

SWARM BEHAVIOR TO MITIGATE REBOUND IN AIR
CONDITIONING DEMAND RESPONSE EVENTS

by

Jason Yasuto Kuwada



A thesis

submitted in partial fulfillment

of the requirements for the degree of

Master of Science in Mechanical Engineering

Boise State University

August 2019

© 2019
Jason Yasuto Kuwada
ALL RIGHTS RESERVED

BOISE STATE UNIVERSITY GRADUATE COLLEGE

DEFENSE COMMITTEE AND FINAL READING APPROVALS

of the thesis submitted by

Jason Yasuto Kuwada

Thesis Title: Swarm Behavior to Mitigate Rebound in Air Conditioning Demand Response Events

Date of Final Oral Examination: 30th August 2019

The following individuals read and discussed the thesis submitted by student Jason Yasuto Kuwada, and they evaluated the presentation and response to questions during the final oral examination. They found that the student passed the final oral examination.

John F Gardner, Ph.D.

Chair, Supervisory Committee

Hoda Mehrpouyan, Ph.D.

Member, Supervisory Committee

Aykut Satici, Ph.D.

Member, Supervisory Committee

The final reading approval of the thesis was granted by John F Gardner, Ph.D., Chair of the Supervisory Committee. The thesis was approved by the Graduate College.

This one's for you Mike. I hope you found your peace.

ACKNOWLEDGMENTS

I would like to thank Dr. John Gardner for your expertise and patience along with the opportunity to work on this project. Your mentorship has helped me grow personally and professionally. Thank you to Dr. Hoda Mehrpouyan and Dr. Aykut Satici for contributing valuable insight.

I would also like to thank my colleagues, Ryan Schwartz and Sukjoon Oh for their foundational work. Additionally, I'd like to recognize Travis Pruitt and the rest of the team at the IAC. Your companionship has been key to the completion of this research. Thank you to my family and friends for their continued support. In particular, I would like to thank Cierra for her ongoing encouragement throughout the entirety of my work.

Finally, I would like to thank the Department of Energy's Industrial Assessment Center for funding my work and making all of this possible.

ABSTRACT

Thermostatically Controlled Loads (TCLs) have shown great potential for Demand Response (DR) events. However, it has been commonly seen that DR events using TCLs may cause demand rebound, especially in homogeneous populations. To further explore the potential for DR events, as well as the negative effects, a stability and resilience analysis were performed on multiple populations and verified with agent based modeling simulations.

At the core of this study is an added thermostat criterion created from the combination of a proportional gain and the average compressor operating state of neighboring TCLs. Where DR events in TCLs are commonly controlled by set point manipulation, the modified thermostat behavior proposed in this study alters the effective dead band of each individual TCL. Previous work has shown the effectiveness of the proposed behavior to mitigating the demand rebound.

By adding the average operating state of neighboring TCLs and a proportional gain, the systems feedback is changed, opening the possibilities to creating an unstable response. Stability limit are found from linearized systems, differing in delay schemes and connection architecture. The stability analysis was verified through agent-based modeling simulations on MATLAB. The linearization assumption was tested by simulating the systems while altering the parameters of population size and thermostat dead band.

Resilience of several systems, differing in connection architecture, is computed and

compared to results of a simulated denial of service attack on the system. Resilience for each architecture was calculated using the algebraic connectivity of the graph. The simulated attack is completed by removing the TCLs ability to communicate with in the agent based model.

The stability analysis showed the effect of the gain value on the performance of the system and that the stability limit was directly affected by the effective deadband. As the deadband size was increased, the predicted results found from the analysis aligned with simulations of the system. Contrarily the resilience analysis was validated by simulations with smaller deadband sizes. Simulations of cyber-attacks also showed optimal attacks based on operating state of thermostats, as well as locations within the population.

TABLE OF CONTENTS

ABSTRACT	vi
LIST OF TABLES	xi
LIST OF FIGURES	xiii
1 Introduction	1
1.1 Background	1
1.2 New Technologies and Electrical Distribution Models	5
1.2.1 Centralized Control	7
1.2.2 Decentralized Control	8
1.3 Demand Rebound	9
2 Modeling Approach	12
2.0.1 Heat Transfer Dynamics of Agents	13
2.0.2 Peer to Peer Communication	14
2.0.3 Modified Thermostat Behavior	17
2.1 Delays and Discrete Time Domain	19
2.1.1 Discretization of the Heat Transfer Equation	19
2.1.2 Implemented Delays	21
2.2 System Parameters	23
2.3 Baseline Behavior	26

3	Stability	28
3.1	Background	28
3.1.1	What is a System	28
3.1.2	Stability	29
3.2	Calculating Stability	30
3.2.1	Linearization	30
3.2.2	Approach	36
3.3	Validation of the Linearized System	37
3.3.1	Gain Value effect on Dead Band	42
3.4	Heterogeneous Simulations	48
3.5	Stability Analysis Conclusions	49
3.6	Future Work	51
4	Resilient System Design	52
4.1	Security And Resilience	52
4.2	Simulated Attack	55
4.3	Results	57
4.4	Resilience Conclusions	61
4.5	Future Work	63
5	Conclusions	64
	REFERENCES	66
A	Stability Metrics	69
A.1	Indices of Performance for Varying Populations	69

B Resilience Metrics	75
C Source Code	88
C.1 Ring Lattice Generator	89
C.2 Square Lattice Generator	90
C.3 Eigenvalue Analysis	91
C.4 Create MATLAB Animation	94
C.5 Homogenous Linear	95
C.6 Homogenous Non-Linear	100
C.7 Heterogeneous Parameters	107
C.8 Heterogeneous Varying Ambient Temperature	109
C.9 Deadband Analysis	114

LIST OF TABLES

2.1	Population Parameter Values	25
3.1	Gain Values at Stability Limit for Varying Population Sizes	35
3.2	Gain Values at Stability Limit for Varying Deadband Sizes	35
3.3	Eigenvalue Analysis Compared to Linear Simulation	37
3.4	Indices of Performance for Varying Gain Values	40
4.1	Percent Increase for Varying Population with 20% Failure	58
4.3	Number of Homes till Failure with Varying Location	60
4.4	Summary of Location Effect from Cyber Attacks	61
A.5	Indices of Performance, Ring, Last M, $\delta = 0.2$	74
B.1	IOP of Resilience with Concurrent, Square Lattice, & $\delta = 0.5$	76
B.2	IOP of Resilience with Concurrent, Ring Lattice, & $\delta = 0.5$	77
B.3	IOP of Resilience with Last m, Square Lattice, & $\delta = 0.5$	78
B.4	IOP of Resilience with Last m, Ring Lattice, & $\delta = 0.5$	79
B.5	IOP of Resilience with Full Delay, Square Lattice, & $\delta = 0.5$	80
B.6	IOP of Resilience with Full Delay, Ring Lattice, & $\delta = 0.5$	81
B.7	IOP of Resilience with Concurrent, Square Lattice, & $\delta = 0.2$	82
B.8	IOP of Resilience with Concurrent, Ring Lattice, & $\delta = 0.2$	83
B.9	IOP of Resilience with Last m, Square Lattice, & $\delta = 0.2$	84

B.10 IOP of Resilience with Last m, Ring Lattice, & $\delta = 0.2$	85
B.11 IOP of Resilience with Full Delay, Square Lattice, & $\delta = 0.2$	86
B.12 IOP of Resilience with Full Delay, Ring Lattice, & $\delta = 0.2$	87

LIST OF FIGURES

1.1	CASIO Duck Chart	3
1.2	Example of demand rebound in a homogeneous population.	10
2.1	Example of common graph topology.	17
2.2	Example of (a) Homogeneous Neighborhood and (b) Heterogeneous Neighborhood in Boise, Idaho via Google Earth.	24
2.3	Example of hourly TMY data for Boise (July 21st) fitted with a spline.	25
2.4	Example of demand rebound in a homogeneous population.	26
2.5	Example of demand rebound in a Heterogeneous population.	27
3.1	Visual representation of linearized thermostat.	30
3.2	Example of an unstable response of the linear system.	36
3.3	Predicted unstable response of the non-linear system.	38
3.4	Indices of performance for varying gain values.	41
3.5	Grouping of temperature trajectories.	42
3.6	Rapid cycling of TCLs.	43
3.7	Effective deadbands with varying gain values.	44
3.8	Example of rapid cycling due to deadband changes.	45
3.9	Rapid cycling between hours 2 and 4 of simulation.	46
3.10	Operating state of rapid cycling TCL.	47
3.11	Histogram of \tilde{m} values during two hour period.	48

3.12	Example response of heterogeneous population.	49
3.13	Indices of performance for heterogeneous populations.	50
4.1	Algebraic Connectivity for Varying Populations of Homes.	57
4.2	Deadband effects on percent increase.	59
4.3	Locational effects on percent increase.	61

CHAPTER 1

INTRODUCTION

1.1 Background

Electric utilities have their origins in the late 19th century and have evolved in the technological and social environment of the early 20th century. This has led to the current situation in which consumers have come to expect that electricity will always be available, regardless of when and how a consumer requires it, and utilities accept the responsibility of meeting those requirements. Utilities pull from a variety of generation sources to produce the required power. A subset of these generation sources (i.e nuclear, coal) continually operate, utilizing large thermal masses that are cost efficient at supplying a constant amount of power to the grid, but are unable to respond quickly to changes in demand.

The rapid deployment of intermittent renewable energy generation on the electric grid has presented significant challenges to the entities that are responsible for maintaining the reliability of our electric distribution system. In 2016, China led the expansion for solar photovoltaic (PV) growth, pushing the global PV capacity up by 50% [1]. With this spike in solar capacity, renewables account for nearly two-thirds, or approximately 165 gigawatts of net new power in the 2016. Renewables are forecasted to continue to grow through the year 2022 due to technology cost reductions and changing market dynamics. With its new policy changes, China continues to pioneer

the growth capacity for renewables, already surpassing its 2020 solar PV target, (as of 2017). The two main challenge facing China's growth in renewables are the cost of renewable subsidies and grid integration.

Since the grid has very little inherent ability to store energy, the variable nature of wind and solar energy forces the utility to constantly compensate for those variations by modifying the output of the resources they do control. Typically, this compensation is done by changing the output of hydropower or natural gas generators, which can reach full generation in a matter of minutes, allowing for tracking of peak loads or filling in the valleys caused by a renewable generation source suddenly losing power production (e.g. cloud covers solar farm, wind quickly dies down, etc.).

Similarly, California is facing problems with the increase of renewables on the grid. In 2013, the California Independent System Operator (CAISO) introduced the "Duck Chart" [2]. Figure 1.1 shows a series of lines, each representing the net energy demand in the California grid during a spring day (31 March) on each of 8 years. Net energy is the energy provided by the electric utilities, which is equal to the energy consumed minus the solar energy produced by distributed (non-utility owned) solar resources. The base line of the graph (labeled 2012 (actual)) is prior to California's incentives to promote installation of customer-owned solar generation. The general shape of that line, with a minimum around 3 AM, a peak mid morning, a lull in mid-afternoon and a high peak around dinner time, is typical of an aggregated electrical demand curve throughout the developed world. Additional lines show the impact that the increasing penetration of solar (2013) and that of the growth of solar penetration predicted at the time of the study. As solar generation increases (because more facilities are built) the net demand that utilities must meet changes drastically during the daylight hours. However, the load required at night is not impacted.

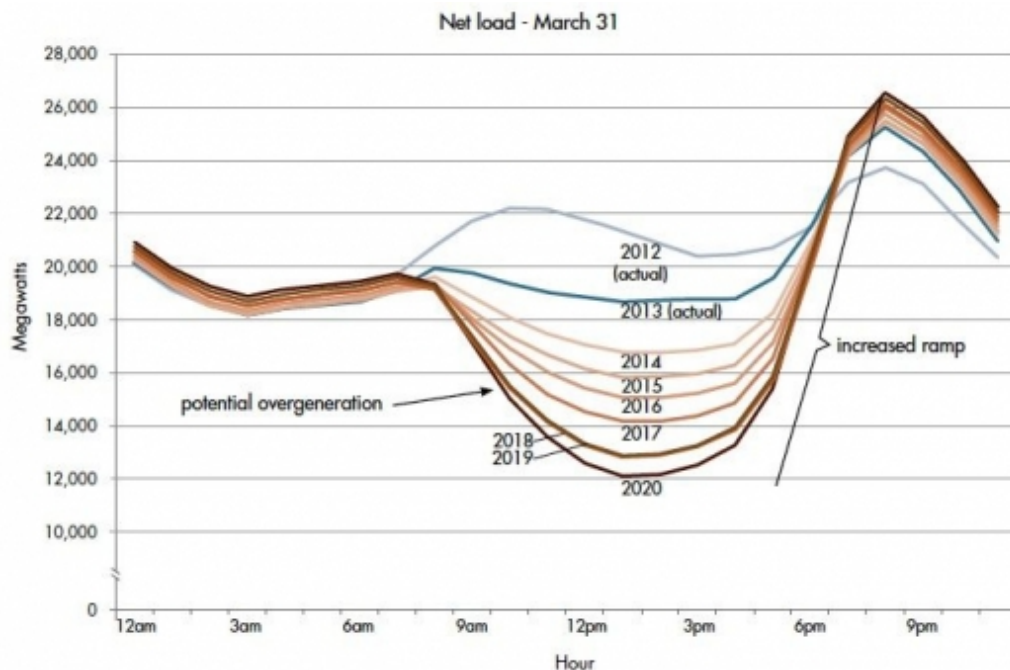


Figure 1.1: CASIO Duck Chart

There are two major areas of concern that the duck curve highlights. One is that the greatly reduced net demand at mid-day might lead to over generation due to other constraints on grid operation. Over generation is an issue because the so-called base load generating stations cannot be economically ramped up and down to meet this relatively short period of high solar generation. As an attempt to mitigate over-generation the system operator curtails the amount of power that it's VG can produce. The act of curtailment is done by changing the blade pitch on a wind turbine or reducing the output from an inverter on a solar PV. Having this type of control over VG sources is typical for larger power plants but not always an option for smaller systems, such as roof top or distributed systems. Curtailment also reduces the benefits of VG on the grid. For "every unit of VG curtailed represents a unit of energy not sold on to the grid and a unit of fossil fuel not avoided." [2]

The second issue is a rapid change in demand that utilities must compensate for as the sun sets across the region. Even those generators that can change their output have limitations as to how fast those changes can occur. Therefore, the concern is that there can come a time when the steepness of the ramp exceeds the ability of utilities to meet rapidly changing needs.

While California works to mitigate the evolution of the Duck Chart, Hawai'i is already facing the effects of over-generation from solar PV [3]. Where the predictions of the Duck Chart forecast reduced demand, Hawaii has seen system wide demand below zero. In 2013 the local utility, Hawaii Electric Company (HECO), recorded instances where the grid receives enough solar generation to drive demand plunge below zero on a system wide level, causing power to back feed into certain distribution circuits. In response, HECO placed new interconnection requirements which halted the growth of solar PV.

Unlike California, Hawai'i's challenges with over generation from solar PV comes mainly from the large amount of distributed systems on the grid. While revisiting the CASIO study, Scott Madden found that California's Duck Chart is not driven from distributed systems, but from utility scale solar [4]. In this 2016 study, it was also found the predictions depicted in the duck chart are occurring, and faster than originally forecasted. This shows that not only states with large penetration from distributed systems should be concerned with effects similar to the Duck Chart.

In addition to these problems exposed by California and other system operators relative to solar energy, other utilities and independent operators find related challenges to the growing impact of wind generation capacity. These concerns are the main motivations for developing grid-level energy storage solutions in the form of electro-chemical batteries. Grid-scale battery deployment is seen as the main solution

to these problems, although they come at considerable expense and the long-term environmental impacts and costs associated with these technologies remain uncertain [5].

1.2 New Technologies and Electrical Distribution Models

Batteries are only one way to tame the duck curve. A growing number of energy observers are pointing out that by enlisting the cooperation of electricity users, and incentivizing changes in their behavior, we can impact the problem by changing consumption at critical times of the day [6]. Increasing development of the electrical smart grid offers unprecedented opportunity for more complex electrical supply and demand interactions in a relationship that has been historically unilateral. The smart grid allows for the application of modern communication technology, such as the internet of things, to improve or modify widespread electrical transmission and distribution.

In addition to developments related to the smart grid, utilities have begun to embrace Demand Side Management (DSM) as a way of helping it meet customer needs. In the broadest sense, DSM refers to those programs implemented by a utility (or independent grid operator) in which they endeavor to modify (or manage) customer use of electricity in some way. DSM programs mainly fall into two categories: (1) Energy Efficiency programs such as LED lighting or insulation upgrades, and (2) Demand response programs utilities use to incentivize deferral of energy consumption during times of high demand.

This shift in the utility-customer relationship has led observers to speculate that we are at the dawn of a new era in which the relationship between energy providers

and customers is re-defined in a way that empowers individual consumers to provide services back to the grid in addition to simply consuming. The terms used to describe this new relationship include the producing consumer, or *prosumer* [7], and the *Energy Internet* [8]. The speculation around the future of the prosumer is, until now, limited to the relationships between individual prosumers and the utility (or with third party aggregators acting on behalf of the utility).

Much of this activity is focused on thermostatically-controlled loads (TCLs) such as those systems used for space heating and cooling, hot water or refrigeration and food storage. Systems that use electricity in this manner are normally designed to maintain temperature, not at a single constant set point, but within a range of temperatures, known as the thermostat deadband.

In a typical DR application, residential air conditioning compressors (but not the circulating fan) are temporarily turned off under control of the utility, allowing the inside temperature to rise above the thermostat setting. The grid is relieved of the load that the AC compressor would have drawn and the homeowner (if present) experiences a small and possibly noticeable increase in indoor temperature. These programs are nearly universally used to shed load at times of very high demand, but they can also be used to increase consumption at times of energy surplus, resulting in a somewhat cooler home than the set point would imply. A common interpretation of this effect is that energy is being stored in the thermal mass of the home. In this respect, the home acts like a thermal battery, albeit a leaky one with limited storage capacity.

The ability of any individual TCL to impact the energy balance of the grid is limited. Therefore, most applications entail the aggregation of many hundreds and thousands of loads, coordinated by a central controller. The dynamics and control of

such aggregated loads is the subject of significant research [6], [9], [10] and grid operators across the country have significant experience using aggregated DR to manage peak loads.

1.2.1 Centralized Control

Grid operators typically exert central control over large populations of TCLs through radio links or power line carrier protocols. For example, Idaho Power has an AC Cool Credit program where individual consumers, in exchange for a small reduction in their monthly electric bill, give the utility permission to install equipment on their AC unit [11]. This equipment can receive a signal from Idaho Power that shuts off the unit's compressor for a short period of time. After that time has expired, the unit is allowed to turn back on while another population is disconnected, thus reducing the overall demand for as long as required and preventing uncomfortable conditions from occurring within any individual homes.

Another common method of controlling TCLs is through set-point control. Callaway [6] makes the case that changing thermostat set-points of a population of TCLs can be used to follow the variability of wind generation. Building on this work, Bashash and Fathy [12, 13] developed a model that uses a centralized controller to broadcast a uniform signal to vary the thermostat set-point temperature of the population of TCLs. This enables the tracking of a real wind power trajectory.

A centralized controller can coordinate many agents without knowing individual agent states. By comparing a reference to a received aggregate output value, the controller determines what signal to broadcast to all agents. Each agent then makes a decision based on the signal and defined probabilities [14].

Similarly, Zhang et al. [15] developed a control scheme where the centralized control signal is broadcast to all agents. The agents then decide how to implement the signal based on their local temperature and power state.

A priority-stack-based control strategy can be an effective way to control TLCs [16]. Sorting the population of TLCs by temperature into two stacks, one where the TLCs are off and the other where the TLCs are on, allows for the most appropriate selection of the next TCL to turn on or off. When the grid has excess power, perhaps due to an increase in wind generation, the centralized controller can send a signal to the TLCs. The TCL with the highest priority in the off stack will turn on first and then continue down the stack until the excess power is being utilized. This also works the other way. When the grid is trying to reduce load, the TLCs in the on column turn off sequentially, in order of priority, until the desired reduction has been met. The downside of this model is the need to have the information of all agents in the system accessible to the central controller to sort correctly.

1.2.2 Decentralized Control

In decentralized control, agents take in information and make decisions as individuals, rather than being told what to do by one controller.

One method of decentralized demand response is to use price signalling and adaptive mechanisms, coupled with smart meters, to prevent loads from syncing up and creating high peak demand [17].

With the rising number of electric vehicles (EVs), an additional opportunity for demand response has been created. EV charging management allows for the vehicles to charge at times of low demand or to match renewable energy production. Xydas et al. [18] developed a model to demonstrate the effectiveness of "responsive" EVs.

These vehicles determine their charging schedule according to a signal that takes power demand and generation forecasts into account. Their model demonstrated that responsive EVs could reduce the peak charging demand of all EVs, including unresponsive EVs, by shifting demand to a time when the unresponsive EVs were finished charging. They also demonstrated the ability of responsive EVs to charge in response to a real time photovoltaic (PV) generation profile.

1.3 Demand Rebound

DR events in populations of TCLs have the potential to create a rebound effect in the aggregate power. This rebound is caused by the synchronization of the individual TCLs. Homogeneous populations are especially prone to synchronization due to each home having the same time constant, or time in which the houses take to heat and cool. An example of demand rebound in a population of homogeneous TCLs is shown in Fig.1.2. Details on the simulation will be covered in the following chapter, but it is important to note that this simulation models 100 homogeneous TCLs starting at an equilibrium point. The equilibrium point is created by a distribution of indoor temperatures throughout the population and causes a steady response for the aggregate power. At two hours into the simulation, a DR event is introduced. The DR event forcibly turns the AC compressor of all the TCLs in the population to an Off state for 15 minutes. During this 15 minute time period, all the TCLs heat up or move to the upper limit of the deadband and wait to turn back on. After the DR event all the houses turn On as soon as they are allowed, this is the cause for the synchronous behavior, creating large ramps and falls in the aggregate power.

Schwartz showed the potential for local communication to mitigate rebound in

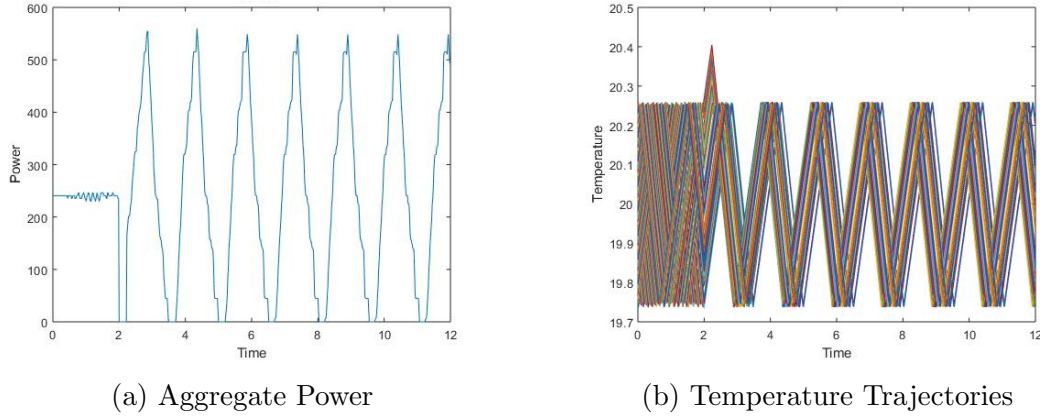


Figure 1.2: Example of demand rebound in a homogeneous population.

air conditioning demand response events [19]. While Schwartz’s work showed the positive impact of changing the system dynamics using local communication, it raised questions of stability and resilience. The added connectivity of TCLs within a population allow increased freedom for individual TCLs to move between operating states. While the increased freedom has been shown to mitigate the synchronization in TCLs (which is the cause of demand rebound), the method of implementation introduced feedback which can lead to instability of the population. Additionally, this technique depends on mesh-like communication networks which may create vulnerabilities for cyber-physicals attacks on the system, leading to the need for resilience to be inherently designed into the system.

The thesis of this work is to develop a framework to understand the role that local communication, or a ‘mesh network’ can play in mitigating demand rebound and other beneficial behaviors. As such, this work aims to tie the dynamics of the system to theoretical bases in both stability and resilience. The rest of the paper is organized as follows. Chapter 2 describes the modeling approach for the simulations used in this study as well as the proposed thermostat modification. Chapter 3 shows

the effect the proposed thermostat modification has on the stability of the system. Chapter 4 give insight into the resilience benefits of the mesh network and finally, Chapter 5 reviews the important findings of this research as a well as future work.

CHAPTER 2

MODELING APPROACH

The majority of the modeling techniques used by Schwartz are incorporated into the techniques used to lay the groundwork for the this study [19], including the use of Agent-based Modeling (ABM) techniques. ABM has been a fixture of the social sciences for many years and has proven to be an effective means of understanding emergent behavior from a large number of individual actors. More recently, ABM has been identified as a means of understanding complex physical systems, such as the electric grid, where large numbers of individual homes or even appliances can be modeled in a coherent framework that allows for easy control of the statistical distribution of agent parameters and behavior [15].

In order to show the aggregate power of the population, as well as observe the individual effects on each TCL, agent based modeling was chosen. Agent based modeling allows the observer to define agents and their interactions with the environment and each other. The agents in the model are residential houses equipped with with a smart thermostat with the ability to communicate to it's nearest neighbors [20]. The thermostats control the houses' air conditioning (AC) units based off of interactions with the environment, (ambient outdoor temperature), and communications with neighboring houses.

2.0.1 Heat Transfer Dynamics of Agents

The proposed work grows on previous research. At the base of this work is the numerical approximation to the coupled Fokker-Planck equations derived from Malhamé and Chong [9]. The Fokker-Planck equations describe the statistical distribution of the individual agents and how the distributions evolve over time. There are no closed form solutions to these equations, as such they are approximated numerically.

Callaway furthers this work by contributing equivalent parameters for the first order model and showing populations of thermostatically controlled loads can act as virtual storage devices by collectively managing them [6]. Through this work, Callaway found load populations with greater heterogeneity are better candidates for set-point control of thermostatically controlled loads, where previous work focused on homogeneous populations. Bashash and Fathy modeled each thermostatically controlled load with their own first order differential equation and used Monté Carlo simulations to represent the community [12]. The heat transfer dynamics for a single house, as defined by Bashash and Fathy, are shown in

$$\dot{T} = \frac{1}{RC} \left(T_{\infty} - T(t) + R(Q_I - m(t)Q) \right) \quad (2.1)$$

where:

$T(t)$	Indoor Air Temperature	°C
T_{∞}	Outdoor Air Temperature	°C
C	Thermal Capacitance of Building	kWh/°C
R	Thermal Resistance of Building	°C/kWh
Q_I	Internal Heat Gain	kW
Q	Load Cooling Capacity	kW
$m(t)$	Discrete State of AC Power	-

It should be noted that the variable $m(t)$ is a discrete variable representing the

operating state of the air conditioning unit, having a value of one if the unit is operating and a value of zero if the unit is off. This is shown mathematically in Equation 2.2.

$$m(t) = \begin{cases} 0, & \text{if } T(t) \leq T_{min} \\ 1, & \text{if } T(t) \geq T_{max} \\ m(t^-), & \text{otherwise} \end{cases} \quad (2.2)$$

where T_{min} and T_{max} are the lower and upper limits of the thermostat deadband, δ . The setpoint temperature, T_{sp} , is related to these limits as shown in Equation 2.3.

$$T_{min} = T_{sp} - \frac{\delta}{2}, \quad T_{max} = T_{sp} + \frac{\delta}{2} \quad (2.3)$$

Considering a population containing N , number, of TCLs, the total load can be expressed as

$$P_{TCL}(t) = \sum_{i=1}^N \frac{1}{\eta_i} \bar{Q}_i m_i(t) \quad (2.4)$$

where η_i is the coefficient of performance (COP) of the i^{th} load.

2.0.2 Peer to Peer Communication

The core of this approach is a small amount of information sharing between thermostats that are in close proximity. For this study, thermostats allow each home to be aware of the on/off state of the compressors in the four nearest homes. The selection of the connections may be defined by the layout of the neighborhood (e.g. the

next door neighbors, the house across the street and over the fence in the backyard) or they may be defined by the topology of the electric distribution system.

The logistics of information sharing are not covered in this study, but it is clear that a large number of options are available covering a spectrum of technologies from internet-based server models where the connections can be implemented and programmed centrally, to local communication protocols such as Zigbee, Bluetooth and power-line carrier methods [21].

It is also important to address privacy concerns in information sharing situations. These concerns are addressed by noting that the information being shared (whether the compressor is on or off) is something a typical next door neighbor can observe by opening the window that is nearest the neighbor's outdoor condenser unit. In some locations, that information could be deduced from a careful observer on the public sidewalk. This information is likened to be similar to whether or not inside lights are on. While it is not usually noticed, it is clearly observable from the outside.

Graph theory is the mathematical lens through which a network of connected houses can be viewed. As Barabási [22] explains, in graph theory the network (or graph) is described as a set of nodes (the agents) and edges (links between agents). The degree, d , of a node describes the number of connections that node has to other nodes. These connections between agents can be *directed* or *undirected*. In a directed link, connection is established in one direction from one agent to another, similar to citations in a paper or a webpage linking to another webpage. Other networks utilize undirected links, like the power grid where transmission line current can flow both directions. The connections between residential thermostats in this model are assumed to be undirected because the information (ON/OFF state of the compressors) flows both ways on the edges of the graph.

Networks of connections are often represented as an adjacency matrix, \mathcal{A} . For a network containing N nodes, the adjacency matrix has N rows and N columns containing elements that follow the rules:

$$\mathcal{A}_{ij} = \begin{cases} 0, & \text{if nodes } i \text{ and } j \text{ are not connected to each other} \\ 1, & \text{if nodes } i \text{ and } j \text{ are connected to each other} \end{cases} \quad (2.5)$$

For an undirected network the adjacency matrix is symmetric, $\mathcal{A}_{ij} = \mathcal{A}_{ji}$, and since a house is not connected to itself, the diagonal consists of zeros. The adjacency matrix can be used to find the degree of house i by summing either the column or the row corresponding to that house:

$$d_i = \sum_{j=1}^N \mathcal{A}_{ij} = \sum_{j=1}^N \mathcal{A}_{ji} \quad (2.6)$$

The graph structures utilized in this study are the square lattice and ring lattice. Both of these connection architectures are applicable to decentralized control. An example of a square lattice is shown in Fig. 2.1a and an example of a ring lattice is shown in Fig. 2.1b. For reference, a star lattice, Fig. 2.1c, is a common architecture for centralized control. In a star lattice, the central node acts as the nucleus of the graph; in application to a demand response system, this central node would represent the utility provider.

Now consider a situation in which the state of the AC unit, $m(t)$, for each house can be communicated from its thermostat to nearby connected thermostats. The variable \tilde{m}_i is introduced to represent the average state of the thermostats communicating with agent i . The adjacency matrix representing connected agents can be used to easily calculate all of these values simultaneously:

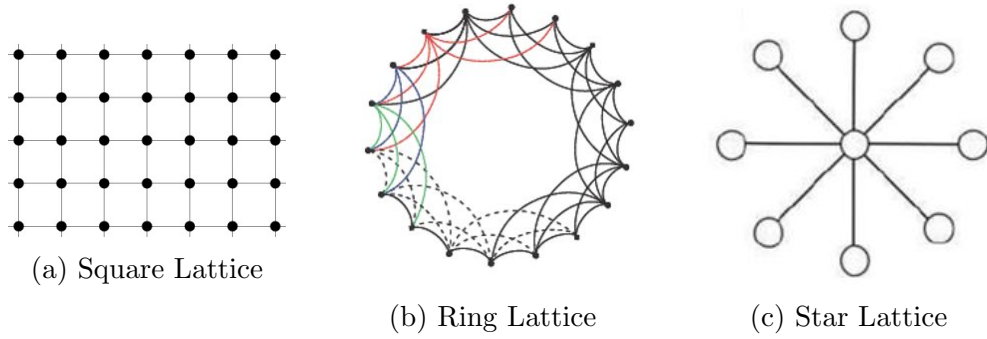


Figure 2.1: Example of common graph topology.

$$\tilde{\mathbf{m}} = \frac{1}{d} \mathcal{A} \mathbf{m} \quad (2.7)$$

In Equation 2.7, \tilde{m} and m are vectors of N , number of homes, representing the entire population. \mathcal{A} is a matrix of $N \times N$ size, as defined as in Equation 2.5. The number of connected homes, or degree, is denoted by d . For this study, d is a constant throughout the population equal to four, ($d = 4$). This means that each house is connected to four neighboring houses.

2.0.3 Modified Thermostat Behavior

Consider a new non-dimensional temperature variable defining a normalized position within the deadband, θ_i , where the bottom of the deadband is $\theta_i = 0$ and the top of the deadband is $\theta_i = 1$.

$$\theta_i = \frac{T_i - T_{min,i}}{T_{max,i} - T_{min,i}} = \frac{T_i - (T_{sp,i} - \frac{\delta_i}{2})}{\delta_i} \quad (2.8)$$

Now typical thermostat behavior can be described in terms of this normalized parameter instead of individual house temperatures and deadbands:

$$m_i(t) = \begin{cases} 0, & \text{if } \theta_i \leq 0 \\ 1, & \text{if } \theta_i \geq 1 \\ m_i(t^-), & \text{otherwise} \end{cases} \quad (2.9)$$

Here, a new addition to the thermostat model is proposed which uses the average state of the surrounding units, \tilde{m} , to inhibit operation based on the number of connected units that are operating.

$$m_i(t) = \begin{cases} 0, & \text{if } \theta_i - K_{gain}\tilde{m}_i \leq 0 \\ 1, & \text{if } \theta_i - K_{gain}\tilde{m}_i \geq 1 \\ m_i(t^-), & \text{otherwise} \end{cases} \quad (2.10)$$

The variable K_{gain} , is a proportional gain to the control feedback introduced. In the simplest term, K_{gain} is a knob that lets the user adjust the effectiveness of the \tilde{m} modification.

The addition of the average ON/OFF state of connected neighbors allows agents to reduce overall demand by causing an earlier entry to the OFF state and a later entry to the ON state, essentially shifting the deadband up, if a larger number of neighbors turn ON. For example, consider a network with $d = 4$, where two of a house's neighbors are ON, resulting in a \tilde{m} of 0.5. Assuming $K_{gain} = 1$, this house will turn OFF as soon as $\theta = 0.5$, or halfway through the deadband, instead of the standard $\theta = 0$, and won't turn ON until $\theta = 1.5$. With a connection degree of four, each thermostat has the ability to use five upper and five lower bounds depending on the state of neighboring thermostats.

2.1 Delays and Discrete Time Domain

While 2.1 accurately describes the heat transfer dynamics of a house using a simple thermostat, it is unlikely, in implementation, that the average state of neighboring thermostat will be available concurrently for the modified thermostat criteria. While it is possible to simulate a model which uses the average state of neighboring thermostats concurrently, this adds an algebraic loop because the newly computed state of thermostat is required to compute the state of adjacent thermostats. By moving from the continuous time domain to the discrete time domain, delays are able to be incorporated without introducing additional non-linearities, while making it possible to eliminate the implicit nature of the computation.

2.1.1 Discretization of the Heat Transfer Equation

The discretization follows the process outlined by Ogata [23], starting from the general state space model shown as:

$$\dot{x}(t) = Ax(t) + Bu(t) \quad (2.11)$$

$$y(t) = Cx(t) + Du(t) \quad (2.12)$$

The discretization was completed on a single agent, or a single state system. As such, the variables A , B , C , and D are scalar variables. These variables are defined in Equations 2.13-2.17. The population of TCLs are created by compiling N number of state equations. The same result can be seen from creating the state equations to represent vectors with N number of states. Equations 2.11 and 2.11 represent a single order equation, the state parameters are:

$$A = \frac{-1}{RC} \quad (2.13)$$

$$B_I = \frac{1}{RC} \quad (2.14)$$

$$B_{II} = \frac{Q}{C} \quad (2.15)$$

$$C = 0 \quad (2.16)$$

$$D = \frac{1}{\eta} \bar{Q}_i \quad (2.17)$$

It should be noted the the system has multiple inputs, of ambient outside temperature and HVAC operating state. Equations 2.14 & 2.15 define the variables for both inputs, ambient outside temperature and HVAC operating state, respectively. With equal sampling intervals, t_{samp} , it is assumed that Equations 2.11 & 2.12 the discrete time system can be shown in the form,

$$X((k+1)t_{samp}) = G(t_{samp})X(kt_{samp}) + H(t_{samp})U(kt_{samp}) \quad (2.18)$$

$$Y(kt_{samp}) = CX(kt_{samp}) + DU(kt_{samp}) \quad (2.19)$$

where $G(t_{samp})$ and $H(t_{samp})$ are through the solution of the continuous time system.

The general form of these equations can be seen below.

$$G(t_{samp}) = e^{At_{samp}} \quad (2.20)$$

$$H(t_{samp}) = (e^{At_{samp}} - I)BA^{-1} \quad (2.21)$$

It is to be noted that $H(t_{samp})$ can take this form if and only if the matrix A is invertible. For the house model, indoor air temperature, T , is the only state variable.

Ambient outdoor temperature, T_∞ , and compressor operating state, $m(t)$, are both inputs to the system. For a single home and a constant sampling interval of one minute we find Eq. 2.20 and Eq. 2.21 to be:

$$G(t_{samp}) = e^{\frac{t_{samp}}{RC}} \quad (2.22)$$

$$H_I(t_{samp}) = \left(e^{\frac{t_{samp}}{RC}} - I \right) \left(\frac{1}{RC} \right) \left(\frac{1}{RC} \right)^{-1} \quad (2.23)$$

$$H_{II}(t_{samp}) = \left(e^{\frac{t_{samp}}{RC}} - I \right) \left(\frac{Q}{C} \right) \left(\frac{1}{RC} \right)^{-1} \quad (2.24)$$

Plugging in Equations 2.22 - 2.24 into the discrete state space form, the complete equation for the heat transfer dynamics can be seen in Eq 2.25.

$$T(k+1) = GT(k) + H_I T_\infty + H_{II} m(k) \quad (2.25)$$

The number of TCLs simulated are based of Equation 2.26. L is chosen to change the size of the population to ensure the total number of TCLs in a model is a squared number. This is done due to the restrictions of constructing a square lattice.

$$N = L \cdot L \quad (2.26)$$

For any N , number of homes, in a simulation Equation 2.25 is calculated N times at each time step.

2.1.2 Implemented Delays

In the discrete time domain, delays are now able to be added without adding non-linearities. Delays are added to θ and \tilde{m} in the thermostat switching criteria in

different combinations. Three different delay schemes are introduced to the thermostat criteria: concurrent, last average operating state, and full delay.

The first of the three delay schemes is the most similar to that used in the continuous time domain. Here the current thermostat state is a function of the current values of neighboring states. This implementation is unlikely to be able to know the average operating state with the use on decentralized control. The switching criteria for the concurrent thermostat is shown in Ep. 2.27.

$$m_i(k) = \begin{cases} 0, & \text{if } \theta_i(k) - K_{gain}\tilde{m}_i(k) \leq 0 \\ 1, & \text{if } \theta_i(k) - K_{gain}\tilde{m}_i(k) \geq 1 \\ m_i(k-1), & \text{otherwise} \end{cases} \quad (2.27)$$

The second delay scheme uses a concurrent temperature reading and the last average operating state of neighboring compressors. For the remainder of this report, this delay scheme will be referenced as ‘last m’ delay. This delay scheme is the most likely to be implemented and can be seen in Ep. 2.28.

$$m_i(k) = \begin{cases} 0, & \text{if } \theta_i(k) - K_{gain}\tilde{m}_i(k-1) \leq 0 \\ 1, & \text{if } \theta_i(k) - K_{gain}\tilde{m}_i(k-1) \geq 1 \\ m_i(k-1), & \text{otherwise} \end{cases} \quad (2.28)$$

While the final delay scheme is also unlikely to be implemented, it is imported to understand the effect of the delays on the thermostat reading. As such the final delay scheme is a full delay, both the thermostat reading and average operating state of neighboring compressors are delayed. The full delay switching criteria can be found in Eq. 2.29.

$$m_i(k) = \begin{cases} 0, & \text{if } \theta_i(k-1) - K_{gain}\tilde{m}_i(k-1) \leq 0 \\ 1, & \text{if } \theta_i((k-1)T) - K_{gain}\tilde{m}_i(k-1) \geq 1 \\ m_i(k-1), & \text{otherwise} \end{cases} \quad (2.29)$$

These three delay schemes are implemented in simulations and analyzed throughout the remainder of the report.

2.2 System Parameters

Demand rebound, or the synchronization following a demand response event, is particularly common in populations of homogeneous TCLs. The majority of the study assumes homogeneity for the house parameters, such as thermal resistance and capacitance. A homogeneous population is assumed because having a population of identical TCLs in a distributed Demand Response (DR) event is the most likely case for synchronization, or more simply, the worst case possible. Parameter values were adopted from the [6] study and are displayed in the Table 2.1. While populations of homogeneous houses are not entirely realistic, they are a fair approximation for tract houses, also referred to as cookie cutter neighborhoods. An example of tract housing found in Boise, Idaho is shown in Figure 2.2a.

While simulating homogeneous houses gives insight to a more severe case, neighborhoods generally resemble Figure 2.2b. Heterogeneous houses are simulated to reinforce the result from the homogeneous populations. By confirming results found in the homogeneous populations, this study will test the feasibility of implementing the proposed thermostat modification.



Figure 2.2: Example of (a) Homogeneous Neighborhood and (b) Heterogeneous Neighborhood in Boise, Idaho via Google Earth.

Homes with varying size and construction material are estimated by creating a statistical distribution around the homogeneous values. A parameter spread of $\pm 15\%$ is desired, which results in the normal standard deviations found in Table 2.1. The values were generated using MATLAB's `RANDN()` function. The `RANDN()` function returns a sample of random numbers with a normal distribution based off of a mean and standard deviation provided.

The energy transfer rate of a house's AC unit is sized depending upon the thermal dynamics of the house. The homogeneous population of houses' 14 kW is equivalent to a 4 ton unit ($1 \text{ ton} = 3.5 \text{ kW}_{\text{th}}$), which, for these parameters, means that the cooling rate is $0.8 \text{ }^\circ\text{C/hr}$, or the temperature moves from the upper limit of the deadband to the lower limit in about 37.5 minutes. The necessary tonnage to achieve this cooling time for the heterogeneous population was calculated and then rounded up to the nearest half-ton to reflect sizes commercially available. The resulting range in unit sizes is 3.5-5 tons ($12.25\text{-}17.5 \text{ kW}_{\text{th}}$). Rounding up of the unit size results

in slight over sizing, which means some houses will cooler faster than 37.5 minutes and therefore cycle more often than their homogeneous counterpart. The minimum cooling time for a heterogeneous house is 30.4 minutes.

Table 2.1: Population Parameter Values

Parameter	Value	Standard Deviation (Heterogeneous)
R , Thermal resistance	2 °C/kW	0.1 °C/kW
C , Thermal capacitance	10 kWh/°C	0.5 kWh/°C
P , Energy transfer rate	14 kW	-
η , Load efficiency	2.5	0.125
T_{sp0} , Initial setpoint temperature	20 °C	1 °C
T_{∞} , Ambient temperature	32 °C	-
δ , Thermostat deadband	0.5 °C	0.025 °C

It should be noted that for the homogenous populations, the ambient temperature, T_{∞} , is kept at a constant 32°C. While this is not realistic, it allows the study to focus on the effect of the input, $m(t)$. In order to properly test the feasibility of the study, the heterogeneous simulation has an ambient temperature that will vary akin to the temperature changes on a single day (e.g. Figure 2.3). To improve upon their model,

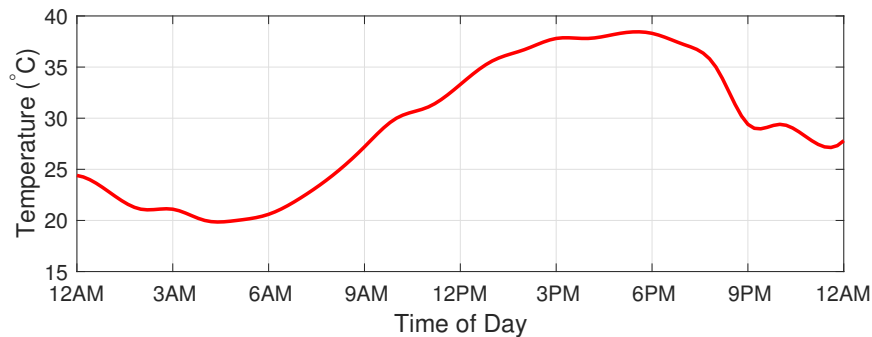


Figure 2.3: Example of hourly TMY data for Boise (July 21st) fitted with a spline.

hourly typical meteorological year (TMY) data for Boise, Idaho [24] for the week of July 21 to July 28 is used to represent a realistic summer temperature profile.

2.3 Baseline Behavior

Baseline behavior shows the demand rebound effect that this study aims to mitigate. In simulations of the baseline behavior, N number of homes are simulated using the techniques described above. The aggregate power starts at an equilibrium point by setting 42.8% of the homes to the On state, as found by Bashash and Fathy [12]. At two hours into the simulation, a DR event is introduced. The introduced DR event will forcibly turn Off all of the AC compressors and hold them in the Off state for 15 minutes.

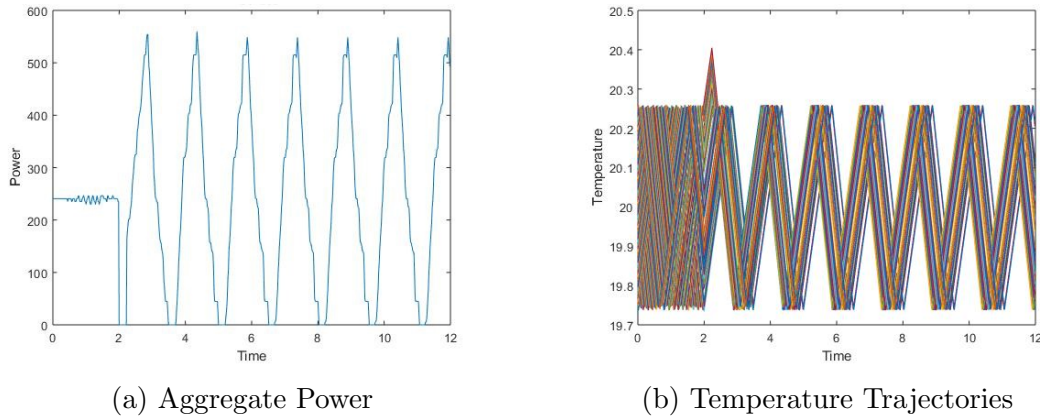
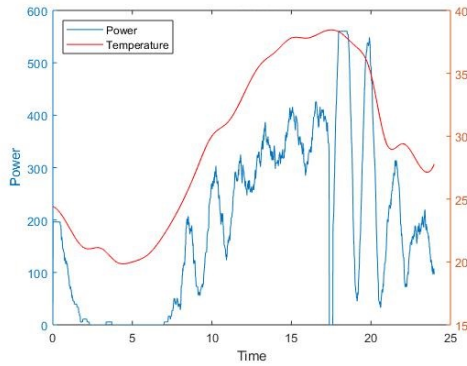


Figure 2.4: Example of demand rebound in a homogeneous population.

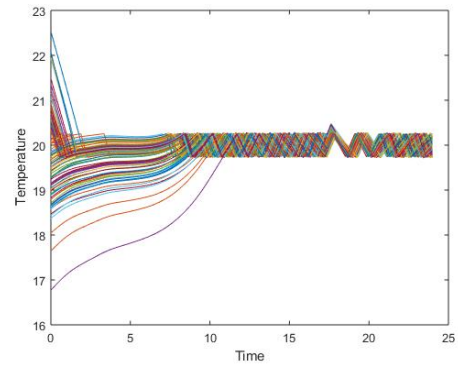
It can be seen that the DR event syncs the TCLs. This happens because during the 15 minutes all of the agents heat up moving to the upper limit of the deadband. When the DR event has ended, all of the agents wish to turn back, creating large ramps in the aggregate power. Since all of the houses have the same parameters in the homogeneous population, they all heat and cool at the same rate. This maintains the synchronized behavior after the DR event.

The baseline response for the heterogeneous population is shown in Figure 2.5. Here the DR event is applied 10 minutes before the maximum ambient temperature

is experienced. Similar effects of demand rebound can be seen in the heterogeneous population, however, the variance of the house parameters and the changing outdoor temperature have a damping effect on the aggregate power.



(a) Aggregate Power



(b) Temperature Trajectories

Figure 2.5: Example of demand rebound in a Heterogeneous population.

CHAPTER 3

STABILITY

3.1 Background

3.1.1 What is a System

Kulakowski et al. provides a definition of a system as any combination of components which act together toward a common objective [25]. Kulakowski et al. also gives a more philosophical definition of a system as an isolated part of the universe which holds interest. While we are able to describe the isolated system using mathematical equations, a system can never truly be separated from its environment. The interactions between systems and their environments can be categorized into two types of variables, independent and dependent. Independent variables are generated outside of the system and are not affected by the dynamics within the system. These variables are recognized as inputs to the system. Independent variables that which do not interact with the system are neglected, as no interest is found from them. Dependent variables are generated within the system and those which have significant interest to the observer are considered outputs.

For the sake of this study, the system is defined as a population of residential thermostatically controlled loads (TCLs). Recalling that a system is a combination of components acting toward a single goal, a population of TCLs are only able to

be described as a single system by adding a form of interaction between individual components. Where a single TCL operates as an independent system, isolated from neighboring TCLs, a population of TCLs with added communication interact with each other to form a common objective; this objective is demand response.

Independent variables for the population of TCLs would be the ambient outside air temperature, as well as any disturbances to the system, such as demand response events. The output variable would be the aggregate power used by the TCLs. The goal of demand response is to reduce demand peaks; therefore, it is essential that the output of our system is stable.

3.1.2 Stability

While techniques for evaluating stability on non-linear control systems exist, such as Lyapunov Functions (such as [26]), stability of linear control systems have been extensively studied. Studying the natural response of a linear time-varying system, Nise defines a stable system if the natural response approaches zero as time approaches infinity [27]. Similarly, a system is unstable if the natural response approaches infinity as time approaches infinity. Lastly, in regard to the natural response, a system is considered marginally stable if the natural response is constant or oscillates as time approaches infinity.

Looking at the total response of the system, or the sum of the natural response and the forced response, a stable system is defined as having a bounded output for every bounded input. If any of the outputs from a bounded input are unbounded the system is considered unstable.

Focusing on the natural response, we are able to predict the stability of a linear and continuous system if the system poles fall on the left half plane, or rather, the

poles possess negative real parts. For a system in the discrete time domain, stability is found when all poles fall within a unit circle centered at the origin, or the largest magnitude of the system pole is less than one.

3.2 Calculating Stability

3.2.1 Linearization

With the system defined, the issue of solving stability is now addressed, particularly, the impact the proportional gain, K_{gain} has on the stability of the system. As previously mentioned, there are numerous methods to solving a non-continuous control system but because the aim of this study is to analyze the entire population as a whole, as well as to avoid excess computing levels, a linearization for the discrete input m is introduced. Allowing m to equal any value within the range $[0,1]$, the input is now continuous. This is done by setting m equal to the position within the dead band, θ . Visually this can be seen in the Figure 3.1.

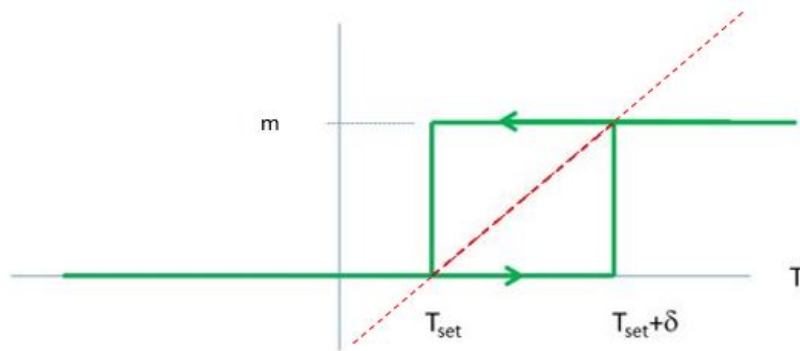


Figure 3.1: Visual representation of linearized thermostat.

The assumption of linearization by setting the compressor power equal to a continuous value can be seen in commercial HVAC units as variable compressors are introduced to the market. As previously mentioned, in the continuous time domain stability is found by analyzing the eigenvalues of the system matrix. If the eigenvalues fall on the left-hand side of the imaginary axis, or have negative real parts, the system is stable. In the discrete time domain, the eigenvalues must fall within the unit circle, or have a magnitude less than one. For each of the three styles of delays the system matrix has been solved to be able to compute the eigenvalues. The closed loop system matrix for each delay style is calculated from the open loop matrices shown in Equations 2.22 - 2.24. It is to be noted the the variable I found in the new system matrices are the identity matrix. The closed loop form for each delay scheme was found by defining the input m in terms of the system's state variables. By redefining the input, a feedback loop is created. With the input m written in terms of state variables, the system matrix can be re-written to incorporate these dynamics.

The closed loop system matrix of the concurrent delay style is a $N \times N$ matrix. The linear input $m(k)$ is defined in terms of the position within the deadband $\theta(k)$ shown in Equation 3.1.

$$m_i(k) = \theta_i(k) - K_{gain} \tilde{m}_i(k) \quad (3.1)$$

Substituting Equation 2.7 into Equation 3.1, which yields:

$$m(k) = \theta_i(k) - K_{gain} \frac{1}{d} \mathcal{A} m(k) \quad (3.2)$$

Like terms are combined and the input $m(k)$ is written as a function of $\theta(k)$.

$$m(k) = \left(I + K_{gain} \frac{1}{d} \mathcal{A} \right)^{-1} \cdot \theta(k) \quad (3.3)$$

Finally, substituting Equation 3.2 into the vector form of the system equation (Equation 2.25), we can find the closed loop system matrix to be the matrix multiplied by the vector of state variables. This is shown in Equation 3.4.

$$G_{cl,com} = G + \frac{1}{\delta} H_{II} \left(I + \frac{K_{gain}}{d} \cdot \mathcal{A} \right)^{-1} \quad (3.4)$$

The closed loop system matrix for the last m delay scheme is a $4*N \times 4*N$ matrix. Due to the introduced delays, three states are introduced. The state variables for the last m and full delay schemes are $\theta(k)$, $\theta(k-1)$, $m(k)$, and $m(k-1)$, respectively. The derivation of the system matrix for the last m delay scheme is shown below. Starting with the linear input:

$$m(k+1) = \theta(k+1) - K_{gain} \tilde{m}(k) \quad (3.5)$$

Re-write \tilde{m} from Equation 3.5 in terms of the adjacency matrix.

$$m(k+1) = \theta(k+1) - \frac{K_{gain}}{d} \mathcal{A} m(k) \quad (3.6)$$

The system equation described in Chapter 2 is shown below.

$$T(k+1) = GT(k) + H_I T_\infty + H_{II} m(k) \quad (3.7)$$

The system equation is re-written by substituting the definition of θ and shown below.

$$\delta * \theta(k+1) + T_{set} = GT(k) + H_I T_\infty + H_{II} m(k) \quad (3.8)$$

Equation 3.5 is plugged into Equation 3.8 to get $\theta(k+1)$ in terms of the state variables.

$$\theta(k+1) = G[\theta(k) + T_{set}] + \frac{H_I}{\delta}T_{\infty} + \frac{H_{II}}{\delta}m(k) - \frac{T_{set}}{\delta} \quad (3.9)$$

With $\theta(k+1)$ described in state variables, plug Equation 3.9 into Equation 3.6.

$$m(k+1) = \left[G[\theta(k) + T_{set}] + \frac{H_I}{\delta}T_{\infty} + \frac{H_{II}}{\delta}m(k) - \frac{T_{set}}{\delta} \right] - \frac{K_{gain}}{d}\mathcal{A}m(k) \quad (3.10)$$

As $\theta(k)$ and $m(k)$ are state variable, the identity matrix is used to write them in terms of themselves. Writing the state equations in matrix form we find:

$$\begin{bmatrix} \theta(k+1) \\ \theta(k) \\ m(k+1) \\ m(k) \end{bmatrix} = \begin{bmatrix} G & 0 & \left(\frac{1}{\delta}\right)H_{II} & 0 \\ I & 0 & 0 & 0 \\ G & 0 & \left(\frac{1}{\delta}\right)H_{II} - \left(\frac{K_{gain}}{d}\right)\mathcal{A} & 0 \\ 0 & 0 & I & 0 \end{bmatrix} \begin{bmatrix} \theta(k) \\ \theta(k-1) \\ m(k) \\ m(k-1) \end{bmatrix} + \begin{bmatrix} \frac{1}{\delta}H_I \\ 0 \\ \frac{1}{\delta}H_I \\ 0 \end{bmatrix} T_{\infty}(k) + \begin{bmatrix} \frac{1}{\delta}(G-I) \\ 0 \\ \frac{1}{\delta}(G-I) \\ 0 \end{bmatrix} \quad (3.11)$$

As such, the close loop system matrix is shown as follows:

$$G_{cl,lastM} = \begin{bmatrix} G & 0 & \left(\frac{1}{\delta}\right)H_{II} & 0 \\ I & 0 & 0 & 0 \\ G & 0 & \left(\frac{1}{\delta}\right)H_{II} - \left(\frac{K_{gain}}{d}\right)\mathcal{A} & 0 \\ 0 & 0 & I & 0 \end{bmatrix} \quad (3.12)$$

Note, the full delay scheme uses the same state variables as the last m delay scheme. The derivation for the system matrix of the full delay scheme is shown below. Starting with the linear input for this delay scheme:

$$m(k+1) = \theta(k) - K_{gain}\tilde{m}(k) \quad (3.13)$$

Re-write \tilde{m} from Equation 3.13 in terms of the adjacency matrix.

$$m(k+1) = \theta(k) - \frac{K_{gain}}{d}\mathcal{A}m(k) \quad (3.14)$$

Note the Equation 3.14 is already in terms of the state variables described at the beginning of this section. The equation for θ is described in Equation 3.8. Moving to matrix form, the state equation is shown in Equation 3.15.

$$\begin{bmatrix} \theta(k+1) \\ \theta(k) \\ m(k+1) \\ m(k) \end{bmatrix} = \begin{bmatrix} G & 0 & \left(\frac{1}{\delta}\right)H_{II} & 0 \\ I & 0 & 0 & 0 \\ I & 0 & -\left(\frac{k}{d}\right)\mathcal{A} & 0 \\ 0 & 0 & I & 0 \end{bmatrix} \begin{bmatrix} \theta(k) \\ \theta(k-1) \\ m(k) \\ m(k-1) \end{bmatrix} + \begin{bmatrix} \frac{1}{\delta}H_I \\ 0 \\ 0 \\ 0 \end{bmatrix} T_{\infty}(k) + \begin{bmatrix} \frac{1}{\delta}(G-I) \\ 0 \\ 0 \\ 0 \end{bmatrix} \quad (3.15)$$

As such the final close loop matrix for the full delay is shown below:

$$G_{cl,full} = \begin{bmatrix} G & 0 & \left(\frac{1}{\delta}\right)H_{II} & 0 \\ I & 0 & 0 & 0 \\ I & 0 & -\left(\frac{k}{d}\right)\mathcal{A} & 0 \\ 0 & 0 & I & 0 \end{bmatrix} \quad (3.16)$$

By incrementally increasing the gain value, K_{gain} , until the magnitude of the largest eigenvalue falls outside of the unit circle, the gain value at the stability limit is found. This test was repeated for both a square and ring lattice, as the connection architecture, as well as varying population size and delay style. a summary of these

values are shown on Fig. 3.1. All values utilize a deadband of $\delta = 0.5^\circ\text{C}$.

Table 3.1: Gain Values at Stability Limit for Varying Population Sizes

$\delta = 0.5$							
Population Size		Square Lattice			Ring Lattice		
L	N	Con	Last M	Full	Con	Last M	Full
5	25	1.2073	0.9767	0.9767	1.7485	0.9767	0.9767
10	100	0.9767	0.9767	0.9767	1.7364	0.9767	0.9767
15	225	0.9985	0.9767	0.9767	1.7365	0.9767	0.9767
20	400	0.9767	0.9767	0.9767	1.7364	0.9767	0.9767
25	625	0.9845	0.9767	0.9767	1.7364	0.9767	0.9767
30	900	0.9767	0.9767	0.9767	1.7364	0.9767	0.9767
35	1225	0.9807	0.9767	0.9767	1.7364	0.9767	0.9767
40	1600	0.9767	0.9767	0.9767	1.7364	0.9767	0.9767
45	2025	0.9791	0.9767	0.9767	1.7363	0.9767	0.9767
50	2500	0.9767	0.9767	0.9767	1.7364	0.9767	0.9767

The stability limit is effected greatly by the change for concurrent to either delays. The concurrent systems are also the only delay scheme found to be effected by change in population size. To test the effect of the deadband on the stability limit, a single population size, $N = 100$ was analyzed for both the square and ring lattice. The results from this analysis is shown in Table 3.2.

Table 3.2: Gain Values at Stability Limit for Varying Deadband Sizes

$L = 10 \ \& \ N = 100$						
δ	Square Lattice			Ring Lattice		
	Con	Last M	Full Delay	Con	Last M	Full Delay
2.0	0.994	0.994	0.994	1.768	0.994	0.994
1.5	0.992	0.992	0.992	1.764	0.992	0.992
1.0	0.988	0.988	0.988	1.757	0.988	0.988
0.5	0.977	0.977	0.977	1.736	0.977	0.977
0.4	0.971	0.971	0.971	1.726	0.971	0.971
0.3	0.961	0.961	0.961	1.709	0.961	0.961
0.2	0.942	0.942	0.942	1.674	0.942	0.942
0.1	0.883	0.883	0.884	1.570	0.883	0.884

From Table 3.2, it can be seen that the size of the deadband greatly affects the stability limit of the system.

3.2.2 Approach

In order to validate the eigenvalue analysis of the linearized systems, a simple simulation was conducted and compared. Simulations were composed using a combination of dynamics systems and agent-based modeling. Using the equations provided in Section 3.2 these simulations were created and tested within MATLAB.

It is to be noted that the linearized system has the capability to go unstable. An example of the an unstable response for the linear system is shown in Fig. 3.2, where Fig. 3.2a is the aggregate power of the population and Fig. 3.2b is the individual temperature trajectories over the course of the simulation. For this example, the simulation is comprised of 100 homes, with a deadband of $\delta = 0.5$, and a square lattice structure for communication. The simulation also utilizes the last m delay criteria and a gain value of $K_{gain} = 1.0$, eigenvalue analysis predicts the unstable response.

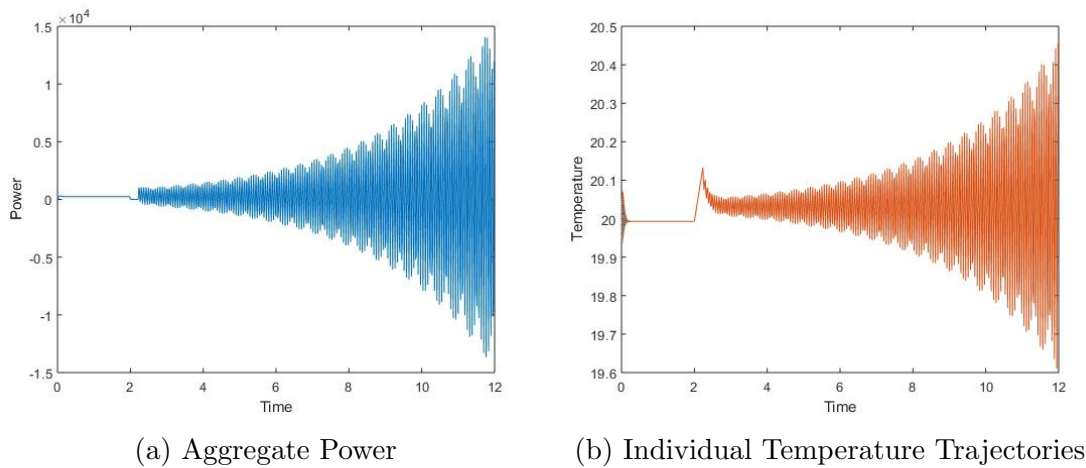


Figure 3.2: Example of an unstable response of the linear system.

Table 3.3: Eigenvalue Analysis Compared to Linear Simulation

Square & Concurrent			
δ	ζ	Lin Sim	Diff
2.0	0.994	0.994	0.000
1.5	0.992	0.992	0.000
1.0	0.988	0.988	0.000
0.5	0.977	0.976	0.001
0.4	0.971	0.970	0.001
0.3	0.961	0.961	0.000
0.2	0.942	0.941	0.001
0.1	0.883	0.883	0.000

Square & Last M			
δ	ζ	Lin Sim	Diff
2.0	0.994	0.994	0.000
1.5	0.992	0.992	0.000
1.0	0.988	0.988	0.000
0.5	0.977	0.977	0.000
0.4	0.971	0.971	0.000
0.3	0.961	0.962	-0.001
0.2	0.942	0.943	-0.001
0.1	0.883	0.890	-0.007

Square & Full Delay			
δ	ζ	Lin Sim	Diff
2.0	0.994	0.994	0.000
1.5	0.992	0.992	0.000
1.0	0.988	0.988	0.000
0.5	0.977	0.976	0.001
0.4	0.971	0.970	0.001
0.3	0.961	0.960	0.001
0.2	0.942	0.939	0.003
0.1	0.884	0.875	0.008

Ring & Concurrent			
δ	ζ	Lin Sim	Diff
2.0	1.768	0.994	0.774
1.5	1.764	0.992	0.772
1.0	1.757	0.988	0.769
0.5	1.736	0.976	0.760
0.4	1.726	0.970	0.756
0.3	1.709	0.961	0.748
0.2	1.674	0.941	0.733
0.1	1.570	0.883	0.687

Ring & Last M			
δ	ζ	Lin Sim	Diff
2.0	0.994	0.994	0.000
1.5	0.992	0.992	0.000
1.0	0.988	0.988	0.000
0.5	0.977	0.977	0.000
0.4	0.971	0.971	0.000
0.3	0.961	0.961	0.000
0.2	0.942	0.943	-0.001
0.1	0.883	0.890	-0.007

Ring & Full Delay			
δ	ζ	Lin Sim	Diff
2.0	0.994	0.994	0.000
1.5	0.992	0.992	0.000
1.0	0.988	0.988	0.000
0.5	0.977	0.977	0.000
0.4	0.971	0.970	0.001
0.3	0.961	0.960	0.000
0.2	0.942	0.939	0.003
0.1	0.884	0.875	0.008

3.3 Validation of the Linearized System

In order to validate the linearization and eigenvalue analysis, analysis of the non-linear system was completed. Because the non-linear system is a population of limit

switching agents defining the characteristics of instabilities raises issues. While the linear system offers a decent representation of the behavior of variable speed units, the stability predictions must be validated through nonlinear simulations. It is also important to note that the linear system was able to go unstable for certain gain values. For the example of an unstable response for the linear system, a simulation using the same model parameters is completed on the non-linear system. The non-linear response is shown in Fig 3.3.

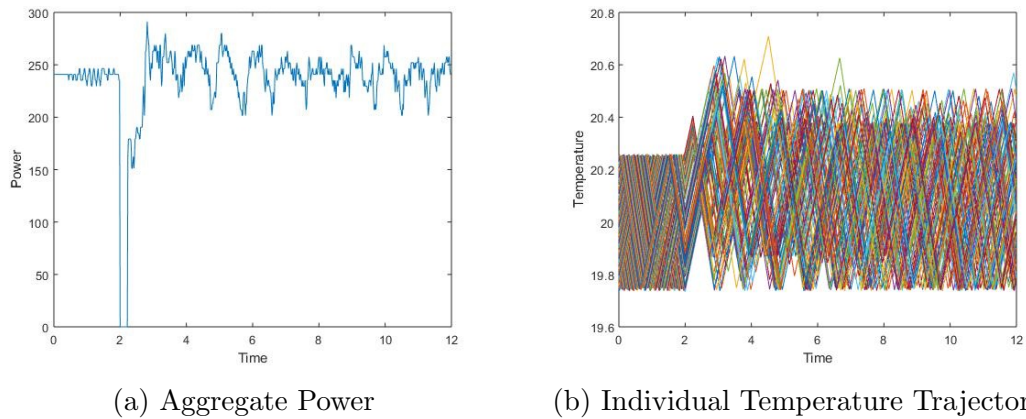


Figure 3.3: Predicted unstable response of the non-linear system.

While our linearized analysis predicts that the system would be unstable at $k=0.8$, the simulation shows no clear sign of instability. Indices of performance were chosen to further inspect the effect of the proportional gain, K_{gain} , on the non-linear system. Maximum aggregate power, number of compressor cycles following the DR event, and standard deviation of the individual temperature trajectories were chosen as these indices of performance (Equations 3.17-3.19). These indices were collected from populations with varying gain values and deadband sizes.

The max power, or peak demand, is the maximum aggregate power of the population of TCLs over the course of the simulation.

$$\text{Max Power} = \max(P_{TCL}) \quad (3.17)$$

Number of cycles is the average number of cycles a house experiences in the population, after the DR event is imposed on the system. This is the cost at which the system operates. As the criteria is stronger individual TCLs will experience larger number of cycles and while it is not included in the simulation, larger number of cycles are correlated to a less efficient operation and shortened lifetime of AC units.

$$\text{Number of Cycles} = \frac{1}{N} \sum_{i=1}^N \text{Cycles}(DREvent^+) \quad (3.18)$$

Standard deviation of the individual temperature trajectories was taken as the average of the standard deviation of the nominal position within the deadband for the population of TCLs. 'Stability' of the aggregate power is related to the distribution of the houses throughout the deadband. Grouping, or a small standard deviation is related to unwanted behavior seen in 1.2.

$$\text{Std Dev} = \frac{1}{N_{steps}} \sum_{k=1}^{N_{steps}} \sqrt{\frac{1}{N-1} \sum_{i=1}^N |\theta_i(k) - \mu|^2} \quad (3.19)$$

Where μ is defined as mean of the populations position within the deadband at the time. shown below:

$$\mu = \frac{1}{N} \sum_{i=1}^N \theta_i(k) \quad (3.20)$$

Maximum aggregate power and average number of cycles are shown graphically in Fig. 3.4. There are several notable characteristics to be seen in the maximum power and number of cycles. After approximately a gain of $K_{gain} = 1.0$, the system is no

$L = 10 \& N = 100$			
Square, Last M, & $\delta = 0.5$			
k	Power	Cycle	StdDev
0.10	504	8.82	0.279
0.20	453.6	8.73	0.293
0.30	375.2	8.61	0.302
0.40	341.6	8.52	0.309
0.50	302.4	8.37	0.319
0.60	296.8	8.19	0.325
0.70	285.6	8.18	0.334
0.80	296.8	8.15	0.322
0.90	302.4	7.99	0.347
1.00	291.2	7.97	0.351
1.10	280	7.86	0.351
1.20	280	7.85	0.361
1.30	280	7.72	0.376
1.40	280	7.73	0.372
1.50	280	7.91	0.344
1.60	280	7.97	0.361
1.70	280	7.91	0.371
1.70	280	7.91	0.371
1.80	280	7.75	0.373
1.90	280	7.96	0.364
2.00	280	8.68	0.375
2.10	280	8.92	0.396
2.20	324.8	10.21	0.414

(a)

$L = 10 \& N = 100$			
Square, Last M, & $\delta = 0.5$			
k	Power	Cycle	StdDev
2.30	296.8	9.99	0.399
2.40	296.8	9.76	0.403
2.50	296.8	9.79	0.405
2.60	291.2	9.29	0.426
2.70	291.2	9.29	0.429
2.80	291.2	9.3	0.439
2.90	291.2	9.3	0.442
3.00	291.2	9.3	0.454
3.10	291.2	9.15	0.463
3.20	291.2	9.09	0.466
3.30	291.2	9.05	0.467
3.40	291.2	8.93	0.471
3.50	291.2	8.88	0.474
3.60	291.2	8.85	0.470
3.70	291.2	8.83	0.503
3.80	319.2	10.45	0.480
3.90	336	16.72	0.548
4.00	375.2	20.59	0.513
4.10	420	31.1	0.459
4.20	464.8	58.72	0.426
4.30	560	117.23	0.293
4.40	436.8	39.63	0.458
4.50	560	132.64	0.236

(b)

Table 3.4: Indices of Performance for Varying Gain Values

longer positively effected by the rising gain. Shortly after the gain of $K_{gain} = 2.0$ an increase in both maximum power and number of cycles is experienced. Lastly, as the gain values approached four, the system witnesses a large increase in both maximum power and number of cycles.

The anomalous behavior observed through the indices of performance was not predicted by the eigenvalue analysis and there was no clear signs of the system

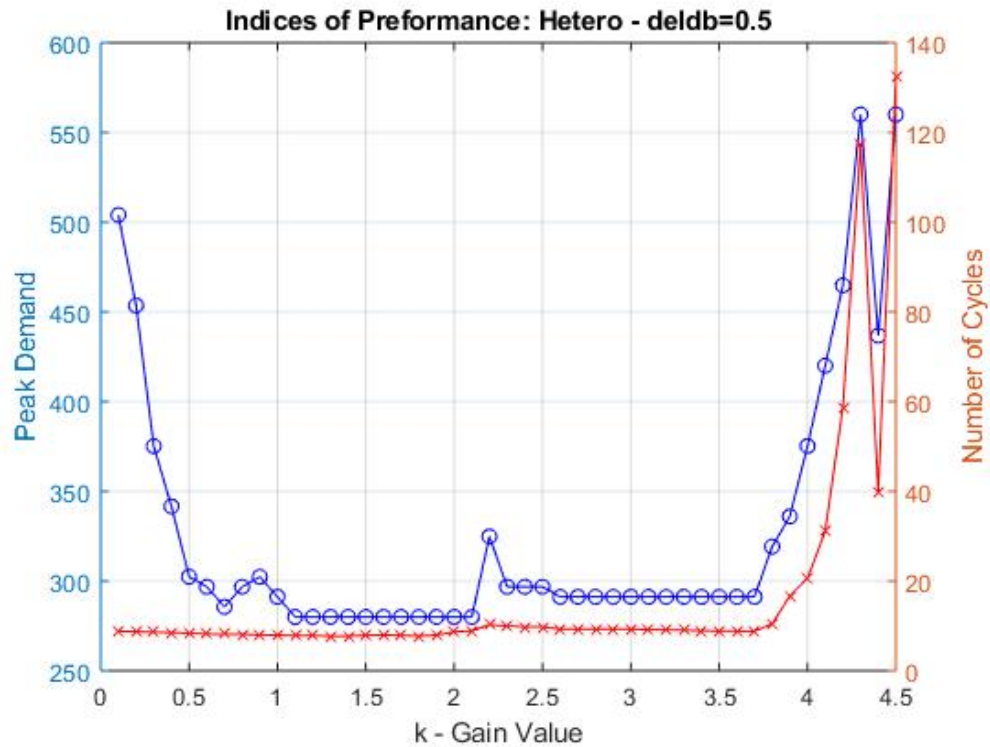


Figure 3.4: Indices of performance for varying gain values.

exhibiting an unstable response. It is to be noted that the metric of standard deviation did not accurately represent the system, as two groupings of houses became apparent. An example of this behavior is shown in the Fig. 3.5. The two groupings of temperature trajectories caused a higher standard deviation, usually related to a better distribution of temperatures across the population. More than one grouping was not anticipated prior to the testing.

It was mentioned that as the gain value approaches four, a drastic spike in maximum power and number of cycles is witnessed. Temperature trajectories for a system of 100 houses with a deadband of $\delta = 0.5$ and $K_{gain} = 4.0$ is shown in Fig. 3.6. After the DR event occurs, it can be seen the some of the TCLs experience rapid cycling. While commercial HVAC units utilize control to avoid rapid cycling, this

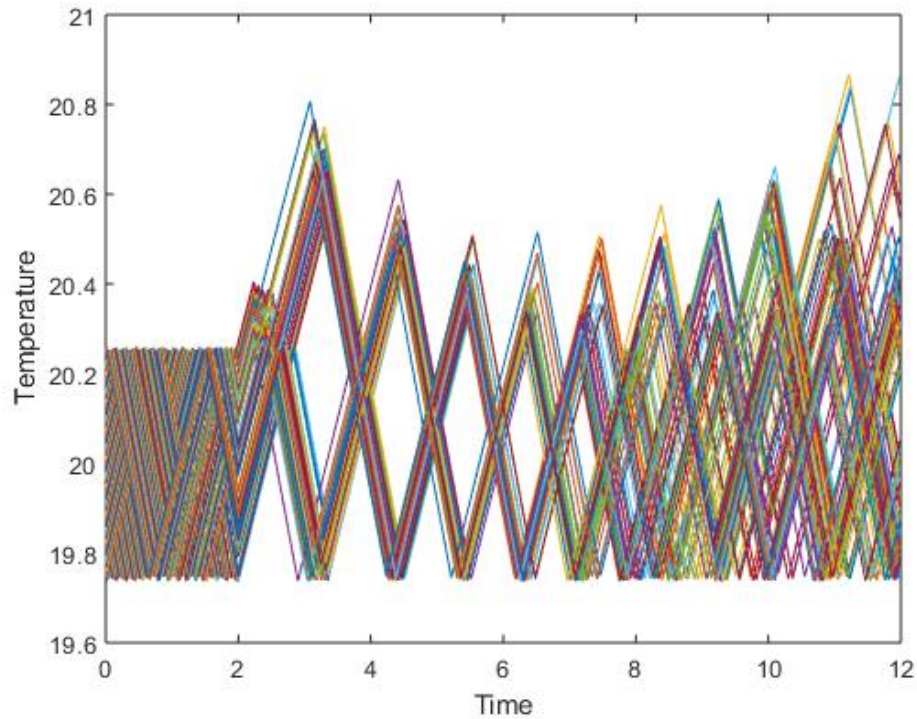


Figure 3.5: Grouping of temperature trajectories.

feature has been omitted from the simulation. The delay introduced by the lock out feature would have been difficult to incorporate into our linearized model. However, the model does represent the worst case scenario.

3.3.1 Gain Value effect on Dead Band

To get further insight to the responses of the non-linear system, the effect of varying gain values on the upper and lower bounds of the deadband were observed. To begin the analysis, first a deeper look into the \tilde{m} values must be completed. Due to the discrete nature of the operating state, m , and the constant number of connections, the average value of neighboring thermostats is constricted to five discrete values. With a constant degree of four, the possible \tilde{m} values are shown in Eq. 3.21.

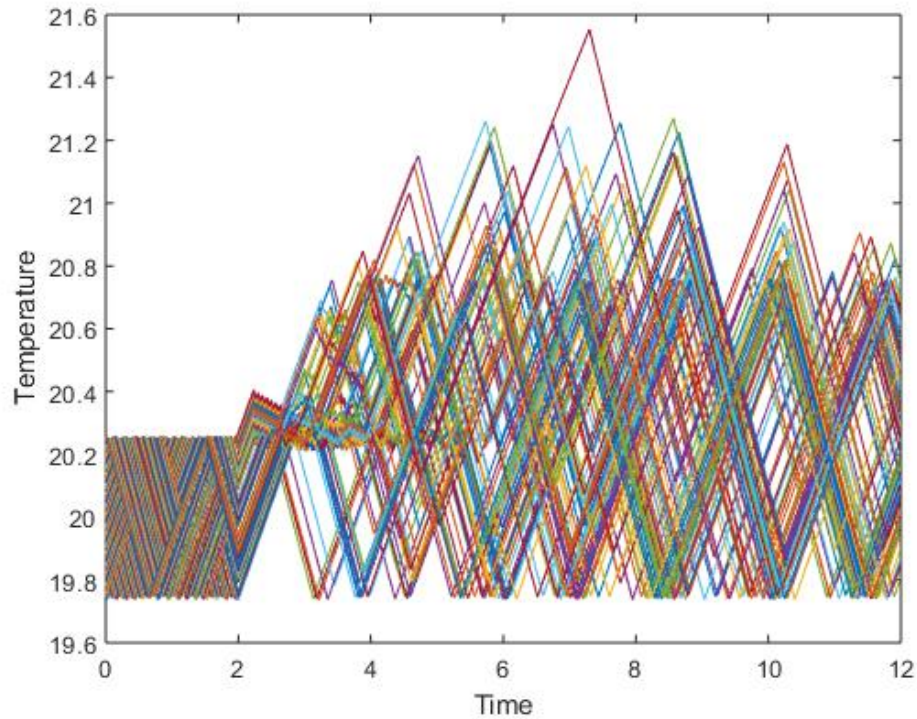


Figure 3.6: Rapid cycling of TCLs.

$$\tilde{m}_i = \begin{cases} 0, & \text{if 0 of the connected houses are on.} \\ 0.25, & \text{if 1 connected house is on.} \\ 0.5, & \text{if 2 connected houses are on.} \\ 0.75, & \text{if 3 connected houses are on.} \\ 1.0, & \text{if 4 connected houses are on.} \end{cases} \quad (3.21)$$

In combination to the gain value, \tilde{m} changes the switching criteria of the thermostats effectively changing the deadband to five different possible values. Each of these values vary as the gain varies. The effective deadbands are plotted for varying gain values.

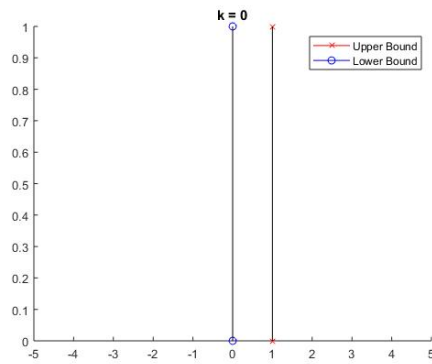
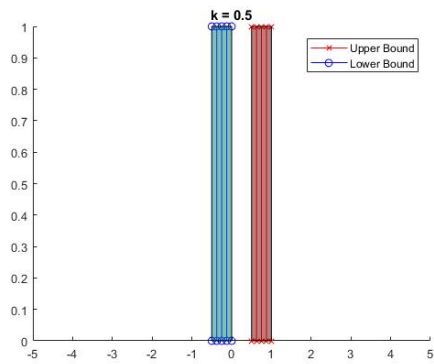
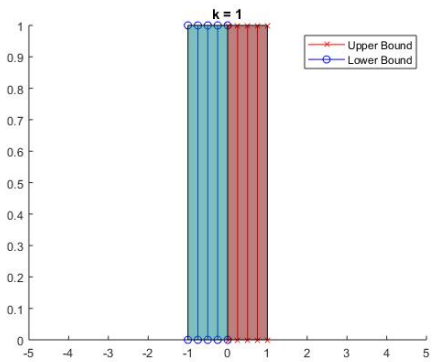
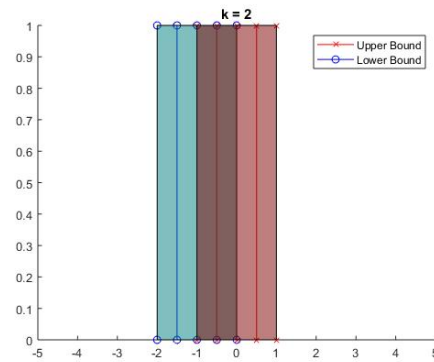
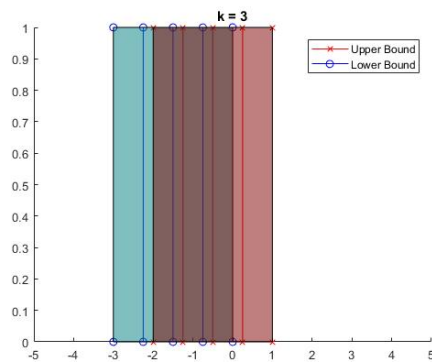
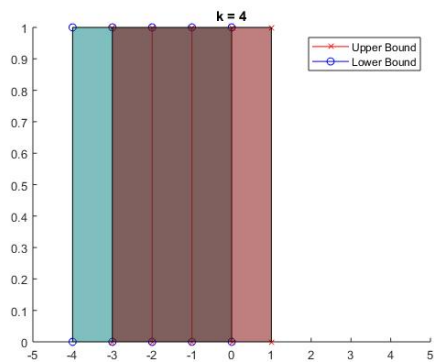
(a) $K_{gain} = 0.0$ (b) $K_{gain} = 0.5$ (c) $K_{gain} = 1.0$ (d) $K_{gain} = 2.0$ (e) $K_{gain} = 3.0$ (f) $K_{gain} = 4.0$

Figure 3.7: Effective deadbands with varying gain values.

At a gain value of zero, the deadbands positions are the same as that of a simple thermostat. As the gain value increases, the effective deadbands move to decreasing

values. At a gain values of $K_{gain} = 1.0$, a possible upper bound overlaps with a possible lower bound. At a gain value of $K_{gain} = 2.0$ a similar trend occurs as two of the deadband bounds overlap and again at $K_{gain} = 4.0$ where three deadband bounds overlap. It is important to note that the \tilde{m} value for each home can change with each time step of the simulation.

As these possible upper and lower bounds overlap, it increases the change of rapid cycling. If a individual home experiences upper and lower bounds that continuously places it's thermostat outside of the allowable deadband, a TCL will switch between operating states at undesirable speeds. An example of this behavior is shown in Figure 3.8.

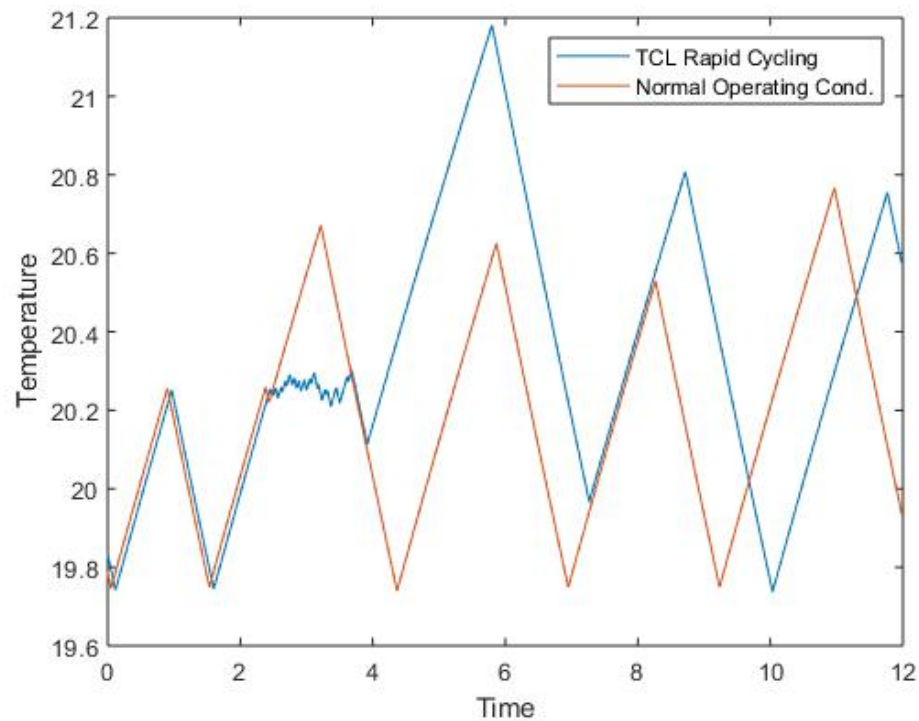


Figure 3.8: Example of rapid cycling due to deadband changes.

In Figure 3.8, two temperature trajectories are shown from a simulation utilizing a

gain of $k = 4.0$, 100 homes, and deadband size of $\delta = 0.5^\circ C$. One of the temperature trajectories witnesses rapid cycling to the changing deadbands, switching between operating states at every time step at its worst point. The rapid cycling occurs for nearly a two hour duration, between hours two and four of the simulation, (shown in Figure 3.9). Contrasting from the TCL rapid cycling, another TCL from the same simulation is plotted showing expected operating conditions. Here, the deadband changes to allow the TCL to desynchronize from its neighbors but does not cause rapid cycling.

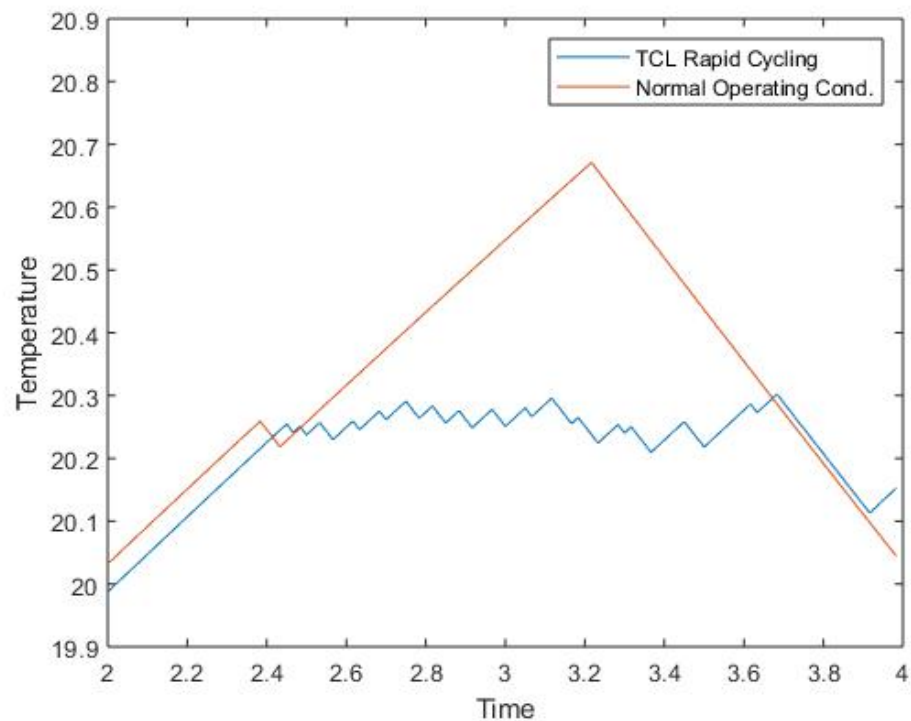


Figure 3.9: Rapid cycling between hours 2 and 4 of simulation.

To better visualize this problem, the operating state, m , of the rapid cycling TCL has been plotted in Figure 3.10. Figure 3.10a shows the operating state and the temperature trajectory of the rapid cycling TCL. During this two hour period, the AC

unit turns on 17 times for an average 2.5 minutes. The shortest time the compressor is On during the two hour period is one minute and the longest the compressor is on is 14 minutes (near the end of the time period).

This rapid cycling occurs because of the added thermostat modification. This effect arises because of the combination of the overlapping deadbands and the way the average operating state, \tilde{m} , is calculated. Because of the modified thermostat behavior, house temperatures can move from within the thermostat deadband to outside the allowable range just based off of the operation of neighboring TCLs. This phenomenon can occur at every time step, (as seen in Figure 3.10a), as the average operating state of neighboring units is calculated. Figure 3.10b shows the effect of \tilde{m} on the operating state of a single TCL experiencing rapid cycling.

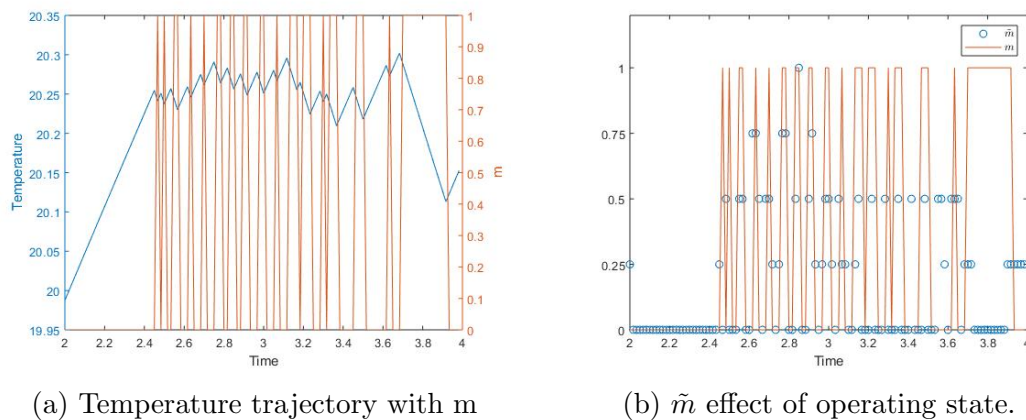


Figure 3.10: Operating state of rapid cycling TCL.

Figure 3.10b suggests that during this two hour period, when \tilde{m} is a non-zero number the AC compressor experiences a change of state. The majority of the time within the two hour period of rapid cycling, the \tilde{m} value is equal to zero. This is seen in Figure 3.11.

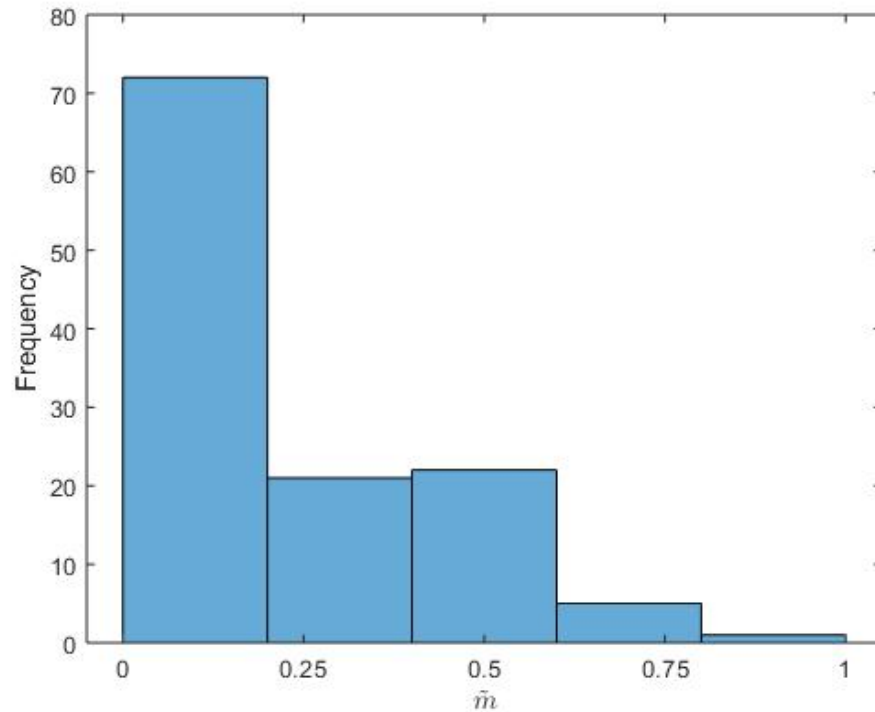


Figure 3.11: Histogram of \tilde{m} values during two hour period.

3.4 Heterogeneous Simulations

In an attempt to further validate the implementation of the proposed criteria, a simulation and analysis was completed for the heterogeneous population with a varying outdoor temperature.

Aggregate power is plotted against ambient outdoor temperature. Fig. 3.12 is of a 100 home simulation which utilizes a deabband of $\delta = 0.5$ and a gain values of $K_{gain} = 0.8$. To observe the effect of the gain value on the heterogeneous population, the indices of performance are shown below.

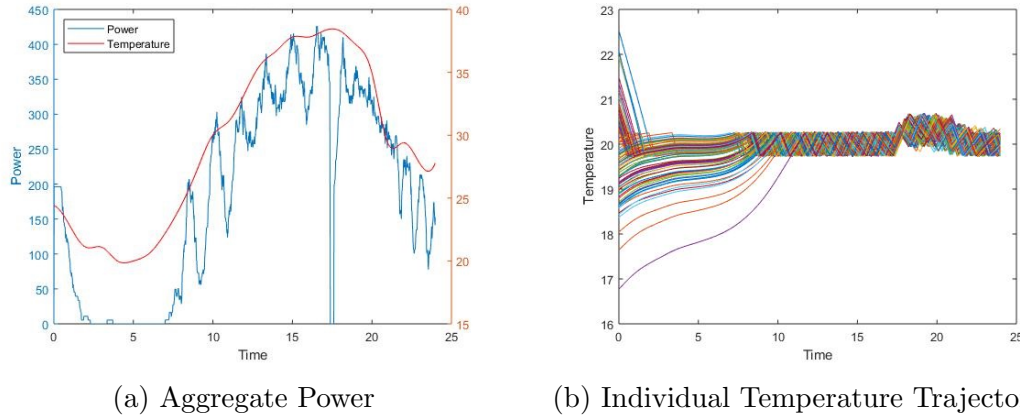


Figure 3.12: Example response of heterogeneous population.

3.5 Stability Analysis Conclusions

Through the linearization assumption, the closed loop system equation was able to be computed from each of the three delay styles introduced in the study. An eigenvalue study was completed from the closed loop systems to predict stability based off of varying gain values. The gain values at the stability limit approached a value of $K_{gain} = 1.0$ as the deadband increased. Agent based models were used to validate the predictions of the eigenvalue study.

While the assumption used to linearize the heat transfer equation was a large one, it did give insight to the dynamics of the population of TCLs. It was found that as the deadband size increases, the linearization assumption becomes more valid. For a TCL with an infinitely large deadband, the first order heat transfer dynamics will act akin to that of the linear first order system. Instability was not seen in the non-linear systems with a gain value above the predicted stability limit and because of this, further analysis of the effect of the gain value on the effective deadband was completed. Due to the discrete nature of \tilde{m} five possible values for each the upper and lower bounds of the deadband are created. In combination with the gain value,

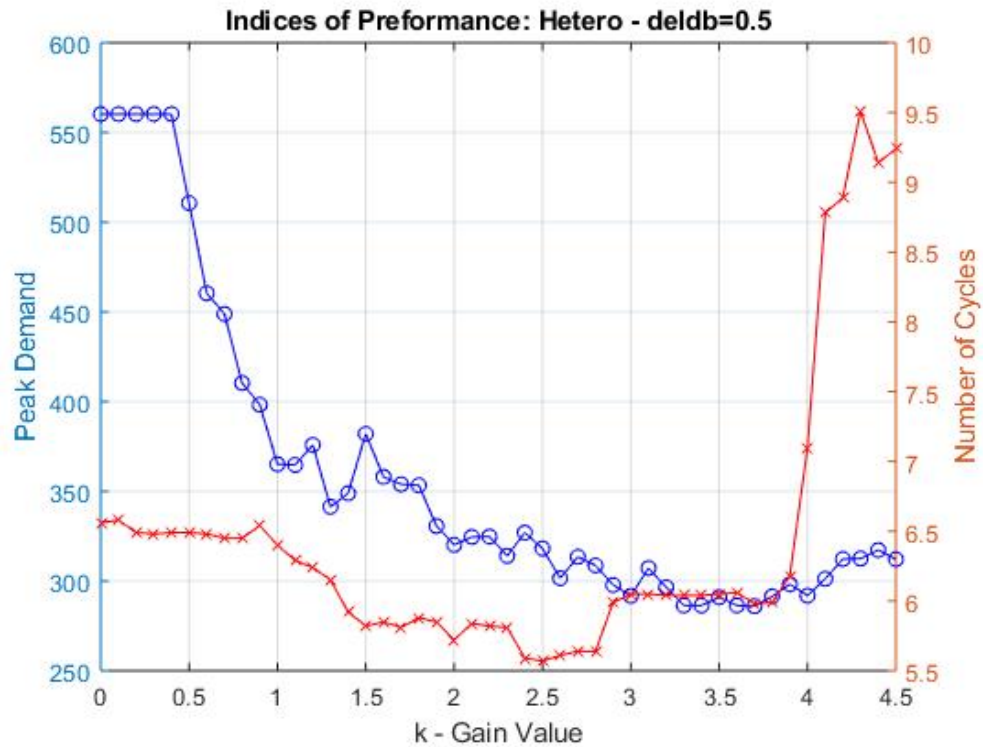


Figure 3.13: Indices of performance for heterogeneous populations.

\tilde{m} , moves these bounds to an new position, creating an effective deadband. At a gain value of $K_{gain} = 1.0$, the position of a lower bound and upper bound coincide, possibly cause the thermostat to rapid cycle. This behavior could have lead to the predicted instability.

At a gain value of $K_{gain} = 2.0$ and $K_{gain} = 4.0$, the deadband bounds coincide again. The overlap of the upper and lower bounds cause an increase in number of cycles, as well as the spike in maximum power. The periodic behavior witnessed through the indices of performance are caused by these overlaps in the upper and lower bounds.

3.6 Future Work

In order to fully understand the effects of the gain value on the system, a refined definition of instability for a population of TCLs needs to be created. Through the indices of performance, the instability prediction from the linear analysis was validated by the deadband analysis. The deadband analysis showed the possibility for rapid cycling, but other than the decreased effectiveness of the the gain on the aggregate power near the predicted stability limit, there were no clear indicators for what instability was for the nonlinear system.

Furthering this point, only a single stability limit was found using the eigenvalue analysis on the linear system. Why the anomalous behavior does not appear in the linear analysis and how to incorporate it into definitions of stability should be addressed.

Further analysis into how each connection architecture effects the unwanted behaviors. It was seen using the square lattice that un-predicted grouping began to occur at varying gain values. This was not consistent for smaller or larger gain values. It is important to understand the cause of these groupings and how to avoid it in the future.

CHAPTER 4

RESILIENT SYSTEM DESIGN

4.1 Security And Resilience

With more and more demand response being added to the power grid in efforts to mitigate over generation issues, issues of security and resilience begin to emerge. Security of cyber physical systems, such as the electric grid, have become a growing topic of interest. Need for security can be seen from examples such as the Ukraine Power Grid Attack [28]. In order to design not only a secure system for demand response, but a secure and resilient system the definitions of both must be understood in terms of cyber physical systems.

Security, as defined by the US Department of Homeland Security, is the act of adding physical means or cyber defense measures to reduce the risk to a critical infrastructures[29]. These defenses protect from intrusions, attacks, or the effects of both natural and manmade disasters. Examples provided for security measures include requiring badge entry at doors and using antivirus software. Through this definition and examples, it can be seen that security of cyber-physical systems works to keep unauthorized users, both intentional and unintentional, from accessing critical information and data from the system. If security is used to keep unauthorized users out of a system, what can be done to reduce the effects of intruders who have successfully infiltrated the system?

Resilience offers systems the ability to endure and recover from attacks, intentional and unintentional, as well as naturally occurring threats[29]. This is done through the effort to prepare and adjust to changing conditions and possessing the ability to recover from system disturbances. An example of a resilience measure is installing a generator to provide back up powering in the case of a power outage[30]. A deeper definition of resilience in terms of cyber-physical systems is provided by Rieger et al through the definition of resilient control systems [31]. Rieger et al defines resilient control systems as a system that maintains state awareness and an accepted level of operational normalcy in response to disturbances, including threats of an unexpected and malicious nature.

Rothrock explains the problem of relying solely on security is that security will not stop all attacks[30]. Because attacks are not always from outside adversaries and the security system cannot prepare for every possible attack, the system cannot be 'locked down' to the point where all attacks are defended against. Resilience is needed for cyber physical systems in order to ensure operation amidst having the systems vulnerabilities exploited.

Vulnerabilities are security flaws and can be found in any system. Any vulnerability can be exploited by an adversary to become an entry point to the system. A common entry point in cyber physical systems is communication between devices and added communication is an integral component to demand response systems. While vulnerabilities of a system are inherently created through the design of the system, albeit generally created unintentionally, the ability to overcome system vulnerabilities can also be incorporated at early stages of the design stage.

Resilience itself is an emergent property of a system [32]. Therefore the resilience of a system can not only be analyzed by studying the interactions between individual

parts of the system, but also be altered by changing those interactions. This means that the resilience of a demand response system can be improved through alterations of the communication used.

Analyzing resilience of our system in early design stages is crucial to the ability of creating a resilient system. The use of complex network theory, aligned with graph spectral theory gives the ability to calculate resilience of complex engineered systems [33]. By using these theories to calculate the resilience of a complex system, Mehrpouyan et al show how design structure can effect the resilience of these systems. This work emphasizes the ability to use graph theory to show the interconnection of nodes within a system. Elements of these theories, such as the adjacency matrix, the degree matrix, and the Laplacian matrix, are used to define metrics of resilience.

The Laplacian matrix is defined as:

$$\mathcal{L} = \mathcal{D} - \mathcal{A} \quad (4.1)$$

Where \mathcal{D} is the degree matrix, or a diagonal $N \times N$ matrix representing the number of connections each node has. The algebraic connectivity, or second smallest eigenvalue of the laplacian matrix, quantifies the average difficulty to isolate a node from the system. A lower algebraic connectivity is related to a higher level of modularity within the system. Mehrpouyan explains the importance of modularity in a system but is clear to highlight that modularity does not increase the systems tolerance to failure but increases the reliability of the system because of the ability to fix and maintain the system's modules individually.

Open Automated Demand Response (OpenADR) is a excellent example of incorporating security and resilience in early design for demand response systems[34].

OpenADR was introduced by the Lawrence Berkeley National Lab in an attempt to standardize the communication protocols for demand response systems. OpenADR 2.0 was developed by the OpenADR Alliance and addressed improved needs through certification processes[35]. Herberg et al's OpenADR 2.0 shows different architectures for the protocol and exclaims the security and resilience benefits compared to the previous version.

OpenADR targets a specific subcategory of demand responses, real time pricing demand response, or the ability of the utility to calculate varying demand rates and provide information to the end user in real time. While OpenADR shows the inclusion of resilience in the design stages of a complex system, it focuses solely on the communication of the system, and does not take into account physical implications. The resilience was also determined by the communication tools used to create the varying architectures and Herberg et al never explicitly defines measures of resilience for this system [35].

4.2 Simulated Attack

The security of the designed system will be tested by analyzing the effects of failures due to cyber-physical attacks through two parts. The first part being an analysis of the algebraic connectivity of varying populations. The second part of the study will be validating the analysis of the algebraic connectivity by simulating a cyber-physical attack in the agent based model. By comparatively analyzing these two parts, improvements in security and resilience of the proposed model will be shown.

The simulated attack will show the effects of a loss of communication between

neighboring thermostats. This simulated attack could possibly occur from communication jamming or a denial of service attack targeting a communication module. This attack assumes the first stage of Industrial Control System (ICS) Kill-Chain has previously been completed and focuses on the second stage. ICS Kill-Chain is defined by Sans in 2015 [28]. By compromising the peer-to-peer communication, it is effectively setting the \tilde{m} value to zero for any attacked thermostat. As mentioned earlier, if the \tilde{m} value is set to zero, the thermostat effectively reverts back to the simple thermostat model, which could lead to a dangerous rebound after a DR event.

For the simulation, \tilde{m} is calculated through the use of the adjacency matrix of the system. The loss of communication of a thermostat from the population will be modeled by setting the row and column of the corresponding thermostat equal to zero. This act of reverting to the simple thermostat model will be recognized as a failure of the thermostat in the demand response system. Different levels of failure, or differing percent of failed nodes out of the total population, are to be tested. Using a percent of zero failure, or 100% operation, as a baseline, the different levels of failure are tested to the point that demand response is not longer effectively reducing the peak demand of the aggregate system. Baseline simulations will be created at each population, including varying connection architectures.

To compare to the baseline, varying populations are simulated with fixed percentages of total number of houses attacked. The indices of performance used in the previous section are used to represent the systems throughout the simulated attacks. It is to be noted that this test case uses varying numbers of homes. Each simulation starts at an equilibrium point with 42.8% of the house's in an 'On' state and the remaining houses 'Off'. The demand response event occurs at two hours into the simulation and has a duration of 15 minutes.

4.3 Results

By solving for the eigenvalues of the Laplacian matrix, the algebraic connectivity of the graph can be found [33]. The algebraic connectivity, or second smallest eigenvalue of the Laplacian matrix, was found for varying populations and varying graph construction. These results can be seen in Figure 4.1.

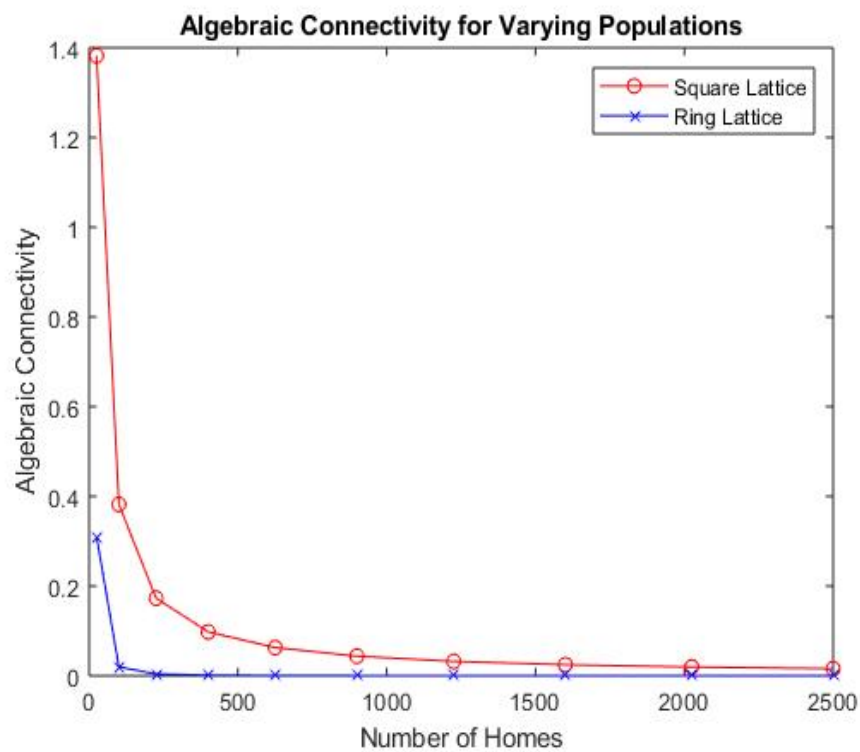


Figure 4.1: Algebraic Connectivity for Varying Populations of Homes.

To test the effect of the resilience of a graph to the desired demand response ability of the population, the simulated nodal attack was completed for varying populations and graph structures. For each simulation percent increase of maximum power is displayed for varying populations, as they experience an attack on 20% of total population. These values are shown on Table 4.1.

Table 4.1: Percent Increase for Varying Population with 20% Failure

$\delta = 0.2$ & Con			$\delta = 0.2$ & Last M			$\delta = 0.2$ & Full		
N	Square	Ring	N	Square	Ring	N	Square	Ring
25	10.53%	16.67%	25	10.53%	10.53%	25	10.53%	5.26%
100	12.16%	13.89%	100	12.16%	3.66%	100	11.69%	4.94%
225	11.59%	6.98%	225	9.04%	4.49%	225	10.84%	4.44%
400	10.67%	5.56%	400	7.33%	4.13%	400	9.43%	2.80%
625	13.10%	6.08%	625	5.72%	3.46%	625	33.68%	2.57%
900	11.89%	4.58%	900	3.71%	3.09%	900	3.96%	1.90%
1225	9.57%	3.76%	1225	1.75%	2.88%	1225	1.11%	1.29%
1600	12.01%	3.42%	1600	3.10%	2.83%	1600	3.19%	0.98%
2025	11.42%	3.32%	2025	4.25%	2.86%	2025	4.35%	0.77%
2500	10.96%	3.19%	2500	4.46%	2.87%	2500	4.42%	0.76%
Avg.	11.39%	6.74%	Avg.	6.20%	4.08%	Avg.	9.32%	2.57%

$\delta = 0.5$ & Con			$\delta = 0.5$ & Last M			$\delta = 0.5$ & Full		
N	Square	Ring	N	Square	Ring	N	Square	Ring
25	14.29%	6.67%	25	14.29%	6.67%	25	21.43%	13.33%
100	15.09%	10.71%	100	13.21%	20.75%	100	12.96%	16.36%
225	12.71%	6.67%	225	12.71%	15.52%	225	11.97%	16.52%
400	14.15%	12.94%	400	13.59%	15.64%	400	13.88%	20.10%
625	17.57%	17.16%	625	15.24%	19.12%	625	16.09%	19.06%
900	13.26%	18.32%	900	14.47%	20.61%	900	15.86%	20.10%
1225	15.74%	14.31%	1225	16.94%	15.43%	1225	16.56%	18.74%
1600	18.09%	20.27%	1600	18.013%	22.06%	1600	20.38%	20.94%
2025	17.01%	21.46%	2025	18.05%	22.12%	2025	16.54%	21.43%
2500	21.76%	20.81%	2500	21.83%	21.45%	2500	2.16%	20.97%
Avg.	15.97%	14.93%	Avg.	15.84%	17.94%	Avg.	14.78%	18.77%

Each population uses a baseline for maximum power from a simulation with zero houses attacked. It should be noted that for each of the population sizes, the baseline varies between connection architecture. For all population with a deadband equal to $\delta = 0.2$ it can be seen that the percent increase for a ring lattice is less than that of a square lattice. As the deadband increases, the simulated response becomes greater removed from the predictions using algebraic connectivity, or simply, the square lattice

experiences smaller percent increases of maximum power. The only population with a deadband equal to $\delta = 0.5$ where the ring lattice shows greater robustness is the population utilizing the a concurrent delay scheme. The deadband effects on percent increase for the last m delay scheme are showed graphically in Figure 4.2.

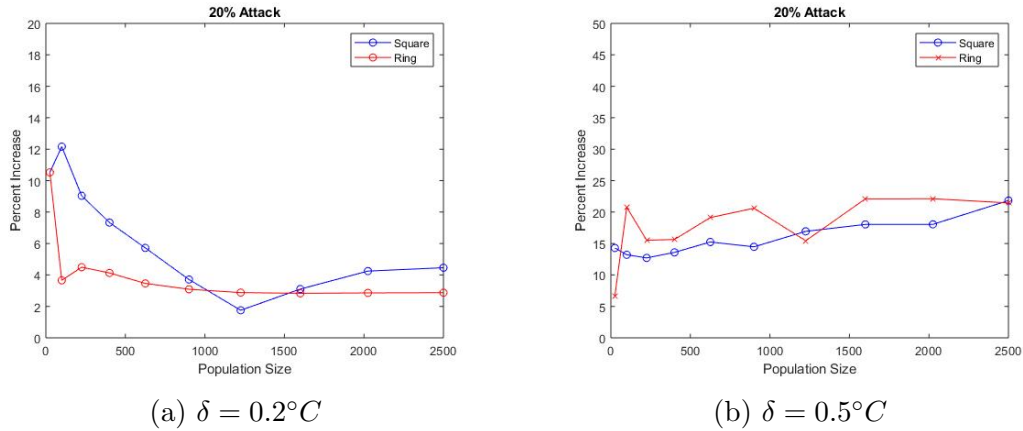


Figure 4.2: Deadband effects on percent increase.

Through the testing of the simulation it was observed that the location of the houses which were attack effected the aggregate demand of the system. This was verified by varying the location of attacked homes on a constant population of 100 homes, connected with a square lattice. Locations chosen for the attacks were the first N number of attacked homes of the population, the last N number of attacked homes in the population, and alternately attacking the first $2*N$ number of attacked homes in the population. An example of the alternating attack would be if two homes in a population were attacked home number one and home number three would lose ability to communicate with neighboring homes. A summary of the effect of varying attack location is shown through percent increase of maximum power. This test for robustness was completed on two deadband sizes, $\delta = 0.2$ and 0.5 , focusing on the last M delay scheme. Percent of attacked homes is also varied, using 10%, 20%, and

30% An example of this work is shown in Table 4.3, for a deadband size of $\delta = 0.5$, square lattice structure, and a percent attacked of 20%.

Table 4.3: Number of Homes till Failure with Varying Location

Population	Front	Back	Alternating
25	14.3%	7.1%	14.3%
100	13.2%	13.2%	22.6%
225	12.7%	11.9%	26.3%
400	13.6%	11.2%	18.9%
625	15.2%	14.0%	31.1%
900	14.5%	14.5%	18.4%
1225	16.9%	13.5%	32.9%
1600	18.0%	17.2%	22.1%
2025	18.0%	16.7%	37.6%
2500	21.8%	19.9%	24.9%
Average	15.8%	13.9%	24.9%

A summary of all the populations tested for the effects of varying location are shown in Table 4.3. The values shown are the average percent increase over varying population sizes. An example of this average is shown on the final row of Table 4.3.

Table 4.4, gives insight into location effects seen in simulated cyber attacks on the proposed DR system. For attacks on systems with deadbands of $\delta = 0.2$, a cyber attack located at the front of the population is more effected by the attack than when the attack is located at the back of the population. For larger percentages of attacked houses, the attack alternating between every other house in front of the population is the most critical. A clear distinction between attacks on the front of the population versus those at the back of a population were not seen for systems with a dead band of $\delta = 0.5$. However, it was clear that simulations following the alternating attack pattern caused the largest percent increase in maximum aggregate power.

Table 4.4: Summary of Location Effect from Cyber Attacks

δ	% Attack	Front		Back		Alternating	
		Square	Ring	Square	Ring	Square	Ring
0.2	10%	5.4%	4.1%	0.9%	0.0%	4.2%	2.9%
	20%	6.2%	4.1%	0.9%	0.0%	12.1%	7.4%
	30%	9.2%	4.6%	1.4%	0.0%	21.0%	17.5%
0.5	10%	6.7%	8.6%	8.5%	8.9%	11.5%	10.6%
	20%	15.8%	17.9%	13.9%	16.7%	24.9%	21.1%
	30%	26.3%	26.8%	24.8%	27.1%	38.9%	26.4%

The locational effects shown in Table 4.4 are plotted for two different deadband sizes. These plots can be seen in Figure 4.3.

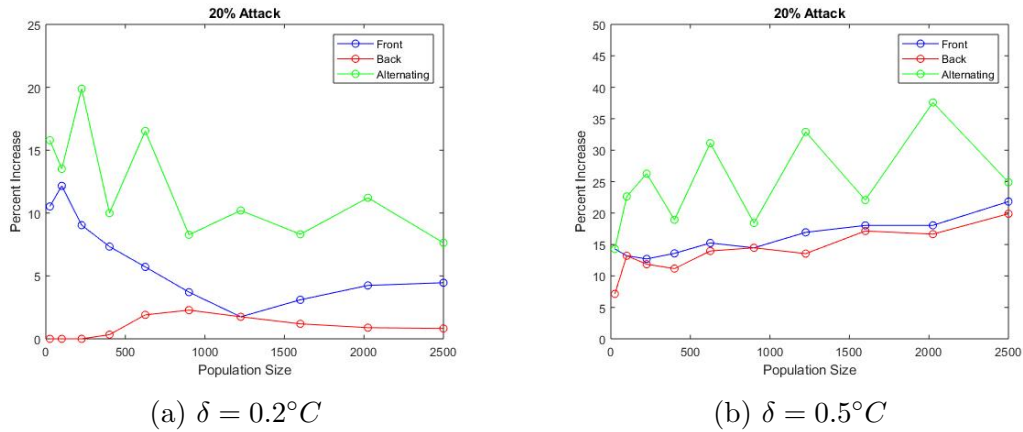


Figure 4.3: Locational effects on percent increase.

4.4 Resilience Conclusions

The results displayed in Figure 4.1 clearly shows that both population size and graph structure effect the resilience of a population. Algebraic connectivity was calculated after the simulated attack, presenting a value of zero for each system. An algebraic connectivity of zeros shows that the graph is not connected. This coincides with simulating an attack to remove the ability to communicate between houses.

Algebraic connectivity for a centralized controller was found to be a value of one. This calculated value for the centralized controller was larger than all values, except 25 homes connected with a square lattice, validating the decision to move toward a de-centralized controller.

The ring lattice, on average, showed both better algebraic connectivity, as well as a better ability, as a whole, to operate with attacked homes. For larger populations and larger percentages of attacked homes the ring lattices saw a smaller percent increase in maximum power. It should be noted that percent increase and maximum are not the same because baseline results were calculated for each population size with square and ring lattices.

The simulated attack created failures in communication by altering the adjacency matrix of the graph, setting row and column of the corresponding houses to zero. For the simulation the attacked houses were the first corresponding houses in the simulation, (i.e. house 1,2,3..N - Number of Attacked Homes). Slight differences were found when the attacked houses were positioned at the end of the population. When further studied, the location of the attack showed larger differences on the total number of houses to failure. The most vulnerable of the test attacks was attacked at the beginning of the population but alternating every other house.

One reason the attack alternating every other house was the most effective is because it effects more houses in the population. By attacking the houses in a close group, only a few of the houses neighboring the edge houses are effected. However, distributing the attack effects every house connected to the attacked houses.

4.5 Future Work

Showing the comparison between lattice structures shows the dependence of the demand response to the graph as well as shows potential reasoning on using one above the other. Only two simple lattice structures were used for the study, further work testing other lattice structures is needed to find the optimal graph for the proposed criteria. It should be noted the study on the effects of location of attacks was completed only with the square lattice, further testing should be done on all additional lattice structures.

For the study the arbitrary gain value in the additional criteria was kept at a constant of $k=0.8$. This was chosen because it fall clearly below the stability limit in a parallel study of the stability of the system. Testing the effect of the gain value on the resilience of the system show a great area of interest.

Another interesting point maybe to look further into the effects of population size. For the larger populations with a ring lattice, it appears that the value of failed homes may be asymptotically approaching a value, but it is unclear at this point.

Finally, the study has looked at the ability of the system to function while under an cyber-physical attack. In moving forward in the advancement of the purposed criteria, the ability of the system to self heal is crucial.

CHAPTER 5

CONCLUSIONS

Through both the eigenvalue analysis of the linear system and the simulations of the nonlinear system it was shown that choice of gain, connection degree, and connection architecture effected the performance of the population in demand response events. It is important to note that undesirable effects can be introduced to the population by the proposed mesh thermostat behavior. With higher values of the proportional gain K_{gain} it was shown to cause unstable response in the linear system, spikes in peak demand for the nonlinear system, and groupings of temperature trajectories on populations utilizing a square lattice.

While the eigenvalue analysis gave insight to the nonlinear system, the behavior of the nonlinear system was much more complex than what we were able to capture with the linear system. However it is important to note that for systems with large deadband values the system acted closer to the linear assumption.

By simulating cyber-physical attacks on the system, we were able to see that the system resilience, as measured by the algebraic connectivity, is sensitive to the method of attack and the type of connection architecture implemented. Contrary to the eigenvalue analysis of stability, the algebraic connectivity was more aligned to the simulations when a smaller deadband was utilized.

One of the limitations to this study is the first order equation used to describe

the heat transfer dynamics. Zhang et al showed the importance for the second order equation in this type of analysis [15]. The second order equation takes into account the internal mass of the TCLs. Moving this work to the second order equation would give a more realistic simulation for the demand response events. Also incorporating a lock-out stage on the AC compressors will help narrow the gap of what is being simulated to what is implemented.

To validate the simulated model, a hardware in the loop (HIL) simulation should be created. The HIL simulation would give insight to the complexity of a real embedded system utilizing the mesh behavior. A HIL simulation may be comprised of several built model homes which have the same time constant as those being modeled in the simulation. The built models will be embedded within the population of simulated models. Model parameters and the order of the model will be validated with the HIL simulation.

Model predictive control is increasingly pursued with demand response events. The ability to incorporate solar penetration into the simple swarm behavior and test it's ability for tracking against model predictive control is desired for the future steps of this project.

REFERENCES

- [1] “Renewables 2017.”
- [2] P. Denholm, M. O’Connel, G. Brinkman, and J. Jorgenson, “Overgeneration from Solar Energy in California: A Field Guide to the Duck Chart,” Tech. Rep. NREL/TP-6A20-65023, National Renewable Energy Laboratory, Nov. 2015.
- [3] J. S. John, “Hawaii’s Solar-Grid Landscape and the Nessie Curve,” Feb. 2014.
- [4] C. Vlahoplus, G. Litra, P. Quinlan, and C. Becker, “Revisiting the Duck Curve,” Oct. 2016.
- [5] L. Vandepaer, J. Cloutier, and B. Amor, “Environmental impacts of Lithium Metal Polymer and Lithium-ion stationary batteries,” *Renewable and Sustainable Energy Reviews*, vol. 78, pp. 46–60, Oct. 2017.
- [6] D. S. Callaway, “Tapping the energy storage potential in electric loads to deliver load following and regulation, with application to wind energy,” *Energy Conversion and Management*, vol. 50, pp. 1389–1400, May 2009.
- [7] J. Rodriguez-Molina, M. Martinez-Nez, J.-F. Martinez, and W. Prez-Aguiar, “Business Models in the Smart Grid: Challenges, Opportunities and Proposals for Prosumer Profitability,” *Energies*, vol. 7, pp. 6142–6171, Sept. 2014.
- [8] K. Zhou, S. Yang, and Z. Shao, “Energy Internet: The business perspective,” *Applied Energy*, vol. 178, pp. 212–222, Sept. 2016.
- [9] R. Malhame and C.-Y. Chong, “Electric load model synthesis by diffusion approximation of a high-order hybrid-state stochastic system,” *IEEE Transactions on Automatic Control*, vol. 30, pp. 854–860, Sept. 1985.
- [10] Z. Xu, J. Ostergaard, M. Togeby, and C. Marcus-Moller, “Design and Modelling of Thermostatically Controlled Loads as Frequency Controlled Reserve,” in *2007 IEEE Power Engineering Society General Meeting*, pp. 1–6, June 2007.
- [11] “A/C Cool Credit.”

- [12] S. Bashash and H. K. Fathy, "Modeling and Control of Aggregate Air Conditioning Loads for Robust Renewable Power Management," *IEEE Transactions on Control Systems Technology*, vol. 21, pp. 1318–1327, July 2013.
- [13] S. Bashash and H. K. Fathy, "Modeling and control insights into demand-side energy management through setpoint control of thermostatic loads," in *Proceedings of the 2011 American Control Conference*, pp. 4546–4553, June 2011.
- [14] L. B. Wood and H. Harry Asada, "Cellular Stochastic Control of the Collective Output of a Class of Distributed Hysteretic Systems," *Journal of Dynamic Systems, Measurement, and Control*, vol. 133, no. 6, p. 061011, 2011.
- [15] W. Zhang, J. Lian, C. Y. Chang, and K. Kalsi, "Aggregated Modeling and Control of Air Conditioning Loads for Demand Response," *IEEE Transactions on Power Systems*, vol. 28, pp. 4655–4664, Nov. 2013.
- [16] H. Hao, B. M. Sanandaji, K. Poolla, and T. L. Vincent, "Aggregate Flexibility of Thermostatically Controlled Loads," *IEEE Transactions on Power Systems*, vol. 30, pp. 189–198, Jan. 2015.
- [17] S. D. Ramchurn, P. Vytelingum, A. Rogers, and N. Jennings, "Agent-Based Control for Decentralised Demand Side Management in the Smart Grid," *International Foundation for Autonomous Agents and Multiagent Systems*, p. 8, 2011.
- [18] E. Xydias, C. Marmaras, and L. M. Cipcigan, "A multi-agent based scheduling algorithm for adaptive electric vehicles charging," *Applied Energy*, vol. 177, pp. 354–365, Sept. 2016.
- [19] R. Schwartz, "Local Peer-to-Peer Communication to Improve Demand Response in Residential Neighborhoods," *Boise State University Theses and Dissertations*, Dec. 2018.
- [20] "Agent-Based Simulation Modeling."
- [21] A. Al-Fuqaha, M. Guizani, M. Mohammadi, M. Aledhari, and M. Ayyash, "Internet of Things: A Survey on Enabling Technologies, Protocols, and Applications," *IEEE Communications Surveys Tutorials*, vol. 17, no. 4, pp. 2347–2376, 2015.
- [22] A.-L. Barabási and M. Pósfai, *Network Science*. Cambridge University Press, July 2016. Google-Books-ID: iLtGDQAAQBAJ.
- [23] K. Ogata, *Discrete-Time Control Systems*. Pearson, 2 ed., Dec. 1994.
- [24] "NSRDB update - TMY3: Alphabetical List by State and City."

- [25] B. Kulakowski, J. Gardner, and J. L. Shearer, *Dynamic Modeling and Control of Engineering Systems*. Cambridge, 2012.
- [26] “Multiple Lyapunov functions and other analysis tools for switched and hybrid systems - IEEE Journals & Magazine.”
- [27] N. Nise, *Control System Engineering*. Wiley, 7 ed., 2015.
- [28] R. M. Lee, M. j. Assante, and T. Conway, “Analysis of the Cyber Attack on the Ukrainian Power Grid,” *EISAC*, Mar. 2016.
- [29] “What Is Security and Resilience?,” Dec. 2012.
- [30] R. Rothrock, “Security vs. resilience: Know the difference.”
- [31] C. Rieger, Q. Zhu, and T. Basar, “Agent-based cyber control strategy design for resilient control systems: Concepts, architecture and methodologies,” in *2012 5th International Symposium on Resilient Control Systems*, (Salt Lake City, UT, USA), pp. 40–47, IEEE, Aug. 2012.
- [32] Y. Y. Haimes, K. Crowther, and B. M. Horowitz, “Homeland security preparedness: Balancing protection with resilience in emergent systems,” *Systems Engineering*, vol. 11, pp. 287–308, Dec. 2008.
- [33] H. Mehrpouyan, B. Haley, A. Dong, I. Y. Tumer, and C. Hoyle, “Resilient Design of Complex Engineered Systems,” *ASME 2013 International Design Engineering Technical Conferences and Computers and Information in Engineering Conference*, no. 55881, p. V03AT03A048, 2013.
- [34] C. McParland, “OpenADR open source toolkit: Developing open source software for the Smart Grid,” in *2011 IEEE Power and Energy Society General Meeting*, (Detroit, MI, USA), pp. 1–7, IEEE, July 2011.
- [35] U. Herberg, D. Mashima, J. G. Jetcheva, and S. Mirzazad-Barijough, “OpenADR 2.0 deployment architectures: Options and implications,” in *2014 IEEE International Conference on Smart Grid Communications (SmartGridComm)*, pp. 782–787, Nov. 2014.

APPENDIX A

STABILITY METRICS

A.1 Indices of Performance for Varying Populations

(a) Indices of Performance, Square, Last M, $\delta = 2.0$ (b) Indices of Performance, Square, Last M, $\delta = 0.5$

L=10 & N=100			
Square, Last M, & deldb=2.0			
K - Gain	Max Pout	Num Cyc.	StdDev
0.100	425.6	2.42	0.2645
0.200	392	2.33	0.2742
0.300	352.8	2.27	0.2819
0.400	330.4	2.22	0.2948
0.500	319.2	2.24	0.3024
0.600	308	2.24	0.3178
0.700	296.8	2.16	0.32
0.800	296.8	2.24	0.3284
0.900	296.8	2.12	0.3251
1.000	291.2	2.09	0.3299
1.100	285.6	2.09	0.3469
1.200	285.6	2.13	0.349
1.300	280	2.08	0.3498
1.400	280	2.07	0.3595
1.500	280	2.07	0.3698
1.600	280	2	0.3793
1.700	280	1.99	0.3819
1.800	280	1.96	0.3836
1.900	280	1.96	0.3839
2.000	280	1.96	0.3852
2.100	280	1.99	0.3864
2.200	280	1.99	0.3871
2.300	280	2.06	0.3886
2.400	280	2.09	0.3898
2.500	280	2.07	0.3914
2.600	280	2.06	0.3933
2.700	280	2.01	0.3957
2.800	280	1.99	0.3992
2.900	280	1.93	0.4033
3.000	280	1.89	0.4068
3.100	280	1.86	0.4111
3.200	280	1.81	0.4145
3.300	280	1.81	0.4176
3.400	280	1.78	0.4214
3.500	280	1.78	0.4247
3.600	280	1.78	0.4281
3.700	280	1.78	0.43
3.800	274.4	1.78	0.4311
3.900	268.8	1.78	0.4323
4.000	263.2	4.01	0.4328
4.100	296.8	17.14	0.4217
4.200	336	23.22	0.4087
4.300	341.6	26.66	0.4012
4.400	341.6	27.12	0.4019
4.500	336	29.22	0.3988

L=10 & N=100			
Square, Last M, & deldb=0.5			
K - Gain	Max Pout	Num Cyc.	StdDev
0.100	504	8.82	0.2796
0.200	453.6	8.73	0.2936
0.300	375.2	8.61	0.3026
0.400	341.6	8.52	0.3093
0.500	302.4	8.37	0.3198
0.600	296.8	8.19	0.3254
0.700	285.6	8.18	0.3347
0.800	296.8	8.15	0.3222
0.900	302.4	7.99	0.3478
1.000	291.2	7.97	0.3511
1.100	280	7.86	0.3626
1.200	280	7.85	0.3612
1.300	280	7.72	0.3766
1.400	280	7.73	0.3723
1.500	280	7.91	0.344
1.600	280	7.97	0.3619
1.700	280	7.91	0.3716
1.800	280	7.75	0.3733
1.900	280	7.96	0.364
2.000	280	8.68	0.3756
2.100	280	8.92	0.3966
2.200	324.8	10.21	0.4143
2.300	296.8	9.99	0.3995
2.400	296.8	9.76	0.4034
2.500	296.8	9.79	0.405
2.600	291.2	9.29	0.4269
2.700	291.2	9.29	0.4299
2.800	291.2	9.3	0.439
2.900	291.2	9.3	0.4421
3.000	291.2	9.3	0.4549
3.100	291.2	9.15	0.4638
3.200	291.2	9.09	0.466
3.300	291.2	9.05	0.4675
3.400	291.2	8.93	0.4713
3.500	291.2	8.88	0.4743
3.600	291.2	8.85	0.4709
3.700	291.2	8.83	0.5039
3.800	319.2	10.45	0.4809
3.900	336	16.72	0.5482
4.000	375.2	20.59	0.513
4.100	420	31.1	0.4595
4.200	464.8	58.72	0.4261
4.300	560	117.23	0.2936
4.400	436.8	39.63	0.4584
4.500	560	132.64	0.2365

(a) Indices of Performance, Square, Last M, $\delta = 0.4$ (b) Indices of Performance, Square, Last M, $\delta = 0.3$

L=10 & N=100			
Square, Last M, & delldb=0.4			
K - Gain	Max Pout	Num Cyc.	StdDev
0.100	515.2	10.84	0.2852
0.200	487.2	10.83	0.2954
0.300	414.4	10.49	0.3028
0.400	358.4	10.59	0.2979
0.500	308	10.65	0.2995
0.600	296.8	10.26	0.3104
0.700	296.8	10.2	0.3348
0.800	296.8	10.34	0.3219
0.900	296.8	9.78	0.3512
1.000	302.4	9.77	0.3349
1.100	291.2	9.78	0.3436
1.200	285.6	9.77	0.3449
1.300	296.8	9.96	0.3506
1.400	291.2	9.49	0.386
1.500	280	9.32	0.366
1.600	280	9.51	0.3797
1.700	285.6	9.17	0.4201
1.800	291.2	9.25	0.4208
1.900	285.6	9.3	0.3886
2.000	280	9.72	0.3874
2.100	280	9.83	0.4018
2.200	296.8	10.2	0.4047
2.300	319.2	10.45	0.4019
2.400	319.2	10.31	0.4146
2.500	313.6	10.14	0.4212
2.600	308	10.16	0.4183
2.700	308	10.13	0.413
2.800	308	10.07	0.4187
2.900	308	10.05	0.4227
3.000	308	10.04	0.4189
3.100	308	10.02	0.4336
3.200	308	10.02	0.4335
3.300	308	10.03	0.4299
3.400	308	10.03	0.4318
3.500	308	9.98	0.4422
3.600	308	10.16	0.4476
3.700	285.6	9.79	0.5075
3.800	324.8	11.5	0.5167
3.900	364	20.79	0.4636
4.000	492.8	44.03	0.4464
4.100	526.4	59.8	0.4437
4.200	560	127.9	0.2036
4.300	560	113.21	0.3013
4.400	560	133.58	0.2205
4.500	560	93.45	0.3416

L=10 & N=100			
Square, Last M, & delldb=0.3			
K - Gain	Max Pout	Num Cyc.	StdDev
0.100	548.8	14.23	0.2924
0.200	515.2	14.1	0.3039
0.300	459.2	13.88	0.314
0.400	414.4	13.51	0.3132
0.500	369.6	13.51	0.3035
0.600	347.2	13.55	0.3151
0.700	308	13.39	0.32
0.800	308	13	0.3259
0.900	296.8	12.99	0.3347
1.000	291.2	12.75	0.3405
1.100	291.2	12.51	0.3427
1.200	291.2	12.49	0.3436
1.300	291.2	12.49	0.3591
1.400	280	12.17	0.3714
1.500	285.6	12.3	0.3721
1.600	280	12.33	0.3866
1.700	291.2	12.12	0.3938
1.800	285.6	12.33	0.391
1.900	280	12.38	0.3974
2.000	291.2	12.13	0.4071
2.100	280	12	0.4083
2.200	280	11.9	0.4607
2.300	280	12.33	0.4219
2.400	280	12.21	0.4353
2.500	285.6	12	0.4648
2.600	285.6	12.51	0.45
2.700	285.6	12.46	0.4616
2.800	285.6	12.16	0.4663
2.900	285.6	12.12	0.4673
3.000	285.6	12.09	0.4671
3.100	285.6	11.91	0.4871
3.200	285.6	11.92	0.4911
3.300	285.6	11.92	0.5054
3.400	285.6	11.93	0.4861
3.500	308	11.33	0.4854
3.600	313.6	12.11	0.4869
3.700	336	13.84	0.4868
3.800	414.4	19.97	0.485
3.900	560	81.02	0.3079
4.000	481.6	37.27	0.4495
4.100	560	138.08	0.2033
4.200	560	109.5	0.2808
4.300	560	145.55	0.1084
4.400	560	145.64	0.1117
4.500	560	147.08	0.099

(a) Indices of Performance, Square, Last M, $\delta = 0.2$ (b) Indices of Performance, Square, Last M, $\delta = 0.1$

L=10 & N=100			
Square, Last M, & delldb=0.2			
K - Gain	Max Pout	Num Cyc.	StdDev
0.100	560	20.75	0.299
0.200	554.4	20.24	0.3035
0.300	526.4	19.94	0.3224
0.400	498.4	19.68	0.3153
0.500	459.2	19.33	0.3381
0.600	459.2	19.07	0.3456
0.700	448	18.87	0.3532
0.800	414.4	18.68	0.344
0.900	392	18.26	0.3532
1.000	392	18.24	0.3572
1.100	392	18.17	0.3536
1.200	392	17.64	0.3975
1.300	392	17.88	0.3626
1.400	392	17.45	0.4139
1.500	392	17.64	0.3763
1.600	392	17.45	0.4014
1.700	392	17.59	0.3919
1.800	431.2	17.94	0.4147
1.900	487.2	21.22	0.4333
2.000	537.6	26.66	0.397
2.100	537.6	24.86	0.4541
2.200	515.2	28.38	0.3914
2.300	509.6	26.91	0.3975
2.400	509.6	25.99	0.4225
2.500	509.6	25.51	0.4386
2.600	509.6	25.41	0.4289
2.700	509.6	23.7	0.4325
2.800	509.6	23.82	0.4642
2.900	509.6	24.73	0.4305
3.000	509.6	24.94	0.4349
3.100	509.6	23	0.4659
3.200	509.6	23	0.4665
3.300	509.6	25.42	0.4728
3.400	509.6	25.23	0.4826
3.500	509.6	24	0.4861
3.600	515.2	35.01	0.4556
3.700	509.6	33.45	0.4689
3.800	554.4	71.81	0.3888
3.900	560	81.6	0.3414
4.000	560	150.59	0.0978
4.100	560	151.12	0.0951
4.200	560	151.13	0.0949
4.300	560	151.13	0.0949
4.400	560	151.13	0.0949
4.500	560	151.13	0.0949

L=10 & N=100			
Square, Last M, & delldb=0.1			
K - Gain	Max Pout	Num Cyc.	StdDev
0.100	560	36.62	0.3458
0.200	560	36.39	0.3486
0.300	560	35.8	0.3578
0.400	560	33.8	0.3799
0.500	560	33.28	0.3887
0.600	560	32.44	0.4028
0.700	560	31.71	0.4126
0.800	560	30.21	0.4088
0.900	560	30.48	0.4317
1.000	560	30.16	0.4421
1.100	560	29.96	0.4492
1.200	560	29.39	0.4408
1.300	560	29.1	0.4577
1.400	560	29.03	0.4679
1.500	560	29.18	0.4825
1.600	560	29.22	0.4911
1.700	560	28.97	0.4798
1.800	560	29.82	0.5087
1.900	560	34.74	0.4654
2.000	560	31.99	0.5052
2.100	560	40.09	0.4919
2.200	560	32.55	0.5033
2.300	560	50.66	0.4611
2.400	560	50.15	0.4683
2.500	560	45.92	0.4827
2.600	560	46.84	0.4928
2.700	560	43.27	0.5067
2.800	560	45.31	0.513
2.900	560	62.13	0.4678
3.000	560	59.84	0.4762
3.100	560	64.78	0.4979
3.200	560	56.03	0.4791
3.300	560	52.61	0.4914
3.400	560	59.28	0.4767
3.500	560	55.84	0.4967
3.600	560	51.15	0.5003
3.700	560	76.79	0.4903
3.800	560	85.87	0.4267
3.900	560	56.01	0.4985
4.000	560	57.14	0.5191
4.100	560	58.66	0.5304
4.200	560	60.29	0.5023
4.300	560	67.99	0.4827
4.400	560	95.32	0.4743
4.500	560	94.4	0.4825

(a) Indices of Performance, Ring, Last M, $\delta = 2.0$ (b) Indices of Performance, Ring, Last M, $\delta = 0.5$

L=10 & N=100			
Ring, Last M, & deldb=2.0			
K - Gain	Max Pout	Num Cyc.	StdDev
0.100	431.2	2.44	0.2628
0.200	386.4	2.37	0.2774
0.300	364	2.31	0.2882
0.400	341.6	2.28	0.2914
0.500	330.4	2.25	0.2964
0.600	336	2.21	0.3002
0.700	319.2	2.16	0.3028
0.800	330.4	2.17	0.3077
0.900	324.8	2.12	0.3146
1.000	324.8	2.1	0.3204
1.100	302.4	2.11	0.3227
1.200	285.6	2.12	0.3281
1.300	280	2.1	0.3341
1.400	308	2.14	0.3547
1.500	285.6	2.12	0.3633
1.600	274.4	2.1	0.3691
1.700	257.6	2.06	0.3646
1.800	240.8	1.98	0.3675
1.900	240.8	1.94	0.3776
2.000	240.8	2.9	0.3747
2.100	240.8	2.9	0.3717
2.200	240.8	2.91	0.3808
2.300	240.8	2.89	0.3855
2.400	240.8	2.9	0.3893
2.500	240.8	2.88	0.3911
2.600	240.8	2.8	0.4109
2.700	240.8	2.85	0.4165
2.800	240.8	2.73	0.4162
2.900	240.87	2.7	0.4149
3.000	240.8	2.74	0.4171
3.100	240.8	2.72	0.4115
3.200	240.8	2.68	0.4228
3.300	240.8	2.71	0.4163
3.400	240.8	2.71	0.4231
3.500	240.8	2.71	0.4246
3.600	240.8	2.71	0.4268
3.700	240.8	2.74	0.4288
3.800	240.8	2.76	0.4307
3.900	240.8	2.76	0.4332
4.000	302.4	75.58	0.2767
4.100	291.2	81.03	0.2824
4.200	375.2	82.52	0.2716
4.300	397.6	83.42	0.2715
4.400	397.6	83.61	0.2712
4.500	403.2	83.81	0.2718

L=10 & N=100			
Ring, Last M, & deldb=0.5			
K - Gain	Max Pout	Num Cyc.	StdDev
0.100	509.6	8.84	0.2796
0.200	436.8	8.74	0.2923
0.300	375.2	8.75	0.2982
0.400	330.4	8.73	0.3018
0.500	291.2	8.57	0.3084
0.600	308	8.57	0.3111
0.700	296.8	8.57	0.3149
0.800	296.8	8.54	0.3192
0.900	291.2	8.42	0.3252
1.000	285.6	8.55	0.3266
1.100	296.8	8.37	0.3367
1.200	291.2	8.35	0.3448
1.300	296.8	8.73	0.3471
1.400	296.8	8.73	0.3471
1.500	302.4	8.68	0.3635
1.600	285.6	8.79	0.3646
1.700	280	8.62	0.3693
1.800	280	8.69	0.3771
1.900	274.4	8.78	0.3854
2.000	308	17.13	0.3781
2.100	324.8	18.18	0.3785
2.200	364	19.8907	0.3723
2.300	358.4	24.7	0.3791
2.400	347.2	23.97	0.3869
2.500	364	23.78	0.3965
2.600	364	22.23	0.4122
2.700	341.6	26.07	0.4209
2.800	358.4	23.28	0.4212
2.900	347.2	20.48	0.4314
3.000	336	21.74	0.4301
3.100	341.6	21.34	0.4302
3.200	336	23.61	0.4349
3.300	330.4	23.38	0.4426
3.400	330.4	25.18	0.445
3.500	324.8	24.9	0.4508
3.600	352.8	24.92	0.4594
3.700	308	32.36	0.4435
3.800	481.6	112.71	0.3291
3.900	436.8	139.17	0.1259
4.000	442.4	142.32	0.1023
4.100	397.6	141.7	0.1226
4.200	560	143.29	0.1096
4.300	560	143.28	0.1124
4.400	504	141.74	0.1339
4.500	509.6	143.15	0.1194

Table A.5: Indices of Performance, Ring, Last M, $\delta = 0.2$

L=10 & N=100			
Ring, Last M, & deldb=0.2			
K - Gain	Max Pout	Num Cyc.	StdDev
0.100	560	20.89	0.2973
0.200	554.4	20.78	0.3061
0.300	526.4	20.56	0.3148
0.400	509.6	20.57	0.3174
0.500	498.4	20.67	0.3192
0.600	498.4	20.39	0.3274
0.700	481.6	20.32	0.3307
0.800	459.2	19.82	0.344
0.900	425.6	19.96	0.3447
1.000	392	20.62	0.3455
1.100	392	19.98	0.3624
1.200	392	20.5	0.3659
1.300	392	20.52	0.3709
1.400	392	20.59	0.3735
1.500	408.8	20.61	0.3777
1.600	425.6	21.84	0.3881
1.700	403.2	22.38	0.3968
1.800	414.4	24.75	0.3952
1.900	436.8	28.94	0.3904
2.000	436.8	35.4	0.386
2.100	431.2	36.25	0.38
2.200	436.8	40.37	0.3858
2.300	476	46.62	0.3741
2.400	481.6	51.31	0.3825
2.500	476	49.17	0.3884
2.600	476	44.13	0.4096
2.700	476	56.3	0.39
2.800	464.8	50.81	0.3981
2.900	453.6	49.09	0.4093
3.000	459.2	49.21	0.4155
3.100	464.8	52.54	0.4215
3.200	464.8	83.46	0.3824
3.300	464.8	102.63	0.3482
3.400	459.2	103.1	0.364
3.500	448	140.7	0.1405
3.600	448	144.06	0.1325
3.700	425.6	143.85	0.129
3.800	560	148.2	0.1104
3.900	560	150.24	0.0988
4.000	560	150.15	0.0997
4.100	560	150.27	0.0995
4.200	560	150.1	0.1012
4.300	560	148.04	0.1159
4.400	543.2	147.6	0.1664
4.500	560	147.68	0.134

APPENDIX B

RESILIENCE METRICS

Table B.1: IOP of Resilience with Concurrent, Square Lattice, & $\delta = 0.5$

L	N	N Attacked	Max Power	Num Cycles	Standard Dev	Percent Incr.
5	25	0%	78.4	7.96	0.34306	
		10%	84	8.24	0.33538	7.14%
		20%	89.6	8.24	0.33158	14.29%
		30%	100.8	8.48	0.31645	28.57%
		40%	100.8	8.6	0.31113	28.57%
		50%	106.4	8.6	0.30421	35.71%
10	100	0%	296.8	8.15	0.31444	
		10%	302.4	8.28	0.33218	1.89%
		20%	341.6	8.31	0.32882	15.09%
		30%	369.6	8.44	0.32468	24.53%
		40%	397.6	8.53	0.31648	33.96%
		50%	414.4	8.64	0.3061	39.62%
15	225	0%	660.8	8.1644	0.33349	
		10%	677.6	8.3289	0.33131	2.54%
		20%	744.8	8.4133	0.32316	12.71%
		30%	817	8.502	0.31969	23.64%
		40%	873.6	8.6	0.30968	32.20%
		50%	935.2	8.64	0.30226	41.53%
20	400	0%	1148	8.2775	0.33074	
		10%	1198.4	8.3225	0.32993	4.39%
		20%	1310.4	8.4075	0.32625	14.15%
		30%	1416.8	8.53	0.31905	23.41%
		40%	1540	8.665	0.31198	34.15%
		50%	1624	8.6475	0.30779	41.46%
25	625	0%	1752.8	8.1968	0.33561	
		10%	1881.6	8.3344	0.32983	7.35%
		20%	2060.8	8.4432	0.32587	17.57%
		30%	2223.2	8.5072	0.31984	26.84%
		40%	2391.2	8.6288	0.31085	36.42%
		50%	2553.6	8.744	0.30119	45.69%
30	900	0%	2576	8.26	0.33173	
		10%	2676.8	8.3233	0.33202	3.91%
		20%	2917.6	8.4189	0.32663	13.26%
		30%	3175.2	8.5044	0.32102	23.26%
		40%	3460.8	8.6011	0.31348	34.35%
		50%	3656.8	8.6944	0.30722	41.96%
35	1225	0%	3416	8.3012	0.33383	
		10%	3623.2	8.3902	0.33057	6.07%
		20%	3953.6	8.4833	0.32575	15.74%
		30%	4295.2	8.5502	0.32068	25.74%
		40%	4670.4	8.6563	0.31326	36.72%
		50%	4972.8	8.7314	0.30713	45.57%
40	1600	0%	4396	8.3281	0.33365	
		10%	4788	8.4087	0.33049	8.92%
		20%	5191.2	8.5044	0.32508	18.09%
		30%	5667.2	8.6019	0.31851	28.92%
		40%	6109.6	8.6906	0.31057	38.98%
		50%	6490.4	8.7688	0.30317	47.64%
45	2025	0%	5628	8.3259	0.33256	
		10%	6003.2	8.3941	0.33056	6.67%
		20%	6585.6	8.482	0.32615	17.01%
		30%	7168	8.5788	0.3196	27.36%
		40%	7733.6	8.7032	0.31032	37.41%
		50%	8176	8.7788	0.30332	45.27%
50	2500	0%	6871.2	8.392	0.3319	
		10%	7694.4	8.4759	0.32813	11.98%
		20%	8366.4	8.558	0.32315	21.76%
		30%	8892.8	8.6276	0.31753	29.42%
		40%	9592.8	8.7164	0.3095	39.61%
		50%	10181	8.7832	0.3037	48.17%

Table B.2: IOP of Resilience with Concurrent, Ring Lattice, & $\delta = 0.5$

L	N	N Attacked	Max Power	Num Cycles	Standard Dev	Percent Incr.
5	25	0%	84	8.24	0.32565	
		10%	89.6	8.4	0.32775	6.67%
		20%	89.6	8.44	0.32646	6.67%
		30%	95.2	8.56	0.32666	13.33%
		40%	106.4	8.64	0.31913	26.67%
		50%	106.4	8.72	0.31014	26.67%
10	100	0%	313.6	8.42	0.31986	
		10%	324.8	8.53	0.32518	3.57%
		20%	347.2	8.6	0.32585	10.71%
		30%	375.2	8.63	0.32447	19.64%
		40%	408.8	8.71	0.31916	30.36%
		50%	414.4	8.75	0.31751	32.14%
15	225	0%	672	8.4844	0.31926	
		10%	655.2	8.5422	0.32488	-2.50%
		20%	716.8	8.64	0.3243	6.67%
		30%	784	8.7067	0.32264	16.67%
		40%	851.2	8.7689	0.31843	26.67%
		50%	912.8	8.8356	0.31391	35.83%
20	400	0%	1125.6	8.635	0.31414	
		10%	1181.6	8.665	0.31998	4.98%
		20%	1271.2	8.6925	0.3215	12.94%
		30%	1377.6	8.7525	0.32021	22.39%
		40%	1512	8.83	0.3153	34.33%
		50%	1596	8.8675	0.31147	41.79%
25	625	0%	1696.8	8.4928	0.32021	
		10%	1820	8.5456	0.32437	7.26%
		20%	1988	8.608	0.32474	17.16%
		30%	2172.8	8.6768	0.32275	28.05%
		40%	2357.6	8.752	0.31825	38.94%
		50%	2480.8	8.7984	0.31444	46.20%
30	900	0%	2536.8	8.72	0.31479	
		10%	2732.8	8.7689	0.319	7.73%
		20%	3001.6	8.8244	0.31896	18.32%
		30%	3264.8	8.8967	0.31646	28.70%
		40%	3500	8.9544	0.31177	37.97%
		50%	3595.2	8.9667	0.30811	41.72%
35	1225	0%	3522.4	8.7829	0.31539	
		10%	3651.2	8.8024	0.32061	3.66%
		20%	4026.4	8.8302	0.32172	14.31%
		30%	4407.2	8.8686	0.32503	25.12%
		40%	4692.8	8.9127	0.31652	33.23%
		50%	4922.4	8.942	0.31274	39.75%
40	1600	0%	4502.4	8.8606	0.31533	
		10%	4967.2	8.863	0.31942	10.32%
		20%	5415.2	8.925	0.31981	20.27%
		30%	5857.6	8.9712	0.31739	30.10%
		40%	6160	9.0175	0.31278	36.82%
		50%	6524	9.0194	0.3091	44.90%
45	2025	0%	5661.6	9.0173	0.31207	
		10%	6232.8	9.0281	0.31719	10.09%
		20%	6876.8	9.0375	0.31856	21.46%
		30%	7341.6	9.0568	0.31742	29.67%
		40%	773.6	9.0928	0.31346	-86.34%
		50%	8142.4	9.0928	0.3102	43.82%
50	2500	0%	7022.4	9.0892	0.31297	
		10%	7694.4	9.0948	0.31781	9.57%
		20%	8484	9.1068	0.31865	20.81%
		30%	9111.2	9.1284	0.31758	29.74%
		40%	9536.8	9.1576	0.31341	35.81%
		50%	10018	9.14	0.30997	42.66%

Table B.3: IOP of Resilience with Last m, Square Lattice, & $\delta = 0.5$

L	N	N Attacked	Max Power	Num Cycles	Standard Dev	Percent Incr.
5	25	0%	78.4	7.88	0.34712	
		10%	84	8.12	0.33578	7.14%
		20%	89.6	8.28	0.32523	14.29%
		30%	100.8	8.4	0.31357	28.57%
		40%	106.4	8.44	0.30963	35.71%
		50%	106.4	8.64	0.29986	35.71%
10	100	0%	296.8	8.15	0.3224	
		10%	324.8	8.46	0.31847	9.43%
		20%	336	8.29	0.33087	13.21%
		30%	364	8.64	0.30813	22.64%
		40%	392	8.48	0.31119	32.08%
		50%	420	8.63	0.30373	41.51%
15	225	0%	660.8	8.0578	0.33965	
		10%	688.8	8.2578	0.33287	4.24%
		20%	744.8	8.3333	0.32778	12.71%
		30%	817.6	8.4444	0.32188	23.73%
		40%	879.2	8.5778	0.31344	33.05%
		50%	924	8.6311	0.30996	39.83%
20	400	0%	1153.6	8.0825	0.33874	
		10%	1181.6	8.195	0.33561	2.43%
		20%	1310.4	8.2825	0.33149	13.59%
		30%	1433.6	8.395	0.32454	24.27%
		40%	1545.6	8.565	0.314	33.98%
		50%	1618.4	8.6175	0.30751	40.29%
25	625	0%	1764	8.096	0.34036	
		10%	1859.2	8.216	0.33691	5.40%
		20%	2032.8	8.3424	0.33261	15.24%
		30%	2223.2	8.4352	0.32573	26.03%
		40%	2385.6	8.56	0.31775	35.24%
		50%	2548	0.30832	0.30832	44.44%
30	900	0%	2553.6	8.1378	0.33977	
		10%	2671.2	8.2489	0.33605	4.61%
		20%	2923.2	8.3767	0.33001	14.47%
		30%	3197.6	8.4644	0.32282	25.22%
		40%	3438.4	8.5833	0.3154	34.65%
		50%	3640	8.6533	0.30786	42.54%
35	1225	0%	3472	8.2718	0.33528	
		10%	3673.6	8.3698	0.33308	5.81%
		20%	4060	8.4367	0.32793	16.94%
		30%	4356.8	8.5224	0.32208	25.48%
		40%	4681.6	8.6286	0.31382	34.84%
		50%	4961.6	8.7102	0.30762	42.90%
40	1600	0%	4440.8	8.1931	0.33907	
		10%	4804.8	8.3225	0.33612	8.20%
		20%	5241.6	8.4069	0.33143	18.03%
		30%	5728.8	8.5069	0.32578	29.00%
		40%	6148.8	8.6156	0.31787	38.46%
		50%	6512.8	8.715	0.30982	46.66%
45	2025	0%	5616.8	8.2523	0.33703	
		10%	6059.2	8.317	0.33561	7.88%
		20%	6630.4	8.4114	0.33045	18.05%
		30%	7201.6	8.5427	0.32312	28.22%
		40%	7800.8	8.6558	0.31397	38.88%
		50%	8254.4	8.7205	0.3082	46.96%
50	2500	0%	6927.2	8.2752	0.33672	
		10%	7722.4	8.396	0.3332	11.48%
		20%	8439.2	8.4952	0.32809	21.83%
		30%	8960	8.5824	0.32189	29.35%
		40%	9626.4	8.688	0.31425	38.97%
		50%	10203	8.7628	0.30717	47.29%

Table B.4: IOP of Resilience with Last m, Ring Lattice, & $\delta = 0.5$

L	N	N Attacked	Max Power	Num Cycles	Standard Dev	Percent Incr.
5	25	0%	84	8.4	0.32332	
		10%	89.6	8.48	0.32991	6.67%
		20%	89.6	8.4	0.3308	6.67%
		30%	95.2	8.48	0.3285	13.33%
		40%	106.4	8.52	0.32243	26.67%
		50%	106.4	8.72	0.31048	26.67%
10	100	0%	296.8	8.54	0.31918	
		10%	324.8	8.53	0.32526	9.43%
		20%	358.4	8.62	0.3245	20.75%
		30%	386.4	8.71	0.32149	30.19%
		40%	414.4	8.79	0.31638	39.62%
		50%	425.6	8.79	0.31285	43.40%
15	225	0%	649.6	8.6844	0.3164	
		10%	700	8.7067	0.32292	7.76%
		20%	750.4	8.76	0.32328	15.52%
		30%	817.6	8.8089	0.32188	25.86%
		40%	890.4	8.8711	0.31715	37.07%
		50%	924	8.8933	0.31372	42.24%
20	400	0%	1181.6	8.6975	0.31454	
		10%	1243.2	8.755	0.31839	5.21%
		20%	1366.4	8.7975	0.3192	15.64%
		30%	1461.6	8.84	0.31794	23.70%
		40%	1573.6	8.8775	0.31217	33.18%
		50%	1635.2	8.905	0.3073	38.39%
25	625	0%	1786.4	8.6112	0.31887	
		10%	1971.2	8.6784	0.3237	10.34%
		20%	2128	8.7392	0.32414	19.12%
		30%	2301.6	8.808	0.32251	28.84%
		40%	2480.8	8.8752	0.31759	38.87%
		50%	2548	8.92	0.31337	42.63%
30	900	0%	2553.6	8.7811	0.3155	
		10%	2828	8.8289	0.31988	10.75%
		20%	3080	8.8856	0.3199	20.61%
		30%	3292.8	8.9522	0.31769	28.95%
		40%	3505.6	9.0044	0.31335	37.28%
		50%	3606.4	9.0156	0.31013	41.23%
35	1225	0%	3483.2	8.9061	0.31277	
		10%	3656.8	8.9233	0.31827	4.98%
		20%	4020.8	8.9371	0.32033	15.43%
		30%	4401.6	8.9804	0.3191	26.37%
		40%	4625.6	9.0229	0.31576	32.80%
		50%	4905.6	9.0212	0.31254	40.84%
40	1600	0%	4384.8	8.9394	0.31567	
		10%	4849.6	8.9569	0.32009	10.60%
		20%	5353.6	9.005	0.31991	22.09%
		30%	5784.8	9.0556	0.31747	31.93%
		40%	6059.2	9.1019	0.31228	38.19%
		50%	6389.6	9.1	0.30873	45.72%
45	2025	0%	5544	9.1067	0.31294	
		10%	6067	9.1062	0.31826	9.43%
		20%	6770.4	9.1022	0.31964	22.12%
		30%	7184.8	9.1215	0.31884	29.60%
		40%	7700	9.1526	0.31479	38.89%
		50%	8080.8	9.1457	0.31167	45.76%
50	2500	0%	6787.2	9.1576	0.31491	
		10%	7509.6	9.1524	0.31974	10.64%
		20%	8243.2	9.1632	0.32048	21.45%
		30%	8786.4	9.1832	0.31944	29.46%
		40%	9503.2	9.2036	0.31571	40.02%
		50%	9900.8	9.1932	0.31216	45.87%

Table B.5: IOP of Resilience with Full Delay, Square Lattice, & $\delta = 0.5$

L	N	N Attacked	Max Power	Num Cycles	Standard Dev	Percent Incr.
5	25	0%	78.4	7.44	0.35534	
		10%	84	7.72	0.3502	7.14%
		20%	95.2	7.8	0.33915	21.43%
		30%	106.4	7.92	0.32468	35.71%
		40%	106.4	7.92	0.31959	35.71%
		50%	112	8	0.30968	42.86%
10	100	0%	302.4	7.79	0.34258	
		10%	319.2	7.86	0.33253	5.56%
		20%	341.6	7.96	0.3323	12.96%
		30%	364	8.09	0.32297	20.37%
		40%	397.6	8.09	0.31183	31.48%
		50%	420	8.14	0.29271	38.89%
15	225	0%	655.2	7.5333	0.34909	
		10%	683.2	7.7956	0.34358	4.27%
		20%	733.6	7.8667	0.34013	11.97%
		30%	800.8	7.9689	0.33286	22.22%
		40%	879.2	7.9378	0.32489	34.19%
		50%	929.6	8.0044	0.316	41.88%
20	400	0%	1170.4	7.6625	0.34616	
		10%	1209.6	7.8175	0.34385	3.35%
		20%	1332.8	7.885	0.33928	13.88%
		30%	1461.6	7.9125	0.33232	24.88%
		40%	1562.4	7.975	0.32039	33.49%
		50%	1646.4	8.0175	0.31455	40.67%
25	625	0%	1775.2	7.6416	0.34802	
		10%	1887.2	7.7872	0.34452	6.31%
		20%	2060.8	7.896	0.3403	16.09%
		30%	2251.2	7.9472	0.33226	26.81%
		40%	2396.8	7.9856	0.32413	35.02%
		50%	2581.6	8.0736	0.31333	45.43%
30	900	0%	2542.4	7.6778	0.34876	
		10%	2688	7.84	0.3436	5.73%
		20%	2945.6	7.9389	0.33808	15.86%
		30%	3214.4	7.9911	0.33072	26.43%
		40%	3455.2	8.0111	0.32278	35.90%
		50%	3701.6	8.1256	0.31239	45.59%
35	1225	0%	3483.2	7.7869	0.34654	
		10%	3673.6	7.8857	0.34371	5.47%
		20%	4060	8.0033	0.33924	16.56%
		30%	4373.6	8.0024	0.33224	25.56%
		40%	4676	8.0278	0.32418	34.24%
		50%	5017.6	8.0947	0.31727	44.05%
40	1600	0%	4396	7.7044	0.34663	
		10%	4849.6	7.8456	0.34426	10.32%
		20%	5292	7.9562	0.33974	20.38%
		30%	5756.8	7.97	0.33287	30.96%
		40%	6188	8.0175	0.32441	40.76%
		50%	6652.8	8.0919	0.31699	51.34%
45	2025	0%	5723.2	7.7625	0.34654	
		10%	6126.4	7.9007	0.34379	7.05%
		20%	6669.6	8.0025	0.33913	16.54%
		30%	7285.6	8.0286	0.33187	27.30%
		40%	7800.8	8.0756	0.3223	36.30%
		50%	8304.8	8.1373	0.31544	45.11%
50	2500	0%	8304.8	6960.8	0.34558	
		10%	7800.8	7.96	0.34183	-6.07%
		20%	8484	8.054	0.33663	2.16%
		30%	8988	8.07	0.33016	8.23%
		40%	9671.2	8.0916	0.3216	16.45%
		50%	10354	8.1716	0.31295	24.67%

Table B.6: IOP of Resilience with Full Delay, Ring Lattice, & $\delta = 0.5$

L	N	N Attacked	Max Power	Num Cycles	Standard Dev	Percent Incr.
5	25	0%	84	7.72	0.3408	
		10%	95.2	7.8	0.34294	13.33%
		20%	95.2	7.88	0.3406	13.33%
		30%	106.4	7.8	0.3365	26.67%
		40%	112	7.88	0.32923	33.33%
		50%	112	8.08	0.31632	33.33%
10	100	0%	308	8.09	0.32699	
		10%	319.2	8.14	0.33183	3.64%
		20%	358.4	8.17	0.33149	16.36%
		30%	386.4	8.09	0.33092	25.45%
		40%	408.8	8.14	0.32325	32.73%
		50%	420	8.16	0.32026	36.36%
15	225	0%	644	8.0889	0.32813	
		10%	700	8.1467	0.33426	8.70%
		20%	750.4	8.2222	0.33451	16.52%
		30%	812	8.2	0.33257	26.09%
		40%	884.8	8.1867	0.3287	37.39%
		50%	935.2	8.2089	0.32344	45.22%
20	400	0%	1142.4	8.1075	0.3273	
		10%	1243.2	8.195	0.33053	8.82%
		20%	1372	8.265	0.33046	20.10%
		30%	1467.2	8.2325	0.32958	28.43%
		40%	1562.4	8.215	0.32376	36.76%
		50%	1640.8	8.2525	0.31793	43.63%
25	625	0%	1792	8.032	0.33027	
		10%	1982.4	8.1152	0.33402	10.63%
		20%	2133.6	8.1984	0.33402	19.06%
		30%	2301.6	8.2	0.3314	28.44%
		40%	2475.2	8.2016	0.32629	38.13%
		50%	2581.6	8.2368	0.32167	44.06%
30	900	0%	2576	8.2656	0.32497	
		10%	2844.8	8.3322	0.32864	10.43%
		20%	3096.8	8.4022	0.32838	20.22%
		30%	3326.4	8.3878	0.32567	29.13%
		40%	3500	8.3756	0.32117	35.87%
		50%	3696	8.3856	0.31664	43.48%
35	1225	0%	3466.4	8.3804	0.32386	
		10%	3740.8	8.431	0.3281	7.92%
		20%	4116	8.4808	0.32868	18.74%
		30%	4452	8.4392	0.32739	28.43%
		40%	4653.6	8.4008	0.32418	34.25%
		50%	4933.6	8.4024	0.31931	42.33%
40	1600	0%	4625.6	8.4219	0.32567	
		10%	5101.6	8.4838	0.32902	10.29%
		20%	5594.4	8.5356	0.32849	20.94%
		30%	6014.4	8.5075	0.32546	30.02%
		40%	6193.6	8.4731	0.32032	33.90%
		50%	6540.8	8.4669	0.31591	41.40%
45	2025	0%	5958.4	8.524	0.32436	
		10%	6585.6	8.5536	0.3292	10.53%
		20%	7235.2	8.5793	0.32978	21.43%
		30%	7565.6	8.5269	0.32831	26.97%
		40%	7733.6	8.4998	0.3239	29.79%
		50%	8282.4	8.4938	0.31902	39.00%
50	2500	0%	7610.4	8.6052	0.32547	
		10%	8388.8	8.652	0.3295	10.23%
		20%	9206.4	8.6928	0.32959	20.97%
		30%	9430.4	8.6512	0.32789	23.91%
		40%	9475.2	8.6	0.32393	24.50%
		50%	10114	8.5888	0.31923	32.90%

Table B.7: IOP of Resilience with Concurrent, Square Lattice, & $\delta = 0.2$

L	N	N Attacked	Max Power	Num Cycles	Standard Dev	Percent Incr.
5	25	0%	106.4	18.48	0.3665	
		10%	112	18.84	0.3608	5.26%
		20%	117.6	19.52	0.3419	10.53%
		30%	128.8	19.6	0.3344	21.05%
		40%	134	19.8	0.3267	25.94%
		50%	134.4	19.92	0.3084	26.32%
10	100	0%	414.4	19.46	0.3412	
		10%	453.6	19.86	0.326	9.46%
		20%	464.8	20.03	0.3207	12.16%
		30%	481.6	20.12	0.3199	16.22%
		40%	526.4	20.23	0.3051	27.03%
		50%	548.8	20.31	0.2902	32.43%
15	225	0%	918.4	18.8578	0.3555	
		10%	1019.2	19.3778	0.3449	10.98%
		20%	1024.8	19.7378	0.3364	11.59%
		30%	1064	19.6844	0.3342	15.85%
		40%	1164.8	20.0089	0.3196	26.83%
		50%	1232	20.0667	0.3073	34.15%
20	400	0%	1680	18.98	0.355	
		10%	1803.2	19.3025	0.3493	7.33%
		20%	1859.2	19.4525	0.344	10.67%
		30%	1926.4	19.6625	0.3357	14.67%
		40%	2044	20.31	0.3103	21.67%
		50%	2144.8	20.39	0.2968	27.67%
25	625	0%	2564.8	19.144	0.3521	
		10%	2788.8	19.3408	0.3492	8.73%
		20%	2900.8	19.6144	0.3411	13.10%
		30%	3012.8	19.728	0.3354	17.47%
		40%	3192	19.9664	0.325	24.45%
		50%	3360	20.0416	0.3128	31.00%
30	900	0%	3673.6	19.1167	0.3532	
		10%	3964.8	19.2944	0.3501	7.93%
		20%	4110.4	19.5244	0.3433	11.89%
		30%	4323.2	19.7444	0.3344	17.68%
		40%	4575.2	19.91	0.3255	24.54%
		50%	4832.8	20.0511	0.3114	31.55%
35	1225	0%	5034.4	19.2971	0.3532	
		10%	5342.4	19.5502	0.3495	6.12%
		20%	5516	19.7037	0.3446	9.57%
		30%	5796	19.8735	0.3371	15.13%
		40%	6204.8	20.0269	0.3274	23.25%
		50%	6557.6	20.1649	0.3123	30.26%
40	1600	0%	6529.6	19.2069	0.3531	
		10%	7022.4	19.44	0.3498	7.55%
		20%	7313.6	19.6062	0.3446	12.01%
		30%	7627.2	19.8281	0.3355	16.81%
		40%	8108.8	19.9975	0.3255	24.19%
		50%	8523.2	20.1275	0.3115	30.53%
45	2025	0%	8237.6	19.2953	0.3522	
		10%	8814.4	19.5462	0.3479	7.00%
		20%	9178.4	19.7388	0.3417	11.42%
		30%	9559.2	19.9388	0.3339	16.04%
		40%	10198	20.119	0.32269	23.80%
		50%	10730	20.226	0.3099	30.26%
50	2500	0%	10276	19.468	0.35042	
		10%	10937	19.72	0.34611	6.43%
		20%	11402	19.892	0.34066	10.96%
		30%	11844	20.042	0.33326	15.26%
		40%	12550	20.221	0.32291	22.13%
		50%	13205	20.279	0.30991	28.50%

Table B.8: IOP of Resilience with Concurrent, Ring Lattice, & $\delta = 0.2$

L	N	N Attacke	Max Power	Num Cycles	Standard Dev	Percent Incr.
5	25	0%	100.8	19.4	0.3452	
		10%	112	19.28	0.357	11.11%
		20%	117.6	19.64	0.3525	16.67%
		30%	128.8	19.88	0.34515	27.78%
		40%	128.8	19.8	0.3392	27.78%
		50%	134.4	20	0.3231	33.33%
10	100	0%	403.2	19.52	0.3435	
		10%	442.4	19.56	0.3475	9.72%
		20%	459.2	19.73	0.3466	13.89%
		30%	470.4	19.97	0.3413	16.67%
		40%	504	20.1	0.3334	25.00%
		50%	526.4	20.1	0.323	30.56%
15	225	0%	963.2	20.2311	0.3346	
		10%	1013.3	20.2756	0.3407	5.20%
		20%	1030.4	20.3556	0.3418	6.98%
		30%	1064	20.4756	0.3378	10.47%
		40%	1125.6	20.4844	0.3308	16.86%
		50%	1192.8	20.5511	0.3188	23.84%
20	400	0%	1713.6	20.34	0.3362	
		10%	1803.2	20.405	0.3408	5.23%
		20%	1808.8	20.53	0.3404	5.56%
		30%	1831.2	20.6225	0.337	6.86%
		40%	1960	20.655	0.3298	14.38%
		50%	2111.2	20.66	0.3187	23.20%
25	625	0%	2671.2	20.5536	0.3349	
		10%	2811.2	20.6528	0.34	5.24%
		20%	2833.6	20.707	0.3416	6.08%
		30%	2856	20.7904	0.3385	6.92%
		40%	3046.4	20.824	0.3324	14.05%
		50%	3309.6	20.8112	0.3209	23.90%
30	900	0%	3908.8	20.8656	0.3324	
		10%	4076.8	20.8833	0.338	4.30%
		20%	4088	20.9156	0.3392	4.58%
		30%	4110.4	20.9578	0.3366	5.16%
		40%	4373.6	21.0022	0.3308	11.89%
		50%	4790	20.9622	0.3201	22.54%
35	1225	0%	5364.8	21.0931	0.3318	
		10%	5566.4	21.0612	0.3386	3.76%
		20%	5566.4	21.0441	0.3409	3.76%
		30%	5577.6	21.102	0.3382	3.97%
		40%	5919.2	21.1502	0.3318	10.33%
		50%	6462.4	21.1118	0.3214	20.46%
40	1600	0%	7044.8	21.326	0.33064	
		10%	7285	21.358	0.3367	3.41%
		20%	7285.6	21.38	0.3386	3.42%
		30%	7296.8	21.392	0.33644	3.58%
		40%	7716.8	21.402	0.33056	9.54%
		50%	8450.4	21.273	0.32057	19.95%
45	2025	0%	8937.6	21.685	0.32738	
		10%	9234.4	21.665	0.33449	3.32%
		20%	9234.4	21.635	0.33718	3.32%
		30%	9245.6	21.619	0.33562	3.45%
		40%	9749.6	21.575	0.33024	9.09%
		50%	10690	21.432	0.32023	19.61%
50	2500	0%	11060	21.956	0.32488	
		10%	11413	21.862	0.3335	3.19%
		20%	11413	21.796	0.33672	3.19%
		30%	11424	21.764	0.33542	3.29%
		40%	12034	21.703	0.33026	8.81%
		50%	13205	21.553	0.32063	19.39%

Table B.9: IOP of Resilience with Last m, Square Lattice, & $\delta = 0.2$

L	N	N Attacked	Max Power	Num Cycles	Standard Dev	Percent Incr.
5	25	0%	106.4	18.04	0.3684	
		10%	112	18.56	0.36683	5.26%
		20%	117.6	18.96	0.35196	10.53%
		30%	128.8	19.2	0.34094	21.05%
		40%	134.4	19.32	0.33345	26.32%
		50%	134.4	19.64	0.31472	26.32%
10	100	0%	414.4	18.68	0.34395	
		10%	453.6	18.85	0.35829	9.46%
		20%	464.8	19.01	0.35613	12.16%
		30%	487.2	19.27	0.34409	17.57%
		40%	526.4	19.54	0.33412	27.03%
		50%	548.8	19.7	0.31969	32.43%
15	225	0%	929.6	18.564	0.35983	
		10%	1013.6	18.893	0.3546	9.04%
		20%	1013.6	19.142	0.34872	9.04%
		30%	1064	19.391	0.34088	14.46%
		40%	1164.8	19.582	0.32895	25.30%
		50%	1232	19.8	0.31394	32.53%
20	400	0%	1680	18.695	0.36284	
		10%	1803.2	18.948	0.35767	7.33%
		20%	1803.2	19.18	0.35209	7.33%
		30%	1853.6	19.39	0.34423	10.33%
		40%	2027.2	19.652	0.33197	20.67%
		50%	2144.8	19.823	0.31787	27.67%
25	625	0%	2643.2	19.011	0.36129	
		10%	2794.4	19.27	0.35662	5.72%
		20%	2794.4	19.491	0.35182	5.72%
		30%	2861.6	19.706	0.3432	8.26%
		40%	3136	19.957	0.33174	18.64%
		50%	3360	19.989	0.31645	27.12%
30	900	0%	3925.6	19.082	0.36104	
		10%	4071.2	19.237	0.35802	3.71%
		20%	4071.2	19.443	0.35287	3.71%
		30%	4127.2	19.69	0.34409	5.14%
		40%	4485.6	19.886	0.33365	14.27%
		50%	4832.8	20.002	0.31893	23.11%
35	1225	0%	5432	19.417	0.35438	
		10%	5527.2	19.529	0.35239	1.75%
		20%	5527.2	19.687	0.34795	1.75%
		30%	5572	19.907	0.3405	2.58%
		40%	6059.2	20.064	0.33013	11.55%
		50%	6557.6	20.17	0.31411	20.72%
40	1600	0%	7039.2	19.246	0.35891	
		10%	7257.6	19.474	0.35541	3.10%
		20%	7257.6	19.662	0.34986	3.10%
		30%	7285.6	19.886	0.34132	3.50%
		40%	7918.4	20.053	0.33054	12.49%
		50%	8532.2	20.159	0.31708	21.21%
45	2025	0%	8836.8	19.304	0.35768	
		10%	9212	19.483	0.35471	4.25%
		20%	9212	19.697	0.34902	4.25%
		30%	9223.2	19.896	0.3404	4.37%
		40%	9962.4	20.073	0.32959	12.74%
		50%	10730	20.162	0.31672	21.42%
50	2500	0%	10920	19.41	0.3566	
		10%	11407	19.641	0.35338	4.46%
		20%	11407	19.826	0.34883	4.46%
		30%	11407	20.018	0.34131	4.46%
		40%	12270	20.214	0.33001	12.36%
		50%	13205	20.277	0.31678	20.92%

Table B.10: IOP of Resilience with Last m, Ring Lattice, & $\delta = 0.2$

L	N	N Attacke	Max Power	Num Cycles	Standard Dev	Percent Incr.
5	25	0%	106.4	19.2	0.34667	
		10%	117.6	19.2	0.35544	10.53%
		20%	117.6	19.6	0.34972	10.53%
		30%	123.2	20.16	0.33802	15.79%
		40%	128.8	20.2	0.32936	21.05%
		50%	134.4	20	0.32013	26.32%
10	100	0%	459.2	19.82	0.34402	
		10%	476	19.98	0.34749	3.66%
		20%	476	20.01	0.34846	3.66%
		30%	476	20.1	0.3446	3.66%
		40%	492.8	20.3	0.33595	7.32%
		50%	526.4	20.36	0.32378	14.63%
15	225	0%	996.8	20.32	0.33851	
		10%	1041.6	20.431	0.34265	4.49%
		20%	1041.6	20.533	0.34288	4.49%
		30%	1041.6	20.604	0.33886	4.49%
		40%	1103.2	20.693	0.33114	10.67%
		50%	1192.8	20.644	0.3207	19.66%
20	400	0%	1764	20.395	0.33984	
		10%	1836.8	20.48	0.34417	4.13%
		20%	1836.8	20.6	0.34432	4.13%
		30%	1836.8	20.663	0.3415	4.13%
		40%	1943.2	20.77	0.33435	10.16%
		50%	2111.2	20.765	0.32312	19.68%
25	625	0%	2749.6	20.752	0.33759	
		10%	2844.8	20.754	0.34316	3.46%
		20%	2844.8	20.83	0.34355	3.46%
		30%	2844.8	20.909	0.34032	3.46%
		40%	3024	20.992	0.3328	9.98%
		50%	3309.6	20.923	0.32252	20.37%
30	900	0%	3987.2	20.987	0.33716	
		10%	4110.4	20.977	0.34285	3.09%
		20%	4110.4	21.022	0.34377	3.09%
		30%	4110.4	21.086	0.34059	3.09%
		40%	4345.6	21.103	0.3339	8.99%
		50%	4760	21.038	0.32362	19.38%
35	1225	0%	5448.8	21.238	0.33567	
		10%	5605.6	21.203	0.34152	2.88%
		20%	5605.6	21.219	0.34287	2.88%
		30%	5605.6	21.271	0.33967	2.88%
		40%	5891.2	21.317	0.33206	8.12%
		50%	6462.4	21.243	0.32148	18.60%
40	1600	0%	7123.2	21.529	0.33328	
		10%	7324.8	21.45	0.34024	2.83%
		20%	7324.8	21.432	0.34186	2.83%
		30%	7324.8	21.446	0.33908	2.83%
		40%	7688.8	21.436	0.33285	7.94%
		50%	8450.4	21.326	0.32264	18.63%
45	2025	0%	9016	21.714	0.33271	
		10%	9273.6	21.636	0.33916	2.86%
		20%	9273.6	21.616	0.34066	2.86%
		30%	9273.6	21.607	0.33817	2.86%
		40%	9721.6	21.581	0.3318	7.83%
		50%	10690	21.452	0.32122	18.57%
50	2500	0%	11138	21.873	0.33251	
		10%	11458	21.793	0.33908	2.87%
		20%	11458	21.749	0.34079	2.87%
		30%	11458	21.718	0.33834	2.87%
		40%	12006	21.673	0.33206	7.79%
		50%	13205	21.53	0.32174	18.56%

Table B.11: IOP of Resilience with Full Delay, Square Lattice, & $\delta = 0.2$

L	N	N Attacked	Max Power	Num Cycles	Standard Dev	Percent Incr.
5	25	0%	106.4	16.72	0.39247	
		10%	112	16.96	0.38787	5.26%
		20%	117.6	17.16	0.37561	10.53%
		30%	128.8	17.24	0.36852	21.05%
		40%	134.4	17.48	0.35545	26.32%
		50%	134.4	17.48	0.34129	26.32%
10	100	0%	431.2	16.97	0.38729	
		10%	464.8	17.19	0.37708	7.79%
		20%	481.6	17.59	0.34988	11.69%
		30%	492.8	17.6	0.36012	14.29%
		40%	526.4	17.73	0.34185	22.08%
		50%	548.8	17.86	0.32385	27.27%
15	225	0%	929.6	17.058	0.38207	
		10%	1024.8	17.311	0.37313	10.24%
		20%	1030.4	17.436	0.36763	10.84%
		30%	1069.6	17.591	0.36287	15.06%
		40%	1164.8	17.968	0.34734	25.30%
		50%	1232	17.72	0.33702	32.53%
20	400	0%	1663.2	17.177	0.38494	
		10%	1797.6	17.275	0.37905	8.08%
		20%	1820	17.425	0.37316	9.43%
		30%	1870.4	17.587	0.36121	12.46%
		40%	2027.2	17.73	0.35155	21.89%
		50%	2144.8	17.805	0.33787	28.96%
25	625	0%	2144.8	17.805	0.33787	
		10%	2844.8	17.502	0.38025	32.64%
		20%	2867.2	17.624	0.37489	33.68%
		30%	2934.4	17.786	0.36632	36.81%
		40%	3136	17.909	3136	46.21%
		50%	3360	17.877	0.34013	56.66%
30	900	0%	3959.2	17.39	0.38552	
		10%	4110.4	17.488	0.38202	3.82%
		20%	4116	17.614	0.37622	3.96%
		30%	4200	17.744	0.36686	6.08%
		40%	4485.6	17.854	0.35562	13.30%
		50%	4832.8	17.908	0.34201	22.07%
35	1225	0%	5566.4	17.679	0.38079	
		10%	5622.4	17.683	0.37834	1.01%
		20%	5628	17.796	0.37252	1.11%
		30%	5723.2	17.882	0.36428	2.82%
		40%	6059.2	18.024	0.35298	8.85%
		50%	6557.6	18.018	0.34056	17.81%
40	1600	0%	7190.4	17.511	0.38197	
		10%	7414.4	17.652	0.37825	3.12%
		20%	7420	17.762	0.37288	3.19%
		30%	7481.6	17.875	0.36384	4.05%
		40%	7884.8	17.98	0.35304	9.66%
		50%	8523.2	18.012	0.33986	18.54%
45	2025	0%	9010.4	17.59	0.38122	
		10%	9396.8	17.706	0.37799	4.29%
		20%	9402.4	17.8	0.37224	4.35%
		30%	9419.2	17.916	0.36311	4.54%
		40%	9968	18.024	0.35205	10.63%
		50%	10730	18.025	0.33924	19.08%
50	2500	0%	11161	17.69	0.37951	
		10%	11637	17.808	0.37652	4.26%
		20%	11654	17.904	0.37107	4.42%
		30%	11654	18.024	0.36276	4.42%
		40%	12314	18.12	0.35174	10.33%
		50%	13205	18.15	0.33921	18.31%

Table B.12: IOP of Resilience with Full Delay, Ring Lattice, & $\delta = 0.2$

L	N	N Attacke	Max Power	Num Cycles	Standard Dev	Percent Incr.
5	25	0%	106.4	17.32	0.37242	
		10%	112	17.48	0.37669	5.26%
		20%	112	17.6	0.3761	5.26%
		30%	123.2	17.72	0.36567	15.79%
		40%	128.8	17.88	0.35817	21.05%
		50%	134.4	18	0.34259	26.32%
10	100	0%	453.6	17.9	0.36813	
		10%	476	17.98	0.37056	4.94%
		20%	476	18.05	0.36957	4.94%
		30%	476	18	0.36693	4.94%
		40%	492.8	18.14	0.3558	8.64%
		50%	526.4	18.24	0.34537	16.05%
15	225	0%	1008	18.542	0.35828	
		10%	1047.2	18.533	0.36296	3.89%
		20%	1052.8	18.636	0.36154	4.44%
		30%	1052.8	18.618	0.35781	4.44%
		40%	1097.6	18.64	0.34904	8.89%
		50%	1192.8	18.542	0.33988	18.33%
20	400	0%	1797.6	18.61	0.35857	
		10%	1842.4	18.517	0.36425	2.49%
		20%	1848	18.608	0.36264	2.80%
		30%	1848	18.593	0.35902	2.80%
		40%	1943.2	18.657	0.35047	8.10%
		50%	2111.2	18.6	0.34052	17.45%
25	625	0%	2828	18.816	0.35768	
		10%	2889.6	18.81	0.36216	2.18%
		20%	2900.8	18.813	0.36235	2.57%
		30%	2900.8	18.808	0.3584	2.57%
		40%	3035.2	18.79	0.35074	7.33%
		50%	3309.6	18.702	0.34134	17.03%
30	900	0%	4121.6	19.016	0.35632	
		10%	4188.8	18.997	0.36099	1.63%
		20%	4200	18.966	0.36124	1.90%
		30%	4200	18.962	0.35697	1.90%
		40%	4368	18.964	0.3492	5.98%
		50%	4760	18.841	0.33982	15.49%
35	1225	0%	5661.6	19.233	0.3544	
		10%	5723.2	19.128	0.36054	1.09%
		20%	5734.4	19.081	0.36149	1.29%
		30%	5734.4	19.045	0.35819	1.29%
		40%	5941.6	19.012	0.35143	4.95%
		50%	6462.4	18.875	0.34187	14.14%
40	1600	0%	7431.2	19.284	0.35591	
		10%	7492.8	19.232	0.36128	0.83%
		20%	7504	19.202	0.36196	0.98%
		30%	7504	19.163	0.35882	0.98%
		40%	7750.4	19.107	0.35228	4.30%
		50%	8450.4	18.952	0.34313	13.72%
45	2025	0%	9430.4	19.613	0.35196	
		10%	9492	19.508	0.35846	0.65%
		20%	9503.2	19.43	0.35986	0.77%
		30%	9503.2	19.369	0.35673	0.77%
		40%	9794.4	19.29	0.35032	3.86%
		50%	10690	19.117	0.34131	13.36%
50	2500	0%	11665	19.754	0.35115	
		10%	11743	19.626	0.35752	0.67%
		20%	11754	19.533	0.35853	0.76%
		30%	11754	19.452	0.35524	0.76%
		40%	12102	19.37	0.3483	3.75%
		50%	13205	19.179	0.33894	13.20%

APPENDIX C

SOURCE CODE

C.1 Ring Lattice Generator

```

function adj = AdjMatrixRingLatticeGen(N_agents,N_connections)

%N_agents = Total number of agents in population (length of Adj
  Matrix)
%N_connections = Number of connections per agent

A = zeros(N_agents,N_agents); %initialize adj matrix

if mod(N_connections,2) == 0    %checks to make sure even number of
  connections

  for i = 1:N_connections/2

    v1 = ones(1,length(diag(A,i)));
    v2 = ones(1,length(diag(A,N_agents-i)));
    A = A + diag(v1,i) + diag(v1,-i) + ...
          diag(v2,N_agents-i) + diag(v2,-(N_agents-i));
  end
  adj = A;
else
  adj = 'Does not compute odd # of connections';

  % Odd number of connections is a more complex matrix since AnyLogic
  % flips which side has 2 connections vs 1 connection in a manner I
  % didn't
  % want to spend time figuring out since we are using 4 or 8
  % connections.

end
end

Not enough input arguments.
Error in AdjMatrixRingLatticeGen (line 6)
A = zeros(N_agents,N_agents); %initialize adj matrix

```

Published with MATLAB® R2018a

C.2 Square Lattice Generator

```

function adj = AdjMatrixLatticeSqGen(N,periodic)
%
% function to create an adjacency matrix for
% a square lattice graph with or without periodic
% boundary conditions
% N - size of the N * N lattice grid
% periodic: boolean
%
Id = eye(N);
Ze = zeros(N,N);
D1 = diag(ones(N-1,1),1);
D2 = Ze;
D2(1,N) = 1;
tmp=ones(N-1,1);
%A = diag(tmp,-1)+diag(tmp,1);
A = diag(tmp,1);
if periodic
    A(1,N) = 1;
end

ad1 = kron(Id, A);
ad2 = kron(D1,Id);
if periodic
    ad3 = kron(D2,Id);
else
    ad3 = kron(D2,Ze);
end
adj = ad1+ad2+ad3;
adj=adj+adj';
if N==2
    adj=[0 1 0 1;1 0 1 0;0 1 0 1;1 0 1 0];
end

Not enough input arguments.
Error in AdjMatrixLatticeSqGen (line 9)
Id = eye(N);

```

Published with MATLAB® R2018a

C.3 Eigenvalue Analysis

Table of Contents

.....	1
All Concurrent	2
All Concurrent jason test 10/21/18	2
Concurrent temp Last M	3
Last temp Last M	3

```

clear all;
format compact

% Model Parameters
R = 2;
C = 10;
Qc = -14.0;
Q=14;
Beta = 2.5;
Tset = 20.0;
delldb = 0.1;
Tinf= 32;
Qi=0;
kl=1;

L = 10; %setting up an LxL grid
N=L*L; %Number of Houses
graphtype = 'S';
%graphtype = 'S';
if graphtype == 'R'
    deg=4;
    LAdj=AdjMatrixRingLatticeGen(N,deg);
elseif graphtype == 'D'
    LAdj=AdjMatrixLatticeSqGen8(L,true);
    deg = 8;
else
    LAdj=AdjMatrixLatticeSqGen(L,true);
    deg = 4;
end

% Simulation times
delt = 1/60; % integrate set size (hr)
nhours = 24; % how long to run
DRHour = 5; % time when the DR event happens
DRDuration = 0.25; % duration in minuts of DR event
DRDuration = round(DRDuration/delt);
%
nsteps = round(nhours/delt);
stepDRStart = round(DRHour/nhours * nsteps);
stepDREnd = stepDRStart + DRDuration;
stepSWEnd = stepDRStart + 8*DRDuration;
stepSWEnd = 5000; %supress swarm end this way

```

```

hrc=-delt/(R*C);

m=zeros(N,nsteps);
rho=zeros(N,nsteps);
Non = round(0.428*N);
m(1:Non,1) = 1; % 42.8% of the devices are ON
T(1:N,1)=Tset - (deldb/2);
dt0ON = deldb/(Non-1);
dt0OFF = deldb/(N-Non-1);
for i=1:Non
    T(i,1)= Tset - (deldb/2) + dt0ON*(i-1);
end
for i = Non+1:N
    T(i,1)= Tset - (deldb/2) + dt0OFF*(i-Non-1);
end

P=zeros(N,1);
P(:,1)=m(:,1)*Q/Beta;
time=zeros(nsteps,1);
mtilde = (LAdj * m(:,1))/deg;
rhotilde = (LAdj * rho(:,1))/deg;

ei=0;
k=0.85;
count=1;
G=expm(hrc*eye(N));
H2=(R*Qc)*(eye(N)-expm((hrc)*eye(N)));

```

All Concurrent

```

while ei <1
Amatrix=G+((1/deldb)*H2*inv(eye(N)+(k/deg)*LAdj));
min(eig(Amatrix));
ei=max(abs(eig(Amatrix)));
k=k+0.0001;
count=count+1;

end
kCon=k-0.0001
Con_i=count;
beep

kCon =
    0.8834

```

All Concurrent jason test 10/21/18

```

ei=0; k=1; while ei <1 Amatrix=G+((1/deldb)*H2*(eye(N)-(k/deg)*LAdj)); min(eig(Amatrix));
ei=max(abs(eig(Amatrix))); k=k+0.001; count=count+1;

end kContest=k-0.001 Con_i=count;

```

Concurrent temp Last M

```
ei=0;
k=0.85;
count=1;
Gnew_T=zeros(4*N);
Gnew_T(1:N,1:N)=G;
Gnew_T(1:N,(2*N+1):3*N)=(1/delldb)*H2;
Gnew_T((N+1):2*N,1:N)=eye(N);
Gnew_T((2*N+1):3*N,1:N)=G;
Gnew_T((3*N+1):4*N,(2*N+1):3*N)=eye(N);

while ei <1
Gnew_T((2*N+1):3*N,(2*N+1):3*N)=(1/delldb)*H2-((k/deg)*LAdj);
ei=max(abs(eig(Gnew_T)));
k=k+0.0001;
count=count+1;

end

kLastM=k-0.0001
LastM_i=count;
beep

kLastM =
    0.8834
```

Last temp Last M

```
ei=0;
k=0.85;
count=1;
Gnew=zeros(4*N);
Gnew(1:N,1:N)=G;
Gnew(1:N,(2*N+1):3*N)=(1/delldb)*H2;
Gnew((N+1):2*N,1:N)=eye(N);
Gnew((2*N+1):3*N,1:N)=eye(N);
Gnew((3*N+1):4*N,(2*N+1):3*N)=eye(N);

while ei <1
Gnew((2*N+1):3*N,(2*N+1):3*N)=(1/delldb)*H2-((k/deg)*LAdj);
ei=max(abs(eig(Gnew)));
k=k+0.0001;
count=count+1;

end

kLastTM=k-0.0001
LastTM_i=count;
beep

kLastTM =
    0.8835
```

C.4 Create MATLAB Animation

```
%  
% Make an avi file from a movie  
%  
filename=uinputfile('*.avi');  
myVideo = VideoWriter(filename);  
myVideo.FrameRate=15;  
open(myVideo);  
writeVideo(myVideo, F);  
close(myVideo);
```

Published with MATLAB® R2018a

C.5 Homogenous Linear

```

clear all;

% Model Parameters
R = 2;
C = 10;
Qc = -14.0;
Q=14;
Beta = 2.5;
Tset = 20.0;
deldb = 0.5;
Tinf= 32;
Qi=0;
k1=0.8

L = 10; %setting up an LxL grid
N=L*L; %Number of Houses
graphtype = 'S';
%graphtype = 'S';
if graphtype == 'R'
    deg=4;
    LAdj=AdjMatrixRingLatticeGen(N,deg);
elseif graphtype == 'D'
    LAdj=AdjMatrixLatticeSqGen8(L,true);
    deg = 8;
else
    LAdj=AdjMatrixLatticeSqGen(L,true);
    deg = 4;
end

% Simulation times
delt = 1/60; % integrate set size (hr)
nhours = 12; % how long to run
DRHour = 2; % time when the DR event happens
DRDuration = 0.25; % duration in minuts of DR event
DRDuration = round(DRDuration/delt);
%
nsteps = round(nhours/delt);
stepDRStart = round(DRHour/nhours * nsteps);
stepDREnd = stepDRStart + DRDuration;
stepSWEnd = stepDRStart + 8*DRDuration;
stepSWEnd = 5000; %supress swarm end this way
hrc=-delt/(R*C);

cycles=zeros(N,1);
m=zeros(N,1);
m_last=zeros(N,1);

rho=zeros(N,1);
Non = round(0.428*N);
m(1:Non) = 1; % 42.8% of the devices are ON
T(1:N,1)=Tset - (deldb/2);
dt0ON = deldb/(Non-1);

```

```

dt0OFF = deldb/(N-Non-1);
for i=1:Non
    T(i,1)= Tset - (deldb/2) + dt0ON*(i-1);
    cycles(i)=1;
end
for i = Non+1:N
    T(i,1)= Tset - (deldb/2) + dt0OFF*(i-Non-1);
end

P=zeros(N,1);
P(:,1)=m*Q/Beta;
time=zeros(nsteps,1);
mtilde = (LAdj * m)/deg;
rhotilde = (LAdj * rho)/deg;

tic
for k = 1:nsteps-1
    for j = 1:N
        T(j,k+1)=exp(hrc)*T(j,k)+(1-exp(hrc))*Tinf+R*(1-exp(hrc))*Qi
        +(R*Qc)*(1-exp(hrc))*m(j);
    end

    rho(:,k+1) = makerhovec(T(:,k+1),Tset,deldb);
    time(k+1)=k*delt;
    mtilde(:,k+1) = (LAdj * m)/deg;
    rhotilde(:,k+1) = (LAdj * rho(:,k+1))/deg;
    m_last(:)=m(:);
    mout(k,:)=m(:);

    for j = 1:N
        % Simple Thermostat Model
        if k < (stepDRStart-1)
            % if rho(j,k+1) > 1
            m(j) = rho(j,k);
            % elseif rho(j,k+1) < 0
            m(j) = rho(j,k+1);
            % end
        end

        % Additional Criteria
        if k > (stepDRStart-1) && k < (stepSWEEnd-1)
            % if (rho(j,k) -k1*mtilde(j,k)) > 1
            m(j) = (rho(j,k+1) -k1*mtilde(j,k));
            % elseif (rho(j,k) - k1*mtilde(j,k)) <= 0
            m(j) = 0;
            % end
        end
    end

    if k>(stepDRStart-1) && k < (stepDREnd-1)
        m(:)=0;
    end

    P(j,k+1)=m(j)*Q/Beta;

```

```

        if (m(j)-m_last(j)) == 1
            cycles(j)=cycles(j)+1;
        end
    end

end

end
toc
pout=sum(P);
figure
plot(time,pout)
ylabel('Power')
xlabel('Time')
title('Discrete')

figure
plot(time,T')
ylabel('Temperature')
xlabel('Time')
% low_temp=round(min(T(:,stepDRStart:end)'),2);
% [lt,bin]=hist(low_temp,unique(low_temp));
% [~,idx]=sort(lt,'descend');
% temp_inst=lt(idx)
% temp_values=bin(idx)
% eff_db=mean(min(T(:,stepDRStart:end)))
% hold on
% reffline(0,temp_values(2))
% hold off
mean(cycles)
max(pout)
devation=mean(std(rho))
beep

k1 =

    0.8000

Elapsed time is 0.046633 seconds.

ans =

    0.4300

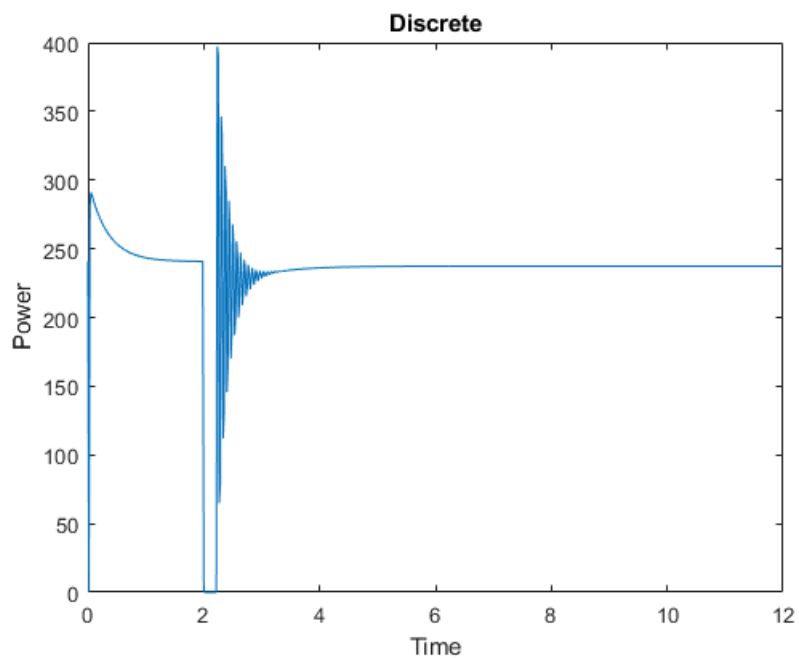
ans =

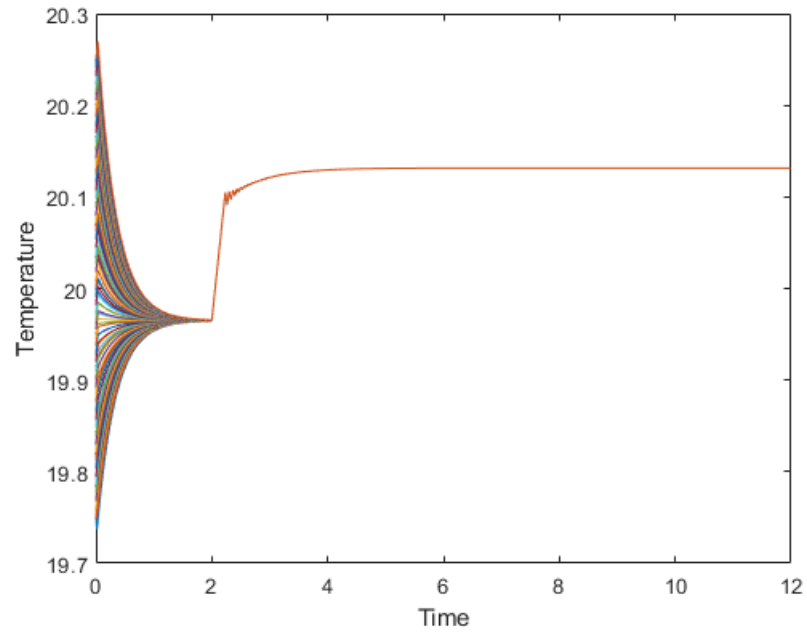
    397.1661

devation =

```

0.0087





Published with MATLAB® R2018a

C.6 Homogenous Non-Linear

Agent Based Model of TCL in Discrete Form

```

clear all;
format compact

% Files needed to include in folder
% -----
% AdjMatrixRingLatticeGen.m
% AdjMatrixLatticeSqGen8.m
% AdjMatrixLatticeSqGen.m
% makerhovec.m
% MakeAVI.m
% -----

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%
%%%%
%If video is desired
vid=0;          %[0] - No to video    [1] - Yes to Video
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%
%%%%

% Model Parameters
% From Callaway
% https://ac.els-cdn.com/S0196890408004780/1-s2.0-
S0196890408004780-main.pdf?_tid=4165d845-5c8b-4b7a-8c59-
ee93f2ba5105&acdnat=1552589032_382c3b81482263bf736457833c524332
R = 2;          % Thermal Resistance
C = 10;         % Thermal Capacity
Qc = -14.0;     % Cooling Capacity of AC
Q=14;          % Redundant Variable for AC
Beta = 2.5;     % COP of AC
Tset = 20.0;   % Homogeneous Set Point Temperature
deldb = 0.5;   % Deadband Size
Tinf= 32;      % Ambient Temperature
Qi=0;          % Internal Heat Gains

%Gain Value
k1=0.80        %Calculated Stability at k=1.0 for larger deadbands

% Population Size (Square Number needed for Square Lattice Graph)
L = 10;        % Setting up an LxL grid
N=L*L;        % Number of Houses

% Choose Graph Type
graphtype = 'S';

if graphtype == 'R'                                %Creates Ring Lattice
    (deg=4)
    deg=4;
    LAdj=AdjMatrixRingLatticeGen(N,deg);

```

```

elseif graphtype == 'D'                                %Creates Square Lattice
    (deg=8)
        LAdj=AdjMatrixLatticeSqGen8(L,true);
        deg = 8;
else                                                    %Default Square Lattice
    (deg=4)
        LAdj=AdjMatrixLatticeSqGen(L,true);
        deg = 4;
end

% Simulation times
delt = 1/60;                                           % integrate set size (hr)
nhours = 12;                                          % how long to run
DRHour = 2;                                           % time when the DR event
    happens
DRDuration = 0.25;                                    % duration in Hours of DR
    event
DRDuration = round(DRDuration/delt);                 % duration in minutes of DR
    event
%
nsteps = round(nhours/delt);
stepDRStart = round(DRHour/nhours * nsteps);
stepDREnd = stepDRStart + DRDuration;
stepSWEnd = stepDRStart + 8*DRDuration;
stepSWEnd = 5000;                                    %supress swarm end this way

% Create Exponential Constant used in Discrete Space
hrc=-delt/(R*C);

% Initialize Simulation Metrics
cycles=zeros(N,1);                                    % Number of Cycles Per Home
m=zeros(N,1);                                         % HVAC Operating State
m_last=zeros(N,1);                                    % HVAC last operating State
rho=zeros(N,1);                                       % Normalized Position within
    DB

% Create Initial Conditions
Non = round(0.428*N);                                 % 42.8% for initial stability
m(1:Non) = 1;                                         % 42.8% of the devices are ON
T(1:N,1)=Tset - (delldb/2);                          % Initialize Indoor
    Temperature
dt0ON = delldb/(Non-1);                               % Incremental Temperature
    Change
dt0OFF = delldb/(N-Non-1);                           % Incremental Temperature
    Change

% Create unifom distributions of intial Temerature
for i=1:Non
    T(i,1)= Tset - (delldb/2) + dt0ON*(i-1);         %Distribution for
    On
end
for i = Non+1:N
    T(i,1)= Tset - (delldb/2) + dt0OFF*(i-Non-1);   %Distribution for
    Off

```

```

end

% Initialize Model Outputs and Average Values
P=zeros(N,1); % Power (kW)
P(:,1)=m*Q/Beta; % Power for Initial States
time=zeros(nsteps,1); % Simulation Time
mtilde = (LAdj * m)/deg; % Average Operating State
rhotilde = (LAdj * rho)/deg; % Average Position within DB

% Create Frames and Video for House States
if vid == 1
    figure
    hold on
    for j=1:N
        hold on
        xpos=floor((j-1)/10)+1;
        ypos=rem(j,10);
        if ypos == 0
            ypos = 10;
        end

        % Red Markers are Off & Blue Markers are On
        if m(j) == 0
            p=plot(xpos,ypos,'or','MarkerFaceColor','red');
        else
            p=plot(xpos,ypos,'ob','MarkerFaceColor','blue');
        end
    end

    name=[num2str(0) ' hrs and ' num2str(0) ' min'];
    title(name)
    hold off
    drawnow
    % Set up Video Frames
    F(nsteps) = struct('cdata',[],'colormap',[]);
    F(1)=getframe(gcf);

    hold on
end

% Start Simulation
for k = 1:nsteps-1
    for j = 1:N
        % Compute Temperature for individual TCLs
        T(j,k+1)=exp(hrc)*T(j,k)+(1-exp(hrc))*Tinf+R*(1-exp(hrc))*Qi
        +(R*Qc)*(1-exp(hrc))*m(j);
    end

    % Update Simulations Values
    rho(:,k+1) = makerhovec(T(:,k+1),Tset,delddb);
    time(k+1)=k*delt;
    mtilde(:,k+1) = (LAdj * m)/deg;
    rhotilde(:,k+1) = (LAdj * rho(:,k+1))/deg;
    m_last(:)=m(:);

```

```

mout(k,:)=m(:);

% Thermostat Criteria for Individual Homes
for j = 1:N
    % Simple Thermostat Model
    if k < stepDRStart-1
        if rho(j,k+1) > 1
            m(j) = 1;
        elseif rho(j,k+1) < 0
            m(j) = 0;
        end
    end
end

% Additional Criteria
if k > (stepDRStart-1) && k < (stepSWEnd-1)
    if (rho(j,k+1) - k1*mtilde(j,k)) > 1
        m(j) = 1;
    elseif (rho(j,k+1) - k1*mtilde(j,k)) <= 0
        m(j) = 0;
    end
    % Count Number of Cycles after DR Event
    if (m(j)-m_last(j)) == 1
        cycles(j)=cycles(j)+1;
    end
end

end

% DR Event
if k>(stepDRStart-1) && k < (stepDREnd-1)
    m(j)=0;
end

% Individual Power Consumption
P(j,k+1)=m(j)*Q/Beta;

end

% Update Video Frames
if vid == 1
    for j=1:N

        xpos=floor((j-1)/10)+1;
        ypos=rem(j,10);
        if ypos == 0
            ypos = 10;
        end

        % Only change plot if states change (Much Faster)
        if (m_last(j)-m(j)) == 1
            plot(xpos,ypos,'or','MarkerFaceColor','red')
        elseif (m(j)-m_last(j)) == 1
            plot(xpos,ypos,'ob','MarkerFaceColor','blue')
        end
    end
    if k>stepDRStart && k < stepDREnd

```

```

        plot(xpos,ypos,'or','MarkerFaceColor','red')
    end
end

    pause(0.05) %Pause to Allow viewer to see times
    drawnow
    name=[num2str(floor((1+k)/60)) ' hrs and '
num2str(rem(k,60)) ' min'];
    title(name)
    F(k+1)=getframe(gcf);
end
end

% Save Video using popup window
if vid==1
    hold off
    MakeAVI
end

% Sum Aggregate Power
pout=sum(P);

% Plot Aggregate Power vs Time
figure
area(time,pout)
ylabel('Power')
xlabel('Time')
title('Discrete')

% Plot Individual Temperature Trajectories
figure
plot(time,T')
ylabel('Temperature')
xlabel('Time')
low_temp=round(min(T(:,stepDRStart:end')),2);
[lt,bin]=hist(low_temp,unique(low_temp));
[~,idx]=sort(lt,'descend');
temp_inst=lt(idx)
temp_values=bin(idx)

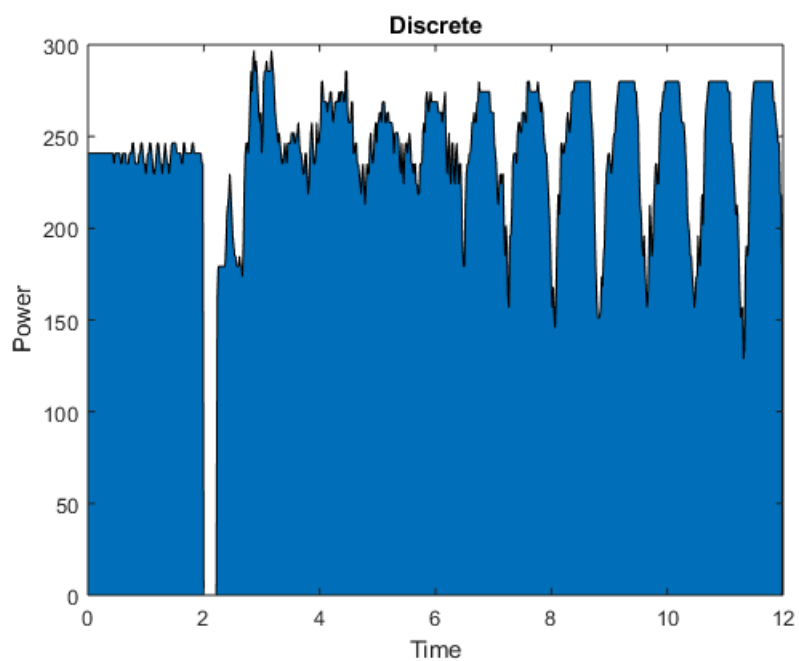
% %Below is some an attempt to find and plot an effective deadband
% eff_db=mean(min(T(:,stepDRStart:end)))
% hold on
% reffline(0,temp_values(2))
% hold off

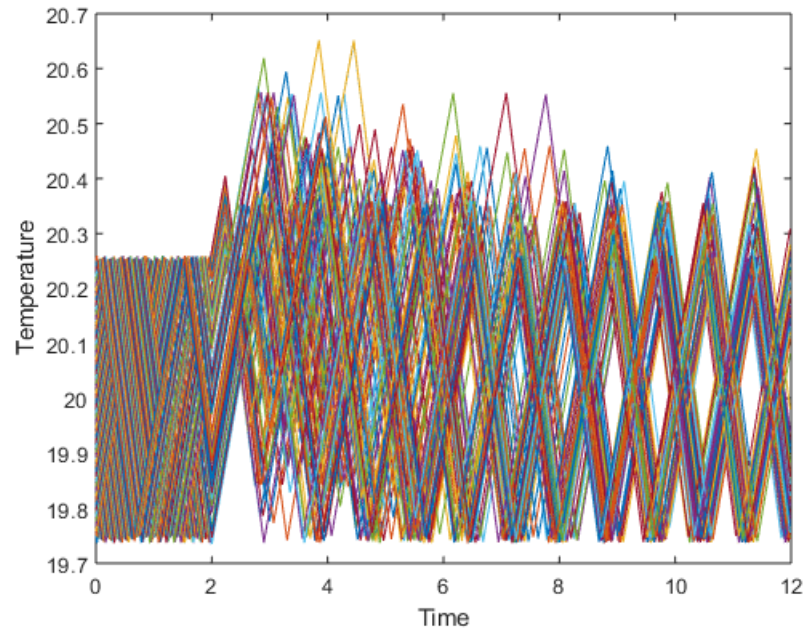
% Display Preformance Metrics
num_cycles=mean(cycles)
max_pow=max(pout)
devation=mean(std(rho))

k1 =
    0.8000
temp_inst =

```

```
95      4      1
temp_values =
  19.7400  19.7500  19.7300
num_cycles =
  8.1500
max_pow =
  296.8000
deviation =
  0.3222
```





Published with MATLAB® R2018a

C.7 Heterogeneous Parameters

Table of Contents

Creating Heterogenous parameters	1
File Naming Structure	1
Defining Means and StdDev	1
Creating Random Arrays	1
Write Data To File	2

Creating Heterogenous parameters

```
clear all
```

File Naming Structure

```
L=5;
N=L*L
name='HeteroParam_test';
filetype='.xlsx';
name=strcat(name,num2str(N),filetype);
Sheet=1;
```

```
N =
```

```
25
```

Defining Means and StdDev

```
R_m=2;      R_s=0.1;      %C/kW
C_m=10;     C_s=0.5;     %kWh/C
P=14;      %kW
Beta_m=2.5; Beta_s=0.125;
T0_m=20;   T0_s=1;      %C
Tinf=32;   %C
Theta_m=0.5;      Theta_s=0.025; %C
% 43% of n home On
m=zeros(N,1);
Non = round(0.428*N);
m(1:Non) = 1; % 42.8% of the devices are ON
m=m(randperm(length(m))); %Randomize location of beginning state
```

Creating Random Arrays

Using Normal Distribution

```
R=normrnd(R_m,R_s,[N,1]);
C=normrnd(C_m,C_s,[N,1]);
Beta=normrnd(Beta_m,Beta_s,[N,1]);
```

```
deldb=normrnd(Theta_m,Theta_s,[N,1]);  
T0=normrnd(T0_m,T0_s,[N,1]);
```

Write Data To File

```
xlswrite(name,'R',Sheet,'A1')  
xlswrite(name,'C',Sheet,'B1')  
xlswrite(name,{'Beta'},Sheet,'C1')  
xlswrite(name,{'Deldb'},Sheet,'D1')  
xlswrite(name,{'T0'},Sheet,'E1')  
xlswrite(name,{'m'},Sheet,'F1')  
xlswrite(name,R,Sheet,'A2')  
xlswrite(name,C,Sheet,'B2')  
xlswrite(name,Beta,Sheet,'C2')  
xlswrite(name,deldb,Sheet,'D2')  
xlswrite(name,T0,Sheet,'E2')  
xlswrite(name,m,Sheet,'F2')
```

Published with MATLAB® R2018a

C.8 Heterogeneous Varying Ambient Temperature

```

clear all;
format compact

% Files needed to include in folder
% -----
% AdjMatrixRingLatticeGen.m
% AdjMatrixLatticeSqGen8.m
% AdjMatrixLatticeSqGen.m
% makerhovec.m           *Please Note This file differs for
    Hetero
% MakeAVI.m
% BoiseTemp_24.xlsx
% HeteroParam_'Size'.xlsx *Sizes are Squares from 25 to 2500
% -----

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%If video is desired
vid=0;           %[0] - No to video   [1] - Yes to Video
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

% Model Parameters Consistant Throughout Populatoin
Qc = -14.0;           % Cooling Capacity of AC
Q=14;                % Redundant variable
Tset = 20.0;         % Thermostat Set Point
Tinf= xlsread('BoiseTemp_24.xlsx'); % 24 Hours of Ambient Temp in
    Boise
Qi=0;                % Internal Heat Gain
k1=0.8;              % Gain Value

% Population Size (Square Number needed for Square Lattice Graph)
L=10;                % Setting up an LxL grid
N=L*L;                % Number of Houses

% Create String for File Name of Heterogenous Parameters
name='HeteroParam_';
filetype='.xlsx';
name=strcat(name,num2str(N),filetype);
Sheet=1;
param=xlsread(name); % Load excel file

% Model Heterogenous Paramters
% Standard Deviations and Means defined in Ryan's Thesis
R=param(:,1);        % Thermal Resistance
C=param(:,2);        % Thermal Capacitance
Beta=param(:,3);     % COP of AC Unit
deldb=param(:,4);    % Thermostat Deadband
T(:,1)=param(:,5);   % Initial Temperature
m=param(:,end);      % Initial State (42.8% On & Randomly
    Assigned)
%-----

```

```

% Choose Graph Type
graphtype = 'S';

if graphtype == 'R'                                %Creates Ring Lattice
    (deg=4)
    deg=4;
    LAdj=AdjMatrixRingLatticeGen(N,deg);
elseif graphtype == 'D'                            %Create Square Lattice
    (deg=8)
    LAdj=AdjMatrixLatticeSqGen8(L,true);
    deg = 8;
else                                                %Creates Square Lattice
    (deg=4)
    LAdj=AdjMatrixLatticeSqGen(L,true);
    deg = 4;
end

% Simulation times
delt = 1/60;                                       % integrate set size (hr)
nhours = 24;                                       % how long to run
DRHour = 17.36;                                    % time when the DR event
    happens | 10 minutes before max temp
DRDuration = 0.25;                                 % duration in Hours of DR
    event
DRDuration = round(DRDuration/delt);              % duration in minutes of DR
    event
%
nsteps = round(nhours/delt);
stepDRStart = round(DRHour/nhours * nsteps);
stepDREnd = stepDRStart + DRDuration;
stepSWEnd = stepDRStart + 8*DRDuration;
stepSWEnd = 5000;                                  % supress swarm end this way

% Create Exponential Constant used in Discrete Space
hrc=-delt./(R.*C);

% Initialize Simulation Metrics
cycles=zeros(N,1);                                 % Number of Cycles Per Home
m=zeros(N,1);                                       % HVAC Operating State
m_last=zeros(N,1);                                  % HVAC last operating State
rho=zeros(N,1);                                     % Normalized Position within
    DB

% Initialize Model Outputs and Average Values
P=zeros(N,1);                                       % Power (kW)
P(:,1)=m*Q./Beta;                                  % Power for Intial States
time=zeros(nsteps,1);                               % Simulation Time
mtilde = (LAdj * m)/deg;                           % Average Operating State
rhotilde = (LAdj * rho)/deg;                       % Average Position within DB

% Create Frames and Video for House States
if vid == 1

```

```

figure
hold on
for j=1:N
    hold on
    xpos=floor((j-1)/10)+1;
    ypos=rem(j,10);
    if ypos == 0
        ypos = 10;
    end

    % Red Markers are Off & Blue Markers are On
    if m(j) == 0
        p=plot(xpos,ypos,'or','MarkerFaceColor','red');
    else
        p=plot(xpos,ypos,'ob','MarkerFaceColor','blue');
    end
end
name=[num2str(0) ' hrs and ' num2str(0) ' min'];
title(name)
hold off
drawnow
% Set up Video Frames
F(nsteps) = struct('cdata',[],'colormap',[]);
F(1)=getframe(gcf);

hold on
end

tic
% Start Simulation
for k = 1:nsteps-1
    for j = 1:N
        % Compute Temperature for individual TCLs
        T(j,k+1)=exp(hrc(j))*T(j,k)+(1-exp(hrc(j)))*Tinf(k)+R(j)*(1-
expm(hrc(j))*Qi+(R(j)*Qc)*(1-exp(hrc(j)))*m(j);
    end

    % Update Simulations Values
    rho(:,k+1) = makerhovec(T(:,k+1),Tset,deltdb(:));
    time(k+1)=k*delt;
    mtilde(:,k+1) = (LAdj * m)/deg;
    rhotilde(:,k+1) = (LAdj * rho(:,k+1))/deg;
    m_last(:)=m(:);

    % Thermostat Criteria for Individual Homes
    for j=1:N
        % Simple Thermostat Model
        if k < stepDRstart-1
            if rho(j,k+1) > 1
                m(j) = 1;
            elseif rho(j,k+1) < 0
                m(j) = 0;
            end
        end
    end
end

```

```

% Additional Criteria
if k > (stepDRStart-1) && k < (stepSWEnd-1)
    if (rho(j,k+1) -k1*mtilde(j,k)) > 1
        m(j) = 1;
    elseif (rho(j,k+1) - k1*mtilde(j,k)) <= 0
        m(j) = 0;
    end
end

% Count Number of Cycles after DR Event
if (m(j)-m_last(j)) == 1
    cycles(j)=cycles(j)+1;
end

end

% DR Event
if k>stepDRStart && k < stepDREnd
    m(:)=0;
end

% Individual Power Consumption
P(j,k+1)=m(j)*Q/Beta(j);

end

% Update Video Frames
if vid == 1
    for j=1:N

        xpos=floor((j-1)/10)+1;
        ypos=rem(j,10);
        if ypos == 0
            ypos = 10;
        end

        % Only change plot if states change (Much Faster)
        if (m_last(j)-m(j)) == 1
            plot(xpos,ypos,'or','MarkerFaceColor','red')
        elseif (m(j)-m_last(j)) == 1
            plot(xpos,ypos,'ob','MarkerFaceColor','blue')
        end
        if k>stepDRStart && k < stepDREnd
            plot(xpos,ypos,'or','MarkerFaceColor','red')
        end
    end

    pause(0.05) %Pause to Allow viewer to see times
    drawnow
    name=[num2str(floor((1+k)/60)) ' hrs and '
num2str(rem(k,60)) ' min'];
    title(name)
    F(k+1)=getframe(gcf);
end

```

```

end
toc

% Save Video using popup window
if vid==1
    hold off
    MakeAVI
end

pout=sum(P);
% figure
% yyaxis left
% plot(time,pout)
% ylabel('Power')
% xlabel('Time')
% title('Discrete')
% hold on
% yyaxis right
% pp = plot(time,Tinf,'-r');
% legend('Power','Temperature','Location','northwest')
% hold off
%
% figure
% plot(time,T,'-')
% ylabel('Temperature')
% xlabel('Time')
% low_temp=round(min(T(:,stepDRStart:end)),2);
% [lt,bin]=hist(low_temp,unique(low_temp));
% [~,idx]=sort(lt,'descend');
% temp_inst=lt(idx)
% temp_values=bin(idx)
% eff_db=mean(min(T(:,stepDRStart:end)))

startDate = datenum(2000,7,21,0,0,0);
endDate = datenum(2000,7,21,24,0,0);
xData = linspace(startDate,endDate,length(Tinf));
%hold on
% yyaxis right
% p = plot(time,Tinf,'-r');
% hold off

MaxPower=max(pout(stepDRStart+1:end))
NumCycles=mean(cycles)
devation=mean(std(rho))

Elapsed time is 0.454670 seconds.
MaxPower =
    410.3702
NumCycles =
    6.4500
devation =
    0.5136

```

C.9 Deadband Analysis

```

m_tilde=[0 0.25 0.5 0.75 1.0];
k=0.0:0.5:5;
X=length(k);
Y=length(m_tilde);

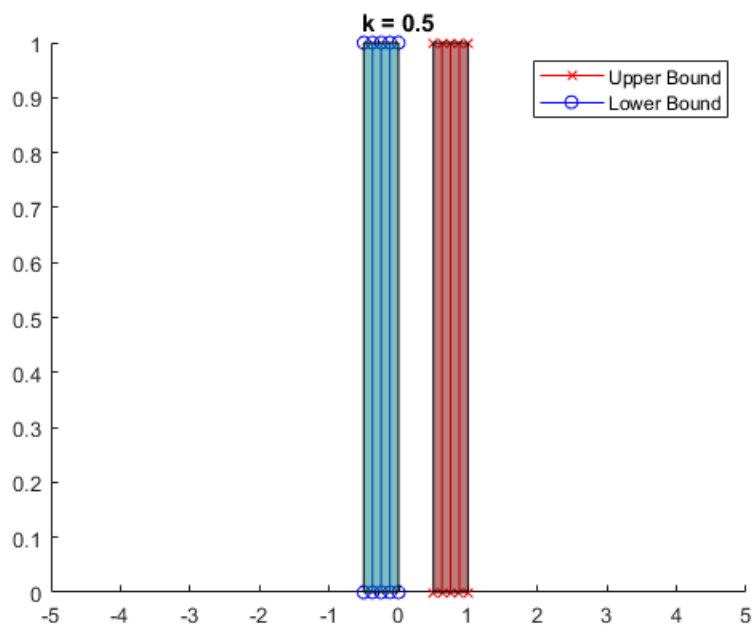
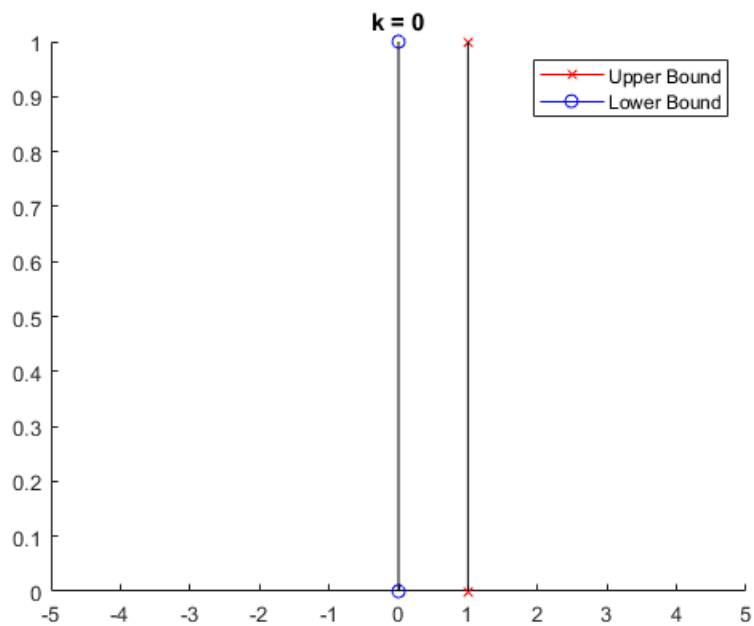
F(X) = struct('cdata',[],'colormap',[]);

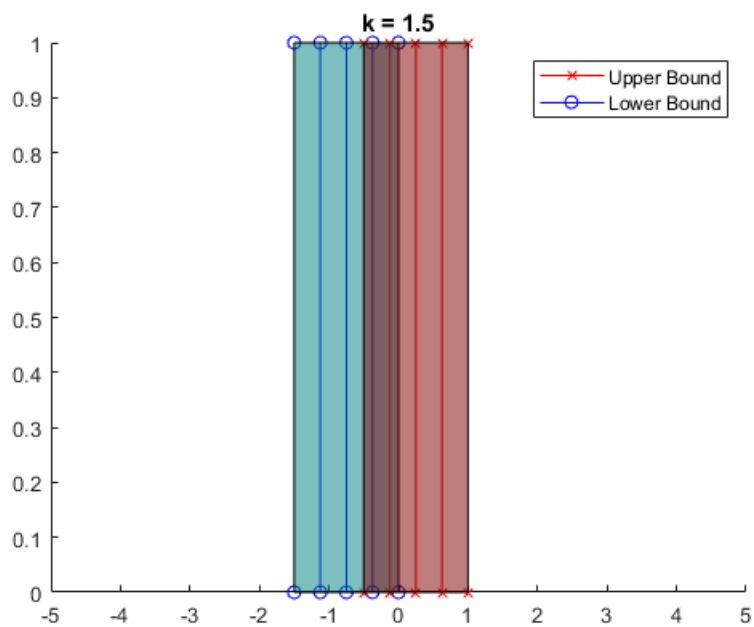
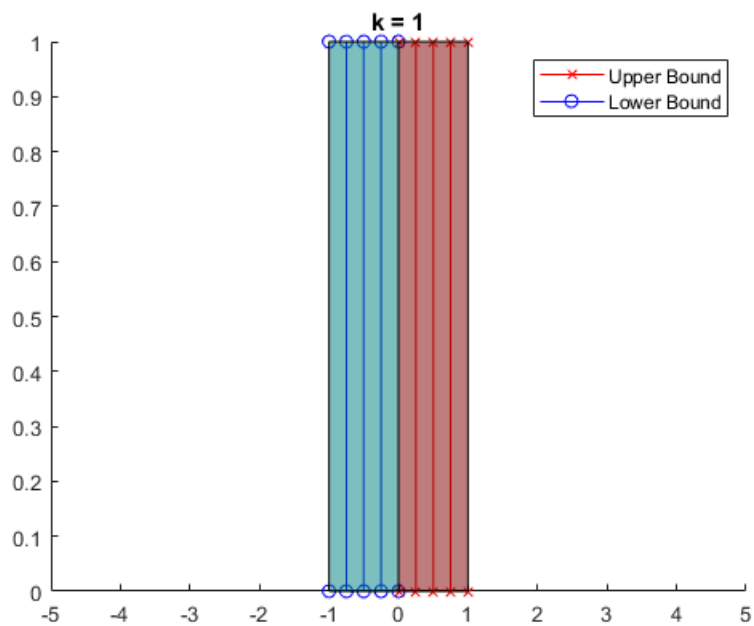
for i=1:X
    figure
    hold on
    for j=1:Y
        upperbound(i,j)=1-k(i)*m_tilde(j);
        lowerBound(i,j)=-k(i)*m_tilde(j);

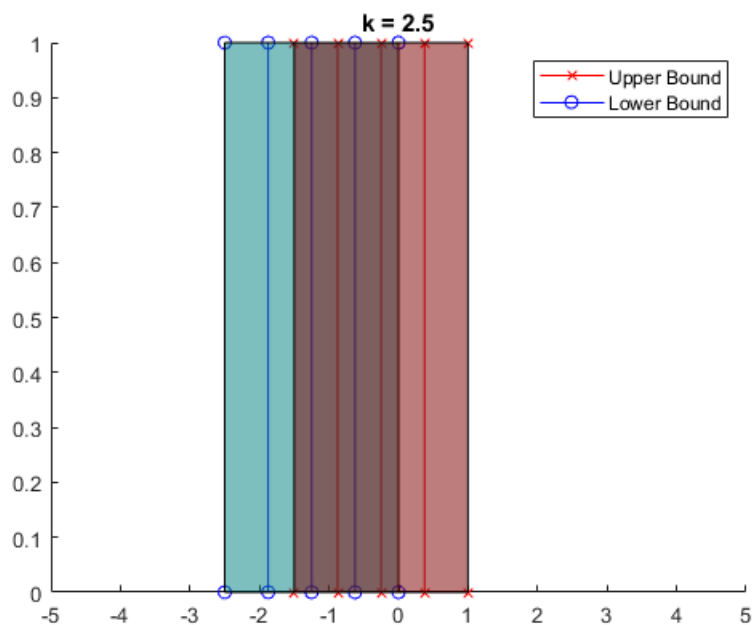
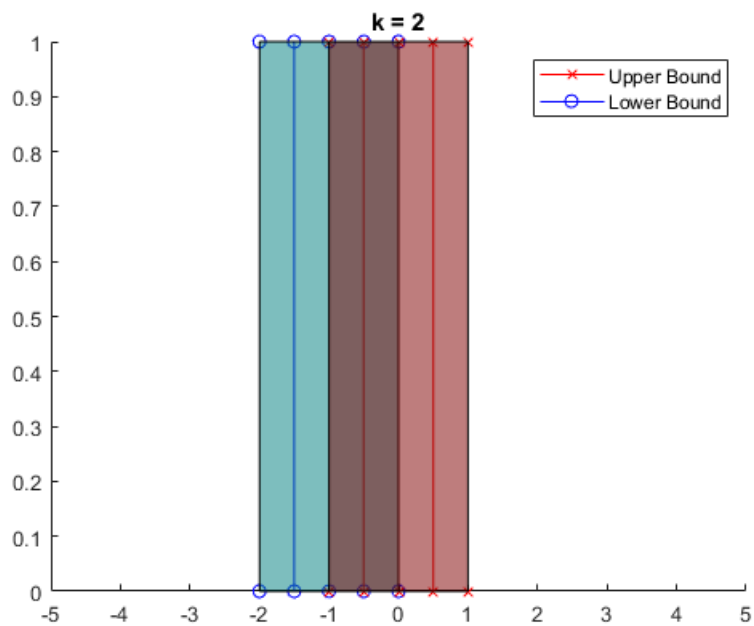
        up=[ upperbound(i,j) upperbound(i,j)];
        lp=[ lowerBound(i,j) lowerBound(i,j)];
        yy=[0 1];
        plot(up,yy,'r-x',lp,yy,'b-o')
        name=['k = ' num2str(k(i))];
        title(name);
        legend('Upper Bound','Lower Bound')
    end
    rectangle('Position',[lowerBound(i,end) 0 abs(lowerBound(i,1)-
lowerBound(i,end)) 1],'FaceColor',[0 0.5 0.5 0.5])
    rectangle('Position',[upperbound(i,end) 0 abs(upperbound(i,1)-
upperbound(i,end)) 1],'FaceColor',[0.5 0 0 0.5])
    axis([-5 5 0 1])
    hold off
    drawnow
    pause(5)
    F(i)=getframe(gcf);
end

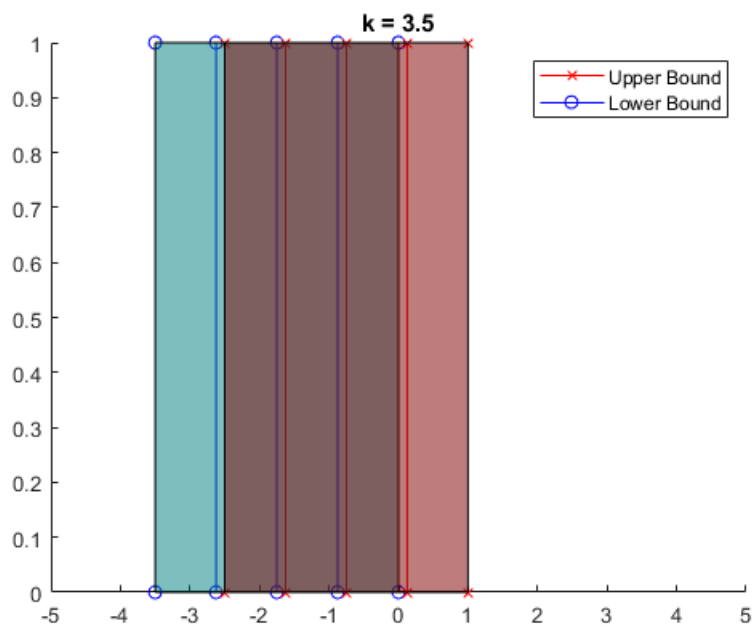
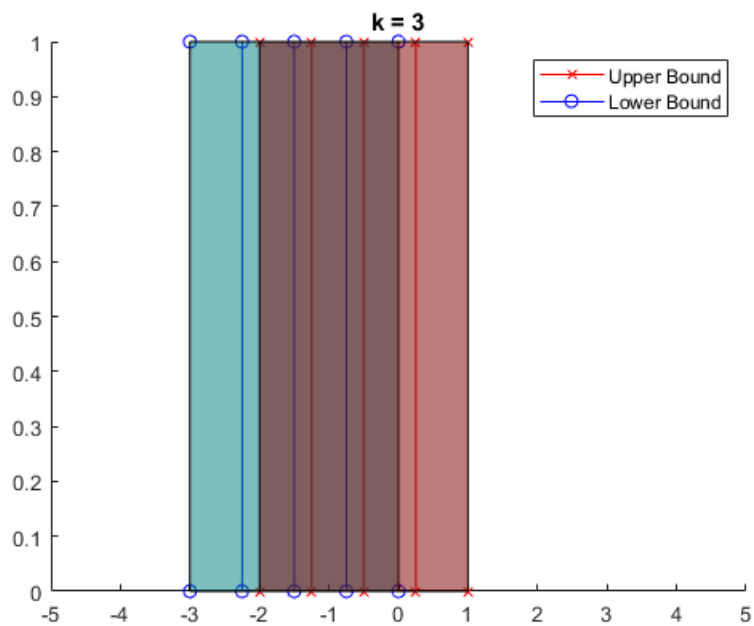
MakeAVI

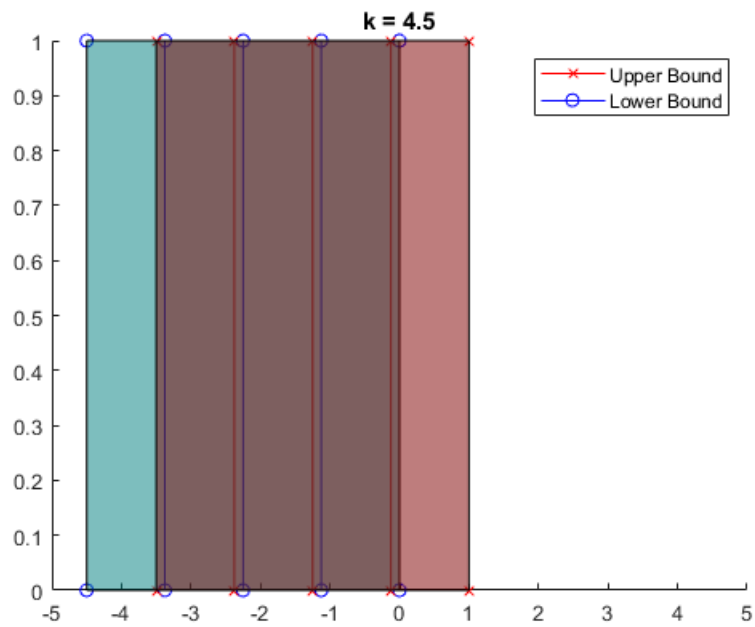
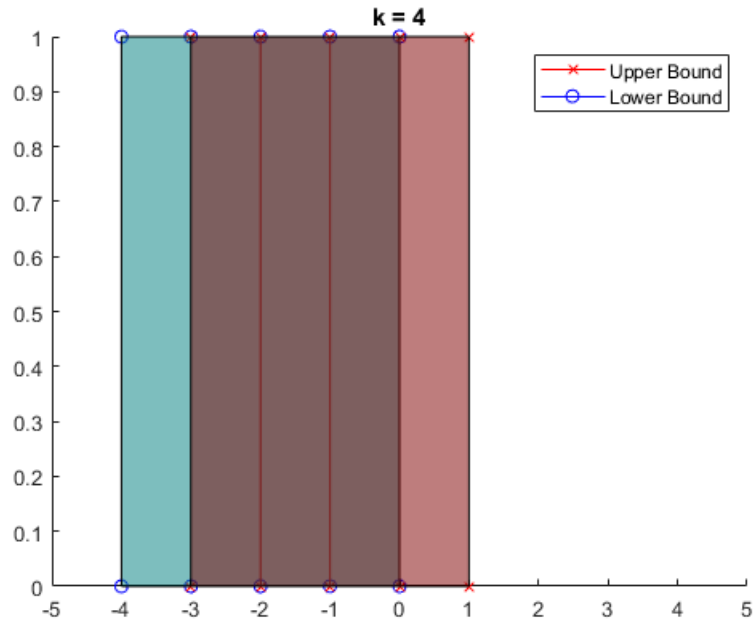
```

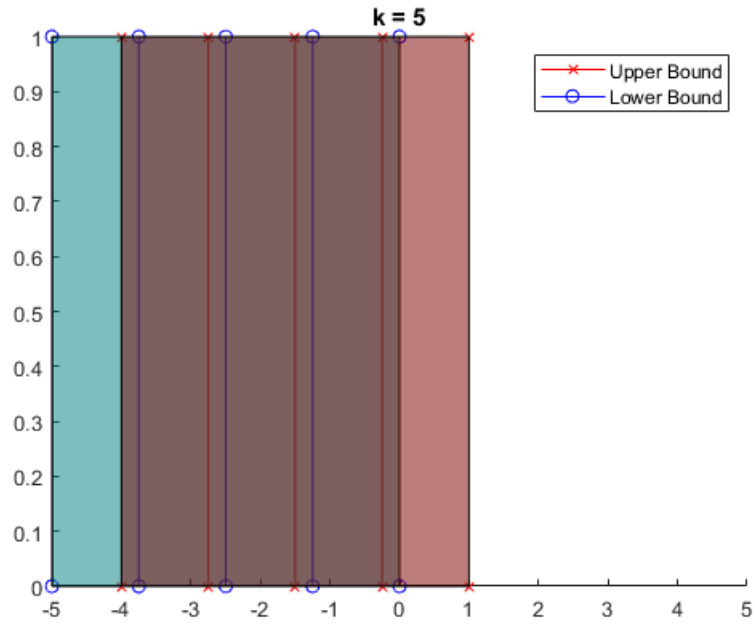













Published with MATLAB® R2018a