
    DateTime end = end_time; // start.AddDays(7); // new
    DateTime(2012, 09, 1);

    LoadHouseList();

    //for (int houseID = 1; houseID <= iMaxHouse; houseID++)
    int houseID = int.Parse(cmbHouse.SelectedItem.ToString())
        ;
    {
        stsProgress.Value = (houseID * 100 / iMaxHouse);
        LoadData(houseID, start, end);
        if (objSqlDataView.Count == 0) return; //continue;
        double P0 = double.Parse(objSqlDateViewHouselist[
            houseID][ "P0" ].ToString()) * 1000;

        double Q = double.Parse(objSqlDataView[0][ "WPC" ].
            ToString());
        DateTime dt = DateTime.Parse(objSqlDataView[0][ "
            DateTime" ].ToString());
        double Q_bar = (Q < .1) ? 0.25 : 0;
        int on_time = 0;
        int off_time = 0;

        string strStart = "'" + dt.Year + "-" + dt.Month + "-"
            + dt.Day + "_" + dt.Hour + ":" + dt.Minute + "'";
        objSqlDataView.RowFilter = "DateTime>_" + strStart + "
            _and_WPC=" + Q_bar;
    }
}

```

```

while (objSqlDataView.Count > 0)
{
    DateTime new_dt = DateTime.Parse(objSqlDataView[0][ "
        DateTime"].ToString());
    lblStatus.Text = "Analyzing␣until␣" + new_dt;
    TimeSpan delta_t = new_dt - dt;
    double Wd = 0;
    if (Q > 0.1)
    {
        on_time = ((delta_t.Days * 24 + delta_t.Hours) * 60
            + delta_t.Minutes) * 60;
        Wd = FindWd_ave(houseID, T_lo, T_hi, P0, on_time);
    }
    else
    {
        off_time = Math.Min(500 * 60, ((delta_t.Days * 24 +
            delta_t.Hours) * 60 + delta_t.Minutes) * 60);

        Wd = FindWd_ave(houseID, T_hi, T_lo, P0, off_time);
    }
    //Now update datarows
    string strEnd = "'" + new_dt.Year + "-" + new_dt.
        Month + "-" + new_dt.Day + "␣" + new_dt.Hour + ":"
        + new_dt.Minute + "'";
    objSqlDataView.RowFilter = "DateTime␣>=␣" + strStart
        + "␣and␣DateTime<" + strEnd;
    for (int counter = 0; counter < objSqlDataView.Count;
        counter++) objSqlDataView[counter]["WD"] = Wd *
        60000;
}

```

```

        dt = new_dt;
        Q = double.Parse(objSqlDataView[0]["WPC"].ToString());
        ;
        Q_bar = (Q < .1) ? 0.25 : 0;
        strStart = "'" + dt.Year + "-" + dt.Month + "-" + dt.
            Day + "_" + dt.Hour + ":" + dt.Minute + "'";
        objSqlDataView.RowFilter = "DateTime>_" + strStart +
            "_and_WPC=" + Q_bar;
    }

    //Now store household list
    SqlCommandBuilder objCommandBuilder = new
        SqlCommandBuilder(objSqlDataAdp);
    objSqlDataAdp.FillSchema(selectedData, SchemaType.
        Mapped);
    objSqlDataAdp.Update(selectedData, "Meter" + houseID);
    }
}

stsProgress.Value = 0;
stsProgress.Visible = false;
}

private double FindWd_ave(int houseID, double T_start, double
    T_end, double Q, double delta_t)
{
    //Wd is m3/s
    double Wd1 = -20 / 60000.0;
    double Wd2 = 20 / 60000.0;

```

```

double G = double.Parse(objSqlDateViewHouselist[houseID]["G
    "].ToString());
double C = double.Parse(objSqlDateViewHouselist[houseID]["C
    "].ToString());

int counter = 0;
while (++counter < 100)
{
    double R1 = 1 / (ro * c_p * Wd1 + G);
    double R2 = 1 / (ro * c_p * Wd2 + G);

    double y1 = function(R1, Q, T_start, T_end, delta_t, C, G
        );
    double y2 = function(R2, Q, T_start, T_end, delta_t, C, G
        );
    if (y1 * y2 > 0) return 0;
    if (Math.Abs(y1) < .1) return Wd1;
    if (Math.Abs(y2) < .1) return Wd2;
    double Wd3 = (Wd1 + Wd2) / 2;
    double R3 = 1 / (ro * c_p * Wd3 + G);
    double y3 = function(R3, Q, T_start, T_end, delta_t, C, G
        );
    if (y1 * y3 < 0) Wd2 = Wd3;
    else if (y2 * y3 < 0) Wd1 = Wd3;
    else MessageBox.Show("Coudl not find root");
}
return 0;
}

```

```

private double function(double R, double Q, double T_start,
    double T_end, double delta_t, double C, double G)
{
    //This function returns T equation for different values of
    R
    //Q is in watt
    //delta_t is in second

    double result = T_start * Math.Exp(-delta_t / (R * C)) + (G
        * R * T_out + (1 - G * R) * T_in + Q * R) * (1 - Math.
        Exp(-delta_t / (R * C))) - T_end;

    return result;
}

```

A.5 Statistic-Based Load Forecast

```

private void ThreadTrain()
{
    ons = new ArrayList();
    offs = new ArrayList();
    DateTime start = new DateTime(startCalendar.Value.Year,
        startCalendar.Value.Month, startCalendar.Value.Day);
    DateTime end = new DateTime(endCalendar.Value.Year,
        endCalendar.Value.Month, endCalendar.Value.Day);
    if (end <= start) end = start.AddDays(1);
    int total_minutes = ((end - start).Days * 24 + (end - start).
        Hours) * 60 + (end - start).Minutes;
}

```

```

int houseID = -1;
try { houseID = int.Parse(cmbHouse.SelectedItem.ToString());
    }
catch { return; }

SqlCommand objSqlCmd = new SqlCommand();
SqlDataAdapter objSqlDataAdp = new SqlDataAdapter();

//Load telemetry data for houseID since start to end
objSqlDataAdp.SelectCommand = new SqlCommand();
objSqlDataAdp.SelectCommand.CommandText = "select rxTime,
    cum_DEWH_from_house" + houseID + " where rxTime>=@start
    and rxTime<@end order by rxTime";
objSqlDataAdp.SelectCommand.Parameters.AddWithValue("@start",
    start);
objSqlDataAdp.SelectCommand.Parameters.AddWithValue("@end",
    end);
objSqlDataAdp.SelectCommand.CommandType = CommandType.Text;
objSqlDataAdp.SelectCommand.Connection = objSqlCnt;
DataSet history = new DataSet();
try { objSqlDataAdp.Fill(history, "House" + houseID); }
catch (Exception ee) { } // { MessageBox.Show("Error loading
    histofy...\n" + ee.Message); }
if (history.Tables.Count == 0) return;

DataView dvHistory = new DataView(history.Tables[0]);
int total_ons = 0;
int total_offs = 0;

```

```

int previous_delta = 0;

while (dvHistory.Count > 0)
{
    int state = double.Parse(dvHistory[0]["cum_dewh"].ToString
        ()) > 0 ? 1 : 0;
    DateTime elapsed = DateTime.Parse(dvHistory[0]["rxTime"].
        ToString());

    int elapsed_minute = ((elapsed - start).Days * 24 + (
        elapsed - start).Hours) * 60 + (elapsed - start).Minutes
        ;
    stsProgress.Value = (int)(elapsed_minute * 100 /
        total_minutes);

    string filter = "";
    if (state == 0)
        filter = "cum_DEWH>0□and□rxTime>' " + elapsed + "'";
    else
        filter = "cum_DEWH=0□and□rxTime>' " + elapsed + "'";
    dvHistory.RowFilter = filter;
    if (dvHistory.Count > 0)
    {
        DateTime now = DateTime.Parse(dvHistory[0]["rxTime"].
            ToString());
        int delta_t = ((now - elapsed).Days * 24 + (now - elapsed
            ).Hours) * 60 + (now - elapsed).Minutes;
        if (state == 0)
        {

```

```

        offs = InsertInterval(offs , delta_t);
        total_offs += 1;
        ons = UpdateNextInterval(ons, delta_t , previous_delta);
    }
    else
    {
        ons = InsertInterval(ons, delta_t);
        total_ons += 1;
        offs = UpdateNextInterval(offs , delta_t , previous_delta
            );
    }
    previous_delta = delta_t;
}
}

//calculate the probabilities of each duration
for (int i = 0; i < ons.Count; i++) { duration d = (duration)
    ons[i]; d.probability = d.count * 100.0 / total_ons; ons.
    RemoveAt(i); ons.Insert(i, d); }
for (int i = 0; i < offs.Count; i++) { duration d = (duration)
    offs[i]; d.probability = d.count * 100.0 / total_offs;
    offs.RemoveAt(i); offs.Insert(i, d); }

DrawOnOffPDF();
stsProgress.Value = 0;
StorePDF();
stsProgress.Visible = false;
btnTrain.Text = "Training";
}

```

```

private ArrayList InsertInterval(ArrayList a, int interval)
{
    ArrayListCompare c = new ArrayListCompare();
    duration d = new duration();
    for (int i = 0; i < a.Count; i++)
    {
        d = (duration)a[i];
        if (d.minutes == interval)
        { d.count += 1; a.RemoveAt(i); a.Add(d); a.Sort(c); return
            a; }
    }
    d.minutes = interval;
    d.count = 1;
    d.next_interval = new ArrayList();
    a.Add(d);
    a.Sort(c);
    return a;
}

```

```

private ArrayList UpdateNextInterval(ArrayList a, int
    next_interval, int current_interval)
{
    for (int i = 0; i < a.Count; i++)
    {
        duration d = (duration)a[i];
        if (d.minutes == current_interval)
        {
            if (d.next_interval == null || d.next_interval.Count ==
                0) d.next_interval = new ArrayList();

```



```

btnSingleForecast.Text = "Stop";
int houseID = -1;
try { houseID = int.Parse(cmbHouse.SelectedItem.ToString());
    }
catch
{
    btnSingleForecast.Text = "Single_FLC";
    stsProgress.Value = 0;
    stsProgress.Visible = false; return;
}

allHouses.RowFilter = "HouseID=" + houseID;
if (allHouses.Count == 0 || allHouses[0]["Nominal"].ToString
    ().Length == 0 || double.Parse(allHouses[0]["Nominal"].
    ToString()) < 2)    //if house ID is invalid or its not a
    DEWH
{
    btnSingleForecast.Text = "Single_FLC";
    stsProgress.Value = 0;
    stsProgress.Visible = false;
    allHouses.RowFilter = "";
    return;
}
allHouses.RowFilter = "";
DateTime start = new DateTime(startCalendar.Value.Year,
    startCalendar.Value.Month, startCalendar.Value.Day);
DateTime end = new DateTime(endCalendar.Value.Year,
    endCalendar.Value.Month, endCalendar.Value.Day);
if (end <= start) end = start.AddDays(1);

```

```

int total_minutes = ((end - start).Days * 24 + (end - start).
    Hours) * 60 + (end - start).Minutes;

//creat FLC every 5 minutes from start to end
for (DateTime dt = start; dt < end; dt = dt.AddMinutes(5))
{
    Forecast(houseID, dt);

    //update the progress bar
    int elapsed_minute = ((dt - start).Days * 24 + (dt - start)
        .Hours) * 60 + (dt - start).Minutes;
    stsProgress.Value = (int)(elapsed_minute * 100 /
        total_minutes);

} //end of date loop

btnSingleForecast.Text = "Single FLC";
stsProgress.Value = 0;
stsProgress.Visible = false;
}

private void Forecast(int houseID, DateTime dt)
{
    //houseID changes from 1..iMaxHouse

    //load the current status and the duration it was in this
    state and the duration of the previous status

```

```

double Q = GetQ(houseID, dt);
int state = (Q > 0) ? 1 : 0;
int state_bar = (Q > 0) ? 0 : 1;

//Get the time when the DEWH changed state
DateTime last = GetLastStateChange(houseID, Q, dt);
if (last > DateTime.Now) return;
int alpha_0 = ((dt - last).Days * 24 + (dt - last).Hours) *
    60 + (dt - last).Minutes;

//now determine previous s_bar interval
DateTime pre_last = GetLastStateChange(houseID, state_bar,
    last);
int gamma = ((last - pre_last).Days * 24 + (last - pre_last).
    Hours) * 60 + (last - pre_last).Minutes;

//Now calculate the expected value of staying in this state
int t = CalcEstimatedTime(houseID, alpha_0, gamma, Q);
//int t = CalcEstimatedTime(houseID, alpha_0, Q);

//Create the forecast string
GenerateStoreForecastString(houseID, dt, (Q > 0) ? 1 : 0, t);
}

private int CalcEstimatedTime(int houseID, int alpha_0, double
    Q)
{
    if (all_ons[houseID - 1] == null) return 0;
    double t_hat = 0;
    int n0 = 0;

```

```

if (Q > 0)  //DEWH is on for alpha_0 minutes
{
    //get the number of incidents at which t_s >= alpha_0
    for (int i = 0; i < all_ons[houseID - 1].Count; i++) {
        duration d = (duration)all_ons[houseID - 1][i]; if (d.
            minutes >= alpha_0) n0 += d.count; }

    for (int i = 0; i < all_ons[houseID - 1].Count; i++)
    { duration d = (duration)all_ons[houseID - 1][i]; if (d.
        minutes >= alpha_0) t_hat += d.minutes * d.count * 1.0 /
            n0; }
}

else      // DEWH is off for alpha_0 minutes
{
    //get the number of incidents at which t_s >= alpha_0
    for (int i = 0; i < all_off[houseID - 1].Count; i++) {
        duration d = (duration)all_off[houseID - 1][i]; if (d.
            minutes >= alpha_0) n0 += d.count; }

    for (int i = 0; i < all_off[houseID - 1].Count; i++)
    { duration d = (duration)all_off[houseID - 1][i]; if (d.
        minutes >= alpha_0) t_hat += d.minutes * d.count * 1.0 /
            n0; }
}

//remaining time at this state
int est = (int)t_hat - alpha_0;
return est;
}

```

A.6 ANN-Based Aggregated Load Forecast

```
function
forecast_water_heater_power=Neural_Network_forecast_algorithm(
    future_forecast_horizon ,start_forecast ,Input ,Target ,
    actual_Input ,household_Id_final ,...
    store_FLC_in_database)

%% p is the input for neuron network; t is the target for the
    neuron network %%
%% NE: number of epochs = 40 (number of epochs is imposed only
    when the training stage with validate data doesn't converge)
%% NHL: number of hidden layers =2
%% NN: number of neurons for both first hidden layer and second
    hidden layer = 100
%% NI: number of inputs = past forecast horizon
%% NO: number of outputs= future forecast horizon
%% Transfer function for the first hidden layer: tangent function
    tansig
%% Transfer function for the second hidden layer: tangent
    function    tansig
%% Transfer function for the output layer: linear function
    purelin
%% Network training function that updates weight and bias values:
    scaled conjugate gradient method    trainscg    ; Note that
    Levenberg–Marquardt cannot be used here because it requires a
    large computing time and a large memory.

%% check if there is a missing data
```

```

no_data=0;
m=0;
for i=1:1:size(actual_Input,1)
    if isnan(actual_Input(i,1))
        m=m+1;
        if m>5
            no_data=1;
            break
        end
    end
end

%% neural network
if (isempty(household_Id_final)==0) && (no_data==0)
    p=Input(:,1:end);
    t=Target(:,1:end);
    net=feedforwardnet(100,'trainscg');
    net.numLayers=3;
    net.numInputs=1;
    net.biasConnect=[1;1;1];
    net.inputConnect=[1;0;0];
    net.layerConnect=[0 0 0;1 0 0;0 1 0];
    net.outputConnect=[0 0 1];
    net.layers{2}.dimensions=100;
    net.layers{1}.transferFcn='tansig';
    net.layers{2}.transferFcn='tansig';
    net.layers{3}.transferFcn='purelin';
    net.layers{2}.name='Hidden';
    net.layers{3}.name='output';

```

```

net.trainParam.epochs=100;
net.divideParam.trainRatio=0.85;
net.divideParam.valRatio=0.15;
net.divideParam.testRatio=0;
net.trainParam.showWindow = false;
net.inputs{1}.processFcns={'fixunknowns','removeconstantrows',
    'mapminmax'};
[net tr]=train(net,p,t);
if tr.best_vperf==0
    clear net tr
    net=feedforwardnet(100,'trainscg');
    net.numLayers=3;
    net.numInputs=1;
    net.biasConnect=[1;1;1];
    net.inputConnect=[1;0;0];
    net.layerConnect=[0 0 0;1 0 0;0 1 0];
    net.outputConnect=[0 0 1];
    net.layers{2}.dimensions=100;
    net.layers{1}.transferFcn='tansig';
    net.layers{2}.transferFcn='tansig';
    net.layers{3}.transferFcn='purelin';
    net.layers{2}.name='Hidden';
    net.layers{3}.name='output';
    net.trainParam.epochs=40;
    net.divideFcn='';
    net.trainParam.showWindow = false;
    net.inputs{1}.processFcns={'fixunknowns','removeconstantrows',
        'mapminmax'};
    [net tr]=train(net,p,t);

```

```

end

forecast_water_heater_power=net(actual_Input);

else

if store_FLC_in_database==1

for i=1:1:(future_forecast_horizon-start_forecast)

forecast_water_heater_power(i,1)=NaN;

end

else

for i=1:1:5

forecast_water_heater_power(i,1)=NaN;

end

end

end

ANN = forecast_water_heater_power;

%Run the Kalman

%Now use the Kalman Filter to estimate the error of the jth

forecast

Q = 1e-6;

R = .1;

M = 1;

e = 1;

FLC = ANN;

for ahead = 1:horizon

for k=1+ahead:length(t)

%Update the previous error estimation

z = measurement(k)-ANN(k-ahead, ahead);

G = M/(M+R);

e = e + G*(z-e);

```

```

M = (1-G)*M;

%estimate for current
%      e = e;
M = M + Q;

%Now create a new forecast
FLC(k, ahead) = History(k+ahead)+e;
end
end

private void ConvertCtrl2Unctrl(DateTime start, double P_c)
{
    //Whenever a PSU-power is created, this function is called to
    create another PSU that if no control action was executed,
    how PSU should be
    //Todo so, load the command history and use the reverse payback
    model to convert controlled load profile to uncontrolled
    load profile

    //The number of houses in each group as listed in the report on
    30 Aug. 2014
    int [] members = { 24, 27, 26, 22, 18, 19, 17, 17, 17, 19 };
    double P_u = P_c; //the estimated uncontrolled power
    double impact = 0; //the impact of each control action in the
    past will be acumulated

    DataView dvCommands = GetLcm(start.AddHours(-1));
    //now calculate the impact of each command on this interval

```

```

for (int i = 0; i < dvCommands.Count; i++)
{
    //get the number of houses in that group
    string grpName = dvCommands[i]["groupname"].ToString();
    //grpNo will change the number from the database into the
        index number
    int grpNo = int.Parse(grpName[grpName.Length - 1].ToString())
        - 1;

    //now find the time difference between command time and
        payback time
    int difference = 0;
    DateTime command_time = DateTime.Parse(dvCommands[i]["start"
        ].ToString());
    if (start - command_time > new TimeSpan(0, 45, 0)) difference
        = 2;
    else if (start - command_time > new TimeSpan(0, 30, 0))
        difference = 1;
    else difference = 0;
    int index_in_day = start.Hour * 4 + start.Minute;
    impact = impact + (payback_common_sets[index_in_day,
        difference] - payback_common_sets[index_in_day, difference
        + 1]) * (members[grpNo] / 200);
}
P_u = P_c / (1 + impact);

//now update this value in the table
SqlConnection objSqlCnt = new SqlConnection(connectionString);
SqlCommand objSqlCmd = new SqlCommand();

```

```

objSqlCommand = new SqlCommand("update PSU set unCtrlValue=@val
    where starttime=@start and type='power'");
objSqlCommand.Connection = objSqlCnt;
objSqlCommand.Parameters.AddWithValue("@start", start);
objSqlCommand.Parameters.AddWithValue("@val", P_u);
try { objSqlCnt.Open(); }
catch { }
try { objSqlCommand.ExecuteNonQuery(); }
catch (Exception ee) { MessageBox.Show(ee.Message); }
try { objSqlCnt.Close(); }
catch { }

}

```

A.7 Load Control

```

private void thread_control()
{
    SqlConnection objSqlCnt = new SqlConnection(connectionString);
    SqlCommand objSqlCommand = new SqlCommand();
    SqlDataAdapter objSqlDataAdp = new SqlDataAdapter();

    DateTime dt = new DateTime(DateTime.Now.year, DateTime.Now.
        Month, DateTime.Now.Day,
        DateTime.Now.Hour, DateTime.Now.Minute, 0);

    //Run the controller
    {
        index_counter += 1;
    }
}

```

```

//load GDI
DataView gdi = LoadGDI(dt, dt.AddMinutes(16));
if (gdi.Count == 0) continue;
double target_power = 1000 * double.Parse(gdi[0]["targetValue
    "].ToString());

//load forecasted power
DataView flc = LoadFLC(dt, dt.AddMinutes(16));
double forecasted_power = 4000 * double.Parse(flc[0]["value_c
    "].ToString());

//determine R_D and R_u
CalcReserveSet(dt);

//if the difference is not more than one DEWH, ignore this
    command and continue
if (Math.Abs(forecasted_power - target_power) >= 3)
{

    //Select which houses should turn off and on
    double error = forecasted_power - target_power;

    int number_of_loads = (int)(Math.Abs(error) / 3);

    if (error > 0)    //should turn off some heaters
    {
        for (int i = 0; i < Math.Min(ramp_down.Count,
            number_of_loads); i++)

```

```

        {
            //turn off this house
            ShedHouse((int)ramp_down[i], dt);
        }
    }
    else //should turn back on something
    {
        //we don't need to unshed, because they automatically
        turn on
    }
}

//update PSU
UpdateControlledPSU(dt);

}

btnSimFollowLoad.Text = "Load Following";
lblStatus.Text = "Idle";
stsProgress.Value = 0;
stsProgress.Visible = false;
//lblStatus.Text = "Idle";
}

private void CalcReserveSet(DateTime start_interval)
{
    ramp_up = new ArrayList();
    ramp_down = new ArrayList();
    string up_list = "";
    string down_list = "";

```

```

for (int i = 1; i < iMaxHouse; i++)
{
    if (excluded.Contains(i)) continue;
    //load on/off state of the DEWH at the given time
    if (LoadDEWHState(i, start_interval)) //if the uncontrolled
        house is on
    {

        if (ControlledDuration(i, start_interval) <
            max_allowed_control) //if it still can be kept off
        { ramp_down.Add(i); down_list = down_list + i + ", "; }
        if (CurrentlyControlled(i, start_interval)) //if it was
            controlled in previous interval, it can be used to
            increase the load
        { ramp_up.Add(i); up_list = up_list + i + ", "; }
    }
}

//now store the reserve sets in the database
//SQL variables
SqlConnection objSqlCnt = new SqlConnection(connectionString);
SqlCommand objSqlCmd = new SqlCommand();
objSqlCmd = new SqlCommand("insert into ReserveSet (time,
    reserve_up, list_up, reserve_down, list_down) VALUES(@date,
    @up, @lstup, @down, @lstdown)");
objSqlCmd.Connection = objSqlCnt;
objSqlCmd.Parameters.AddWithValue("@date", start_interval.
    ToString());
objSqlCmd.Parameters.AddWithValue("@up", ramp_up.Count * 3);

```

```

objSqlCommand.Parameters.AddWithValue("@lstop", up_list);
objSqlCommand.Parameters.AddWithValue("@down", ramp_down.Count * 3)
    ;
objSqlCommand.Parameters.AddWithValue("@lstdown", down_list);
try { objSqlCnt.Open(); objSqlCommand.ExecuteNonQuery(); objSqlCnt.
    Close(); }
catch (Exception ee) { MessageBox.Show(ee.Message); }
}

private void LoadFollowingControl(DateTime start)
{
    //This function should be called 2 minutes before each 15
    //minutes sharp due to LCM communication delay
    //it will compare the GDI with the forecasted load, then decide
    //how many houses must be turned off for the next 15 minutes

    //at each action time, the R_u group will be turned back on for
    //the next time interval
    //so the power consumption will be flc+R_u, this value should
    //be compared with the GDI to select houses to be off

    //get the GDI for the next interval
    DataView dvGDI = GetGDI(start, start.AddMinutes(15));
    if (dvGDI.Count == 0) return;
    double gdi = double.Parse(dvGDI[0]["targetvalue"].ToString());

    //get the forecast for the next interval
    DataView dvFLC = GetMostRecentFLC(start);

```

```

double flc = double.Parse(dvFLC[0]["forecastedload"].ToString()
    );
int group_number = -1;

DataView dvDummyPSU = GetMostRecentDummyPSU(start);
double dummyPSU = double.Parse(dvDummyPSU[0]["value"].ToString()
    ());
int total_included_house = dvDummyPSU[0]["houseList"].ToString()
    ().Split(',').Length - 1;

double error = flc - gdi;

if (Math.Abs(error) < 0.015) //if error is less than 15 Kw, do
    nothing
    return;

if (gdi > flc) //turn on some groups if possible
{
    //We are not implementing this part for now
    //only the GDIs less than the forecasted load will be
        followed
}
else //turn off another group
{
    int required_house = (int)(total_included_house * error / flc
        );

    //select some groups to turn off the required number of
        houses

```

```

    group_number = GetGoodGroups(required_house , start);
    string[] shed_list = { "PSA Group" + group_number };

    ShedLoad(shed_list);
}

// Now turn on groups that were turned off in the previous
// action
if (R_u.Count > 0)
{
    ArrayList houseIDs = new ArrayList();
    int group_counter = 0;
    for (int i = 0; i < R_u.Count; i++)
    {
        int number = (int)R_u[i] + 1; //because index starts from 0
        //but group names start from 1
        if (group_number == number) continue;
        houseIDs.Add("PSA Group" + number);
    }
    R_u.Clear();
    string[] hIDs = new string[houseIDs.Count];
    for (int i = 0; i < houseIDs.Count; i++) hIDs[i] = (string)
        houseIDs[i];
    UnshedLoad(hIDs);
}
}

```

```

private bool IsSelectable(int group_number, DateTime start)
{
    //find out how many times this group was turned off in 2 hours
    //before start time

    string group = "PSA Group" + (group_number + 1);
    SqlConnection objSqlCnt = new SqlConnection(connectionString);
    SqlCommand objSqlCmd = new SqlCommand();
    SqlDataAdapter objSqlDataAdp = new SqlDataAdapter();

    //Load house list information
    objSqlDataAdp.SelectCommand = new SqlCommand();
    objSqlDataAdp.SelectCommand.CommandText = "select    cmd, start,
        GroupName from lcm_commands where cmd='Shed' and start>=@st
        and start<=@end";
    objSqlDataAdp.SelectCommand.Parameters.AddWithValue("@st",
        start.AddHours(-2));
    objSqlDataAdp.SelectCommand.Parameters.AddWithValue("@end",
        start);

    objSqlDataAdp.SelectCommand.CommandType = CommandType.Text;
    objSqlDataAdp.SelectCommand.Connection = objSqlCnt;
    DataSet dsHouseList = new DataSet();
    try { objSqlDataAdp.Fill(dsHouseList, "lcms"); }
    catch (Exception ee) { return false; }
    if (dsHouseList.Tables.Count == 0 || dsHouseList.Tables[0].Rows
        .Count <= 4)
        return true;
    return false;
}

```

```

private int GetGoodGroups(int required_number , DateTime
    action_start)
{
    //return group ranges from 1..10
    int houseIds = -1;

    //check which n_group is close to the required_number among
        those which are not in R_u
    int [] distance = new int[n_groups.Length];
    int min = 1000;
    int min_index = -1;

    for (int i = 0; i < distance.Length; i++)
    {
        if (!IsSelectable(i, action_start)) continue; //if it was not
            turned off more than 1 hours in the past two hours
        if (!R_u.Contains(i)) distance[i] = Math.Abs(required_number
            - n_groups[i]);
        else distance[i] = 1000;
        if (distance[i] < min) { min = distance[i]; min_index = i; }
    }

    houseIds = (min_index + 1);
    R_u.Add(min_index);
    return houseIds;
}

```

B Forecast Evaluation

The two forecasting methods were applied in a pilot containing approximately 100 electric water heaters. Every 5 minutes, each algorithm creates a load forecast for each individual load for the next 4 hours in 5 minutes intervals. So each forecast contains 48 samples. The forecasted profiles are analyzed and the results are illustrated in the following sections.

B.1 Single Load Forecast Evaluation

Table B.1: Average performance of the forecasting methods for single load.

	Statistic-Based			ANN-Based		
index	$S_R(j)$	$E\{\tilde{s}\} = \eta$	$E\{(\tilde{s})^2\} = \gamma^2$	$S_R(j)$	$E\{\tilde{s}\} = \eta$	$E\{(\tilde{s})^2\} = \gamma^2$
1	0.97	0.02	0.03	0.78	0.19	0.21
2	0.93	0.04	0.06	0.78	0.19	0.22
3	0.9	0.08	0.1	0.77	0.2	0.22
4	0.9	0.08	0.1	0.76	0.2	0.21
5	0.88	0.09	0.1	0.76	0.2	0.22
6	0.84	0.12	0.12	0.76	0.21	0.24
7	0.82	0.16	0.19	0.75	0.22	0.24
8	0.8	0.18	0.2	0.76	0.22	0.24
9	0.8	0.21	0.22	0.75	0.21	0.24
10	0.78	0.19	0.23	0.74	0.22	0.24
11	0.78	0.2	0.22	0.74	0.25	0.25

12	0.76	0.21	0.22	0.75	0.23	0.25
13	0.75	0.23	0.25	0.76	0.24	0.25
14	0.77	0.22	0.24	0.75	0.24	0.25
15	0.76	0.24	0.21	0.75	0.24	0.25
16	0.74	0.25	0.21	0.74	0.22	0.24
17	0.75	0.22	0.21	0.73	0.25	0.25
18	0.71	0.25	0.21	0.72	0.26	0.25
19	0.73	0.26	0.22	0.71	0.26	0.28
20	0.75	0.24	0.22	0.68	0.28	0.3
21	0.72	0.26	0.23	0.69	0.28	0.3
22	0.74	0.26	0.23	0.68	0.29	0.3
23	0.71	0.28	0.24	0.68	0.28	0.3
24	0.7	0.28	0.23	0.67	0.28	0.3
25	0.68	0.29	0.24	0.68	0.28	0.3
26	0.64	0.31	0.24	0.69	0.3	0.3
27	0.63	0.34	0.27	0.68	0.3	0.3
28	0.65	0.34	0.28	0.68	0.3	0.31
29	0.67	0.35	0.28	0.68	0.31	0.33
30	0.63	0.36	0.28	0.69	0.31	0.33
31	0.63	0.36	0.28	0.7	0.3	0.33
32	0.61	0.39	0.3	0.69	0.3	0.33
33	0.63	0.37	0.3	0.68	0.3	0.33
34	0.6	0.38	0.31	0.67	0.31	0.33
35	0.61	0.36	0.31	0.68	0.3	0.33
36	0.62	0.36	0.31	0.67	0.31	0.32
37	0.61	0.39	0.32	0.68	0.3	0.32
38	0.59	0.4	0.32	0.66	0.32	0.33
39	0.62	0.38	0.31	0.66	0.31	0.33
40	0.58	0.39	0.31	0.67	0.31	0.32
41	0.59	0.4	0.32	0.67	0.32	0.33
42	0.61	0.38	0.31	0.66	0.32	0.33

43	0.6	0.39	0.31	0.67	0.33	0.33
44	0.59	0.39	0.31	0.68	0.32	0.33
45	0.59	0.4	0.32	0.69	0.31	0.33
46	0.57	0.41	0.31	0.68	0.3	0.33
47	0.58	0.41	0.31	0.68	0.3	0.33
48	0.57	0.41	0.32	0.67	0.3	0.33

B.2 Sets of *ON* Loads

The aggregated payback model presented in Section 3.4.3 depends on the average number of the loads that consumes power over consecutive time intervals.

Through direct measurements of the load power consumption of approximately 100 electric water heaters, the number of *on* loads at each time interval was measured (named $n(A_0)$). Then, the number of the loads that remained *on* during the next interval was measured (named $n(A_0, A_1)$), and so on for the next 5 time intervals. The results are listed in Table B.2.

Table B.2: Average number of the loads that remains *on* during consecutive time intervals

time interval	$n(A_0)$	$n(A_0, A_1)$	$n(A_0, \dots, A_2)$	$n(A_0, \dots, A_3)$	$n(A_0, \dots, A_4)$	$n(A_0, \dots, A_5)$
00:00-00:15	12.52	5.64	1.80	1.28	1.08	0.92
00:15-00:30	11.60	5.08	1.60	1.20	0.96	0.88
00:30-00:45	12.28	4.72	1.56	1.16	1.00	0.92
00:45-01:00	9.88	4.08	1.36	1.12	1.00	0.92
01:00-01:15	9.00	3.60	1.32	1.16	1.08	0.92

01:15-01:30	8.96	3.76	1.24	1.16	0.96	0.80
01:30-01:45	9.64	3.76	1.24	1.00	0.84	0.76
01:45-02:00	8.52	3.64	1.08	0.92	0.84	0.76
02:00-02:15	8.72	3.48	1.04	0.96	0.84	0.68
02:15-02:30	8.80	3.56	1.12	0.96	0.76	0.60
02:30-02:45	8.72	3.80	1.08	0.88	0.68	0.56
02:45-03:00	8.32	3.16	0.88	0.68	0.56	0.48
03:00-03:15	8.52	3.68	0.88	0.68	0.56	0.48
03:15-03:30	8.08	2.76	0.88	0.72	0.56	0.56
03:30-03:45	7.92	3.24	0.88	0.68	0.68	0.60
03:45-04:00	7.72	3.20	0.84	0.76	0.68	0.68
04:00-04:15	8.64	3.60	1.12	0.80	0.76	0.64
04:15-04:30	8.40	3.48	1.04	1.00	0.76	0.64
04:30-04:45	8.28	3.32	1.08	0.84	0.68	0.64
04:45-05:00	7.64	3.44	1.12	0.92	0.84	0.80
05:00-05:15	7.76	3.00	1.24	1.08	0.96	0.80
05:15-05:30	8.48	4.64	1.76	1.52	1.20	1.08
05:30-05:45	11.08	5.88	3.48	2.76	2.56	2.12
05:45-06:00	13.04	7.84	5.08	4.36	3.40	2.60
06:00-06:15	15.80	10.76	7.52	5.96	4.68	3.68
06:15-06:30	18.96	13.40	8.88	6.64	5.16	4.24
06:30-06:45	23.24	16.60	11.20	8.68	7.20	6.32
06:45-07:00	25.44	17.56	11.92	9.72	8.48	7.12
07:00-07:15	26.08	18.40	13.44	11.56	9.44	6.88
07:15-07:30	27.32	19.64	14.44	11.56	8.20	5.92
07:30-07:45	29.04	20.56	14.44	10.00	7.04	5.28
07:45-08:00	30.00	21.08	13.28	9.32	6.92	5.52
08:00-08:15	31.56	21.40	12.96	9.16	7.32	5.84
08:15-08:30	31.48	19.96	12.16	9.28	6.92	5.32
08:30-08:45	29.36	18.64	11.68	8.68	6.56	5.20
08:45-09:00	27.88	17.84	11.72	8.68	6.64	4.80

09:00-09:15	27.76	18.36	11.68	8.84	6.36	4.76
09:15-09:30	27.36	17.80	11.24	8.20	6.16	5.16
09:30-09:45	26.20	17.00	10.84	8.04	6.48	5.16
09:45-10:00	26.32	16.96	10.80	8.32	6.36	4.88
10:00-10:15	25.92	16.36	9.96	7.40	5.48	4.20
10:15-10:30	24.52	14.72	9.24	6.68	5.04	3.92
10:30-10:45	24.24	15.92	9.08	6.88	5.28	4.08
10:45-11:00	24.60	14.56	8.76	6.68	4.92	3.76
11:00-11:15	23.92	14.56	8.84	6.44	4.84	3.92
11:15-11:30	23.56	15.20	8.60	6.40	4.96	3.96
11:30-11:45	23.92	14.32	8.56	6.48	5.04	4.20
11:45-12:00	24.00	14.20	8.40	5.96	4.88	3.80
12:00-12:15	23.28	14.16	7.80	6.08	4.48	3.72
12:15-12:30	23.52	13.40	7.48	5.36	4.32	3.04
12:30-12:45	22.44	13.16	7.48	5.84	3.84	3.16
12:45-13:00	22.92	12.84	7.88	5.00	4.04	3.28
13:00-13:15	21.56	13.48	6.80	5.40	4.24	3.28
13:15-13:30	22.84	12.44	7.20	5.40	4.00	3.16
13:30-13:45	20.84	12.08	7.12	5.24	3.96	3.04
13:45-14:00	20.52	12.20	6.80	4.96	3.48	2.80
14:00-14:15	20.12	11.20	5.84	4.16	3.28	2.68
14:15-14:30	19.16	10.84	5.60	4.32	3.24	2.64
14:30-14:45	18.44	10.04	5.36	3.84	3.00	2.40
14:45-15:00	17.88	10.00	5.28	3.96	3.00	2.32
15:00-15:15	18.04	10.32	5.20	3.80	2.80	2.24
15:15-15:30	19.44	10.88	5.24	3.76	2.88	2.28
15:30-15:45	19.44	10.48	5.08	3.80	3.04	2.60
15:45-16:00	19.76	10.68	5.56	4.16	3.48	2.80
16:00-16:15	18.92	10.40	5.88	4.56	3.56	3.04
16:15-16:30	19.00	10.84	6.24	4.76	3.84	3.04
16:30-16:45	20.08	11.40	6.36	4.92	3.72	3.16

16:45-17:00	22.04	13.00	7.24	5.32	3.92	3.24
17:00-17:15	23.52	13.88	7.16	5.28	4.20	3.52
17:15-17:30	25.40	14.84	8.20	5.92	4.60	3.80
17:30-17:45	27.04	16.40	9.12	6.92	5.52	4.32
17:45-18:00	28.56	17.28	10.12	8.16	6.32	5.04
18:00-18:15	29.24	17.76	11.20	8.56	6.84	5.56
18:15-18:30	29.12	19.04	12.20	9.32	7.68	6.64
18:30-18:45	30.04	19.92	12.28	10.00	8.24	6.20
18:45-19:00	29.52	18.92	12.60	10.20	7.68	6.44
19:00-19:15	27.96	18.44	12.28	9.04	7.24	5.56
19:15-19:30	27.08	18.08	11.32	8.92	6.76	5.48
19:30-19:45	26.64	17.16	11.24	8.68	6.88	5.44
19:45-20:00	25.52	16.44	10.76	8.20	6.28	4.76
20:00-20:15	26.04	17.04	10.00	7.28	5.36	4.16
20:15-20:30	25.52	14.24	8.44	6.16	4.72	3.80
20:30-20:45	21.96	13.24	7.12	5.40	4.32	3.72
20:45-21:00	21.48	12.20	6.84	5.48	4.48	3.52
21:00-21:15	20.04	11.64	6.96	5.56	4.20	3.08
21:15-21:30	20.44	12.36	7.16	5.56	4.12	2.88
21:30-21:45	20.64	11.92	6.64	4.84	3.28	2.48
21:45-22:00	20.88	11.68	6.24	4.04	3.08	2.56
22:00-22:15	20.68	0.00	0.00	0.00	0.00	0.00
22:15-22:30	21.12	0.00	0.00	0.00	0.00	0.00
22:30-22:45	19.24	0.00	0.00	0.00	0.00	0.00
22:45-23:00	17.84	0.00	0.00	0.00	0.00	0.00
23:00-23:15	17.92	0.00	0.00	0.00	0.00	0.00
23:15-23:30	16.44	0.00	0.00	0.00	0.00	0.00
23:30-23:45	14.00	0.00	0.00	0.00	0.00	0.00
23:45-00:00	14.60	0.00	0.00	0.00	0.00	0.00

C Time Constraints for Customer Comfort

C.1 Worst Case Scenario

One of the important constraints in designing the load control program is to maintain customer comfort. The customer comfort in electric water heaters can be characterized with the water temperature. In this dissertation, it is assumed that the measurement of the water temperature is not available; thus, a time limit was imposed to the control system to maintain customer comfort by limiting the time interval in which a water heater could be interrupted.

Customer comfort mostly depends on the user experience during direct use of hot water. Taking showers is one of the most critical time intervals for providing hot water. Several studies [12, 81] have shown that a shower may take approximately 10 – 15 minutes with the amount of 60 – 80 Liters and assuming that the rate of water usage is constant during a shower; results in $W_D \cong 5\text{Liter}/\text{min}$.

Let us define the worst case scenario in a way that two showers within approximately 1 hour happens, as depicted in Figure C.1 (top). This water usage was used as an input to the simulator which was discussed in Chapter 2 to estimate the power consumption of the water heater and its water temperature.

In order to study the impact of load interruption on the water temperature, various control signals were applied to the simulator. Figure C.1 depicts one of these tests

in which $l = 4$ interruption commands of 15 minutes length were issued to the water heater. As can be seen in the resulting temperature, the minimum temperature is dropped from approximately $49^{\circ}C$ to approximately $45^{\circ}C$, which still can be considered warm water for a shower.

This study suggests that considering $l = 4$ samples of 15-minutes intervals in an $L = 8$ samples window will not compromise customer comfort significantly. This time constraint was also imposed in the pilot projects containing approximately 100 water heaters, and during the operation of the control system, no complaints were received from the customers.

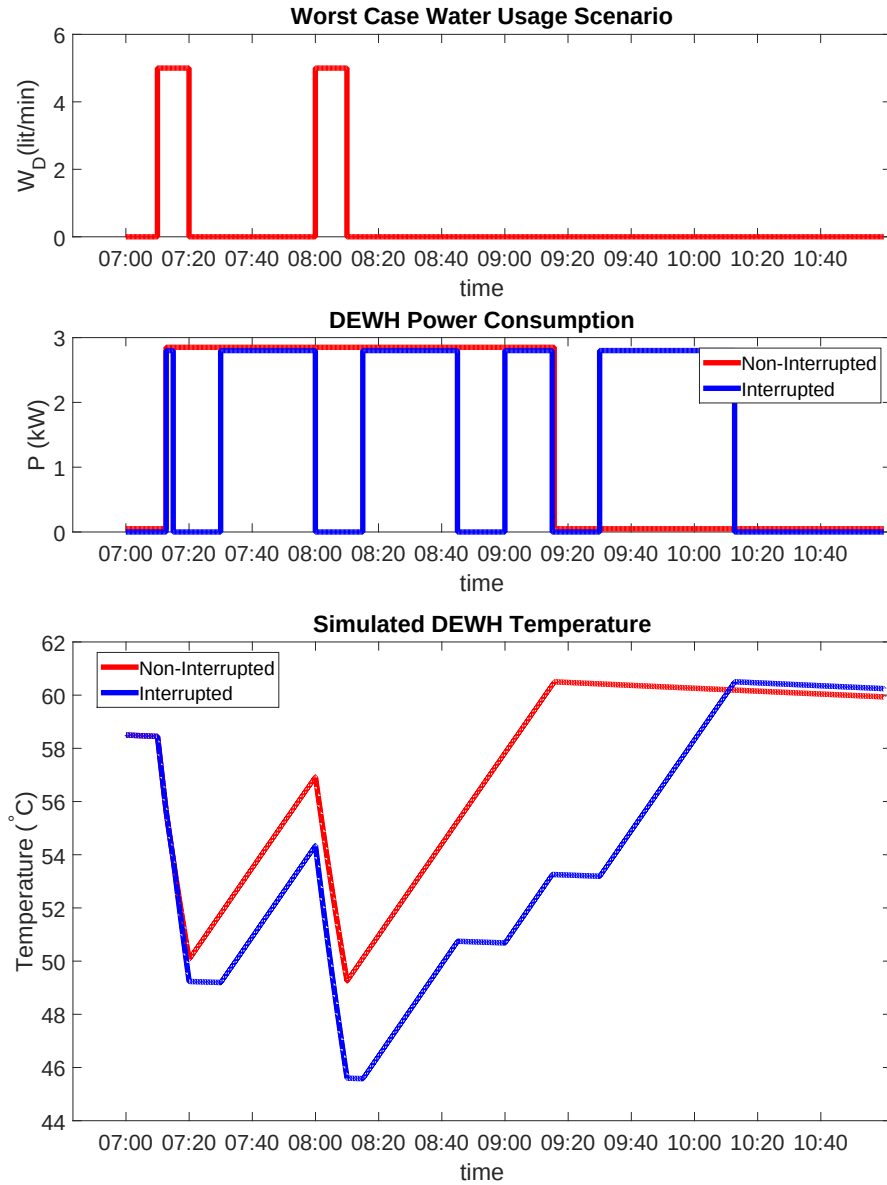


Figure C.1: The worst case scenario for water usage (top figure) and its resulting power consumption (middle) and water temperature(bottom) with and without external control signal.

Vita

Candidate's full name: Mostafa Shad Zolpirani

University attended:

1. Sharif University of Technology, Tehran, Iran, Sep. 1999 - May 2004, B.Sc. Computer Engineering / Hardware Engineering
2. Sharif University of Technology, Tehran, Iran, Sep. 2004 - Nov. 2006, M.Sc. Computer Engineering / Computer Architecture
3. University of New Brunswick, Fredericton, New Brunswick, Canada, Sep. 2009 - Mar. 2016, PhD Electrical and Computer Engineering

Publications:

Conference Presentations:

1. **M. Shaad**, C.P. Diduch, M.E. Kaye and L. Chang, "A Basic Load Following Control Strategy in a Direct Load Control Program," in Proceedings of the 9th IEEE Electrical Power and Energy Conference (EPEC 2015), Nov. 26 – 28, 2015, London, ON, Canada
2. **M. Shaad**, A. Momeni, C.P. Diduch, M.E. Kaye and L. Chang, "Forecasting the Power Consumption of a Single Domestic Electric Water Heater for a Direct Load Control Program," in Proceedings of the 28th Canadian Conference on Electrical and Computer Engineering (CCECE 2015), May 3–6, 2015, Halifax, Canada
3. **M. Shaad**, R. Errouissi, C.P. Diduch, M.E. Kaye and L. Chang, "Aggregate Load Forecast with Payback Model of the Electric Water Heaters for a Direct Load Control Program," in Proceedings of the 8th IEEE Electrical Power and Energy Conference (EPEC 2014), Nov. 12 – 14, 2014, Calgary, Canada

4. C. Diduch, **M. Shaad**, R. Errouissi, M.E. Kaye, J. Meng and L. Chang, "Aggregated Domestic Electric Water Heater Control - Building on Smart Grid Infrastructure," in Proceedings of the 7th International Power Electronics and motion control (IPEMC 2012), June 2 – 5, 2012, Harbin, China
5. **M. Shaad**, A. Momeni, C. P. Diduch, M. Kaye, L. Chang, "Parameter Identification of Thermal Models for Domestic Electric Water Heaters in a Direct Load Control Program," in Proceedings of the 25th IEEE Canadian Conference on Electrical and Computer Engineering (CCECE 2012), April 29 - May 2, 2012, Montreal, Canada
6. **M. Shaad**, W. G. Morsi, C. P. Diduch, H. Li, L. Chang, "A Centralized Genetic Algorithm Controller to Control Power Demands of Electric Water Heaters for Peak Shaving of Residential Consumers," in Proceedings of the 4th IEEE Electrical Power and Energy Conference (EPEC 2010), August 25 – 27, 2010, Halifax, NS, Canada
7. **M. Shaad Zolpirani**, M.-M. Bidmeshki and S.G. Miremadi., "The Effect of Routing-Update Time on Networks Performability," in Proceedings of The 5th ACS/IEEE International Conference on Computer Systems and Applications, AICCSA 2007, May 13 – 16, 2007, Amman, Jordan
8. **M. Shaad Zolpirani**, M.-M. Bidmeshki and S.G. Miremadi., "Improving the Performability of Networks Using Parallel Processing," in Proceedings of the 6th IEEE International Conference on Networking (ICN 07), April 22 – 28, 2007, Martinique, French Caribbean
9. M.-M. Bidmeshki, **M. Shaad Zolpirani**, and S.G. Miremadi., "Performability Estimation of Network Services in the Presence of Component Failure," in Proceedings of the 2nd International Conference on Telecommunications and Networking (TeNe 06), December 2006
10. Y. Sedaghat, S. Hessabi, and S. G. Miremadi and **M. Shaad Zolpirani**, "A Method to Improve the Performability of NOC Using Dual Source-Router Conjunction," in Proceedings of 11th. International CSI Computer Conference (CSICC 06), Vol. 1, pp.115 – 121, Tehran, Iran, Jan. 2006. (in Persian)
11. A. Marjovi, S. Bagheri Shooraki, **M. Shaad Zolpirani** and M. H. Radan, "Design and Modeling a Hybrid 4-DOF with the Ability to Perform Special Operations on Branched Poles and it's Controller," in Proceedings of 10th International CSI Computer Conference (CSICC 06), Vol. 1, pp.115 – 121, Tehran, Iran, Jan. 2006. (in Persian)
12. **M. Shad Zolpirani**, R. Majdodin and S. Baghri Shooraki, "Design and Implementation of a 3-dimension CNC Controller Using a Fuzzy Self-Organizer Controller (FUZZY SOC)," in Proceedings of the 13th Annual International Mechanical Engineering Conference (ISME 05), Isfahan University of Technology, Isfahan, Iran, May 2005.

Journal Publication :

1. **M. Shaad**, A. Momeni, R. Errousi, C.P. Diduch, M.E. Kaye and L. Chang, "Identification and Estimation for Electric Water Heaters in Direct Load Control Programs," IEEE Transactions on Smart Grid, vol. 7, no. 99, pp. 1-9, 2015