UNIVERSITY OF CALIFORNIA, SAN DIEGO

Optimized Energy Control in Power Distribution Systems

A dissertation submitted in partial satisfaction of the
requirements for the degree
Doctor of Philosophy

in

Electrical Engineering (Communication Theory and Systems)

by

Alper Sinan Akyurek

Committee in charge:

Professor Tajana Simunic Rosing, Chair
Professor Sujit Dey
Professor Tara Javidi
Professor Ryan Kastner
Professor Miroslav Krstic

2017

The dissertation of Alper Sinan Akyurek is approved, and it is acceptable in quality and form for publication on microfilm and electronically:

_____

_____

_____

_____

_____
Chair

University of California, San Diego

2017

DEDICATION

To my wife Özge - the infinity,
my daughter Gökçe - the sky,
and my family - the Whitehearts.

EPIGRAPH

To know the future, is to be trapped by it.
Knowing there is a trap, is the first step in evading it.
—Paul Muad'Dib Atreidis

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

ACKNOWLEDGMENTS

I would like to start by thanking my advisor Prof. Tajana Rosing for her encouragement, guidance, trust, understanding and support throughout my PhD. Her experience and knowledge has kept me in the right direction throughout my years. I am extremely grateful for her support and understanding in both work and off-work difficulties, that have made this PhD possible. I would like to thank my committee members, Prof. Sujit Dey, Prof. Tara Javidi, Prof. Ryan Kastner and Prof. Miroslav Krstic for their valuable discussions, feedback and contributions to this work.

I would like to thank the Terraswarm Research Center, Qualcomm, NSF Grant 1344153 and UCSD Center for Networked Systems (CNS) for their generous support, which made this research possible.

I would like to thank all my lab mates and friends in SEELab, both current and past, for their comments, feedback, discussions, guidance and lots of good memories. It was a privilege to work with and along-side them. I am grateful to all of them, but I would also like to give special thanks to Baris Aksanli, Jagannathan Venkatesh and Christine Chan.

A special thanks to Prof. Elif Uysal Biyikoglu for her support and guidance over the many years, and for encouraging me to the path of PhD in the first place.

Most importantly, I owe so much to my wife, my daughter, my sister, my parents and all my family. I am the person I am now, because of them. I would like to thank them for their unconditional and unending love, support and encouragement throughout everything. Coming to a distant foreign country, while leaving all comforts behind is a tough commitment and a dark unknown road for a couple. Thank you Özge, for always being there for me and shining your strong light for both (+1) of us.

Chapter 2 contains material from Bengu Ozge Akyurek, Alper Sinan Akyurek, Jan Kleissl and Tajana Simunic Rosing, "TESLA: Taylor Expanded Solar Analog Forecasting", IEEE International Conference on Smart Grid Communications (SmartGridComm), 2014 [1]. The dissertation author was one of the primary

investigators and the second author of this paper.

Chapter 4 contains material from Alper Sinan Akyurek, Baris Aksanli and Tajana Simunic Rosing, "S2Sim: Smart Grid Swarm Simulator", International Green and Sustainable Computing Conference (IGSC), 2015 [2]. The dissertation author was the primary investigator and author of this paper.

Chapter 3 contains material from Alper Sinan Akyurek and Tajana Simunic Rosing, "ECO-DAC: Energy Control Over Divide And Control", IEEE International Conference on Smart Grid Communications (SmartGridComm), 2013 [3]. The dissertation author was the primary investigator and author of this paper.

Chapter 3 contains material from Alper Sinan Akyurek and Tajana Simunic Rosing, "Optimal Nonlinear Distributed Battery Control", IEEE Journal of Emerging and Selected Topics in Power Electronics, Special Issue on Structured DC Microgrids, 2016 [4]. The dissertation author was the primary investigator and author of this paper.

Chapter 5 contains material from Alper Sinan Akyurek and Tajana Simunic Rosing, "Optimal in-network packet aggregation policy for maximum information freshness in wireless sensor networks", European Conference on Networks and Communications, 2016 [5]. The dissertation author was the primary investigator and author of this paper.

Chapter 5 contains material from Alper Sinan Akyurek and Tajana Simunic Rosing, "Optimal Packet Aggregation in Wireless Networks", which was submitted for consideration to IEEE Transactions on Mobile Computing. The dissertation author was the primary investigator and author of this paper.

VITA

2008	B. Sc. in Electrical and Electronics Engineering, Middle East Technical University, Ankara - Turkey
2011	M. Sc. in Electrical and Electronics Engineering, Middle East Technical University, Ankara - Turkey
2017	Ph. D. in Electrical Engineering (Communication Theory and Systems), University of California, San Diego

PUBLICATIONS

A. Sinan Akyurek, A. E. Xhafa, J. Zhou, R. Vedantham "Compression of Internet Protocol Version 6 Addresses in Wireless Sensor Networks", US Patent 20,160,088,517

A. Sinan Akyurek, A. E. Xhafa, J. Zhou, R. Vedantham "Address Generation for Networks", US Patent 20,160,087,936

A. Sinan Akyurek, A. E. Xhafa, J. Zhou, R. Vedantham "System and method for distributed scheduling of transmission resources", US Patent App. 15/054,508

A. Sinan Akyurek and Tajana Simunic Rosing, "Optimal Packet Aggregation in Wireless Networks", IEEE Transactions on Mobile Computing (under review)

Jinseok Yang, A. Sinan Akyurek, Sameer Tilak and Tajana S. Rosing, "Design of transmission manager in heterogeneous WSNs", IEEE Transactions on Emerging Topics in Computing

A. Sinan Akyurek and Tajana Simunic Rosing, "Optimal Distributed Nonlinear Battery Control", IEEE Journal of Emerging and Selected Topics in Power Electronics, Special Issue on Structured DC Microgrids, 2016

Jagannathan Venkatesh, Baris Aksanli, Christine Chan, Alper S. Akyurek and Tajana Rosing, "Scalable Application Design for the Internet of Things", IEEE Software Special Issue on Software Engineering for the Internet of Things, 2016

A. Sinan Akyurek and Elif Uysal-Biyikoglu, "A Depth-Optimal Low Complexity Distributed Wireless Multicast Algorithm", The Computer Journal 2011

Akanksha Maurya, Alper Sinan Akyurek, Baris Aksanli and Tajana Rosing, "Time-Series Clustering for Data Analysis in Smart Grid", International Conference on Smart Grid Communications (SmartGridComm) 2016

A. Sinan Akyurek, Tajana S. Rosing, "Optimal In-Network Packet Aggregation Policy for Maximum Information Freshness", European Conference on Networks and Communications (EuCNC) 2016

Jagannathan Venkatesh, Christine Chan, Alper S. Akyurek, Tajana S. Rosing, "A Modular Approach to Context-Aware IoT Applications", IEEE International Conference on Internet-of-Things Design and Implementation (IoTDI) 2016

A. Sinan Akyurek, Baris Aksanli, Tajana S. Rosing, "S2Sim: Smart Grid Swarm Simulator", International Green and Sustainable Computing Conference (IGSC) 2015

Jagannathan Venkatesh, Christine Chan, Alper S. Akyurek, Tajana S. Rosing, "Sensitive Applications", TechCon 2015

Christine Chan, Alper S. Akyurek, Kalyan Vaidyanathan, Kenny Gross, Tajana S. Rosing, "Optimization of Energy, Cooling and IO Performance for Data-intensive Applications on Enterprise Servers", TechCon 2015

Baris Aksanli, Alper S. Akyurek, Tajana S. Rosing, "User Behavior Modeling for Estimating Residential Energy Consumption", EAI International Conference on Smart Grids for Smart Cities 2015

Baris Aksanli, Alper S. Akyurek, Tajana S. Rosing, "Minimizing the Effects of Data Centers on Microgrid Instability", International Green and Sustainable Computing Conference (IGSC) 2015

Henrique Rodrigues, Richard Strong, A. Sinan Akyurek, Tajana S. Rosing, "Dynamic Optical Switching for Latency", ACM/IEEE International Symposium on Architectures for Networking and Communication Systems (ANCS) 2015

B. Ozge Akyurek, A. Sinan Akyurek, Jan Kleissl, Tajana S. Rosing, "TESLA: Taylor Expanded Solar Analog Forecasting", IEEE SmartGridComm 2014

Baris Aksanli, Alper S. Akyurek, et. al, "Distributed Control of a Swarm of Buildings Connected to a Smart Grid", 1st ACM International Conference on Embedded Systems For Energy-Efficient Buildings (BuildSys) 2014

Henrique Rodrigues, Alper S. Akyurek, C. Guok, I. Monga, Tajana S. Rosing, "Enabling SDN-based Multi-Layer Provisioning and Optimization for Inter Data Center Networks with Unified Packet-Optical Control Plane", OIDA Software Defined Photonic and Data Center Networks Workshop, 2014

A. Sinan Akyurek, Bill Torre, Tajana S. Rosing, "ECO-DAC: Energy Control Over Divide And Control", IEEE SmartGridComm 2013

A. Sinan Akyurek and Elif Uysal-Biyikoglu, "A Low Complexity Distributed Algorithm for Computing Minimum-Depth Multicast Trees in Wireless Networks", MILCOM 2010

A. Sinan Akyurek and Elif Uysal-Biyikoglu, "Kablosuz Aglarda Cogagonderim icin Yeni bir Yol Atama Algoritmasi", IEEE SIU 2008

ABSTRACT OF THE DISSERTATION

Optimized Energy Control in Power Distribution Systems

by

Alper Sinan Akyurek

Doctor of Philosophy in Electrical Engineering (Communication Theory and Systems)

University of California, San Diego, 2017

Professor Tajana Simunic Rosing, Chair

Advancements in communication and computation, energy storage and renewable generation, along with the recent support of policy makers have been transforming the power grid into the Smart Grid. In this dissertation, we propose an optimized control framework that addresses four major interconnected problems present in today's grid. We develop a fast, low complexity and accurate forecast algorithm that supplies solar energy predictions to the control algorithms. We improve the accuracy by up to 50% as compared to the state of the art solutions. We complement this with an optimal distributed nonlinear battery control algorithm that uses a high-accuracy nonlinear battery model. Our solution reduces the utility bill of an actual building by up to 50%, a 25% improvement over heuristic

solutions. Furthermore, we show that state of the art linear optimal control algorithms incur up to 250% higher costs due to model inaccuracies accumulated over time. We develop a smart grid simulator, S2Sim, that enables the co-simulation of distributed control algorithms. Using S2Sim we show that our battery control algorithm can improve the stability of the grid by up to 45%. Finally, we develop an optimal packet aggregation solution that can optimize individual goals, such as information freshness or energy, in a network-wide manner. We show through case studies that our solution has constant performance under increasing congestion, reduces energy consumption by up to 60%, increases information freshness by up to 55% and adapts to dynamic conditions in real-time, an important quality for distributed control in future Smart Grid applications.

Chapter 1

Introduction

Until recently, the electrical power grid has not changed dramatically from its base design a century ago [6]. However, in the recent years, advancements in power generation technology, a shift from centralized to distributed generation and control, transition from manual control to automation and climate change related policies have led to an advancement to a *Smart(er) Grid* [7]. The current power grid is enormous in size and spans multiple interconnected networks including millions of people, commercial buildings and industrial buildings. Massive power generation plants produce electricity in distant locations. The generated energy is transmitted to central locations using high voltage transmission lines and distributed to customers through the low-medium voltage distribution networks. The tolerance of the systems comes from reserve generators and the mechanical inertia of the spinning generators. Control systems are centralized and require information only from important centralized locations for operational observations.

Transition from such a centralized, deterministic and human-dependent system to a distributed, stochastic and automated system is a big research challenge with a multitude of interconnected issues. Renewable resources are, by their nature, hard to predict stochastic generation processes with rapid variations associated with climate conditions. In contrast to conventional power plants, renewable generation, such as photo-voltaic (PV) generation, can be installed in a distributed fashion at residential locations behind the utility substation transformer. Without distributed generation, the utility company can design the substation trans-

former's voltage output according to the line-loss related voltage drops. However, when distributed generation is present, this static design is not feasible and must be addressed to avoid voltage instability. The electrification of the transportation system also has a major impact on the grid. Electrical vehicles require large amounts of power during charging and represent a previously non-existent type of a load. Furthermore, the ability to store and use electrical energy by EVs makes them very flexible devices to shape the consumption time-series according to the needs of the connected loads.

In this thesis, we define four interconnected challenges in the smart grid that we address to obtain a framework. The major criteria addressed for all solutions are 1) scalability: the power grid is already a vast networked system and the transition to smart grid is increasing this network in both size and control granularity, making scalability a crucial property that must be considered in all solutions. 2) optimality: our solutions are based on formal mathematical optimization problems and have provably correct solutions. Due to the vast scale of smart grid, any suboptimal operation should be avoided for limiting losses to a minimum. 3) low computational complexity: the smart grid network works in real-time with a common operating frequency of 50Hz or 60Hz, where the timeliness of decisions is of utmost importance for both decision accuracy and grid stability. Any delay due to computational purposes of a solution should be minimized and bounded. 4) distributed operation: a solution in distributed form enhances scalability, uses local resources for the parallelization of computation, improves privacy by keeping information locally and reduces communication requirements.

An example scenario describing the connections is shown in Figure 1.1. The example consists of three residential buildings that have a PV, electric vehicle (EV) and batteries, along with a PV farm supported with a battery on the same distribution circuit. PVs have volatile generation and need related real-time sensor information, such as temperature or global horizontal irradiance (GHI) in that area, to produce an accurate forecast. Using this forecast and information on the loads, the batteries can coordinate among themselves to devise the optimal control strategy. We must make sure that the result of this actuation should not endanger

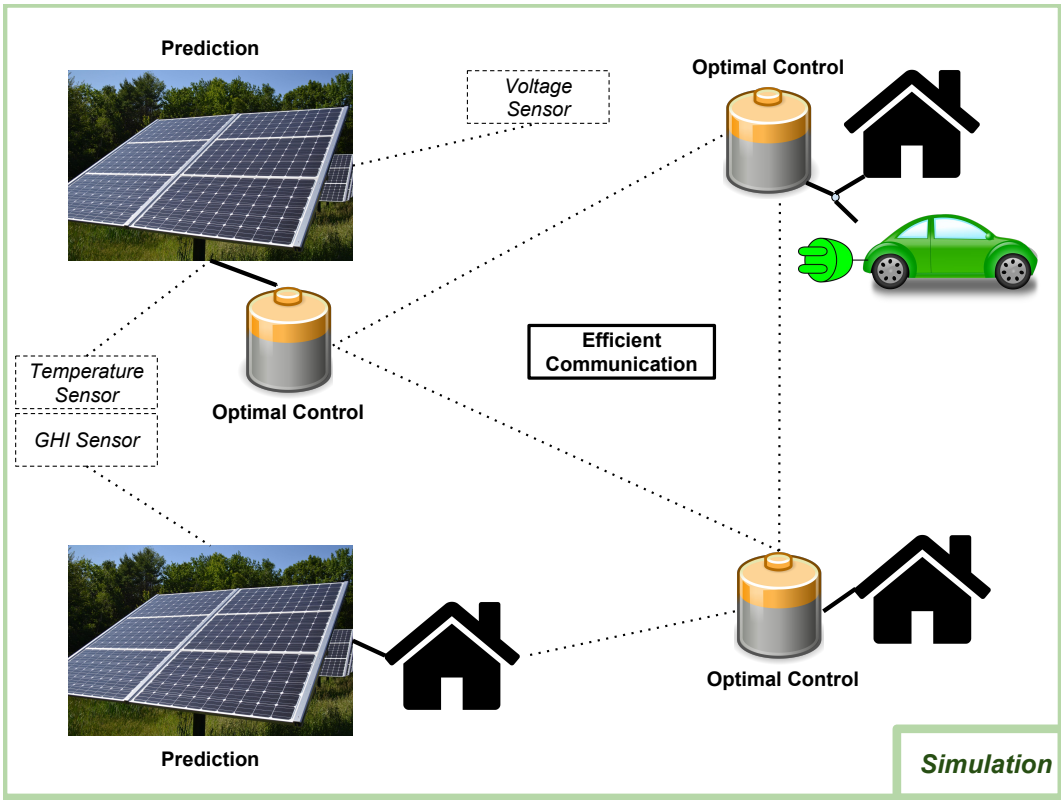

Figure 1.1: The interconnected challenges of distributed generation prediction, distributed control, dynamic smart grid simulation and efficient communication that are addressed in these thesis.

the stability of the grid. This is possible through the use of a fast grid simulator capable of incorporating dynamic actuation decisions and providing grid stability information. Finally, real-time sensor information and control negotiation signals must be efficiently transmitted, while considering the result of their transmission is an addition to the spectrum congestion around them. The transmission instances must be coordinated in a congestion-aware manner, while optimizing for the individual requirements of individual data streams. The challenges are described as:

Distributed Generation and Prediction: One of the biggest challenges for the integration of renewable resources is their hard-to-predict and rapidly varying nature. We propose a fast and accurate forecasting algorithm to obtain hours-ahead predictions. Obtaining accurate prediction is crucial for

planning and design. Most importantly, model predictive control algorithms cannot work without a quality prediction.

Distributed Control of Energy Storage Devices: The addition of EVs and advances in battery technology enable the use of energy storage devices for power consumption shaping purposes. However, the nonlinear charge characteristics and, nonlinear degradation and lifetime of batteries introduce a major research challenge to obtain an optimal and low-complexity control algorithm that works in distributed fashion for scalability.

Distributed Control Simulation in the Smart Grid: The addition of distributed control into smart grid makes the previous generation of grid simulators infeasible as the main focus was on power-flow and involved static (snap-shot) or semi-static (predetermined time-series) scenarios. There is a need for a smart grid simulator that can incorporate distributed control algorithms through a dynamic, unified interface and synchronized time scale.

Efficient Communication in the Smart Grid: Distributed control uses negotiation, prediction and current sensor information to make decisions. Furthermore, prediction algorithms need timely sensor information to train and compute predictions. The main challenge is to bring the communication needs of multiple heterogeneous types of sensors together in a distributed and scalable fashion, closing the loop between the other three challenges.

1.1 Forecasting of Renewable Energy Sources

One of the major leading sources for the speed-up of the transition towards the smart grid is the higher availability, lower costs and climate change based policies and incentives for renewable energy resources. However, integration of renewable energy and possibly replacing the conventional resources brings in major challenges due to the fundamental differences between the conventional resources of coal, natural gas and nuclear power, and renewable resources of solar and wind energy: 1) conventional generators have a higher efficiency as their size gets larger,

resulting in centralized generation with large generators to be more efficient. Renewable energy is implemented in distributed fashion. 2) the power output of conventional generators is deterministic under normal operation and can be adjusted with known dynamics for supply-demand balancing. Renewable resources are stochastic in nature and vary vastly due to weather conditions. As an example, a single cloud passing over a PV is enough to reduce the power output drastically and partial clouds result in rapid variations of the output power in a short amount of time. 3) conventional generators consist of large mechanical spinning parts that have a mechanical inertia. If the generation process is interrupted due to a problem or the supply-demand is not balanced, the inertia results in a slow and smooth decrease in performance of the system, allowing enough time for backup systems to respond. Renewable resources, if not supplied with any kind of energy storage, don't have this type of inertia.

One way to mitigate the disadvantages of renewable resources is to obtain fast and accurate predictions that have smaller uncertainty and enable preparing for any future events. Previous works either focus on numerical weather prediction [8, 9] that models the climate and internal processes numerically or on statistical methods such as artificial neural networks or statistical learning [10, 11]. Numerical weather prediction is successful in incorporating current conditions to obtain long-term forecasts, however their computational complexity is very high and requires specialized hardware for solution. Statistical methods don't require built-in models and can train based on history, yet they miss the connection to the current conditions. We focus on a hybrid method, analog forecasting [12], that compares the similarity between past predictions and current conditions, and utilizes the most similar observations from the past for a prediction. Instead of defining a predetermined similarity function, such as euclidean distance, we define a general function form, expand it using Tailor expansion and train the coefficients using past data. Our solution is Taylor Expanded Solar Analog Forecasting (TESLA) algorithm for solar prediction. TESLA is a low-complexity algorithm, where the order of complexity can be adjusted in trade-off with accuracy. TESLA can obtain forecasts for different time horizons, including the most common horizons of

hour-ahead and day-ahead forecasts. It can use the current conditions as an input to improve accuracy. Compared to the state-of-the-art solutions using Numerical Weather Prediction and Persistence, TESLA reduces the root mean square error (RMSE) of prediction by up to 50% compared to numerical weather prediction, while providing a year long forecasting in less than a second. TESLA can adapt using incremental training data, without the need for complete re-training and can work for different locations and climate conditions. This is presented in Chapter 2.

1.2 Optimal Control of Energy Storage Devices

The biggest challenge for the integration of renewable resources is not the lack of generation but the inability to dispatch energy on demand, as these resources have diurnal cycles resulting from natural climate cycles [13, 14, 15, 16, 17, 18]. Energy storage devices are very flexible devices that can act as a load or generator, enabling the on demand dispatch of energy. Standalone batteries or EVs through vehicle-to-grid (V2G) systems can charge to store energy during maximum generation intervals and discharge to dispatch energy, when the resources are intermittent. This also enables the smoothing of rapid variations, improving the stability of the grid. The reason for the slow adaptation of batteries is due to their high initial cost and nonlinear degradation properties. In order to mitigate the disadvantages, the control algorithm must 1) consider degradation as part of the problem to justify the initial capital investment, 2) consider the nonlinear nature of the batteries since batteries are devices with memory and linearization errors can accumulate to decrease the cost reduction performance, 3) support a low-complexity and distributed solution form to be usable for on-demand dispatch and scalable in the grid.

Recent works model the batteries linearly to reduce the solution complexity [19, 20, 21, 22, 23]. Linear models assume a linear connection between the power output and the state of charge of the battery. Furthermore, linear models neglect degradation completely. However, batteries are inherently nonlinear [24, 25]. Us-

ing a linear model can result significant errors [26]. There are multiple works on the nonlinear modeling of batteries [27, 28, 29, 30, 31], yet the scope of research is limited to battery modeling to determine accurate lifetime predictions, rather than measuring the degradation and deviation of batteries during smart grid applications. We developed the Optimal Distributed Nonlinear Battery Control (ODNBC) algorithm that minimizes any given convex cost function using multiple batteries. Compared to the state-of-the-art battery solutions, ODNBC uses a high accuracy nonlinear battery model for state of charge and state of health calculations with less than 2% error. ODNBC works in distributed fashion for multiple batteries and has exponential convergence properties. In case studies, ODNBC reduces the cost of an actual building by up to 50%, a 30% improvement over the load following heuristic. Linear optimal algorithms that calculate the solution using linear models result in up to 250% sub-optimality compared to our nonlinear optimal solution. We show that this is due to premature state transitions as a direct result of using linear models. This is presented in Chapter 3.

1.3 Distributed Control Simulation in the Grid

The transition towards distributed generation and the increasing trend in smart home automation are causing the control aspect of the smart grid to shift towards the end nodes, requiring distributed control solutions. Distributed control algorithms negotiate and iterate among each other to obtain a solution consensus dynamically. However, the convergence towards a consensus might have unexpected effects on the rest of the grid and vice versa. As an example, the iteration path to a consensus solution can require extreme values at the same time for multiple devices, which may lead to unstable conditions for a short amount of time. In contrast, an unstable event can incur higher costs to the devices resulting in a different steady state consensus point, which would have been ignored if the grid were not considered. Furthermore, it is not enough to show that a distributed control algorithm is performing well locally. It must also be tested with the context of the larger grid with multiple instances of the algorithm. Moreover, there

is currently no industrial standard for control algorithms, which means that a scenario of multiple heterogeneous control algorithms that are not negotiating with each other being present in the grid concurrently is very likely. For example, in the case of heating, ventilating and air conditioning (HVAC) control, there are multiple algorithms in the literature [32][33][34]. Multiple instances of the same algorithm can coordinate with each other, but not with the others. In such a case, there is no guarantee for convergence and can lead to instability. Another example would be the use of linear battery control algorithms that result in high discharge currents [26]. In an ideal setting, this wouldn't affect the solution, but in the grid it would cause instabilities due to sudden voltage drops. For all such scenarios, there is a need for a smart grid simulator to test how algorithms perform in the context of the grid and how they affect the stability of the grid.

Control algorithms in the literature focus on showing their performance on localized setup in rapid prototyping environments such as MATLAB. However, multiple instances must be tested in the grid environment to obtain the more realistic performance results and grid stability related information. Current simulators in the literature are not suitable for this purpose. Current solutions include open source simulators such as OpenDSS [35] and GridLab-D [36] or commercial products like RTDS [37] and Paladin Live [38]. These tools are very fast and efficient in providing the nonlinear power-flow solutions and harmonics analysis. These solutions focus on static or predetermined time-series scenarios and modeling dynamically changing scenarios and dynamic control algorithms become infeasible or constrained by their libraries. We develop Smart Grid Swarm Simulator (S^2Sim), a novel smart grid simulator that enables the co-simulation of distributed control algorithms without constraints and provides a unified communication interface for heterogeneous control algorithms. The unified communication interface uses transfer control protocol (TCP) and Internet protocol (IP) that enables any control algorithm to connect to the simulation over the Internet. This increases parallel computation and co-simulation, and makes the control implementation independent of the simulator. The implementation can be in any language or platform, can represent a real device, can be a simulation or past recorded data. Our simulator

receives data from all controllers, synchronizes them in time, calculates the power flow and sends an adjustable feedback signal to obtain a closed control loop. In our case-studies, we show how distributed control algorithms that perform well in isolated conditions can in fact lead to unstable conditions when they are simulated in the context of the grid. We provide multiple feedback functions, such as time of use pricing or voltage deviation based adaptive pricing. This is presented in Chapter 4.

1.4 Efficient Communication in the Smart Grid

Smart grid is transitioning from a centralized structure to a distributed one. This transition includes the conversion from centralized control to distributed control. Distributed control algorithms, if the control problem is not completely separable, must negotiate and iterate in a timely manner to have a consensus solution. Furthermore, distributed generation requires distributed observation points for stability tracking purposes. Moreover, the prediction of generation needs sensor information from multiple sources to obtain an accurate forecast. Finally, the smart grid is part of a bigger research area, the Internet of Things (IoT), underlining the necessity for scalable and congestion-aware communication. All of these aspects require an efficient communication protocol that considers the different requirements of heterogeneous types of sensors, the limited bandwidth of wireless sensor networks and the increasing levels of congestion, and the physical power and energy constraints of sensor devices. Packet aggregation is a possible solution for these challenges, where multiple packets are aggregated together into a single transmission to save energy, increase spectral efficiency and decrease congestion [39]. However, it is of utmost importance to determine the waiting time for aggregation, since a long waiting time can increase energy savings and spectral efficiency but would result in the expiration and loss of freshness of the data being aggregated. On the opposite side, transmitting whenever a data is available enables quick transmission but is costly in terms of energy consumption, spectral inefficiency and congestion.

Previous solutions in the literature focus on sensor networks with a single type of application that have homogeneous requirements across the network [40, 41, 42, 43]. However, in reality wireless networks can have multiple different types of sensor with different requirements and constraints. As an example, solar forecasting requires temperature sensor data as an input, which is not time critical for continuous operation, however the control iteration between batteries are time sensitive and their delay can cause higher operational costs and endanger grid stability in the worst case. Including the IoT on the same communication platform would increase the types of sensors even more, such as localization and voltage sensors for smart homes and commuter information for EV networks. We developed an Optimal Packet Aggregation algorithm for wireless networks. Our solution defines individual priority functions for each data stream. These functions enable fine-tuned adjustment of what each data stream requires and prioritizes, such as expiration minimization or freshness maximization. Our solution obtains the optimal aggregation time solution with low computational complexity and in a distributed fashion. The optimal solution considers the effects of congestion by and to the neighboring nodes by each node in a global setting and adapts to changing environmental conditions. This is presented in Chapter 5.

Chapter 2

TESLA: Taylor Expanded Solar Analog Forecasting

2.1 Introduction

History repeats itself. Weather is a continuous, data-intensive, multidimensional, dynamic and chaotic process, and these properties make weather forecasting a formidable challenge. With the increasing percentage of renewable energy penetration within the Smart Grid, forecasting the weather accurately gained even more importance. Even now, a group of Smart Grid control algorithms, battery optimization solutions [44], day ahead energy market negotiations and residential energy management systems [45] already rely on the availability of an accurate forecast. High errors in generation forecasts have the danger of disturbing the supply-demand stability within the Smart Grid, which will have to be compensated by expensive generators or in the worst case may even lead to frequency drop and instabilities.

Weather forecasts provide critical information about future weather. There are a wide range of techniques involved in weather forecasting from basic approaches to highly complex computerized models [11]. It is difficult to obtain an accurate result from the weather and solar predictions. Accurate forecasting of solar irradiance is essential for the efficient operation of solar thermal power plants,

energy markets, and the widespread implementation of solar photovoltaic technology. Numerical weather prediction (NWP) is generally the most accurate tool for forecasting solar irradiation several hours in advance [8]. The techniques used in solar forecasting can be categorized as dynamical and empirical methods. Furthermore, NWPs provide another alternative to a national or global scale ground based monitoring network [46]. NWP models provide a comprehensive and physically-based state-of-the-art description of the atmosphere and its interactions with the Earth surface [9]. But, these methods are very computation intensive and require both time and computation resources. In order to enhance the forecast accuracy, there are refining techniques [47].

Most weather prediction systems use a combination of empirical and dynamical techniques. However, a little attention has been paid to the use of artificial neural networks (ANN) in weather forecasting [10, 11]. Since the late 1990s ANNs have seen increased application in the field of solar forecasting [46]. ANNs provide a methodology for solving many types of non-linear problems which are difficult to solve by traditional techniques [48]. Furthermore, ANN modeling offers improved non-linear approximation performance and provides an alternative approach to physical modeling for irradiance data when enough historical data is available.

Another family of methods is the analog method family. It relies on this fact that tomorrow has already happened in the past. In [49], the authors also show that the different applications of analog method usage in their study. They used different types of k-methods in their studies, which gives a good idea on the accuracy and the errors of the methods. In addition to this Hacker [50] also compares the different types of analogue approaches in their studies by indicating the inclusion of model diversity showing an improvement in terms of reliability and the statistical consistency in an analogue method approach study. Analog methods have been applied and tested for forecasting increasingly. Abdel-Aal shows the effect of using different training sets to train a network system [51].

In this work, we analyze the fundamental pieces of the analog forecast method family and show that the distance, which describes the similarity between two analogues is very important for a good forecast. First, we propose an extension

to the euclidean distance based analog method, then generalize the idea to construct a new method called Taylor Expanded Solar Analog (TESLA) forecasting. We show through case studies that we can even beat the persistence method by up-to 50% in terms of solar irradiance root mean square error (RMSE).

2.2 Methodology

Before we begin with the proposed method, we start by explaining the motivation for searching for a better forecasting algorithm by going over the drawbacks of some of the algorithms frequently used in the literature.

2.2.1 Euclidean Distance Analog Method

The analog forecasting method relies upon the fact that the history consists of recurrences. In other words, the future may have already happened in the past. In order to establish a connection between the future and the past, we first need a rough forecast product of the future and a distance of this forecast to the multiple forecast points in the past. This metric will describe how much the forecasted day is **similar** to the days that have happened in the past.

The forecasts produced in the past are grouped under the name of *ensembles*. Each ensemble has also an *observation* associated with it, which is the solar irradiance observed at the time of the ensemble, measured by weather stations. The euclidean distance analog method simply measures the euclidean distance between the forecast and each ensemble and weighs the observations inversely proportional to the distance. We can write this algebraically. First, we need to define the variable names that are going to be used throughout this thesis.

$e_{i,j}$**:** The forecast product has multiple outputs, typically forecasting the states of the weather like temperature at various atmosphere heights, relative humidity or wind speeds. This variable is the j^{th} variable output of the i^{th} hourly forecast ensemble.

o_i**:** The observed/measured solar irradiance associated with the i^{th} forecast ensemble.

$f_{k,j}$**:** The f variable defines the rough forecast of the desired future, thus this variable defines the j^{th} variable output of the k^{th} hour future forecast product.

Figure 2.1 shows an example construction to clarify the concept and the timing of the variables.

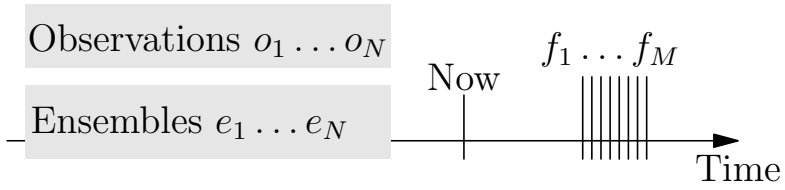

Figure 2.1: An example timeline showing the construction of the Analog Forecasting method family.

Using the defined variables, we can define the euclidean distance method algebraically. The euclidean distance in an N_j dimension universe is defined as:

$$d(x, y) = \sqrt{\sum_{j=1}^{N_j} (x_j - y_j)^2} \tag{2.1}$$

Applying this distance to our i^{th} ensemble and k^{th} forecast:

$$d_{i,k} = \sqrt{\sum_{j=1}^{N_j} (e_{i,j} - f_{k,j})^2} \tag{2.2}$$

The analog forecast output can be defined by weighing the observations inversely proportional to the distance:

$$a_k = \frac{\sum_{i=1}^{N_i} \frac{o_i}{d_{i,k}}}{\sum_{i=1}^{N_i} \frac{1}{d_{i,k}}} \tag{2.3}$$

Note that this method has its drawbacks. First of all, the method relies on the fact that if the distance of two forecast products is small then their observations

should be close to each other, in other words they will be similar days in terms of weather.

To check how well the euclidean distance metric performs, we have constructed an ensemble set of 16343 hours (roughly 15 months). The set is obtained from NOMADS, North American Mesoscale (NAM) [52] forecast data, which consists of 36 hour daily forecasts. We have selected 38 variables from the forecasts for distance calculation. We have sorted all ensembles and corresponding observations in ascending order of observations. Then we calculated the distance of all ensembles to three selected ensembles, 2000, 10000 and 13000 corresponding to a night and two mid day indices. The resulting plots are given in Figure 2.2.

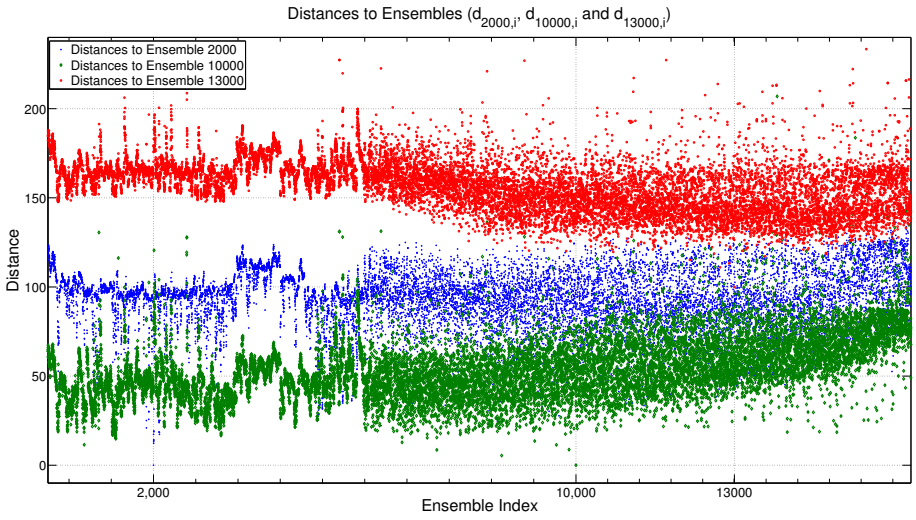

Figure 2.2: The euclidean distance of all sorted ensembles to the Ensemble 2000 (blue), Ensemble 10000 (green) and Ensemble 13000 (red).

In the ideal case, we would expect the ensembles close to the selected ensembles to have a small distance, since their observations are close. As the indices go far from the selected ensembles, the distance metric should increase since the similarity between observations will be lost completely. The figure shows that the real case is very far from the expected ideal case. There is a big band of noise in the figures, furthermore the expected increase in distance is not observed around the selected indices. This non-ideality of the distance metric causes a big problem on the accuracy of the forecasts, as will be shown in the performance section.

Another drawback of this method is that it uses a linear combination of the variables to calculate the distance metric, but in reality this may not be the case. We would only have a linear approximation of the ideal distance.

A final remark on the method is that the weight of each parameter on the distance metric is assumed to be the same, creating a perfect hyper-sphere. Different parameters may have different weighted effects on the distance, creating a hyper-ellipse rather than a hyper-sphere. The application of this idea is explained in the next section as an extension to the euclidean distance method.

2.2.2 Weighted Euclidean Distance Analog Method

In the previous section, we have shown that the distance metric to measure the similarities between ensembles is not a completely reliable metric. In this section, we propose an extension to the euclidean distance analog method by introducing linear weights to incorporate different effects of forecast variables into the distance metric. We can show this algebraically as:

$$dW_{i,k} := \sqrt{\sum_{j=1}^{N_j} w_j \left(e_{i,j} - f_{k,j}\right)^2} \tag{2.4}$$

This new parameter introduces the problem of determining its value. In order to find the optimal weight values, we need to train the system with known outcomes and optimize the system to best fit the expected outcomes. In order to formulate the optimization problem, we need to introduce two more variables.

$t_{k,j}$**:** We use a set of ensembles and their associated observations to train the system weights. This variable defines the j^{th} parameter of the k^{th} training ensemble.

γ_k**:** This variable defines the observation associated with the k^{th} training ensemble.

Given that we have our training ensembles and their associated observations, we can define our optimization problem. The main objective is to maximize the accuracy of our forecasts. We define the accuracy of multiple forecasts as the Root Mean Square Error (RMSE). The RMSE for the training ensembles is given

in the following equation.

$$RMSE = \sqrt{\frac{1}{N_k}\sum_{k=1}^{N_k}\left(\frac{\sum_{i=1}^{N_i}\frac{o_i}{dWT_{i,k}}}{\sum_{i=1}^{N_i}\frac{1}{dWT_{i,k}}} - \gamma_k\right)^2} \tag{2.5}$$

where

$$dWT_{i,k} := \sqrt{\sum_{j=1}^{N_j} w_j \left(e_{i,j} - t_{k,j}\right)^2} \tag{2.6}$$

This equation is optimized using the Optimization Toolbox in MATLAB. Using 14000 training hours we have obtained the optimal weights and tested our new distance metric on the example ensembles from the previous section. The distances are shown in Figure 2.3.

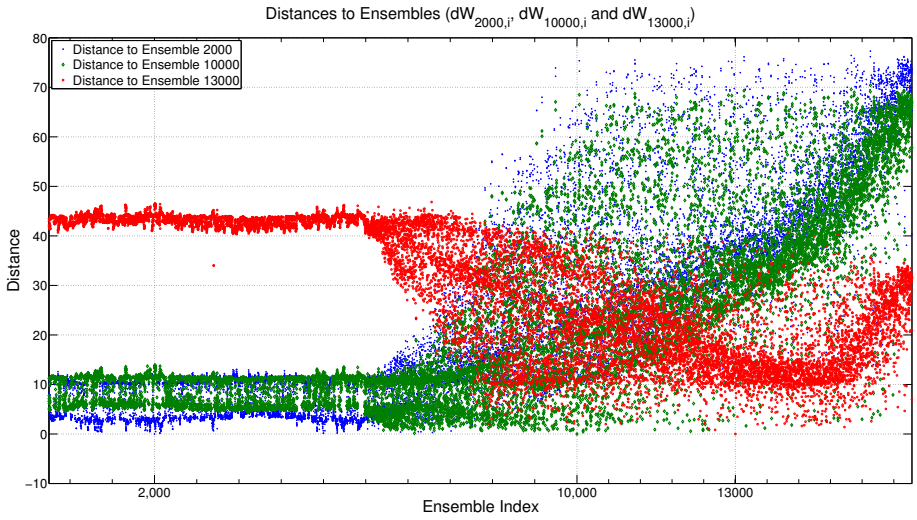

Figure 2.3: The weighted euclidean distance of all sorted ensembles to the Ensemble 2000 (blue), Ensemble 10000 (green) and Ensemble 13000 (red).

It can be clearly seen that the introduction of parameter weights has improved the shapes of the distance metrics to the expected ideal case. For Ensemble 13000, it can be seen that the distance to the closer points is small compared to the farther ensembles, constructing the convex shape that was desired. Although the general trends of the distances have improved, there is still too much noise in the

system that will lead to errors if not handled. In the performance section, we will show that the weighted distance method performs better than its uniform counterpart. In the next section, we define an even better solution, the main method presented in this thesis.

2.3 Taylor Expanded SoLar Analog Forecasting

In the previous section, we have shown that the distance metric showing the similarity between ensembles has problems and needs improvement, because the similarity is the heart of the analog forecasting method family. A second observation that we made in the previous section is that the similarity is calculated as a linear combination of the parameters, which in real life may not be the case. To better address these problems, we have changed our perspective fundamentally. Instead of using a *distance* metric, we introduce a new *similarity* metric. This metric is constructed as a function of two vectors, representing the two ensembles that we are comparing for similarity and outputs a similarity value. This can be formulated as:

$$s_{i,k} = S(\mathbf{e_i}, \mathbf{f_k}) \tag{2.7}$$

We aren't assuming anything regarding how this similarity function should be. Instead, we write the Taylor Expansion of the similarity function around $(\mathbf{0}, \mathbf{0})$.

$$\begin{aligned} S(\mathbf{x}, \mathbf{y}) &= S(\mathbf{0}, \mathbf{0}) + \sum_{j=1}^{N_j} \partial_{x_j} S(\mathbf{0}, \mathbf{0}) x_j + \sum_{j=1}^{N_j} \partial_{y_j} S(\mathbf{0}, \mathbf{0}) y_j \\ &+ \frac{1}{2!} \sum_{j1=1}^{N_j} \sum_{j2=1}^{N_j} \partial_{x_{j1}, x_{j2}} S(\mathbf{0}, \mathbf{0}) x_{j1} x_{j2} \\ &+ \sum_{j1=1}^{N_j} \sum_{j2=1}^{N_j} \partial_{x_{j1}, y_{j2}} S(\mathbf{0}, \mathbf{0}) x_{j1} y_{j2} \\ &+ \frac{1}{2!} \sum_{j1=1}^{N_j} \sum_{j2=1}^{N_j} \partial_{y_{j1}, y_{j2}} S(\mathbf{0}, \mathbf{0}) y_{j1} y_{j2} + \dots \end{aligned}$$

Note that within this expansion, we don't know any of the expansion constants. We can denote the unknowns as a_i, such that;

$$a_1 = S(\mathbf{0}, \mathbf{0}),\ a_{2,j} = \partial_{x_j} S(\mathbf{0}, \mathbf{0}),\ a_{3,j} \partial_{y_j} S(\mathbf{0}, \mathbf{0}), \ldots$$

This expression can be represented in matrix form. Representing all the unknown constants as the $\mathbf{A}$ vector and concatenating the variables into a single vector ψ, the equality becomes:

$$S(\mathbf{x}, \mathbf{y}) = \mathbf{A}^T \cdot \psi \tag{2.8}$$

We need to find a way to obtain the $\mathbf{A}$ parameters. We again use a forecast set for training purposes.

The similarity between an ensemble and a training forecast can be found as:

$$S(\mathbf{e_i}, \mathbf{t_k}) := s_{i,k} = \mathbf{A}^T \cdot \psi(\mathbf{e_i}, \mathbf{t_k})^T = \mathbf{A}^T \cdot \psi_{i,k} \tag{2.9}$$

Concatenating the similarity metrics for all ensembles horizontally:

$$\mathbf{S}(\mathbf{e}, \mathbf{t_k}) := \mathbf{s_k} = \mathbf{A}^T \cdot \begin{pmatrix} \psi_{1,k} & \cdots & \psi_{N_i,k} \end{pmatrix} := \mathbf{A}^T \cdot \mathbf{B_k} \tag{2.10}$$

Using the similarity metric, we define our analog forecast result as:

$$a_k = \sum_{i=1}^{N_i} o_i s_{i,k} = \mathbf{s_k} \cdot \mathbf{o} = \mathbf{A}^T \cdot \mathbf{B_k} \cdot \mathbf{o} \tag{2.11}$$

The final step is to define a vector $\mathbf{M_k} = \mathbf{B_k} \cdot \mathbf{o}$ and concatenating the vectors horizontally for all k values, converting it into a matrix $\mathbf{M}$. The forecast result vector is then simply found as:

$$\mathbf{a} = \mathbf{A}^T \cdot \mathbf{M} \tag{2.12}$$

We want in the ideal case $\mathbf{a} = \gamma$ to have an error-free forecast. In most cases the rank of $\mathbf{M}$ matrix is less than the size of the training set, N_t. This means that we have an under-defined system of equations. Since we are trying to minimize the RMSE, we can solve this system by Least Mean Squares Estimation (LMSE) to get the Taylor Expansion parameters.

After an initial training, the system will have learned the Taylor Expansion terms and use them to calculate the similarity metrics and give the TESLA Forecast

result. Note that, this method is already a super-set of both euclidean distance methods explained in the previous section. Furthermore, if the order of the Taylor Expansion is selected more than 1, the non-linear effects are also being added into the forecast, providing a better forecast.

Before moving into the performance section, an extension is provided in the next section to further improve the forecast results by trading off accuracy with the forecast horizon.

2.3.1 TESLA Forecasting with Moving Horizon Feedback Extension

TESLA method in the previous section uses a training dataset to determine its Taylor Expansion terms. Any new ensemble or observations during normal operation is not used, where it could have been used as an additional feedback parameter to increase the performance of the future forecasts.

An extension idea to TESLA is to use the N observations prior to the current forecast ensemble, as additional parameters to the ensemble parameters. Although it will be shown that the increased number of parameters by adding additional observations improves the performance, the trade-off that we are sacrificing is the forecast horizon. The forecast horizon of the TESLA method is upper-limited by the forecast horizon of the ensembles, denoted hereby by H. At any point in time, the closest observation that we have is the previous interval. The forecast interval that we are going to add our latest observation as an additional parameter, will also limit our forecast horizon. In other words, if we define the time between our latest observation and the forecast interval that we are going to add the observation to as the *delay*, denoted as D, our forecast horizon decreases to D. For an ensemble at t, N observations from time $(t-D)$ to $(t-D-N+1)$ are added as the N additional parameters.. These concepts are described in Figure 2.4.

The smaller the value of D, the better performance we will have. This extension allows the user to determine its own forecast horizon, D, according to the error requirements. Furthermore, we can also run this method H times and varying the delay from 1 to H. By selecting the last forecast at each run, we would

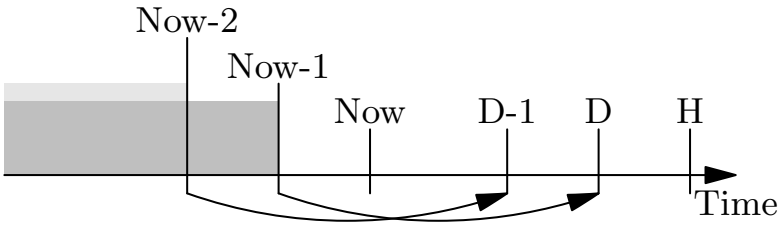

Figure 2.4: For a selected delay D, the closest N observations are used as additional parameters for the D^{th} forecast. The selected window is moved step-by-step and added to the previous forecast parameters.

both get the improvement without sacrificing the forecast horizon.

2.4 Performance

In order to test the performance of our solution, we have performed multiple case studies. This section describes the datasets that we have used and compares the performance of TESLA method with different methods from the literature.

2.4.1 Datasets

To construct our ensemble, training dataset and comparison datasets, we have used the 12 km, hourly NAM forecasts from September 2010 to January 2012, accounting for more than 15 months. The working site has been selected as San Diego, particularly the University of California, San Diego campus. The observation information has been used from Solar Anywhere data [53].

The NAM forecast has a 36 hour forecast horizon. We have extracted 38 parameters to be used within the ensembles. The parameters are Global Horizontal Irradiance, Planetary Boundary Layer height, surface heat flux, latent heat flux, total columnar cloud cover, dew point, surface temperature and pressure. In addition, the height, temperature, relative humidity, x and y components of wind speed have been used for barometric heights of 925, 850, 700, 500 and 200hPa, which correspond to the heights contours with the given pressure values, constituting the 38 parameters in each ensemble, that are believed to have an effect on the solar forecast result physically.

2.4.2 Weighted Euclidean Distance Performance

In order to test the performance of the weighted euclidean distance method, we have selected various sizes for our training set to understand its impact on the forecast performance. The error is calculated as the difference between the forecast product and the real solar irradiance observed on that hour shown in these equations:

$$RMSE = \sqrt{\frac{1}{N}\sum_{i=1}^{N}(forecast_i - observation_i)^2}$$

We have selected a separated training set and comparison set. This means that the weights are determined through the training set and then the error metrics are calculated over the non-overlapping comparison set. The size of the comparison set is selected as 8300 hours.

The results are compared to the NAM and 24 hour persistence forecast results. The RMSE results are shown in Figure 2.5. The 3-parameters in the figure are the highest correlation parameters with the observation. It can be clearly seen that the euclidean distance with 38 parameters performs the worst as expected from Section 2.2. The three parameters are selected as the highest correlation parameters of the ensembles with the observations, which performs very close to the weighted euclidean distance case. All methods still need improvement as they are much worse compared to the 24 hour persistence forecast method.

2.4.3 TESLA Forecasting Performance

We have compared TESLA against three methods. The first method is one of the state of the art methods, the analog method using Delle-Monache [12] distance as the similarity metric. The second method is also another state of the art method, the Persistence Forecast method. We have compared against both 24 hour and 1 hour persistence methods. Note that the forecast horizon of the 1 hour persistence is 1 hour. The third method for comparison is the unmodified NAM forecast. In order to make the comparison under same conditions, the 36-hour horizon of NAM forecasts are cropped to 24 hours.

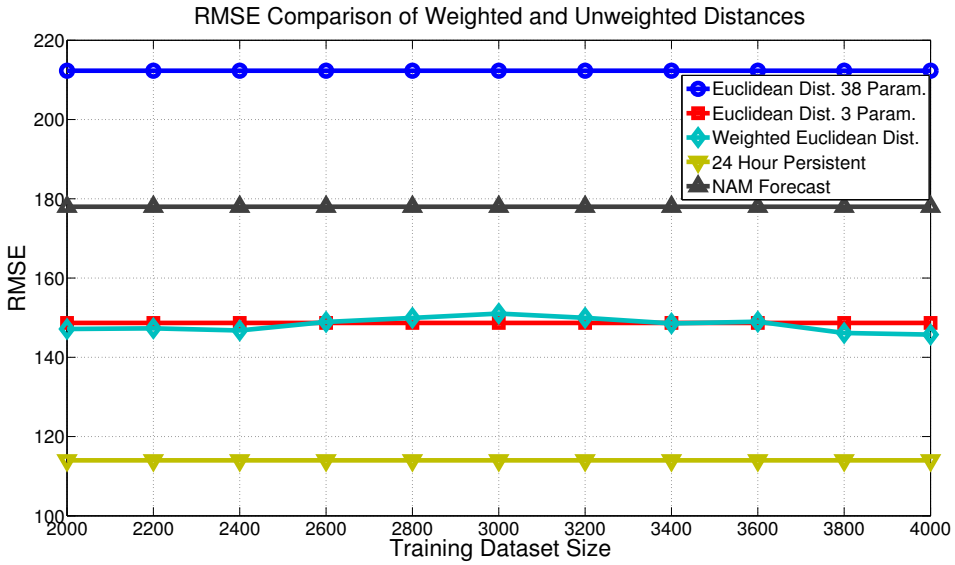

Figure 2.5: RMSE(W/m^2) comparison of weighted Euclidean distance forecast and other various methods with increasing training size. 38 parameter distance has a poor performance as expected from Section 2.2.

Two cases are considered to test the performance of TESLA. The first case uses a training size of 450 days. The size of the overlapping comparison set is varied from 20 to 460 days. The second case uses two completely separate sets for comparison and training/ensemble. 267 days are used for comparison. The training set size is varied from 20 to 200 days. The TESLA forecast parameters are selected as: First and second order Taylor expansion and First order Taylor expansion with the Moving Horizon Feedback extension with 24 previous observations: $D = 24, D = 1$ and delay varied from 1 to 24 hours and best forecasts are combined for 24 hour forecast horizon.

The results of the first case is shown in Figure 2.6. The figure shows that all TESLA methods have a better RMSE than the NAM and 24 persistence method. The second order expansion and all extension results have lower RMSEs than the Delle-Monache and 1 hour persistence methods. When the forecast horizon is decreased to 1 hour ahead, we can have RMSEs as low as $50W/m^2$.

The second case results are shown in Figure 2.7. TESLA requires training to construct its expansion constants. The figure shows that in order to get a

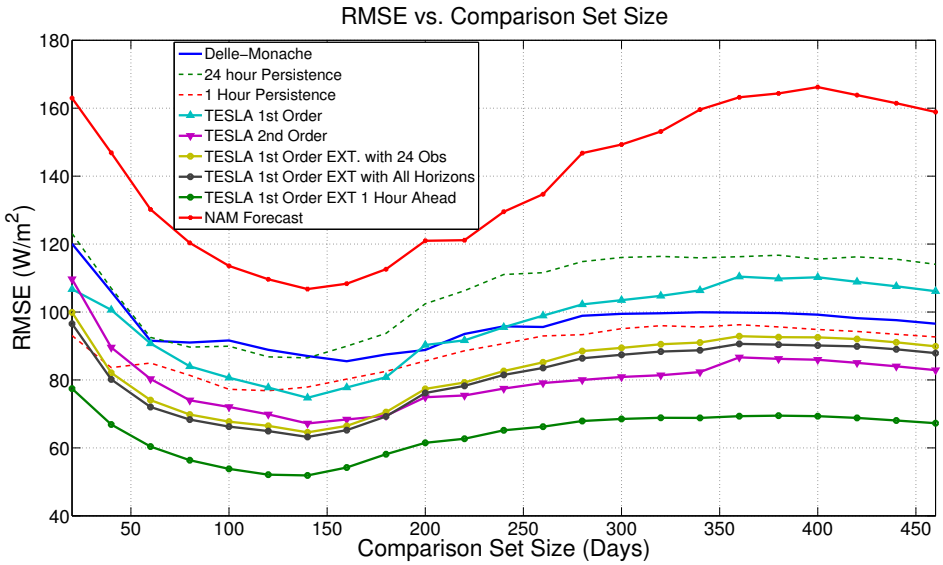

Figure 2.6: Comparison of multiple methods with TESLA. The initial decrease of RMSE is due to the fact that training size also increases with comparison set size and at least 60-80 days are required to settle training.

good forecast, we require 60-80 days of training data. When enough training is used, TESLA performs very close to the 1 hour persistence, while maintaining the 24 hour horizon. If the horizon is decreased to 1 hour ahead, TESLA performs 25% better than the 1 hour persistence method and 50% better than the 24 hour persistence.

2.4.4 Computational Complexity

TESLA computation consists of two stages, the initial training and the actual computation. Training stage, using least squares estimation has a complexity of $O(C^2N)$, where C is the number of parameters and N is the training size in our case. This stage is only performed once. The actual forecasting is a matrix-vector multiplication with a linear complexity of $O(C)$.

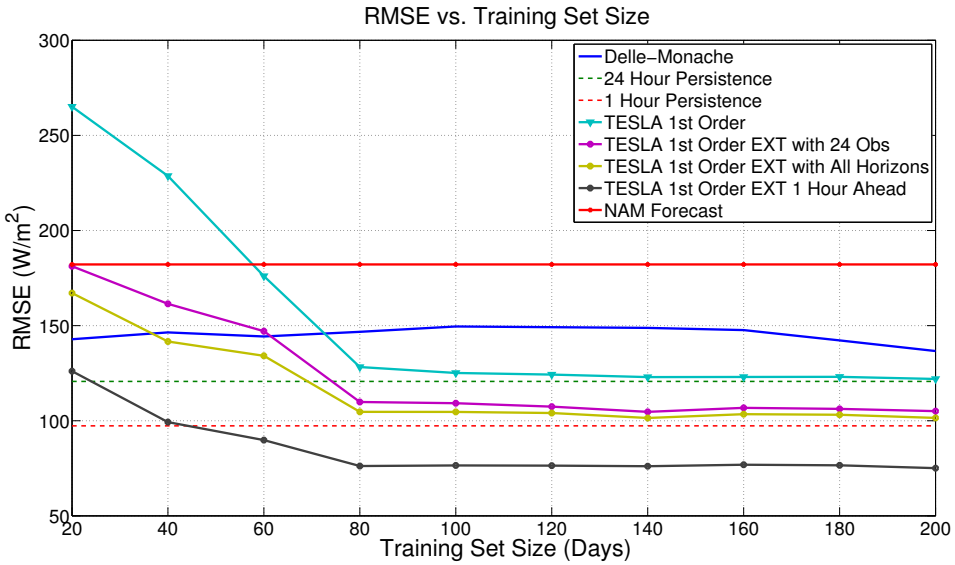

Figure 2.7: Comparison of methods with TESLA under different training sizes. A minimum number of 60-80 days of training is required to get normal results.

2.5 Conclusion

The integration of renewable energy sources into the grid causes many difficulties due to the stochastic and hard-to-predict nature of these resources. Model predictive control algorithms, commonly used for residential automation, energy storage control and demand response control, require high accuracy predictions to give high quality actuation decisions. In this chapter, we provide TESLA, a novel forecasting algorithm. TESLA has high accuracy and low computational complexity. In case studies, TESLA outperforms state of the art solutions by up to 50%. Its accuracy can be adjusted by changing the expansion order in trade-off with computational and training complexity. TESLA can provide a year worth of predictions in less than a second. It can use current conditions and past predictions as inputs to improve accuracy. In Chapter 3, we provide a model predictive control solution for optimal battery operation. Our solution, like any other model predictive control algorithm, requires high accuracy forecast to produce high quality actuation decisions, which is provided by TESLA. Furthermore, for TESLA to operate correctly, it requires current and past sensor information in a timely

manner for which an optimal data communication scheme is provided in Chapter 5.

This chapter contains material from Bengu Ozge Akyurek, Alper Sinan Akyurek, Jan Kleissl and Tajana Simunic Rosing, "TESLA: Taylor Expanded Solar Analog Forecasting", IEEE International Conference on Smart Grid Communications (SmartGridComm), 2014. The dissertation author was the primary investigator and second author of this paper.

Chapter 3

Optimal Distributed Nonlinear Battery Control

Energy storage plays a more important role than ever before, due to the transition to smart grid along with higher penetration of renewable resources. In Chapter 2, we presented an accurate forecast algorithm for solar energy. However, knowing when and how much energy is to be generated is not the only challenge in the integration of renewable resources. In fact, the dispatch of the generated energy at the right time and amount to balance the load requirements is the main problem. In this chapter, we describe our optimal nonlinear battery control algorithm that can handle multiple batteries connected to the grid with renewable generation in a distributed and cost-optimal fashion, while maintaining low complexity of $O(N^2)$. In contrast to the state-of-the-art, we use a high accuracy nonlinear battery model with 2% error. We present three distributed solutions: 1) Circular negotiation ring, providing convergence rates independent of number of batteries, 2) Mean circular negotiation ring, converging very quickly for a low number of batteries, 3) Bisection method has a convergence rate independent of battery capacities. We compare our algorithm to the state-of-the-art and show that we can decrease the utility cost of an actual building by up to 50% compared to the batteryless case, by 30% over the load-following heuristic and by 60% over a state-of-the-art optimal control algorithm designed using a linear battery model. For a constant load profile, optimal linear control incurs costs higher by 150% for MPC and by

250% for single trajectory solutions, as compared to our algorithm.

3.1 Introduction

Smart grid brings concepts of load automation and distributed generation. The current structure of the power grid is designed to have generation handled by large centralized generators and the consumption to be within predictable limits. Small mismatches in generation and consumption are balanced through multiple controllers and immediate changes are damped by the inertia of a spinning mechanical generator [54].

On the load side, automation transforms the predictable, average daily human behavior into a more complex function due to load shifting, two-way communication, pricing mechanisms [55] and demand response [56]. Renewable distributed generation brings new challenges with it. There are many forecasting algorithms for wind [57] and solar [1] energy, but errors are inevitable due to their highly varying nature.

Energy storage devices (ESDs) are used either as stand-alone batteries [58] or indirectly as a part of Electrical Vehicles (EVs), also known as Vehicle to Grid (V2G) [59]. The advantage of batteries is their flexibility to be used as either load or generator at adjustable power levels. The phase of the injected power can be adjusted to control the reactive power [60]. The power output can be adjusted for voltage stability [61]. Disadvantages of batteries include the high capital costs and the nonlinear characteristics that makes the profitability harder to predict [62].

In this thesis, we target the finite horizon optimal economic dispatch problem of how multiple inherently nonlinear batteries with nonlinear power and degradation characteristics connected to the grid should be used and controlled, such that the aggregate electricity cost of a distribution circuit, such as an energy sharing neighborhood or a microgrid, is minimized. We obtain the optimal timeseries solution of each battery's power consumption to minimize the aggregate costs under different electricity cost schemes. We consider both centralized and distributed

implementations. Our main contributions are: 1) We use a nonlinear battery model for lithium-ion batteries to describe charging/discharging and degradation characteristics and find a low-complexity optimal centralized solution for a generalized cost function using dynamic programming. The centralized solution is converted into distributed algorithms without any loss of optimality. The region of convergence and the convergence rates of the solutions are studied and verified. 2) We obtain 50% cost reduction compared to the batteryless case and 30% improvement over the state-of-the-art battery management technique of load-following. 3) We provide a comparison with an optimal solution based on a linear battery model, and show that if it were to be used on a realistic nonlinear battery, it would deviate by up to 60% in terms of cost reduction from the nonlinear optimal solution. For a constant profile, we show that this deviation can be higher than 150%. This is caused by the accumulation of model errors in time, resulting in premature constraint violations. 4) We provide a voltage stability study for a real neighborhood circuit, where we show that we can decrease the maximum voltage deviation by up to 45% using multiple batteries in the neighborhood, improving stability.

3.2 Related Work

Increased penetration of renewable resources with rapid variations and increased usage of electric vehicles shifted many studies towards energy storage systems and their control. We divide the literature into battery *control* and *modeling* the electrical behavior of different battery systems. There is a growing gap between the more complex and accurate battery models and, fast and optimal control strategies. Complex battery models try to model every detail of internal battery characteristics, making it practically impossible to implement feasible large scale control algorithms. On the other hand, large scale control algorithms oversimplify the battery models in order to obtain optimal solutions for battery control problems.

An important part of the smart grid is the integration of renewable distributed energy resources. There have been many studies on photo-voltaic (PV) [13,

14, 15, 16] and wind power [17, 18] in the recent years. The common conclusion is that the fast changing, hard to predict nature and the stability concerns of renewable energy sources need to be addressed in order for the renewable energy sources to become a larger fraction of the overall energy supply. A good solution is through the usage of energy storage devices.

3.2.1 Related Work on Battery Control

Batteries can smooth out the intermittent nature of the renewable sources and decrease the peak power usage of the system by charging at non-peak hours and discharging at peak hours [14]. Different kinds of energy storage techniques currently used in smart grid development is given in [63]; examples range from 34MW Sodium Sulfur down to 25kW residential area batteries. For a good summary of current energy storage uses, see [21, 22, 23]. In [64], the authors emphasize that it is possible to get three times more energy from the current wind farms if there was enough storage and a good control algorithm implementation. We divide the battery control algorithms into two categories: 1) heuristics for fast and suboptimal control; 2) optimal control algorithms with linear battery models. In [16], a price-based energy management solution is proposed for a system with PV and batteries minimizing the financial loss. In [14], ramp rate control is proposed, which holds the rate of change in PV power output within a stable boundary. In [65], the authors propose a 10kWh Lithium-Polymer battery with a home solar array with a heuristic control algorithm to decrease peak power usage and to store unused solar energy. They introduced different operation modes for the battery, which are determined depending on the solar power, required load power and current state of charge of the battery. The authors in [66] propose charging the battery during peak PV times and discharging at peak load times. They add a load forecasting method that uses six different inputs, such as temperature or season. They show that the system efficiency, in terms of the capital and environment cost functions, is increased by using load forecasting. In [67], authors determine (dis)charging regimes for frequency and voltage regulation based on a threshold.

In order to obtain the optimal solution, existing papers in the literature

focus on linearized battery models to simplify the constraints of the optimization problem. In [15], the authors use a Thermal Energy Storage (TES) in combination with Concentrating Solar Power (CSP) and propose an optimization problem to maximize its economic performance. CSP uses solar thermal energy to drive a heat engine and, the unused thermal energy is stored in TES for higher demands. The problem is solved by convex optimization to obtain the benefits in the long run, not in real-time. In [68], the authors model a PV system with battery and provide a dynamic programming solution that requires the complete knowledge of the future to the load flow problem. In [20], extensive work was done on the stability, peak shaving and capital cost performances of batteries using a linear battery model with quadratic programming (QP). The authors show for a residential area with PV that batteries increase the integration of PV, improve stability and simplify the over-voltage problem. In [69], the authors look into the problem of coordinated electric vehicle charging. The optimization problem is defined using a linear battery model and a quadratic cost function, representing the market balance and user discomfort. For further information on electric vehicles, please see the survey [70]. The authors of [71] propose an energy storage system for wind power smoothing. They utilize a model predictive control, where the battery is modeled linearly and the cost quadratic. The problem is converted into a quadratic programming problem and solved in a centralized manner. In [72], the authors approach the battery control problem from a DC microgrid perspective. The battery is modeled such that the nonlinear connection between the battery voltage and battery power is explicitly accurate. This connection is then leveraged in a hardware implementation to control the voltage levels at the utility substation transformer. A recent paper [19] considers a residential neighborhood with PV and linearly modeled batteries. The authors propose 4 different algorithms: a heuristic, centralized model predictive control (MPC), a decentralized MPC and a novel market maker distributed MPC. In all cases, the objective is to flatten out the aggregated power profile drawn from the grid.

3.2.2 Related Work on Battery Modeling

The simplest and most frequently used battery model is linear, where the change in battery state of charge (SoC) is a linear function of the battery's terminal power consumption. This is expressed in (3.1), where $\eta_C < 1$ is the battery efficiency constant for charging and $\eta_D = \eta_C^{-1} > 1$ for discharging. SU is the charge capacity of the battery.

$$\text{State of Charge}(t) = \text{SoC}(0) + \frac{\eta_{C,D}}{SU} \int_{t'=0}^{t} \text{Battery Power}(t')dt' \tag{3.1}$$

However, it is shown in [26] that the linear model has high errors for high discharge currents. Since renewable resources are intermittent and have rapid variations, high currents are frequent in grid related uses. There are many accurate and complex battery models proposed in the literature. See [24] for a comprehensive survey and [25] for an overview on different levels of degradation models.

In [73], the authors introduce a circuit based model. The paper introduces two degradation models based on temperature, self discharge and capacity fading, along with the consumption values. A Ph.D. thesis [74] models the battery (dis)charging memory effects and degradation effects based on the internal chemistry of lithium-ion batteries. Many different cycle testing cases are required to create the detailed model. The Riso Report [30] further discusses degradation effects based on the battery's consumption values in the form of analytic functions. The model is based on the internal chemistry of the battery. National Renewable Energy Laboratory (NREL) has a report [31] that models the battery from the perspectives of capacity degradation, depth of discharge memory effect and nonlinear charging and discharging characteristics. All properties are provided in the form of analytic functions making it easier to include within mathematical optimization. In [75], an analytic model is provided for degradation, including temperature, depth of discharge and state of charge based capacity fading. The authors use a linear SoC model along with the degradation model to solve an economic dispatch problem in [76]. In [77] the authors provide a state of charge and depth of discharge based degradation model. This is then used with a linear

state of charge model to solve an electric vehicle optimization problem. Effect of temperature is neglected based on steady operating temperature conditions.

As opposed to analytics based model, there are also degradation estimation methods based on the internal physics. In [27], [28], [78], [29] and [79], the unwanted chemical side reactions inside a lithium ion battery are modeled to obtain particle level aging estimation for very accurate degradation modeling during design time. These include recoverable and unrecoverable fading effects. In [80] a cell level optimization is described based on 2D physics based model. However, for larger scale and real time applications these models become very complex.

As seen in previous works, modeling the battery is a tradeoff between control quality and computational simplicity. We consider two aspects of the battery; state of charge and state of health (SoH). SoC defines the charge level of the battery. SoH is the measure of battery's charge holding capacity degradation. SoH declines slowly from 1 (brand new) to 0 (completely dead). We could include internal chemical reactions, physics and hysteresis effects to obtain a very accurate complex battery model, but the solution complexity increases by multiple orders of magnitude. In this thesis, we discard the hysteresis effects to keep the computational complexity to $O(N^2)$, but we retain the nonlinearity in SoC and SoH for our battery model, with an error of only 2% compared to empirical data (Section 3.5).

The widely used linear model, shown in (3.1), defines a linear SoC dependence that fails to represent the nonlinearity at high discharge currents, known as the Peukert's effect. Previous works show that linear models ignoring this effect can have an error as high as 43% [26]. Our SoC model, based on [26][74][3], is shown in (3.2), which states that higher discharge currents lead to an exponentially smaller effective capacity (SoC). α is the Peukert exponent and describes the exponential nonlinearity of the SoC relation.

$$\text{SoC}(t) = \text{SoC}(0) + \frac{1}{SU} \int_{t'=0}^{t} \text{Battery Power}^{\alpha}(t')dt', \alpha > 1 \tag{3.2}$$

As an example, if the battery output power is doubled, the amount of charge lost is increased by 2.2x for $\alpha = 1.15$. Solutions neglecting this effect can be misleadingly biased towards the use of high discharge currents causing accumulative control

errors of more than 150% as shown in Section 3.5. From the SoH perspective, the linear battery model assumes no degradation, keeping SoH= 1. A model without degradation would neglect aging and loss in capital value of the battery completely. Although degradation occurs slowly in time, the effect is noticeable over longer intervals. The SoH model for lithium-ion batteries used in this thesis is based on discharge throughput degradation [30][31] and, temperature and state of charge degradation [75][76], is shown in (3.3).

$$\begin{aligned} \text{SoH}(t) = SoH(0) \quad &- \quad \left[\int_{t'=0}^{t} \beta \exp\left(\gamma \text{ Battery Power}(t') - \theta\right) dt' \right] \\ &- \quad [\phi_1 SoC_{avg}(t) + \phi_2] \\ &- \quad \left[\int_{t'=0}^{t} \sigma_1 \exp\left(-\sigma_2 (T_{amb} + \sigma_3 |\text{Battery Power}|)^{-1}\right) dt' \right. \\ &+ \quad \left. \sigma_4(T_{amb}) \right] \end{aligned} \tag{3.3}$$

The first bracket expression states that the SoH degradation is exponential with the discharge throughput and linearly related to the discharge amount for a fixed time interval. β is the exponential scale of degradation, γ is the linear modification from discharge amount to Amperes-Hour (AH) discharge throughput and θ is the inefficiency coefficient for converting an amount of charge into actually stored charge. These parameters are device dependent and can be obtained from either battery data sheets that provide depth of discharge and cycle life results or from experimental results. More information on this process is provided in [31]. The second bracket expression is degradation due to the SoC level, modeled linearly in [75]. The coefficients in the original model have been converted to the time resolution used in this thesis and are obtained by fitting to the experimental measurements in [81]. The last bracket expression is the degradation due to temperature. It models the temperature change in the battery linearly through a thermal resistance and adjusts the degradation with respect to the current ambient temperature (T_{amb}). The coefficients in the original model [76] have been combined into ambient temperature dependent function of σ_4 and constants of $\sigma 1, \sigma_2$ and σ_3 for convenience in representation. These coefficients are obtained by fitting to the

experimental measurements in [81]. Throughout this thesis, we have used lithium-iron-phosphate battery chemistry for all model fits, case studies and, accuracy, complexity and performance results. The parameters are $\alpha = 1.15, \beta = 25, \gamma = 0.017, \theta = 19.9, \sigma_1 = 1.4 \times 10^{-4}, \sigma_2 = -75, \sigma_3 = 0.1, \phi_1 = -10^{-3}, \phi_2 = 10^{-8}$. However, the underlying optimal control solution is valid for different analytic battery models using the same control solution. The main purpose of this thesis is not to advance the literature on battery models, but rather provide a novel optimal nonlinear control strategy capable of accounting for nonlinear battery models to increase accuracy.

3.3 Optimal Centralized Battery Control

3.3.1 System Model

The main goal of this section is to solve the optimal economic dispatch problem, where we solve for the power level of each grid-connected battery at every time interval, such that the resulting power profiles are cost-optimal in a finite horizon. Our system model consists of a circuit with 3 types of devices: loads, distributed generators and batteries. We aggregate all consumption and generation values into a single time-series, but leave batteries separate to study the effect of nonlinear battery characteristics. We assume that the line-losses in our aggregate model are constant, resulting in a mean absolute error of only 0.8% as shown in Section 3.5. Aggregation is widely used in the literature for battery control problems [67][69][19]. Note that a battery at a particular location can physically consist of a battery bank, however we refer to them as batteries for simplicity in notation throughout this thesis.

There are two types of variable definitions arising from this division: Interval long and Interval end variables. Interval long variables define the total of a quantity within the defined time interval. As an example, the consumption of a battery (bat_t) is an interval long variable and represents the total energy consumption within the defined interval. Interval end variables define quantities for the end of time intervals. As an example, State of Charge (SoC) of a battery defines the

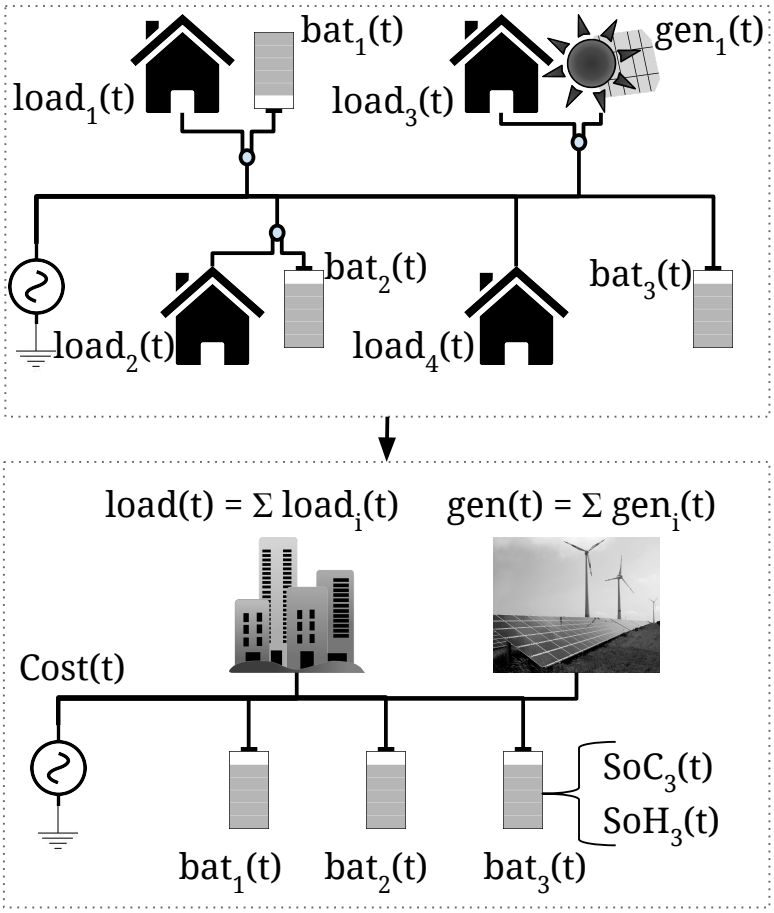

Figure 3.1: System model overview and load aggregation

Table 3.1: Symbols and Interval Types of the used variables

Symbol	Description	Interval Type
bat	Battery consumption	Interval Long
load	Load consumption	Interval Long
gen	Generator generation	Interval Long
SoC	State of Charge	Interval End
SoH	State of Health	Interval End

level of energy in a particular battery at the *end* of a specific time interval.

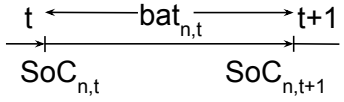

Figure 3.2: Interval Type example for Interval Long and Interval End variables

We begin our formulation by dividing time into discrete intervals of equal length. Any variable $x_{n,t}$ represents the quantity x for the n^{th} battery at the t^{th} time interval. *load*, *gen* and *bat* represent consumption, generation and battery consumption values, respectively. We jointly represent the total load and generation as: $p_t = \sum_k \text{load}_{k,t} + \sum_m \text{gen}_{m,t}$, since they are not controllable. The overview of the system is shown in Figure 3.1. Next, we define constraints to represent the battery's physical nature: 1) Power Limit: The discharge and charge power of a battery is limited, thus the energy consumption per interval is also bounded by $L_n \leq \text{bat}_{n,t} \leq U_n, \forall n, t$. 2) Charge Limit: SoC, is upper-limited by a value depending on the health of the battery. A lower limit is also enforced to avoid quick lifetime depletion, represented by $SL_n \leq SoC_{n,t} \leq SU_n, \forall n, t$. Each battery model has three state variables: energy consumption $\text{bat}_{n,t}$, state of charge $SoC_{n,t}$ and state of health $SoH_{n,t}$. Only $\text{bat}_{n,t}$ is independently controllable, whereas SoC and SoH dependent on the energy consumption of the current interval and their previous values, respectively. Our battery model uses the discrete time forms of (3.2) and (3.3):

$$SoC_{n,t+1} = SoC_{n,t} + soc_{n,t} = SoC_{n,t} + \frac{1}{SU_n}(\text{bat}_{n,t})^{\alpha_n} \tag{3.4}$$

$$\begin{aligned} SoH_{n,t+1} = SoH_{n,t} + soh_{n,t} = SoH_{n,t} - \beta_n e^{(\gamma_n |\text{bat}_{n,t}| - \theta_n)} - \frac{\phi_{n,1}}{SU_n}(\text{bat}_{n,t})^{\alpha_n} \\ -\phi_{n,2} - \sigma_{n,1} \exp\left(-\sigma_{n,2}(T_{amb} + \sigma_{n,3}|\text{bat}_{n,t}|)^{-1} + \sigma_{n,4}(T_{amb})\right) \end{aligned}$$

In practice SoH is a very slowly varying property. Thus in our solution, we update SoH only over long time intervals (24-hours for simulations).

3.3.2 Problem Formulation

We define two different cost factors: 1) a general memoryless function, C, that depends on the total consumption of the considered loads, generators and batteries (eg. the utility bill of a microgrid) 2) the cost of degradation, which is the loss in the invested capital value. We define K_n as the capital cost of battery n, thus the degradation of the battery at the end of t intervals is defined by: $\text{Deg}_{n,t} = K_n\,(SoH_{n,0} - SoH_{n,t})$. Consumption of any load k, generator m and

battery n at interval t is represented by $\text{load}_{k,t}$, $\text{gen}_{m,t}$ and $\text{bat}_{n,t}$, respectively, converted into a cost value by the C function. All intervals are summed up along with the degradation costs of each battery to get the total cost of the system. Since $\text{bat}_{n,t}$ is the only controllable parameter, The optimization problem for a finite interval of T is:

$$\min_{\forall b} \sum_{n} \text{Deg}_{n,T} + \sum_{t=1}^{T} C\left(p_t + \sum_{n} \text{bat}_{n,t}\right) \text{ s. t.}$$
$$SL_n \leq SU_n \cdot SoC_{n,t} \leq SU_n, \quad L_n \leq \text{bat}_{n,t} \leq U_n, \forall n \tag{3.5}$$

We solve this problem using its Lagrangian dual:

$$\mathcal{L} = \sum_{t=1}^{T} C\left(p_t + \sum_{n} \text{bat}_{n,t}\right) + \sum_{n} K_n \left(SoH_{n,0} - SoH_{n,T}\right)$$
$$+ \sum_{n}\sum_{t=1}^{T} A_{n,t}\left(SU_n SoC_{n,t} - SU_n\right) + B_{n,t}\left(SL_n - SU_n SoC_{n,t}\right)$$
$$+ \sum_{n}\sum_{t=1}^{T} X_{n,t}\left(\text{bat}_{n,t} - U_n\right) + Y_{n,t}\left(L_n - \text{bat}_{n,t}\right) \tag{3.6}$$

The KKT multipliers defined for the constraints are explained in Table 3.2, along with necessary conditions. Note that constraint functions are convex and we assume C to be also convex to satisfy Slater's condition for strong duality. Using $C_t = C\left(p_t + \sum_m \text{bat}_{m,t}\right)$, $Z_{n,t} = Y_{n,t} - X_{n,t}$ and $\lambda_{n,t} = B_{n,t} - A_{n,t}$, the optimal solution for any interval t is:

$$\frac{\partial C_t}{\partial \text{bat}_{n,t}} = Z_{n,t} + \sum_{t'=1}^{T} \lambda_{n,t'} SU_n \frac{\partial SoC_{n,t'}}{\partial \text{bat}_{n,t}} \tag{3.7}$$

Two major implications arise from this expression: 1) The equation is independent of past decisions and can be solved using dynamic programming: starting at T and iteratively solving back until $t = 1$. This property becomes the basis of our low complexity solution explained later in this section; 2) The right hand side of (3.7) depends on a *single* battery, but is equal to a function of the summation of *all* batteries. This becomes the basis of our distributed solution in Section 3.4.

Table 3.2: KKT Multiplier definitions and regions

Multiplier	Description	Condition
$A_{n,t}$	Battery n is full at time t	$A_{n,t} > 0$
$B_{n,t}$	Bat. n is empty at time t	$B_{n,t} > 0$
$X_{n,t}$	Bat. n charges with max. power at time t	$X_{n,t} > 0$
$Y_{n,t}$	Bat. n discharges with max. power at time t	$Y_{n,t} > 0$

3.3.3 Centralized Solution

Dynamic programming is a high complexity and generalized solution algorithm. To mitigate its disadvantages, we use the sparsity in our specific problem definition to obtain a new low-complexity solution. We use C', soc' and soh' to denote the derivatives of C, soc and soh in (3.4), and use them in (3.7):

$$C'_t = Z_{n,t} + SU_n soc'_{n,t} \sum_{t'=t+1}^{T} \lambda_{n,t'} = R_n(\text{bat}_{n,t}) \tag{3.8}$$

We define the right hand side of the equation as the individual constraint function since it only involves the individual properties of a battery. $Z_{n,t}$ and $\lambda_{n,t}$ variables dictate the state of the battery. A negative $Z_{n,t}$ means the battery is at its discharge power limit, whereas for a positive value the battery is at its charging power limit. $\lambda_{n,t}$ introduces three states: A negative $\lambda_{n,t}$ means that the battery is *empty*, a positive value means the battery is *full* and $\lambda_{n,t} = 0$ means the battery is transitioning from one state to another, thus *transient*. For a practical illustration, we present three figures for the battery state cases of empty, full and transient. In each figure, the x-axis represents the possible energy consumption values for a particular battery in the system, $\text{bat}_{n,t}$, while the y-axis represents the output values of the battery's individual constraint function, $R_n(\text{bat}_{n,t})$. Equation (3.8) states that every battery must match their individual constraint value, R_n, to the global cost value, C'_t. Note that the global cost is theoretically unbounded and must be matched by $R_n(\text{bat}_{n,t})$, which has a finite set of input values, $\text{bat}_{n,t}$. To match the unbounded cost with finite input, the battery must adjust its KKT multiplier values differently for each case as described below. Figure 3.3 represents the possible values of R_n when battery are **full**. The battery is bounded by $\text{bat}_{n,t} \in [L_n, 0]$, where $L_n = -155Wh$ represents the maximum discharge energy in this

example. For R_n to have a higher value beyond $R_n(\text{bat}_{n,t} = 0)$, $\lambda_{n,t} > 0$ must increase to compensate. Similarly, in order for R_n to have a lower value beyond $R_n(L_n)$, $Z_{n,t} < 0$ must decrease to compensate. This allows R_n to be unbounded, while $\text{bat}_{n,t}$ satisfies its bounds. If $\text{bat}_{n,t}$ is within its limits then the multipliers, $\lambda_{n,t} = Z_{n,t} = 0$. This region is defined as the transient region in the figures. For the **transient** state, the battery is bounded by $\text{bat}_{n,t} \in [L_n, U_n]$, where $U_n = 155Wh$ is the maximum charge energy. For $R_n > R_n(U_n)$, $Z_{n,t} > 0$ must increase and for $R_n < R_n(L_n)$, $Z_{n,t} < 0$ must decrease, as illustrated in Figure 3.4. Finally, for the **empty** state, the battery is bounded by $\text{bat}_{n,t} \in [0, U_n]$. For $R_n < R_n(0)$, $\lambda_{n,t} < 0$ must decrease and for $R_n > R_n(U_n)$, $Z_{n,t} > 0$ must increase, as illustrated in Figure 3.5. Batteries adjust their $bat_{n,t}$ values to match their R_n values to the cost value during transient regions and stop during full and empty regions. We

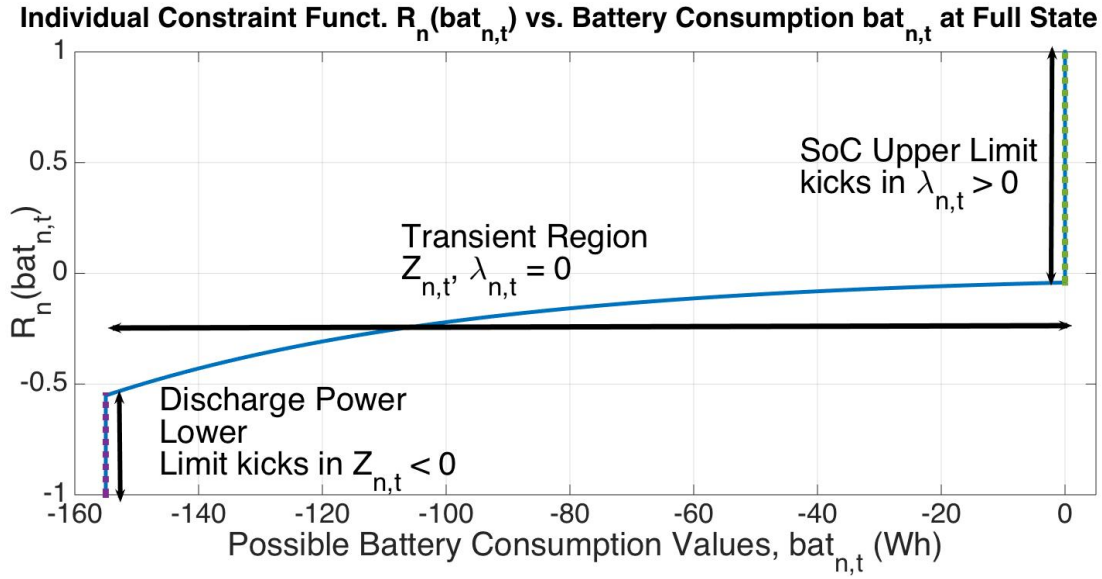

Figure 3.3: Optimal solutions at a single time step for a full battery. The battery cannot be charged further and the individual constraint function is dominated by $\lambda_{n,t}$ on the positive side and by $Z_{n,t}$ on the negative side.

normalize (3.8) by $soc'_{n,t}$ and take a finite difference at time t to obtain $\lambda_{n,t}$:

$$\lambda_{n,t} = \frac{C'_{t-1} - Z_{n,t-1}}{SU_n soc'_{n,t-1}} - \frac{C'_t - Z_{n,t}}{SU_n soc'_{n,t}}, \forall n, t \tag{3.9}$$

Equation (3.9) dictates the state of every battery. For a transient state ($\lambda = 0$), if the battery model were linear, the denominator would be 1, resulting in a solution

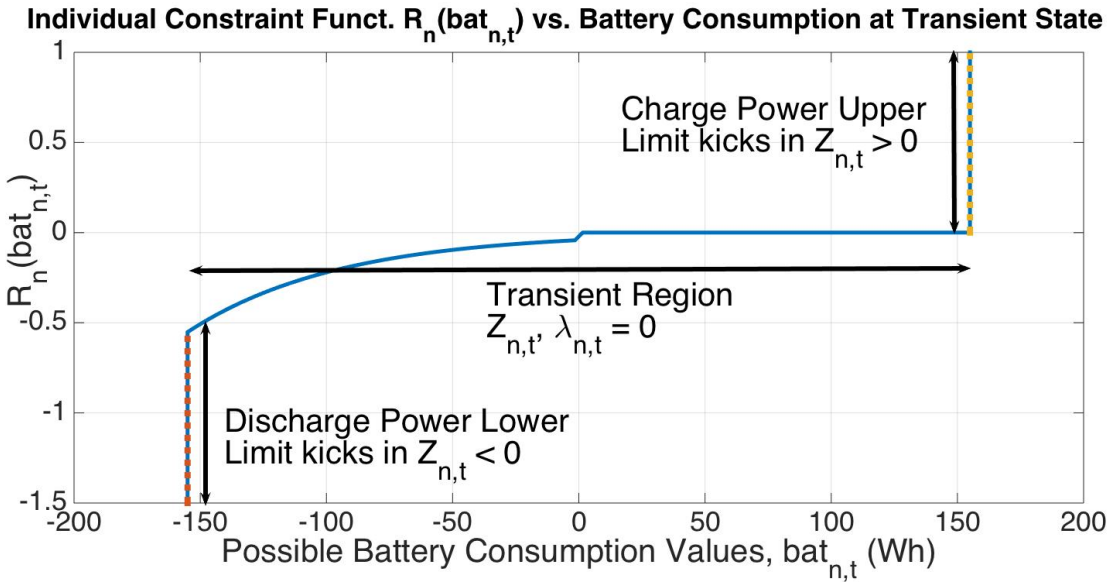

Figure 3.4: Optimal solutions at a single time step for a transient state battery, neither full nor empty. The battery is limited by the power limits and the individual constraint function is dominated by $Z_{n,t}$ on both limits.

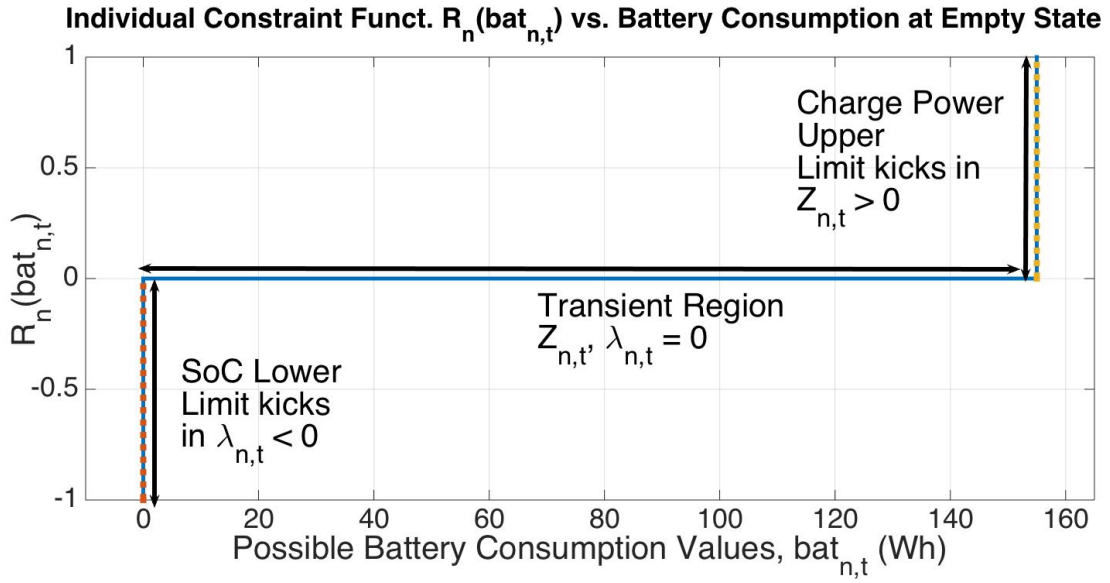

Figure 3.5: Optimal solutions at a single time step for an empty battery. The battery cannot be discharged further and the individual constraint function is dominated by $\lambda_{n,t}$ on the negative side and by $Z_{n,t}$ on the positive side.

of constant cost profile, whereas in our nonlinear solution, the *cost per used charge* is kept constant. For a full state ($\lambda > 0$), the cost per used charge should be

decreasing and for an empty state ($\lambda < 0$), increasing. Most importantly, this equality dictates that all batteries must have the same state at the same time for the optimal solution. We prove this by contradiction. Assume that battery n is in a transient state, while battery m is full:

$$\frac{C'_{t-1}}{SU_n soc'_{n,t-1}} = \frac{C'_t}{SU_n soc'_{n,t}} = \frac{C'_{t+1}}{SU_n soc'_{n,t+1}} \quad \text{transient} \tag{3.10}$$

$$\frac{C'_{t-1}}{SU_m soc'_{m,t-1}} > \frac{C'_t}{SU_m soc'_{m,t}} > \frac{C'_{t+1}}{SU_m soc'_{m,t+1}} \quad \text{full} \tag{3.11}$$

The derivative soc' is always positive since we cannot increase/decrease the charge by discharging/charging. Thus the sign of C' values must be the same. Furthermore, $b_{m,t} = 0$ since the battery is full at interval t and $soc'_{m,t} = 0$ as its result. This means that the signs of C'_{t-1} and C'_{t+1} are opposite, contradicting with the first equation, proving that they must be either full or transient at the same time. Extension of the proof for empty state and power limit cases follow similarly. Note that this is not a requirement for the system to operate, but rather the condition of optimal operation. Even though our solution considers heterogeneous battery chemistries and sizes, under real operating conditions, this requirement might not be satisfied due to hidden constraints or model irregularities, resulting in suboptimal operation. However, under our stated assumptions, the provided solutions are guaranteed to converge while satisfying this requirement. This concept has a similar precedent applied at a smaller scale in every cell of the battery [82] commonly used in battery management systems. Furthermore, even though this solution appears to neglect degradation at first, degradation is in fact inherently present within the SU_n capacity. As a battery degrades, SU_n decreases, resulting in the most degraded battery to be used less and the least degraded one to be used more. To show this, we created a scenario with two identical batteries, except the degradation coefficient of one of them has been increased by 25x. The long term solution is shown in Figure 3.6. It can be seen that the solution relies less on the fast degrading battery and more on the normal battery as the capacity fades in time.

Since each battery enters the full (empty) state at the same time, the transient region ($\lambda = 0$) must ensure that the accumulation of charge reaches full

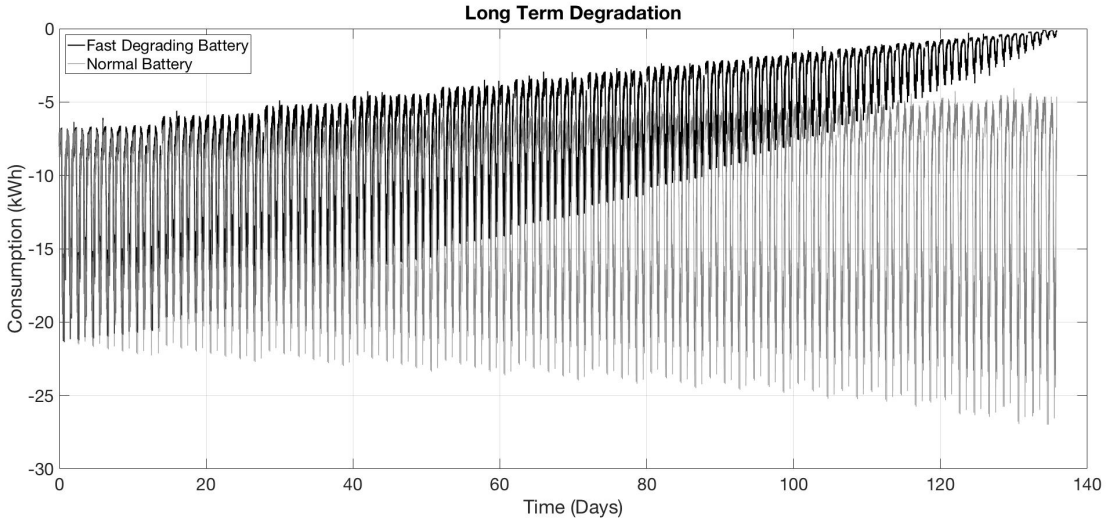

Figure 3.6: Long term degradation inherently present within the solution.

(empty) state at the same time as all batteries. The centralized solution directly follows this concept. If a battery is empty (full), the next step can either be charging (discharging) or staying idle at the same state. Once the battery controller leaves a full (empty) state, the optimal solution is obtained by $\lambda = 0$ until it reaches another full (empty) state. Since each full (empty) state is dictated by the KKT conditions, the solver must make sure that the end of a transient period ends up in a full (empty) state that satisfies the KKT conditions. So, at the beginning of a full (empty) state, the solver assumes virtually that a transient period can start. The transient period is calculated using (3.9) for all batteries at each time step, until an empty or full state is reached. The solver then checks whether the result satisfies the KKT conditions in (3.9). If it does, the initial assumption of transient region is correct and the transient region becomes the actual solution for the next time step. If not, the batteries stay idle for a single interval. This process is repeated until the horizon is reached. The resulting computational complexity is $O(N^2)$, meaning that the number of iterations required for the solution is upper-limited by the square of the time horizon. Note that this is the same complexity as a matrix multiplication. The flow chart of the centralized solution is shown in Figure 3.7. During the transient region, the solver solves the global cost equation on line 12 and continues until the batteries reach a full/empty state. This

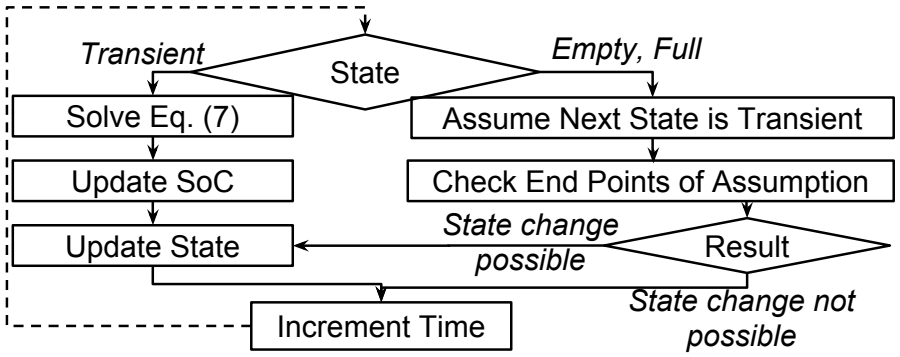

Figure 3.7: Algorithm flowchart for the centralized solution.

means that the solution tries to shave the highest cost values during the transient regions.

$\lambda \leftarrow 0$
for $t = 1 : T$ **do**
  if state_n == *Empty State* **or** *Full State*, $\forall n$ **then**
    for $t' = t : T$ **do**
      possible $\leftarrow$ CheckStateChange(t,t',n)
      if possible **then**
        $\text{state}_n \leftarrow$ *Transient State*
        Solve $C'(p_t + \sum_m \text{bat}_{m,t}) = R_n(\text{bat}_{n,t})$
      end if
    end for
  else
    Solve $C'(p_t + \sum_m \text{bat}_{m,t}) = R_n(\text{bat}_{n,t})$
    $\text{bat}_{n,t} = \min(\max(\text{bat}_{n,t}, L_n), U_n)$
  end if
  UpdateSoC()
  if $SoC_{n,t} == SU_n$ **then**
    $\text{state}_n \leftarrow$ *Full State*
  else if $SoC_{n,t} == SL_n$ **then**
    $\text{state}_n \leftarrow$ *Empty State*
  end if

21: **end for**

Since the solution is based on a finite horizon, it has two methods of applicability in practice: 1) the algorithm is executed once for a long horizon to get the optimal solution for long term planning; 2) MPC, where the algorithm is executed at each time interval based on the predicted horizon values and dynamically updated at the next decision interval.

3.4 Optimal Distributed Battery Control

The centralized solution is easily convertible to a coordination based distributed solution by construction. The distributed solution has multiple advantages over the centralized solution: 1) Computational complexity is reduced by an order of magnitude. All steps performed by the centralized solution are divided between all battery controllers in a fair manner; 2) No initial system charting is required. Battery related information is only used by the battery itself; 3) The cost coordinator, possibly implemented by the utility company, can close its cost model to the users. An energy sharing neighborhood or microgrid can minimize its utility bill using this solution.

To formulate the distributed solution, we leverage the requirement for the optimal solution: batteries must be in the same state at the same time due to (3.9). This requires that any transient region starts from an empty/full state and end at another empty/full state synchronously across all batteries. Even though this seems counter-intuitive from a heterogeneous degradation point across batteries, the degradation is inherently present within SU_n in (3.9), resulting in a more degraded battery to be used less. To achieve this condition, the charged or discharged energy by each battery at each interval must have the same effect on their total capacities:

$$\gamma_t \triangleq soc_n(\text{bat}_{n,t}) = soc_m(\text{bat}_{m,t}), \forall n, m \tag{3.12}$$

The ratio, γ_t, is a single value to be satisfied by all batteries within an interval. We use this property to create three distributed negotiation based solutions. All solutions use steps similar to the centralized solution. At each interval, if the

batteries are at an empty/full state, they assume that a transient region is possible. The battery power levels are then computed using (3.9) and (3.12), until an empty or full state is reached. The solution checks whether the sign of (3.9) is satisfied at the end points. If it is, the solution uses the assumed transient region, otherwise all batteries stay idle for a single interval. All steps require consensus between the batteries, where the power level of each battery must be solved satisfying (3.9) and (3.12). We present three negotiation schemes to solve (3.9), while automatically satisfying (3.12): 1) Circular negotiation ring that has a convergence rate independent of the number of batteries, 2) Mean circular negotiation ring, which converges very quickly for a small number of batteries, 3) Bisection method that has a convergence rate independent of the battery capacities, providing an upper bound to all. All solutions guarantee in a distributed way that all battery states are synchronous with each other and are determined such that (3.9) is solved in their respective regions of convergence.

3.4.1 Circular Negotiation Ring

Based on a circular communication pattern between the batteries, this solution requires a fully connected graph with a minimum node degree of 2. Since the batteries are connected to the grid and the grid is a fully connected structure, this requirement is satisfied under most circumstances. The process is described in Figure 3.8. Each battery starts with an initial consumption guess (e.g. set to the previous consumption value). At each negotiation step, every battery sends its consumption value and their current $\gamma_{n,t}$ ratio calculated using (3.12) to the next battery. The receiving battery updates its own consumption based on $\gamma_{n,t}$ such that $\mathrm{bat}_{n,t} = soc_n^{-1}(\gamma_{n-1,t})$. The result is concatenated to the consumption values received from other batteries and sent to the next battery. When the initializing battery receives the results of the current iteration, it either calculates C'_{t+1} by itself if the cost function is known, or communicates with the pricing node (e.g. the utility) for C'_{t+1}. The cost values are used for the next guess value and the next iteration starts.

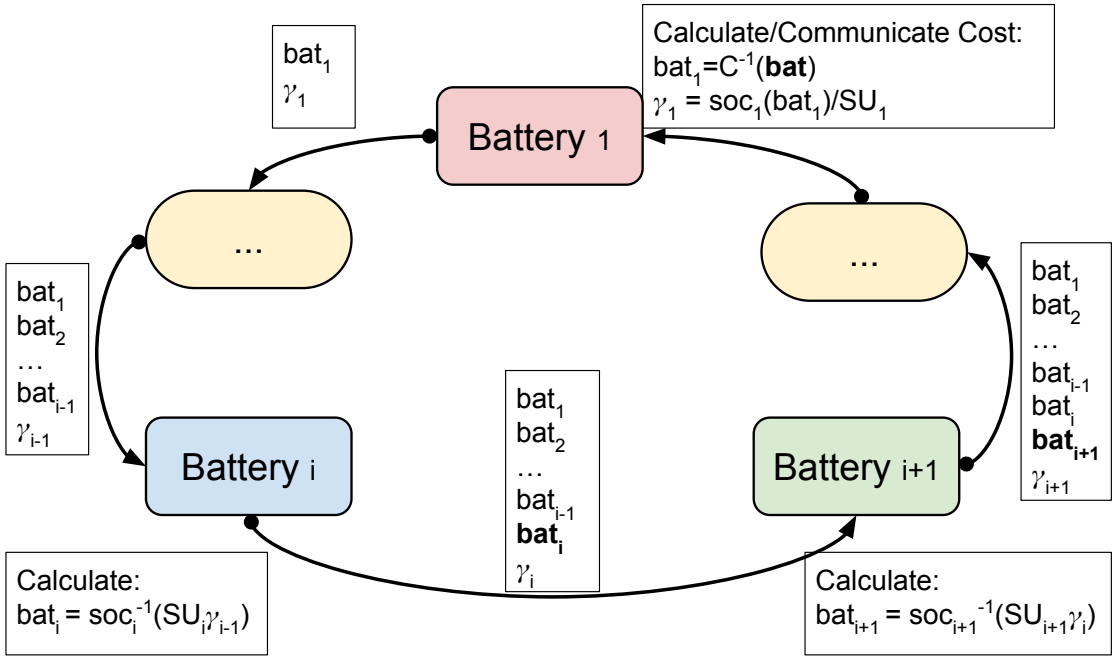

Figure 3.8: The circular negotiation ring illustration, where each node calculates their own γ ratio based on the consumption value of the previous battery.

Convergence

The solution starts by using the γ ratio of the previous battery to obtain its own consumption, given by:

$$\mathrm{bat}_{n+1,t} = soc_{n+1}^{-1}\left(soc_n(\mathrm{bat}_{n,t})\right) \tag{3.13}$$

This chain equation has a closure at the initial node to calculate cost and the whole system becomes a fixed point equation, solvable by fixed point iteration:

$$\mathrm{bat}_{1,t} = C'^{-1}\left(\mathrm{bat}_1, soc_n^{-1}\left(soc_1(\mathrm{bat}_{1,t})\right), \ldots\right) \tag{3.14}$$

We use the following theorem for region of convergence:

Theorem 3.4.1. *Given a fixed point iteration, $x_{t+1} = sys(x_t)$, where $sys(x)$ is a continuous function, the solution converges to a unique fixed point if $sys(x)$ has bounded input $[L, U]$ and maps it to the same bounded output $[L, U]$.*

Proof. If $\mathrm{sys}(L) = L$ or $\mathrm{sys}(U) = U$, then the fixed point is found. Otherwise, $\mathrm{sys}(L) - L > 0$ and $\mathrm{sys}(U) - U < 0$ state that there must be a point satisfying the equation between L and U, due to intermediate value theorem. □

In our case, if C'^{-1} is continuous and stays within the bounds of the battery energy limits, the iteration converges. We continue with Banach's Fixed Point Theorem to bound the convergence rate. We first define contraction mapping:

Theorem 3.4.2. *Let T be a mapping from $X \to X$, where X is the input set in metric space. T is called a contraction mapping if $d(T(x), T(y)) \leq kd(x, y), x, y \in X$ and $0 \leq k < 1$.*

Theorem 3.4.3. *Let $x_{n+1} = T(x_n)$ be a fixed point iteration and T a contraction mapping. Rate of convergence is bounded by $d(x^*, x_n) \leq \frac{k^n}{1-k} d(x_1, x_0)$, where x^* is the solution.*

For our case the magnitude of the Jacobian is:

$$\max_i \frac{soc_i'(\text{bat}_{i,t})}{soc_1'(soc_1^{-1}(soc_i(\text{bat}_{i,t})))} \simeq \max_i \frac{SU_i}{SU_1} \tag{3.15}$$

This means that the convergence rate is upper limited by the maximum capacity ratio of any two batteries. The ratio of 1 is obtained when any two batteries have the same capacity values and the solution will diverge. But, as long as the batteries are not the same, the iteration converges exponentially by the ratio of their capacities independent of the number of batteries. To show this, an example is given in Figure 3.9. In the example, the SoC values for all batteries are assigned such that each battery has a factor multiple capacity of the previous battery (ex. 100, $1.1 \cdot 100$, $1.1^2 \cdot 100$, ...). For three different batteries, all negotiations converge in the same number of iterations, because the convergence rate for the proposed solution is independent of the number of batteries.

3.4.2 Mean Circular Negotiation Ring

Similar to the circular negotiation ring, this solution also requires that each battery sends its consumption and γ_t ratio to the next battery during every iteration cycle. The main difference is in the calculation of γ_t. Rather than using the γ ratio of the previous battery directly to calculate its consumption, an additional parameter of average γ, γ_m, from the previous iteration is sent and used to determine the consumption: $b_{n,t} = soc_n^{-1}(\gamma_m)$. The process is described in Figure 3.10.

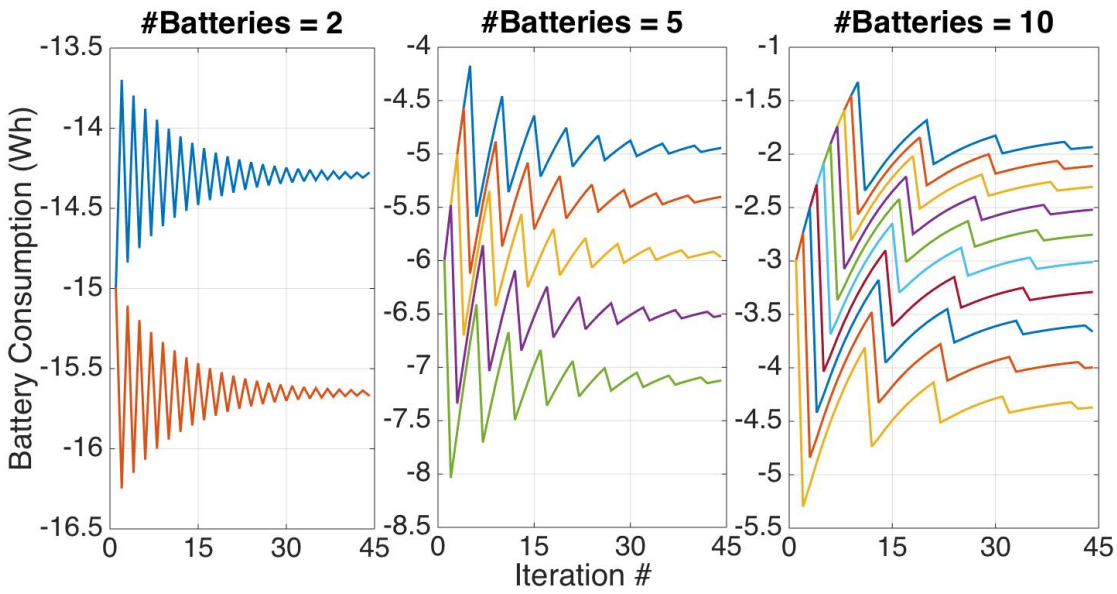

Figure 3.9: For circular negotiation ring, the same capacity ratio has been maintained for different numbers of batteries. The resulting convergence rate is the same for all cases, independent of the number of batteries as expected.

The extra averaging enables an order of magnitude faster convergence for a small number of batteries.

Convergence

Each battery uses the average γ_m value from the previous iteration: $\text{bat}_{n,t} = soc_n^{-1}(\gamma_m)$. After each battery calculates its own consumption, the closure is achieved at the initializing node to calculate the mean ratio for the next iteration, γ'_m, where N is the number of batteries:

$$\gamma'_m = \frac{1}{N}\sum_{n=1}^{N} soc_{n,t} = \frac{soc_{1,t}}{N} + \gamma_m \frac{N-1}{N} \tag{3.16}$$

Combining this expression with the consumption we obtain:

$$\text{bat}_{n,t} = soc_n^{-1}\left(\frac{soc(\text{bat}_{1,t})}{N} + \frac{N-1}{N}\gamma_m\right) \tag{3.17}$$

The Jacobian is either dominated by $\frac{SU_n}{NSU_1}$ or $\frac{N-1}{N}$, depending on the configurations. In the event of a large number of batteries, the second term dominates and the convergence rate slows down as the number of batteries increases. This

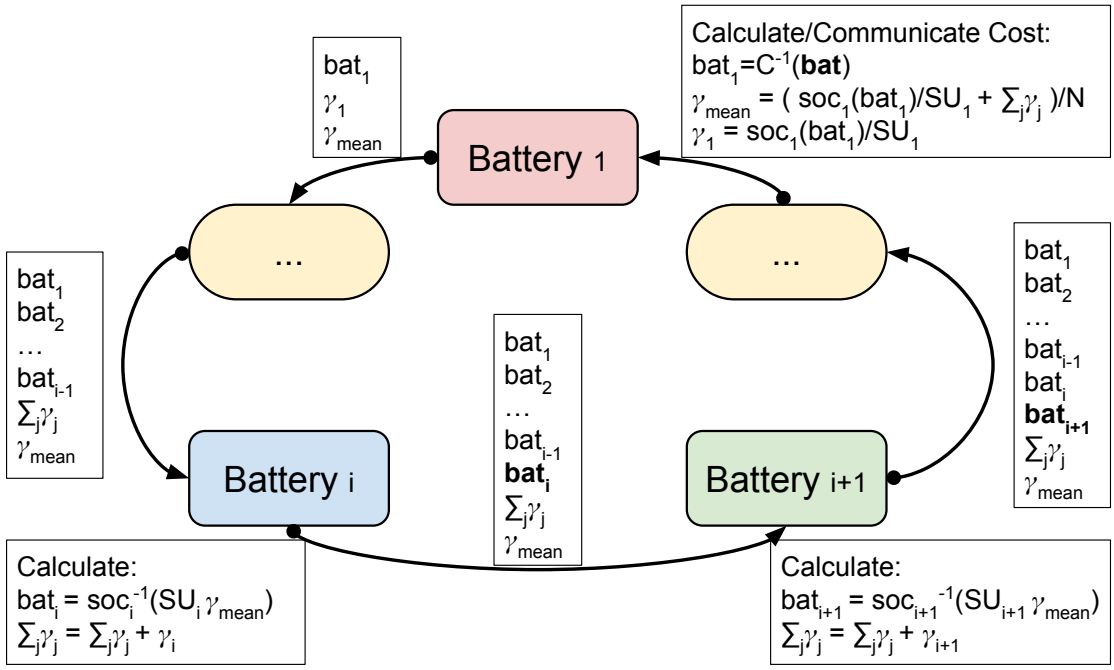

Figure 3.10: The mean circular negotiation ring illustration, where each node calculates their consumption based on the mean γ ratio from the previous iteration.

is a disadvantage compared to the previous algorithm, but for a low number of batteries and the case of identical batteries, the first term dominates and we have an order of magnitude faster convergence than the first method. Thus, the algorithm selection depends on the number of batteries and their configuration. An example is illustrated in Figure 3.11. A region of divergence exists for a small area, due to the discontinuous nature of: $soc''(x) = \alpha(\alpha-1)x^{\alpha-2}/SU$ for $x < 0$. This discontinuity in the calculation of R_n, causes the negotiation to oscillate around its neighborhood, defined by: $\lambda_n \alpha_n (\lambda_n - \text{load}_n)^{\alpha-1} \leq 1 \cap \lambda_n \leq \text{load}_n$. Although this region corresponds to small discharge powers of a few watts for typical configurations, the Bisection method solves these problematic regions.

3.4.3 Bisection Method

This iteration scheme is a modification of the mean circular negotiation ring method. The iteration is not determined by a reevaluation of previous values, but by partitioning using previous results, illustrated in Figure 3.12. The communi-

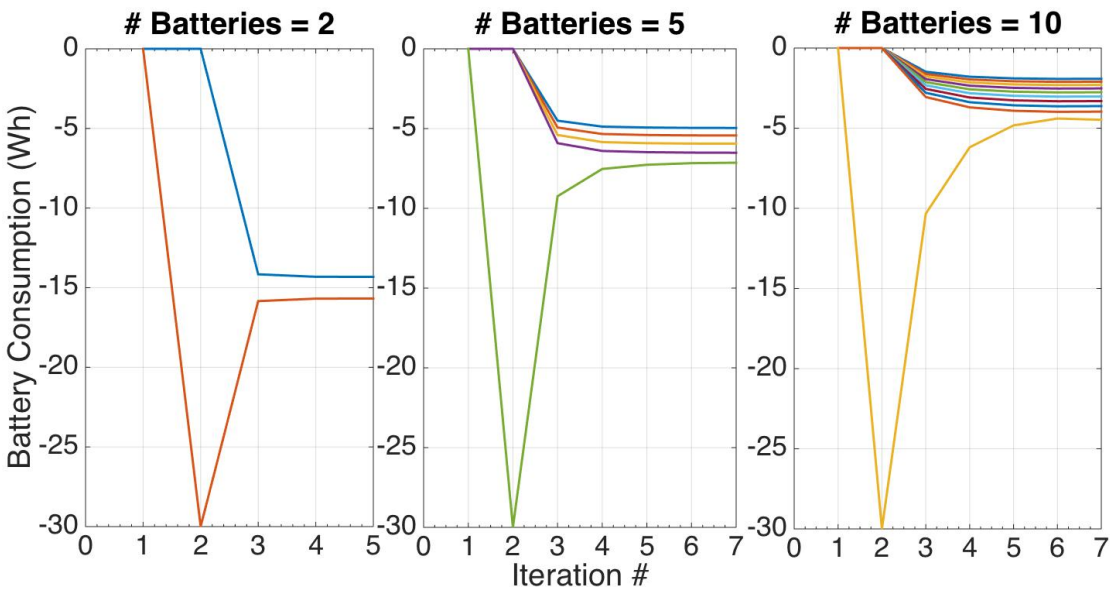

Figure 3.11: The convergence rate of the mean circular ring negotiation is increasing with the number of batteries, but has a lower value compared to the circular negotiation ring case in Figure 3.9.

cation among batteries remains unchanged, where the consumption is calculated using γ_m. The difference is that the initial battery does not compute the next iteration consumption directly using the inverse of the cost function, but starts with the widest consumption interval possible: the upper and lower power limits. To satisfy optimal conditions, the individual constraint function must be equal to the total cost function in (3.7). The difference between them is considered the error in this case. Due to the mean value theorem, the solution in the viable region must have an error value of zero and its neighboring points must have opposite signs. The bisection method uses this knowledge to partition the valid consumption range into smaller regions with opposite signs. Since the range is fixed, the number of partitions and convergence rate are also fixed.

Convergence

Since the batteries are power constrained, the viable region can only be partitioned into finite subregions, upper limiting the number of iterations. For an error tolerance of δ, the number of iterations can be obtained as $\log_2(U - L)/\delta$.

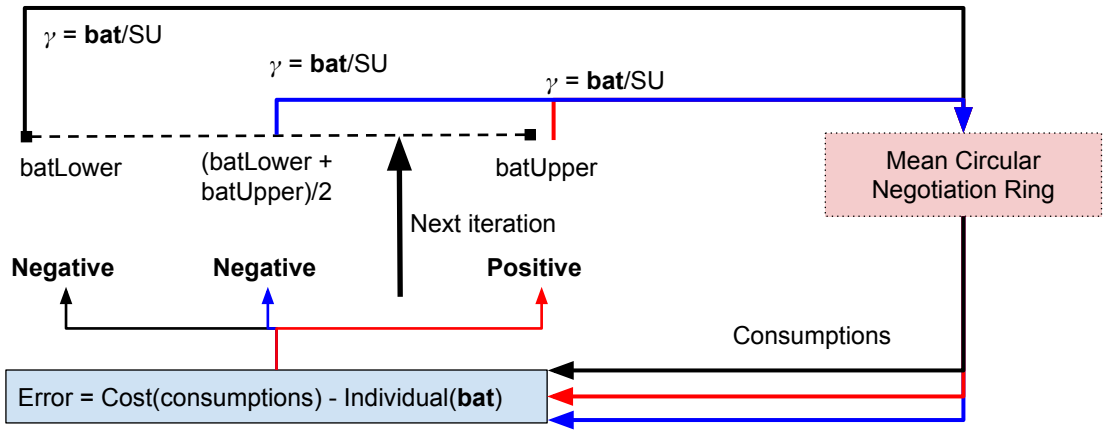

Figure 3.12: The bisection method workflow illustration.

For example, a battery with power limits of $\mp 150W$ converges in 8 iterations for a tolerance of $1W$.

3.5 Results

3.5.1 Experimental Setup

We consider multiple use-case scenarios to compare our control algorithm. For comparison purposes we have selected the widely used heuristic of load following and a recent algorithm in the literature [19] that provides a linear optimal centralized and two decentralized solutions. The linear algorithm in [19] is used for both single trajectory calculation and MPC separately. Load following algorithm discharges the battery as the inverse of the load, such that their summation is constant. In all cases except the voltage deviation studies, the load profile is a residential building at the University of California, San Diego (UCSD) that houses 350 students to showcase the applicability of our solution for a single building. The profile has a typical trend, where the peak occurs in the evening after classes end and a smaller peak in the morning before classes start. All algorithms have a solution horizon of 24 hours. MPC solutions are recalculated every 15 minutes. The Bisection Method form of our algorithm with a single trajectory is used due to constant iteration performance, even though all forms would have converged to

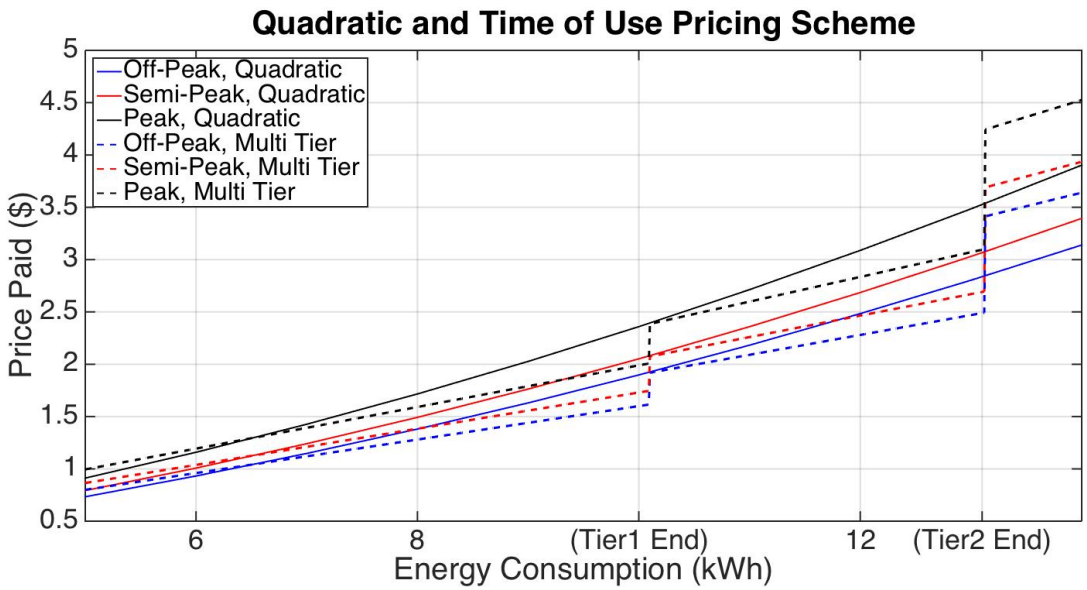

Figure 3.13: Quadratic pricing scheme used in simulations

the same optimal result.

The cost function is a combination of time of use pricing and quadratic pricing. Time of use pricing is one of the most widely used electricity tariff in residential buildings. We use values from San Diego Gas and Electric (SDGE) residential rates [83]. A quadratic pricing profile is widely used for electrical stability based studies [19]. Note that our algorithm is guaranteed to provide the optimal solution for any convex cost function, including time of use and multi-tier functions. The quadratic function is selected since the quadratic nature punishes the user for high power values, incentivizes a flatter profile and allows us to do a fair comparison with [19], designed for quadratic pricing. The cost parameters were obtained by fitting a quadratic polynomial to multi-tier SDGE prices: $0.016\ \$/kWh^2p^2 + 0.096p\ \$/kWh$, where p is the total energy consumption in a time step. Each time step is 15 minutes. We adjust the coefficients according to the time of use as shown in Figure 3.13. We analyzed three scenarios in MATLAB: 1) we show the solution profiles of our solution for different cost functions and battery configurations, 2) we study how the number of batteries affect the total electricity cost, 3) how the capacity of the batteries affect the total electricity cost. In both studies, the degradation cost and electricity cost are studied separately.

3.5.2 Battery Model Accuracy

Before testing our algorithm, we measure the accuracy of our battery model and fine-tune its parameters. Our battery model is verified against NASA battery prognostics repository in [81] for Lithium-ion batteries. The repository contains various experiment scenarios and physical measurements cycling batteries until their capacity is reduced below the industry standard of 80% of their original capacity. We simulated the same scenarios using our model and compared our SoC and SoH estimates with the measurements. These scenarios contain 3 different ambient temperatures, 3 levels of output power and various levels of depth of discharge. Our model captures the nonlinear SoC behavior with less than 2% error and the nonlinear SoH degradation with 1.6% error. In contrast to our model, linear models in the literature don't consider the nonlinearity of SoC and ignore the degradation. Figure 3.14 shows the percentage error between experimentally measured state of charge and the battery model and Figure 3.15 shows the state of health degradation for the measured and model calculated values.

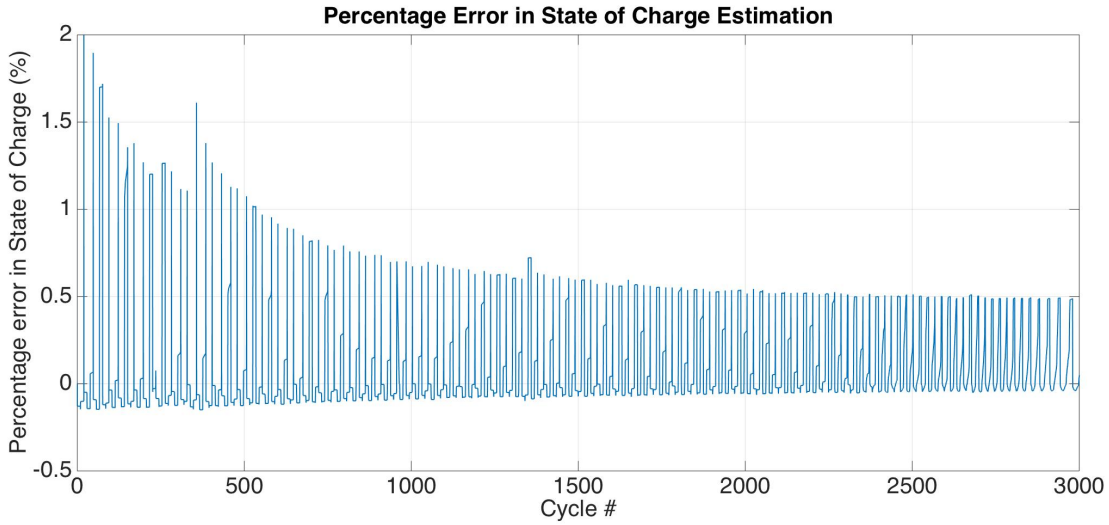

Figure 3.14: State of charge error percentage profile of the model's output.

The SoH comparison is shown in Figure 3.15. The measured data in the SoH figure is nonlinear as expected from the battery's nature and shows the hysteresis profile. Our model only deviates by 4.2% from the measured data. Note

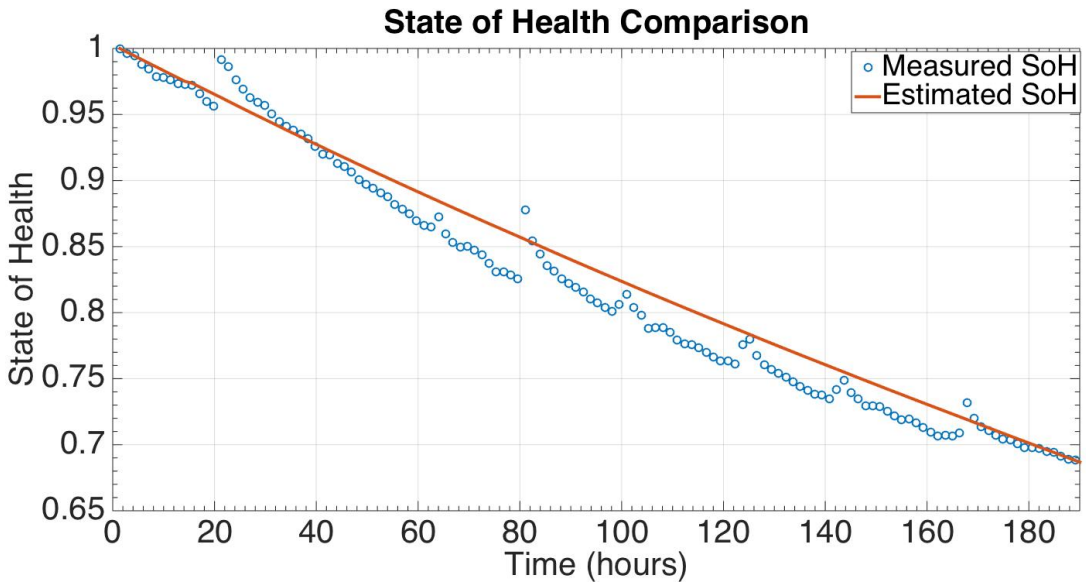

Figure 3.15: SoH profiles of measured data and model output.

that it is possible to model the hysteresis profile more accurately by using two exponential hysteresis [31], but it increases the complexity and is out of the scope of this thesis. In contrast to our model, linear models in the literature don't consider the nonlinear nature of SoC and ignore the degradation completely, causing large errors as shown in the next sections.

3.5.3 Effect of Cost Function and Battery Nonlinearity

We created multiple case studies with different cost functions and battery nonlinearities to show the versatility of our solution. The load time series is the same across all cases and we used two battery banks, where the first one has twice the capacity of the second one (500kWh). The solution time series for both batteries and SoCs are given in all results. Figure 3.16 shows the solution for the case of a time of use pricing, showcasing that our solution is capable of different cost functions, including discrete jump functions. The jumps in the solution mark the expensive and inexpensive time intervals.

Figure 3.17 shows the solution for the quadratic cost function case. The resulting profile is a flatter one compared to the original load due to the quadratic

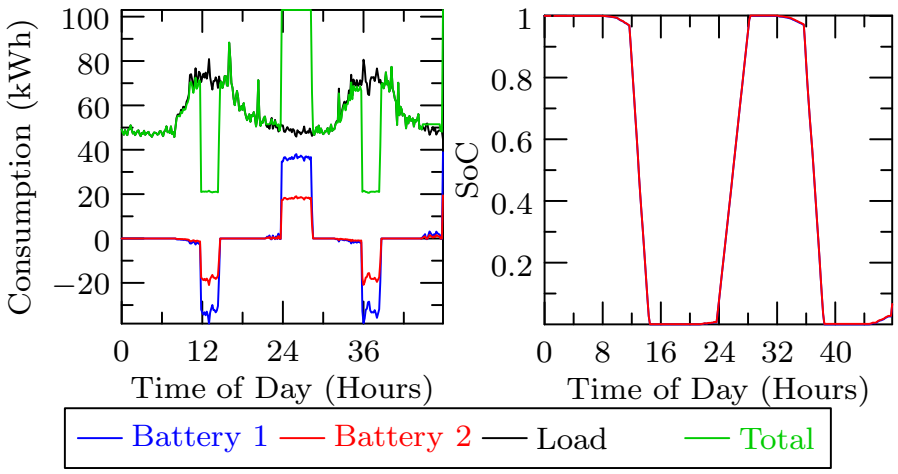

Figure 3.16: Optimal solution for a time of use pricing with nonlinear battery model

nature punishing spikes in the consumption. However, the profile is not completely flat due to the nonlinear nature of the battery punishing high discharge powers.

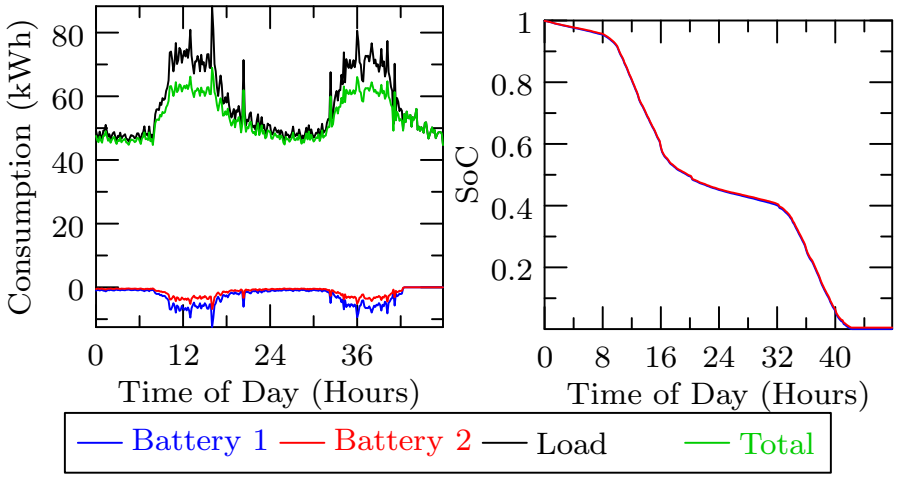

Figure 3.17: Optimal solution for quadratic pricing with nonlinear battery model

Figure 3.18 shows the solution for the quadratic cost function and a linear battery model by setting the Peukert exponent $\alpha = 1$. The resulting profile is flat as expected, since the linear model does not distinguish between different discharge powers.

Figure 3.19 shows the solution for a fourth order cost function and a nonlinear battery model. The resulting profile is flatter compared to the case of quadratic

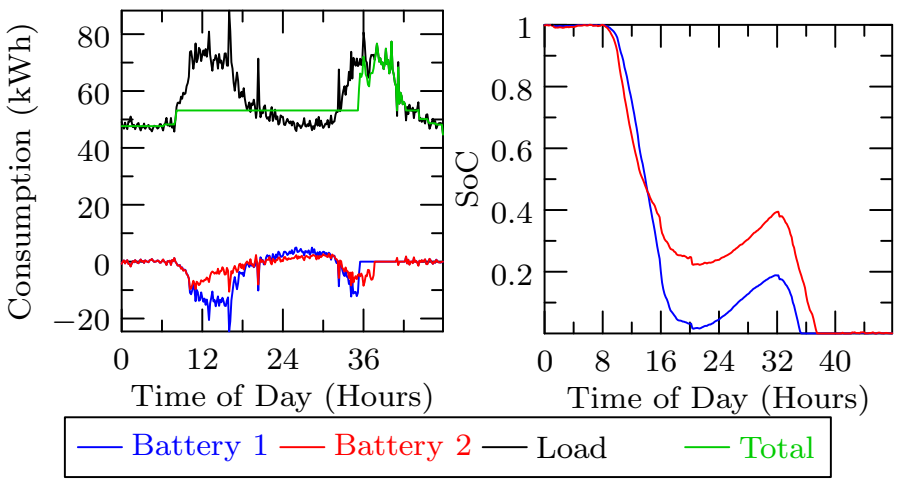

Figure 3.18: Optimal solution for quadratic pricing with linear battery model

cost function, due to a higher punishment. However, the nonlinearity of the battery model results in the solution to be not completely flat as in the quadratic cost case

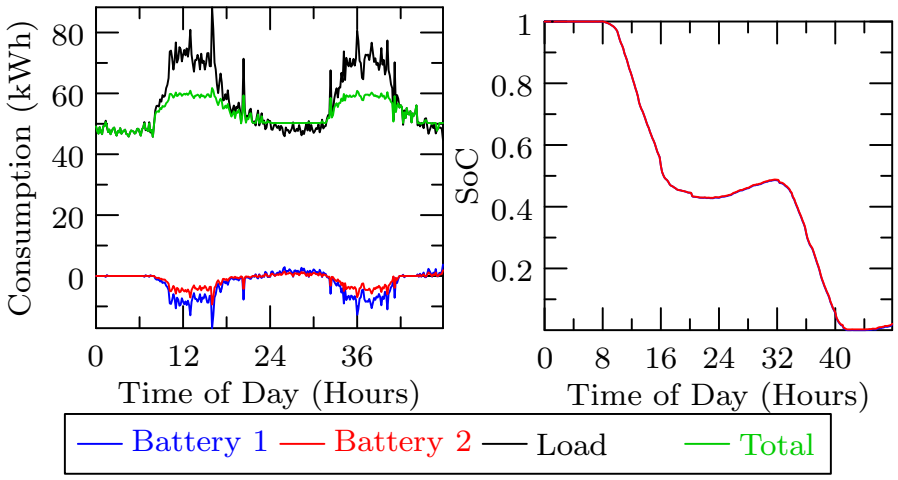

Figure 3.19: Optimal solution for fourth order pricing with nonlinear battery model

Figure 3.20 shows the solution for a quadratic cost function using a very lossy nonlinear battery model ($\alpha = 2$). The consumption of the battery has been automatically reduced as any high consumption value is drastically punished by the battery model. Under the same conditions, if the linear optimal algorithm in [19] is used, the resulting profile is shown in Figure 3.21. The profile differs

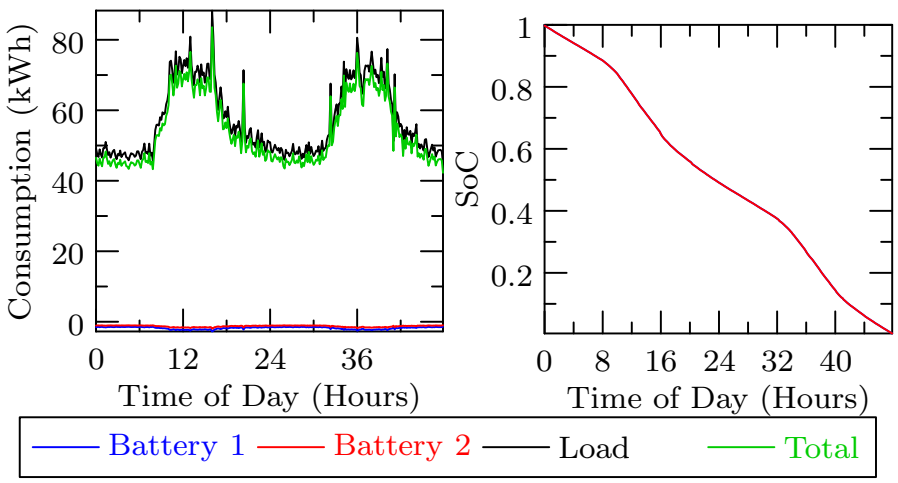

Figure 3.20: Optimal solution for quadratic pricing with lossy nonlinear battery model

significantly compared to our nonlinear optimal solution as expected and results in large cost differences. The figures show that our distributed optimal solution

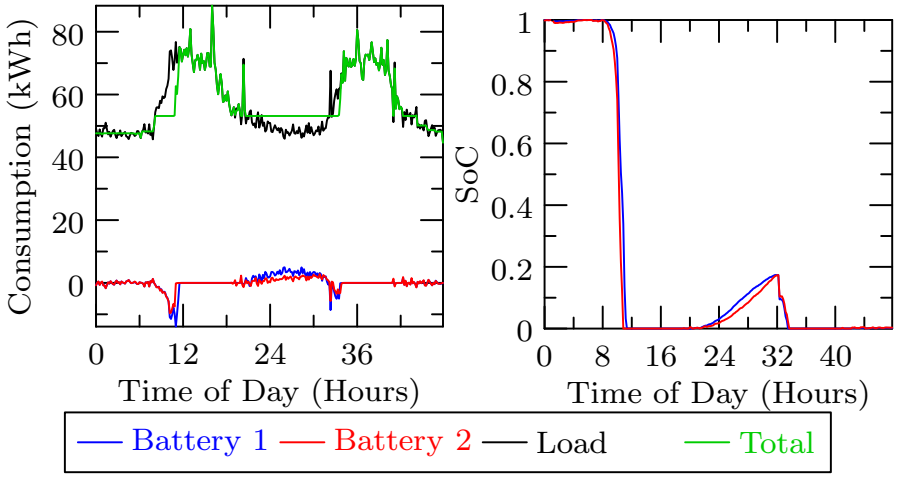

Figure 3.21: Linear optimal solution for quadratic pricing with a lossy nonlinear battery

can operate using different cost functions and battery model selections.

3.5.4 Effect of Battery Capacity on Cost Reduction

For the first case scenario we use a single battery and change its capacity to obtain cost reduction sensitivity with respect to the battery capacity. We compare

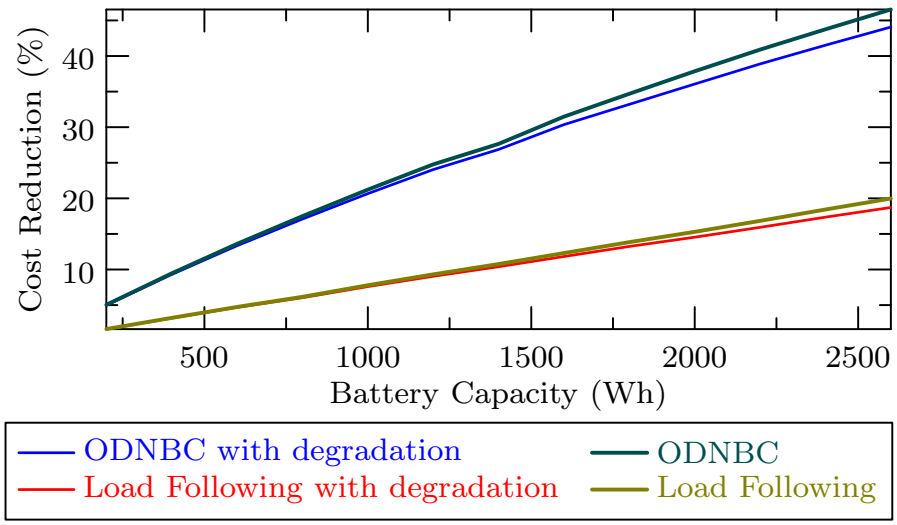

Figure 3.22: Cost reduction in electricity bill for ODNBC and load following.

our method with the no battery case and load following heuristic. Two different cost reduction values are studied: 1) decrease in electricity cost, 2) decrease in electricity cost adjusted by the battery degradation. To obtain the second metric, we add the degradation in the battery capital as an additional cost for using the battery. We set a linear dependence between the capital cost of the battery and its capacity [84] as \$500 per kWh capacity. The cost reduction and the degradation effect results are given in Figure 3.22. The figure showsthat our algorithm, denoted as ODNBC, outperforms load following by up to 30% and results in a cost reduction of up to 50%. The degradation cost results are slightly higher for ODNBC, because the optimal solution requires the battery to be discharged over a longer time interval.

3.5.5 Effect of Number of Batteries on Cost Reduction

Our second case study is based on the same consumption profile and cost function as in the previous study. In this case, we hold the total capacity of multiple batteries constant and change their individual capacities to understand the cost reduction dependence on the number of batteries used. To show the difference between the different solutions, a 24 hour portion of the optimal solution for a single battery and 32 batteries with equal total capacities of 2000Wh is shown in

Figure 3.23.

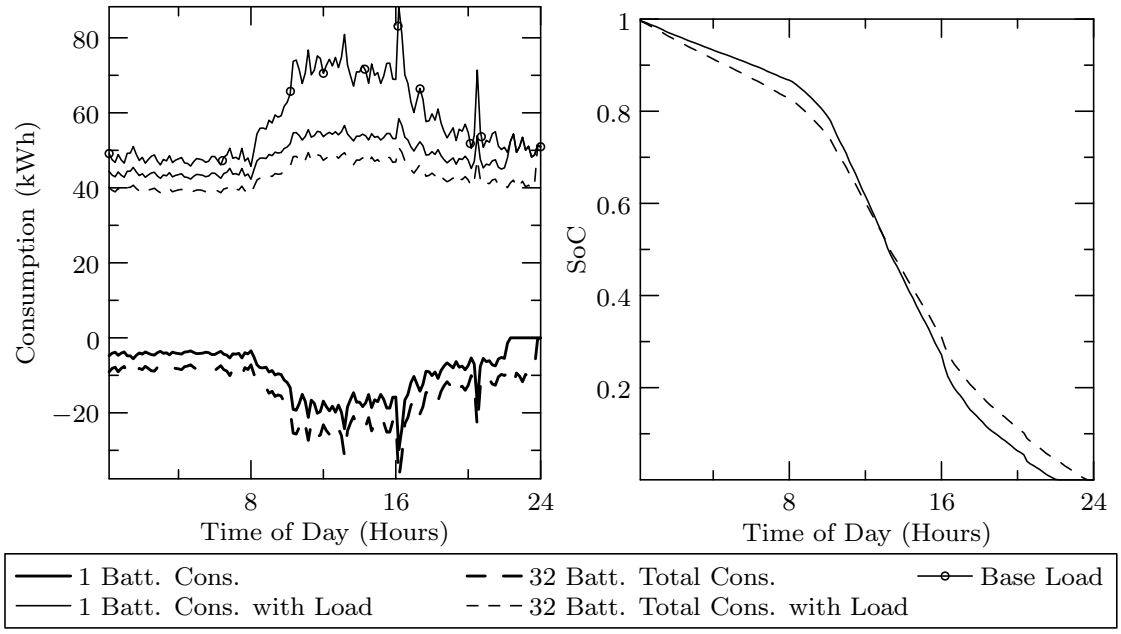

Figure 3.23: 24 hour solution time series for single battery and 32 batteries cases.

The results for cost reduction and degradation in Figure 3.24 show that as the number of batteries increases, the cost and degradation performance of the solution improves. As the batteries get smaller, their power consumption and the effect of nonlinearities also decrease, enabling a higher effective total capacity, hence a higher cost reduction and lower degradation.

3.5.6 Comparison with Linear Optimal Methods

We compare our algorithm against 3 algorithms from recent state-of-the-art work in [19]. The centralized optimal algorithm uses QP to solve the system of batteries. The decentralized algorithm uses QP for each battery to get a solution for its local load separately. The optimality is not guaranteed and deviates from the global optimum for batteries with different characteristics. The third algorithm, a novel market maker MPC is a coordinated solution, where the coordination is achieved by a dynamic price. The optimality and convergence are not guaranteed. All parameters for the algorithms were obtained from [19], whereas the battery model was substituted with our nonlinear model. The reduction in cost relative

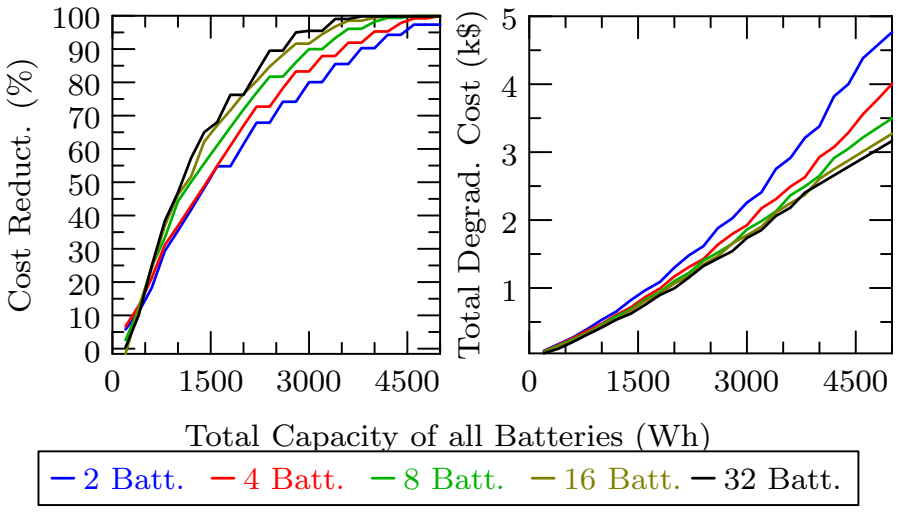

Figure 3.24: Effect of number of batteries on cost reduction and degradation.

to the no battery case is shown in Figure 3.25. The results show the significant

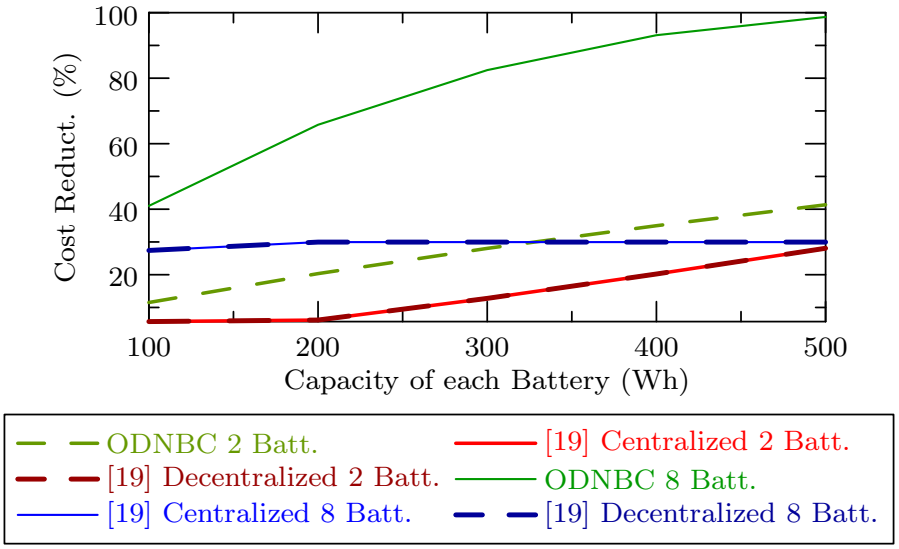

Figure 3.25: Percentage reduction in cost compared to the bateryless case.

error between the linear optimal algorithm and our nonlinear optimal solution. There are two factors: 1) negligible linearization errors at each interval due to the nonlinearity in the SoC function as given in (3.9); 2) accumulation of linearization errors leading into a wrong decision making. The accumulation of small errors in the memory of the battery model causes it to change states prematurely. An

example is illustrated in Figure 3.26. During a discharge period, nonlinearity causes the battery to discharge slightly less than the linear solution, causing it to last longer; whereas the linear solution assumes that the battery will last longer according to its model, causing premature transition to empty state.

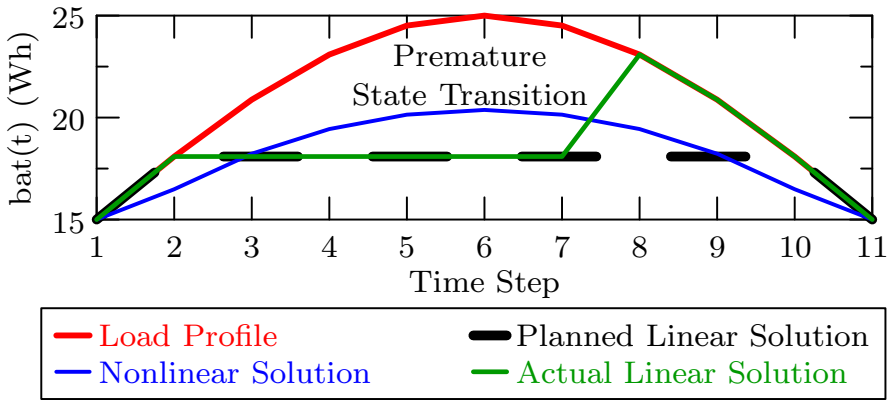

Figure 3.26: Example of linear model assumption leading to premature state transition.

To further study this error caused by nonlinearity, we measure the error between the linear optimal solution and our nonlinear optimal solution for a flat load profile. In this case, we also add the MPC usage of the linear optimal solution [19]. All results with different number of batteries are shown in Figure 3.27. The percentage of error is calculated by: $100\% \times (\text{Cost}_{\text{linear}} - \text{Cost}_{\text{nonlinear}})/\text{Cost}_{\text{nonlinear}}$. The results show that the linear solutions, single trajectory or MPC, have significant errors. This shows that even though the linearization error at each interval is small, their accumulation in time can cause more than a 150% error for MPC and 250% for single trajectory solution compared to the nonlinear solution. These error values increase with increasing number of batteries.

3.5.7 Forecast Error Study

Our simulation setup consists of three different case studies. For the first case, one of the dormitory buildings designed for 80 students is used for the load and the rooftop solar panel is used as the renewable energy source. For the second case, the load is changed to a research building with approximately five times the

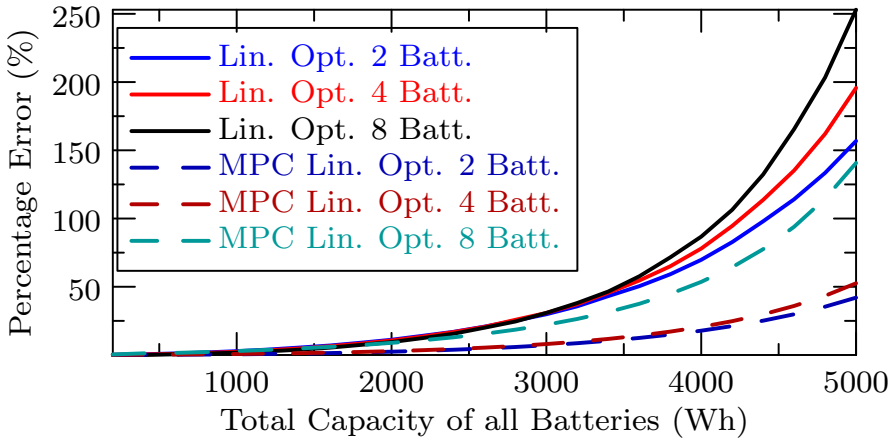

Figure 3.27: Error of linear model solutions, compared to our optimal solution.

demand of the first building, representing a different kind of load profile. The last case is the medical institution. The mean power demand is thirty times larger than the first building, measured within 48 hours. For the renewable energy, the rooftop solar panel is used. All data are taken from real measurements.The error in the prediction is modeled as a Gaussian random variable with zero-mean and the deviation set to the mean power demand, scaled by the error percentage value. In all cases, the simulation is performed using the predicted data and is compared with real data, which is the addition of the noise to the forecast depending on the error percentage value. The reduction in cost and variance are with respect to the no-battery case.

Residential Building

Three different numbers of batteries are selected as $7, 13$ and 30 to observe the effect of number of batteries on the performance. The cost reduction percentage results with respect to the no-battery cases are shown in Figure 3.28. ODNBC saves 16% of the cost of energy drawn from the grid, when 30 batteries are used and continues to be beneficial up to 50% forecast error. If only 7 batteries are used, our solution still saves more than 10% of the electricity cost and remains beneficial up to 40% forecast error. Normally, forecast error is expected to be in the $10-20\%$ range. For this range, our solution looses only 1% of its performance.

On top of these results, the variance in the power drawn from the grid is reduced by 32.6% in the 13 battery case and 22% in the 7 battery case to help the stability of the system.

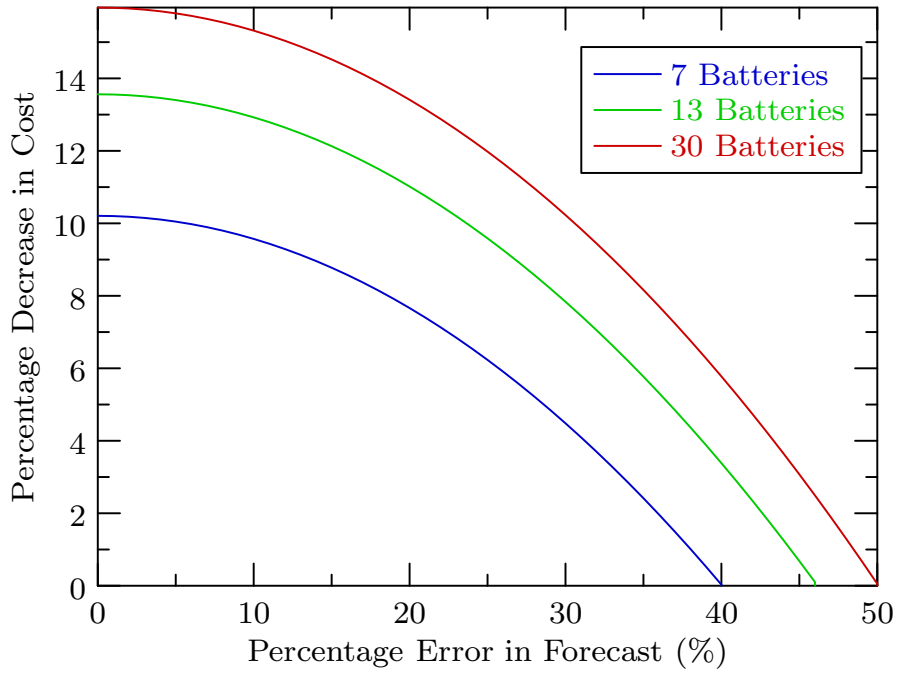

Figure 3.28: Percentage decrease in electricity cost with respect to the no-battery case for the residential building

Research Building

The second case study is a research building with a different power profile than the first case. The mean power demand is roughly 5 times of the residential building.Three different number of batteries are selected as $25, 50$ and 100. These numbers are selected to be approximately 5 times the numbers in the residential case. The cost reduction percentage results with respect to no-battery cases are given in Figure 3.29. ODNBC saves 11% of the cost of energy drawn from the grid, when 100 batteries are used and continues to be beneficial up to 80% forecast error. The percentage decrease in cost is lower than the residential case. The reason is that, although the number of batteries is approximately scaled by five, only the capacity of the storage is scaled, due to the battery interconnection type.

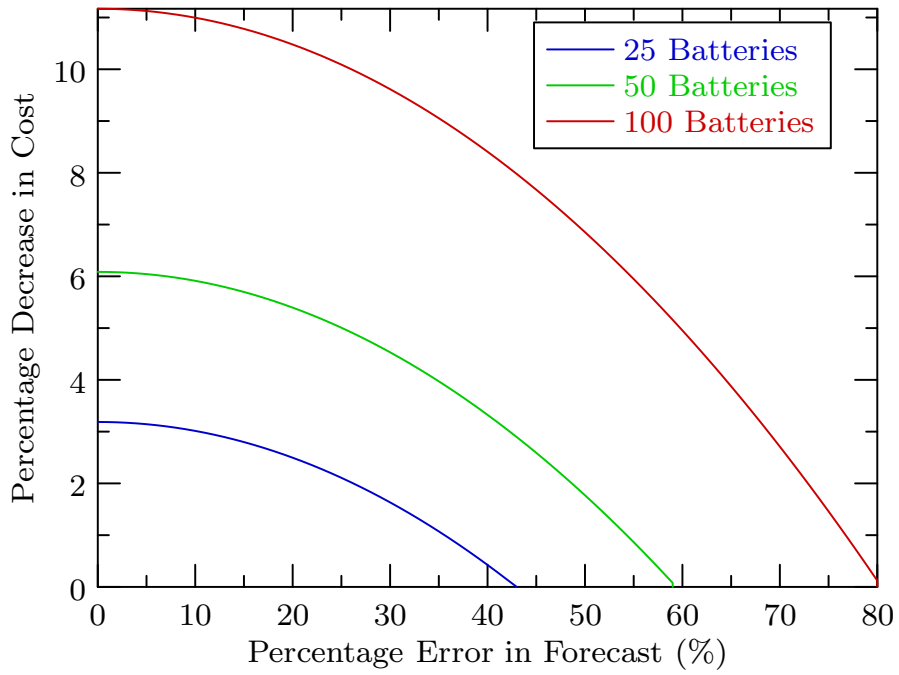

Figure 3.29: Percentage decrease in electricity cost with respect to the no-battery case for the research building

The output power is not enough to decrease the cost further as in the first case. In addition, the variance in the grid power is reduced by up to 31% in the 50 battery case and 21% in the 25 battery case. For the expected forecast error region, ODNBC looses only less than a percent of its performance.

Medical Institution

Our third study tests the performance of ODNBC on a large load, the medical institution. The mean power demand is 30 times larger than the residential building. In order to have a comparable and working result, five batteries are connected to form clusters that give higher power output. The number of clusters for the simulation are selected as $250, 500$ and 1000, obtained by scaling the numbers in the residential case by 30. The percentage decrease in cost with respect to the no-battery case is given in Figure 3.30. ODNBC saves 21% of the cost of energy drawn from the grid, when 1000 clusters are used. Furthermore, the variance of the grid power is reduced by up tp 92% in the 1000 cluster case and by 58% in

the 250 cluster case. Since the number of batteries is very high, the results aren't effected by the forecast error significantly.

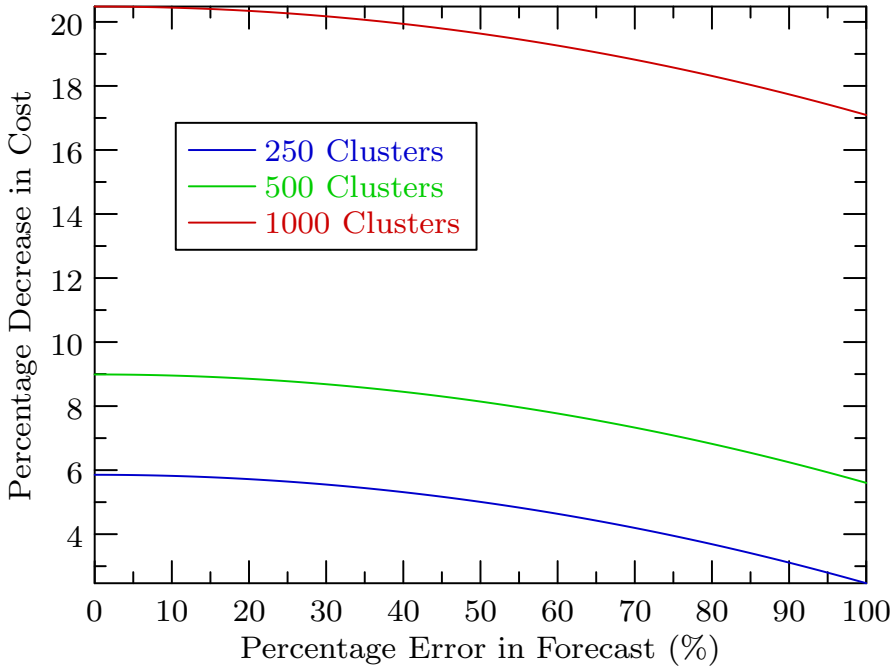

Figure 3.30: Percentage decrease in electricity cost with respect to the no-battery case for the medical institution

3.5.8 Voltage Deviation Comparison

We compare our control algorithm to single trajectory and MPC based linear optimal algorithms from the perspective of circuit stability. We performed a voltage stability analysis on a residential neighborhood to understand how the voltage deviation is affected by the use of batteries. The electrical circuit is taken from NREL, which represents a real neighborhood in south Los Angeles, shown in Figure 3.31. There are 23 houses and their load profiles are created using HomeSim, a residential energy simulator [85]. All homes contain daily appliances, PV and 20% of them contain large loads such as an EV or a pool pump. PV profiles are obtained from UCSD's rooftop PV measurements. EV profile is created using the charging profile of a Nissan Leaf. All other appliance profiles are created by HomeSim that schedules start and stop events based on measured appliance statistics. Voltage

stability is obtained through S^{2}Sim [2], a smart grid simulator that calculates the power flow solution using OpenDSS [7]. The details of the simulator is described in Chapter 4. The battery locations are marked as B on Figure 3.31. The error introduced due to the aggregation of loads and generation time-series is also presented. Even though residential energy sharing is not implemented currently in practice, there is an increasing amount of research for energy sharing neighborhoods [86, 87, 88]. Three cases with $2, 5$ and 10 batteries are studied, where

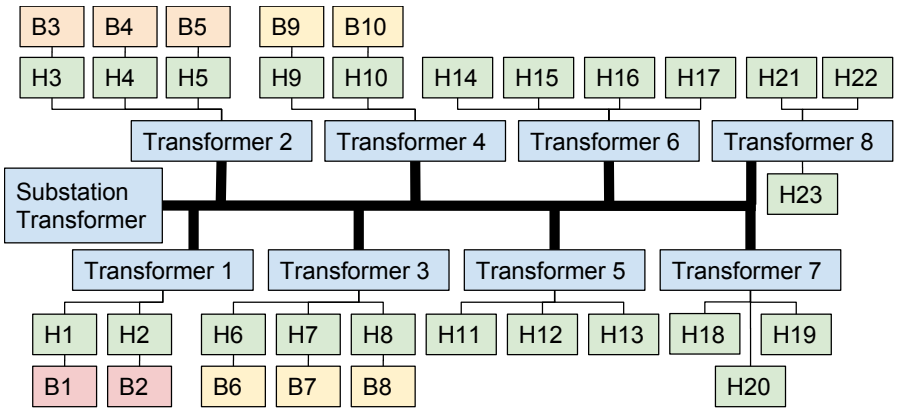

Figure 3.31: Neighborhood circuit with battery locations

each battery has a capacity of $500, 200$ and 100Wh, respectively, so that the total amount of battery capacity is the same for all cases. The maximum absolute voltage deviation observed at any terminal is shown in Figure 3.32, where we focus on the afternoon hours as only those have significant deviation. The widely accepted maximum allowed voltage deviation is 10%. Thus we use the following metric to emphasize the voltage deviation difference *relative* to the maximum allowed value: Relative Reduction = $(\text{MaxDeviation}_{\text{No Battery}} - \text{MaxDeviation})/(10\%\ \text{Limit})$. At peak hours, our algorithm achieves a relative voltage deviation reduction of up to 45%, helping the voltage stability significantly. At peak hours, our algorithm achieves a voltage deviation of only 2.4%. Compared to the 6.85% voltage deviation of the bateryless case, the difference of 4.45% results in a relative reduction of 45%. As the number of batteries increase, the voltage deviation values decrease since more terminals contain batteries as stabilizing active devices. Table 3.3 shows the relative voltage deviation reduction values for all cases, along with their actual

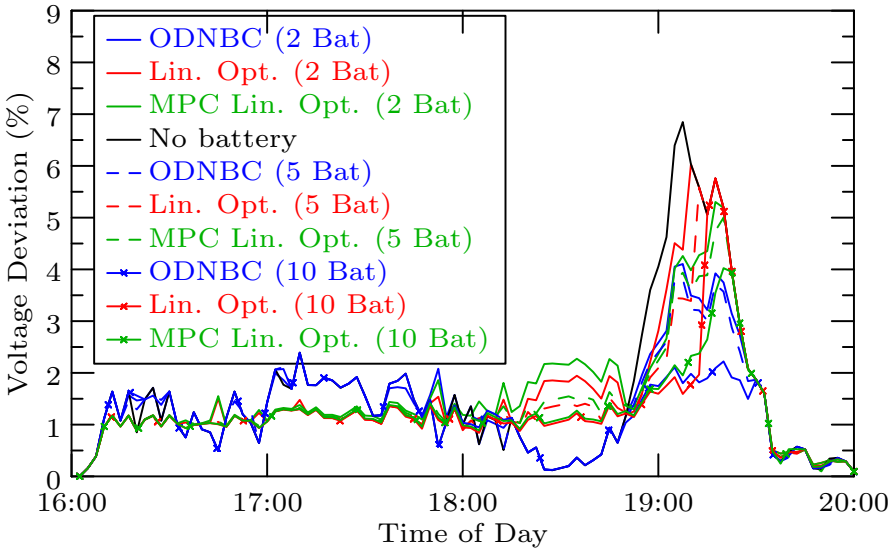

Figure 3.32: Absolute voltage deviation profile comparison of all solutions.

Table 3.3: Voltage Deviation Reduction Relative to Deviation Limit

N	Batteryless	**ODNBC**	Linear Opt.	MPC Lin. Opt.
2 Batt.	0% (6.9%)	28% (4.1%)	8% (6.0%)	16% (5.3%)
5 Batt.	0% (6.9%)	30% (3.8%)	21% (5.8%)	19% (5.0%)
10 Batt.	0% (6.9%)	45% (2.4%)	21% (5.8%)	24% (4.0%)

deviation values in parenthesis. The results show that our algorithm has the highest reduction, lowest voltage deviation and as the number of batteries increase, the reduction improves even more, since more terminals contain batteries as stabilizing active devices.

Finally, we investigate the error introduced by aggregating the loads. We look at the total line losses of the same neighborhood scenario and measure the mean absolute error compared to the actual nonlinear power flow solution. The error is normalized by the total consumption to provide a ratio of lost energy to useful energy: $\frac{1}{T}\sum_{t=1}^{T}|\frac{\text{Loss}_{\text{nonlinear}}(t)-\text{Loss}_{\text{constant}}(t)}{\text{Consumption}(t)}|$. The result is only 0.84% for our solution and 1.24% for the linear optimal solution, justifying the use of aggregation.

3.6 Conclusion

Energy storage systems enable the on-demand dispatch of energy to compensate for generation and consumption volatility, especially arising due to the integration of renewable resources. Our optimal distributed battery control handles multiple batteries, with low computational complexity of $O(N^2)$. In contrast to previous work, we use a higher accuracy nonlinear battery model with only 2% error. We provide three iteration mechanisms to implement the distributed optimal solution with proven convergence regions and rates: 1) circular negotiation ring has convergence rate independent from battery count; 2) mean circular negotiation ring converges an order of magnitude faster than the first scheme for low number of batteries; 3) bisection method has a convergence rate independent of battery capacities.

We show in a case study that optimal algorithms designed for a linear battery model induce an error of up to 60% in terms of cost reduction, due to the nonlinear nature of the battery. For the case of a constant load profile, we show that this error exceeds 150%.

In the previous chapter we presented a forecast algorithm for solar energy to accurately predict the amount of energy to be generated. We found the forecast accuracy to be 10% for a 24 hour horizon. However, predicting irregularities in generation is not enough to solve the integration of renewable resources. Our optimal battery control solution can be used in conjunction with renewable forecasting to mitigate these disadvantages. Our forecast error studies show that, if TESLA and ODNBC were to be used together, the degradation in cost is only 1% of the original reduction.

Furthermore, the use of distributed control algorithms can either help, as in our case by up to 45%, or worsen the stability of the grid due to slow convergence. Any distributed control algorithm must also be tested in the context of the grid itself. We present a smart grid simulator in Chapter 4 that can help to achieve this.

This chapter contains material from Alper Sinan Akyurek and Tajana Simunic Rosing, "ECO-DAC: Energy Control Over Divide And Control", IEEE Interna-

tional Conference on Smart Grid Communications (SmartGridComm), 2013. The dissertation author was the primary investigator and author of this paper.

This chapter contains material from Alper Sinan Akyurek and Tajana Simunic Rosing, "Optimal Nonlinear Distributed Battery Control", IEEE Journal of Emerging and Selected Topics in Power Electronics, Special Issue on Structured DC Microgrids, 2016. The dissertation author was the primary investigator and author of this paper.

Chapter 4

S^2Sim: Smart Grid Swarm Simulator

In Chapter 2 we described the challenges for the integration of renewable resources due to their stochastic and uncontrollable nature. These rapid variations coupled with distributed generation can lead to voltage swings in the grid, endangering the stability of the system. In Chapter 3, we described an optimal control strategy for batteries integrating renewable resources and minimizing overall system cost in a distributed fashion. Both solutions reflect the distributed nature of the smart grid and rely on the underlying grid structure to remain stable and unchanged with the actuation decision. Other distributed control algorithms at different scales within the grid, such as demand response, EV charging and home automation, are being developed and deployed; yet their effects on each other and the grid's health and stability have not been sufficiently studied due to the lack of a capable simulator. Simulators in the literature can solve the power flow by modeling the physical system, but fail to address the cyber physical aspect of the smart grid with multiple agents. To answer these questions, we have developed S^2Sim: Smart Grid Swarm Simulator. S^2Sim allows any object within the grid to have its own independent control, transforming physical elements into cyber-physical representations. Objects can be of any size ranging from a light bulb to a whole microgrid and their representative data can be supplied from a real device, simulation, distributed control algorithm or a database. S^2Sim shields the

complexity of the power flow solution from the control algorithms and directly supplies information on system stability. This information can be used to give feedback signals like price or regulation incentives by virtual coordinators to form closed-loop control. Using three case studies specifically possible with our simulator, we illustrate how different distributed control algorithms can have varying effects on system stability, where the first case shows, why it is necessary to have closed loop control for grid stability. The second case shows that we cannot justify the performance of a control algorithm under isolated conditions alone, without testing it within the grid picture. The third case shows that we can use S^2Sim to compare the performance of different heuristics using our tool.

4.1 Introduction

With the growth in information technology and increasing demand for power, interest in the smart power grid has risen rapidly. As smarter loads, devices, appliances, storage elements and generators, or, in general, a swarm of *objects* with sensing and/or actuation capabilities connect to the grid, the need for scalable, stable and distributed control algorithms rises rapidly. There is large body of research on the control of both the client side and the utility-provider side of the smart grid separately. This multi agent system is shifting the physical electrical grid into a Cyber Physical System (CPS). One of the most important aspects of the smart grid is the electrical stability of the system. The classical power grid model has more concrete separation of the demand and generation sides. The smart grid, in contrast, with its CPS side of distributed control, generation and energy storage devices [89] is forced to be smarter to address the dangers of instability that can cause major problems, e.g. blackouts.

The elements in the smart grid are moving towards a more autonomous and distributed structure, with diverse control algorithms. Home automation [90], offices with HVAC controllers, microgrids, datacenters using Photovoltaic (PV) [91] or energy storage devices at substation levels [92] are examples of increasing autonomy. But, majority of the control algorithms are designed from a local perspective,

modeling the grid as an uninterruptible power supply. This approach has two major flaws: 1) The cross-effect of multiple controllers on each other is not studied, 2) The cumulative-effect of the control decision on the grid itself is left out. It is crucial to test and evaluate any control solution not only in an isolated environment, but also with respect to the bigger picture of the constituent smart grid in order to get a more realistic success metric. Recently, a white paper from a multi-institutional collaboration mentions a need for a smart grid simulator that can connect loads from different physical locations, including real hardware to bring the physical aspect into the loop [93]. In order to achieve these goals, there is a need for a smart grid simulation tool, which can handle the swarm of objects with distributed, diverse (possibly heterogeneous) control algorithms in a dynamic fashion, without introducing any constraints on the objects.

In order to address these needs, we designed and implemented S^2Sim , Smart Grid Swarm Simulator. S^2Sim allows real-time co-simulation of distributed control algorithms within the smart grid and studying the grid's behavior and health under various desired conditions. To the best of our knowledge, existing simulators in the literature either don't support dynamic, real-time object behavior [35] or constrain the object control strategies to predefined libraries with predefined behavior [36][38][37]. Section 4.2 has a detailed analysis of existing tools and their limitations. Our main contributions are:

1) A smart grid simulator, capable of evaluating independent distributed control algorithms to analyze stability and control issues in the smart grid with heterogeneous objects connected to it. The simulator shields the complexity of the non-linear power flow equations from the control algorithms.

2) A multitude of objects within the grid can be represented as an external (possibly real-time) data stream, a real hardware, simulation code or control algorithm over a reliable TCP/IP connection. These objects can represent any type of grid element, ranging from loads, generators, microgrids to energy storage elements at any scale, such as a single light bulb or a whole microgrid. In contrast to classical simulators, objects enable the simulation of smart grid's CPS aspect.

3) Multiple coordinators can connect and access system-wide information to emu-

late coordination logics such as the microgrid or a home control hub. Coordinators can provide feedback such as pricing or stability information to objects.
4) S^2Sim handles time synchronization among objects despite different time constants such as an air conditioner and a PV.
5) Our simulator provides an application layer communication protocol for remote access over any network interface. This enables objects that are physically distant from each other to form a virtual grid, enabling parallel computing capabilities. We used this property to perform a US-wide case study.

4.2 Related Work

There are various smart grid power flow simulators in the literature: open source simulators OpenDSS [35] and GridLab-D [36] or commercial products as RTDS [37] and Paladin Live [38]. The objects in these simulators are static objects with fixed behavior, predefined with a time series throughout the simulation. This static behavior prevents any reaction from either the objects or the utility, making it impossible to co-simulate distributed control algorithms. The only way to overcome this is to set the simulation time to a single step and readjust the scenario for the next time step. One common point of the mentioned simulators is that they can all solve the complex non-linear power flow equations efficiently.

OpenDSS and GridLab-D represent the grid by impedances and lines connecting them. There are two ways to control object behavior. The first pre-loads the object behavior as time series before the simulation. The second uses a Dynamic Link Library (DLL) that represents the object behavior during the simulation. The main disadvantage of the first method is the static simulation, where the objects cannot react to anything due to preset object behavior. The second method adds dynamism to the object behavior, but is constrained by the implementation guide of the DLL.

None of the simulators have an interface for a coordinator that can give feedback signals like price or regulation incentives back to the objects. These simulators are thus limited to an open-loop control in nature. RTDS is very powerful

in terms of connecting actual devices to the simulation environment. But it is again limited to the libraries provided by the simulator and thus constrains the control application scenarios. Paladin Live allows real-time system monitoring and provides tools to analyze the system health. However, its simulation mode is for general power system design and is not able to do distributed control simulations.

Other studies on specific load models and their real time simulations also exist in the literature [94], but they fail to consider general and heterogeneous control cases, but rather concentrate on specific scenarios. In [95], the authors introduce a real-time combined power flow simulator and electromagnetic simulator, but the scenarios and the system are all static, i.e. flow of simulation is preset before run-time. In [96], Real Time Digital Simulator (RTDS) [37] has been used to simulate a fuel cell vehicle, where the operation is limited to the specific load scenario of a fuel cell vehicle.

As a summary, existing simulators have very powerful non-linear power flow solvers that can calculate the voltage drops efficiently in the physical system. Yet, they lack the ability and the interface to connect and test dynamic online scenarios, distributed control algorithms, reactive control algorithms and feedback based (closed-loop) control algorithms, representing the emerging cyber physical aspect of the smart grid. Furthermore, classical simulators fail to address time synchronization since the scenario is a static simulation. To answer all these missing points and still maintain the powerful aspects, we have developed S^{2}Sim.

4.3 S^{2}Sim Architecture

The classical power grid is a network of many different grid elements connected to each other over the electrical lines. This graph is mostly represented by an impedance matrix. We use this *physical* circuit as the basis of our architecture. But, with the emergence of Smart Grid, we need to add additional concepts on top of the physical electrical circuit in order to represent the resulting *CPS*.

The first concept we introduce with S^{2}Sim in the *object*. An object is the cyber/virtual representation of a physical circuit element. It controls the behavior

and the loadshape of the physical element it is representing. It is of crucial importance to represent these elements correctly for power system simulation [97]. The second concept is the optional component of *coordinator*. The coordinator is a completely virtual entity, which implements the feedback logic that will be present in the CPS. Figure 4.1 shows an example scenario for the overall architecture of S^2Sim.

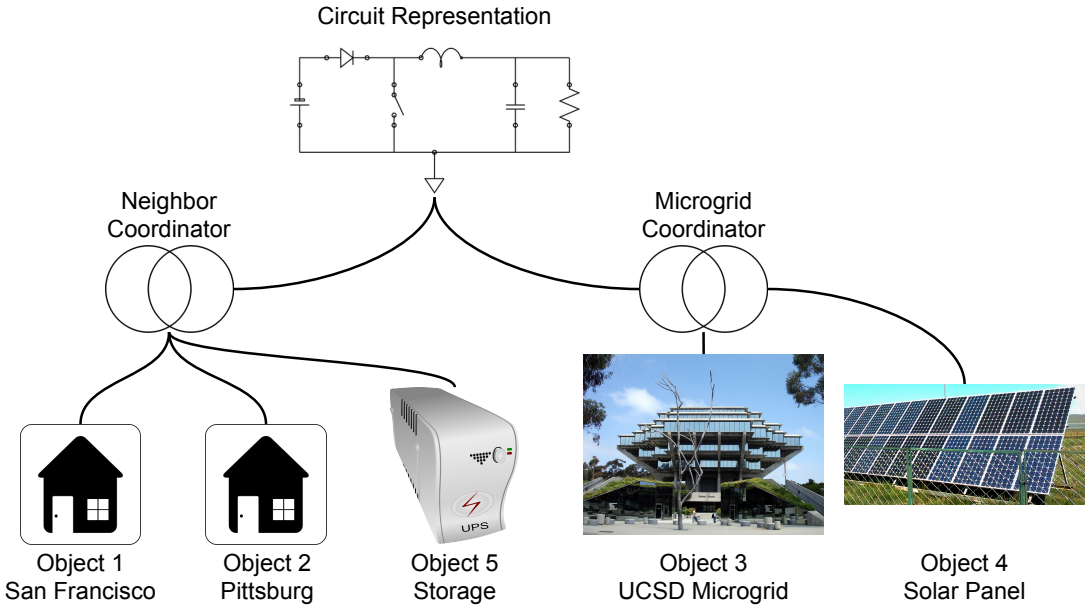

Figure 4.1: Example Architecture of 3 main elements

4.3.1 Object

An object is the cyber representation of any physical entity defined on the physical circuit. It can represent any type, such as loads, generators, energy storage devices or a combined system as a single entity. Objects can be of any size in the grid, ranging from a toast machine to a whole microgrid. Objects can be self-aware and implement distributed control algorithms to adjust their behavior, such as real-time consumption of a building, output of a solar panel, charging characteristic of a battery or the output of any simulation. Object behavior is controlled over a TCP/IP communication interface, which allows it to be virtually

anywhere. S^{2}Sim does not implement the behavior of the objects, but provides the communication framework the objects have to use in order to co-simulate their outputs. The minimum requirement for an object implementation is for the object to output consumption values, which is used to adjust the physical representation of it. There is no limitation on the frequency of the output as all time synchronization is done by S^{2}Sim.

4.3.2 Coordinator

A coordinator is a special virtual element that can oversee and get information regarding the whole or a part of the grid at any time. It is an optional component that provides feedback information to the objects, such as dynamic pricing or stability related sensor information. As an example, it can represent the grid perspective, a home control hub or a microgrid coordinator and serves as a feedback provider to its intended operation region. The coordinator constructs the missing link in a closed-loop control scheme, providing various feedback signals required for normal daily operation on different scales. The simplest of these is price. Each coordinator has a different strategy for different types of consumers. Another common signal is regulation incentive, which guides the consumption of participating customers by giving incentives. The specific implementation of a coordinator is external to S^{2}Sim , but requires a specific communication framework to connect to the simulation. The complex solution of the power flow equations is completely shielded from the coordinators and is handled by S^{2}Sim.

4.3.3 Electrical Circuit

The only physical and static part of the simulator is the electrical circuit itself. The circuit represents the networked connection of objects and is determined statically before the simulation starts. This component represents the classical power flow simulators. Any electrical grid item can be defined ranging from loads, generator, energy storage devices to transformers and circuit breakers. The electrical circuit is implemented within the simulator and is one of the essential

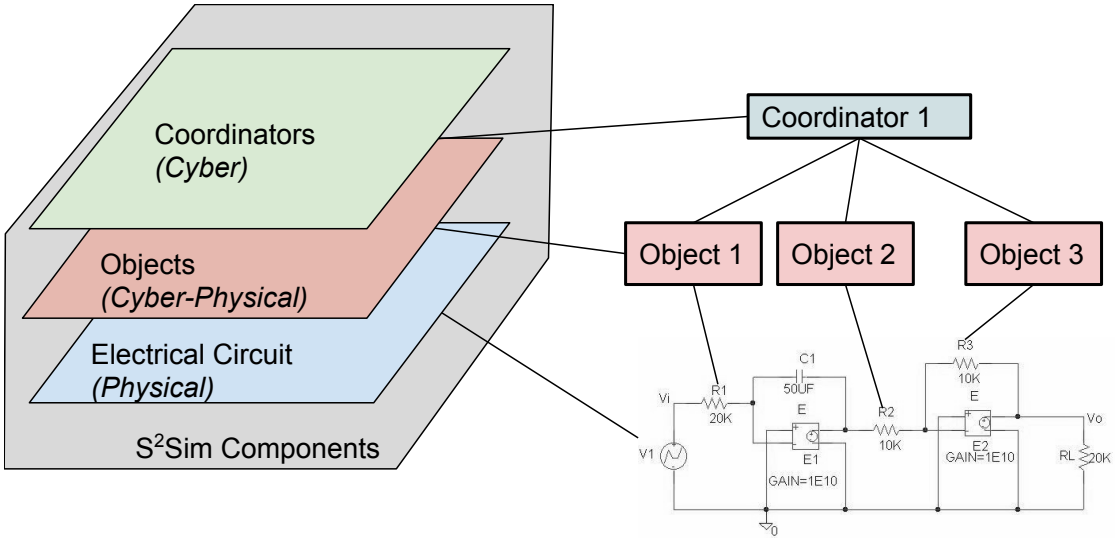

Figure 4.2: CPS Layers of S^{2}Sim.

components. The three layers of components of the composite simulated CPS is shown in Figure 4.2.

4.3.4 Internal Architecture

For modular structure and flexibility, the internal structure of S^{2}Sim has three major engines, corresponding to three major tasks: Communication, Time Synchronization and Power Flow Engines. Figure 4.3 shows how these engines interact with each other as well as the external components, namely objects and coordinators. At any time, the simulation's information flow starts from the objects into the simulator. The incoming information is parsed and processed through the Communication Engine and is supplied into the Time Synchronization Engine. The data is then time filtered and time synchronized, and passed to the Power Flow Engine in order to obtain the power flow solution for the current time interval. The Electrical Circuit component is modeled within the Power Flow Engine. All obtained information is then forwarded to the respective Coordinators for feedback calculation. Finally, any feedback is sent back to the Objects, closing the information flow loop. During the process, the information flow with the external components can happen asynchronously, as this is handled by the Time

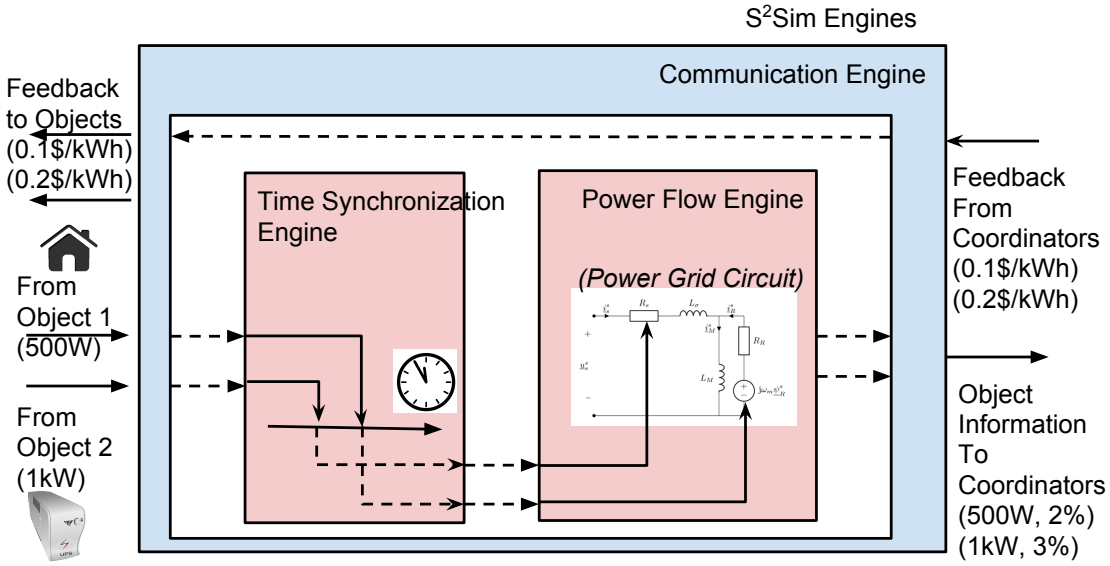

Figure 4.3: Functionalities of S²Sim divided among its internal components.

Synchronization Engine.

Communication Engine

Communication engine maintains end-to-end communication using TCP/IP between the objects and S²Sim. Since the exchanged data can be sensitive such as the consumption information of a residential building, we provide an optional security layer with end-to-end encryption. The encryption is performed using Secure Sockets Layer (SSL).

We create a common message structure as an application layer messaging protocol to regulate the communication between the simulator and the objects. The protocol is flexible, extensible, lightweight, low overhead and has minimal dependency on the underlying infrastructure. This protocol establishes the minimum framework required by the implementation of every object. Every communication command is represented by a separate packet. Examples to these commands are registration message, consumption reporting message or price notification message. Although each command has a unique internal structure, all messages have a common header and ending for message identification. Any control algorithm that wishes to be represented in the system, needs only to implement this communica-

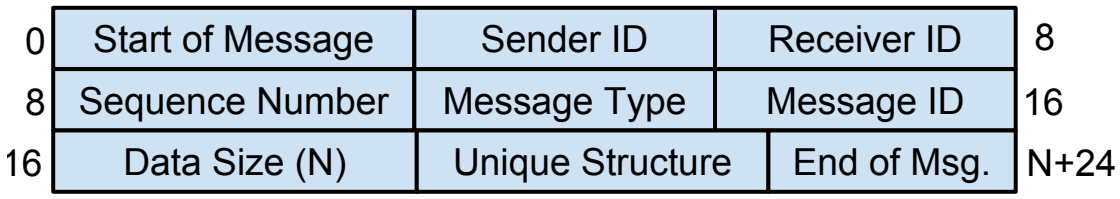

Figure 4.4: Common structure of S^{2}Sim messages.

tion protocol framework. The common structure is shown in Figure 4.4.

Start and End of Message: A communication protocol must be independent of the layers underneath it. These two fields mark the start and end of a single message. Since a periodic field may endanger the keys of the encryption, this field must be transmitted unencrypted.

Sender & Receiver IDs: Each object is assigned a unique identification number when it registers to the simulator. This ID will be used for every communication for end-to-end identification.

Message Type & Message ID: The unique structure of a message is decoded through a two-level hierarchy. Message Type defines the higher level (e.g. System Messages), whereas the Message ID value determines the lower level (e.g. Registration Message). The specific values are defined in the Interoperability Document of the simulator in a separate document, obtainable from the author.

Time Synchronization Engine

Multiple Objects with various behaviors imply a distributed sense of timing. They may have different time resolutions and time constants. Consider 2 objects: A phasor measurement unit (PMU) connected to a PV and a simulated office building with heating ventilating and air conditioning (HVAC). PMU provides high resolution, on the order of seconds, near real-time data and has a small time constant due to rapid solar variations. In contrast, HVAC simulation has low time resolution, on the order of hours, may provide simulated information for the future and has a large time constant due to the slow adapting nature of thermodynamics. Time Synchronization Engine enables both objects to connect in real-time

and be represented in the same simulation environment.

Time Synchronization Engine filters out past incoming data for the passive connections, stores future data and provides the current information for all connections. It integrates different resolutions by linear interpolation and time averaging for low and high resolutions respectively. It uses any future information as a prediction and provides it to the coordinator as an input. The prediction can be updated if the actual information changes as time advances. For the previous example, the PMU's measurements are processed in real-time, whereas the low-resolution information from the HVAC is interpolated to obtain the missing points compared to the high resolution PMU. To represent a broad scale of objects, S^{2}Sim defines 2 types of object connections:

Active Connection: The object is time synchronized to the simulator and provides real-time information or future prediction. In return, the object receives feedback information sent by the local coordinator. If the object fails to communicate within a time interval, previously sent information or prediction is used automatically.

Passive Connection: The object connects to the system, uploads bulk consumption data and disconnects. The bulk data is filtered and processed. But the coordinator does not provide feedback, as the object is disconnected and is assumed to be irresponsive to any feedback. This type of connection enables the connection of consumption databases or data sources requiring no feedback and is an easy way to represent an object without any control or automation.

Figure 4.5 explains the timing with an example. The passive object provides bulk consumption data for the whole simulation, whereas the active object is time synchronized and determines its behavior based on the feedback signals it receives at every time step.

Power Flow Engine

Since one of the purposes of S^{2}Sim is to provide abstraction of the power flow problem to the coordinators, the coordinator may require an additional "sandbox" or "playground" environment for power flow solutions. The Power Flow Engine

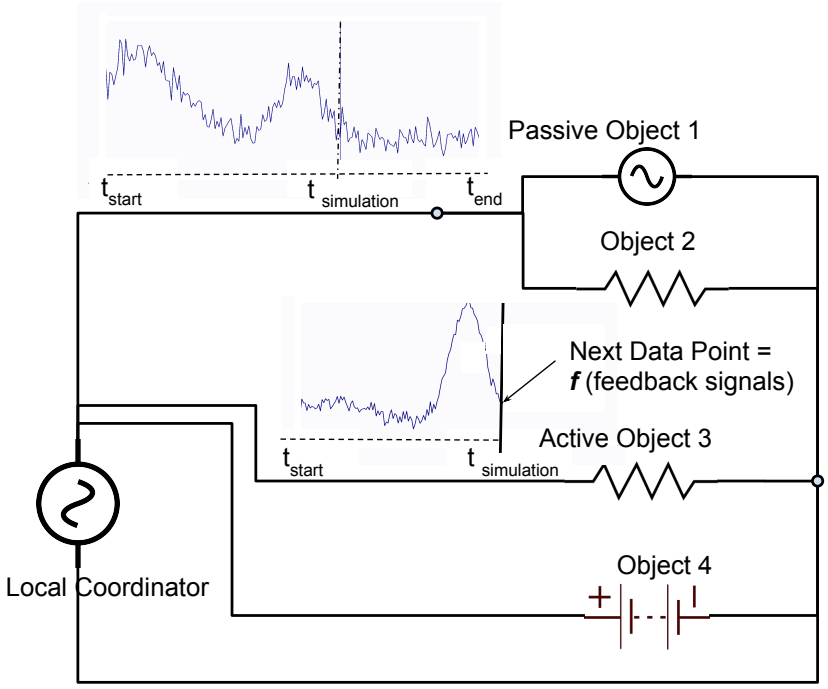

Figure 4.5: Example circuit with passive and active connections

uses the power flow solution interface of OpenDSS [35] over a DLL and constantly maintains two parallel instances of it. The real circuit handled on the first instance is modified only to reflect the actual behavior of objects and any modification represents actual snapshots of the physical circuit. The second instance has the exact same circuit at the beginning of each time interval as the real circuit, but is used as a sandbox to be modified and reset multiple times to answer different "What if" scenarios that the coordinator might be interested in.

4.4 Distributed Control Simulation

In this section, we demonstrate multiple case-studies how our simulator, S^2Sim, can be used to show how heterogeneous distributed control algorithms can affect each other and the grid. We first simulate an average sized U.S. town to show that a complete greedy distributed control of loads may lead to unstable conditions given static time of use (ToU) pricing. In response, we show that introducing adaptive pricing heuristic on the coordination side to guide the grid to stable

operating regions can avert this situation. In the second case, we use the test bed of a joint project between 6 universities [98] to test a distributed heterogeneous control scenario. Each university from different regions of the United States deploys its own control algorithm. S^{2}Sim combines and synchronizes all objects and provides a smart pricing heuristic from the coordinator to guide the grid to stable operating regions. In the third case, we use HomeSim [90], a residential energy simulator to simulate multiple houses in a neighborhood to test various control strategies.

4.4.1 Validation and Performance Overhead

The simulator has been validated against University of California, San Diego campus Microgrid measurements, by comparing measured and simulated voltage deviation information at building terminals.

To give an estimate for the communication overhead, we look at a sample problem size of 100.000 simultaneous objects. At each simulation time step, the default communication overhead is the consumption message from every object to the simulator and, a price and a regulation message from the simulator to every object. The messages are only 28 bytes in total. This results in $56N$ bytes of overhead for N objects in every time step. The default setting runs one time step per second, so for a circuit with 100000 simultaneous objects, this results in $5.6MB/s$ of communication overhead, easily maintainable with an everyday home network.

The processing overhead of the 3 main engines are as follows: Communication Engine has $O(N)$ message processing complexity for parsing and distributing messages. Time Synchronization Engine has $O(N)$ complexity for filtering and interpolation. Power Flow Engine has at least $O(N^3)$ due to the power flow solution. Extra overhead caused by S^{2}Sim besides the power flow solution is only $O(N)$.

4.4.2 Time of Use vs. Adaptive Pricing

We use a university campus distribution circuit with both residential and office buildings as the loads. The average total grid consumption is $10MW$, about the size of an average U.S. town with 81 buildings represented as individual objects.

Each object runs a distributed control algorithm, unaware of its surroundings or the grid and only uses the price signal provided by the utility to adjust its consumption. The distributed control algorithm of the objects is a greedy heuristic, which adjusts the consumption in proportion with the ratio of the average price to the current price. The remaining consumption is adjusted to fix the total energy consumption, in order to give a fair comparison among different pricing strategies. The algorithm at i^{th} step is given below:

$$\text{AdjustedPower}_i = \text{Power}_i \frac{\text{Avg(Price)}}{\text{Price}_i} \tag{4.1}$$

$$\text{Power}_j = \text{Power}_j + \frac{\text{AdjustedPower}_i - \text{Power}_i}{N-j+1}, \forall j \in (i, N) \tag{4.2}$$

This scheme is a simple heuristic assuming an energy storage device connected to the load, capable of reacting to price changes. We consider two pricing strategies: 1) Completely static pricing, open loop without feedback and distributed control case; 2) adaptive consumption, dynamic price guided, distributed closed control loop case.

Static pricing uses a ToU pricing scheme with 3 price regions dividing the day into 4 intervals representing peak, off-peak and super off-peak hours [83]. The price is static as it doesn't react to the state of the grid and is the same for every object. Adaptive pricing computes a dynamic price for each individual object. The heuristic uses the information of object's terminal voltage deviation as a stability metric, then multiplies it with the object's current consumption and maps the value to a price range. The heuristic not only penalizes high consumption, but also takes into account the voltage deviation, which is affected by every object in the grid. High deviation caused by any object thus has a higher price effect on all objects, yet the object that has caused the condition will have the highest penalty. To avoid rapid variations in pricing, we pass the immediate price values through an exponentially weighted moving average filter to smooth out the price decisions. We take the maximum voltage deviation within the grid as our stability metric and mark the widely accepted 10% value as the limit of danger and start of instability. Figure 4.6 shows that the result of combined greedy behavior under ToU pricing in a completely distributed scenario leads to unstable system behavior, pushing the voltage deviation beyond its safe limits. The initial spike is largely due to the

fact that the controllers are unaware of each other and react to the low price in a greedy manner.

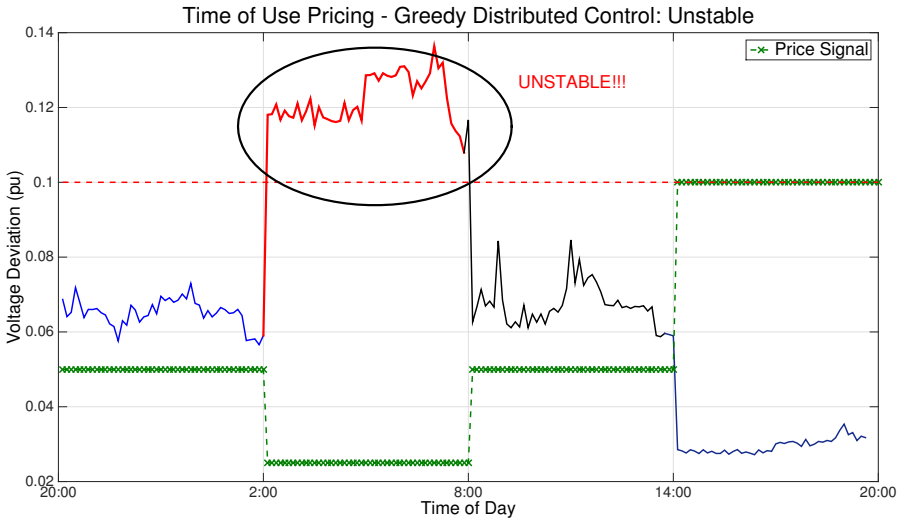

Figure 4.6: Time of Use Pricing resulting in unstable system behavior

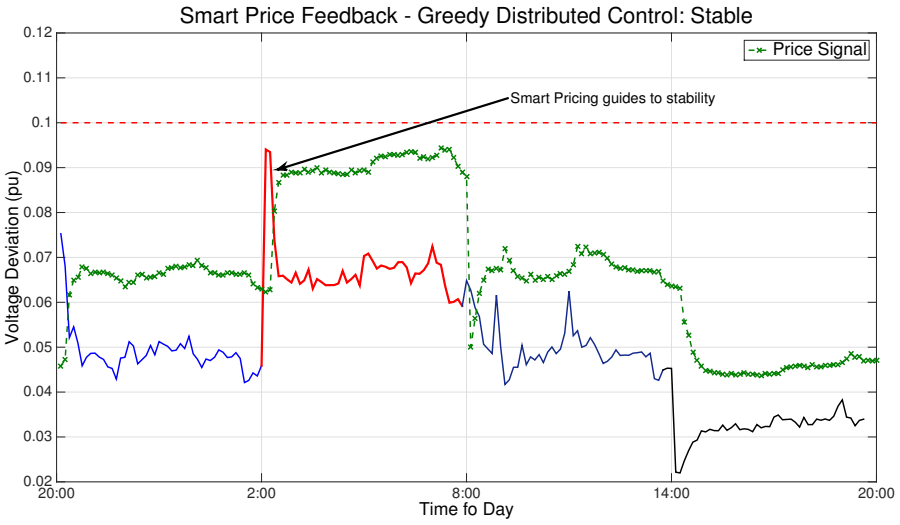

Figure 4.7: Adaptive Pricing resulting in stable system behavior

Figure 4.7 shows the results for the adaptive pricing scheme. As with previous results, there is a spike in consumption due to the greedy distributed control in the low price region. However, the price adapts to consumption and stability

values and, guides the system to be within stable boundaries to avoid instability. Although both control algorithms are simple heuristics, we show that good performance for a control algorithm under isolated conditions is misleading. S^{2}Sim enables each algorithm to be simulated within the whole grid, exploring cross-correlated effects in depth.

4.4.3 Distributed Heterogeneous Control

We use a university Microgrid circuit with 12 major buildings represented by a combination of real and simulated objects from 6 different universities. Their physical locations are in California, Michigan and Pennsylvania, connected remotely over TCP/IP to S^{2}Sim. We use home automation controller simulation [90], actual battery bank controller [99], real-time consumption of an actual building with actuation [100] and 3 different HVAC control simulations with different strategies [32][33][34], summarized in Figure 4.8.

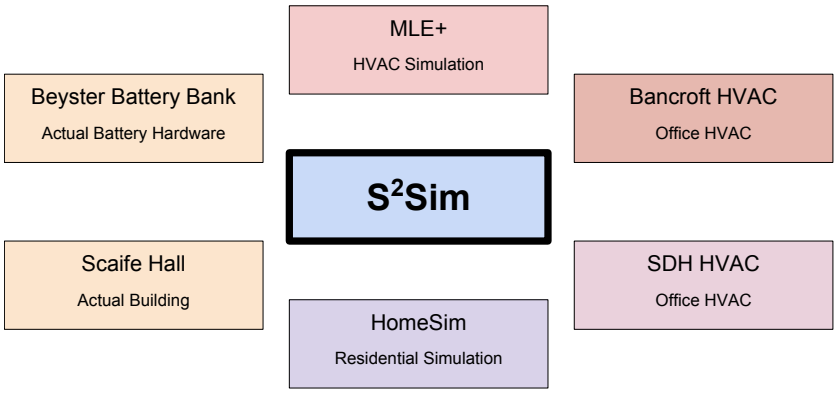

Figure 4.8: List of heterogeneous controllers in a multi-university collaboration

We use the same heuristic pricing as in the previous section. Figure 4.9 shows that the independent distributed controllers increase their consumption leading to increasing voltage deviation (solid) within the system, endangering the system health by coming close and exceeding the 10% limit (horizontal). The adaptive price (dashed) increases to guide the system back into the stable region and later achieves it. This study shows that, well performing algorithms in isolated situations, may lead to an unstable system, when working together.

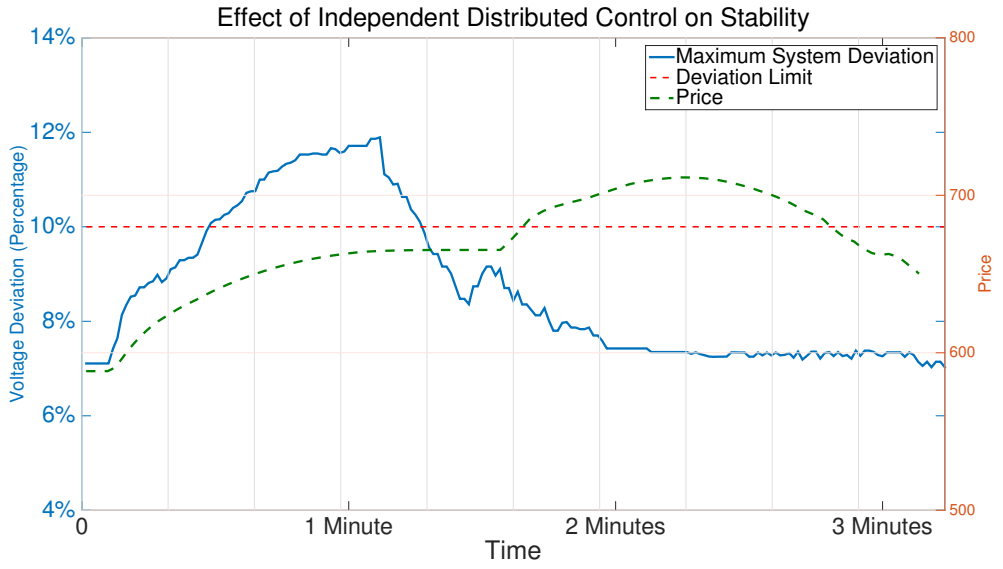

Figure 4.9: Effect of Distributed Heterogeneous Control on system stability

4.4.4 Neighborhood Simulation

To further demonstrate the abilities of our simulator, we extend the first case study for time of use pricing by eliminating the coordinator entirely and randomizing the consumption intervals in order to distribute the total consumption over time. We consider a residential neighborhood with 160 buildings. Consumption values are obtained from a residential simulator called HomeSim [90]. To decrease the probability of a high consumption correlation, each building selects a random shifting amount without any further knowledge and shifts its consumption value by the selected value in time. The random value is a uniformly distributed value drawn from three different intervals for the three cases considered: 1) $[0, 1]$, 2) $[0, 2]$, 3) $[0, 3]$ hours. Furthermore, we use the two algorithms used in the first case, where the buildings implement greedy distributed control and the coordinator is providing static and dynamic pricing feedback. 50 iterations have been averaged to get stable results. Figure 4.10 shows the results for all 5 control algorithms considered. The maximum observed deviation values are shown in Figure 4.11. The only algorithm that fails the voltage deviation limit is the greedy control case with static pricing as in the first case. Active feedback manages to keep the sta-

bility within the limit as in the first case. The additionally tested randomization algorithms without feedback manage to decrease the deviation and the peak is decreased by increased randomization.

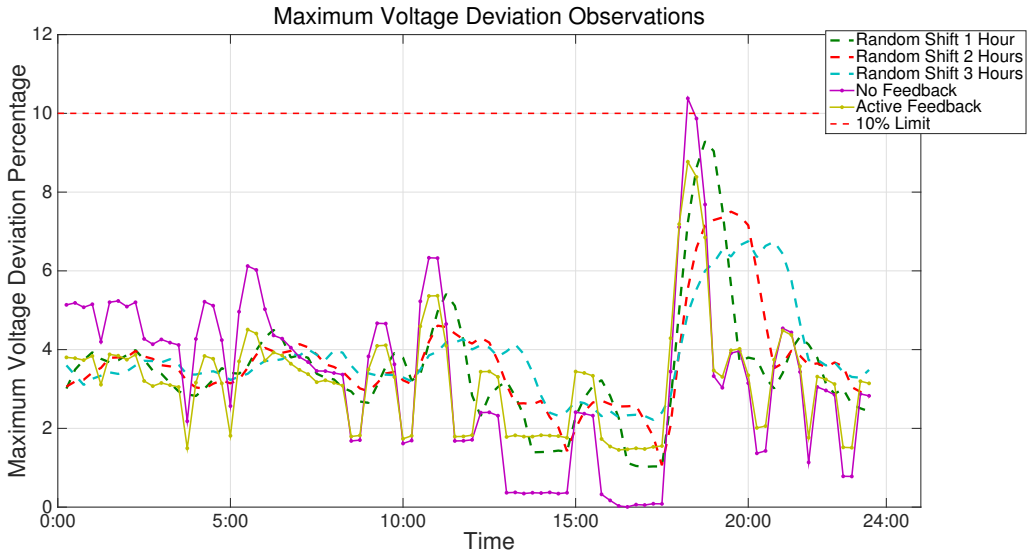

Figure 4.10: Neighborhood case study with distributed heterogeneous controls

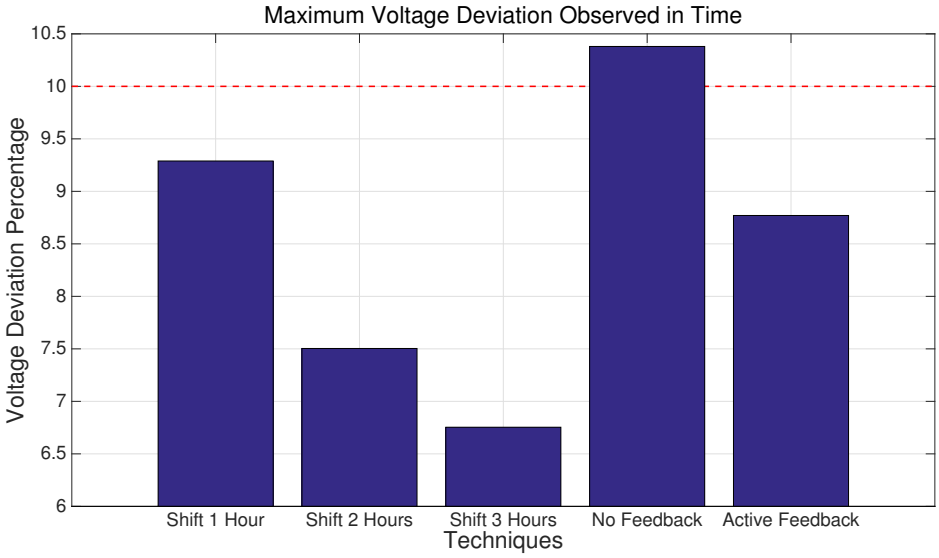

Figure 4.11: System stability of the neighborhood case study

4.5 Conclusion

The classical power grid is transforming into a cyber physical system, the smart grid. Distributed control algorithms for different platforms are being developed and deployed in different scales. In Chapter 3 we presented a distributed battery control solution, producing dynamic actuation outputs at different locations of the grid. In the design of distributed control algorithms, it is a common assumption that the underlying grid structure to remain unchanged and stable with the actuation decisions. However, the effect of the distributed generation and distributed actuation decisions must be studied in the context of the grid. Existing grid simulators solve the power flow of the physical aspect of the grid efficiently, but fail to address the co-simulation of distributed control algorithms, thus the CPS aspect of the smart grid. There is a need for a flexible simulator to co-simulate and test independent distributed control algorithms in order to observe their effects on both each other and the health of the system. To answer this need, we have developed S^2Sim. S^2Sim allows the co-simulation of any object connected over TCP/IP, which can represent any type and any size of grid elements, with distributed independent control strategies. S^2Sim takes care of communication, time synchronization and introduces an interface for multiple coordinators to construct closed loop feedback controlled system. S^2Sim is extensible, scalable and has low overhead. We present 3 different case studies specifically possible with our simulator, where the first case shows, why it is necessary to have closed loop control for grid stability. The second case shows that we cannot justify the performance of a control algorithm under isolated conditions alone, without testing it within the grid picture. The third case shows that we can use S^2Sim to compare the performance of different heuristics.

All chapters so far require an efficient and fast communication infrastructure. Our forecasting algorithm in Chapter 2 requires timely delivery of current sensor information to operate. Our optimal battery control algorithm in Chapter 3 requires quick delivery of negotiation information for fast convergence and stable operation. In Chapter 5, we present a congestion-aware optimal packet aggregation algorithm that enables optimization of individual stream goals, such as

information freshness or delay. Using our solution, slowly changing sensor information can be sent as best-effort, time critical negotiation iteration information can be optimized for maximum freshness and quickly changing sensor data such as voltage measurements can be transmitted with minimum delay. Our solution finds the optimal solution for the whole network considering the congestion level in the network.

This chapter contains material from Alper Sinan Akyurek, Baris Aksanli and Tajana Simunic Rosing, "S2Sim: Smart Grid Swarm Simulator", International Green and Sustainable Computing Conference (IGSC), 2015. The dissertation author was the primary investigator and author of this paper.

Chapter 5

Optimal Packet Aggregation in Wireless Networks

One of the most critical emerging problems for 5G and Internet of Things is the handling of machine-to-machine communication including the smart grid. Wireless sensor networks are deployed every day, resulting in a more distributed infrastructure, where the communication and processing are handled by energy, bandwidth, and processing constrained devices. Aggregation of multiple packets flowing over the same path increases spectral efficiency, energy efficiency, and resource utilization. In this chapter, we address the problem of determining the optimal waiting time to maximize the utility within the network. We provide a general framework, where the utility function can be user-defined for each individual application stream and packet. This allows the user to optimize for energy, delay or expiration rate in the resolution of individual streams. Our algorithm calculates the optimal time for any given condition on-the-fly and can adapt to changing conditions with low computational complexity. We also provide an optimal multi-hop distributed and scalable under congestion versions of our algorithm. Our simulations in ns3 show that we outperform state-of-the-art policies by 1.55x in terms of information freshness. Our solution reduces average power consumption by more than 60%. Our congestion-aware solution shows constant performance with increasing congestion levels, whereas the state-of-the-art solutions degrade by up to 70% under the same conditions.

5.1 Introduction

The 5th generation of mobile networks (5G) and the Internet of Things (IoT) are the next major phases in the research area of wireless telecommunications. Even though 5G does not have an established standard as of yet, the vision for 5G has the concept of connectivity anywhere for anyone and anything, including the machine-to-machine communication [101]. One of the ongoing projects for 5G by the European Union is METIS-II [102]. METIS envisions three service types, two of which are machine-to-machine communication: 1) reliable, low latency communication, 2) low-energy, scalable communication. IoT and Wireless Sensor Networks (WSNs) have much in common with the second service type. However, this vision shows that it is important to keep communication management protocols and policies as flexible and as dynamic as possible to address the ever-changing scenarios. Most current work focus on the waveform design of 5G [103, 104] to accommodate the higher link speeds and massive multi-input multi-output (MIMO) communications. Transport level improvements in 5G stay as a relatively understudied area. In this chapter, we consider the technique of in-network packet aggregation on the transport level to improve the efficiency and effectiveness of resource utilization within the whole communication stack.

In-network packet aggregation is a simple concept that combines multiple packets, possibly from different sources with heterogeneous requirements and characteristics, into a single packet to be transmitted to the next node. This technique inherently saves energy consumption by transmitting fewer packets, increases efficiency by eliminating recurring lower layer headers and indirectly lowers latency by creating less interference within the shared spectrum. How these metrics are affected is a direct function of when the policy decides to transmit the aggregated packet. If it waits for a long period, more packets are aggregated together, however the latency of each packet increases during this process, which might be undesirable for the application. On the other hand, if the policy waits only for a short period, fewer packets are aggregated and the gains of the packet aggregation technique are diminished. The optimal value of the transmission instance depends on multiple factors, ranging from individual application streams to network level conditions.

In this chapter, we provide an optimal packet aggregation algorithm that allows individual tuning on an application stream level, packet level and network-wide level using a concept of *gain functions*. Gain functions describe a positive metric that the designer wants to maximize. It is a mathematical function and as long as it is concave and positive, our solution is guaranteed to converge to the optimal aggregation time without knowing the details of the function formulations.

The concept of the gain function also allows us to adjust the solution to different requirements without having to re-design the algorithm making our solution extensible to future designs. Our algorithm calculates the optimal result numerically with a low computational complexity of $O(A)$, where A represents the number of application streams on a node. We provide special consideration and specialized solutions for a congested 5G Radio Access Network (RAN) and a multi-hop Wireless Sensor Network (WSN).

5.2 Related Work

Packet aggregation, data aggregation and data fusion have been studied from multiple aspects in the literature. In [39], the authors have shown that packet aggregation increases the efficiency of Transfer Control Protocol (TCP) streams, improves fairness and reduces end-to-end delays. Similar advantages were also observed in [105], where the authors show that data aggregation in a Voice over IP (VoIP) system decreases jitter and increases the number of concurrent calls in the network. One of the most popular and earliest data aggregation solutions is LEECH [106], in which the time division based Media Access Control (MAC) protocol creates clusters to aggregate data. The authors provide simulations to show the tradeoff between the number of clusters and performance. A more recent work in [107] approaches the aggregation problem from the sensor's perspective, in which the sensor aggregates multiple measurements into a single transmission to save energy. The sensor produces a single type of data stream with known characteristics and the solution is optimized for a particular platform. The work in [42] focuses on using duty-cycling at multiple hierarchical layers in the network

to aggregate data and save energy. However, in our previous work [5], we showed that if the data streams are constrained by deadlines, the energy per successful data bit is constant. This means that any improvement in energy consumption will lead to an equal degradation in the number of expired information.

Another class of solutions attack the problem from the routing perspective, either through structuring the network into clusters around aggregation points or structure-free dynamic solutions. A survey [108] provides a good overview on multiple algorithms for both cases. It discusses that structure based solutions perform well in static conditions, whereas dynamic solutions consume more energy but are more sustainable in dynamic environments. Most solutions consider only a single type of application stream in the network. The authors in [41] approach from the multi-hop periodic transmission perspective, providing two timeout based algorithms, focusing on both energy consumption and delay. In their results, they show the importance of the timeout value and how it effects the performance. In [109], the authors use Markov Decision Process (MPD) to model the aggregation problem and the use genetic algorithm to solve it numerically. For a comprehensive survey, see [40].

In the current state of the literature, most solutions focus on a single application data stream, dealing with a homogeneous environment and optimize for a single performance metric, mostly energy consumption. Furthermore, aggregation algorithms are designed focusing on WSNs. However, applications may have heterogeneous characteristics, requirements and optimization metrics. Recently the very high number of machine-to-machine communication vision was introduced also into the cellular networks with the 5th generation, such that there is a need for a more flexible aggregation solution. In this thesis, we provide an optimal aggregation solution that can be tuned on 1) the individual stream level, 2) individual packet level and 3) network level; allowing for a flexible, scalable and optimal solution that supports heterogeneous applications as opposed to the single application WSN mindset in the literature.

5.3 Optimal In-Network Packet Aggregation

In this section, we provide the description of the system that we are considering and the definition of the optimal aggregation problem.

5.3.1 System and Problem Definition

We envision a network with a single data sink and multiple nodes communicating with the sink, possibly over multiple hops. Multiple applications on each node generate streams of data. Each application is associated with a deadline process and a generation process. The deadline process determines the lifetime of each individual chunk of data within the data stream and can be stochastic. The arrival process determines when and how many chunks of data are generated for a specific stream. The network setup can also be generalized into multiple sinks. The routing algorithm would handle the division of the network into individual trees with a single sink node at the root. A cross layer optimization of network and transport layers is also possible, however it is out of the scope of this thesis.

Each application stream has a gain function that describes the virtual gain of a specific chunk of data. Gain is a user-defined metric that is non-decreasing with the delay of a data chunk. For example, a sensor board is equipped with a temperature and an intrusion sensor. It is crucial that the intrusion information reaches the sink before the tight deadline and the outcome of the decision is binary (arrived on time or expired), whereas the temperature information should be as fresh as possible. The user assigns a linear information freshness gain function to the temperature stream and a binary expiration gain function to the intrusion stream.

The only requirement for a gain function is that it should be positive and concave within the operating region of possible transmission instances. The gain function has three inputs: 1) waiting time after generation, 2) expected deadline, 3) network related information. The maximum value of the gain function is expected to be unity, however the freedom to increase the importance of a specific stream is also given to the application designer by assigning a higher-than-unity gain

function. The gain function is only an internal metric and is not transmitted within the packet. It is used by the policy to determine the optimal transmission instance.

All application stream gain functions are accumulated and multiplied by the packet gain function that specifies the significance of an individual packet. The task of the policy is to maximize the aggregation of all packets within the network by determining the transmission times of each node. The system architecture of a single node is described in Figure 5.1.

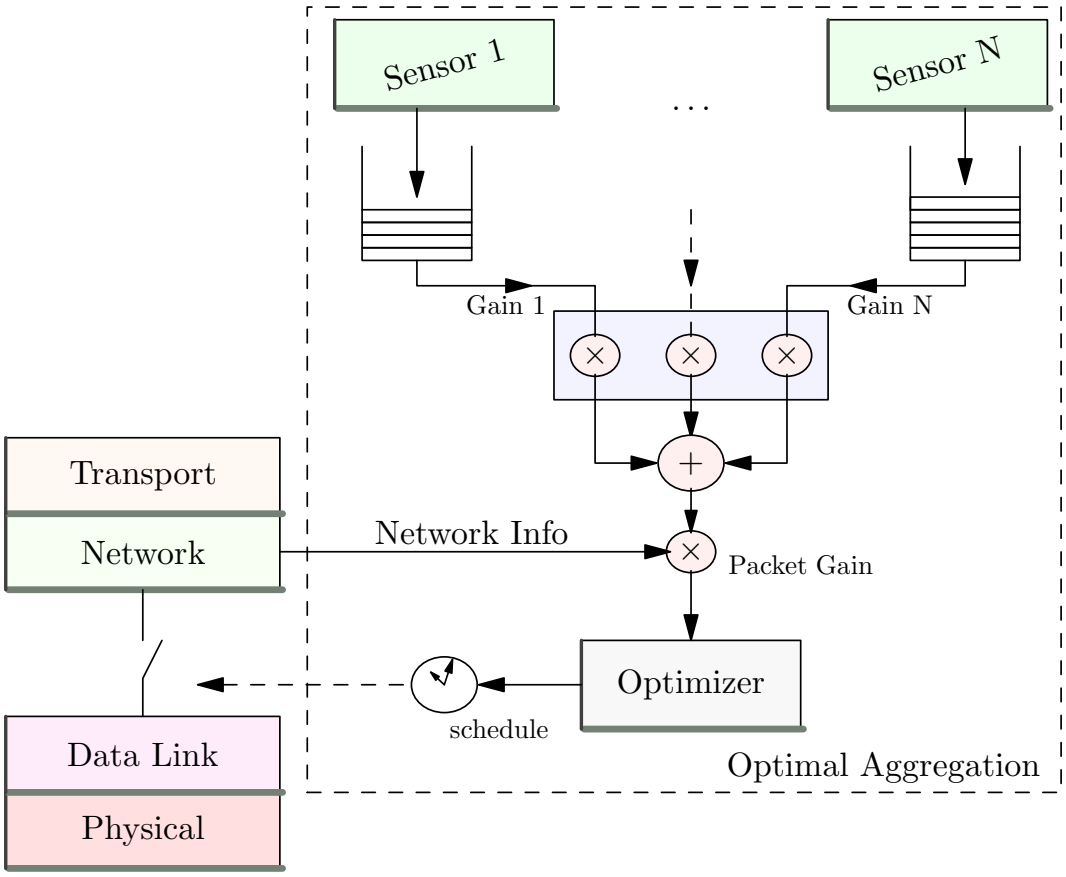

Figure 5.1: Architecture of our solution within a single node.

The figure shows how the individual stream gains, the packet gain and network information are used for the optimization to decide on the transmission schedule. The network is formed by multiple nodes containing our solution. Each node has multiple application streams with heterogeneous characteristics and de-

mands. Each application has a deadline and a generation process. The significance of the application is determined by a gain function. The aggregated packet is then further multiplied by the packet gain function for global optimization. Finally, the optimizer finds the optimal aggregation time and reports it to the lower layers.

When the policy decides to transmit, it aggregates all generated information into a single packet. This aggregation can be implemented in multiple ways. Internet Protocol (IP) tunneling is the most straightforward technique to achieve aggregation, retaining all original information. However, it is also the most inefficient way. IP header compression is an additional layer that would improve the efficiency significantly as most of the headers are the same as they share the same destination sink. A completely novel protocol is also possible to maximize the efficiency. The only required information is the source ID, application stream ID and the data itself. The actual implementation is out of the scope of this paper.

We define the number of nodes in the network by N. Any node i has A_i number of application streams requiring communication. Each application j has a probabilistic generation process, where the probability of generating k packets, t seconds from the decision time is defined by $p_{i,j,k}(t)$. Each application stream also has a random deadline process that generates the deadline D with a probability of $q_{i,j}(D)$. The transmission is planned to occur x_i seconds from the decision time, which is the optimization variable to be determined. The gain of each stream is determined by $s_i(D, x_i, t, y)$, where y defines the network conditions. The network conditions are defined differently depending on the type of the network to be explained in this section. The gain of the packet is defined by the function $g_i(x_i, y)$.

We can write the total gain (G) in the network as:

$$G(\overline{x}, y) = \sum_{i=1}^{N} g_i(x_i, y) \sum_{j=1}^{A_i} \sum_{k=0}^{\infty} \int_{t=0}^{x_i} \int_{D=0}^{\infty} k p_{i,j,k}(t) \, [\\ q_{i,j}(D) s_i(D, x_i, t, y) dD dt] \tag{5.1}$$

We simplify this using the following assumptions:

- **Time independent gain function:** We assume that the gain function for each application stream is not an absolute function of time (t), but a function

of time elapsed since generation until transmission $(x_i - t)$. This results in $s_i(D, x_i, t, y) = s_i(D, x_i - t, y)$. To the best of our knowledge, most metrics, such as delay, freshness or expiration, use a relative time since generation, rather than the absolute time.

- **Stationary generation process:** We assume that the generation process is stationary within the aggregation waiting time period, removing the dependence on time. This results in $p_{i,j,k}(t) = p_{i,j,k}$. If the generation process is not stationary, the integral over t can be further divided into smaller intervals as separate smaller processes, such that they can be assumed stationary within the smaller time scale.

Furthermore, we define the expected generation rate as $\lambda_{i,j} = \sum_{k=0}^{\infty} k p_{i,j,k}$. Using these assumptions and definitions, we can rewrite the total gain as:

$$G = \sum_{i=1}^{N} g_i(x_i, y) \sum_{j=1}^{A_i} \lambda_{i,j} \int_{t=0}^{x_i} \int_{D=0}^{\infty} q_{i,j}(D) s_i(D, x_i - t, y) dD dt \tag{5.2}$$

Changing variables $t' = x_i - t$ in the outer integral to simplify the expression, we obtain:

$$G = \sum_{i=1}^{N} g_i(x_i, y) \sum_{j=1}^{A_i} \lambda_{i,j} \int_{t'=0}^{x_i} \int_{D=0}^{\infty} q_{i,j}(D) s_{i,j}(D, t', y) dD dt' \tag{5.3}$$

We combine the expectation and the outer integral into a single expected gain expression, denoted by $S_{i,j}(x_i, y)$:

$$G(\overline{x}, y) = \sum_{i=1}^{N} g_i(x_i, y) \sum_{j=1}^{A_i} \lambda_{i,j} S_{i,j}(x_i, y) \tag{5.4}$$

We want to select the transmission times that would maximize the total gain expression. To find the optimal values, we take the Jacobian of the total gain. The result for an arbitrary node m is given as:

$$\begin{aligned} \frac{\partial G}{\partial x_m} = J_m &= g_m(x_m, y) \sum_{j=1}^{A_m} \lambda_{m,j} S'_{m,j}(x_m, y) \\ &+ g'_m(x_m, y) \sum_{j=1}^{A_m} \lambda_{m,j} S_{m,j}(x_i, y) = 0 \end{aligned} \tag{5.5}$$

The gain functions must be positive: $g > 0, S > 0$ and concave: $g'' < 0, S'' < 0$. In order for a solution to exist, the Jacobian must be zero, $J_m = 0$. Since both gain functions, g & S are positive by design, only one of the multiplied terms must be negative:

$$\sum_{j=1}^{A_m} \lambda_{m,j} S'_{m,j}(x_m, y) < 0 \text{ or } g'_m(x_m, y) < 0 \tag{5.6}$$

This can also be represented with a single multiplication, as only one of them should be negative:

$$g'_m(x_m, y) \sum_{j=1}^{A_m} \lambda_{m,j} S'_{m,j}(x_m, y) < 0 \tag{5.7}$$

We take the Jacobian again to prove that if a solution exists, it is unique and it is the global maximum:

$$\begin{aligned} \frac{\partial^2 G}{\partial x_m^2} &= g_m(x_m, y) \sum_{j=1}^{A_m} \lambda_{m,j} S''_{m,j}(x_m, y) \\ &+ g''_m(x_m, y) \sum_{j=1}^{A_m} \lambda_{m,j} S_{m,j}(x_i, y) \\ &+ 2g'_m(x_m, y) \sum_{j=1}^{A_m} \lambda_{m,j} S'_{m,j}(x_i, y) \qquad (5.8) \\ \frac{\partial^2 G}{\partial x_m, x_n} &= 0 \qquad (5.9) \end{aligned}$$

Note that the matrix is non-zero only on the diagonal. The diagonal elements consist of three summations as shown above. The first summation, $g_m(x_m, y) \sum_{j=1}^{A_m} \lambda_{m,j} S''_{m,j}(x_m, y)$, is always negative as $g > 0$ and $S'' < 0$ by design. The second summation, $g''_m(x_m, y) \sum_{j=1}^{A_m} \lambda_{m,j} S_{m,j}(x_i, y)$ is also always negative as $S > 0$ and $g'' < 0$ by design. The sign of S can be allowed to be negative if $g'' = 0$. The final summation was also found to be negative in Equation (5.7), which is the condition for the existence of the solution. As a result, we have a negative definite matrix and if the solution exists, this proves that the solution is unique and is the global maximum.

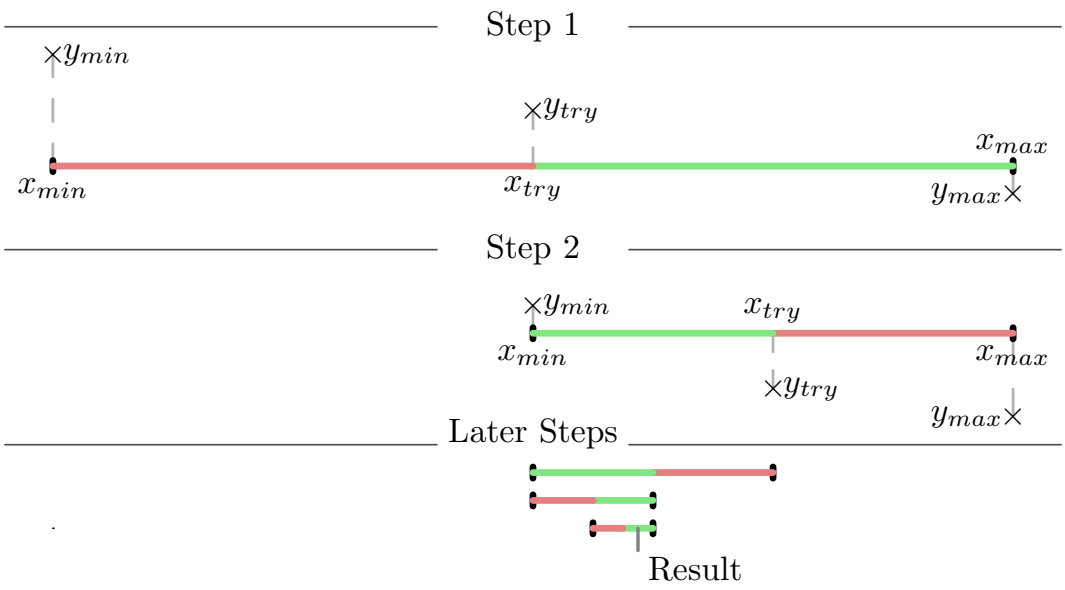

Figure 5.2: An example for the Bisection Method. The interval is divided into half at each iteration, making the search interval smaller every time. The algorithm stops, when the desired solution resolution is achieved.

Now that we have proven that the solution to Equation (5.5) is the global maximum and is unique, we need a method to solve it numerically. Since there is only a single solution, Equation (5.5) must cross the axis only once, with one side having negative values and the other side having positive values by the Mean Value Theorem. So, we employ the bisection method to obtain the solution numerically.

The algorithm is explained in Figure 5.2. The bisection method searches a predetermined interval by dividing the interval in half at every iteration. The operating region is defined by the interval $[0, x_{max}]$. The solution either exists at the boundary points $J_m(0) = 0, J_m(x_{max}) = 0$ or inside the interval. We assign $x_{max}/2$ as a new boundary point and replace one of the old boundary points with which it shares the same sign. This way, the interval gets smaller by 50% at each iteration until the desired solution resolution is achieved. The convergence rate of the method is $log_2(x_{max}/\text{Resolution})$.

Common Gain Function Derivations

For convenience we provide the derivation and analysis of commonly used metrics that are also used in the simulations section of this chapter. Our solution is not limited to these gain function and any type of gain function can be used. For other gain functions, the user needs to provide the expected gain function and its derivative in order for our solution to obtain the optimal aggregation time.

Delay: Delay is a commonly used metric that quantifies how much time has elapsed since the generation of the data. However, since gain has to represent a positive quantity, we consider the inverse sign of it. The time since generation was defined by t', so we calculate the gain of aggregation time x as:

$$S(x) = \int_{t'=0}^{x} \int_{D=0}^{\infty} q(D) - t'^{-1} dD dt' = -x^2/2 \tag{5.10}$$

$$S'(x) = -x \tag{5.11}$$

$$S''(x) = -1 \tag{5.12}$$

Delay is independent of the deadline, so the D integral vanishes to 1. The result is negative and its second derivative is always negative. It achieves its maximum at $x = 0$, the asymptotic result telling us that the minimum delay is achieved by transmitting as fast as possible. This gain function can only be used for a packet gain function selection with $g'' = 0$.

Expiration Ratio: Since each application has a deadline assigned to it, if the data waits for a long period of time, the deadline can be violated resulting in the expiration of the data. Like delay, expiration is a negative quantity, so we consider the success or non-expiration. The only non-expired data are those with

deadline D, greater than the remaining time until transmission:

$$S(x) = 1 - \int_{D=0}^{x} q(D)(1 - D/x)dD \tag{5.13}$$

$$S'(x) = -x^{-2} \int_{D=0}^{x} q(D)DdD \tag{5.14}$$

$$S''(x) = 2x^{-3} \int_{D=0}^{x} q(D)DdD - x^{-1}q(x) \tag{5.15}$$

The expression depends on the probability distribution, but is always non-negative. This gain function maximizes the success ratio of any measurement. It is also possible to maximize the number of non-expired data in a single transmission, which simply involves multiplying the expiration ratio by the total aggregation time:

$$S(x) = x + \int_{D=0}^{x} q(D)(D - x)dD \tag{5.16}$$

$$S'(x) = \int_{D=x}^{\infty} q(D)dD \tag{5.17}$$

$$S''(x) = -q(x) \tag{5.18}$$

This expression is always non-negative and the second derivative is negative.

We have previously shown [5] that the energy consumption per non-expired data is constant, so that any improvement on energy consumption results in an equally worse expiration rate, and vice-versa. We use our general gain function framework to prove this result.

We utilize the expiration gain function for the streams' gain functions. We make no assumptions regarding the network topology, the number of application streams or their rates. Our objective is to minimize the energy per deadline satisfied data ratio or to maximize the ratio of data satisfying deadlines per energy consumption. Each packet transmission is assumed to have a constant duration, thus consumes constant energy per transmission, we set the packet gain function

to be x^{-1}. First, we obtain the optimal aggregation time using Equation (5.5):

$$\sum_{j=1}^{A_m} \lambda_{m,j} \int_{D=x_m}^{\infty} q_{m,j} dD = 0, \forall m$$

$$= \sum_{j=1}^{A_m} \lambda_{m,j} \left[1 - \int_{D=0}^{x_m} q_{m,j} dD \right] \rightarrow \int_{D=0}^{x_m} q_{m,j} dD = 1 \tag{5.19}$$

The second step in Equation (5.19) is obtained by the sum of all probabilities being equal to 1 and due to the fact that all λ values are positive. This equation dictates that the transmission instance should be selected such that the probability of expiration is non-zero. This optimal result is then used in the actual not-expired data per energy expression:

$$\sum_{i=1}^{N} x_i^{-1} \sum_{j=1}^{A_i} \lambda_{i,j} \left[x_i + \int_{D=0}^{x_i} q_{i,j}(D - x_i) dD \right]$$

$$= \sum_{i=1}^{N} x_i \sum_{j=1}^{A_i} \lambda_{i,j} + \sum_{i=1}^{N} \sum_{j=1}^{A_i} \lambda_{i,j} \int_{D=0}^{x_i} q_{i,j} D dD$$

$$- \sum_{i=1}^{N} x_i \sum_{j=1}^{A_i} \lambda_{i,j} \int_{D=0}^{x_i} q_{i,j} dD = \sum_{i=1}^{N} \sum_{j=1}^{A_i} \lambda_{i,j} E(D_{i,j}) \tag{5.20}$$

The resulting maximum successful data bits per energy consumption is independent of the value of the transmission time. This result proves that the successful data per unit energy is only maximized if the transmission instances are set to guarantee that data will not expire. Any transmission instance selection in this region has no effect on the successful data per unit energy metric, therefore any improvement in energy or expiration rate results in an equal degradation in the other. This is the main reason why we have considered the new metric of information freshness.

Information Freshness: Information freshness defines how much time a data chunk has left until its deadline. We preferred this metric over the expiration rate as it gives a better resolution in a linear scale, compared to the binary output

of expiration. It is a positive quantity and the gain function is obtained as:

$$S(x) = E(D)x - \frac{x^2}{2} \tag{5.21}$$

$$S'(x) = E(D) - x \tag{5.22}$$

$$S''(x) = -1 \tag{5.23}$$

$E(D)$ is the expected deadline value. If the policy waits for a short amount of time, the amount of data inside the packet gets smaller, even though each data chunk is very fresh. If the policy waits for a long amount of time, the amount of data in the packet grows, but the freshness also drops as the waiting time nears the deadlines. The expression has a maximum at $x = E(D)$.

Other Gain Functions: The user can select any gain function as long as the gain function requirements of non-negativity and concavity are satisfied. The algorithm ensures that the optimal transmission instance is selected. Packet gain functions can be adjusted to boost the importance of desired nodes. As an example, it can be selected as depleted amount of energy to increase the importance of nodes with low battery levels.

The network information can be used in a number of ways. In multi-hop networks, it can be used to add the additional delay of forwarding the packets. In crowded networks, it can be used to introduce the additional delay due to access times or to increase the importance of desired nodes. More details are presented next.

5.3.2 Optimal Policy for 5G Radio Access Network

In the previous section, we have introduced the main framework of our algorithm, proved that the solution is unique. We provided a fast method to obtain the solution without any detailed information about the gain functions with proven convergence properties. However, we have not used the network information present in all gain functions to adapt the solution to the changing conditions of the system. In this section, we describe how the current framework can be specialized for 5G RAN cells.

We analyze a single cell of a base station, where all nodes are connected to the base station and the network is at a single hop distance. We modify our total gain expression so that the network parameter in the packet gain function depends on all transmission instances and the application gain functions are independent of the network. We justify this assumption by the fact that the transmission instances directly affect the congestion in the air by a per-packet basis, rather than per-application basis. The total gain expression becomes:

$$G(\overline{x}) = \sum_{i=1}^{N} g_i(\overline{x}) \sum_{j=1}^{A_i} \lambda_{i,j} S_{i,j}(x_i) \tag{5.24}$$

The Jacobian of the new expression is:

$$\begin{aligned} \frac{\partial G}{\partial x_m} = J_m &= g_m(\overline{x}) \sum_{j=1}^{A_m} \lambda_{m,j} S'_{m,j}(x_m) \\ &+ \sum_{i=1}^{N} \partial_m g_i(\overline{x}) \sum_{j=1}^{A_i} \lambda_{i,j} S_{i,j}(x_i) \end{aligned} \tag{5.25}$$

We use ∂_m as an abbreviation for the partial derivative with respect to x_m. We assume that the packet gain function for any node is the same to establish fairness between the nodes:

$$\begin{aligned} \frac{\partial G}{\partial x_m} = J_m &= g(\overline{x}) \sum_{j=1}^{A_m} \lambda_{m,j} S'_{m,j}(x_m) \\ &+ g'(\overline{x}) \sum_{i=1}^{N} \sum_{j=1}^{A_i} \lambda_{i,j} S_{i,j}(x_i) = 0 \end{aligned} \tag{5.26}$$

If a solution exists, it is obtained by solving:

$$g(\overline{x}) \sum_{j=1}^{A_m} \lambda_{m,j} S'_{m,j}(x_m) = -\partial_m g(\overline{x}) \sum_{i=1}^{N} \sum_{j=1}^{A_i} \lambda_{i,j} S_{i,j}(x_i) \tag{5.27}$$

We have proven that if the gain function is positive and concave, the optimization is guaranteed to converge to the unique solution. We model the congestion based interference based on the work in [110]. The authors in [110] provide a full interference and well-fitting heuristic to find the relation between effective throughput and user demand, thus congestion, in a heterogeneous cellular network.

To obtain average user demand, we utilize x_i^{-1}, the average number of packets sent by node i per second, where x_i is the waiting time for the transmission of a packet. The average demand per user is then $\frac{1}{N}\sum_{i=1}^{N} x_i^{-1}$, which can be used in the congestion model to obtain the drop in effective throughput. We use an exponential fit to the results in [110] such that the gain in throughput is defined as:

$$g(\overline{x}) = \exp\left(-\frac{\alpha}{N}\sum_{i=1}^{N} x_i^{-1}\right) \tag{5.28}$$

We define α as an arbitrary fitting coefficient. The gain function is positive due to the exponential. The Jacobian of the gain function is found as:

$$\frac{\partial g}{\partial x_m} = \exp\left(-\frac{\alpha}{N}\sum_{i=1}^{N} x_i^{-1}\right)\frac{\alpha}{N x_m^2} \tag{5.29}$$

In order for a solution to exist, the sign of $\sum_{j=1}^{A_m} \lambda_{m,j} S'_{m,j}(x_m)$ must be the opposite of $\partial_m g$, which is positive for any solution, so the former must be negative. The only gain function with a negative first order derivative is the Information Freshness. It is possible to mix multiple applications so that the resulting summation is still negative. Otherwise, the solution converges to the minimum x values possible as all gain functions would be ever increasing. Note that this condition is only necessary if a variable packet gain function is used. If the packet gain function is set to a positive constant value, concavity is the only requirement for the stream gain functions as noted earlier. For maximum clarity, we will be using the information freshness gain function for the 5G RAN case. This results in:

$$\sum_{j=1}^{A_m} \lambda_{m,j}\left(E(D_{m,j}) - x_m\right) < 0 \tag{5.30}$$

$$x_m > \frac{\sum_{j=1}^{A_m} \lambda_{m,j} E(D_{m,j})}{\sum_{j=1}^{A_m} \lambda_{m,j}} = \overline{D_m} \tag{5.31}$$

This inequality sets a minimum value for the transmission instance as a weighted average of the expected deadlines of the applications on a node. Continuing with

the analysis of the packet gain function, we take the second derivative to obtain:

$$\frac{\partial^2 g}{\partial x_m^2} = \exp\left(-\frac{\alpha}{N}\sum_{i=1}^{N} x_i^{-1}\right)\left(\frac{\alpha^2}{N^2 x_m^4} - \frac{2\alpha}{N x_m^3}\right) \tag{5.32}$$

$$\frac{\partial^2 g}{\partial x_m x_n} = \exp\left(-\frac{\alpha}{N}\sum_{i=1}^{N} x_i^{-1}\right)\frac{\alpha^2}{N^2 x_n^2 x_m^2} \tag{5.33}$$

This is a negative-definite matrix. To prove this, we look at the summation of the absolute values of the off-diagonal components for any row:

$$\sum_{n\neq m}\left|\frac{\partial^2 g}{\partial x_m x_n}\right| = \exp\left(-\frac{\alpha}{N}\sum_{i=1}^{N} x_i^{-1}\right)\sum_{n\neq m}\frac{\alpha^2 x_n^{-2} x_m^{-2}}{N^2} \tag{5.34}$$

If this summation is less than the absolute value of the diagonal element of the same row for every row, this matrix is a diagonally dominant matrix:

$$\left|\frac{\partial^2 g}{\partial x_m^2}\right| = \frac{2\alpha N x_m - \alpha^2}{N^2 x_m^4}\exp\left(-\frac{\alpha}{N}\sum_{i=1}^{N} x_i^{-1}\right) \tag{5.35}$$

$$\frac{2\alpha N x_m - \alpha^2}{N^2 x_m^4} \geq \frac{\alpha^2}{N^2 x_m^2}\sum_{n\neq m} x_n^{-2} \tag{5.36}$$

$$x_m \geq \frac{\alpha}{2N}\sum_{n}\left(\frac{x_m}{x_n}\right)^2 \tag{5.37}$$

The maximum value for the right hand side is obtained when all x_n have their minimum possible value, which was found to be the weighted average of the expected deadlines of all applications on the node. We denote this value as $\overline{D_n}$ for simplicity. The upper-bound for the worst case and the lower-bound for existence are given as:

$$\overline{D_m} \leq x_m < 2\frac{1}{\frac{\alpha}{N}\sum\limits_{n\neq m}\left(\overline{D_n}\right)^{-2}} \tag{5.38}$$

The transmission instance must be less than twice the harmonic mean of the square of $\overline{D}$s for a solution to exist in the worst-case. Since all diagonal values are negative and we have a diagonally strictly dominant matrix, the matrix is also negative definite, meaning that the solution is unique and globally maximum.

Now that we have obtained the necessary conditions for a solution to exist and be unique, we continue with our communication mechanism to implement a

fast, distributed and converging method. We start with the solution expression in Equation (5.27):

$$-\frac{g(\overline{x})}{\partial_m g(\overline{x})}\sum_{j=1}^{A_m}\lambda_{m,j}S'_{m,j}(x_m)=\sum_{i=1}^{N}\sum_{j=1}^{A_i}\lambda_{i,j}S_{i,j}(x_i)=C \tag{5.39}$$

The right hand side of this equality is independent from x_m and is a constant global value designated by C. Using our congestion gain function, this expression can be further simplified as:

$$-x_m^2\sum_{j=1}^{A_m}\lambda_{m,j}S'_{m,j}(x_m)=\frac{\alpha}{N}\sum_{i=1}^{N}\sum_{j=1}^{A_i}\lambda_{i,j}S_{i,j}(x_i)=C \tag{5.40}$$

We can utilize the Bisection method to solve this global optimization in very few steps. The base station starts the iteration by sending three C values: 0, C_{max} and $C_{max}/2$. When the nodes receive the C values, they start their own Bisection method iterative solutions. Once three x_m solutions are obtained, the nodes calculate their $\sum_{j=1}^{A_i}\lambda_{i,j}S_{i,j}(x_i)$ values and send them to the base station. Once the base station collects all results, it accumulates them to obtain the resulting C values. The error of each trial is the difference between the calculated C values and the initially sent values. The boundary that has the same error sign of $C_{max}/2$ is replaced by $C_{max}/2$ and the resulting interval is again divided into half for the next iteration. This is repeated until a desired error is achieved. The number of steps for convergence is constant at $log_2(C_{max}/C_{\text{Error Resolution}})$. The iteration is guaranteed to converge as long as the predefined boundaries are adhered to by the user. The number of iterations can be drastically reduced by sending multiple C values at once.

The resulting optimization finds the optimal transmission times for each node in distributed fashion, such that the congestion and the information freshness of every application is balanced globally. The iteration only needs to run during the initialization phase. The optimization method would also work for other gain function selections, as long as the required conditions are satisfied.

Iterative Addition of Nodes

In the previous section, we provided the solution for the case of many nodes connecting to a base station cell at once. However this is a less likely scenario, as in reality the nodes are connecting or disconnecting one-by-one at different times. Thus, there is a need for a solution that handles the dynamic connection of individual nodes, instead of using the complete iteration solution with every node provided in the previous section. We achieve this using a new one-to-one negotiation scheme between the base station and the incoming node to obtain the same optimal result, while reducing the communication overhead drastically. To achieve this, we propose that the base station sends a vector of possible C values as a vector to the incoming node and uses the response to obtain the accumulation result with the least error in Equation (5.40) in a single iteration. However, the base station needs to store the responses to each C value from all connected nodes in a table.

In the case of an incoming node, the base station sends the same C vector to the incoming node and gets the response vector for the provided C vector. The left hand side of Equation (5.40) requires no network knowledge other than the C value and can calculate its response using only its local application information. Finally, the base station calculates the new C value by updating its previous accumulation using:

$$\overline{\Delta C} = \frac{1}{N}\left[\sum_{i=1}^{N-1} \text{Response}_i(\overline{C}) + \text{Response}_{new}(\overline{C})\right] - \overline{C}$$

$\overline{C}$ is used to describe the vector of C values. The C value with the minimum ΔC becomes the new optimal value. In the case of a disconnecting node, the base station does an internal calculation using the opposite of the incoming node case:

$$\overline{\Delta C} = \frac{1}{N-1}\left[\sum_{i=1}^{N} \text{Response}_i(\overline{C}) - \text{Response}_{\text{old}}(\overline{C})\right] - \overline{C}$$

Again, the C value with the minimum error becomes the new optimal value. The resulting optimal value can either be broadcast to all nodes at each change or if the change exceeds a threshold to reduce the overhead. Note that the computation is distributed among the nodes and the base station. Each node, including the

base station, calculates only a single iteration of the Bisection method, resulting in a constant computational complexity. Since the solution is unique on the local and global equalities, the iteration is guaranteed to converge by the Mean Value Theorem.

5.3.3 Optimal Policy for Multi-hop WSNs

In contrast to the 5G RAN case, we concentrate on the effects of forwarding based delays in the multi-hop WSN case. The packet gain function reverts back to its original form, where it depends only on the node's transmission instance, rather than the global information. However, we modify the stream gain function to incorporate the multi-hop related delays. Furthermore, any forwarding node must now consider the data streams of the previous node that are to be forwarded to the next node. The total gain for a node can be written as:

$$G_i(x_i) = g_i(x_i) \sum_{j=1}^{A_i} \lambda_{i,j} S_{i,j}(x_i, H_i(\overline{x})) \tag{5.41}$$

H_i represents the time it takes for a packet by node i to be forwarded to the sink node. This value is added to the delay, if the gain function uses any delay related metric. Furthermore, we modify the transmission instances, x_i, by $x + \sum_{j=2}^{i} \Delta x_j$. Δx_j are optimization variables that define the offset of transmission after receiving a packet from the previous node and x is the transmission instance of the farthest away node to the sink. If the optimal Δx_j values are constant in time, this means that the optimal solution is when the whole network transmits with the same transmission instance, separated by constant offsets of Δx_j. Otherwise, individual transmission instances can still be calculated by accumulating the offsets. The nodes are numbered such that node 1 is the farthest and N is the closest to the sink, without the loss of generality. The total gain in the network can then be written as:

$$G(\overline{x}) = \sum_{i=1}^{N} g_i(x) \sum_{j=1}^{A_i} \lambda_{i,j} S_{i,j}(x, \sum_{h=\text{path}_i} \Delta x_h) \tag{5.42}$$

Path for any node is determined by a routing algorithm. We also add constraints on the offsets, such that they are lower-bounded by a fixed value of

propagation and processing delays. To do this, we use the Lagrangian dual and add K.K.T. multipliers to describe the inequality constraints. We define μ_m as the K.K.T. multiplier for the inequality of $\Delta x_m \geq P_m$, where P_m is the propagation and processing delay of node m. We take the partial derivatives to obtain the optimal solution:

$$\begin{aligned}\frac{\partial G}{\partial x} = &\sum_{i=1}^{N} g_i'(x) \sum_{j=1}^{A_i} \lambda_{i,j} S_{i,j}(x, \sum_{h=\text{path}_i} \Delta x_h) \\ &+ \sum_{i=1}^{N} g_i(x) \sum_{j=1}^{A_i} \lambda_{i,j} \partial_1 S_{i,j}(x, \sum_{h=\text{path}_i} \Delta x_h) = 0 \end{aligned} \tag{5.43}$$

$$\begin{aligned}\frac{\partial G}{\partial \Delta x_m} = &\sum_{i=\text{path}_m} g_i(x) \sum_{j=1}^{A_i} \lambda_{i,j} \partial_2 S_{i,j}(x, \sum_{h=\text{path}_m} \Delta x_h) \\ &+ \mu_m = 0 \end{aligned} \tag{5.44}$$

Next, we assume that the delaying of transmission (Δx_h), beyond the optimal transmission instance (x) has a negative effect on the stream gain function, such that $\partial_2 S_{i,j} < 0$. Since the packet gain function must be positive valued, μ_m must be positive in order to obtain zero in Equation (5.44). By complementary slackness, $\mu_m > 0$ can only happen if $\Delta x_m = P_m$ equality of boundary is satisfied. This means that the optimal values of the transmission offsets are their lower-bounds. Since the lower-bounds are fixed values, the network should have a single transmission period, x, where each node transmits by adding its own data streams into the forwarded packet after the minimum offset possible, P_m. The common transmission period is obtained by:

$$\begin{aligned}&\sum_{i=1}^{N} g_i'(x) \sum_{j=1}^{A_i} \lambda_{i,j} S_{i,j}(x, \sum_{h=\text{path}_i} P_h) \\ &+ \sum_{i=1}^{N} g_i(x) \sum_{j=1}^{A_i} \lambda_{i,j} S_{i,j}'(x, \sum_{h=\text{path}_i} P_h) = 0 \end{aligned} \tag{5.45}$$

To solve for the optimal transmission time, we need global information from every node on the path to the sink. P_h can only be calculated as we go towards the sink node. The solution starts by the sink node broadcasting a time stamp and a vector of x values. Any receiving node sets its P_h value to the difference of

current time and the received time stamp. Then, it calculates its own gain value using the received vector by:

$$g_i'(\overline{x}) \sum_{j=1}^{A_i} \lambda_{i,j} S_{i,j}(\overline{x}, \sum_{h=\text{path}_i} P_h) \tag{5.46}$$

$$+ g_i(\overline{x}) \sum_{j=1}^{A_i} \lambda_{i,j} S'_{i,j}(\overline{x}, \sum_{h=\text{path}_i} P_h) \tag{5.47}$$

The node rebroadcasts the packet to its children that are farther away from the sink. This process is repeated until the leaf nodes are reached. Note that low mobility is assumed in the network, as expected in most WSNs, such that the received information is assumed to not have changed during the relay process. The leaf nodes transmit their calculated gain vector to their parent nodes. Once the parent nodes have received from all children, they accumulate the gain vectors with their own gain vector and forward it to their respective parents. Only the accumulated vector is transmitted, not individual vectors of the children. This iteration is repeated until the sink node receives the accumulated vector of the network. The gain value closest to zero, along with its corresponding aggregation time value, x, is selected as the common transmission period of the network. This process could also have been implemented iteratively as in Bisection method, however the cost of broadcasting in the multi-hop network is much more as compared to the 5G RAN case. The solution is guaranteed to converge either way, as long as the gain functions adhere to the mentioned rules. Our solution requires network-wide messaging as an initial overhead. Even though such a messaging can be considered a high overhead, essential network-wide services, such as routing or addressing, also have an overhead at the same scale. It is also possible to implement the same solution using a distance vector based routing algorithm, where the cost metric would be additive and its value should be replaced by P_h. The routing algorithm's messages would contain enough information to determine the optimal solution. This way, mobility, dynamic conditions and reliable global information dissemination can be handled by the routing algorithm.

In the case of changing conditions such as a new data stream, change in the rate of the stream or topography change due to mobility, the resulting change in the

gain function can be reported to the sink as a difference from the previous vector. The sink node then updates its aggregation time, x, selection if the difference with the current period is beyond a user-defined threshold to avoid unnecessary broadcasting and energy waste.

5.4 Sensitivity Analysis

In this section, we study the sensitivity of our algorithm parameters and compare it to the state of the art. This provides a bridge between the mathematical optimization results and the more complex and realistic scenarios in the next section. All our results are obtained via simulations with the network simulator ns3 [111]. IEEE802.11 standard was used as the underlying Media Access Control (MAC) protocol, with IP as the network layer protocol using predetermined IP addresses from a global pool. The channel model uses Friis propagation loss and a constant speed propagation delay models. Since the main focus of this section is the analytically backed simulation results concentrated on individual performance criteria, the propagation models are kept simple. Nodes are distributed in a radial manner around the sink node. Destination sequenced distance vector (DSDV) routing algorithm is used and since there is no mobility the tables stay the same. The update rate of the tables is set at every 6 minutes. For the generation process, we have used the Poisson distribution as it is one of the most widely used distributions for queuing analysis. We normalize the energy consumption of a transmission to 1J for to simplify analysis. The power consumption results can be scaled by the actual power consumption of a real node. For example, for TI CC-2650, the energy consumption of a transmission corresponds to 0.3mJ approximately [112]. We study three distinct cases:

- **Random deadlines:** This case is used to study the effect of randomness levels in the deadlines on the overall performance.
- **Heterogeneous gain functions:** Different types of gain functions are used on the same node concurrently, creating a multi-objective optimization.

- **Increasing number of applications:** The number of applications is varied to observe the effects.

We compared the performance of our algorithm with the state-of-the-art solutions. We implemented the Cascading Timer solution with fixed period and cascading period variations in [41], denoted as CT in the results. Fixed period algorithm uses a periodic transmission instance, where the period is the average sampling rate of the applications. Cascading algorithm also uses the same periodic approach, but modifies the period by one third of the hop distance to the sink, to accommodate for the forwarding delays. To understand the effect of the period selection, we have also run simulations for 2x and 4x the period of the CT algorithm. We also implemented the Attribute-Aware Data Aggregation protocol [43], denoted as ADA in the results. ADA adjusts the timing of transmissions using an adaptive timer that depends on the state of a node's internal queue. The timer is determined using the following equation (Eq. 14 in [43]):

$$T_u(t) = T_F \times \left[1 - \frac{N_u(t)}{S \times \phi}\right] \tag{5.48}$$

Transmission timing, T_u of node u is determined using how full the current state of the queue is. $T_F = 0.3s$ is a time constant, N_u is the number of packets currently in the queue, $S = 32$ is the maximum queue length and $\phi = 0.9$ is the fraction determining the level of fullness triggering an immediate transmission to avoid packet drops. All values are provided in [43].

For all cases, we investigate four metrics: 1) average power consumption per node, 2) total freshness per transmission, 3) average expiration rate of deadlines, 4) average power consumption per non-expired data. For the sake of the analysis, the energy consumption of a transmission is normalized at $1J$, such that all power consumption results can be scaled by the actual consumption of a specific device to obtain the actual values.

5.4.1 Random Deadlines

In the random deadline scenario, we use Poisson distribution for the generation process, widely used for queuing analysis, and uniform distribution for the

deadline values to observe the effect of maximum randomness. We create a single data stream, but vary the deadline boundaries, while keeping the expected deadline at 10 seconds. We start with the theoretical sensitivity analysis considering information freshness and expiration gain functions separately. Let L and U be the lower and upper bounds of the uniform distribution for deadlines. Then, the probability distribution function can be obtained by:

$$q(D) = (U - L)^{-1}, L \leq D \leq U \text{ and } 0, \text{ otherwise} \tag{5.49}$$

Since the expected deadline is $E(D) = 10$, the information freshness gain function becomes $S(x) = 10x - 0.5x^2$. Since there is a single stream, the optimal solution is obtained at $x = 10$, constant for all deadline distributions. We obtain the expiration ratio for this scenario using the expiration gain function, by substituting the uniform distribution and $x = 0.5(U + L) = 10$:

$$S(x) = 1 - \int_{D=0}^{x} q(D)(1 - D/x)dD = \frac{3U + 5L}{4(U + L)} \tag{5.50}$$

This is expected number of successful packets in the interval of an aggregation, x. The expiration ratio is obtained by subtracting from 1 to convert success ratio into expiration:

$$\text{Expiration Ratio} = \frac{U - L}{4(U + L)} \tag{5.51}$$

This result shows that we can expect the expiration ratio to linearly increase with increasing randomness and no expiration, when the deadline is deterministic. The average power consumption is proportional to x^{-1}, thus constant across all distributions. The average power consumption per non-expired data is obtained by:

$$\text{Energy per non-expired data} = \frac{2(U - L)}{3U + 5L} + \frac{2}{U + L} \tag{5.52}$$

The expression achieves its minimum when $U = L$ resulting in a deterministic case and increases linearly with increasing randomness. Finally, the information freshness per transmission is obtained using the information freshness gain function:

$$S(x) = E(D)x - x^2/2 = x^2/2 = 50 \tag{5.53}$$

The information freshness is constant since the expected deadline is constant. We repeat the steps for an expiration gain function based optimization. The optimal solution is any value between 0 and L, we select L as it consumes the least energy. Using the solution in the expiration gain function gives $S(x) = 1$, meaning no expiration as expected. The average power consumption becomes $1/L$ and the power consumption for non-expired data becomes $1/L$. Finally, the information freshness can be calculated using the information freshness gain function:

$$S(x) = E(D)x - x^2/2 = 10L - L^2/2 \tag{5.54}$$

Note that when the deadline is deterministic, the information freshness is the same for both optimal solutions.

We compare our optimal solution using information freshness and expiration gain functions with the CT and ADA algorithms with simulation. The results are given in Figure 5.3. The results show that our freshness optimal configura-

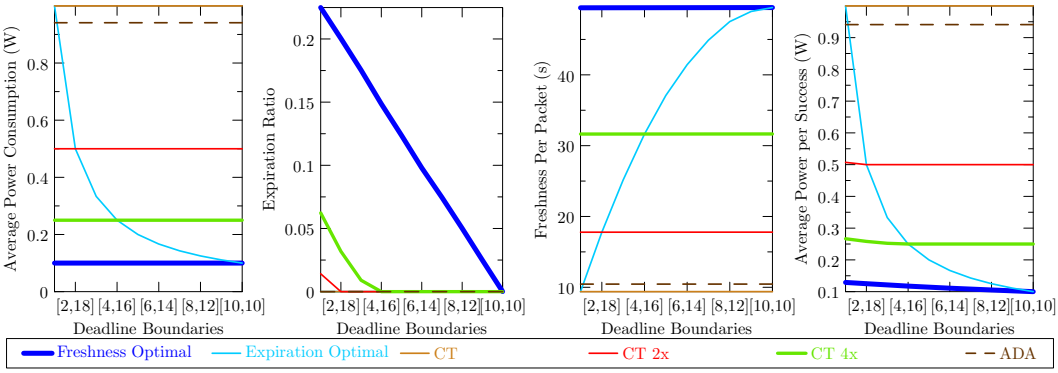

Figure 5.3: Effect of deadline randomness on the performance. Our freshness optimal solution has the highest information freshness, lowest energy consumption and lowest energy per non-expired data, while our expiration optimal solution has no expiration.

.

tion has the highest information freshness - more than 1.55x, the lowest energy consumption - more than 60% reduction and lowest energy consumption per non-expired data. The statistics are calculated with respect to the closest state-of-the-art algorithm (CT 4x). Furthermore, our expiration optimal configuration has no expiration at all as expected. Increasing randomness increases the power consumption to achieve optimal expiration ratio, whereas all the other results have higher

expiration ratios with increased randomness, except ADA. This is due to the fact that it becomes harder to predict the deadlines of the generated data streams. ADA also achieves no expiration due to its frequent transmission, however in cost of energy consumption. Compared to our expiration optimal solution, the very frequent transmission of ADA becomes a waste of energy as both solutions achieve no expiration. The reason is due to the fact that our algorithm directly considers the heterogeneous deadlines of each individual application and ADA acts on the queue size of the whole node.

5.4.2 Heterogeneous Gain Functions

In our second scenario, we consider two application streams, where one stream has an information freshness gain function and the other one has an expiration gain function. We fix the generation rate of one of the applications to once per second and vary the second one to observe the change in the weighted decision making of our algorithm. The first stream has a constant deadline of 1s and the second stream's deadline is fixed at 10s.

The optimization problem is a multi-objective optimization as a combination of freshness and expiration. The optimal solution is obtained using:

$$\lambda_1 S_1'(x) + \lambda_2 S_2'(x) = 0 = (1 - x) + \lambda_2(u(10 - x)) \tag{5.55}$$

$$x = 1 + \lambda_2, \lambda_2 \leq 9 \text{ and } x = 10, \lambda_2 > 9 \tag{5.56}$$

The solution shows a balance depending on the rate of the second stream. Until $x = 10$ seconds, the second stream has no expiration, such that the aggregation time is shifted from 1 second (freshness optimal) to 10 seconds (expiration optimal) as the rate is increased. After 10 seconds, any increase in aggregation time results in a decrease in both gain functions, so the solution stays constant for large arrival rates, converging into the expiration rate optimal result. Using the expiration gain function, the expiration ratio of a deterministic deadline is:

$$S(x) = 1 - u(x - D)(1 - D/x) \tag{5.57}$$

For the first stream, since the aggregation time is always greater than 1 second, it converges to $1 - x^{-1}$. For the second stream, the aggregation time is always less

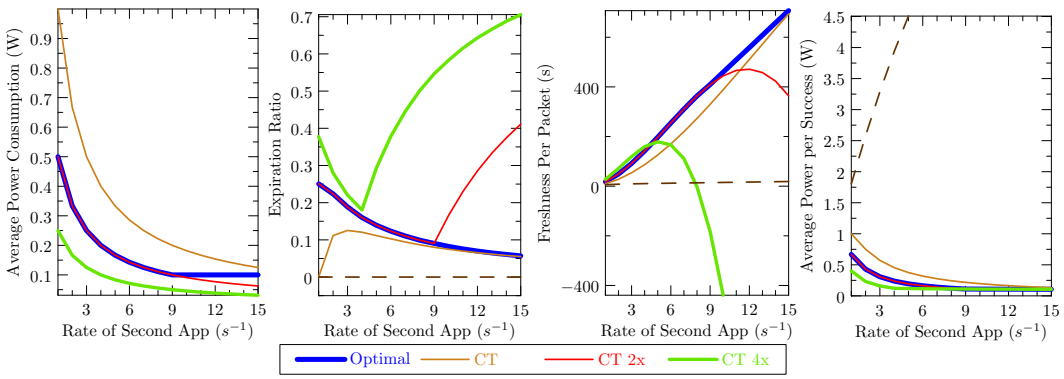

Figure 5.4: Combination of heterogeneous application streams. As the generation rate of the second application stream increases, the weighted average of the gain functions favors towards the second stream. The power consumption is not shown for ADA as it grows very fast relative to the other solutions. However, it is observable in the average power per success figure, since its success rate is 1.

.

than 10 seconds, so the expiration rate is zero. The combined expiration rate is a weighted combination: $\lambda_2(1+\lambda_2)^{-2}$ for $\lambda \leq 9$ and $0.9(1+\lambda_2)^{-1}$ otherwise. The information freshness per transmission can be calculated using the gain function as: $0.5(1+18\lambda_2-\lambda_2^2)(1+\lambda_2)$ for $\lambda_2 \leq 9$ and $50\lambda_2-40$ otherwise. The simulation results confirming the theoretical results are shown in Figure 5.4. The results show that our optimal solution is a weighted combination of information freshness and expiration. As the generation rate of the second stream increases, the balance shifts towards the expiration function as expected. At the rate of $9s^{-1}$, the solution is at a complete balance at $x = 10s$ - transmission every 10 seconds. If the rate is further increased, the resulting aggregation time cannot increase further as that would result in the expiration gain function to fail since the 10 second deadline would be violated. The solution automatically ignores the loss of the information freshness gain and switches to the expiration gain at larger rates, as the gain of the latter exceeds the loss of the former. The sharp turns for the expiration ratios of the CT algorithm is due to the average sampling rate, which is the transmission period of the CT algorithm, exceeds the 10 second deadline of the second application resulting in a sudden increase in expiration ratios. ADA has a very high transmission rate resulting in a costly result for average power

consumption, however it also results in no expiration. In all performance criteria, our algorithm shows a very balanced result as the optimization considers the two metrics of expiration and information freshness.

5.4.3 Increasing Number of Applications

We next study the effects of the number of application streams. Each stream has a uniform deadline distribution in the interval $[4, 10]$s. All streams have a Poisson distributed generation process with a rate of $1s^{-1}$. We increase the number of applications from 2 to 20.

Since we have a homogeneous node, the optimal solution can be calculated by: $x = E(D) = 7s$, independent of the number of applications. This results in an average power consumption at 7^{-1}W. Using the expiration gain function, the expiration rate is constant at 3/28 and the power consumption per non-expired data becomes 0.16W. Finally, the information freshness is the sum of all freshness gain functions, resulting in $24.5N$, where N is the number of applications. The information freshness is expected to increase linearly since more data can be transmitted inside the same transmission.

The simulation results are given in Figure 5.5. The results show that the information freshness increases linearly with the number of application streams as expected. More data is generated and can be aggregated into a packet in a single transmission, increasing the information freshness. Since all applications have the same deadline distribution and expected deadline values, the optimal transmission instance stays constant. This results in constant power consumption and expiration rate metrics. Our information freshness optimal solution has the highest information freshness - by at least 23% higher, 12% expiration rate, the lowest power consumption - saving more than 44% and the lowest power consumption per non-expired data, compared to the closest algorithm (CT 4x).

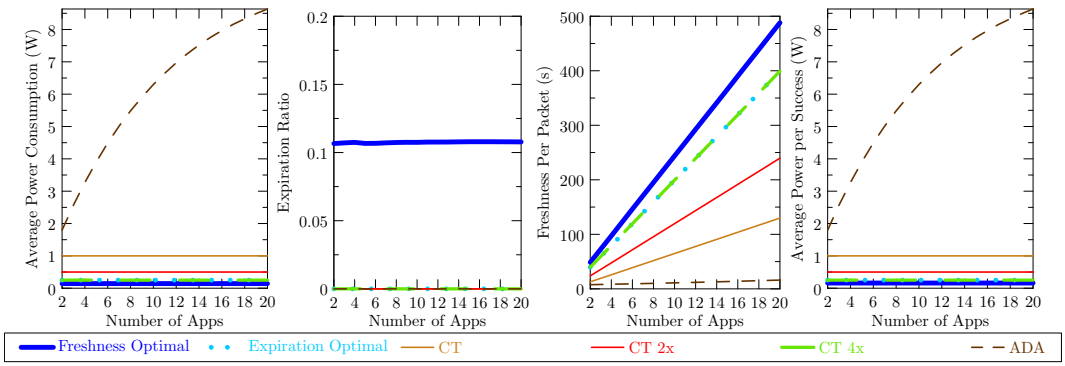

Figure 5.5: Effect of number of application streams on performance. Information freshness increases linearly with the number of streams since more data can be aggregated onto packets, while the expiration rate and power consumption stay constant as the period stays constant. Expiration Optimal and CT 4x have the same period, thus their results are overlapping. In the expiration ratio figure, only Freshness Optimal result has a non-zero expiration rate.

.

5.4.4 Effect of Congestion

For the fourth scenario, we do a comparison between our congestion aware solution and the optimal solution without any packet gain function to represent the congestion. Each node has a single application with a constant deadline of 5 seconds. We varied the number of nodes from a single node to 2000 nodes, a congested sector of a base station. The solution without the gain function has no incentive to change its transmission instance as the network gets crowded, as the solution of each node is independent from each other as in Section 5.3. However, in reality, as the number of nodes increase the access times also increase it becomes impossible to guarantee the calculated transmission instance by the lower layers. This is included in our 5G RAN solution with the packet gain function representing the congestion. The results are given in Figure 5.6.

The result shows that the congestion aware optimal solution slightly increases the transmission instance with increasing number of users. The effect of the increase is observable in the congestion gain function. If the congestion is cutoff at the 0.1 level, the congestion aware optimal solution achieves the same performance with 1000 more active users. The aggressiveness of the algorithm to

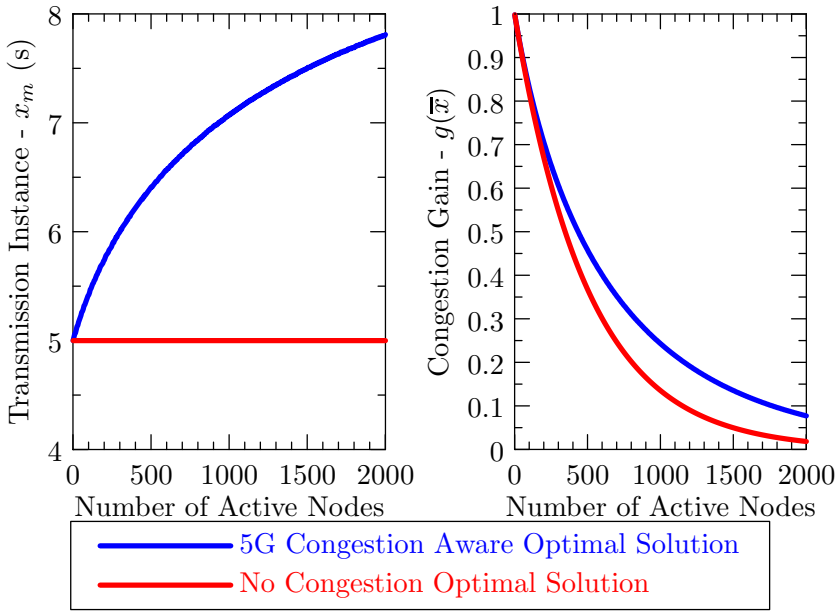

Figure 5.6: Effect of congestion on the optimal transmission instance selection. The congestion aware 5G solution enables higher gains compared to the unaware solution.

.

the congestion can be adjusted using the α parameter in the gain function.

5.4.5 Homogeneous Gain Functions with Congestion

This scenario consists of a single hop network, where each node has two applications. The first application has a constant deadline of 1s and the second application's constant deadline is 9s. The congestion aware mode of our solution is running on all nodes. Nodes are added iteratively, where they communicate with the base station to determine their optimal transmission instance based on the congestion levels, where the details were explained in Section 5.3.2. Both streams have a Poisson distributed generation with a mean rate of $1s^{-1}$ and information freshness as the gain function. We ran multiple cases, where we have increased the number of users from 10 to 210 in order to understand the connection to congestion levels. The results are given in Figure 5.7. The results show that our congestion-aware 5G RAN solution increases the transmission period to mitigate the congestion, resulting in a decrease in power consumption with more crowded

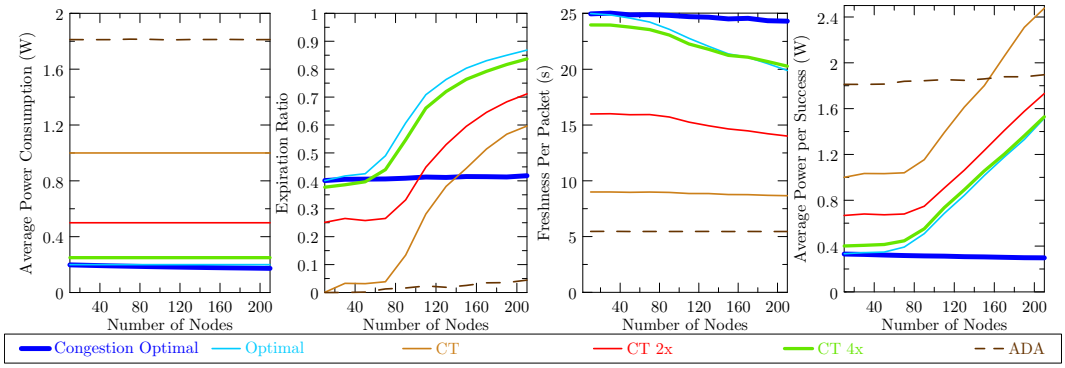

Figure 5.7: 5G network simulation results are provided to study the effect of congestion on the optimal decision. Stream 1: $1s^{-1}$ rate, information freshness gain function, 1s deadline. Stream 2: $1s^{-1}$ rate, information freshness gain function, 9s deadline.

.

networks. The congestion-unaware regular optimal solution and the CT solutions' power consumptions are independent of congestion. With increased congestion, packet losses increase and throughput decreases. If the algorithm tries to send with the same period, more of its packets are lost with increasing congestion. However, our congestion-aware solution's slight adjustment results in constant expiration ratio independent of the number of nodes, enabling a scalable solution. For the information freshness metric, the results show that our congestion-aware optimal solution has the highest performance, 1.25x better than the state-of-the-art solutions under congestion. Furthermore, the degradation with higher congestion is negligible compared to the other solutions. Finally, our 5G RAN solution has the best average power consumption per non-expired data performance, further decreasing with higher congestion, while the other solutions degrade drastically.

5.4.6 Heterogeneous Gain Functions with Congestion

We modify the previous scenario in Section 5.4.5 by changing the second stream's gain function to an expiration gain function. In Section 5.4.2, we performed a theoretical analysis of the performance metrics under no-congestion conditions. For $\lambda_2 = 1$, the expected average power consumption is 0.5W, expiration ratio is 0.25 and information freshness is 18s. The aggregation time is then ad-

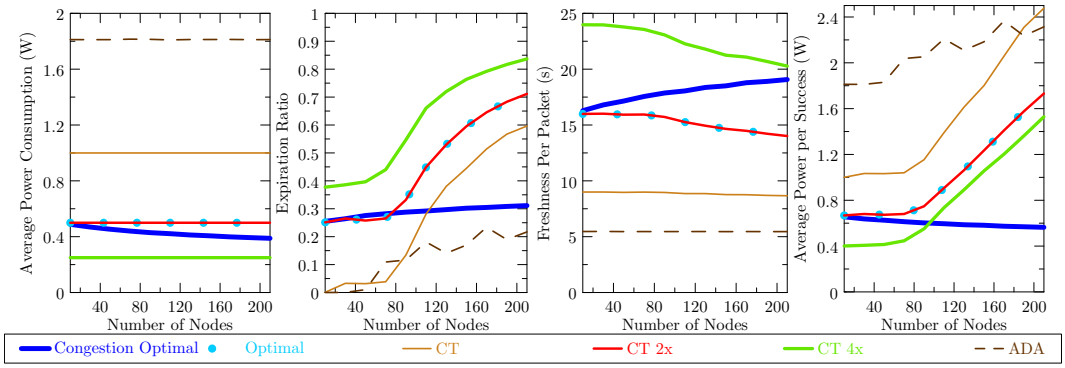

Figure 5.8: 5G RAN heterogeneous gain function simulation results. Stream 1: $1s^{-1}$ rate, information freshness gain function, 1s deadline. Stream 2: $1s^{-1}$ rate, expiration gain function, 9s deadline. Optimal and CT 2x cases have the same period, thus the results are overlapping.

.

justed by increasing congestion. The results for the second scenario is given in Figure 5.8. This scenario differs from the previous scenario in the sense that the optimization goal is a fair combination of information freshness and expiration rate. The average power consumption result shows that our 5G RAN solution adjusts its transmission rate in response to higher congestion. This in effect results in the expiration ratio to stay low with increasing congestion; lower by more than 50% compared to the state-of-the-art solutions. The information freshness of our congestion-aware solution increases with increased number of nodes as the transmission period is adjusted. Finally, for average power consumption per non-expired data, our algorithm stays constant with increased congestion, while the other algorithms' performance is degrading. At the highest level of congestion, our algorithm consumes less than 40% average power as compared to the other algorithms.

5.5 Results

In the previous section, we presented sensitivity analysis, both analytically and simulation-wise to understand the behavior of our optimal solution for different scenarios. In this section, we increase the complexity to obtain the performance

relative to the state of the art solutions for more realistic scenarios. We focus on the two more complex versions of our solutions:

- **5G Radio Access Network:** We create a 5G RAN with different congestion levels, controlled through the number of users. This is used to simulate a sector of a base station placed in the center of a city.
- **Multi-hop Network:** We create a multi-hop network to study the performance of our algorithm in the case of added propagation and forwarding delays.

5.5.1 5G Radio Access Network Case

We simulate a smart city downtown, where multiple sensor nodes are deployed in a small area with high congestion levels, including mobile sensors such as smart phones. Our purpose is to understand the scalability of our solution and the state of the art solutions in a high congestion environment. We picked a T-Mobile cellular base station serving the San Diego downtown area, which has a reported height of 39.2 m. We measured the building and street block dimensions from Google Earth. We used this information to create a 10x10 block grid in ns3. Each node is randomly distributed at different locations and heights. A 2-D Brownian random motion mobility model is used to simulate movement. Even though there is movement present in the simulation, the transmission power guarantees reception at each point of the map. However, interference is dynamic due to mobility and results in different packet losses at different locations. The hybrid building propagation loss model consisting of Okumura-Hata [113], ITU-R 1411 and ITU-R 1238 [114] loss models. The model includes the loss through building walls for both line-of-sight (LoS) and no LoS cases. The topology map with propagation losses is shown in Figure 5.9. We assigned two streams on each node where both data streams have a rate of $1s^{-1}$, an information freshness gain function and the first stream with a 1s deadline and the second stream with a 9s deadline. We simulated different numbers of nodes ranging from 10 to 750 in order to observe the effect of congestion. The results are shown in Figure 5.10. The results agree with the previ-

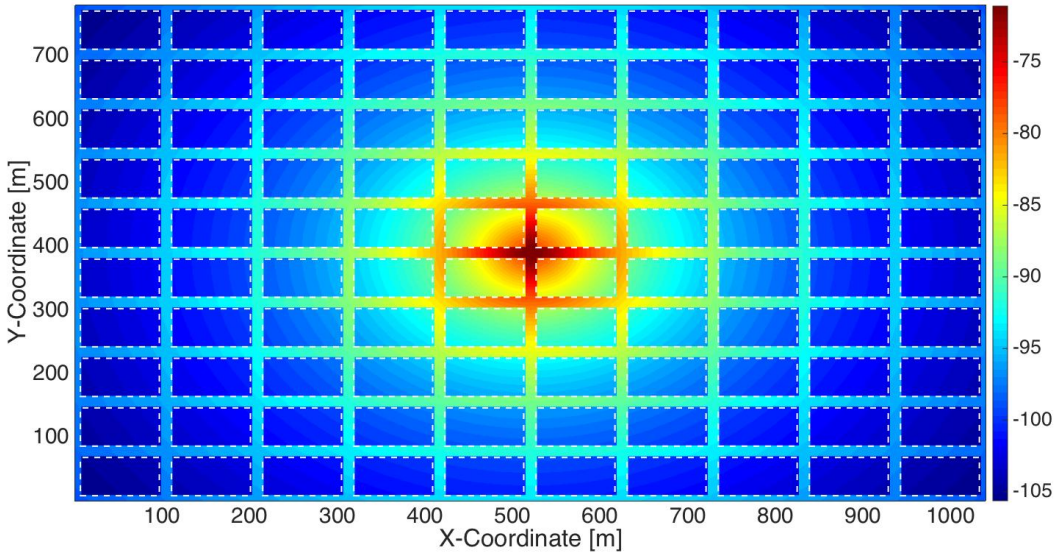

Figure 5.9: 10x10 grid topology of the San Diego Downtown, with the base station in the middle. The color map represents different levels of propagation losses in dB.

.

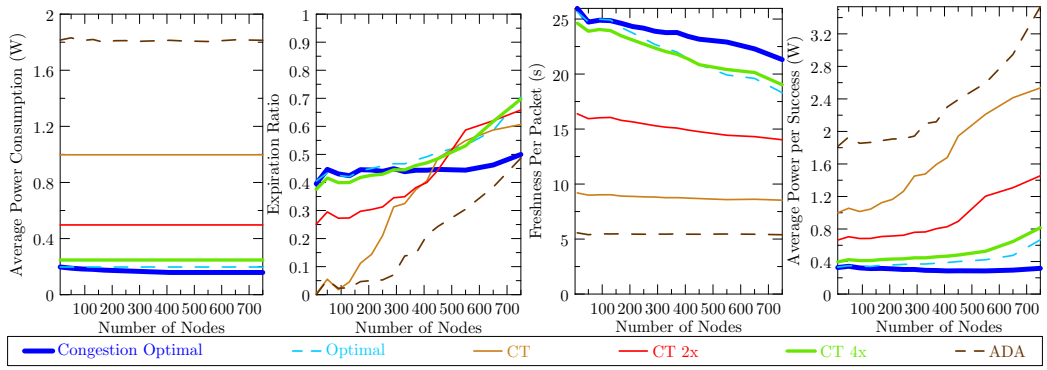

Figure 5.10: San Diego Downtown case with two streams on each node. Number of nodes is increased to observe the effect of congestion.

.

ous static case shown in Figure 5.8, where our congestion-aware 5G RAN solution increases the transmission period to mitigate the congestion, resulting in a decrease in power consumption with more crowded networks. For the expiration ratio metric, it can be observed that the performance of our solution stays roughly constant across increasingly crowded cases, whereas the other solutions degrade by more than 75% compared to their uncongested cases. Our congestion aware solution has the maximum information freshness for all cases, more than 23% compared to the closest solution (CT 4x), and degrades slower compared to the other solutions. Finally, for the average power consumption per non-expired data metric, our 5G RAN solution has the best performance with the lowest value. Furthermore, the performance stays constant with increased congestion levels, whereas the state-of-the-art solutions have degraded by more than 100% compared to their uncongested cases.

In real life, the generation rates are not constant. There are sensor types, such as occupancy, localization, intrusion detection or anomaly detection, where the sensor constantly updates at a low rate. However, when an event is detected, the rate of transmission is increased to report high resolution information. Our optimal solution adapts to these dynamic conditions. We extended the previous downtown case, such that the first stream with a 1s deadline has a variable rate: for a uniformly distributed random time between 60s and 90s, the stream generates data with a low rate of one packet per second. Once this period ends, an event is assumed to occur for a uniformly distributed random interval between 15s and 20s, where the generation rate is increased to three different cases of: 5x, 10x and 20x. The second stream still has a constant rate with one packet per second and 9s deadline. Both streams use an information freshness gain function. The results are shown in Figure 5.11.

The results for the case, where the high data rate interval is 5x the low rate, are similar to the previous constant rate case for our solution. However, all other solutions, except ADA, have lower information freshness. Our congestion-aware optimal solution shows close to constant performance, with a small degradation for information freshness towards very high number of nodes. When the rate is

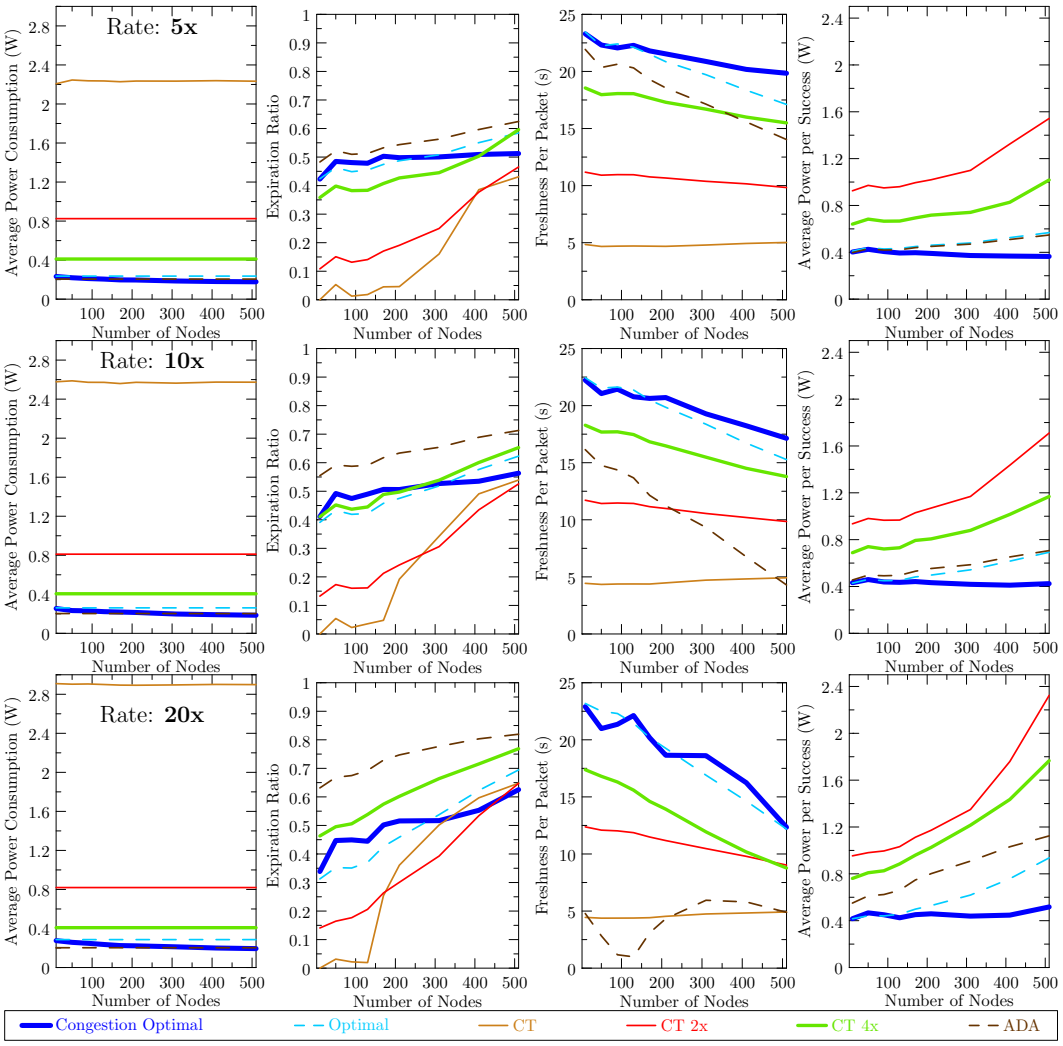

Figure 5.11: San Diego Downtown case with two varying speed streams on each node. Number of nodes is increased to observe the effect of congestion. The top four figures are for the 5x case, middle four figures are 10x and bottom four figures are 20x.

.

increased to 10x and 20x, note that all solutions degrade considerably in performance due to very high levels of congestion. Our solution has the lowest average power consumption and lowest average power consumption per successful data, showing constant performance with increasing congestion. Under highest congestion levels, our solution reduces the average power consumption per successful data by more than 56% compared to the closest solution (ADA). For expiration rate, the constant performance under congestion starts to degrade at higher rates, yet still has the smallest degradation slope among the solutions. Our solution also has the highest information freshness among all solutions. However, note that our congestion-aware optimal solution's information freshness gets closer to the unaware optimal solution as the generation rate increases. The main reason for this is due to the fact that lost packets are not counted in the information freshness calculation (they would have minus infinity freshness). As the number of lost packets increase to significant levels, the performance difference between the two optimal solutions closes. Yet, for the highest congestion case, our solution has at least 1.55x information freshness compared to the closest algorithm (CT 4x).

Overhead: In order to maintain global optimality for the congestion-aware solution, each node updates its information by sending an update message to the base station and the base station broadcasting the new C value if the change is significant enough, as described in Section 5.3.2. This creates a messaging overhead for our congestion-aware solution that is not present in the other solutions. The overhead ratio is calculated by dividing the total number of update messages by the total number of data transmissions. The average overhead across all cases is only 2.5% and the maximum overhead observed is 5%. Furthermore, the computational complexity of our solution requires constant number of iterations for the nodes.

5.5.2 Multi-hop WSN Case

Our multi-hop simulation case consists of a network with multi-hop distances to the sink node to understand the added effects of propagation and forwarding delays. We fix the number of nodes across all cases, but change the topology to alter the maximum number of hops in the network. We consider a

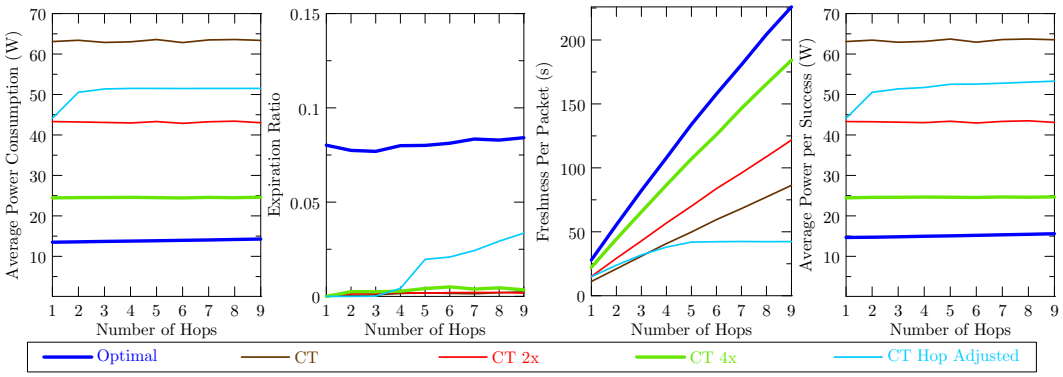

Figure 5.12: Multi-hop simulation results for the linear topology are given to study the additional effects of forwarding delays.

.

linear topology where the nodes are distributed on a line with the sink node at the beginning. This scenario is used to simulate a branch of the routing tree in an ad-hoc wireless sensor network deployment. Each node runs two applications using a Poisson distributed generation process with a mean rate of $10s^{-1}$. The deadlines for the applications are 1s and 9s, respectively and the information freshness gain function is used. The underlying propagation loss model is the same as in Section 5.4. The results for the linear topology are shown in Figure 5.12.

As the number of hops increases, the average power consumption of our solution increases. This is due to the fact that the algorithm is adjusting itself to mitigate the additional forwarding delays by increasing its transmission period. Yet, our solution has the lowest average power consumption. It achieves more than 40% reduction compared to the state-of-the-art solutions. Our solution performs best for the information freshness metric, at least 13% higher than the closest solution (CT 4x). The information freshness increases with the number of hops, since the number of applications generating data is also increasing, such that more data can be aggregated into a packet. Finally, our algorithm has the lowest average power consumption per non-expired data. It achieves 40% reduction on average compared to the closest algorithm (CT Hop Adjusted).

5.6 Conclusion

In previous chapters, we developed different solutions for distributed generation and distributed control in the Smart Grid. All solutions require an efficient underlying communication infrastructure that can guarantee the timely delivery of measurements, forecasts and control iteration messages. Considering that the Smart Grid is expanding to include more sensors and more actuation nodes, scalability of the infrastructure is vital for normal operation. Packet aggregation is proven to be a promising and efficient technique in the literature. Aggregation increases spectral efficiency, reduces delay and reduces congestion. In this chapter, we provide an optimal packet aggregation algorithm that uses the concept of *gain functions* to optimize for a heterogeneous combination of requirements. Gain functions can be any mathematical function and can represent any desired optimization goal, such as delay minimization, energy minimization or freshness maximization. State of the art solutions mainly focus on energy consumption. However, we theoretically prove that energy consumption and data expiration rate have a balanced tradeoff: one cannot be improved without degrading the other in the same amount. Our solution can also optimize energy consumption, but it can also optimize different performance metrics on various levels of resolution. We provide three levels of optimization flexibility: 1) per application data stream, 2) per packet and 3) under different network conditions. Our optimal solution is guaranteed to converge to the optimal result with low computational complexity of $O(A)$, where A is the number of applications on the node. We show optimal distributed solutions for two applications: 1) crowded 5G RANs, considering the congestion based degradation. 2) multi-hop WSNs, designing for the forwarding and processing delays at each hop. In case studies, we show that our algorithm outperforms the compared state-of-the-art algorithms by 60% in energy savings and 1.55x higher in information freshness. Our 5G RAN case study shows that our congestion-aware gain function results in a constant performance with increasing congestion, showing that our algorithm enables scalability with the number of nodes, whereas the state-of-the-art algorithm performances drastically degrade. The overhead to maintain this scalability is only 2.5% of the total transmissions.

This chapter contains material from Alper Sinan Akyurek and Tajana Simunic Rosing, "Optimal in-network packet aggregation policy for maximum information freshness in wireless sensor networks", European Conference on Networks and Communications, 2016. The dissertation author was the primary investigator and author of this paper.

This chapter contains material from Alper Sinan Akyurek and Tajana Simunic Rosing, "Optimal Packet Aggregation in Wireless Networks", which was submitted for consideration to IEEE Transactions on Mobile Computing. The dissertation author was the primary investigator and author of this paper.

Chapter 6

Summary and Future Work

The power grid is changing fundamentally. Higher penetration of distributed generation through renewable resources, increased randomness in generation, distributed sensing and control and increased communication requirements require a fundamental change in the power grid, giving way to the next generation *smart grid*. Due to the vast size of the smart grid, it is of crucial importance to optimize the operation of the grid. However, the challenges in smart grid consist of multiple interconnected aspects such as distributed generation, distributed control and distributed communication. In this thesis, we focus on novel solutions that are optimal, distributed, scalable and low-complexity to provide a framework.

6.1 Thesis Summary

This thesis proposes optimal control strategies for the smart grid, addressing four interconnected problems to create an optimization framework. We addressed the distributed generation problem by developing a fast and accurate prediction algorithm called TESLA in Chapter 2. TESLA is a low complexity forecasting algorithm. It is easily usable for different locations with various climates. We showed through case studies that TESLA has an RMSE reduction performance of up to 50% compared to the state of the art solutions of Numerical Weather Prediction and Persistence forecast. TESLA can calculate the forecast of one year in only a second. Furthermore, TESLA is versatile enough to be used as a machine

learning algorithm, where it was applied to residential user presence and activity detection [115], and also residential energy and flexibility prediction [116]. The accurate prediction timeseries output of TESLA is input into predictive control algorithms, such as ODNBC, for model predictive control operation. This is of crucial importance from the control perspective, since the quality of the input directly affects the quality of the decision, as shown in Chapter 3.

Beyond the quality of the prediction, it is important to determine how to dispatch energy, which is one of the biggest obstacles for the integration of renewable resources. To solve this problem, we developed ODNBC for the optimal nonlinear control of multiple batteries in a distributed fashion. ODNBC, described in Chapter 3, is a novel optimal distributed nonlinear battery control algorithm. It is a low complexity algorithm that uses a high accuracy nonlinear battery model for state of charge and health degradation with less than 2% error. We introduced three negotiation strategies for distributed operation with proven convergence properties. Circular ring negotiation converges independently from the number of batteries, making it a good candidate for very large numbers of batteries. Mean circular ring negotiation converges very quickly for a low number of batteries, complementing the first algorithm. Bisection method has a constant time convergence that can be used in situations where the convergence time must be guaranteed. In case studies, we showed that ODNBC can reduce the cost of a residential building by up to 50% and outperforms the load following heuristic by 30% under the same conditions. Furthermore, when compared against a linear optimal solution that calculates its output using a linear battery model, we have shown that it can incur up to 250% higher costs as compared to our nonlinear optimal solution. If the linear optimal solution is iterated at each time step as an MPC solution, it incurs up to 150% higher costs. This situation arises due to both linearization errors and most importantly premature state transitions. The linearization errors at each time step are accumulated due to the memory of the battery charge, resulting in an empty or full battery before it was planned to be, creating a large cost difference. This property shows how important the nonlinear modeling of the battery is.

Distributed generation and distributed control algorithms mostly assume that there is an underlying stable grid infrastructure, irrespective of the actuation decisions. However, our case studies in Chapter 4 show that it is not sufficient to test the performance of an algorithm locally. We need a smart grid simulator to test distributed control algorithms on a bigger scale. It is important to understand how distributed algorithms perform in the context of the grid. To achieve this, we developed S^2Sim , a novel smart grid simulator. S^2Sim is a low overhead smart grid simulator that enables the co-simulation of distributed control algorithms. It provides the complete cyber physical representation of smart grid through its usage of objects representing actual controllable devices, coordinators representing virtual feedback entities and the electrical circuit representing the physical connections. S^2Sim defines a unified communication interface without constraining the control side implementations and handles the time synchronization among controllers, differentiating itself from currently available simulators. We presented three case studies. In the first case, we demonstrated that open-loop control can lead to unstable conditions, whereas closed-loop control keeps within stable boundaries. In the second case, we simulated control algorithms that perform well under isolated conditions, but we show that their performance degrades when simulated within the larger grid context, proving that the quality of a control algorithm cannot be justified unless tested in the larger scale. The setup is a multi-university collaboration achieved through TCP/IP connections showcasing the parallel computing property of our simulator. The third case shows a comparison between different heuristic control algorithms and how well they perform in terms of voltage stability. All three cases are possible due to S^2Sim and were not possible with previous grid simulators.

Finally, all solutions presented in this thesis require a reliable, congestion-aware and efficient underlying communication infrastructure. Sensor information must be delivered in a timely fashion to obtain good predictions from TESLA. Negotiation information for ODNBC must be sent as quickly as possible to ensure convergence and minimize its effect on the grid stability. A glue between all the components of distributed generation, distributed control and distributed sensing

is communication. To solve this, we developed a novel optimal packet aggregation algorithm in Chapter 5. Packet aggregation is a proven method in the literature, shown to improve the energy consumption, spectral efficiency and scalability. Our algorithm introduces a novel concept of individual gain functions that enable one to describe the priorities of each data stream. The low complexity solution obtains the optimal aggregation time of each node in a distributed fashion, such that the total utility of the network is maximized. We extend the algorithm to be congestion-aware, such that the transmission times are adjusted to mitigate packet losses. We show through case studies that our algorithm has constant performance under increasing congestion, whereas state of the art algorithms degrade by 70% in terms of freshness and more than 160% for energy consumption per non-expired data. In low congestion scenarios, we show that our solution has at least 1.55x more information freshness and more than 60% energy savings compared to the state of the art solutions. We also extend our solution towards multi-hop wireless sensor networks, where the performance is affected by forwarding and processing delays. It provides the optimal aggregation times considering all forwarding delays. Our congestion-aware optimal solution enables efficient and timely transmission of control and sensor data required by distributed control and prediction algorithms.

6.2 Future Directions

Smart grid continues to evolve introducing new challenges, such as connection to the Internet of Things, user-centric applications, automated demand response and data-driven control strategies. In this section, we provide future directions for further improvement on the two topics discussed in this thesis: optimal control and effect of communication on control.

6.2.1 Optimal Nonlinear Energy Storage Control

In this thesis, we showed the importance of model accuracy on control quality for batteries. Our optimal solution is based on a specific model with tunable parameters and requires experimental data for tuning. The quality of tuning de-

termines the accuracy of the model and the quality of the control. Instead of using a predetermined analytic model, we could leverage TESLA to generate a model automatically using measurements or experimental data. The advantage of such a framework would be that the model can be updated and improved upon each measurement and formal scalability and convergence properties can be obtained for a general version of coefficients. The data-driven model concept uses the best properties of the two fields: 1) adaptive solution of data-drive control using new information, 2) fast initialization, provable formal properties and robustness of model-driven control. The accuracy of the model can be determined by the Taylor order and recursive least squares estimation can be utilized for fast adaptive training, without having to retrain for all data points. This eliminates the requirement for storing all measurements.

Our control algorithm is developed for batteries, where the constraints are determined by the physical constraints of the batteries. However, for the EV charging problem, there are also charging deadlines provided by the users that need to be adhered to when solving the optimization problem. Nonlinear state of charge models are missing for the current state of the art EV optimization solutions and must be considered based on the results of this thesis to have an accurate result. Any algorithm can be directly tested in S^2Sim for its effect on the stability of the grid and effects of presence of non-compliant EVs, representing the heterogeneous control cases. The addition of a deadline constraint can be modeled using Lagrangian multipliers in addition to currently used KKT multipliers.

Load shaping is not the only way to reduce costs when using batteries. It is possible to use batteries for demand response. The flexible nature of batteries enable determining stricter bounds on the consumption profile, improving the probability of joining a demand response event. However, the current optimization problem must be converted into a stochastic setup due to the randomness of the arrival of a demand response event. Instead of cost minimization, a risk analysis is necessary to decide the control of batteries. In this risk analysis, the demand response arrival process must be used to minimize the expected total cost, instead of a deterministic value. This introduces an additional dimension to the optimization

problem.

Furthermore, user-centric applications are also another possible direction, where user-context information can be used to improve prediction accuracy and reduce statistical uncertainty. Any additional context information can be learned using TESLA. On the actuation side, user preference and user comfort can be added as additional optimization goals.

6.2.2 Effect of Communication on Control

An important open research problem is the effect of communication inefficiencies on the quality of the control output. Communication mediums and protocols are not perfect. Therefore, delays are unavoidable and the effect of delay must be studied. From a centralized perspective, the solution must be transmitted to the actuator nodes and from a distributed perspective, iterations require constant communication among the nodes. Furthermore, peaks can be missed due to communication delays, endangering the stability of the grid. S^2Sim can be used for this purpose by reducing its time step resolution to observe the effects of minor communication delays.

Another possible direction to tackle this challenge is by increasing the amount and quality of local decisions. This can be achieved by using context information, predicting the global consensus point and reducing the number of iteration steps. TESLA would be a good candidate to combine user context information and past iteration patterns to predict a faster convergence to the consensus solution.

Bibliography

[1] B. O. Akyurek, A. S. Akyurek, J. Kleissl, and T. S. Rosing, "Tesla: Taylor expanded solar analog forecasting," in *Smart Grid Communications (SmartGridComm), IEEE International Conference on*, 2014.

[2] A. S. Akyurek, B. Aksanli, and T. Rosing, "S2sim: Smart grid swarm simulator," in *Green and Sustainable Computing Conference (IGCC), International*, 2015.

[3] A. Akyurek, B. Torre, and T. Rosing, "Eco-DAC energy control over divide and control," in *Smart Grid Communications (SmartGridComm), IEEE International Conference on*, 2013, pp. 666–671.

[4] A. S. Akyurek and T. S. Rosing, "Optimal distributed nonlinear battery control," *IEEE Journal of Emerging and Selected Topics in Power Electronics*, vol. PP, no. 99, pp. 1–1, 2016.

[5] A. S. Akyurek and T. S. Rosing, "Optimal in-network packet aggregation policy for maximum information freshness," in *2016 European Conference on Networks and Communications (EuCNC)*, June 2016, pp. 89–93.

[6] G. W. Arnold, "Challenges and opportunities in smart grid: A position article," *Proceedings of the IEEE*, vol. 99, no. 6, pp. 922–927, 2011.

[7] R. F. Arritt and R. C. Dugan, "Distribution system analysis and the future smart grid," in *Proc. IEEE Rural Electric Power Conf.*, 2011.

[8] P. Mathiesen and J. Kleissl, "Evaluation of numerical weather prediction for intra-day solar forecasting in the continental united states," *Solar Energy*, vol. 85, no. 5, pp. 967–977, 2011.

[9] J. Ruiz-Arias, C. Gueymard, J. Dudhia, and D. Pozo-Vázquez, "Improvement of the weather research and forecasting (wrf) model for solar resource assessments and forecasts under clear skies," *Proc. Amer. Sol. Energy Soc., Denver, USA*, 2012.

[10] R. J. Kuligowski and A. P. Barros, "Localized precipitation forecasts from a numerical weather prediction model using artificial neural networks." *Weather & Forecasting*, vol. 13, no. 4, 1998.

[11] I. Maqsood, M. R. Khan, and A. Abraham, "Intelligent weather monitoring systems using connectionist models," *Neural Parallel and Scientific Computations*, vol. 10, no. 2, pp. 157–178, 2002.

[12] L. Delle Monache, T. Nipen, Y. Liu, G. Roux, and R. Stull, "Kalman filter and analog schemes to postprocess numerical weather predictions." *Monthly Weather Review*, vol. 139, no. 11, 2011.

[13] K. F. Katiraei and J. R. Agüero, "Solar PV integration challenges," *IEEE Power and Energy Magazine*, vol. 9, no. 3, pp. 62–71, 2011.

[14] C. A. Hill, M. C. Such, D. Chen, J. Gonzalez, and W. M. Grady, "Battery energy storage for enabling integration of distributed solar power generation," *IEEE Transactions on Smart Grid*, vol. 3, no. 2, pp. 850–857, 2012.

[15] S. H. Madaeni, R. Sioshansi, and P. Denholm, "How thermal energy storage enhances the economic viability of concentrating solar power," *Proceedings of the IEEE*, vol. 100, no. 2, pp. 335–347, 2012.

[16] C. L. Nge, O. Midtgard, and L. Norum, "PV with battery in smart grid paradigm: Price-based energy management system," in *Proc. 38th IEEE Photovoltaic Specialists Conf. (PVSC)*, 2012, pp. 000 575–000 579.

[17] M. Glinkowski, J. Hou, and G. Rackliffe, "Advances in wind energy technologies in the context of smart grid," *Proceedings of the IEEE*, vol. 99, no. 6, pp. 1083–1097, 2011.

[18] L. Xie, P. M. S. Carvalho, L. A. F. M. Ferreira, J. Liu, B. H. Krogh, N. Popli, and M. D. Ilić, "Wind integration in power systems: Operational challenges and possible solutions," *Proceedings of the IEEE*, vol. 99, no. 1, pp. 214–232, 2011.

[19] K. Worthmann, C. Kellett, P. Braun, L. Grune, and S. Weller, "Distributed and decentralized control of residential energy systems incorporating battery storage," *IEEE Transactions on Smart Grid*, vol. 6, no. 4, pp. 1914–1923, 2015.

[20] J. Tant, F. Geth, D. Six, P. Tant, and J. Driesen, "Multiobjective battery storage to improve PV integration in residential distribution grids," *IEEE Transactions on Sustainable Energy*, vol. 4, no. 1, pp. 182–191, 2013.

[21] M. A. Elgenedy, A. M. Massoud, and S. Ahmed, "Energy in smart grid: Strategies and technologies for efficiency enhancement," in *Smart Grid and Renewable Energy (SGRE), 2015 First Workshop on*, March 2015, pp. 1–6.

[22] I. D. Serna-Surez, G. Ordez-Plata, and G. Carrillo-Caicedo, "Microgrid's energy management systems: A survey," in *2015 12th International Conference on the European Energy Market (EEM)*, May 2015, pp. 1–6.

[23] S. Ci, N. Lin, and D. Wu, "Reconfigurable battery techniques and systems: A survey," *IEEE Access*, vol. 4, pp. 1175–1189, 2016.

[24] A. Seaman, T.-S. Dao, and J. McPhee, "A survey of mathematics-based equivalent-circuit and electrochemical battery models for hybrid and electric vehicle simulation," *Journal of Power Sources*, vol. 256, pp. 410–423, Jun 2014.

[25] Y. Zou, X. Hu, H. Ma, and S. E. Li, "Combined state of charge and state of health estimation over lithium-ion battery cell cycle lifespan for electric vehicles," *Journal of Power Sources*, vol. 273, pp. 793–803, 2015.

[26] B. Aksanli, T. Rosing, and E. Pettis, "Distributed battery control for peak power shaving in datacenters," in *International Green Computing Conference (IGCC)*, 2013, pp. 1–8.

[27] M. Safari, M. Morcrette, A. Teyssot, and C. Delacourt, "Multimodal physics-based aging model for life prediction of li-ion batteries," *Journal of The Electrochemical Society*, vol. 156, no. 3, pp. A145–A153, 2009.

[28] R. Fu, S.-Y. Choe, V. Agubra, and J. Fergus, "Development of a physics-based degradation model for lithium ion polymer batteries considering side reactions," *Journal of Power Sources*, vol. 278, pp. 506 – 521, 2015. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0378775314020874

[29] M. Ouyang, X. Feng, X. Han, L. Lu, Z. Li, and X. He, "A dynamic capacity degradation model and its applications considering varying load for a large format li-ion battery," *Applied Energy*, vol. 165, pp. 48 – 59, 2016.

[30] H. Bindner, T. Cronin, P. Lundsager, J. F. Manwell, U. Abdulwahid, and I. Baring-Gould, "Lifetime modelling of lead acid batteries," Riso, Tech. Rep., 2005.

[31] B. J. S. Drouilhet, "A battery life prediction method for hybrid power applications," NREL, Tech. Rep., 1997.

[32] W. Bernal, M. Behl, T. X. Nghiem, and R. Mangharam, "Mle+: A tool for integrated design and deployment of energy efficient building controls," in *Proceedings of the Fourth ACM Workshop on Embedded Sensing Systems for Energy-Efficiency in Buildings*, ser. BuildSys '12. New York, NY, USA: ACM, 2012, pp. 123–130.

[33] M. Maasoumy, C. Rosenberg, A. Sangiovanni-Vincentelli, and D. Callaway, "Model predictive control approach to online computation of demand-side flexibility of commercial buildings hvac systems for supply following," in *American Control Conference (ACC), 2014*, June 2014, pp. 1082–1089.

[34] V. Raman, A. Donze, M. Maasoumy, R. Murray, A. Sangiovanni-Vincentelli, and S. Seshia, "Model predictive control with signal temporal logic specifications," in *Decision and Control (CDC), 2014 IEEE 53rd Annual Conference on*, Dec 2014, pp. 81–87.

[35] Open distribution system simulator. [Online]. Available: sourceforge.net/projects/electricdss/

[36] http://www.gridlabd.org/. [Online]. Available: http://www.gridlabd.org/

[37] Real time digital system simulator. [Online]. Available: www.rtds.com

[38] Paladin live. [Online]. Available: poweranalytics.com/paladin-software/paladin-live

[39] J. Karlsson, A. Kassler, and A. Brunstrom, "Impact of packet aggregation on tcp performance in wireless mesh networks," in *World of Wireless, Mobile and Multimedia Networks Workshops. (WoWMoM). IEEE International Symposium on a*, June 2009, pp. 1–7.

[40] E. Fasolo, M. Rossi, J. Widmer, and M. Zorzi, "In-network aggregation techniques for wireless sensor networks: a survey," *IEEE Wireless Communications*, vol. 14, no. 2, pp. 70–87, April 2007.

[41] I. Solis and K. Obraczka, "In-network aggregation trade-offs for data collection in wireless sensor networks," *International Journal on Sensor Networks*, vol. 1, no. 3/4, pp. 200–212, Jan. 2006.

[42] S. S. Sran, L. Kaur, G. Kaur, and S. K. Sidhu, "Energy aware chain based data aggregation scheme for wireless sensor network," in *Proc. Int. Conf. Energy Systems and Applications*, Oct. 2015, pp. 113–117.

[43] F. Ren, J. Zhang, Y. Wu, T. He, C. Chen, and C. Lin, "Attribute-aware data aggregation using potential-based dynamic routing in wireless sensor networks," *IEEE Transactions on Parallel and Distributed Systems*, vol. 24, no. 5, pp. 881–892, May 2013.

[44] A. Akyurek, B. Torre, and T. Rosing, "Eco-dac energy control over divide and control," in *Smart Grid Communications (SmartGridComm), 2013 IEEE International Conference on*, Oct 2013, pp. 666–671.

[45] J. Venkatesh, B. Aksanli, J.-C. Junqua, P. Morin, and T. Rosing, "Homesim: Comprehensive, smart, residential electrical energy simulation and scheduling," in *Green Computing Conference (IGCC), 2013 International*, June 2013, pp. 1–8.

[46] R. H. Inman, H. T. Pedro, and C. F. Coimbra, "Solar forecasting methods for renewable energy integration," *Progress in Energy and Combustion Science*, vol. 39, no. 6, pp. 535–576, 2013.

[47] A. Kaur, H. T. Pedro, and C. F. Coimbra, "Ensemble re-forecasting methods for enhanced power load prediction," *Energy Conversion and Management*, vol. 80, pp. 582–590, 2014.

[48] I. Maqsood, M. R. Khan, and A. Abraham, "An ensemble of neural networks for weather forecasting," *Neural Computing & Applications*, vol. 13, no. 2, pp. 112–122, 2004. [Online]. Available: http://dx.doi.org/10.1007/s00521-004-0413-4

[49] M. Bannayan and G. Hoogenboom, "Weather analogue: A tool for real-time prediction of daily weather data realizations based on a modified k-nearest neighbor approach," *Environmental Modelling & Software*, vol. 23, no. 6, pp. 703–713, 2008.

[50] J. P. Hacker and D. L. Rife, "A practical approach to sequential estimation of systematic error on near-surface mesoscale grids." *Weather & Forecasting*, vol. 22, no. 6, 2007.

[51] R. Abdel-Aal, "Improving electric load forecasts using network committees," *Electric Power Systems Research*, vol. 74, no. 1, pp. 83–94, 2005.

[52] G. K. Rutledge, J. Alpert, and W. Ebisuzaki, "Nomads: A climate and weather model archive at the national oceanic and atmospheric administration," *Bulletin of the American Meteorological Society*, vol. 87, no. 3, pp. 327–341, 2014/05/11 2006.

[53] C. P. Research. Solaranywhere. [Online]. Available: solaranywhere.com

[54] J. Zhu, G. Jordan, and S. Ihara, "The market for spinning reserve and its impacts on energy prices," in *Power Engineering Society Winter Meeting, 2000. IEEE*, vol. 2, 2000, pp. 1202–1207 vol.2.

[55] Y. Agarwal, B. Balaji, S. Dutta, R. Gupta, and T. Weng, "Duty-cycling buildings aggressively: The next frontier in HVAC control," in *Information Processing in Sensor Networks (IPSN), 10th International Conference on*, 2011, pp. 246–257.

[56] J. H. Yoon, R. Baldick, and A. Novoselac, "Dynamic demand response controller based on real-time retail price for residential buildings," *IEEE Transactions on Smart Grid*, vol. 5, no. 1, pp. 121–129, 2014.

[57] L. Xie, Y. Gu, X. Zhu, and M. G. Genton, "Short-term spatio-temporal wind power forecast in robust look-ahead power system dispatch," *IEEE Transactions on Smart Grid*, vol. 5, no. 1, pp. 511–520, Jan 2014.

[58] D. Zhu and G. Hug-Glanzmann, "Real-time control of energy storage devices in future electric power systems," *2011 IEEE Trondheim PowerTech*, Jun 2011.

[59] A. Y. Saber and G. K. Venayagamoorthy, "Optimization of vehicle-to-grid scheduling in constrained parking lots," *2009 IEEE Power & Energy Society General Meeting*, Jul 2009.

[60] C. Lin, Y. Shiao, C.-L. Huang, and P. Sung, "A real and reactive power control approach for battery energy storage system," *IEEE Transactions on Power Systems*, vol. 7, no. 3, pp. 1132–1140, 1992.

[61] T. Ohtaka and S. Iwamoto, "Optimal allocation and pq output ratio of nas batteries for improving voltage stability in distribution systems," *2001 Power Engineering Society Summer Meeting. Conference Proceedings (Cat. No.01CH37262)*, 2001.

[62] S. Ried, P. Jochem, and W. Fichtner, "Profitability of photovoltaic battery systems considering temporal resolution," *2015 12th International Conference on the European Energy Market (EEM)*, May 2015.

[63] B. P. Roberts and C. Sandberg, "The role of energy storage in development of smart grids," *Proceedings of the IEEE*, vol. 99, no. 6, pp. 1139–1144, 2011.

[64] P. J. Werbos, "Putting more brain-like intelligence into the electric power grid: What we need and how to do it," in *Proc. Int. Joint Conf. Neural Networks IJCNN 2009*, 2009, pp. 3356–3359.

[65] I. beom Song, D.-Y. Jung, Y.-H. Ji, S.-C. Choi, S.-W. Lee, and C.-Y. Won, "A residential 10kwh lithium-polymer battery energy storage system," in *Proc. IEEE 8th Int Power Electronics and ECCE Asia (ICPE & ECCE) Conf*, 2011, pp. 2625–2630.

[66] Y. Goto, T. Suzuki, T. Shimoo, T. Hayashi, and S. Wakao, "Operation design of PV system with storage battery by using next-day residential load forecast," in *Proc. 37th IEEE Photovoltaic Specialists Conf. (PVSC)*, 2011, pp. 002 369–002 374.

[67] S.-J. Lee, J.-H. Kim, C.-H. Kim, S.-K. Kim, E.-S. Kim, D.-U. Kim, K. Mehmood, and S. Khan, "Coordinated control algorithm for distributed battery energy storage systems for mitigating voltage and frequency deviations," *IEEE Transactions on Smart Grid*, to be published.

[68] Y. Riffonneau, S. Bacha, F. Barruel, and S. Ploix, "Optimal power flow management for grid connected PV systems with batteries," *IEEE Transactions on Sustainable Energy*, vol. 2, no. 3, pp. 309–320, 2011.

[69] K. Mets, F. De Turck, and C. Develder, "Distributed smart charging of electric vehicles for balancing wind energy," in *Smart Grid Communications, 2012 IEEE Third International Conference on*, 2012, pp. 133–138.

[70] W. Su, H. Eichi, W. Zeng, and M.-Y. Chow, "A survey on the electrification of transportation in a smart grid environment," *IEEE Trans. Ind. Inf.*, vol. 8, no. 1, pp. 1–10, Feb 2012.

[71] M. Khalid and A. Savkin, "Optimization and control of a distributed battery energy storage system for wind power smoothing," in *Control & Automation (MED), 2011 19th Mediterranean Conference on*, 2011, pp. 39–43.

[72] S. Sikkabut, P. Mungporn, N. Poonnoi, M. Phattanasak, P. Thounthong, P. Tricoli, B. Nahid-Mobarakeh, S. Pierfederici, B. Davat, and L. Piegari, "A nonlinear control algorithm of li-ion battery substation for DC distributed system," in *Power Electronics, Electrical Drives, Automation and Motion (SPEEDAM), 2014 International Symposium on*, 2014, pp. 1065–1070.

[73] J.-H. Kim, S.-J. Lee, E.-S. Kim, S.-K. Kim, C.-H. Kim, and L. Prikler, "Modeling of battery for ev using emtp/atpdraw," *Journal of Electrical Engineering and Technology*, vol. 9, no. 1, pp. 98–105, jan 2014.

[74] J. Groot, "State-of-health estimation of li-ion batteries: Cycle life test methods," Ph.D. dissertation, Chalmers University of Technology, 2012.

[75] A. Hoke, A. Brissette, D. Maksimovi, A. Pratt, and K. Smith, "Electric vehicle charge optimization including effects of lithium-ion battery degradation," in *2011 IEEE Vehicle Power and Propulsion Conference*, Sept 2011, pp. 1–8.

[76] A. Hoke, A. Brissette, K. Smith, A. Pratt, and D. Maksimovic, "Accounting for lithium-ion battery degradation in electric vehicle charging optimization," *IEEE Journal of Emerging and Selected Topics in Power Electronics*, vol. 2, no. 3, pp. 691–700, Sept 2014.

[77] H. Farzin, M. Fotuhi-Firuzabad, and M. Moeini-Aghtaie, "A practical scheme to involve degradation cost of lithium-ion batteries in vehicle-to-grid applications," *IEEE Transactions on Sustainable Energy*, vol. 7, no. 4, pp. 1730–1738, Oct 2016.

[78] M. Doyle and J. Newman, "Analysis of transference number measurements based on the potentiostatic polarization of solid polymer electrolytes," *Journal of The Electrochemical Society*, vol. 142, no. 10, pp. 3465–3468, 1995.

[79] P. Ramadass, B. Haran, P. M. Gomadam, R. White, and B. N. Popov, "Development of first principles capacity fade model for li-ion cells," *Journal of The Electrochemical Society*, vol. 151, no. 2, pp. A196–A203, 2004. [Online]. Available: http://jes.ecsdl.org/content/151/2/A196.abstract

[80] L. L. Lam and R. B. Darling, "Determining the optimal discharge strategy for a lithium-ion battery using a physics-based model," *Journal of Power Sources*, vol. 276, pp. 195 – 202, 2015.

[81] K. Goebel, B. Saha, A. Saxena, J. Celaya, and J. Christophersen, "Prognostics in battery health management," *IEEE Instrumentation & Measurement Magazine*, vol. 11, no. 4, pp. 33–40, Aug 2008.

[82] S. Chakraborty, A. Jain, and N. Mohan, "Novel converter topology and algorithm for simultaneous charging and individual cell balancing of multiple li-ion batteries," *10th International Workshop on Computational Electronics (IEEE Cat. No.04EX915)*, 2004.

[83] [Online]. Available: www.sdge.com

[84] A. Sakti, J. J. Michalek, E. R. Fuchs, and J. F. Whitacre, "A techno-economic analysis and optimization of li-ion batteries for light-duty passenger vehicle electrification," *Journal of Power Sources*, vol. 273, pp. 966–980, Jan 2015.

[85] J. Venkatesh, B. Aksanli, J.-C. Junqua, P. Morin, and T. Rosing, "Homesim: Comprehensive, smart, residential electrical energy simulation and scheduling," in *Green Computing Conference (IGCC), 2013 International*, 2013, pp. 1–8.

[86] W. Tushar, B. Chai, C. Yuen, S. Huang, D. B. Smith, H. V. Poor, and Z. Yang, "Energy storage sharing in smart grid: A modified auction-based approach," *IEEE Transactions on Smart Grid*, vol. 7, no. 3, pp. 1462–1475, May 2016.

[87] K. Mets, T. Verschueren, F. D. Turck, and C. Develder, "Exploiting v2g to optimize residential energy consumption with electrical vehicle

(dis)charging," in *Smart Grid Modeling and Simulation (SGMS), 2011 IEEE First International Workshop on*, Oct 2011, pp. 7–12.

[88] D. T. Nguyen and L. B. Le, "Joint optimization of electric vehicle and home energy scheduling considering user comfort preference," *IEEE Transactions on Smart Grid*, vol. 5, no. 1, pp. 188–199, 2014.

[89] E. P. R. Institute, "EPRI Smart Grid Demonstration Initiative Two Year Update," Electric Power Research Institute, Tech. Rep., 2010. [Online]. Available: http://smartgrid.epri.com

[90] J. Venkatesh, B. Aksanli, J.-C. Junqua, P. Morin, and T. Rosing, "Homesim: Comprehensive, smart, residential electrical energy simulation and scheduling," in *Green Computing Conference (IGCC), 2013 International*, June 2013, pp. 1–8.

[91] B. Aksanli, J. Venkatesh, and T. Rosing, "Using datacenter simulation to evaluate green energy integration," *Computer*, vol. 45, no. 9, pp. 56–64, Sept 2012.

[92] A. Akyurek, B. Torre, and T. Rosing, "Eco-dac energy control over divide and control," in *Smart Grid Communications (SmartGridComm), 2013 IEEE International Conference on*, Oct 2013, pp. 666–671.

[93] E. Dall'Anese, J. Doyle, S. Dhople, B. Johnson, N. Li, S. Low, B. Lundstrom, A. Paylan, R. Sherick, and A. Wierman, "Federated smart grid testbed," Carnegie Mellon University, Tech. Rep., March 2015.

[94] J. Arrinda, J. Barrena, and M. Rodriguez, "Distribution network simulation method based on a combination of dynamic power-flow simulation and electro-magnetic simulation," in *EUROCON, 2013 IEEE*, 2013, pp. 1336–1343.

[95] F. Angstmann, A. Bracher, S. Bhat, and S. Ramaswamy, "A smart grid simulation framework for electricity trading," in *Networking, Sensing and Control (ICNSC), 2013 10th IEEE International Conference on*, 2013, pp. 609–614.

[96] Y. Deng, S. Foo, and H. Li, "Real time simulation of power flow control strategies for fuel cell vehicle with energy storage by using real time digital simulator (rtds)," in *Power Electronics and Motion Control Conference, 2009. IPEMC '09. IEEE 6th International*, 2009, pp. 2323–2327.

[97] W. Price, C. Taylor, and G. Rogers, "Standard load models for power flow and dynamic performance simulation," *IEEE Transactions on Power Systems*, vol. 10, no. 3, pp. 1302–1313, Aug 1995.

[98] B. Aksanli, A. S. Akyurek, M. Behl, M. Clark, A. Donzé, P. Dutta, P. Lazik, M. Maasoumy, R. Mangharam, T. X. Nghiem, V. Raman, A. Rowe, A. Sangiovanni-Vincentelli, S. Seshia, T. S. Rosing, and J. Venkatesh, "Distributed control of a swarm of buildings connected to a smart grid: Demo abstract," in *Proceedings of the 1st ACM Conference on Embedded Systems for Energy-Efficient Buildings*, ser. BuildSys '14. New York, NY, USA: ACM, 2014, pp. 172–173.

[99] X. Jiang, S. Dawson-Haggerty, P. Dutta, and D. Culler, "Design and implementation of a high-fidelity ac metering network," in *Proceedings of the 2009 International Conference on Information Processing in Sensor Networks*, ser. IPSN '09. Washington, DC, USA: IEEE Computer Society, 2009, pp. 253–264.

[100] A. Rowe, M. Berges, G. Bhatia, E. Goldman, R. Rajkumar, J. Garrett, J. Moura, and L. Soibelman, "Sensor andrew: Large-scale campus-wide sensing and actuation," *IBM Journal of Research and Development*, vol. 55, Jan 2011.

[101] M. Fiorani, P. Monti, B. Skubic, J. Mårtensson, L. Valcarenghi, P. Castoldi, and L. Wosinska, "Challenges for 5g transport networks," in *Proc. IEEE Int. Conf. Advanced Networks and Telecommuncations Systems (ANTS)*, Dec. 2014, pp. 1–6.

[102] "Mobile and wireless communications enablers for twenty-twenty (2020) information society: metis-ii.5g-ppp.eu." [Online]. Available: metis-ii.5g-ppp.eu

[103] M. Agiwal, A. Roy, and N. Saxena, "Next generation 5g wireless networks: A comprehensive survey," *IEEE Communications Surveys Tutorials*, vol. 18, no. 3, pp. 1617–1655, thirdquarter 2016.

[104] D. Muirhead, M. A. Imran, and K. Arshad, "A survey of the challenges, opportunities and use of multiple antennas in current and future 5g small cell base stations," *IEEE Access*, vol. 4, pp. 2952–2964, 2016.

[105] K. Kim, S. Ganguly, R. Izmailov, and S. Hong, "On packet aggregation mechanisms for improving voip quality in mesh networks," in *Vehicular Technology Conference, 2006. VTC 2006-Spring. IEEE 63rd*, vol. 2, May 2006, pp. 891–895.

[106] W. B. Heinzelman, A. P. Chandrakasan, and H. Balakrishnan, "An application-specific protocol architecture for wireless microsensor networks," *IEEE Transactions on Wireless Communications*, vol. 1, no. 4, pp. 660–670, Oct 2002.

[107] J. L. Zechinelli-Martini, P. Bucciol, and G. Vargas-Solar, "Energy aware data aggregation in wireless sensor networks," in *Proc. 2nd Int Wireless Communication, Vehicular Technology, Information Theory and Aerospace Electronic Systems Technology (Wireless VITAE) Conf*, Feb. 2011, pp. 1–5.

[108] S. Siddiqui, A. A. Khan, and S. Ghani, "A survey on data aggregation mechanisms in wireless sensor networks," in *Proc. Int. Conf. Information and Communication Technologies (ICICT)*, Dec. 2015, pp. 1–7.

[109] I. H. Brahmi, F. Maire, S. Djahel, and J. Murphy, "Planning acting: Optimal Markov decision scheduling of aggregated data in wsns by genetic algorithm," in *Proc. IEEE 26th Annual Int Personal, Indoor, and Mobile Radio Communications (PIMRC) Symp*, Aug. 2015, pp. 2066–2071.

[110] B. Błaszczyszyn, M. Jovanovic, and M. K. Karray, "Performance laws of large heterogeneous cellular networks," in *Proc. 13th Int Modeling and Optimization in Mobile, Ad Hoc, and Wireless Networks (WiOpt) Symp*, May 2015, pp. 597–604.

[111] "Network simulator 3: nsnam.org." [Online]. Available: nsnam.org

[112] Cc-2650: Simplelink multi-standard 2.4 ghz ultra-low power wireless mcu. [Online]. Available: ti.com/product/CC2650

[113] A. Medeisis and A. Kajackas, "On the use of the universal okumura-hata propagation prediction model in rural areas," in *Vehicular Technology Conference Proceedings, 2000. VTC 2000-Spring Tokyo. 2000 IEEE 51st*, vol. 3. IEEE, 2000, pp. 1815–1818.

[114] "International telecommunication union: itu.int." [Online]. Available: itu.int

[115] J. Venkatesh, C. Chan, A. S. Akyurek, and T. S. Rosing, "A modular approach to context-aware iot applications," in *IEEE First International Conference on Internet-of-Things Design and Implementation (IoTDI)*, April 2016, pp. 235–240.

[116] J. Venkatesh, B. Aksanli, C. Chan, A. S. Akyurek, and T. S. Rosing, "Scalable application design for the internet of things," *IEEE Software Special Issue on Software Engineering for the Internet of Things*, vol. to appear, 2016.