Machine Learning Based Simulation and Control Framework for Power Distribution Systems with Transactive Elements

by

Boming Liu

B.S., North China Electric Power University, 2014

M.S., University of Pittsburgh , 2016

Submitted to the Graduate Faculty of

the Swanson School of Engineering in partial fulfillment

of the requirements for the degree of

Doctor of Philosophy

University of Pittsburgh

2020

UNIVERSITY OF PITTSBURGH

SWANSON SCHOOL OF ENGINEERING

This dissertation was presented

by

Boming Liu

It was defended on

September 17, 2020

and approved by

Murat Akcakaya, Ph.D., Associate Professor, Department of Electrical and Computer Engineering

Thomas E. McDermott, Ph.D., Pacific Northwest National Laboratory

Brandon M. Grainger, Ph.D., Assistant Professor, Department of Electrical and Computer Engineering

Ervin Sejdic, Ph.D., Associate Professor, Department of Electrical and Computer Engineering

Natasa Miskov-Zivanov, Ph.D., Assistant Professor, Department of Electrical and Computer Engineering

Zhi-Hong Mao, Ph.D., Professor, Department of Electrical and Computer Engineering

Dissertation Directors: Murat Akcakaya, Ph.D., Associate Professor, Department of Electrical and Computer Engineering

Thomas E. McDermott, Ph.D., Pacific Northwest National Laboratory

Machine Learning Based Simulation and Control Framework for Power Distribution Systems with Transactive Elements

Boming Liu, PhD

University of Pittsburgh, 2020

Transactive Energy (TE) has been recognized as a promising combination of techniques for improving the efficiency of modern power grids through market-based transactive exchanges between energy producers and energy consumers. It is of significant interest to identify optimal strategy to control the transactive load in TE systems. The behaviors of transactive loads are affected by the energy market values which in return impact the operation and stability of the distribution system. To evaluate the benefits and impacts of transactive loads and new control mechanisms, time series simulations are commonly used. These simulations consider the pricing response and the physical constraints of the system simultaneously. Such simulations are computationally demanding due to the information exchange among various participants and the complex co-simulation environments.

This dissertation first explores the reduced order models to support quasi-static time-series (QSTS) simulations for power distribution systems with independent dynamic non-responsive load to address the limitations of the order reduction methods. Further, a reduced order model for transactive systems with responsive load is proposed. The proposed model consists of an aggregate responsive load (ARL) agent which utilizes two Recurrent Neural Networks (RNN) with Long Short-Term Memory units (LSTMs) to represent the transactive elements in TE systems. The developed ARL agent generates load behavior for transactive elements and interacts with the electricity market. In addition, for individual transactive elements, a control strategy for the residential Heating, Ventilation, and Air Conditioning (HVAC) is introduced through the solution of an optimization problem that balances between the energy cost and consumer's dissatisfaction. A reinforcement learning (RL) algorithm based on Deep Deterministic Policy Gradients (DDPG) is used to obtain the optimal control strategy for the HVAC systems. The reduced order model and the DDPG RL-based control are both implemented in the Transactive Energy Simulation Platform (TESP). The reduced order model is able to produce transactive behaviour very close to the full

simulation model while achieving significant simulation time reduction. Moreover, simulation results demonstrated that the proposed control method for HVACs reduces the energy cost and improves the customers' comfort simultaneously.

Table of Contents

List of Tables

List of Figures

List of Abbreviations

P_t	Bernoulli trial parameter
M_t	Load shape/behavior (a factor between 0 and 1 that changes with time t)
X_t	Appliance ON time distribution, after it switch from OFF to ON
$N_{t(on)}$	The ratio of appliances ON at time t in GridLAB-D output
$N_{t(off)}$	The ratio of appliances turning off just after time t in stochastic simulation
s_t^i	Status (ON, OFF) of appliance i at time t
P_{WH}	Heating element capacity of water heater (kW)
$V^0_{[in/out],i}$	No-load voltage of segment i (V)
$I^0_{[in/out],i}$	No-load current of segment i (A)
$V^n_{[in/out],i}$	Voltage of segment i when $S^{i\neq n} = 0$ (V)
$I^n_{[in/out],i}$	Current of segment i when $S^{i\neq n} = 0$ (A)
S^n	Loads that follow load shape n (kW)
$S^{i\neq n}$	Loads that follow other load shapes (kW)
$I^n_{c[in/out],i}$	Independent current sources on segment i for load shape n (A)
$G^n_{[in/out],i}$	Compensating power value of segment i for load shape n (kW)
$I^n_{c\ in,i}$,$I^n_{c\ out,i}$	Independent current sources to compensate the voltage drop in segment i due to load behavior of segment n (A)
$M_n(t)$	Independent load shape of the segment n at time t
N	The number of segments
n	The index of the load shape of a different segment
i	The index of a simplified segment in the feeder model, or of an appliance in the stochastic model
j	The index of a downstream segment in the feeder model
Z_i	Simplified equivalent impedance of segment i (Ω), shown as Z_{sim} in Fig. 8(b). The detailed calculation of Z_i is in [3].

$\Delta\bar{V}^i$	Voltage drop across segment i (V)
$P_{cleared}$	price at the last market clearing
$P_{average}$	the historical mean price
$\left\|T_{\max/\min}\right\|$	the allowed range of setpoint variation
$k_{high/low}$	bidding ramp denominator
T_{set}	the thermostat setpoint, including price response (F)

1.0 Introduction

This dissertation introduces reduced order models for the simulation of distribution systems with independent changing load and distribution systems with transactive elements. In addition, a reinforcement learning agent is developed and implemented in Transactive Energy Simulation Platform (TESP) to find to an optimal control strategy for HVAC systems that balances between electricity cost and user comfort by Deep Deterministic Policy Gradients (DDPG)-based deep reinforcement learning algorithm.

Section 1 includes an overview of the research and the background information on power systems and transactive energy. Section 2 introduces the reduced order model by segment substitution for distribution systems with independent changing loads. A method to stochastically model and employ the dynamic behavior of end-use load in QSTS simulations is also presented. Using the stochastically modeled load behavior as an input, the QSTS simulation result of reduced order model and full model are compared. Section 3 provides the reduced order model for transactive systems with responsive loads. The transactive behaviours of HVACs generated by the reduced order model are compared with the TESP full model simulation result. In section 4, a DDPG reinforcement learning based automated control of transactive HVACs in distribution systems is presented, the performance of the method are demonstrated through different simulation scenarios. Section 5 is the summary of the research.

1.1 Research Overview

Many distribution system technologies have time-varying characteristics. One of the tools to investigate the operational characteristics of a modeled distribution system over a period of time is called quasi-static time-series (QSTS) analysis [4]. QSTS analysis with high resolution (small time step) is required to capture the load variability and simulate control signals for the emerging technologies such as responsive load and plug-in electric vehicle charging [5]. However, long-term high-resolution QSTS simulations require a large number of power flow computations. A year-

long simulation at 1-second resolution could take a computational time of 10 to 120 hours using a desktop computer [6]. This may limit sensitivity analysis, design optimization, or exploration of alternatives on the distribution system.

Transactive energy (TE) is a new approach for managing the electric power distribution system based on principles of economic value, with engineering constraints [2]. It is a natural evolution of market deregulation, which resulted in wholesale markets for the big players. Transactive energy is a way of bringing competitive market principles to many more electricity customers and potential participants. With many more new types of participants including end-use load and distributed energy resources in future grids. There are higher requirements for the modeling of end-use load: buildings, appliance to support transactive energy simulations and many valuation metrics need to be predicted by time-series simulation in order to meld physical systems with economic value. At present, most studies mainly focus on the design and analysis of the hierarchy architecture of transactive control schemes. (Transactive control has also been applied in the distributed dispatch or control systems of responsive assets in the grid.) However, the detailed control method of the end-users in the transactive systems needs to be further explored [7].

Simulation for the transactive system is becoming increasingly important in smart grid studies. The modeling of price responsive load in the transactive system helps to evaluate different transactive energy mechanisms and new control strategies. Software tools such as GridLAB-D have detailed physical models to represent end-use behavior loads in a house, e.g. water heaters, heating and cooling systems [8]. Each house has its individual thermal dynamic equivalent model, requiring lengthy computation for large numbers of houses. In the Transactive Energy Simulation Platform (TESP), these houses also have to participate in the electricity market by submitting bids for the transactive elements such as HVACs. Simulations in TESP could take several days for a large system with thousands of houses due to the various participants, different transactive agents, large numbers of individual models, and the complex co-simulation environment.

Accurate and efficient modeling of the end-use load in distribution systems is important for studying the aggregate dynamic behaviours and their impact on the bulk system. The simulation time can be notably reduced by an aggregate model with the collective bidding and responding ability. Therefore, accurate aggregation of large numbers of end-use loads is needed to account for behaviours of the transactive elements and their interactions with the electricity market. To

achieve faster simulations, it is well worth studying the error characteristics of reduced-order models to ensure that simplifications and aggregations are appropriate and represent the system model precisely enough for certain applications and studies. Some studies have focused on the aggregation of thermostatically controlled loads (e.g. HVACs) since they can actively participate in demand response programs. Aggregated dynamic models are developed in [9] for thermostatic loads using stochastic diffusion models by Fokker-Planck partial differential equations (PDEs). An aggregate controllable model for a homogeneous population of HVACs and water heaters considering the state transitions is developed in [10]. Data driven approaches based on Markov chains are proposed in [11] [12], these methods compute the transition probability between different states based on equivalent thermal parameter (ETP) model. The above aggregate models are developed for steady state conditions, not designed for demand response. Researchers have been study on the aggregation of price responsive loads as well. In [13], the author proposed a method to aggregate a group of price responsive loads through a non-parametric analysis of experimental random scenarios to maximize the profit for the retailer/aggregator in a day-ahead market. In [14], an aggregate model for a diverse group of thermostatically controlled loads is proposed, the aggregate model is able to accurately capture the transient dynamics in the collective response under both steady state and severe dynamic conditions. Different from the aforementioned approaches, the reduced order model in TESP needs to have bidding and responding functions to interact with different market participants, thereby produce similar behavior for the transactive elements to ensure the validity of representing the transactive mechanisms of the full model. Through the reduced order mode in TESP, we're trying to simulate an entire balancing authority or region, such as ERCOT (Electric Reliability Council of Texas) or WECC (Western Electricity Coordinating Council), in order to show the effect of major policy changes and restructurings.

In this dissertation, a reduced order models for the simulation of distribution systems with independent changing load is introduced. Moreover, for the evaluation of top-level design approaches and the impacts of responsive loads on the bulk system, a reduced order model for distribution systems with transactive elements is developed for the aggregation of price responsive loads. The reduced order model is able to generate similar transactive behaviors as the full model by formulating bids and reacting to the market clearing price collectively with an aggregate responsive load. The proposed method dramatically reduces the simulation time for distribution feeder models with

larger number of houses in TESP. In addition, for detailed control methods of end-users in the transactive systems, an RL-based HVAC control method is also presented in this dissertation. The RL-based control approach is implemented in and tested through the TESP. Through multiple experiments, the effect of control algorithm parameter selection is demonstrated, and the proposed method is compared with the transactive HVAC controller that is currently implemented in TESP.

1.2 Power System and Transactive Energy

This section contains the background information for the studies in the dissertation.

1.2.1 Power System

This section shows a high-level background about power systems which have some basic information considered to be known that may appear in the rest of the proposal. The composition of the power system and the function of each part is included in this section.

Power systems were designed to deliver power from large power plants to load areas. It is a network of electrical components to supply, transfer, store and use electric power [15]. The systems can be divided into generation, transmission, sub-transmission and distribution. The transmission system connects generating stations to substations and the sub-transmission system is responsible for distributing powers to a specific district. Fig. 1 shows a diagram of a power system that contains key components.

1.2.1.1 Generation

There are several methods of generating electricity. Generally, it is more efficient to generate power at large power plants and distribute power to individual load sites. There also exist electric power generation within distribution networks or on the customer side of the network which is called distributed generations (DG) [16].

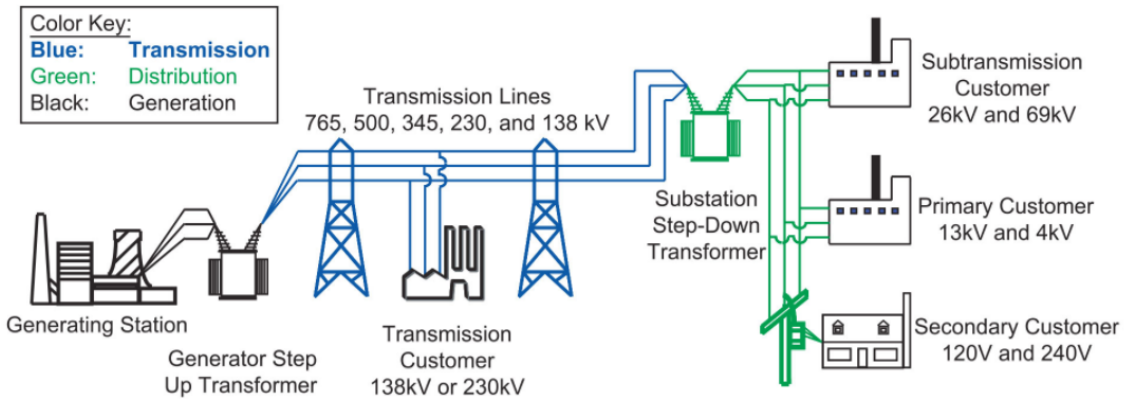


Figure 1: Basic structure of the power system [1]

1.2.1.2 Transmission

The transmission system connects generations and substations at high voltage level in order to reduce line losses. Transmission systems deliver bulk power over long distances, but only a few very large customers are directly connected to transmission. The power going through the transmission lines must be reduced to lower voltage levels by electricity distributors before it can be delivered to a residence or business. So most of the transactive energy simulations and control framework focus on distribution systems, which deliver electricity to more customers who are potential new market participants.

1.2.1.3 Distribution

At distribution substations, transformers reduce the voltage to a medium-voltage level (1 kV to 35 kV) and supply one or more feeders. A feeder is a radial or meshed topology that connects substation and customer loads which is the last link of the power delivery, usually serving several hundred residential consumers.

Modeling of power distribution systems can be used to understand distribution system behavior without monitoring. They can also be used to study transients or other system behaviors. Some examples are: Infrastructure upgrades [17], distribution energy integration: energy storage, plug-in electric vehicle (PEV) applications, solar or other distributed generation resources [18–21],switching transients [22], lightning transients [23], harmonics [24, 25].

There exist different kinds of simulations such as transient simulations and load-flow simulations for different purposes. Transient simulations are performed on the order of micro-seconds to seconds in duration to study corresponding events. Load-flow calculations can be used to study the steady-state behavior of a system or system stability. Harmonics can be studied by performing steady-state load-flow calculations over a range of system frequencies.

1.2.1.4 Utilization

Utilization is the final result of the generation, transmission and distribution. The power then will be turned into light, heat, or some other useful work. Utilization also refers to circuits behind the meter, i.e., inside the residence or place of business. Generally, utilization circuits are not modeled in detail, only as a collection of loads i.e., residential appliances, behind the meter. These circuits are the customer's responsibility, not the utilities. To ensure appropriate planning and operation of the power system, understanding and characterizing the utilization of electric power is important.

1.2.1.5 Demand Response

The power system could become stressed when there is high demand of electricity. Demand response (DR) provides a way for end-use consumers to reduce or shift their electricity usage during peak hours in response to time-based electricity price or other incentives [26]. There are multiple benefits associated with DR. Consumers are able able to reduce the electricity bills by participating in DR programs. A market-wide electricity price reduction is expected by cutting the demand from much expensive electricity generating units. DR programs also help to reduce the risk of outages, reduce price volatility and improve system reliability [27]. An example of direct local control program to release the electric supply pressure is the customers give permission to power companies to control the on and off of air conditioners and water heaters during rush hours in exchange for financial incentives. With advanced information and communications technologies, household consumers now have the potential to participate in demand response more actively. The consumers may adjust their demand in numerous ways to provide the power grid with more flexibility: (1) reduce consumption, (2) increase consumption, (3) shift load to off-peak periods, (4) store electricity, (5) generating electricity in small generation units (self-supply).

Many studies have emphasized the importance of time-based dynamic pricing at the distribution level to motivate consumers to participate in DR programs [28].

1.2.2 Transactive Energy

With large scale deployment of distributed energy resources (DERs) and high penetration level ofrenewable energy sources (RES) in future smart grid, new control method for the power system operations are in demand so that the responsive assets of different participants can be well investigated [7]. The Gridwise Architecture Council (GWAC) defines transactive energy (TE) as "a set of economic and control mechanisms that allows the dynamic balance of supply and demand across the entire electrical infrastructure using value as a key operational parameter." TE also refers to the economic and control techniques used to manage the exchange of energy within an electric power system in regards to economic and market based standard values of energy [29]. It is a concept that is used in an effort to improve the efficiency and reliability of the power system, heading towards a more intelligent and interactive future grid. A TE system uses principles of value to coordinate responsive supply and demand in energy systems. In transactive energy systems, price signals are passed throughout the system as communication bridges between different prosumers, enabling the optimal utilization of energy resources by distributed decision making of prosumers. In the way, different participants perform individually and interacting in near real time to achieve system objectives. An example of an application of a transactive energy technique is the double auction market used to control responsive demand side assets in the GridWise Olympic Peninsula Project [30].

Existing TE mechanisms have been reported in the literature such as double auction [31], TeMix [32], and PowerMatcher [33]. For different parts of power systems described in section 1.2.1, market mechanisms are already used on the transmission level to make decisions for system operation in modern power systems. For the distribution level of the power systems, as the distributed energy resources (DER) become mature, it's necessary to find appropriate market mechanisms for variable DER energy exports. On the other hand, there are still uncertainties and challenges in the coordination of the increasing number of intelligent devices for a resilient and efficient system. These devices have different objectives and different measurement of value. TE

concepts are introduced into the distribution level to balances the trade-offs among participants with different objectives while considering the available changing resources over time [29]. The operation of flexible devices or transactive elements are controlled by individual local controllers (agents), these controllers optimize the operation economically based on market value with the preference of end users. Different from the simple price reactive approach which is an one way communication sending dynamic price signals to end users, local controllers of the consuming device communicate their willingness to pay by submitting bids based on market conditions and the preference of end users. Through this approach, prices can be determined collaboratively from the predictable reactions of all participates to balance the supply and demand. The approach makes good use of the response potential of flexible devices and protects the end user's privacy since only prices and energy quantities are used for communication.

A TE system with transactive control and coordination has advantages in integrating different transactive elements such as smart homes, buildings in the market at the distribution system level, it makes most use of the response potential. Different participates engage into the market based on energy quantities and price in two-way communications. Some other advantages of TE can be summarized as follows:

1. Lower costs during peak demand conditions
2. Improve the resilience and reliability, reduce the length and frequency of outage
3. Increase the flexibility of personal energy use with more information
4. Increased use of cost-effective, renewable energy generation, enable the optimal integration of RES and DERs in the distribution systems.
5. Reduce the need for building new power plants

Specific examples of transactive control applications:

1. Some appliances such as HVACs, dishwashers, dryers, water heaters which use a large amount of electricity. These appliances can be programmed to be on and off at different time of the day based on the consumers' desire for the balance between convenience and cost.

2. Electric vehicle batteries can be charged when the electricity market price is the least expensive. The charging can be coordinated with the availability of renewable energy resources (charging on a windy day or a sunny day). Besides, the charged batteries can sell power back to the system when the market price is high under certain conditions.

1.2.3 Transactive Energy Simulation Platform

A simulation environment TESP (Transactive Energy Simulation Platform) has been developed by Pacific Northwest National Laboratory (PNNL) to simulate alternative transactive energy mechanisms, to co-simulate effects across distribution and transmission, and to allow interplay between existing wholesale market mechanisms and new transactive distribution systems [34]. The Transactive Energy Simulation Platform (TESP) incorporates time-series electrical simulations of the distribution grid with agent-based market simulations and transactive energy valuations.

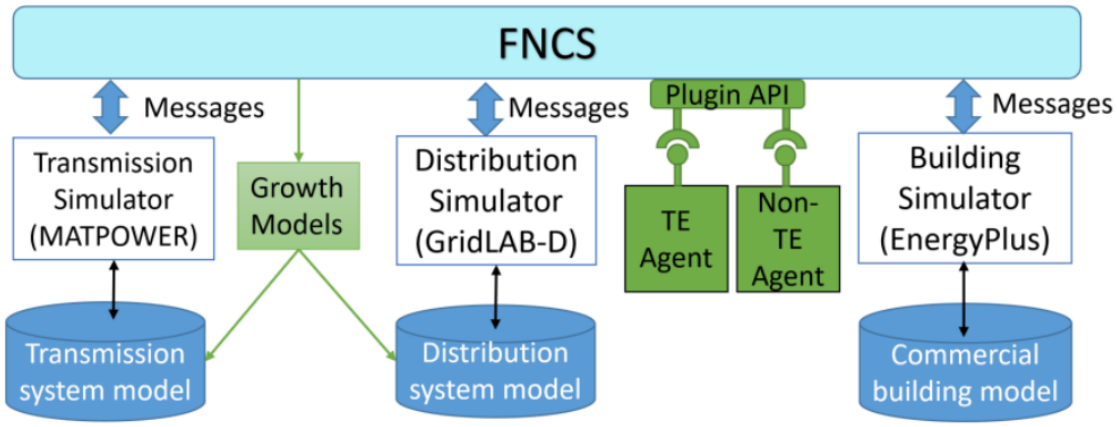


Figure 2: TESP architecture [2]

Figure 2 shows the simulation architecture of TESP. Distribution simulator GridLAB-D [8] covers the simulation of electric power distribution circuits and weather. Transmission simulator MATPOWER or PYPOWER covers the bulk power generation and transmission [35]. Building simulator EnergyPlus represents large commercial buildings [36], which were not addressed in this dissertation. TESP uses these three simulators together in a transactive simulation. The in-

tegrating Framework for Network Co-simulation (FNCS) manages the time step synchronization and message exchange among all of the federated simulation modules [37]. TESP also provides plugin application program interfaces (APIs) which allow users to develop different modularized software control, bidding and market clearing agents.

Some TE agents such as double-auction market, dual-ramp thermostat controller were previously hard coded in GridLAB-D in C/C++ [31]. TESP enables the design and test of new TE mechanisms with separate TE agents implemented in Python, Java or C/C++. One can modify the TE agent code in without rebuilding all of GridLAB-D. Currently, TESP includes TE agents for residential customers and distributed energy resource (DER) owners. Examples of some TE agents are described in the following section.

1. Double Auction TE agent: A double-auction market agent in TESP is a two-way market. It collects bids from both suppliers and demanders (sellers and buyers), including price and quantity simultaneously [31]. The double auction TE agent aggregates the bids and publishes a market cleared price and sends the information back to all the participants.
2. HVAC controller TE agent: With the HVAC controller TE agent, the consumers are able to set the desired temperature and an acceptable temperature range. The HVAC controller agent will formulate bids based on a user-defined ramp parameter associated with the trade-off between cost and comfort at every market cycle. After each market clearing, the HVAC controllers use that subscribed clearing price, compared to their bid price and adjust the HVAC setpoint accordingly.

The sequence of interactions between different simulators and controller agents in TESP are shown in Fig. 3. Messages are communicated via FNCS between different simulators and TE agents. A Substation agent encapsulates the above-mentioned HVAC controllers and the double auction market. Δt is the FNCS time step and T is the time step for market clearing and optimal power flow by PYPOWER. We envision real-time markets clearing at 5 to 15 minutes intervals, along with hourly day-ahead markets. These time frames are similar to some existing wholesale markets on bulk systems. So, there is still a need for ramping reserves and frequency control, which could also be services offered in the market, but not in the scope of this dissertation.

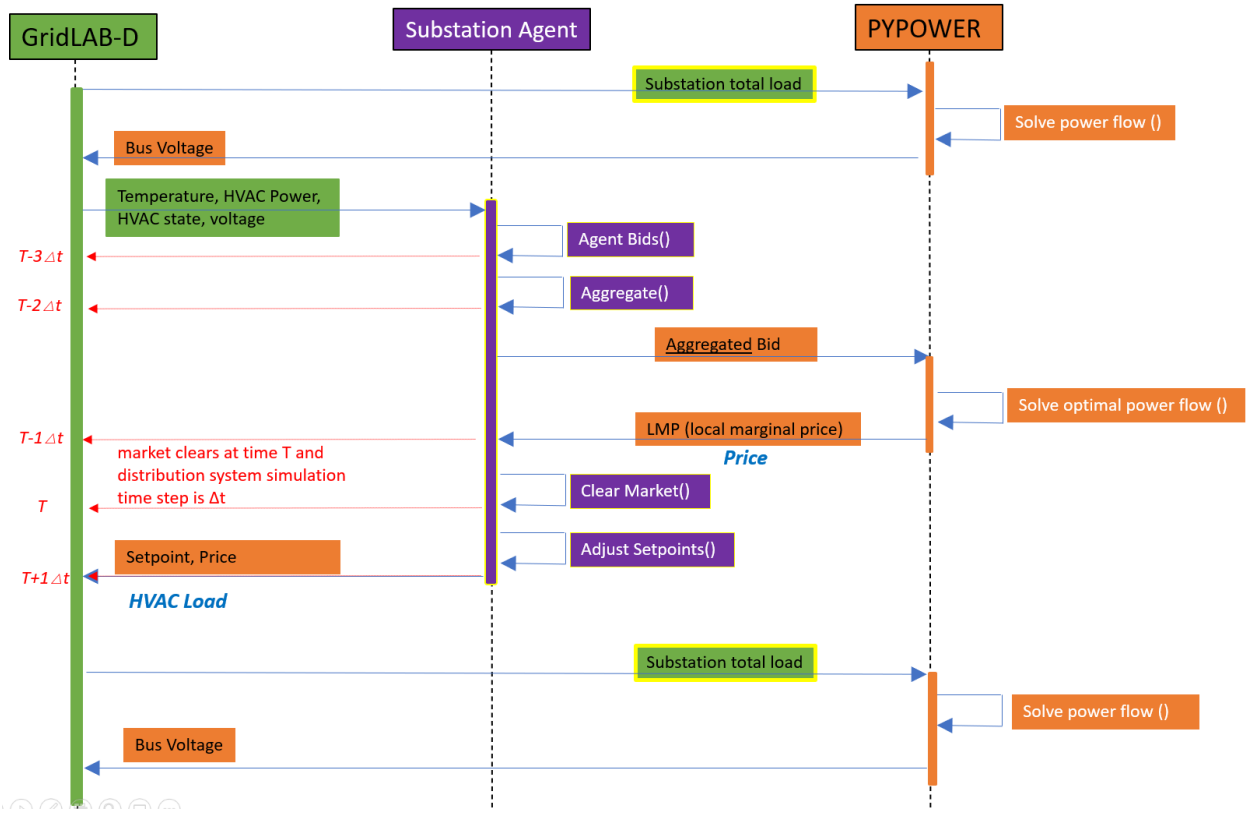


Figure 3: Message flow around every market clearing cycle

2.0 Model Order Reduction for Distribution Systems with Independent Varying Loads

2.1 Motivation and Related Work

Today, many distribution system technologies have time-varying characteristics. One of the tools to investigate the operational characteristics of a modeled distribution system over a period of time is called quasi-static time-series (QSTS) analysis [4]. QSTS analysis at small time step is required to capture the load variability and simulate control signals for the emerging technologies such as responsive load and plug-in electric vehicle charging [5]. However, long-term high-resolution QSTS simulations require a large number of power flow computations. A year-long simulation at 1-second resolution could take a computational time of 10 to 120 hours using a desktop computer [6]. This may limit sensitivity analysis, design optimization, or exploration of alternatives on the distribution system. To achieve faster simulations, it is well worth studying the error characteristics of reduced-order models to ensure that simplifications are appropriate and represent the system precisely enough for a certain application.

This model order reduction work was motivated by the needs of two parallel research projects, to be reported in detail later.

1. Regional Transactive System Study: The study is set in Texas, with a synthetic grid comprising up to 200 substations and 800 feeders. To develop and test new market and control mechanisms over a period of one year, we needed to reduce the number of distribution system nodes retained in the model, while keeping the same number of loads. The total simulation times have been shorter by a factor of 3, saving approximately 10 computing days per simulated month on this large system.
2. Island Resilience with Storage Study: The distribution system comprises 8 feeders with a utility-scale battery, some solar resources and a combustion turbine. When disconnected from the mainland, critical loads cannot be supplied indefinitely without new control methods. To develop and test these methods, we needed to simplify non-responsive parts of the system with reasonable accuracy, while retaining all of the independent and dynamic agents.

2.1.1 Related Work - Load Modeling

To support QSTS simulations, high-resolution temporal and spatial input data are required, including the end-use load behaviors. Accurate representation of the end-use load behaviors can help to completely represent the behavior of distribution systems including state transitions, to assess the influence of the increasing smart grid technologies and transactive control techniques, and to optimize the integration of renewable energy sources (RES) and distributed energy resources (DER) [7, 38–42]. The most commonly used method to represent end-use load is to combine constant impedance, constant current, and constant power elements, known as a ZIP model [5, 43]. These loads are often scaled by feeder load profiles to represent changing load at each QSTS time step. In contrast, a dynamic load model switches on and off according to independent schedules or control signals. Such detailed load behaviors are useful for simulations since the majority of the end-use loads have sub-minute time scale behaviors, not usually captured in feeder load profiles [44–46]. To evaluate their impact on residential distribution system operations, end-use load models on residential distribution feeders that are changing with time are required [45], yielding a detailed load model for peak and off-peak periods.

Apart from the above mentioned deterministic methods, some stochastic methods have been used as well. Hidden Markov models (HMM) have been used to represent load behaviors [47–50], where the probability of transition between the state of the appliance is assumed to be dependent on the current state of the appliance. Reference [47] presents a method of modeling the sequence of operational states of an appliance from measurements of power consumption using HMM. In [48], the author introduced an HMM model to describe and identify thermostatically controlled automatic appliances and fixed-operation human-controlled appliances. In contrast to the HMM models, this work uses and provides visibility to the load's operational states.

Software tools such as GridLAB-D have detailed physical models to represent end-use behavior loads in a house, e.g. water heaters, heating and cooling systems [8]. In [5], a multi-state load model is presented with a detailed equivalent thermal parameter model (ETP) for heating, ventilating and air conditioning (HVAC) units. These models have thermodynamic states and differential equations relating to many random thermal inputs over the population of houses. The simulations are deterministic with detailed physical models of each load component, requiring lengthy com-

putation for large numbers of houses. As the input of QSTS simulation, the load behaviors are influenced by many random input variables. To avoid simulating every single load component, it is reasonable to use a stochastic method to obtain the load behaviors for the inputs to QSTS simulations [51].

2.1.2 Related Work - Feeder Model Order Reduction

To reduce the computational burden of QSTS simulations on large distribution systems, in [52], the A-Diakoptics methodology is presented to reduce the simulation time by separating feeders into sub-networks and solving them in parallel using multi-core machines. A segment substitution method for distribution system model simplification is introduced in [3] eliminating buses on large distribution feeders. The simplified model from that method greatly reduces the simulation time with an acceptable simplification error for the study of photovoltaic (PV) output fluctuations. In a local area, these PV fluctuations tend to be similar in shape and correlated in time, as they originate from the same solar irradiance and cloud cover. However, it is a reasonable simplification for loads only if they all follow the same load shape. When loads at different locations follow different load shapes, the error will increase rapidly. In [53], the author introduced a linear sensitivity model to estimate the impact of load and PV profiles on the magnitude of phase voltage as well as the states of voltage control devices to speed up the simulation. In [54], the authors used a linearized matrix method with weighting factors to transfer loads and generators onto retained buses. Neither considered that loads may follow different profiles at different locations, and this may degrade the resulting accuracy of estimation.

Solar generation is known to vary locally, but the usual practice is to assume that loads vary uniformly and slowly. This is not really the case, especially in small circuit segments with air conditioners, water heaters, microwaves, and other loads turning on and off. Existing QSTS simulations tend to use a uniform load shape for all the loads of the same types. This will potentially capture the action of a substation transformer Tap Changer that regulates the whole feeder, shown to the left in Fig. 4. Loads distributed out on the feeder actually vary independently as shown, but in large numbers, these diverse loads may be aggregated at the feeder level with acceptable results. However, this assumption is less valid in small numbers, and it usually won't capture the behaviors

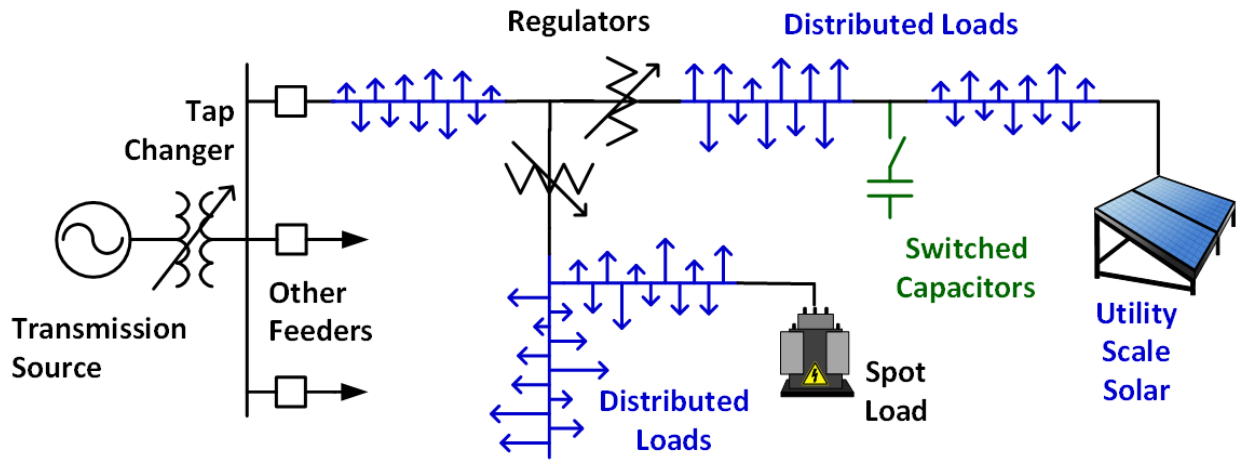


Figure 4: Feeder model with independent load and generation components. Distributed load arrow lengths represent the instantaneous load magnitude at each location, normalized to its own peak.

of line regulators and capacitors with local controls that are located out on the feeder. For example, if the switched capacitors in Fig. 4 operate on local voltage control or reactive power control, then their behavior is largely determined by the downstream solar generation and segment of distributed load, at the right of Fig. 4. The upstream line regulator responds to the distributed load, switched capacitors and utility scale solar to its right. However, the bottom-most line regulator in Fig. 4 responds to its own downstream distributed load, plus its large downstream spot load. Those two regulators respond differently because the solar generation depends on weather, while the spot load depends on customer schedules, and these are loosely correlated if at all. Furthermore, at the instant shown in Fig. 4, the switched capacitor sees less than its share of distributed load while the bottom-most regulator sees more than its share.

This local stochastic behavior is masked by diversification over the whole feeder, which has been acceptable when considering the whole feeder serving passive load. As feeders become more active due to solar generation, storage, and responsive loads, it will be more important to represent these local stochastic behaviors. Otherwise, errors may occur in the analysis of grid constraints, controls and operating conditions. We aim to alleviate this risk by enabling faster simulation tools, but within acceptable error bounds.

In [3], we presented a simplification method that achieved 0.2% error, or less, when the ZIP loads vary uniformly and the PV outputs also vary uniformly. However, when the ZIP loads in each segment do not vary uniformly, the errors in voltage can be higher than 1.0%, which is not good enough for voltage control and violation studies. The example in [3] reduced a feeder with 1225 primary line segments to 23 segments, retaining the key bus connections for devices that help control voltage. Each segment contained one equivalent ZIP load representing its internal uniform load (which varies with time), and a pair of independent sources representing all the PV that vary with time. We improve the error to an acceptable limit by generalizing the superposition of sources on a simplified model, and also speed up the simulations with stochastic models of independent loads. Even with many more independent sources, the number of simultaneous equations to solve in the power flow model is the same as before, which helps to retain most of the computational efficiency.

In this section, we propose two complementary approaches to improve the computational complexity of QSTS simulations in more realistic models of large distribution systems in two different aspects. Accordingly, we first introduce a stochastic method to efficiently generate load behaviors of end-use appliances to support QSTS simulations, providing a general procedure to generate stochastic load behaviors for a given number of independent loads (end-use appliances) without simulating every load instance in detail. We show that the proposed method is able to accurately capture the time correlation of the appliance behaviors in large systems. The results for water heaters are promising; along with air conditioners and heat pumps, these have the most impactful stochastic load behaviors on distribution feeders [2, 11]. Work is underway to apply and validate the method for composite load types, to be reported in a future publication. Secondly, considering the problem described in section 2.1.2, to reduce the QSTS simulation time, a new segment simplification method for large distribution system models is introduced. The proposed simplification method is applicable for QSTS simulations with independent dynamic load behaviors at different parts of the distribution circuit by extending the method presented in [3]. It is demonstrated that independent load behavior can be integrated into the QSTS simulations with the proposed distribution system simplification approach. We also show that other methods for circuit simplification are not as accurate as the proposed method for simulation purposes. Note here that the proposed approach is novel and transformative such that our approach can be applied to models with multi-

ple distributed sources with independent time series profiles, such as PV, energy storage systems and electric vehicles. In this study, we present the methodologies for independent load behavior generation and circuit simplification and demonstrate the performance of the proposed methods on a distribution model simulated in OpenDSS [55]. We compare the proposed approach in terms of computational time and simulation accuracy with a QSTS simulation approach that does not account for independent load behavior [3]. Note here that circuit simplification for computational complexity reduction of QSTS simulation under independent load behavior is not trivial, and this study provides solutions to this important problem.

2.1.3 Systematic Flowchart

Fig. 5 summarizes the overall implementation of the proposed Algorithms 1 and 2 with the usage of and interaction among different simulation tools. We utilized three main software tools in this work as follows:

1. MATLAB: Used to produce and simulate stochastic end use load models, e.g., water heaters, not connected to an electric power distribution system.
2. OpenDSS: Used as a fast and robust QSTS power flow solver of the distribution system, but with combinations of constant impedance, constant current and constant power (ZIP) loads only [5, 43].
3. GridLAB-D: Used for QSTS power flow with climates, thermal envelopes of buildings, and load behavioral models connected to an electric power distribution system [8].

GridLAB-D generates temporally varying water heater data, which is then utilized in Algorithm 1 in MATLAB to obtain stochastic aggregated load variation data for any given number of water heaters, represented as $M_n(t)$ in this figure. The details of Algorithm 1 are in Section 2. The full feeder model is segmented and fed to the proposed Algorithm 2. In each segment, Algorithm 2 generates aggregated impedance and load values (represented as Z_i and S_i, respectively) together with compensating currents and gains (represented as I_c and G_c) in Fig. 5. Compensation currents and gains are used to account for changes in load behavior. The details of Algorithm 2 are provided in Section 3. Aggregated loads and compensation currents for each segment are presented in Fig.2 as the Simplified Feeder Model. Finally, Simplified Feeder Model and compensation gains

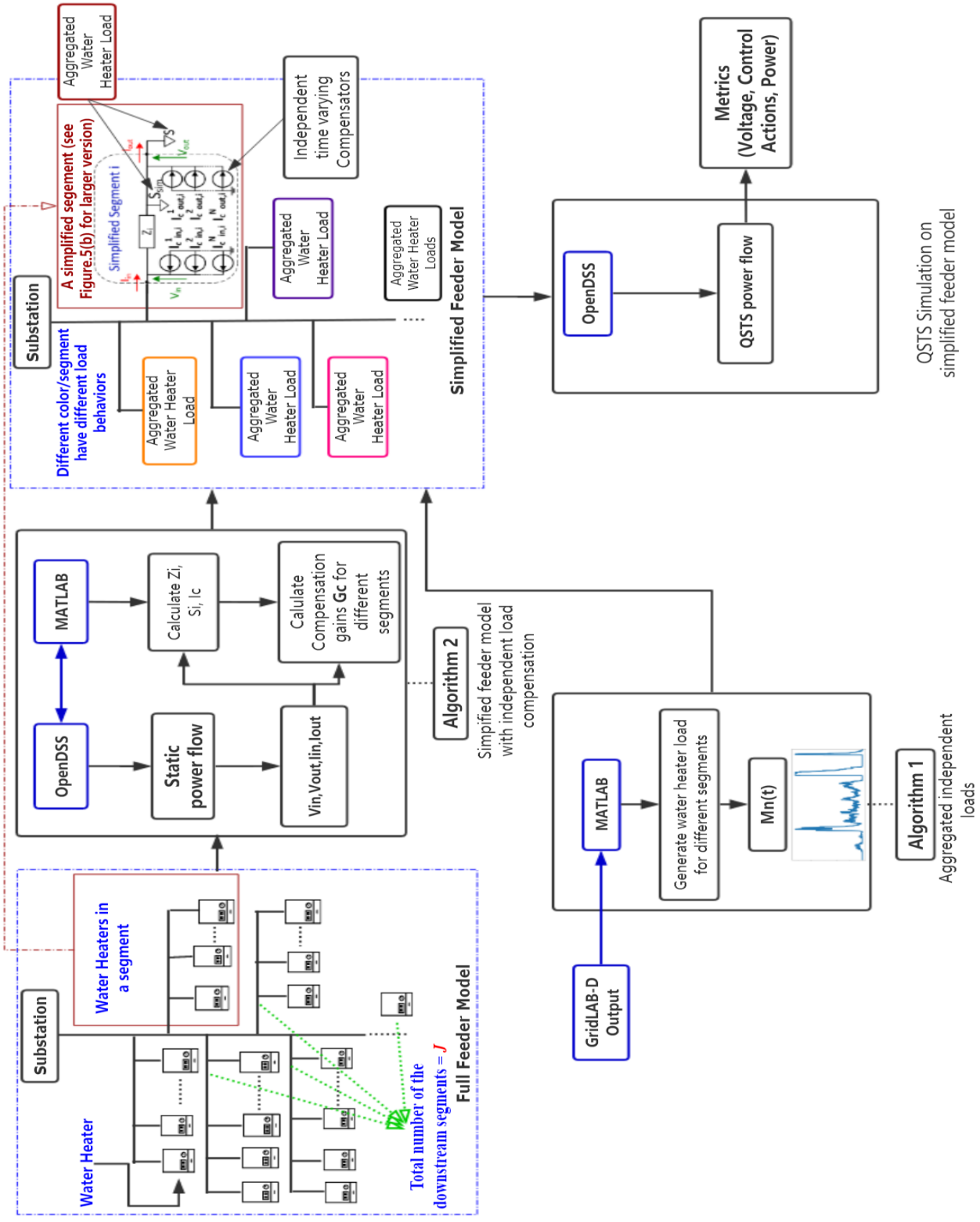


Figure 5: A systematic flowchart of the proposed algorithms with a demonstration of simulation scenario and feeder topology.

(G_c) together with the output of Algorithm 1 ($M_n(t)$) are used in OpenDSS to obtain the QSTS simulation results. The output voltage is used as the performance metric to compare the simplified model to the full model (OpenDSS QSTS) simulation.

2.2 Stochastic Modeling for Load Shapes of Behavioral Load

GridLAB-D has physical models for houses including many kinds of appliances such as dishwasher, dryer, electric vehicle (EV) charger, freezer, water heater and HVAC. The simulations are deterministic; every house model instance has a thermal heat flow circuit and its ETP heat balance equations [8]. Each type of appliance's state in each time step during the simulation will depend on the solution of the equations associated not only with thermally-related variables of the house such as floor area, window-to-wall ratio, air temperature and air exchange rate but also random variables of each type of end-use appliance. Many of these variables are driven by the activity of the occupants, which GridLAB-D represents with randomized schedules. When large numbers of houses are simulated on distribution feeders, computational time will increase dramatically. Thus, our goal is to make the simulations more efficient by using stochastic parameters with reduced-order models. By the proposed method in this section, the stochastic load behaviors can be obtained efficiently for any given number of the same type of appliance. Stochastic simulation in MATLAB

Table 1: Time-Dependent Parameters for Stochastic Load Modeling

Symbol	Definition
P_{t+1}	Bernoulli trial parameter of each OFF appliance's probability of turning on before time $t+1$
$N_{t(on)}$	The ratio of appliances that are on at time t in GridLAB-D output
$N_{t(off)}$	The ratio of appliances turning off just after time t in stochastic simulation
X_t	Probability distribution of the time each appliance stays on, having switched its state from off to on after time step t

is used to build aggregate behavioral load shapes of a certain type of appliance based on GridLAB-D simulation results of larger numbers of houses and appliances. These GridLAB-D simulations comprise the appliances only, each one controlled by randomized end-use schedules (e.g. showers in early morning and dishwashing in evening influence water heater behavior). The feeder power flow analysis is not included at this stage. The procedure to generate aggregated load behaviors is described in the following steps:

1. A preliminary step is extracting appliance behavior from GridLAB-D, specifically, obtaining time-dependent probability distribution parameters that are summarized in Table I from the original simulation results in GridLAB-D.
2. Calculating P_{t+1}: for each appliance, the probability that each one of them that is off at time step t will switch from off to on before time step $t+1$, conditioned on the number of appliances that are off at time t. Equation 2.1 and Equation 2.2 are the mathematical expressions to compute P_{t+1} for each time step t:

$$P_{t+1} = \frac{N_{t+1(on)} - (N_{t(on)} - N_{t(off)})}{1 - (N_{t(on)} - N_{t(off)})} \qquad t \geq 1 \tag{2.1}$$

$$P_1 = \frac{N_{1(on)}}{1} \qquad t = 0 \tag{2.2}$$

   Where $N_{t(on)}$ is the ratio of appliances that are on at time t in GridLAB-D output. $N_{t(off)}$ represents the ratio of appliances turning off just after time t in stochastic simulation. For instance, assume the total number of appliances is five. Three of them are on at 6:00, so at time point t, $N_{t(on)}$ is 0.6. One of them is turning off just after time point t at 6:05 in the stochastic simulation (time t is the end of ON duration of this appliance), so $N_{t(off)}$ is 0.2. Then for the next time point $t+1$, four of them are on in the GridLAB-D data, so $N_{t+1(on)}$ is 0.8. Using Equation 2.1, P_{t+1} will be 0.67 in this example. Equation 2.2 is used to calculate P for the first time step, which is a special case for Equation 2.1. Since we are assuming all the appliances are off as the initial state, $N_{t(on)}$ and $N_{t(off)}$ are both zero when $t = 0$.
3. Calculating X_t: the probability distribution of the on-time duration for each appliance that switches its state from off to on between time step t and time step $t+1$. Example on-time distributions of water heaters at three different times of the day are shown in Fig. 6.

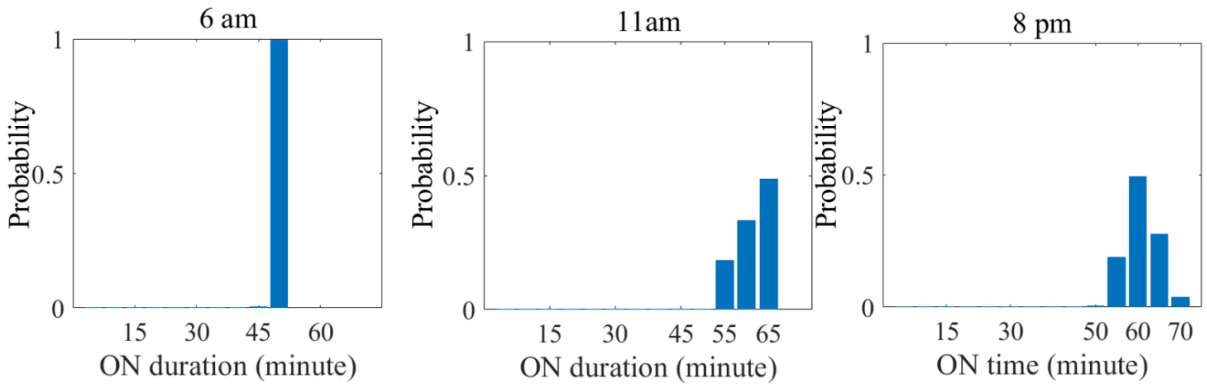


Figure 6: An example of calculating X_t for different times of the day:(left) if the water heater turns ON at 6 am, it will stay on for around 45 minutes with almost probability 1; (middle) at 11am, the length of ON duration is distributed non-uniformly between 55 and 65 minutes; (right) at 8 pm, the length of ON duration is distributed between 50 and 70 minutes.

Note that each appliance type (e.g. water heaters with different vintages and tank sizes are different types) will have different P since they could have different behaviors; similarly, the on-time probability distribution X_t also depends on the characteristics of the appliance. Once these parameters are obtained, we have the ability to generate load behaviors and power consumption values for both individual appliances and, if the initial GridLAB-D population is large enough, for large numbers of appliances having the same distribution of significant load parameters (e.g. residence floor area, temperature setpoint, water demand schedule, etc.).

The stochastic load behaviors could be generated by the following procedure without simulating the appliances in GridLAB-D every time.

1. Determine the type of the appliance. This may be randomized from a uniform distribution of available types, or appliances may be treated systematically by type (e.g. water heater, microwave, refrigerator, lights). It's also possible to aggregate different kinds of appliance into the process. In this method, we aggregate different sizes of water heaters to learn P_t and X_t from GridLAB-D simulations. The recursive approach to calculate P_t (the probability for each appliance to switch from OFF position to ON position) is demonstrated in equations (1) and (2). X_t (the probability distribution for the length of ON duration when an appliance changes from

OFF to ON position) is calculated for different times of the day using GridLAB-D simulations as explained in Fig. 6. Note that X_t can be different for different type of appliance.

2. Generate a stochastic load behavior for an individual appliance of a certain type. The procedure is summarized in Algorithm 1.

Algorithm 1: Stochastic modeling of behavioral load

Data: P_{t+1}, $N_{t(on)}$, $N_{t(off)}$,X_n, n: number of appliance , $P_{WH}(kW)$: heating element capacity of the water heater

Result: s_t^i: Status of appliance i at time t

$L_{WH}(t)$: Aggregate load behavior of a certain type of appliance

```
for t = 1; t ≤ t_max do
    Obtain N_t(off) (assuming all OFF at the first time step and use (2) for P_1);
    Calculate P_{t+1};
    for i = 1; i ≤ n do
        if s_t^i = 0 (1:ON,0:OFF) then
            Sample from P_{t+1} to determine s_{t+1}^i;
            if s_{t+1}^i = 1 then
                Sample from X_{t+1} to get a on time duration k;
                s_{t+1:t+1+k}^i = 1 (Appliance i will be on during this period);
            end
        end
    end
    L_WH(t) = Σ_{i=1}^{n} (s_t^i) × P_WH
end
```

For each individual instance of a specific type of appliance, at time step t, if the appliance is OFF, sample to determine whether it will turn ON. Following the earlier example, each of the three water heaters that are OFF at the end of 6:00 period (two were already OFF, one switches to OFF at the end of the 5 minute period) has a 0.67 chance of turning ON by 6:05. Whenever the appliance turns ON, sample from the on-time probability distribution for that time point to obtain an on-time duration, e.g. Fig. 6. The appliance will be ON throughout this duration

and turn OFF afterward. Once it does turn OFF, it may turn ON later according to the time-dependent probability in Equation 2.1. If the appliance won't switch from OFF to ON, go to the next time step and sample again. Repeat this process until the end of the simulation time. Then the aggregate stochastic load L_{WH} at each time step t could be obtained by adding the load of each individual appliance that turned ON at t or stayed ON from an earlier time step.

Our approach requires for each time frame t, an aggregated probabilistic distribution that will characterize the switching between OFF to ON conditions (P_t), and the length of ON condition when an appliance changes from OFF to ON condition (X_t). This method recursively calculates these distributions from GridLAB-D simulations, and it samples from these distributions to generate a temporally varying load behavior. More specifically, P_{t+1} and the probability distribution X_t need to be calculated indirectly by the status (ON or OFF) of the same type of appliance over a discrete time interval, which is obtained from GridLAB-D simulations. Note that the GridLAB-D appliance simulation only needs to be run one time to obtain P_{t+1} and X_t, using $N_{t(on)}$ and $N_{t(off)}$ from the simulation results to get the canonical behaviors for that type of appliance. Once the stochastic model is built, it will be scalable to generate any number of stochastic load behaviors for a specific sample set of appliances, without re-running GridLAB-D. We randomly place water heaters of different types and sizes within a feeder model, according to probability distributions that depend on the region, type of dwelling and individual dwelling's floor area. Given that, water heater loads depend on randomized schedules of occupant behavior, which we also scale and shift in time according to probability distributions. The stochastic model described in this section addresses the schedule-dependent water heater behavior, and we chose sample set sizes approximately equal to the number of electric water heaters found on a single distribution feeder.

2.3 Segment Substitution for Models with Independent Loads

2.3.1 Previous Work on Distribution System Model Simplification

Many analysis tasks do not require the full electrical state (bus voltage and branch current) at each point on a full distribution feeder. Fig. 7 is a simple demonstration of the segment substitution

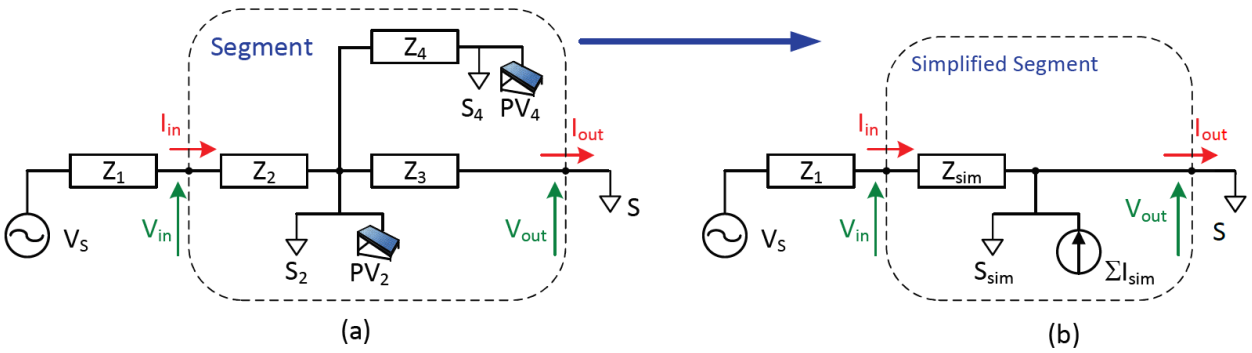


Figure 7: Segment substitution Demonstration

method that we developed in our previous work [3]. This existing method is able to greatly reduce simulation time by replacing the full segment topology, which is often more complex than Fig. 7(a), in the distribution system with simplified segments in Fig. 7(b). We showed that through this method under a constant load assumption, the computation time of QSTS simulation is reduced by 95% with a voltage error as low as 0.2%. As mentioned above this method assumes a constant load scaling across the segments, which means that the loads in different segments follow the same global load shape, i.e., they are not independent. Moreover, even though PV compensation exists through a current source, all the PVs in different segments are assumed to follow the same temporal behavior. Utilities usually have data for these global load shapes. However, loads in different segments generally have different load behaviors, for example, weather and temperature characteristics and corresponding user preferences may vary with time and location. These differences will introduce temporally changing and location-dependent load behaviors. These load behaviors could increase the error during QSTS simulations on the simplified topology in Fig. 7(b) to analyze voltage violations and voltage regulator operations. We also show this increase in error through our simulation results, see Section 2.4.3.

2.3.2 Segment Substitution for Models with Independent Loads

To address the error introduced due to the temporally varying load behavior in the simulation of distribution systems through the previous simplification method, in this study, we present a new simplification methodology as shown in Fig. 8(b). More specifically, for more accurate simula-

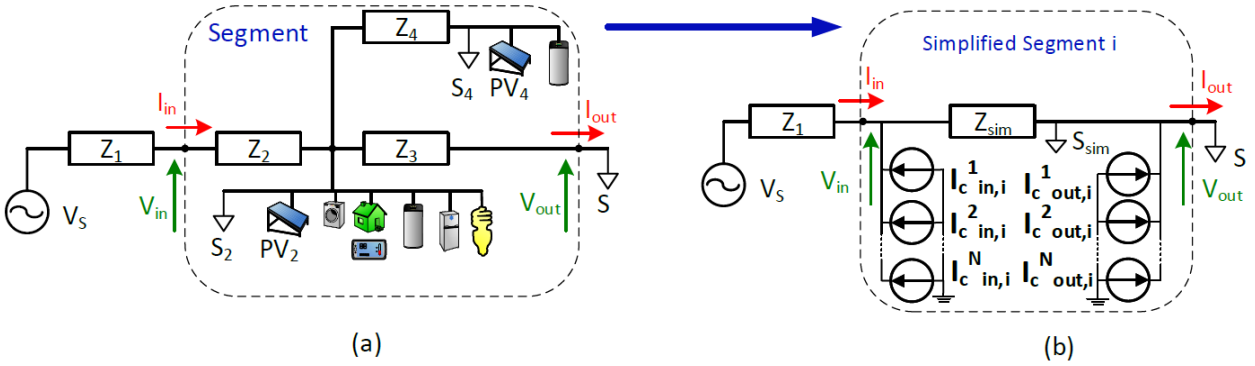


Figure 8: Topological realization of segment substitution for models with independent loads, per phase, expanded from Fig. 7 of [3].

tions, loads within each area corresponding to each segment in the simulation model should have independent changing load behaviors. Models with independent load shapes have a similar effect to PV on the voltage drop across the series impedance of simplified segments [3]. If the loads of each segment follow different load shapes, the corresponding load currents in different segments throughout the feeder will not change in proportion to each other. Independent load shapes of each segment can affect the shunt current, and the voltage drop across the segment will be influenced by different load shapes of both upstream and downstream segments. The voltage drop across the simplified segment should be an appropriate approximation to the voltage drop across the full segment in a distribution model with independent loads. To accomplish this, the simplified segment model should have elements to compensate the effect of independent load behaviors from the other segments as shown in Fig. 8(b). Following the model illustrated in Fig. 8(b), with the current source compensating the influence from the other segments, the voltage drop across each segment i is:

$$\Delta\bar{V}^i = Z_i((-1 * \sum_{n=1}^{N} M_n(t) I^n_{c\ out,i}) + \sum_{j}^{J}(-1 * \sum_{n=1}^{N} M_n(t)(I^n_{c\ in,j} + I^n_{c\ out,j}))) \tag{2.3}$$

$I^n_{c\ in,i}$,$I^n_{c\ out,i}$	Independent current sources to compensate the voltage drop in segment i due to load behavior of segment n
$M_n(t)$	Independent load shape of the segment n at time t
N	The number of segments
n	The index of the load shape of a different segment
i	The index of the simplified segment
j	The index of the downstream segment
Z_i	Simplified equivalent impedance of segment i, shown as Z_{sim} in Fig. 8(b), the detailed calculation of Z_i is in [3].
$\Delta\bar{V}^i$	Voltage drop across segment i

Each pair of $I^n_{c\ in,i}$,$I^n_{c\ out,i}$ added to the topology will compensate the effect of the load behaviors on the voltage drop in segment i due to the independent load behavior in segment n. For example, if the full circuit model has ten segments after simplification, each segment will have 10 pairs of I_cs as shown in Fig. 8(b) to compensate different load behaviors from all segments. In other words, the compensating currents account for other segments, not for different load types. Stochastic load contributions in those other segments would have already been calculated. The total segment number N is determined by the number of the buses to be preserved in a full model; components between two preserved buses form a segment. Through equations 2.4 to 2.7, we demonstrate how to compute values of the compensating current I_c for each segment in the proposed simplified topology. The procedure is also summarized in Algorithm 2. With ten segments, there are 100 power flows performed within the loop, at Line 5, and two before the loop, at Lines 1 and 2.

$$I^n_{c\ in,i} = Z_i^{-1}[(V^n_{in,i} - V^n_{out,i}) - Z_i I^n_{in,i}] \tag{2.4}$$

$$I^n_{c\ out,i} = (I^0_{in,i} - I^0_{out,i}) - (I^n_{in,i} - I^n_{out,i}) - I^n_{in,i} \tag{2.5}$$

$$G^n_{in,i} = V^n_{in,i}(I^n_{c\ in,i})^* \tag{2.6}$$

$$G^n_{out,i} = V^n_{out,i}(I^n_{c\ out,i})^* \tag{2.7}$$

Algorithm 2: Independent load compensation

Data: Total number of segments=Total number of independent load shapes=N

Result: Independent load compensators G

Solve the full system model at no load (e.g. in GridLAB-D or OpenDSS), obtain the no load voltage and current, $V^0_{in,i}$,$V^0_{out,i}$,$I^0_{in,i}$,$I^0_{out,i}$,of each segment ;

Solve the full system model at full load. Determine the series impedance and the full load of each segment (internal load plus the junction load) using the method in [21] ;

while $n \leq N$ **do**

  for $i = 1; i \leq N$ **do**

    Solve the full system model power flow with S^n at full output and $S^{i \neq n} = 0$;

    (S^n: loads that follow load shape n, $S^{i \neq n}$: all the other loads);

    Obtain $V^n_{in,i}$, $V^n_{out,i}$,$I^n_{in,i}$, $I^n_{out,i}$;

    Calculate $G^n_{[in/out],i}$ (Fig. 5) by Eq. (4)-(7) ;

  end

  $i = i + 1$;

end

Compute $G^{n,k}_{[in/out],i}$ for each segment i for different time values k as in Eq. (8) to compensate for the independent time-varying load behavior of segment n.

Where $V^0_{[in/out],i}$ and $I^0_{[in/out],i}$ are the no load voltages and currents of a segment i. In all these equations, V_{in} represents the voltage at the initial point of a segment, V_{out} represents the voltage at the end of a segment, I_{in} represents the input current of a segment, I_{out} represents the output current from the end of a segment and i is the segment index. $V^n_{[in/out],i}$ and $I^n_{[in/out],i}$ are the voltages and currents of segment i when all the loads that don't follow load shape i are set to zero, i.e., $S^{i\neq n} = 0$, $I^n_{c[in/out],i}$ are independent current sources on segment i for load shape n, $G^n_{[in/out],i}$ are the compensating power values of segment i for load shape n. Equation 2.4 and 2.5 are used to find the compensation source values for each segment i associated with each independent load shape n, some of which were generated stochastically using Algorithm 1. Once the values of I_cs are obtained, the full system segments i can be replaced with the corresponding simplified segment shown in Fig. 8(b). Each pair of currents sources $I^n_{c[in/out],i}$ associated with $V^n_{[in/out],i}$ compensates for each independent load shape n. The compensator values $G^n_{[in/out],i}$ obtained through Equations 2.4- 2.7 differ among the segments but these are not yet time-varying. For the purpose of compensating the independent time-varying load behavior generated as described in Section 2.2, the stochastic dynamic load behavior for a single day for each segment is divided into windows of a certain time length. Then within each window, a pth percentile value (p can be 25, 50, 75...) of the normalized load shape M_n is selected to represent the time-changing load behavior. Fig. 9 illustrates a window for which the percentile value is chosen empirically. Equation 2.8 shows how the compensating power value $G^{n,k}_{[in/out],i}$ is changing for different windows to compensate the time-varying load, where k represents different windows, so that the left-hand side of Equation 2.8 is time-varying. In quasi-static time-series simulation, for all segments i, each compensator follows the load shape corresponding to segment n, $M_n(t)$. The selected window size and the percentile value will affect the simulation result as discussed in Section 2.4, based on a numerical example of load behavior for 900 water heaters.

$$G^{n,k}_{[in/out],i} = M_n^{percentile,k} G^n_{[in/out],i} \tag{2.8}$$

By using the proposed new topology, instead of a global load shape for each segment in [3], we introduce independent dynamic loads on the simplified distribution system segmented model. This method ensures the usability of segment substitution under the condition that loads of different areas have different behaviors. Furthermore, the method can be extended to compensating different

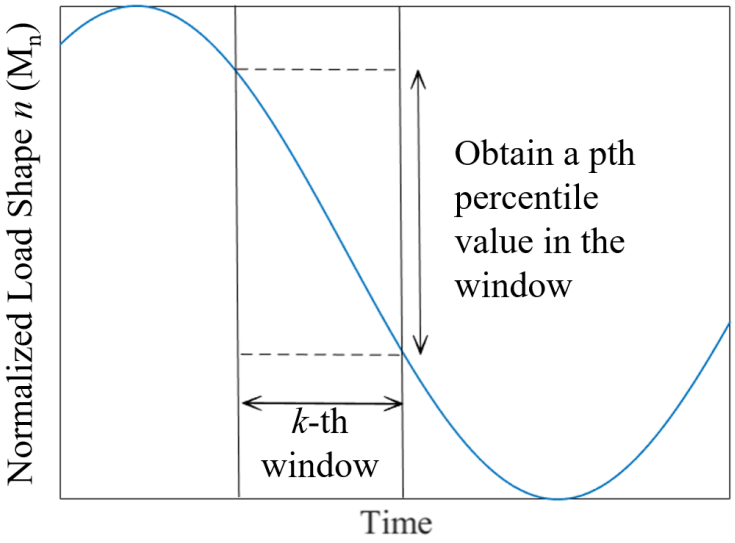


Figure 9: A demonstration of how to obtain $M_n^{percentile,k}$.

sources with variable time series profiles, such as load, PV, battery and electric vehicle charging, all of which have independent behaviors injected into the system at different locations. They would all contribute to the totals as in Fig. 9 or other results presented later.

2.3.3 Voltage Drop and Shunt Current Relationships

This section describes how the added current sources compensate the effect of independent load in the other segments on the voltage drop on segment i. In [3], for a system simplified from Fig. 7(a) to Fig. 7(b), the voltage drop $\Delta\bar{V}^i$ across each simplified segment i is an approximation of the voltage drop ΔV^i across each full segment i under the same loading conditions:

$$\Delta\bar{V}^i \approx \Delta V^i \tag{2.9}$$

The voltage drop is determined by the cumulative voltage drop resulting from branch currents across the series branches between the input and output buses. The voltage drop and shunt current are proportional to a linear combination of the segment's compensating load currents and the segment output current. In the simplified segment of Fig. 8(b), the voltage drop is determined by the input current across the series impedance.

$$\Delta\bar{V}^i \approx Z_i\bar{I}^i_{in} \tag{2.10}$$

Where $\Delta\bar{V}^i$ is the voltage drop across the simplified segment i, and $\bar{I}^i_{in}$ is the input current. The load multiplier, M, is normally applied to the complex power of a load. The load voltages are assumed to change proportionately at different load conditions such that for each segment i:

$$M^i(t)(\frac{S_i}{V_i})^* \approx M^i(t)I_i \tag{2.11}$$

Where $M^i(t)$ is the time series of load multipliers known as the load shape for segment i, S_i is the total complex power drawn by segment i, I_i is the total current drawn by segment i, and V_i is the voltage associated with this segment. Under this assumption, the segment load current follows the segment load shape. Therefore:

$$\Delta\bar{V}^i = Z_i(M^i(t) * \bar{I}^i_{in}) \tag{2.12}$$

The input current is the sum of the output current and current I_i drawn by the complex power S component of segment i. The output current is the sum of the shunt currents of all downstream load objects (segments, junctions, and unsimplified loads) j.

$$\Delta\bar{V}^i = M^i(t) * Z_i(I_i + \sum_j I_j) \tag{2.13}$$

When the segments follow different load shapes, such as the stochastic load behaviors discussed in this study, the influence of these different load behaviors on the segment voltage drop is shown in Equation 2.14, which can be further extended to Equation 2.3:

$$\Delta\bar{V}^i = M(t) * Z_i(M_i(t)I_i + \sum_j M_j(t)I_j) \tag{2.14}$$

Where $M_j(t)$ is the specific load shape for segment j. In the full system model, the shunt current is determined by the loads throughout the segment (neglecting any no-load loss). Similar to voltage drop, the shunt current $\Delta\bar{I}^i$ across a simplified segment i is an approximation of the shunt current $\Delta\bar{I}^i$ across the full segment i:

$$\Delta\bar{I}^i \approx \Delta I^i \tag{2.15}$$

More simply, using Equation 2.3 as described above, the shunt current across a simplified segment without independent load compensation, for a global load shape $M(t)$, is equal to the load multiplier times the segment load:

$$\Delta\bar{I}^i(t) = M(t)I^i \tag{2.16}$$

To compensate for any independent effects on the load,

$$\Delta\bar{I}^i(t) = -1 * \sum_{n=1}^{N} M_n(t)(I^n_{c\ out,i} + I^n_{c\ in,i}) \tag{2.17}$$

$I^n_{c\ in,i}$ and $I^n_{c\ out,i}$ in Equation 2.3 and Equation 2.17 are the compensating current values, which follow the load shape M_n representing the effect of load behaviors of segment n on the voltage drop of segment i.

2.4 Numerical Results

In this section, we first demonstrate the performance of Algorithm 1 (see Fig. 5) that is used for stochastic load modeling. Specifically, we compare the stochastically generated load behaviors in MATLAB with the GridLAB-D water heater simulation results. We also show the generalization of Algorithm 1 to a different number of different water heaters.

In the second part of this section, we demonstrate the performance of the proposed framework, specifically, the combination of Algorithms 1 and 2 for model order reduction. As shown in Fig. 5, we identify different segments of the full feeder model and replace them with the simplified segment topology to obtain a simplified feeder model by Algorithm 2. The load behaviors obtained

through Algorithm 1 are used as the loads of water heaters on a test circuit in OpenDSS for QSTS. The compensators obtained from Algorithm 2 in the simplified feeder are changing according to the load behaviors obtained through Algorithm 1 to compensate for independent time-varying behaviour in QSTS of the simplified feeder model. To evaluate the performance of the new simplification method, the results are compared with QSTS on the full feeder model and simplified feeder model without the independent load compensators. The interactions between MATLAB and OpenDSS were performed using the OpenDSS automation interface from MATLAB as in [3].

2.4.1 Example for Stochastic Load Model: Water Heater

This section shows an example to implement the stochastic method to get aggregate load behaviors for water heaters. The training data set has 300 water heaters for each of three different tank sizes. Fig. 10 compares a stochastic water heater (not power flow) simulation result in MATLAB with the original output from GridLAB-D. The daily simulation with 30-second time step,

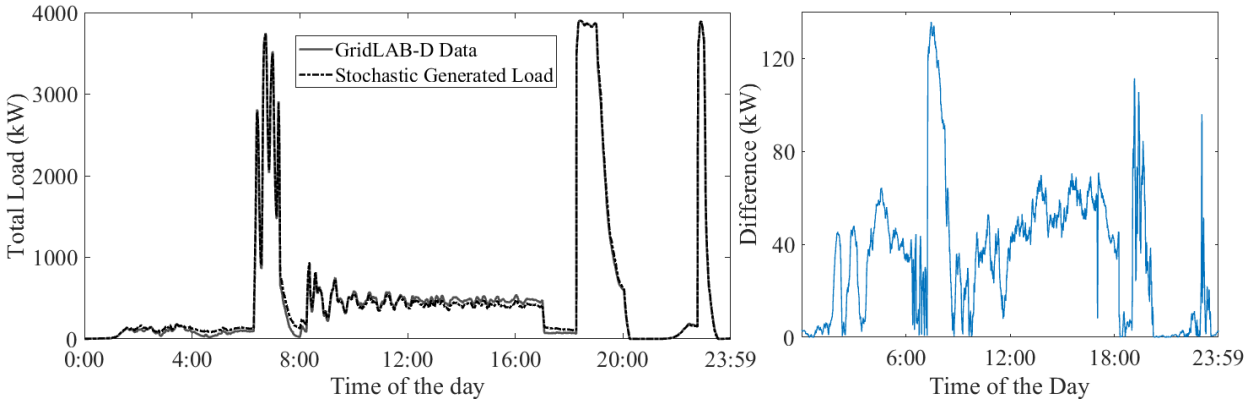


Figure 10: Stochastic simulation result for 900 water heaters.(The blue line represents the difference between stochastically generated load and the original simulation data from GridLab-D)

which was chosen based on typical heating times and time constants for water heaters, shows the changing load of 900 water heaters in the same area throughout a typical day in spring. The maximum error in the aggregate load at peak is about 3.4%. The results showed that the method is able to obtain very similar behaviors as the original output, including the time correlation. Both the

morning and evening peaks are well matched. Regarding computation time, generating a one-day load profile of 900 water heaters in GridLAB-D takes 140s, while subsequent stochastic simulation takes no more than 15s (89.3% reduction) to generate similar behaviors. Furthermore, once the water heater behaviors are extracted from GridLAB-D, it is faster to generate the aggregate power consumption for any other number of water heaters (having the same distribution of types) in a different circuit segment. Specifically, some simulation results for different numbers of water heaters are shown in Fig. 11 and Fig. 12 below, in which the similarities between the GridLAB-D and the proposed stochastic load behaviors are evident. The same stochastic model provides good fidelity to new GridLAB-D simulation results of 9, 30, 150 or 300 water heaters. The approach for water heaters can be extended to other behavioral load types as well, using GridLAB-D simulations to generate the training data sets as appropriate for the type of day, weather, customer schedules, etc.

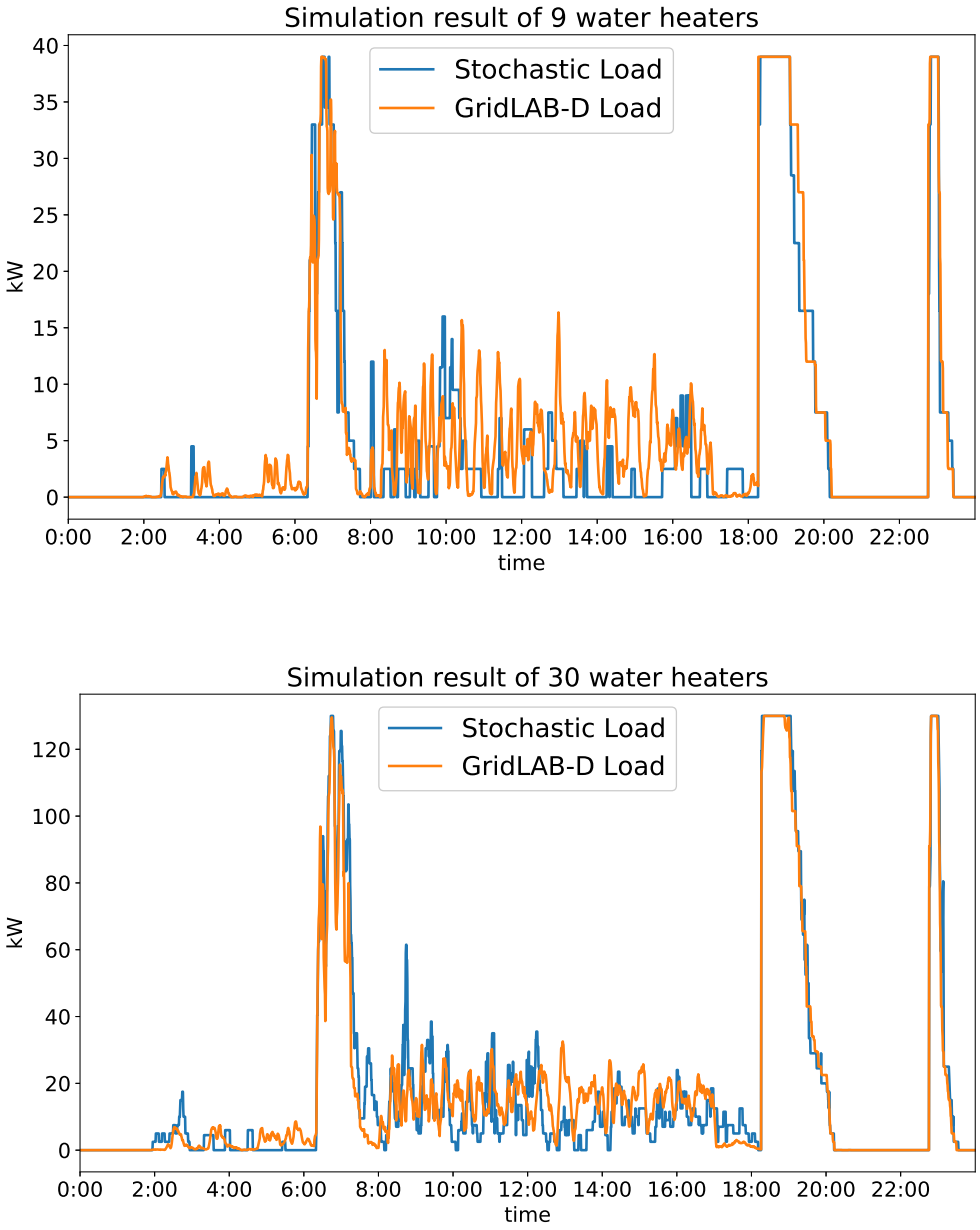


Figure 11: Stochastic load behaviors of 9 and 30 water heaters

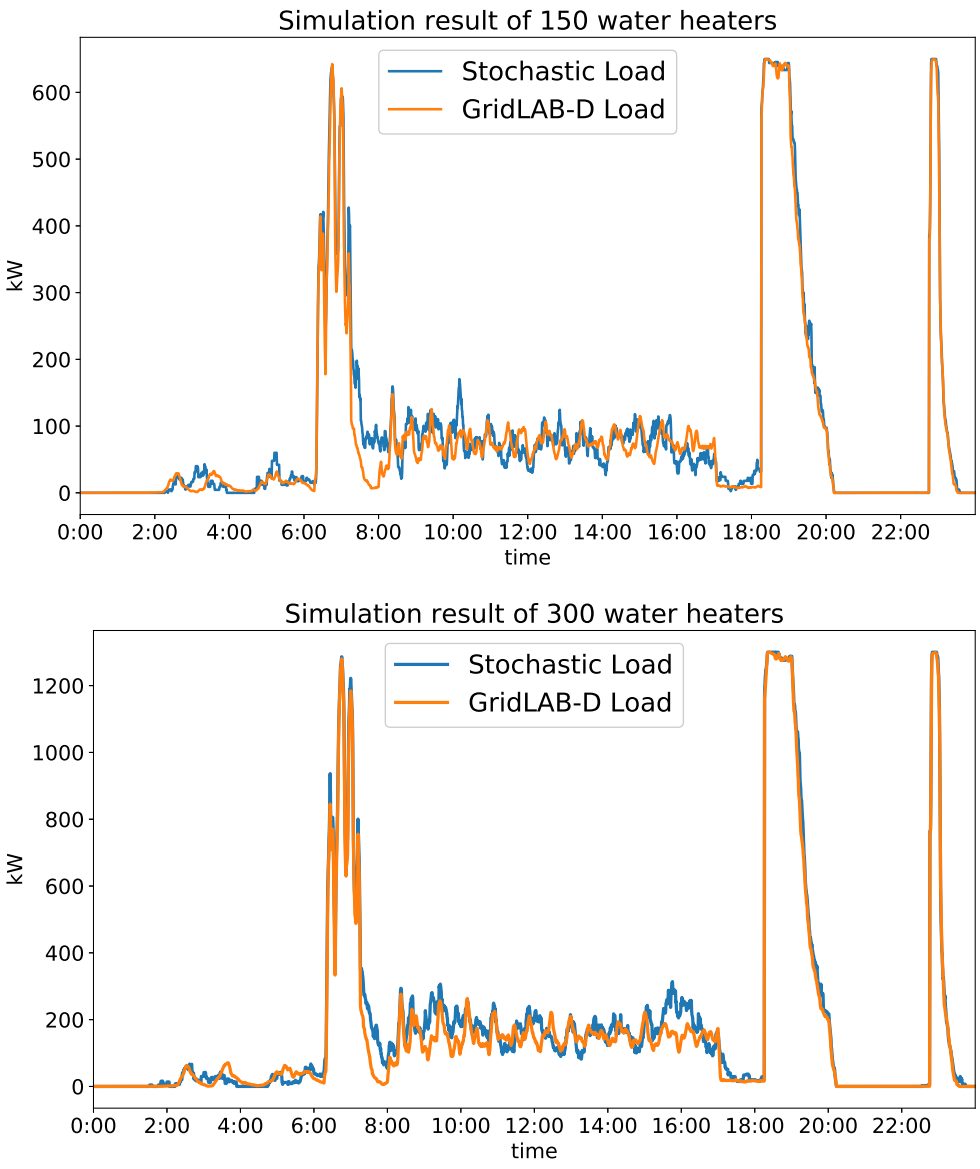


Figure 12: Stochastic load behaviors of 150 and 300 water heaters

2.4.2 Segment Substitution with Independent Loads on a Feeder Model

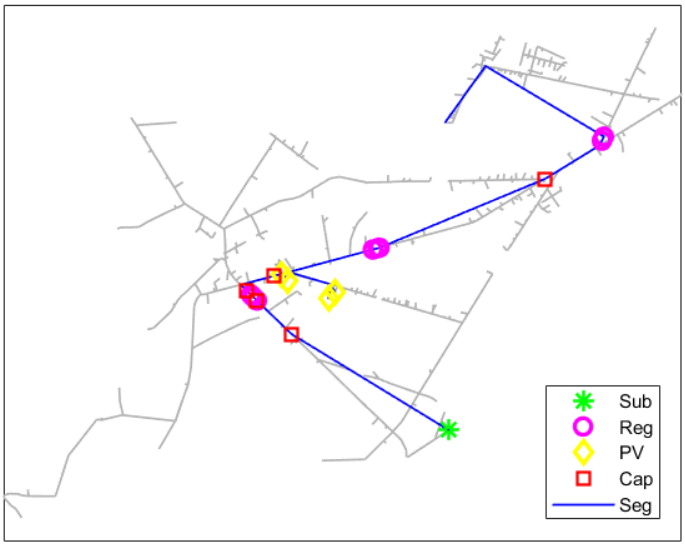


Figure 13: Layout of the EPRI J1 feeder before and after simplification [3].

In this section, segment substitution with independent dynamic loads is demonstrated on a full electric power distribution model that has houses represented. Assuming the only loads in the feeder model are water heaters, after simplification, each segment follows a unique independent load shape to represent load behaviors of the water heaters inside the segment.

The load behaviors are generated by the stochastic method. The test feeder model is EPRI feeder J1, a model in OpenDSS released to the public at [56]. Fig. 13 shows the feeder layout before and after simplification. The simplification procedure steps, including disable old component, enable each segment to calculate the compensator parameters, insert new components, and test performance were all performed using the OpenDSS automation interface from MATLAB as in [3]. The model has 23 segments after the segment substitution. The simplified model has 529 sources, which took 531 load flow solutions to set up. Buses connected to capacitors and regulators, i.e. voltage control devices, are preserved. The full model's three-phase voltage profile is shown in Fig. 14. The three phase voltages are per unit values vs. the distance from the substation

at the feeder head. Fig. 15 shows voltage profiles at full load after substitution with independent load compensators, 23 pairs of them in each segment. This result demonstrates that the simplified model gives a similar result compared to the full model with respect to voltage profile.

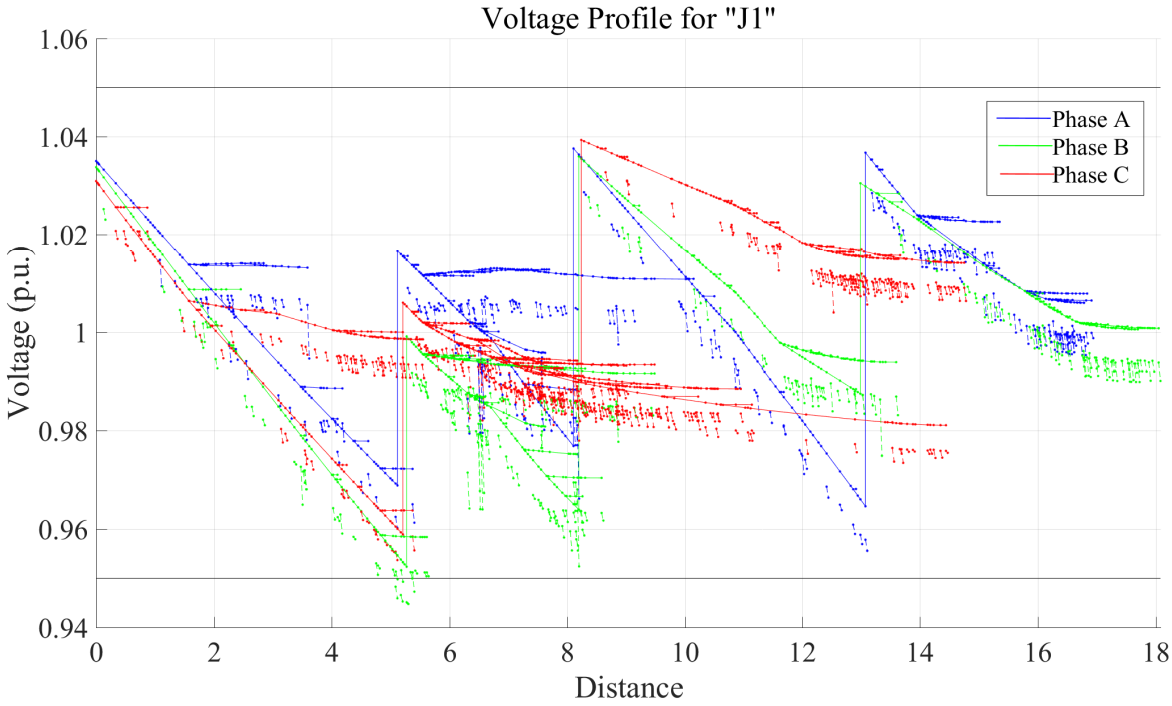


Figure 14: Three-phase unbalanced voltage profile for the full EPRI J1 feeder model.

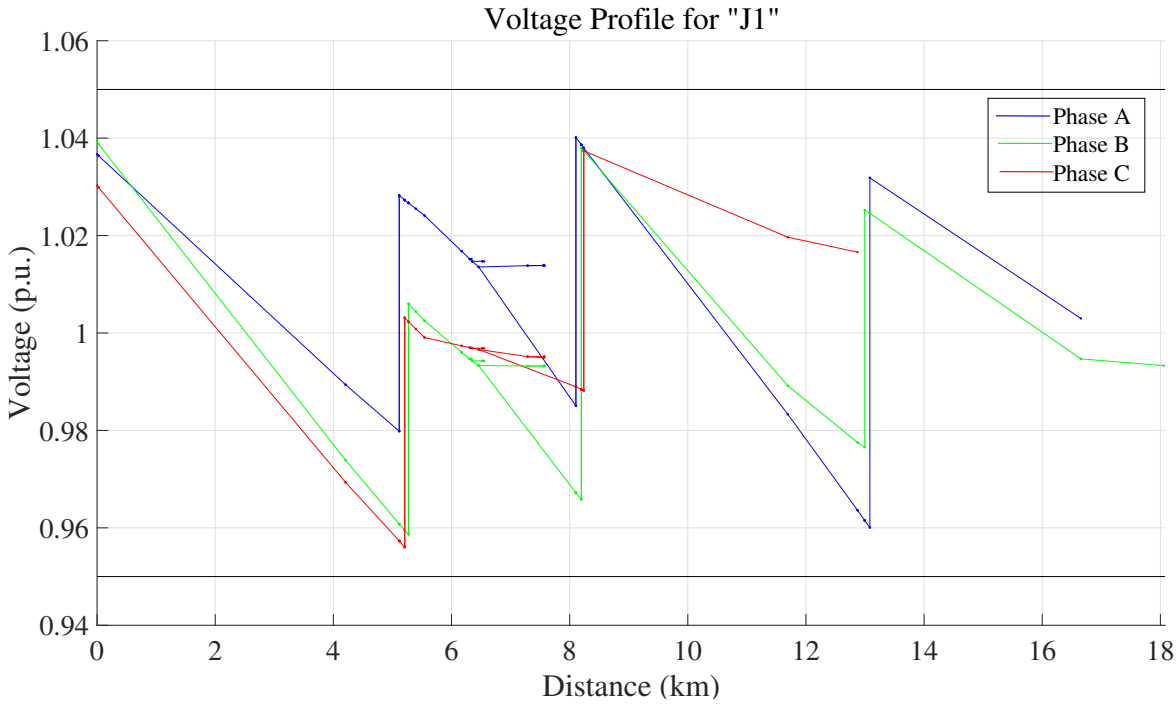


Figure 15: Three-phase unbalanced voltage profile for the simplified EPRI J1 feeder model by independent compensator substitution.

2.4.3 QSTS Simulation Result with Independent Load Behaviors

This section shows the result of daily QSTS simulations performed on the full model, the simplified model with independent load compensation and the simplified model without compensation. To generate the compensating gains, we obtain normalized load shapes using a method that is described in Section 3.2 under equation (8). Specifically, we apply this method to the load behavior obtained in Fig. 10. As before, the time step is 30 seconds for water heaters. Loads in 23 segments follow 23 different load shapes generated by the stochastic method.

Fig. 16 shows the voltage of the full model, the simplified model without independent load compensation and the simplified model with independent load compensation at one specific segment. The voltage error is calculated by Equation 2.18:

$$Error = \frac{abs(V_{Full} - V_{Simplified})}{V_{base}} \tag{2.18}$$

Fig. 17 is the plot of the average error of the 23 segments. The proposed simplification method generates a smaller overall error. It outperforms the old simplification method during the time that the loads are at lower level, and the error during the evening peak is also reduced to an acceptable range. Without independent load compensation, the average error ranges from about 0.01 to 0.02 pu, or 1.2 to 2.4 volts on a 120-volt base, which is unacceptable for studies in which the voltage is important. For example, the worst error exceeds typical bandwidth settings on line voltage regulators. With independent load compensation, the worst average error is about 0.007 pu, or about 0.8 volts on a 120-volt base, which is near the limit of acceptability. In this example, the simplified model is useful for voltage studies only with the independent load compensation.

The simulation result can be affected by different window sizes and percentile values for load behaviors as discussed around Fig. 9. Here we used several sizes of windows to process the stochastic load, from 1 minute, 2 minutes and 5 minutes up to 50 minutes window length. A specific percentile load value was taken from each window (e.g. 75%) to represent the load value of all the other time points inside the associated window.

Fig. 18 and Fig. 19 show some simulation results of the effect of different window sizes and percentile values applied on the stochastic load behaviors for the simplified model. Since the estimations become less precise with larger window size, the errors and their standard deviations

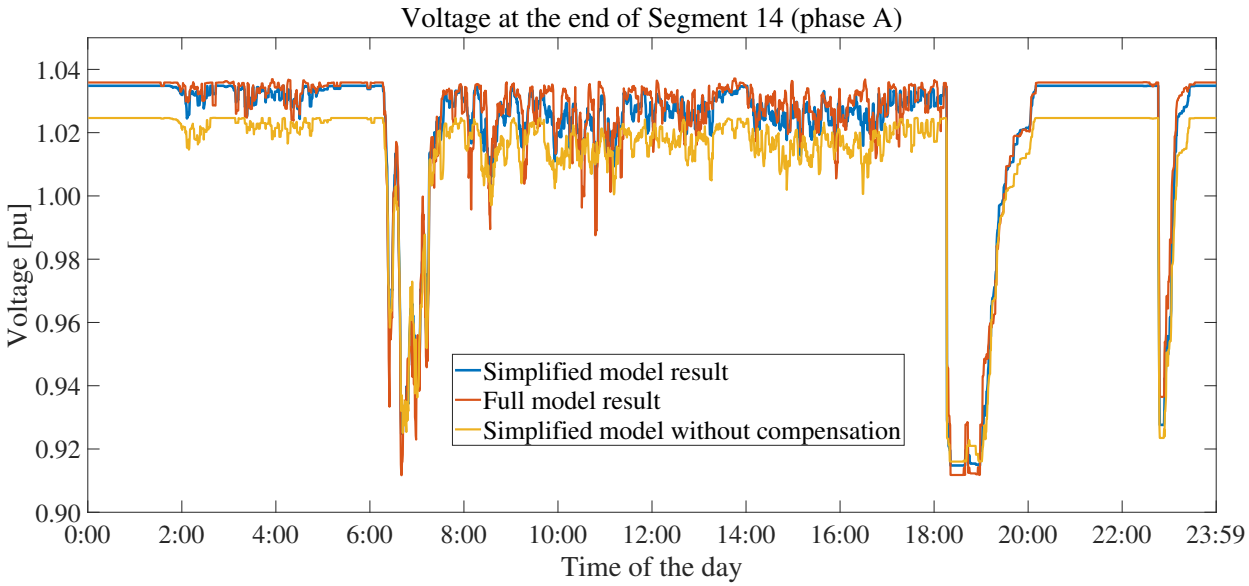


Figure 16: Voltage at the end of segment 14 .

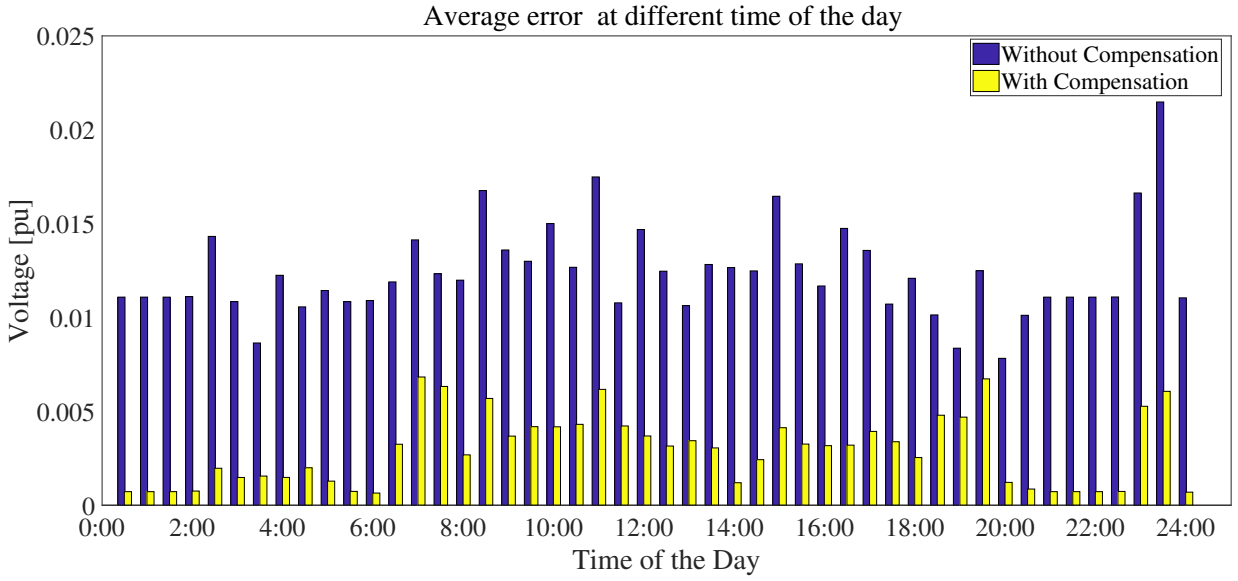


Figure 17: Average voltage error vs. time of day with or without independent load compensation on the simplified feeder model.

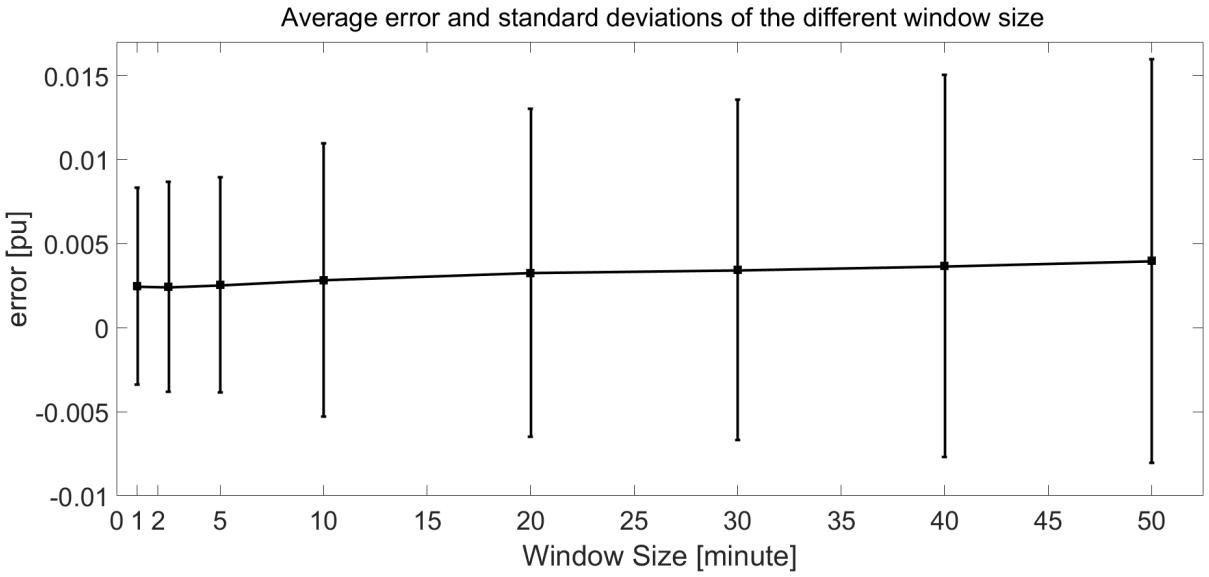


Figure 18: Average error and its standard deviations (averaging among all buses and times of day) vs. window size (75 percentile value for each window).

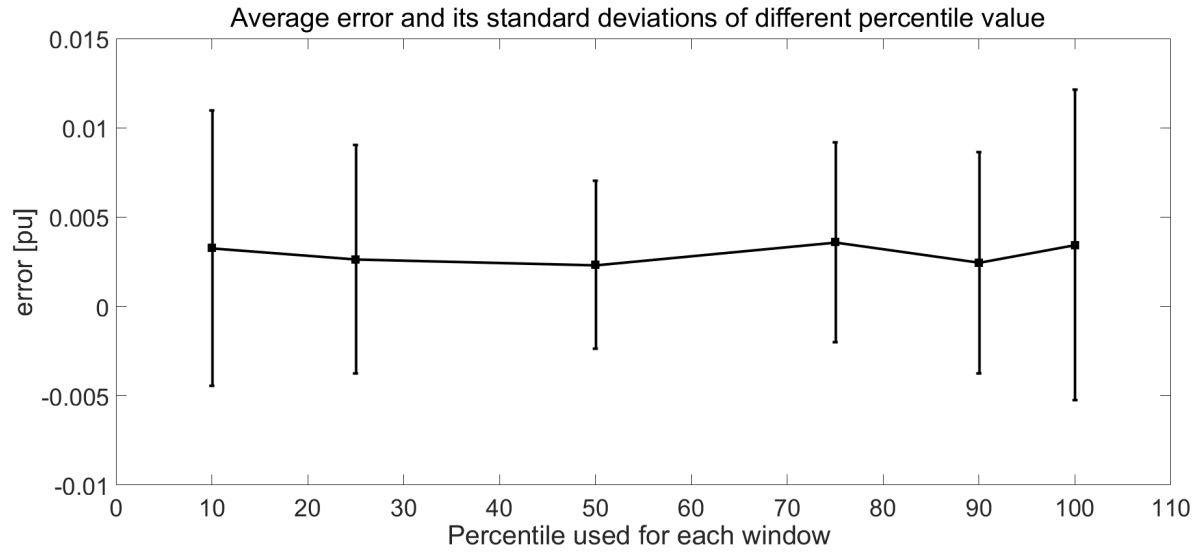


Figure 19: Average error and its standard deviations (averaging among all buses and times of day) vs. percentile value for 5-minutes window.

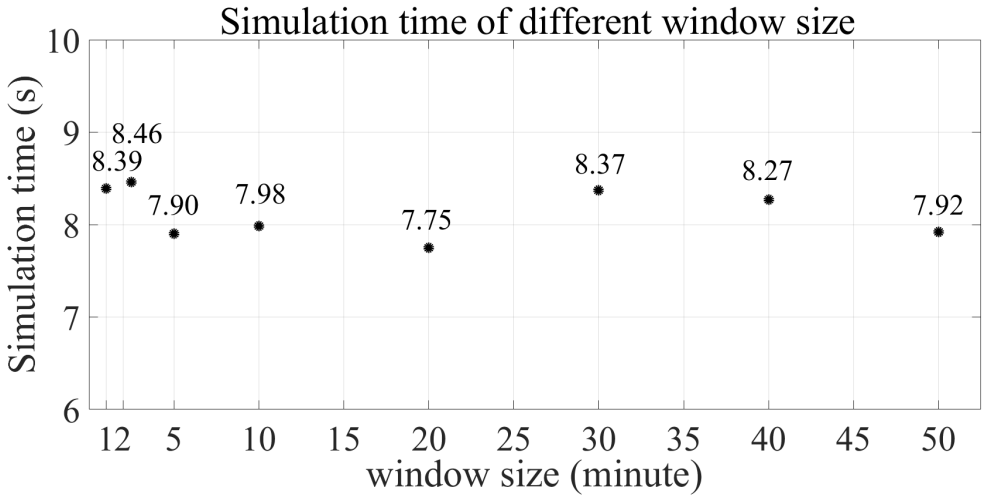


Figure 20: Proposed method's simulation time at different window sizes.

tend to increase as the window size increases. The errors of different percentile values are empirical results, and the smallest error may occur at different percentile values in a different case. Fig. 20 shows that the simulation time of the proposed method remains about the same with different window sizes. Therefore, it is better to use a smaller window size for more precise estimates.

Table 2: Average error with standard deviation

Error (p.u.)	Proposed method	Previous method	Improvement
Phase A	0.0021±0.0047	0.0132±0.0089	6 times
Phase B	0.0018±0.0048	0.0081±0.0083	4 times
Phase C	0.0028±0.0057	0.0116±0.0069	4 times

The best result with the smallest error is summarized in Table 2. The new method's error is 4 to 6 times smaller than the previous method's error. The maximum average error is now approximately 1/3 volt on a 120-volt base; this should be good enough for system optimization functions, such as conservation voltage reduction accounting for stochastic load behavior. The previous maximum average error was approximately 1.6 volts; this might have been good enough to detect voltage violations, but not for system optimization.

2.4.4 Computation Time Reduction

The computational savings factor is defined as:

$$CSF = \frac{T_{Full} - T_{Simplified}}{T_{Full}} \tag{2.19}$$

Where T_{Full} is the benchmark simulation time of the full model, $T_{Simplified}$ is the simulation time required by the simplified model. The result is shown in Table 3.

Table 3: Time Reduction

Metric	Full Model	Proposed Method	Previous Method
Simulation Time	21.352s	7.619s	0.867s
CSF		0.643	0.959

The proposed new method required more time to simulate than the previous method because it contains many more load compensation sources in the model, see Fig. 8. Nonetheless, it improved the accuracy and reduced the QSTS power flow simulation time by 65%, producing much smaller error under the condition of independent temporal loads.

3.0 Model Order Reduction for Transactive Elements

In the recent years, with the widespread application of advanced information and communication technologies, buildings and household appliances have become more intelligent, having the potential to operate more efficiently to achieve higher cost savings by participating in demand response (DR) programs. Transactive energy (TE) extends demand response to operate on faster time scales with multilateral market participation by responsive loads [7]. PNNL has developed a Transactive Energy Simulation Platform (TESP) for TE system simulations with the transactive market and new control mechanisms [2]. Accurate and efficient modeling of the end-use load in a large distribution system is important for studying the aggregate dynamic behaviour and its impact on the bulk system. Previous work on model order reduction for distribution systems with non-responsive loads are introduced in [57], but that approach did not consider the reduction of responsive loads. For the large-scale and complex co-simulation environment of transactive systems in TESP, it is less computationally expensive and more efficient to use aggregated models of responsive loads to evaluate top-level design approaches and to analyze its impacts on the bulk system.

As shown in Fig. 21, TESP includes a distribution simulator GridLAB-D [58], a transmission simulator MATPOWER [59] and a building simulator with multiple transactive agents, and the integrating Framework for Network Co-Simulation (FNCS) [37] that manages the message exchange among different simulators and transactive agents. TESP simulations with multiple distribution feeders in a larger system are time consuming. A single distribution feeder in TESP may have thousands of houses. Each of these houses uses an equivalent thermal parameters model in GridLAB-D to generate the load consumption profiles for both responsive and unresponsive loads [60]. These loads/appliances follow different schedules, thereby representing different behaviours of each end-user. In the TESP, each residential house is a demander which submits a bid, including price and quantity, into a double auction market. The HVAC in a house, as an example of the transactive loads, participates in the market as a transactive element with a thermostat controller agent. The thermostat controller agent subscribes to the air temperature, HVAC power state, and the HVAC power from GridLAB-D. The agent then uses this information to formulate a bid for the

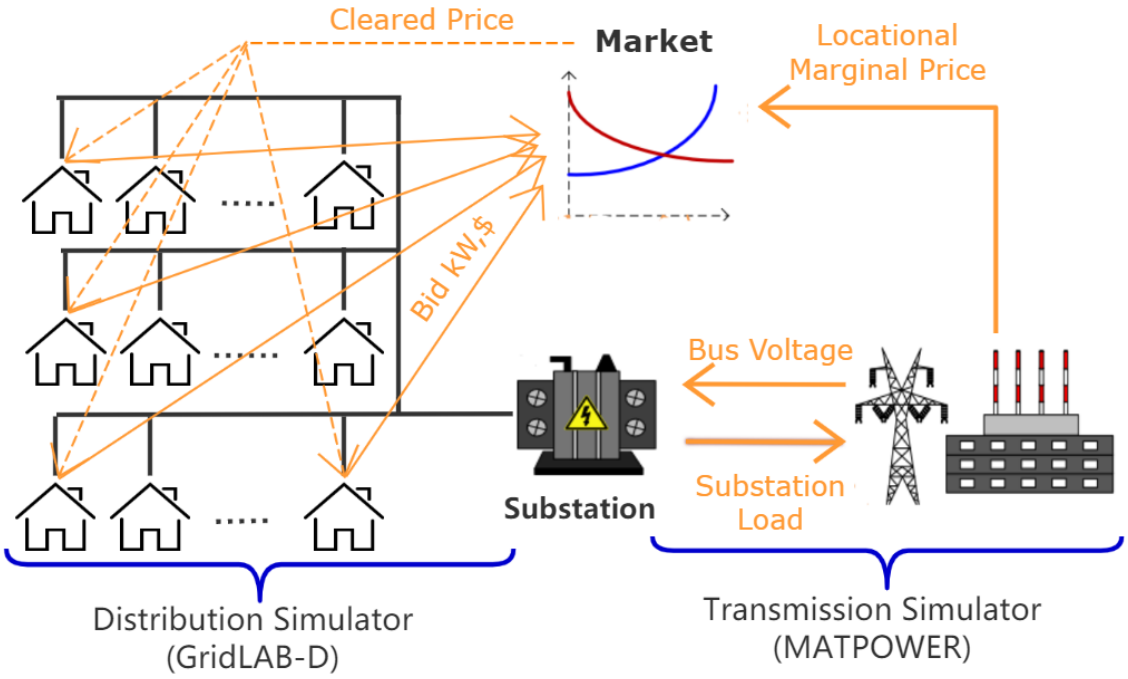


Figure 21: TESP architecture

next cleared market value. After receiving the clearing price from the double auction market, the thermostat controller agent adjusts the temperature setpoint based on the cleared price and the base schedule [31]. Simulation platform like TESP have to model the transactive behaviors (bidding and responding) of over 10 thousands of houses for a larger system, which will lead to days of simulation time. A reduced order model that has the similar transactive function and behaviors to the full model can dramatically reduce the simulation time and computational burden. We develop and test such a reduced order model in this study.

Here we argue that the simulation time can be notably reduced through an aggregate model with the collective bidding and responding ability. Therefore, accurate aggregation of large numbers of end-use loads is needed to account for the transactive behaviours and to model interactions among different aggregated loads within large systems. Such a reduced aggregate model, in addition to having lower computation costs and providing faster simulation results, should have also minimum simulation error compared to the full model. Lower errors will ensure that the simplifications and aggregations is achieved while representing the larger system model precisely. Ex-

isting studies have focused on the aggregation of thermostatically controlled loads (e.g. HVACs) since they can actively participate in demand response programs. Aggregated dynamic models are proposed in [9] for thermostatic loads using stochastic diffusion models by Fokker-Planck partial differential equations (PDEs). An aggregate controllable model for a homogeneous population of HVACs and water heaters considering the state transitions is developed in [10]. Data driven approaches based on Markov chains are proposed in [11] [12], these methods compute the transition probabilities among different states based on the equivalent thermal parameter (ETP) model. All the above existing aggregate models have been developed for steady state conditions i.e. when there is no demand response. The system does not consider the dynamic demand response characteristics of transactive interactions and the heterogeneity in the population. More recently, there have been studies on the aggregation of price responsive loads as well. For example, in [13], a method to estimate the aggregate behavior of a group of price responsive loads is proposed to maximize the profit for the retailer/aggregator with optimal bidding price in a day-ahead market. The communication flow between the retailer and consumers is one-directional, the price responsive load of the consumers is estimated by the conditional distribution given a retail price. In [14], an aggregate model for a diverse group of thermostatically controlled loads is proposed, this aggregate model is able to accurately capture the transient dynamics in the collective response under both steady state and severe dynamic conditions, there is no market and bids involved.

Different from the aforementioned approaches, we aim to obtain a simplified structure for simulations in order to reduce the simulation time, and is able to collectively generate bids which accurately represents the aggregate transactive behaviors of consumers in the transactive systems to participate in the electricity market. The reduced order model for TESP requires bidding and responding functions to interact with different simulators, thereby produce similar behavior for the transactive loads to ensure the validity/accuracy of representing the transactive mechanisms of the full system model which was illustrated in Fig. 21. In order to develop a reduced order model with low approximation error for faster simulation in the TESP, an aggregation of different houses is necessary. The aggregate model should have the ability to formulate bids that represent the functions of transactive loads to participate in the double auction electricity market, and to adjust the loads based on market clearing price which represents the feature of price responsive.

In this study, we develop a method to build a reduced order model for TESP. The method is implemented in TESP as an aggregate responsive load (ARL) agent to replace all the detailed house models and accordingly to achieve faster simulation results. The ARL agent utilizes two recurrent neural networks (RNN) with long-short term memory units (LSTMs) to collectively generate bids and adjust the aggregate responsive load in every market clearing cycle. Through simulation results we show that the reduced order model with the ARL agent is able to significantly reduce the simulation time and generate simulation results with low error. Once the ARL agent was trained, it is able to produce simulation results with low error for large number of houses for different days/weeks/months as well as generalize the simulation results for different number of houses. We demonstrate the error characteristics of the reduced order model and its generalization capabilities by comparing it with the full order model.

3.1 Framework of the Reduced Order Model

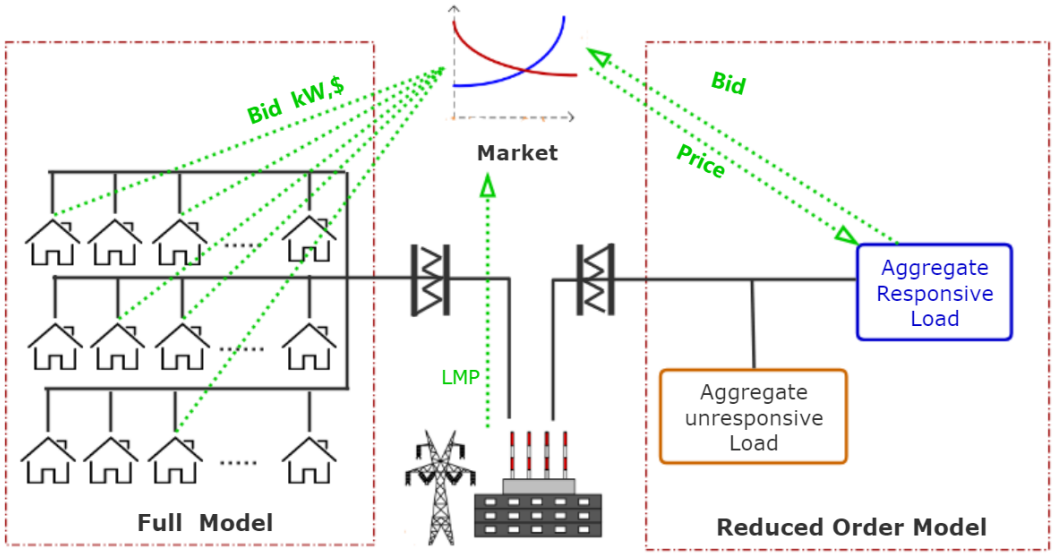


Figure 22: Full model and reduced order model

For comparison, the TESP full model and the proposed reduced order model are shown in Fig. 22. In the full model, every house is modeled in detail with different individual settings (temperature schedule, size) and controllers for transactive elements. Each house formulates bid, and adjusts load based on received cleared price from the auction market, then accordingly changes the temperature setting to adjust the load consumption individually in every market clearing cycle based on occupant's preference. The simulation time will increase with the number of houses. In the proposed reduced order model, the main components is an aggregate response load (ARL) agent which contains two recurrent neural networks with Long Short-Term Memory Units (LSTMs). Instead of hundreds or thousands of houses, only two elements exist in the reduced order model. All the detailed house models which use the equivalent thermal parameter model are removed and replaced by the aggregate responsive load and the aggregate unresponsive load. All the transactive elements in the full model are aggregated and represent by the blue box in the reduced order model as an aggregate responsive load (ARL) agent, see Fig 22. An ARL agent has the ability to submit bids that represent different houses and provide responses to the market clearing price with a price dependent load. In this way, a single ARL agent behaves the same as the transactive elements in hundreds or thousands of houses. All the other loads in the full model which are not price responsive are represented by the aggregate unresponsive load following a certain load profile. With this proposed method, every distribution feeder model in a bulk system can be replaced with a reduced order model. We show that such a model has similar simulation results to the fill model and the error between the outcomes is very low. Moreover, the proposed method is able to reduce the simulation time for the feeder model with larger number of houses. For example, through the proposed method, one can use a full detailed model for one feeder for observation and change all the other feeders to reduced order models.

3.2 Method

In this section, we first introduce the proposed reduced order model. Then we present background information on RNNs and LSTMs, followed by the simulation of transactive load in TESP and the application of RNN with LSTMs in the reduced order model.

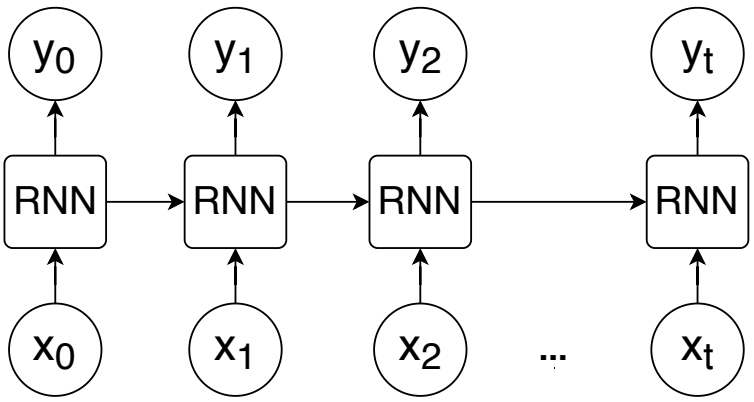


Figure 23: Recurrent neural network structure

3.2.1 Recurrent Neural Network (RNN)

RNN is used as the bid and response neural network in the ARL as described in the previous section. This section briefly introduces RNN and why it is used to imitate the bidding and responding behaviours of transactive elements.

Recurrent neural network is a type of feed forward neural network which has an internal memory [61]. As shown in Figure. 23, RNN performs the same function for every input but the output of the current input is related to the past computations. Every time RNN producing a output, it will be sent back into the recurrent neural network. When generating a output, RNN make use of the current input and also the output that it has learned from the previous input. Unlike the other neural networks where all the inputs are independent from each other, inputs in RNN are related to each other. More specifically, while generating an output, RNN structure makes use of the current input and also the output that it has learned from the previous inputs. Therefore, unlike the other feed-forward neural networks, RNN structures can be used to model correlations among inputs from different time steps and learn dependencies across different outputs and inputs.

However, it was shown that RNN structures have several shortcomings [62] [63]. The training of an RNN could be a difficult process due to the gradient vanishing. If the partial derivation of error is less than 1 and becomes much smaller during iterations, the contribution from the earlier steps becomes insignificant in the gradient for the RNN unit, the long-term dependencies will be ignored during training. If we can not find the gradient, we can not adjust weights in the direction to

minimize error. It is very difficult to find how important some of the remote inputs are to the current output. Also, RNN is not able to process very long sequences. Sometimes the current output needs the information from further back, the gap between the useful information and the point where is it needed can be very large, RNN may not have the ability to connect the information.

3.2.2 Long Short-Term Memory Units (LSTMs)

A variation of RNN called Long Short-Term Memory units is proposed to solve the vanishing gradient problem [64] [65]. LSTMs have powerful memories and are capable of learning long-term dependencies which bring in more flexibility in modeling the input-output relationships. For example, RNNs with LSTMs are widely used for time series prediction [66] [67]. LSTMs help reduce the training error which can be back-propagated through time and layers. Accordingly, LSTMs enable the recurrent neural nets to learn over very long sequences, and make the networks possible to link the error to remote units.

Standard RNN structures include a chain of repeating modules of neural networks as shown in Figure. 23. These modules usually are designed using simple structures including only a single activation layer that uses for example hyperbolic tangent function (tanh). On the other hand, LSTMs have the same chain structure, but every module has a more complicated design with three gates: input, forget and output gates. Through these gates, LSTMs have the ability to update and control/model the information flow. Specifically, the LSTM cell learns when to allow data to enter, leave or be deleted by these gates through the training process, back-propagating error, and adjusting weights via gradient descent.

σ denotes a sigmoid activation function, tanh denotes a hyperbolic tangent activation function. The lower left of the cell operates a concatenation of the new input X_t and the previous output y_{t-1}. The line running through the top of each cell is the cell state of LSTMs. Along the entire chain, LSTMs are able to add or remove information to the cell state by the regulation of different gates as shown in Figure. 24. Different from the normal RNN, LSTM cell has more information, these information can be stored, read and modified by different gates. These gates will open and close to block and pass the signal they received based on the importance, which is determined by the weights that are adjusted by the learning process. The following equations are used to determine

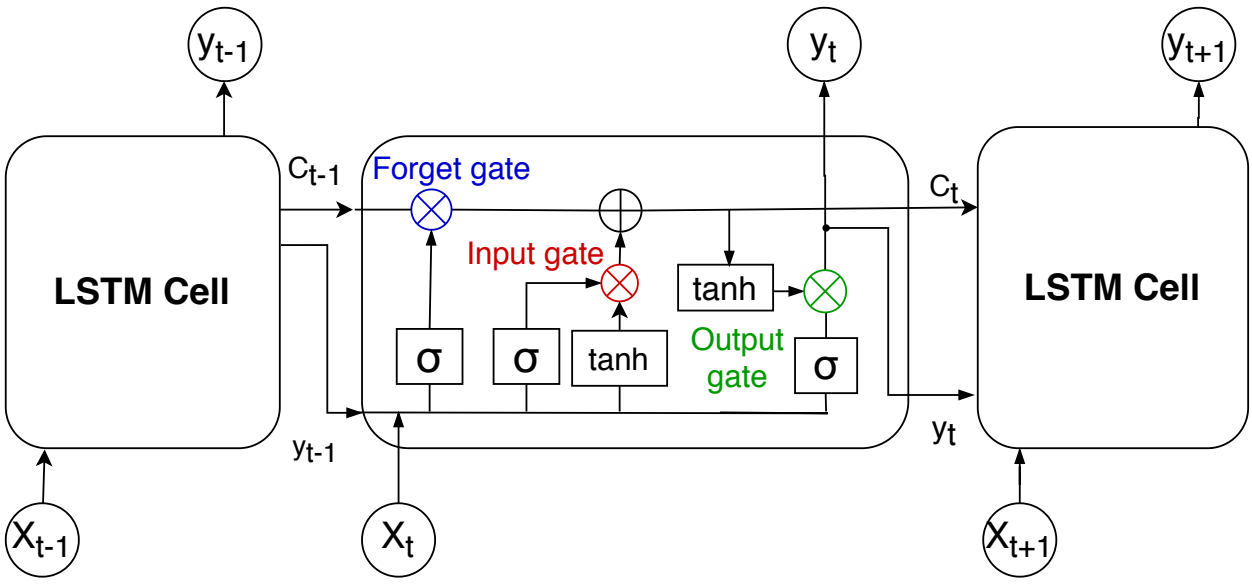


Figure 24: Connection of LSTM cells and its internal structure

the gate state, W represent the weight matrices connecting x_t to different gates, b is the bias term of different gates.

forget gate:

$$f_t = \sigma(W_f \cdot [y_{t-1}, x_t] + b_f) \tag{3.1}$$

The forget gate in Figure. 24 determines what information to be thrown away from the cell state. It will take the previous output and current input, output a number between 0 (delete all) and 1(keep all), then acting on information in the previous cell state C_{t-1} to control it.

input gate:

$$i_t = \sigma(W_i \cdot [y_{t-1}, x_t] + b_i) \tag{3.2}$$

cell input:

$$\widetilde{C}_t = \tanh(W_c \cdot [y_{t-1}, x_t] + b_c) \tag{3.3}$$

The input gate determines which new information is going to be added to the cell state, the sigmoid function will select which value to go through, the tanh function will give the level of importance to these values.

update cell state:

$$C_t = i_t * \widetilde{C}_t + f_t * C_{t-1} \tag{3.4}$$

Multiply f_t with the old cell state C_{t-1} and combined the updated old state with $i_t * \widetilde{C}_t$ from the input gate is the new cell state value C_t. Then information in the new cell state will be used to determine the output.

output gate:

$$o_t = \sigma(W_o \cdot [y_{t-1}, x_t] + b_o) \tag{3.5}$$

A sigmoid function is applied to find which part of the cell state will be output. The cell state will go through a tanh function ranging the value from -1 to 1 and multiply it with the output of sigmoid function to get y_t.

$$y_t = o_t * \tanh(C_t) \tag{3.6}$$

LSTM cell structure helps error to be back-propagated through time and layers. By maintaining a more constant error, LSTMs make it possible for RNN to learn over many time steps by opening a channel to link causes and effects remotely. Due to the powerful features and its recurrent nature, a single layer of LSTM units can be considered as a deep neural network.

3.2.3 Components and Functions of the Aggregate Responsive Load (ARL^a)

Figure. 25 is an enlarged model of ARL. There are two main components in the ARL, the bid RNN and the response RNN. The bid RNN manages to generate bids for the given input features which are related to the bidding formulation in the original TESP mechanisms. ARL then sends these bids to the market shown as (1) in Figure. 25, and the market agent will send a cleared price back to ARL after each market clear. Then the response RNN will receive the cleared price as one of the input features, combined with some other input features, the response RNN generates transactive loads for individual houses shown as (2) in Figure. 25. The summation of these transactive loads is used to represent the aggregate responsive load in the feeder shown as (3) in Figure. 25. The two main functions of the proposed ARL are:

1. Formulate bids (bid RNN):
    a. Obtain the previous market cleared price.
    b. Generate bids(price, power) for a given number of houses.
    c. Send the bids to the market agent.
2. Adjust loads based on cleared price (response RNN):

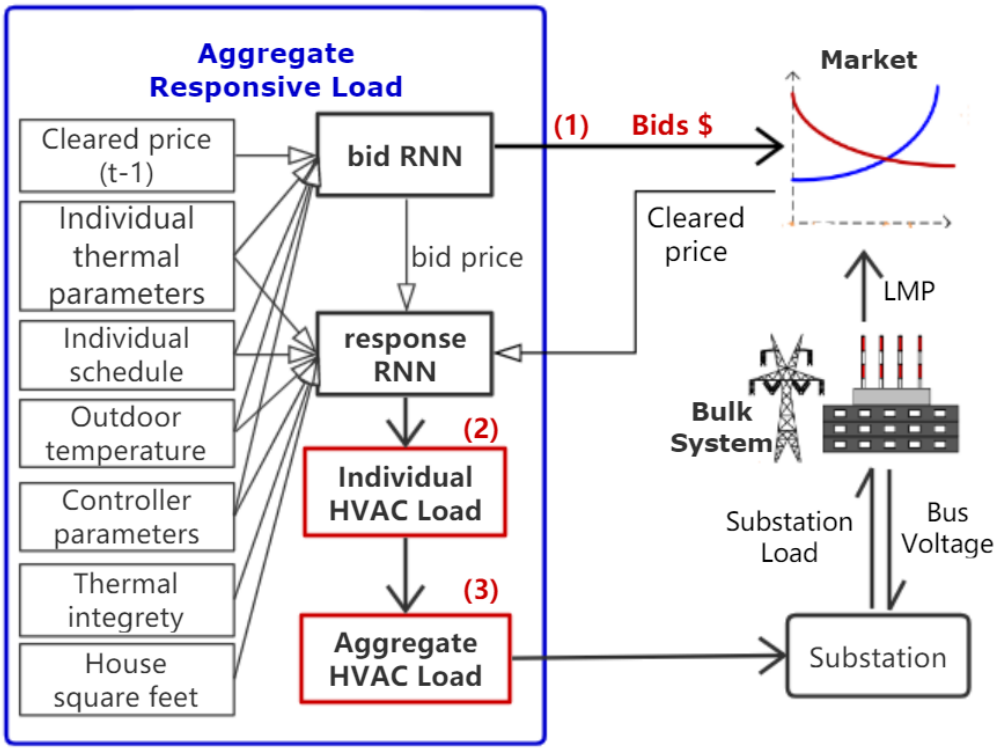


Figure 25: Detailed model of the aggregate responsive Load

a. Receive the cleared price from the auction market.

b. Determine the aggregate transactive load based on the cleared price.

c. Adjust the load value in GridLAB-D.

Equation 3.7 is how every single house in TESP formulate individual bid price. Where $P_{average}$ is the mean value of historical price, $k_{high/low}$ is a bidding ramp denominator, σ_{price} is the standard deviation of historical price, $\left|T_{\max/\min}\right|$ is the allowed range of set point variation. These factors can be summarized as the historical price, and individual thermal settings.

$$P_{bid} = P_{average} + \frac{(T_{room} - T_{schedule}) \times k_{high/low} \times \sigma_{price}}{\left|T_{\max/\min}\right|} \tag{3.7}$$

After sending bids to the market, the substation will receive a cleared price and send back to the HVAC controller of each house. Every house then adjust the temperature based on Equation 3.8 so that the transactive load (HVAC load in this case) is changed in responding to the cleared price.

$$T_{set} = T_{schedule} + \frac{(P_{cleared} - P_{average}) \times \left|T_{\max/\min}\right|}{k_{high/low} \times \sigma_{price}} \tag{3.8}$$

From the above equations, it can be observed that the bidding of HVAC is related to the thermal environment of each house and the price information. Since the thermal environment of a house is highly related to the previous states, the bidding price of the individual HVAC can be treated as a sequence of data which has dependency over different time steps. Specifically, a bidding price at time t could be affected by the bid price, cleared price of some time earlier in the same day. Therefore, we take the key factors that contribute to the formulation of bid price as input features to train an RNN to learn the function of bidding. After training, one RNN can be used to obtain the bid price for different houses with different input parameters such as the cleared price from the previous time step, the pre-defined temperature schedule and the thermal parameters of houses etc.. As mentioned in Section 3.2.2, by the RNN with LSTMs, not only the current inputs but also important information like the thermal environment from historical inputs will be considered when deciding the output. It allows the ARL agent have similar behaviors when formulating the bidding price as the original simulation.

Similarly, RNN can be applied to perform the response-to-price function of the transactive elements in the same way. The thermal state parameters of a house are key factors to determine the HVAC power consumption. The room temperature of a house is determined by the HVAC power output and the thermal environment of the previous time step. Thus, the load of each individual HVAC is also a sequence of values that has time series relations. Taking the cleared price as an input, together with thermal parameters and schedule of different houses, an RNN can be trained to generate the HVAC load given a cleared price.

In the reduced order model as shown in Fig. 25, two main components, bid RNN and response RNN keep the same functions as described in Equation 3.7 and 3.8. These RNNs are trained by the simulation data of different houses from the full model simulation in TESP. After training, the ARL agent is able to generate rational simulation result of different days with different weather and schedule inputs. Input features for bid RNN and response RNN are listed in Table 6 as ARL^a. Note that the thermal integrity (TI) and the size of the house (SF) are easily obtained while U_a, C_a, C_m, H_m are difficult to acquire. These parameters are readily available from GridLAB-D simulation, but in real life they might have to be estimated [68].

Input features for different networks are listed in Table 4 and Table 5.

Table 4: Input features for bid RNN

Input Features of Bid RNN	
Cleared price(t-1)	$P_{cleared(t-1)}$
Outside Temperature	T_o
Temperature Schedule	$T_{desired}$
Controller ramp	$Ramp$
Temperature range	T_{range}
Thermal envelope of the house	U_a
Total air mass	C_a
Total thermal mass	C_m
Interior mass surface conductance	H_m

Table 5: Input features for response RNN

Input Features of Response RNN	
Cleared price(t)	$P_{cleared}$
Outside Temperature	T_o
Temperature Schedule	$T_{desired}$
Controller ramp	$Ramp$
Temperature range	T_{range}
Thermal envelope of the house	U_a
Total air mass	C_a
Total thermal mass	C_m
Interior mass surface conductance	H_m
Thermal integrity	TI
Size of the house	sqft

With RNN and LSTMs, not only the current inputs but also important information like the thermal state from multiple previous inputs will be considered when deciding the current output. It makes the ARL model have similar features as the original simulation. The design of the ARL agent is listed as follows :

ARL agent attribute:

1. House number : the number of houses with transactive HVAC controller
2. Bid RNN : a recurrent neural net which generate bids
3. Response RNN : a recurrent neural net which generate loads in response to cleared price
4. Unresponsive load : all the unresponsive load in the feeder in kW
5. Cleared price : the cleared market price of the current market clearing cycle $/kwh

ARL agent functions:

1. Inform_bid : set the cleared price attribute.

   Arguments: price

2. Generate_bid: output the list of bidding price,bidding quantity and the states on HVACs for the given time step.
   Arguments: day, t, input tensor (input features for bidding)
   Return: bids

3. Set_hvac_load : set the aggregate HVAC load attribute
   Arguments: day, t, input tensor (input features for response RNN)

3.2.4 Aggregate Responsive Load for Different Number of Houses (ARL^b)

In TESP, as the total number of houses increases, the load on this feeder may have a minor impact on the bulk system, which could lead to congestion and thus affect the market cleared price. Examples of bulk market operations can be found in [69]. On the other hand, with more houses, there is more demand on the feeder which lead to higher market clearing price, and consequently higher bidding price for some consumers during certain time of the day. The increasing number of houses may also lead to congestion during peak demand periods. Due to these factors, the bidding behaviors for the same house can be different when the total number of houses in the feeder changes. Consequently, the individual HVAC load behavior is also affected by the total number of houses, and the aggregate HVAC load will not simply scale up with house number which can be observed from the TESP simulation result in Section 3.4.2 . Here, we propose we extend the ARL model from Section II.D to capture the bid and response behaviors affected by the total house number and produce simulation results of any given number of houses after training.

Fig. 26 is a variation of the ARL agent which has several different input features for the bid RNN and response RNN. The main components and the steps (1) (2) (3) are the same as described for Figure. 25 in Section.3.2.3, but the inputs for the RNNs are different. Specifically, the bid RNN and the response RNN both take the total house number and the behaviors from case n_{max} as additional inputs to learn the impact of house number on both bidding and price responsive behaviours. All input features of the RNNs are listed in Table 6 as ARL^b. For the training of networks in Fig. 26, TESP simulation result of different cases with total house number $n = \{n_1, n_2, n_3, ..., n_{max}\}$ are used. In the modified ARL agent, the networks learn the bid and response behaviors of case $n_1, n_2, n_3, ...$ from case n_{max}. Comparing with the networks in Section 3.2.3 which generalize the

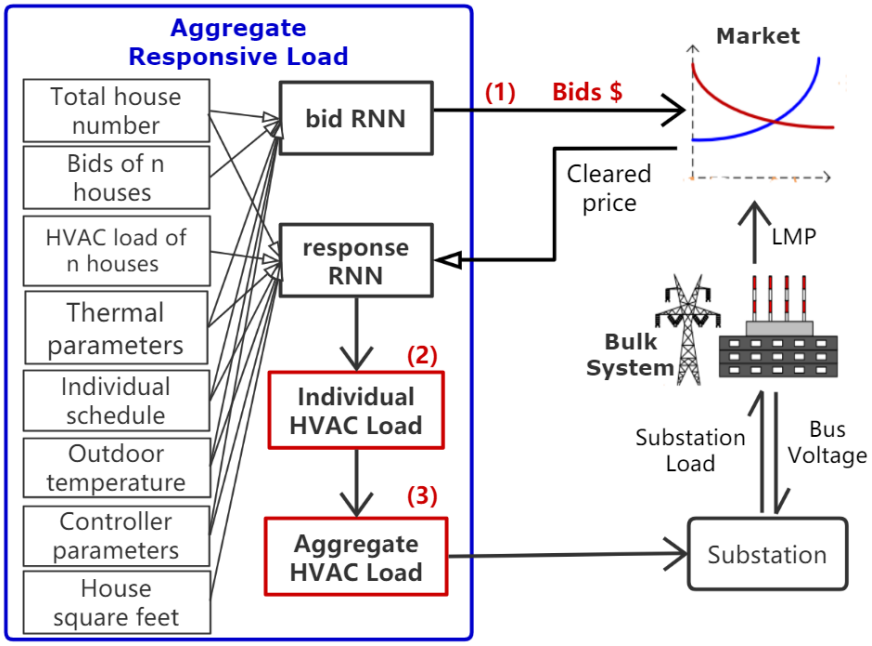


Figure 26: A variation of the bid and response networks

result of different days, the changes in Fig. 26 enable the ARL agent to have better performance in the generalization of simulation cases with different numbers of houses. The inputs of different ARL are summarized in the Table 6 below.

Table 6: Input features for bid RNN and response RNN

Input Features	ARL^a		ARL^b	
	Bid RNN	Response RNN	Bid RNN	Response RNN
Cleared price	$P_{cleared}$	$P_{cleared}$	$P_{cleared}$	$P_{cleared}$
Outside Temperature	T_o	T_o	T_o	T_o
Temperature Schedule	$T_{schedule}$	$T_{schedule}$	$T_{schedule}$	$T_{schedule}$
Controller ramp	$k_{high/low}$	$k_{high/low}$	$k_{high/low}$	$k_{high/low}$
Temperature range	$T_{\max\,/\,\min}$	$T_{\max\,/\,\min}$	$T_{\max\,/\,\min}$	$T_{\max\,/\,\min}$
Thermal envelope of the house	U_a	U_a	U_a	U_a
Total air mass	C_a	C_a	C_a	C_a
Total thermal mass	C_m	C_m	C_m	C_m
Interior mass surface conductance	H_m	H_m	H_m	H_m
Thermal integrity		TI		TI
Size of the house		SF		SF
Total house number			n	n
Bids of n_{max} houses			Bid	
HVAC load of n_{max} houses				$Load$

3.3 Performance of the Proposed Method

In this section, functions and performance of the proposed two RNN with LSTMs networks are shown separately. The bid RNN is responsible for generating bids and the response RNN is responsible for determining the load value of transactive elements (HVACs in this case).

Figure. 27 is the performance of the bid RNN compared with the original simulation result with detailed house model in TESP. Different sub-figures show the HVAC bidding price of different houses during a 24 hours time period. It can be observed that the bidding price of different houses have a similar trend but some are far from each other at certain time of the day. This is due to the slightly different schedule and some random parameters of the house model. With the bid RNN, we are able to generate very similar bid price for different houses by a single network. Some of the small fluctuation is not captured very well but the overall accuracy is acceptable for representing the bidding function of HVAC controller and house model in the original TESP simulations. Similarly, Figure. 28 shows the training result of the bidding price for 1500 houses.

Once the ARL generate the bids of different house at every market cycle, the market agent then collects all the bids from the ARL agent and formulate a bidding curve which accumulates the sets of price and quantity. The order of the curve is descending by the bidding price. In Figure. 29, we compare the bidding curves generated by bid RNN and the original HVAC bids in TESP full model at different time of the day. The bid RNN is able to reproduce very similar result for most of the time, but failed to match the original result at the end of the day. This is due to the reason that the bid price obtained by RNN is not very well matched. As shown in Figure. 27, some peak values between time step 200 and 300 are not captured by the RNN. Therefore, some of the bidding price generated from RNN is lower than TESP, and the RNN bidding curve is far below the TESP bidding curve as can be observed in the bottom part of Figure. 29.

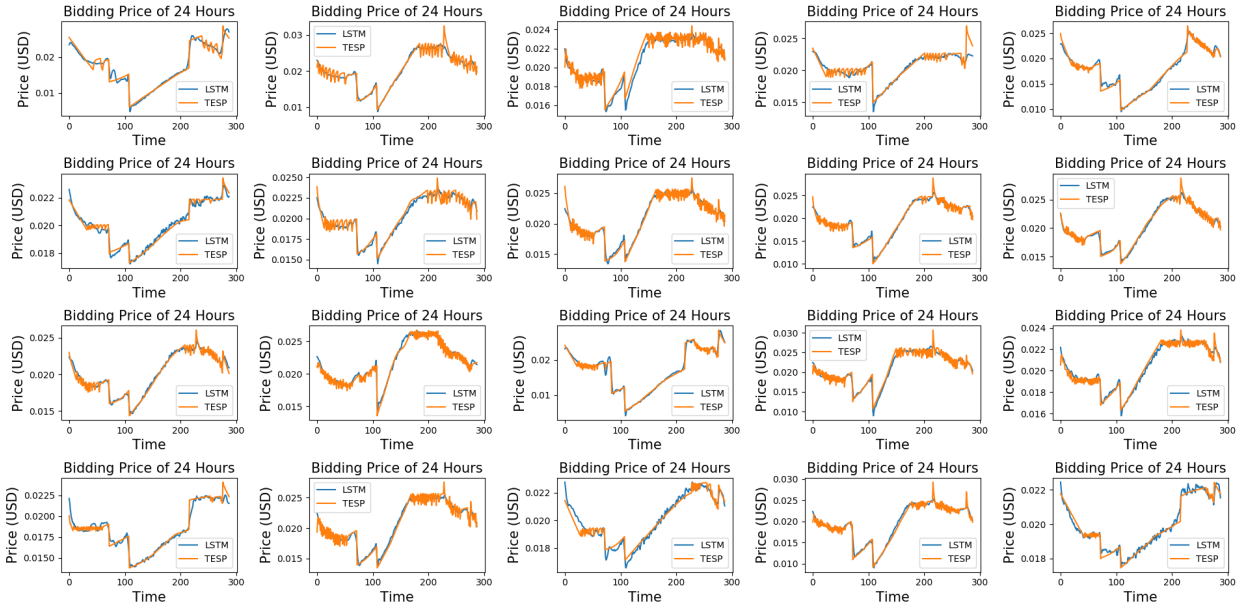


Figure 27: (1) Bidding price of individual house generated from LSTMs (306 houses)

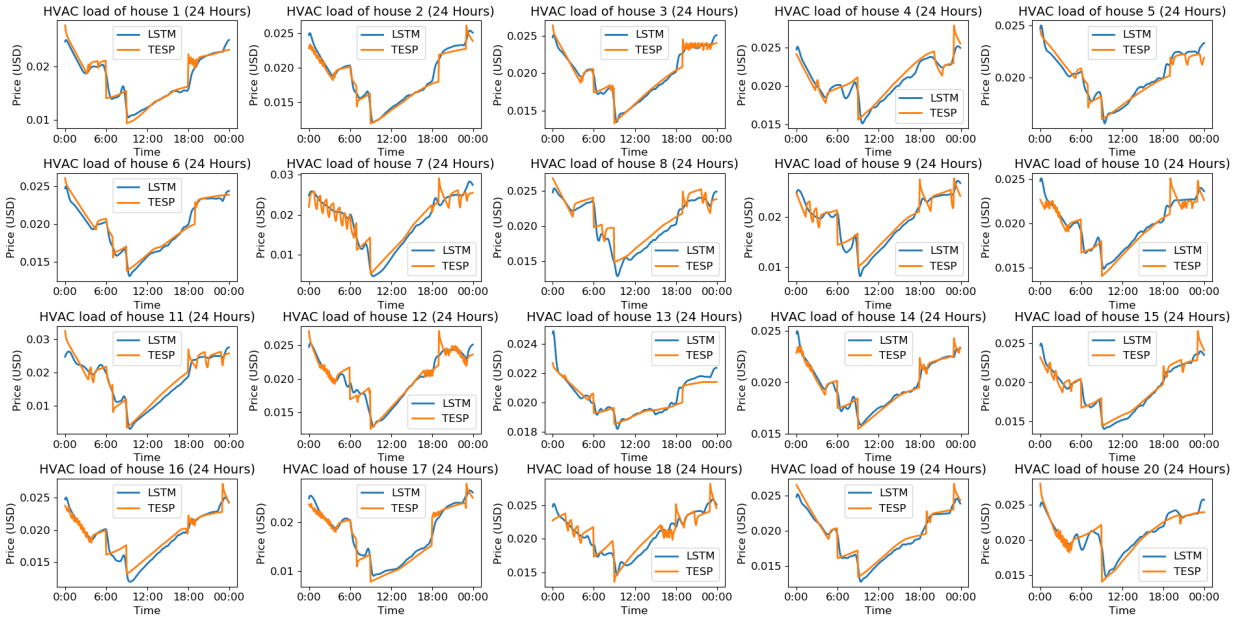


Figure 28: (1) Bidding price of individual house generated from LSTMs (1500 houses)

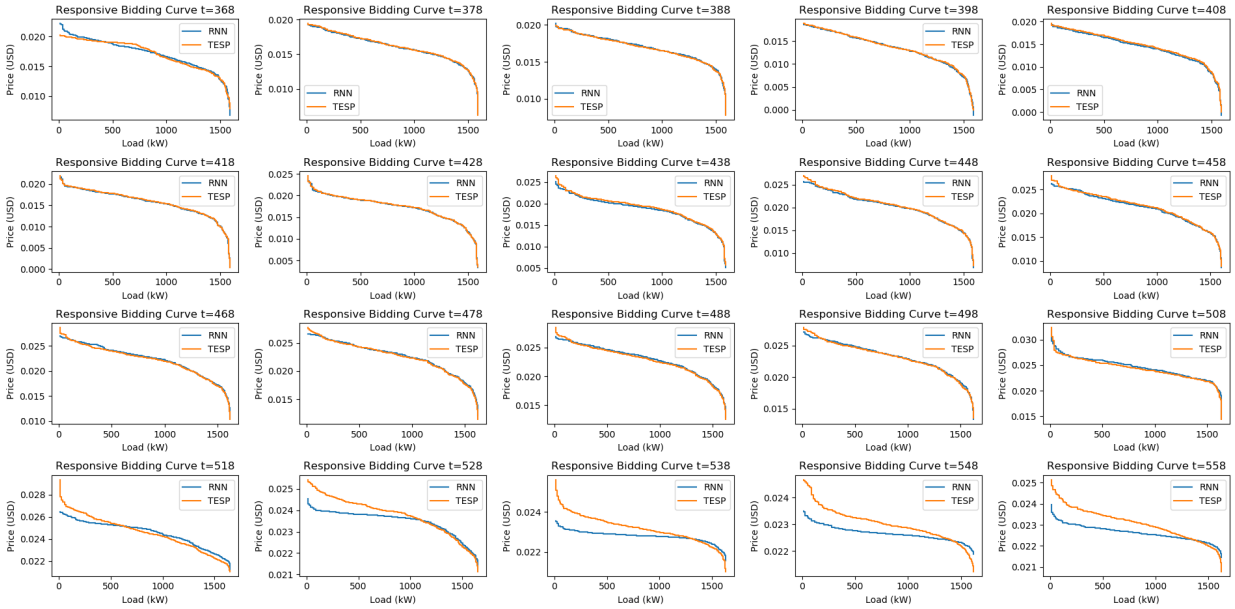


Figure 29: Bidding curve comparison (306 houses)

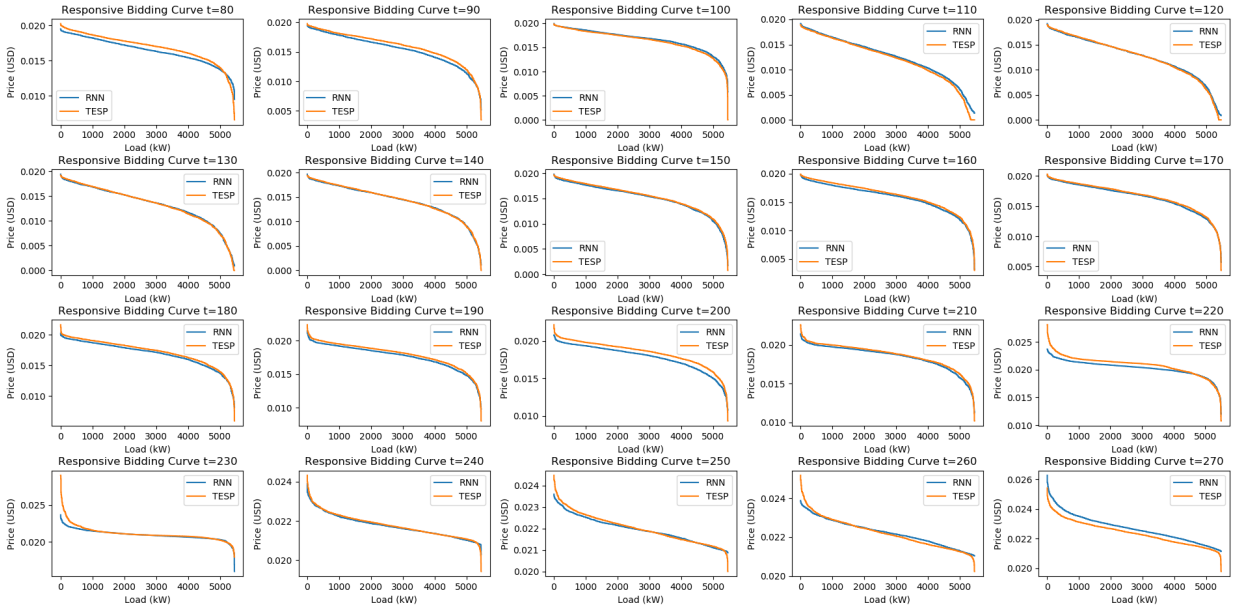


Figure 30: Bidding curve comparison (1500 houses)

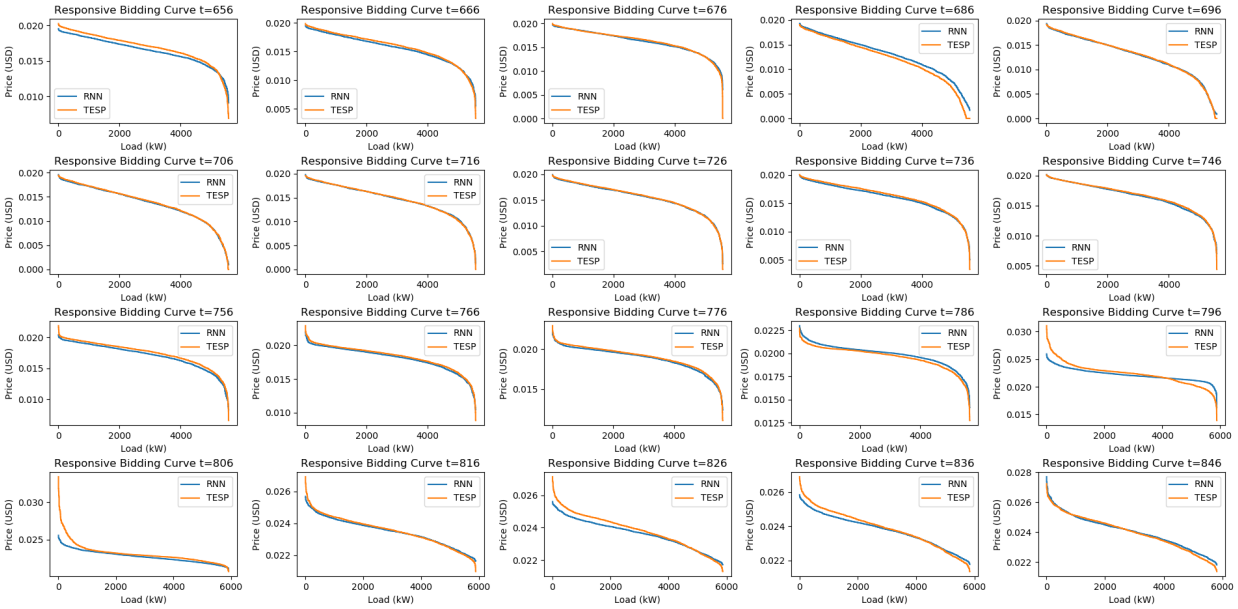


Figure 31: Bidding curve comparison (1500 houses)

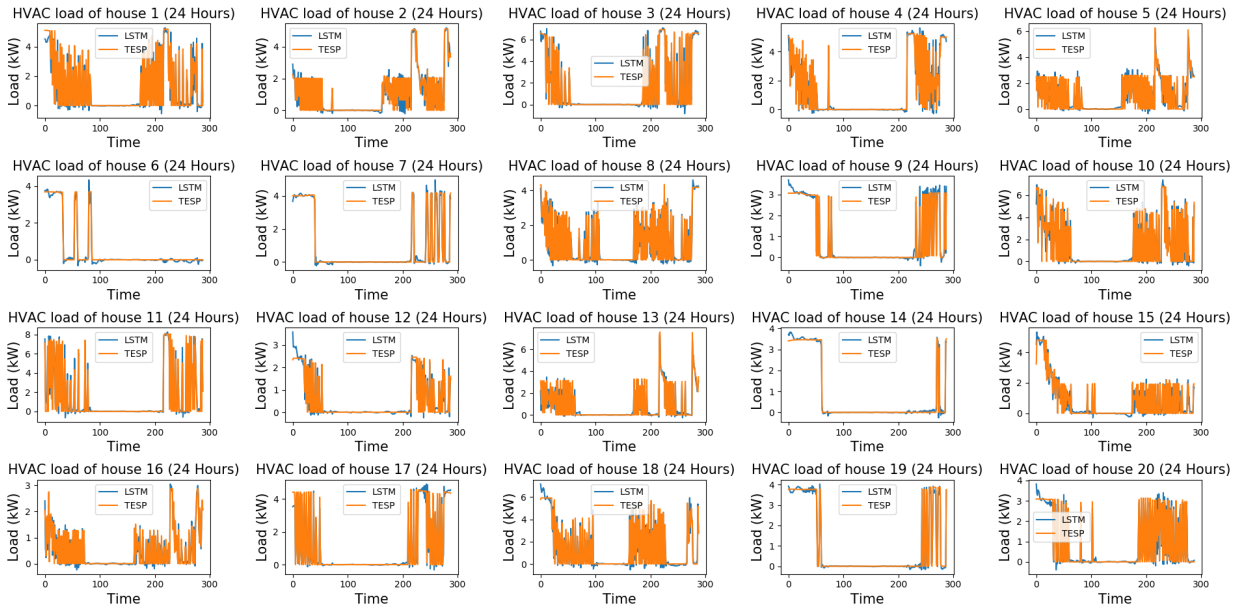


Figure 32: HVAC load of individual house generated from LSTM given the cleared price

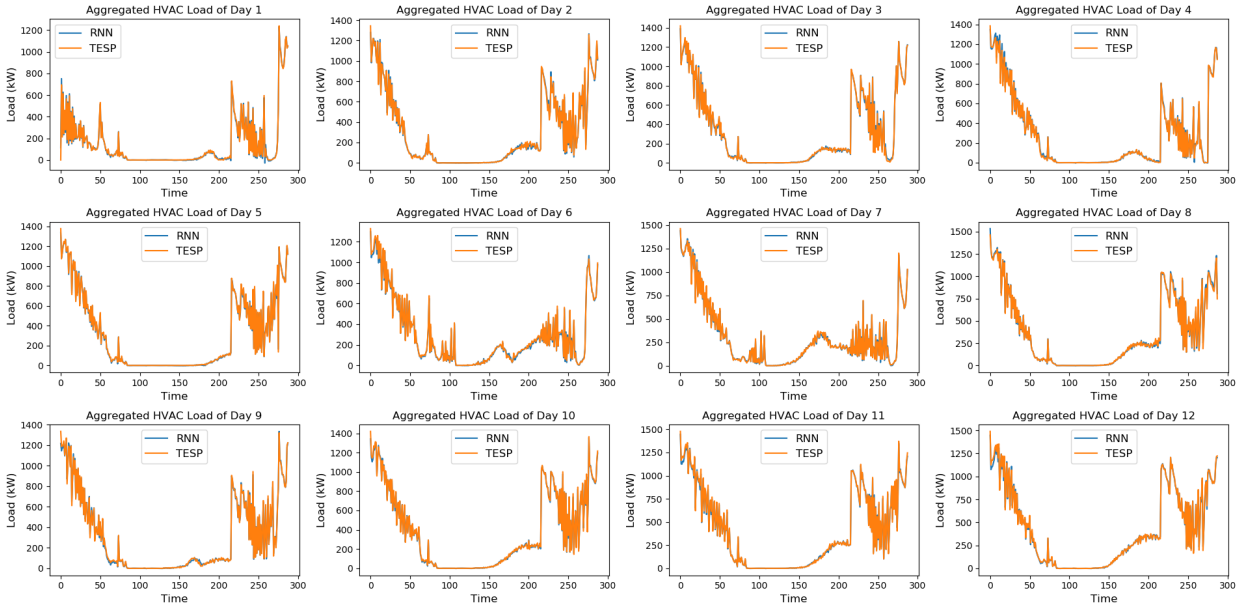


Figure 33: Aggregate HVAC load given the cleared price (306 houses)

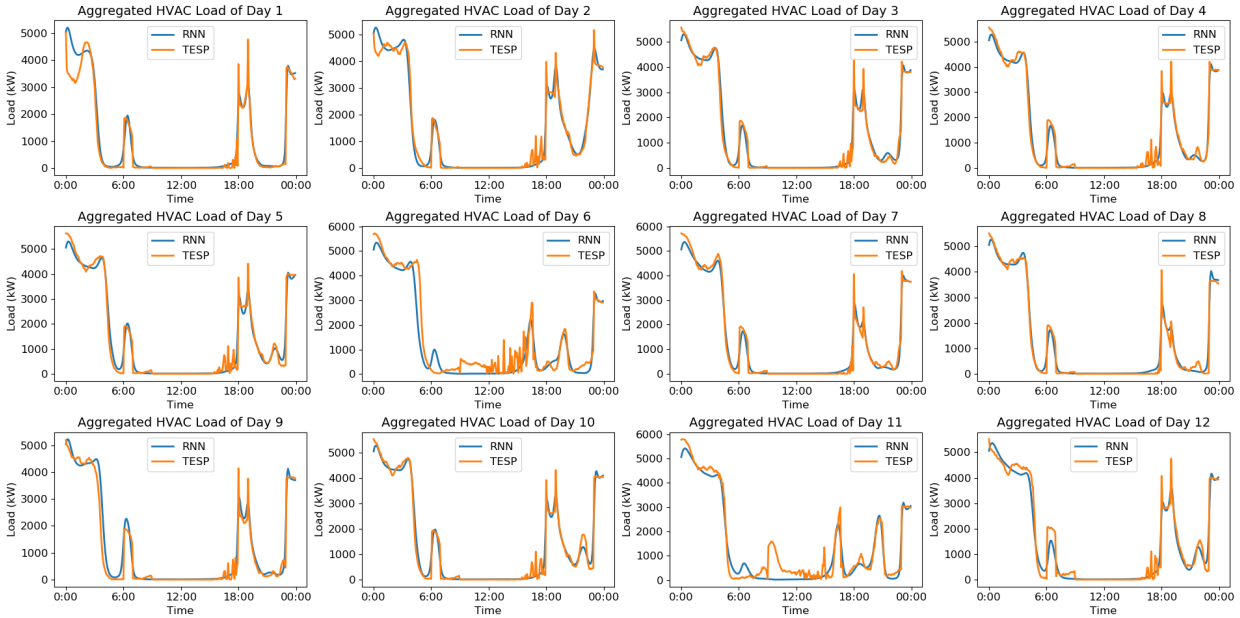


Figure 34: Aggregate HVAC load given the cleared price (1500 houses)

Figure. 30 and Figure. 31 shows the comparison of bidding curve for 1500 houses at different time of the day between the TESP simulation result and the bid RNN result. The bidding curves are well matched for most of the time. When there is capacity limit, a vertical line at the capacity limit will cut through the bidding curve, the price value at the intersection of the bidding curve and the vertical line will be the cleared price for the current market clearing cycle. Therefore, there will not be a significant difference between the cleared price of the reduced order model and the full model if the bidding curve is well matched.

Figure. 32 shows the HVAC load of each individual house of a day generated by the response RNN after training, with the cleared price from TESP as one of the input features, a response RNN can produce similar load behaviours of the HVAC in different houses. It can be observed that the load behaviours of different houses have big difference due to the temperature schedule and thermal parameters. The response RNN is able to learn to adjust the load value of different houses by the input features listed in Table 5.

As described in Figure. 25. The ARL needs to collect all the HVAC loads to obtain an aggregate transactive load. This step can be done by adding up all the HVAC loads at different time steps. Figure. 33 shows the result of aggregate HVAC load of different days. These are the results using the training data, the aggregate HVAC load generated by the response RNN is very close to the original simulation result in TESP with the detail house models.

Figure. 34 shows the aggregate HVAC load of 1500 houses of different days generated by the response RNN, the results are compared with the TESP full model simulation result. The aggregated load are smoothed out due to the increasing number of houses. The response RNN is able to generate similar 1500 houses result as the TESP benchmark.

3.4 TESP Simulation with the Reduced Order Model

In this section, we describe the simulation scenarios and present numerical results. The performance of the proposed reduced order model is compared with the simulation result of a full model. The reduced order model is implemented in TESP and it communicates with the market agent and substation agent as shown in Figure. 21. All the houses are removed in the reduced or-

der model, replaced by an aggregate responsive load (ARL) agent and an aggregate unresponsive load in GridLAB-D. Aggregate unresponsive load follows a given temporally changing load shape representing all the loads which are not transactive. Aggregated responsive load represents all the transactive loads and has two functions, generating bids for a given number of houses and adjusting the aggregate responsive load (total HVAC load) based on the cleared price of the market. Note that the transactive load in this simulation is HVACs, other responsive load such as water heaters can also participate [70]. The bid and response RNNs in ARL use updated cleared price from TESP simulation in real time as one of the input features as shown in Table 6.

Benchmark simulations in TESP take TMY3 data of Miami, Florida as the weather input [71]. The simulation data are generated for different cases with 600, 900, 1000, 1400, 1500 houses for training ARL. The training of ARL^a as introduced in Section 3.2.3 and ARL^b as described in Section 3.2.4 use different training data, details are described in the following sections.

In different simulation cases, the parameters of individual houses are different as summarized in Table 7. There are three kinds of houses with different mean square footage. For different regions of the USA, the proportion of each type of the house is different. Singe family houses range from 2209 sq.ft. to 2951 sq.ft.; apartments range from 798 sq.ft. to 901 sq.ft.; and mobile homes range from 1054 sq.ft. to 1093 sq.ft.. The HVAC setpoint variation range $T_{\max\,/\,\min}$ ranges from 2°F to 4°F. The bidding ramp of an individual HVAC controller $k_{high/low}$ ranges from 0.5 to 3. The thermal integrity of different houses are all uniformly chosen from the set: {very little, little, below normal, normal, above normal, good, very good}; these values are associated with the insulation R-values of the thermal envelope of the house which affects inside-outside air exchange rates. The cooling set points of the HVACs are randomly chosen within different ranges; these ranges change regionally. In our simulations, for this set point we a range from 65°F to 85°F. The detailed parameter definitions are available on github[1].

[1] https://github.com/pnnl/tesp/blob/master/src/tesp_support/tesp_support/feederGenerator.py

Table 7: Simulation parameters of different houses

House parameter	Range / Set
$T_{\max\,/\,\min}$ (setpoint variation range)	2°F $\sim$ 4°F
$k_{high/low}$ (bidding ramp)	0.5 $\sim$ 3
Cooling Setpoints	65°F $\sim$ 85°F
Thermal Integrity	Very little $\sim$ Very good
Type of house	Single family, Apartment, Mobile home

The reduced order model attempts to speed up the simulation without affecting the simulation performance metrics. To evaluate the error, we use simulation results from TESP full models as benchmark. The absolute difference between the metrics of interest (cleared price, aggregate HVAC loads) for the full model and reduced order model can be computed for each simulation time step. The mean absolute error (MAE) is calculated as shown in Equation 3.9 for different simulation cases.

$$A_{err}^{MAE} = \sum_{t=0}^{n} \frac{1}{n} \left(\frac{\left| A_t^{sim} - A_t^{full} \right|}{A_{\max}} \right) \tag{3.9}$$

Where A is the metrics of interest, n is the total number of simulation time steps.

3.4.1 Reduced Order Model for the Generalization across Different Days (ARL^a)

For the training of the bid and response RNNs that are presented in Fig. 25, we used 5 weeks of simulation data from May 28^{th} to July 2^{th} generated by the TESP full model with 1500 houses. For testing we assumed 1500 houses, and we used the TESP full model simulation results for the weeks of July 9^{th} and August 6^{th}. Specifically, cleared price, bids and HVAC loads of each individual house were extracted from the simulation results and used for training. After training, the model is tested across different months. The training results are shown in Figures 35-37, the test results are shown in Figures 38-41. We compared the full model and reduced order model results based on mean absolute bidding price error and mean absolute HVAC load error as shown in Fig. 42.

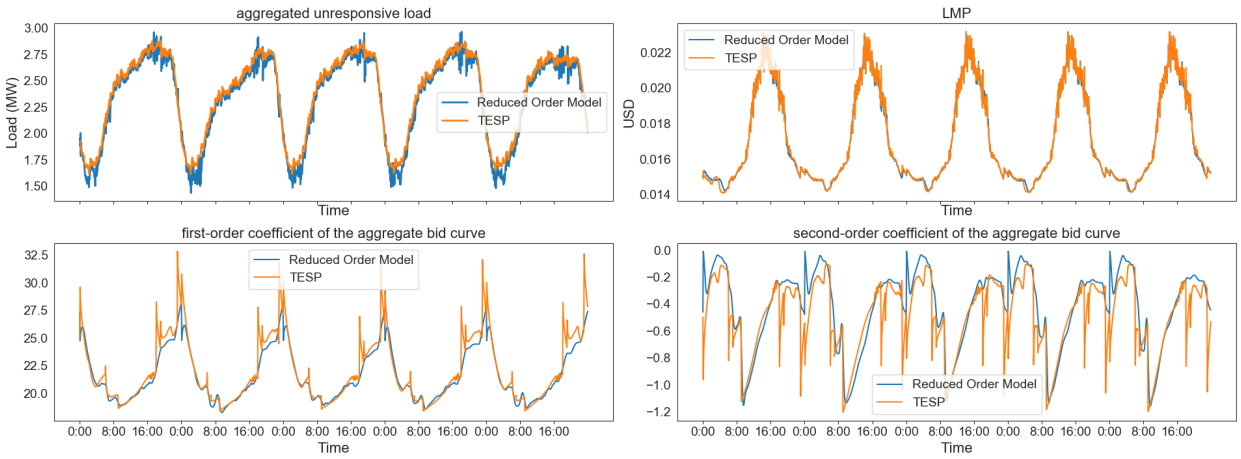


Figure 35: Aggregate parameters of 1500 houses from the reduced order model and the full model

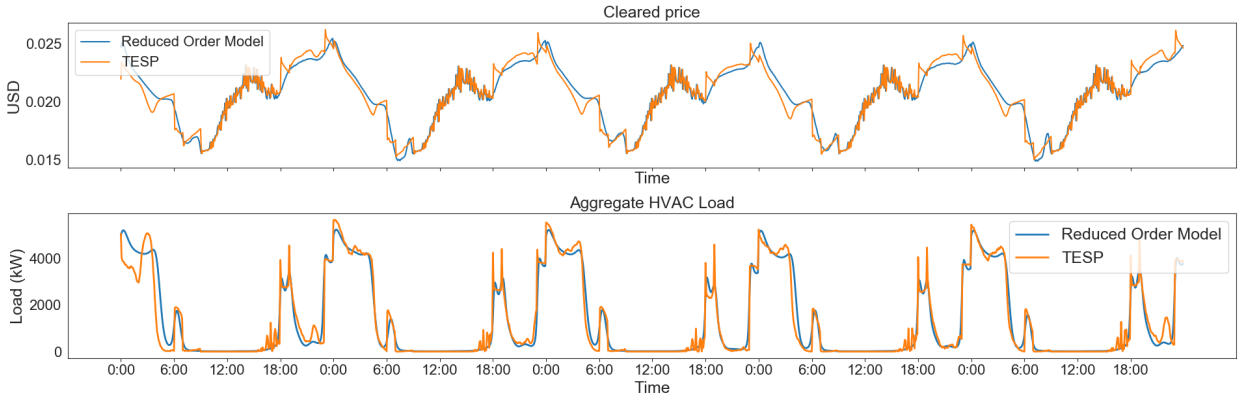


Figure 36: Transactive Load and the cleared price on 1500 houses

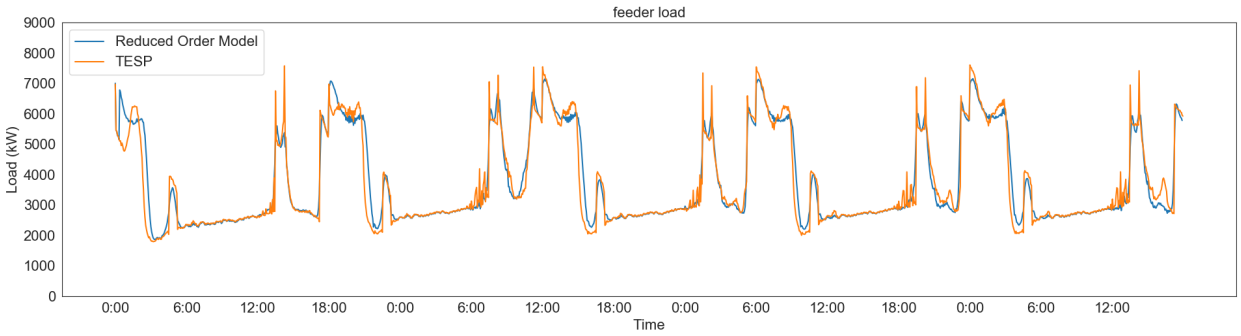


Figure 37: Total feeder load

The cleared price is affected by the bids from transactive loads and the aggregate transactive load is changed based on the cleared price. Therefore, the cleared price and the aggregate HVAC load are two main metrics to be observed in order to test the validity of the fully connected simulation model with the proposed ARL agent. Fig. 36 shows the simulation result of five day cleared price and aggregated HVAC load of the reduced order model and full TESP model on training data sets. Fig. 35 shows the transactive load related variables in the simulation of the reduced order model and the full model. In the reduced order model, these variables are calculated with the bids generated from the ARL agent, accurate estimation of these variables with ARL agent can improve the consistency of cleared price with the full model. In this case, after the market collects bids from the aggregate responsive load, the aggregated unresponsive bidding load are calculated by subtracting all the HVAC loads (determined by the response RNN from last market clearing cycle) from the total substation load.

It can be observed from Figure. 36 that simulation of reduced order model with the ARL agent is able to produce similar load consumption patterns of transactive HVACs as the full model of TESP. The morning peak and evening peak are well captured. The reduced order model is also able to produce the relatively small fluctuation such as the short peak at 6am every day. Correspondingly, the cleared price of the reduce order model also matches with the TESP simulation result which validates the bidding function of the proposed ARL agent

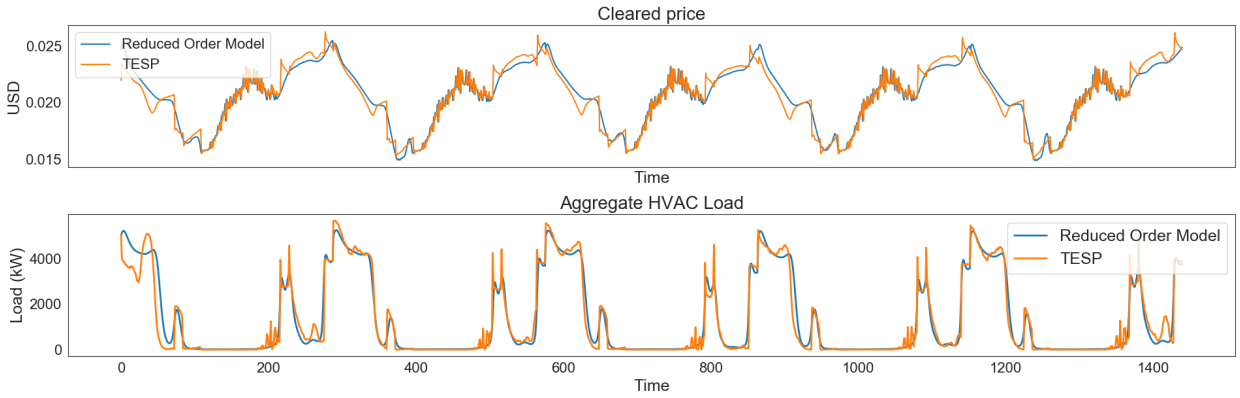


Figure 38: Transactive Load and the cleared price on 1500 houses (test data set 1)

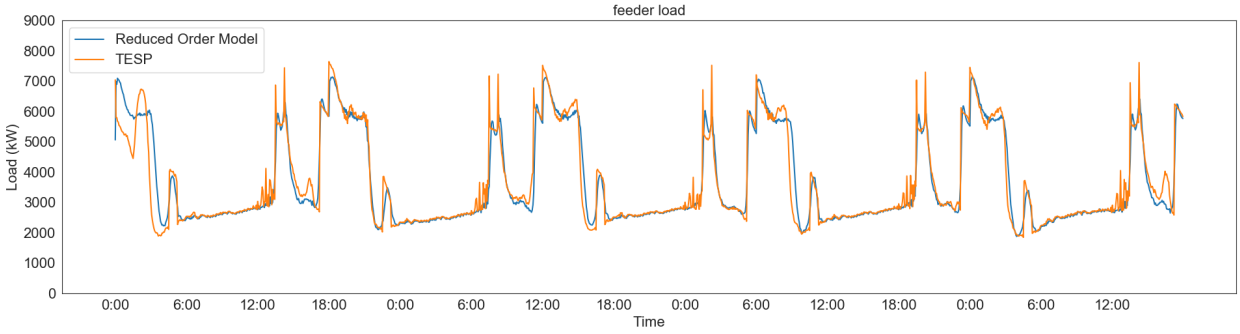


Figure 39: Total feeder load (test data set 1)

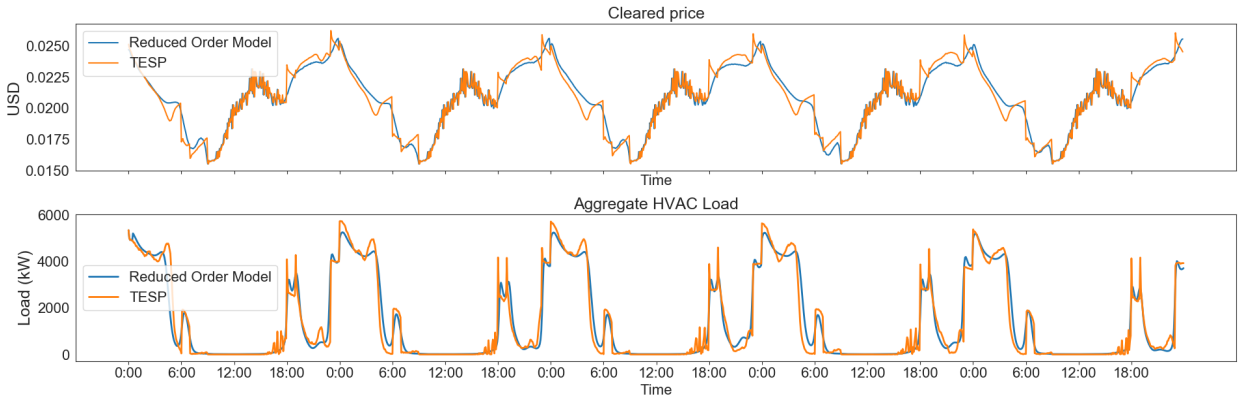


Figure 40: Transactive Load and the cleared price on 1500 houses (test data set 2)

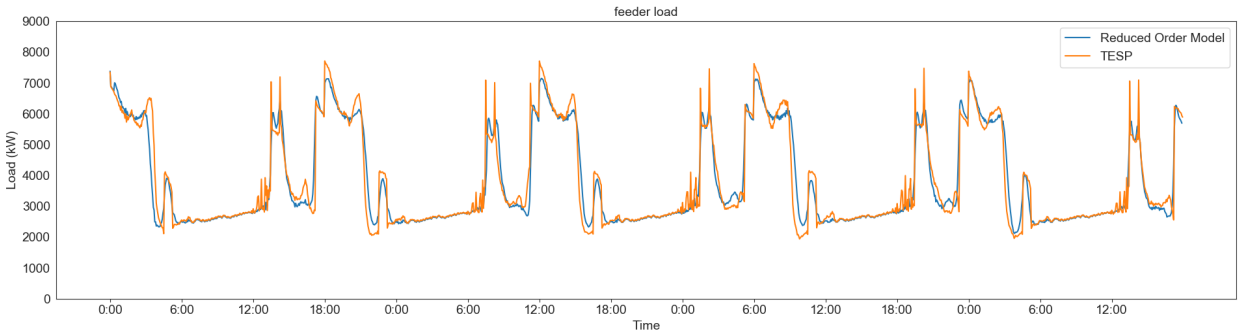


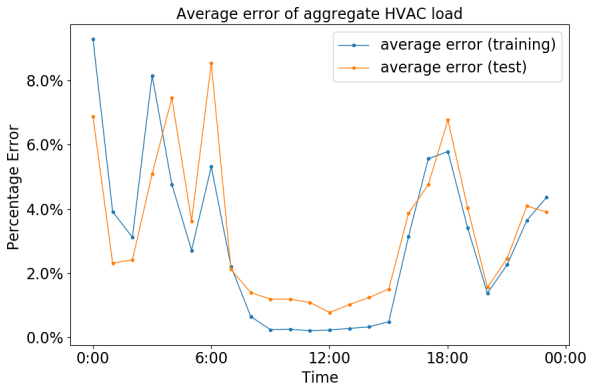
Figure 41: Total feeder load (test data set 2)

Fig. 38-41 shows the simulation results of cleared price and aggregated HVAC load of the reduced order model and full TESP model in two different test cases, one is the result from the week of July 9^{th}, the other is from the week of August 6^{th}. These results demonstrate that the ARL agent is able to learn the bidding and responding functions of the transactive load and generate appropriate simulation result of days from different months under different outside temperature.

To summarize the training results, the MAEs of bidding price and aggregated HVAC load across different days are shown in Fig. 42. Maximum bidding price error is less than 3% and maximum aggregate HVAC load error is less than 8%. It can be observed that the maximum errors occur around 6:00 and/or 18:00. This is due to the slight time shift between the ARL and full model results as shown in Figures 36, and 38. Nonetheless, the average error is acceptable and the overall trends of the full and reduced order model transactive behaviour are very well matched.

The error of the reduced order model is summarized in Table 8. The reduced order model has steady performance over different cases. There is no significant difference between the error of training and testing cases.

(a) Average mean absolute error of bidding price from different houses across a day.



(b) Average mean absolute error of aggregate HVAC load across a day.

Figure 42: Average error changing across a day

Table 8: MAE of training and testing cases

Mean Absolute Error (MAE)		
	cleared price	Aggregate load
Training - June	0.027	0.053
Testing 1 - July	0.030	0.054
Testing 2 - August	0.024	0.042

3.4.2 Reduced Order Model for the Generalization across Different Number of Houses (ARL^b)

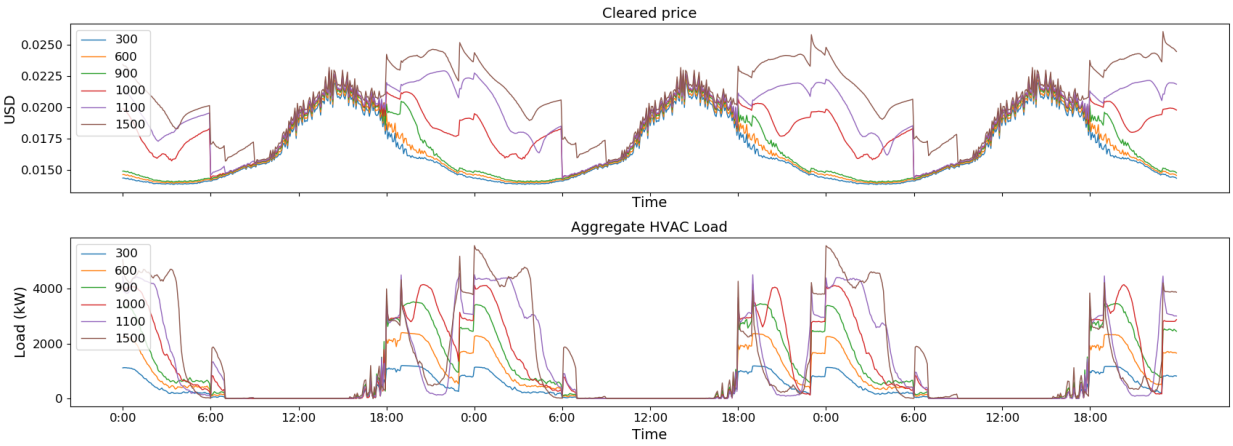


Figure 43: Cleared price and aggregate HVAC load from TESP full model

Fig. 43 shows the TESP simulation result of different number of houses. As described in Section 3.2.4, there are obvious distinctions on cleared price among different cases and the aggregate HVAC load doesn't scale up as the number of houses keeps growing. The ARL agent in Fig. 25 are not able to capture the bidding and responding behaviors affected by the changing number of houses. For the generalization across different number of houses, the ARL model in Fig. 26 are used. In this case, TESP simulation data from 600, 900, 1000, 1400, 1500 houses are chosen as the training data. After training, the ARL agent is tested in the reduced order model with different number of houses and compared with the simulation results of TESP full model. The test set results are shown in Fig. 44-48 with 498, 699, 798, 1200, 1299 houses respectively. The average performance metrics across different days are shown in Fig. 49.

Fig. 44-48 shows the simulation results of the reduced order model after training. When the house number is below a certain limit, there is little impact on the bulk market operation as the house number increases, tiny variations of the cleared price can be observed in Fig. 44-46. When the house number is beyond a limit such as the case in Fig. 47 and Fig. 48, the increasing load (even small amount) triggers congestion on the bulk system, leading to abnormal cleared price. We

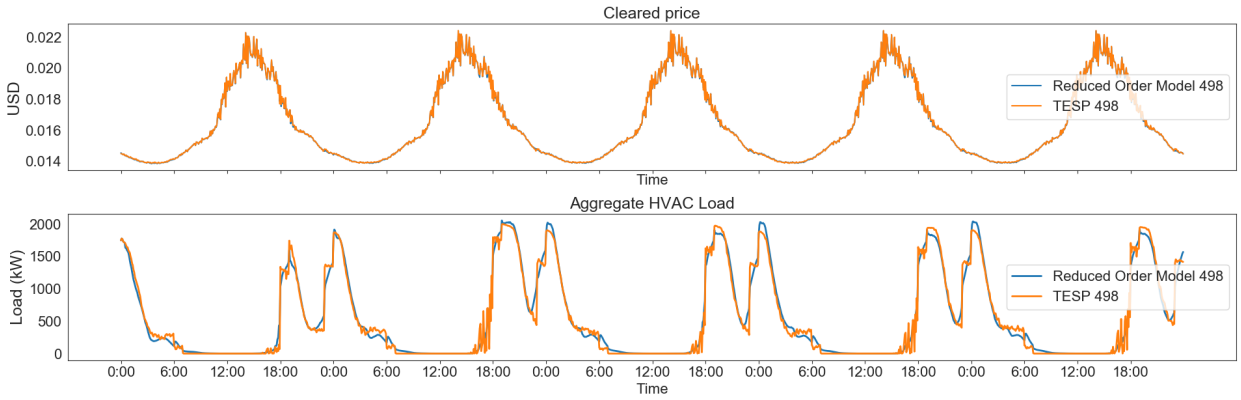


Figure 44: test case: 498 houses

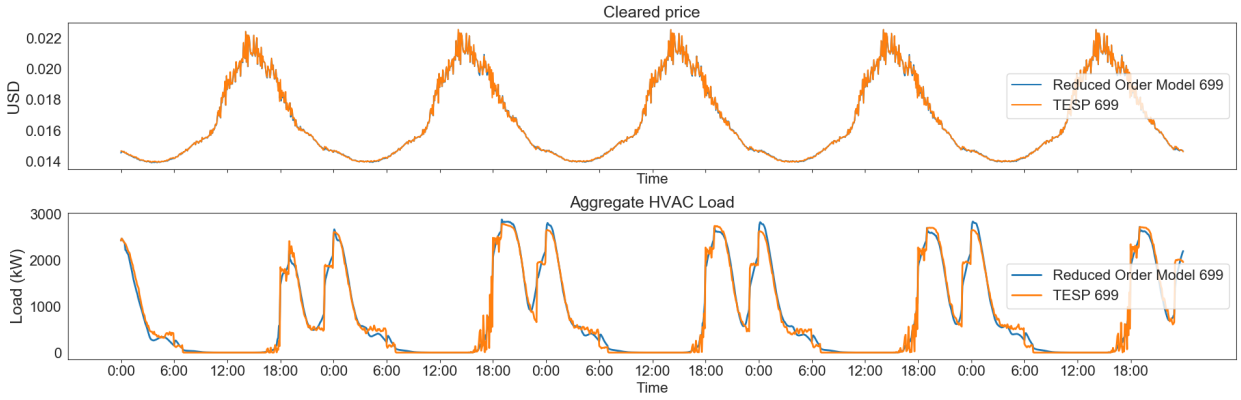


Figure 45: test case: 699 houses

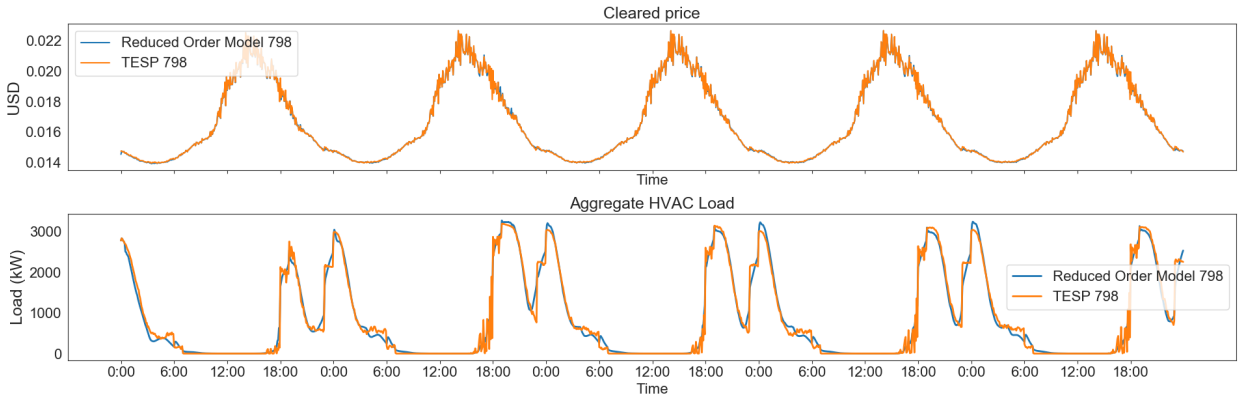


Figure 46: test case: 798 houses

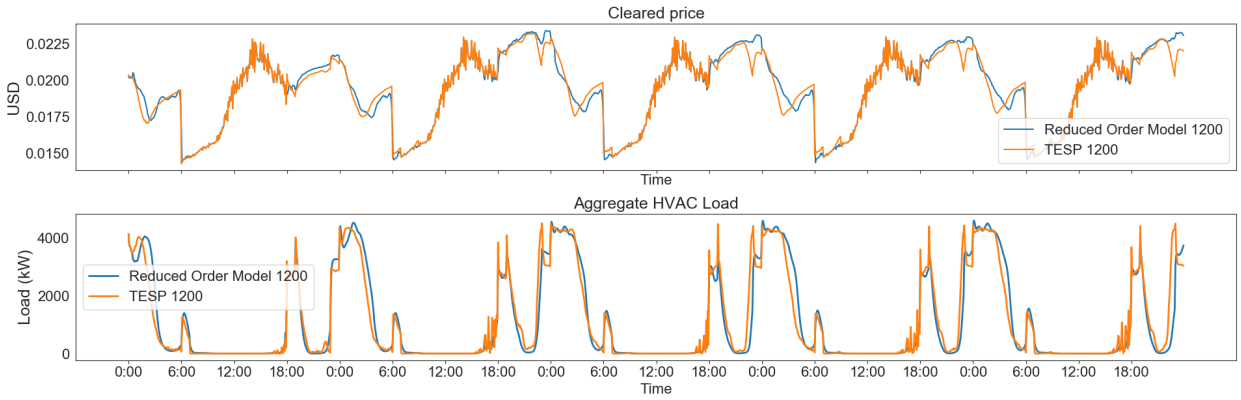


Figure 47: test case: 1200 houses

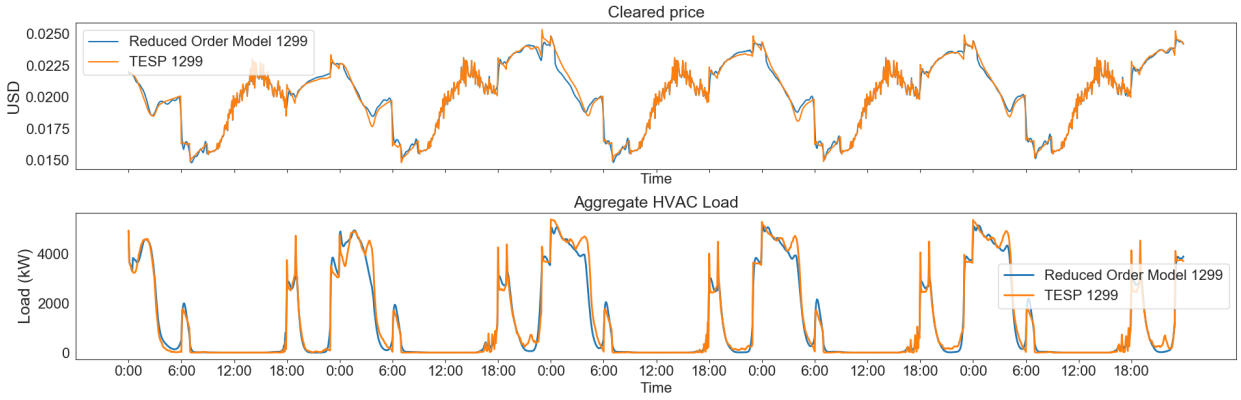


Figure 48: test case: 1299 houses

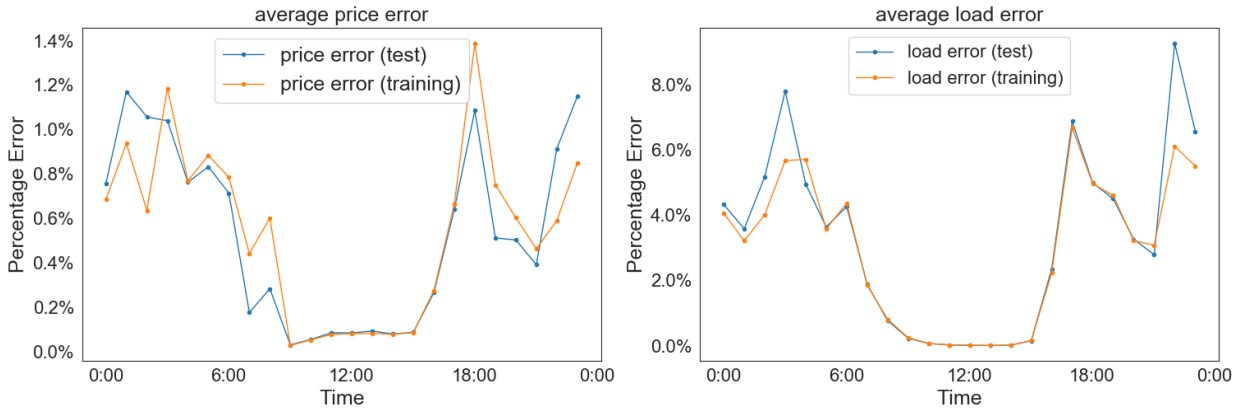


Figure 49: Average Error (mean absolute error (percentage))

compare the simulation results of the proposed reduced order model with the original simulation result in the full TESP model. Similar to the full TESP model simulation results in Fig. 43, these results are different test cases with different number of houses on the feeder. It can be observed that the aggregate HVAC load doesn't simply scale up with house number, the HVAC load behaviors change as the total number of houses grows. The cleared price also varies in different cases. In Fig. 47 and Fig. 48, the cleared price is affected by the congestion starting around 6pm every day, the ARL agent is able to predict it based on the learning from the training data. These figures demonstrate that the reduced order model is able to generate results similar to the full TESP model. Fig. 44-48 also illustrate that the proposed reduced order model generalizes well on feeder models with different number of houses. The cleared price matches well in different cases which verifies the accuracy of the bid RNN's bidding function in the reduced order model. The aggregate HVAC load is also consistent with the trend of total HVAC load in the full TESP model. Small spikes and different peaks are well captured which verifies the function of the response RNN to rationally adjust HVAC load based on cleared price according to the TESP transactive mechanisms. To summarize the results, the MAEs of cleared price and aggregated HVAC load across different days are shown in Fig. 49. Maximum cleared price error is less than 1.4% and maximum aggregate HVAC load error is less than 8.5%. The maximum errors occur around 6:00 and/or 18:00 due to the slight time shift between the ARL and full model results as shown in Figures 47, and 48.

Table 9: MAE of different cases

Mean Absolute Error (MAE)		
Case	Cleared price	Aggregare load
300 houses	0.0001	0.032
498 houses	0.0012	0.030
600 houses	0.0014	0.029
699 houses	0.0015	0.028
798 houses	0.0017	0.028
900 houses	0.0029	0.028
1000 houses	0.0083	0.028
1200 houses	0.0094	0.033
1400 houses	0.0082	0.032
1500 houses	0.0062	0.028

The errors of different cases are summarized in Table 9. There is an increasing trend in the average price deviation as the number of houses increases, the highest average price deviation is still less than 0.01 in the case with 1200 houses. The average load deviation is stable around 0.03 for all the cases.

3.4.3 Simulation Time Reduction

Table 10: Time Reduction

5-day Simulation Time			
Case	Reduced order model	Full model	Time reduction
300 houses	312.2s	1080.3s	71.1%
498 houses	329.6s	2006.3s	83.5%
600 houses	356.4s	2174.9s	83.6%
699 houses	369.1s	3060.9s	87.9%
900 houses	396.9s	3366.9s	88.2%
1000 houses	419.9s	4061.8s	89.7%
1200 houses	458.6s	4715.1s	90.3%
1400 houses	499.0s	5670.3s	91.2%
1500 houses	513.9s	6681.3s	92.3%

The simulation time of different cases is summarized in Table 10. The reduced order model can save approximately 80% to 90% of simulation time. It can be observed that the time reduction becomes more significant as the number of houses increases. The reduced order model is able to reduce 92.3% of the simulation time for a feeder model with 1500 houses.

Fig. 50 is a plot of simulation time vs. increasing house number with extrapolation. The percentage values are the time reduction achieved by the reduced order model. The simulation time of the full model increases nonlinearly with the number houses due to the messaging between increasing number of agents. It can be estimated that even more time reduction can be achieved as the number of houses continues to increase.

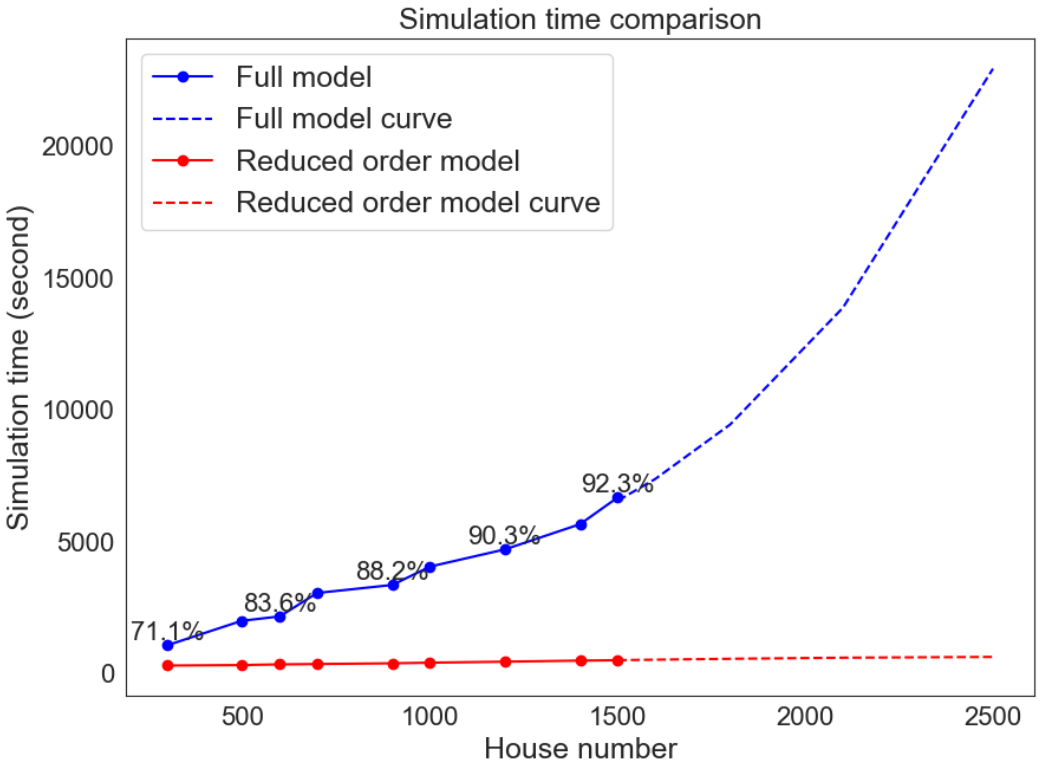


Figure 50: Simulation time comparison

4.0 Automatic control of Transactive HVACs

The increase in population, rapid urbanization, and the usage of various household appliances leads to increasing energy consumption. It is crucial that the energy providers are reliable and flexible based on these increases in the demands. Demand response (DR) of the energy providers motivates the consumers to adapt their energy consumption in response to the market pricing signals [72]. In the recent years, with the widespread application of advanced information and communication technologies, buildings and household appliances have become more intelligent, having the potential to operate more efficiently to adjust their usage based on the DR and also to achieve higher energy savings. Transactive energy (TE) extends DR to operate on faster time scales with multilateral market participation by responsive loads [7]. In this study, we focus on TE systems with HVAC as a responsive load.

Electricity use by residential air conditioners accounts for 14.7% of the total power consumption in the US, which was the largest use of electricity by the U.S. residential sector in 2018 [73]. With the advancements in technology, HVAC systems can be designed to participate in TE systems with energy providers by modifying the temperature levels at each individual residence based on the consumer needs, available energy levels and energy prices. HVAC load can be shifted by pre-heating or pre-cooling the houses providing flexibility to these systems for intelligent operation based on TE [74]. However, consumers are generally willing to pay more for comfort. For example, it was shown that residential consumers will pay two times the actual price for electricity during a power outage [75]. This may be partially due to the fact that the consumers may not be aware of the price changes and/or they may not be willing to compromise on their comfort. However, another factor that contributes to this is that the current HVAC (or other household appliance) technology does not adjust energy consumption patterns that can balance between consumer comfort and energy savings. We argue that future HVAC technology should enhance an intelligent automated operation for active participation of the consumers to achieve this balance between price and comfort [76].

Real-time thermal control is required for the HVAC systems to participate in TE in an automated manner. Traditionally, model-based approaches are used for thermal control problems [77–79], often requiring simplified mathematical modeling of the dynamics of the HVAC systems. However, model-based approaches require time and domain expertise [80] to obtain a robust and generalized approach for HVAC thermal control strategy design due to various randomness originating from individual residences (e.g. size, thermal integrity, window wall ratio and different behaviors of the end users) which introduces additional complexity and uncertainty to the control problem.

In order to address this randomness, artificial intelligence (AI) was applied in many optimal decision-making problems in TE by imitating human behavior and automating the control of the appliances such as HVAC systems. To solve such problems, especially reinforcement learning (RL) was utilized. RL is a machine learning approach with a strong ability to learn and adapt through the interaction with the environment of real world applications. It was shown that with the help of RL, a well designed TE scheme can achieve better performance on the optimal control and decision making of residential appliances. For example, most studies demonstrated the use of a popular RL method, Q-learning [81], in DR and TE [82–84]. Another RL based method was proposed in [85] for the modeling and learning of TE for plug-in electric vehicle (PEV) charging to reduce the long-term cost. Yang et al. used RL to solve the optimal control of a building energy system [86]. In [87], with the predicted future price, the authors proposed a multi-agent RL algorithm to make optimal decisions for the control of various home appliances. In [88] and [89], batch RL algorithms were proposed to schedule thermostatically controlled loads and water heaters participating in a day-ahead market. However, few of the studies modeled the appliances with a high level of detail. Most of the above mentioned approaches did not have a practical way to deal with the continuous space of the controlled state (temperature) of the HVAC systems. Moreover due to limitations in the simulations, these studies failed to provide a high degree of granularity in the precise control of the HVAC.

In this study, we develop an RL-based approach for precise control of HVAC systems that are participating in the energy market as transactive elements in the Transactive Energy Simulation Platform (TESP) [90]. TESP was developed by Pacific Northwest National Laboratory (PNNL) as an open-source simulation platform with transactive market and control mechanisms

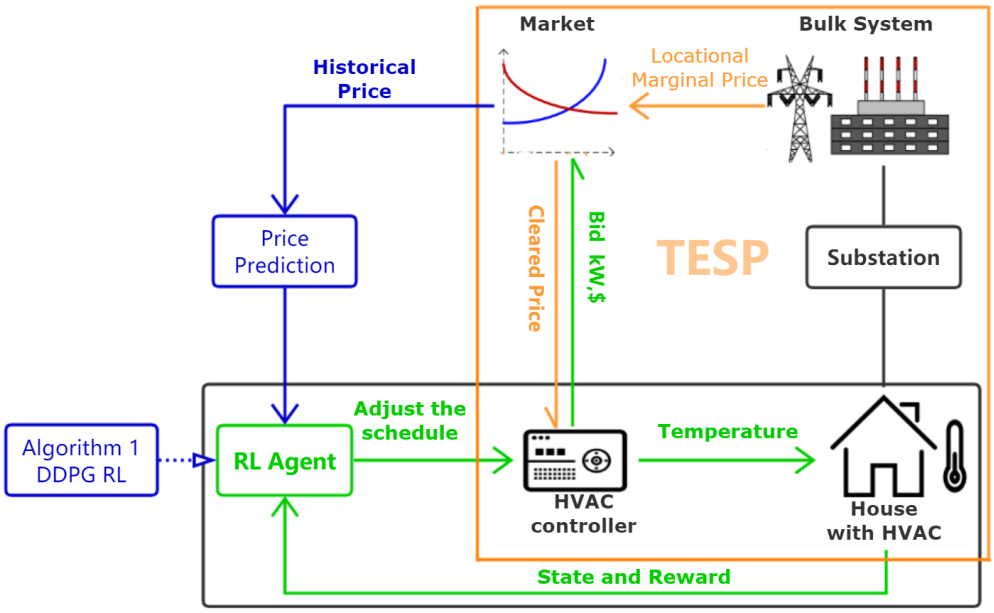


Figure 51: TESP and the implementation of proposed methods

for the grid [2]. TESP includes distribution simulator, transmission simulator and building simulator with multiple transactive agents, and the integrating Framework for Network Co-Simulation (FNCS) [37] that manages the message exchange among different simulators. In order to have an intelligent and granular control of the HVACs, we utilize RL and formulate the control problem as an optimization of cost function that balances between the electricity cost and end-user satisfaction. More specifically, combined with a price prediction method using historical data, we adopt Deep Deterministic Policy Gradients (DDPG) RL algorithm. The methods are implemented as an RL agent in TESP simulations. DDPG is a deep reinforcement learning approach developed for continuous action space; therefore it is naturally suitable for the control of HVAC systems achieving a finer and more precise control. We specifically use DDPG RL to control the base temperature schedule of the HVAC in TESP to make the TESP thermostat controller respond to the cleared market prices more intelligently at each time step to maximize the long term reward that balances between electricity cost and end-user satisfaction.

4.1 TESP HVAC and Problem Formulation

In this section, we describe the formulation of the optimum HVAC control balancing between energy cost minimization and customer satisfaction based on RL. This RL based method relies on the predicted energy price; therefore, a price prediction method based on ANNs is also presented in this section.

4.1.1 HVAC Response and Problem Formulation

In a transactive energy system, residential users are able to participate in TE through a transactive HVAC system. Transactive HVAC systems are flexible, and they can adjust the power consumption by changing the temperature settings in residences. Here, we formulate the HVAC temperature control objective to minimize the electricity cost and the dissatisfaction of the customers caused by the temperature differences between the desired and adjusted temperature settings. We argue that the current room temperature depends on the HVAC state and power, outdoor temperature, and the room temperature of the previous time step. Accordingly, different than the legacy ramp transactive control mechanism used in TESP [31], we formulate the HVAC control through a Markov Decision Process (MDP) to optimize the energy cost and customer satisfaction simultaneously. MDP is a mathematical framework that satisfies Markov property and has four elements: a set of states which represent the environment, a set of possible actions for each state, a reward function to assess the value of each action taken at a certain state, and the rules for the transitions among different states. Below is the description of the state, action, and reward function tailored to the HVAC; the control flow of HVAC based on MDP is shown in Fig. 52.

The HVAC power consumption is influenced by various factors. We consider these factors as the elements of the HVAC state in the MDP model. We denote the HVAC state at time t as S_t, see (4.1). The observable state of an HVAC should contain information about both indoor and outdoor environment as they significantly affect the energy consumption. Therefore, the indoor temperature T^t_{room} and outside temperature T^t_{out} at time t are considered as elements of the HVAC state. In addition, the desired or scheduled base temperature, $T^t_{schedule}$, of the house is included in the HVAC state. Finally, since the HVAC on/off status at time t depends on price-responsive T^t_{set} and the current indoor thermal environment, T^t_{set} is also included in the HVAC state.

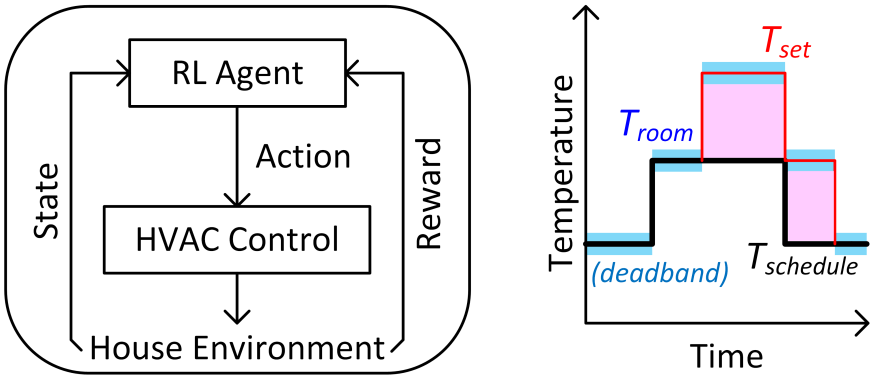


Figure 52: HVAC control flow and settings; discomfort region shaded in pink.

$$S_t = \{T^t_{set}, T^t_{room}, T^t_{schedule}, T_{out}\} \tag{4.1}$$

$$T^t_{set} = T^t_{schedule} + \frac{(P^t_{cleared} - P_{average}) \times \left|T_{\max/\min}\right|}{k_{high/low} \times \sigma_{actual}} \tag{4.2}$$

The relation between T^t_{set} and $T^t_{schedule}$ in TESP is shown in (4.2), where $P_{average}$ is the historical mean price, $\left|T_{\max/\min}\right|$ is the allowed range of set point variation , $k_{high/low}$ is the bidding ramp denominator, σ_{actual} is the standard deviation of the price. Bidding ramps and allowed temperature ranges could be unequal above and below $T^t_{schedule}$ as in [31].

The aim of the HVAC control is to minimize the cost by changing the HVAC temperature setting schedule, $T^t_{schedule}$. Therefore, in our formulation, the learning agent of the RL approach based on MDP assumptions is designed to make changes in the scheduled temperature deviating from the original schedule based on a reward function. The action is the temperature change from the original schedule in a certain adjustable range, e.g. [-5,5] degrees Fahrenheit.

The reward of each action consists of two parts, the penalty for the energy consumed by the HVAC during the time period and the discomfort of the consumer resulting from the control action taken at a given state. The discomfort is the estimated feedback of the occupants' dissatisfaction under the current thermal condition. The reward at each time step is defined as:

$$r_t = -\alpha(E^t_{hvac} \times P^t_{clear}) - (1-\alpha)k \times (T^t_{dev})^2 \tag{4.3}$$

$$T_{dev}^{t} = (T_{room}^{t} - T_{schedule}^{t}) \tag{4.4}$$

where α represents the importance of the cost of energy consumption of the HVAC. E_{hvac}^{t} is the energy consumption of the HVAC during this time step. P_{clear}^{t} is the cleared price from TESP. The cost will be higher if more energy is consumed when the price is relatively high. The second term is the consumers' dissatisfaction cost which is calculated by multiplying a factor k by the squared room temperature deviation T_{dev}^{t} from the original schedule temperature.

4.2 Deep Deterministic Policy Gradient

Model-based or model-free approaches can be used in reinforcement learning to optimize energy cost and/or thermal comfort through the control of HVAC [91]. Model-based approaches require complete information of the HVAC thermal dynamics to represent transition among different states. For example, for the model-based approaches, accurate dynamic interactions between the residence and the surrounding environment may be needed. In contrast, model-free methods are more flexible to overcome the detailed modeling of the HVAC dynamics and accordingly to represent state transitions.

Q-learning, state-action-reward-state-action (SARSA) and deep Q-networks (DQN) are commonly used for model-free RL [92]. However, they cannot be used to solve control problems with both continuous state and action spaces. For instance, in order to utilize DQN for HVAC control, temperature of the HVAC can be discretized finely, resulting in a large number of possible actions. But higher granularity of the action space will decrease the training efficiency dramatically. DDPG is a deep reinforcement learning method which is capable of handling a space of continuous states and actions. There exist other off-policy algorithms like soft actor critic (SAC) [93] and twin delayed DDPG [94] which are variations of the DDPG algorithm. They can also be used to solve the continuous control problem such as HVAC control. In this study, we utilize DDPG for the control purposes as we can show through our numerical results that the reward convergence is robust to the changes in the hyperparameters.

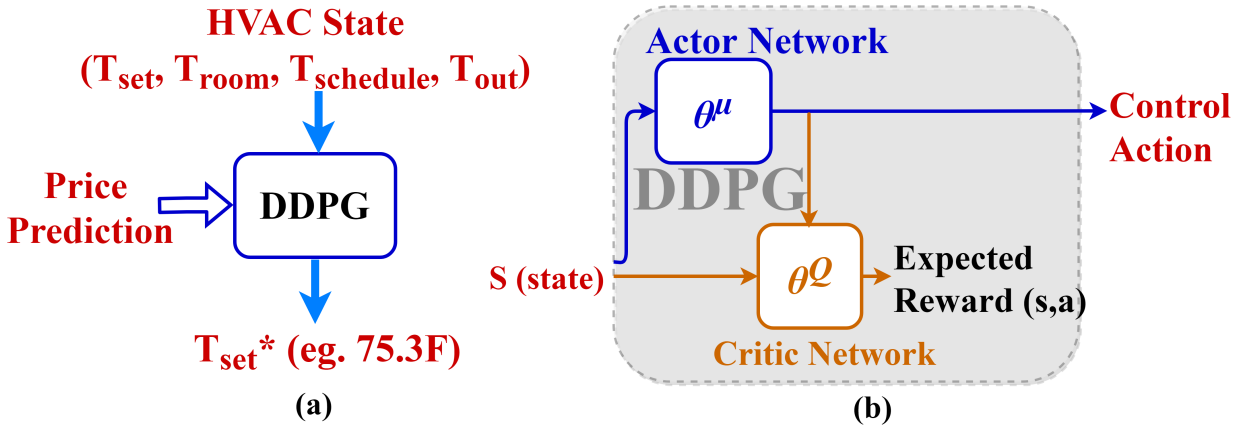


Figure 53: (a) Input HVAC state, DDPG is able to generate continuous action control. (b) The network structure of DDPG implemented as the RL agent. The actor network specifies a control action given the current thermal state and the critic network outputs an evaluation of the action generated by the actor network.

As shown in Fig. 53(a), for any given input state, through the interaction of actor and critic networks, DDPG is able to generate optimal control action directly rather than by fine discretization of the action space. The network structure of the DDPG method is presented in Fig. 53(b). More specifically, DDPG is implemented here through an actor-critic architecture that learns approximations to both policy function, θ^{μ}, and value function, θ^{Q}. An actor is used to tune the parameter θ^{μ} for the policy function (i.e., to decide the best control action A_t given a specific HVAC state S_t, where θ^{μ} represents the weights of the actor neural network). On the other hand, a critic network is used for evaluating the policy function estimated by the actor network. Here, the critic network's parameters are denoted by θ^{Q}. Critic network estimates the action value Q which is the expected reward of taking the control action A_t at state S_t.

The actor network and the critic network are trained through the TESP simulations which enables evaluation of different actions for different HVAC states. After training, during testing, through the interaction between actor and critic networks RL-based control outputs an optimum action that is used by TESP to control the HVAC. The training details of the actor and critic networks are provided in Algorithm 1 and Fig. 54.

Algorithm 1 DDPG

procedure DDPG RL(θ^{μ},θ^{Q})

Initialize memory M of size N;

Initialize the actor network $\mu(S_t|\theta^{\mu})$ and critic network $(S_t, A_t|\theta^{Q})$

with random parameter θ^{μ} and θ^{Q}

Initialize the target network $\mu^{'}$ and $Q^{'}$ with $\theta^{\mu^{'}} \leftarrow \theta^{\mu}$,$\theta^{Q^{'}} \leftarrow \theta^{Q}$;

Input the estimated price $\{\hat{P}_{clear}\}_0^T$;

Define $s_t = \{T_{set}^t, T_{room}^t, T_{out}^t, {T_{schedule}}^t\}$;

Receive the initial HVAC state s_0=$\{T_{set}^0, T_{room}^0, T_{out}^0, {T_{schedule}}^0\}$;

for t=0,1,2,..,T **do**

Select a_t by $a_t = \mu(s_t|\theta^{\mu}) + \mathcal{N}_t$;

Execute a_t on HVAC and obtain the reward $r(s_t, a_t)$ and next state s_{t+1};

Store the transition (s_t, a_t, r_t, s_{t+1}) in M;

Sample K transition from M randomly and calculate the estimated policy value for the sampled transitions i : $y_i = r_i + \gamma Q^{'}(s_{i+1}, \mu^{'}(s_{i+1}|\theta^{\mu^{'}})|\theta^{Q^{'}})$;

Update the critic network θ^{Q} by the gradient $\nabla_{\theta^{Q}} L$ of the MSE over the K size mini-batch and learning rate β_y: $\nabla_{\theta^{Q}} L = \frac{1}{K}\sum_{i=1}^{k}(y_i - Q(s_i, a_i|\theta^{Q}))^2$;

Update the actor network using the sampled policy gradient $\nabla_{\theta^{\mu}} J$ and learning rate β_x: $\nabla_{\theta^{\mu}} J \approx \frac{1}{K}\sum_{i=1}^{K} \nabla_a Q(s, a|\theta^{Q})|_{s=s_i, a=\mu(s_i)} \nabla_{\theta^{\mu}} \mu(s|\theta^{\mu})|_{s_i}$

Update the target networks (τ : updating rate):

$\theta^{\mu^{'}} \leftarrow \tau\theta^{\mu} + (1-\tau)\theta^{\mu^{'}}$;

$\theta^{Q^{'}} \leftarrow \tau\theta^{Q} + (1-\tau)\theta^{Q^{'}}$;

end for

end procedure

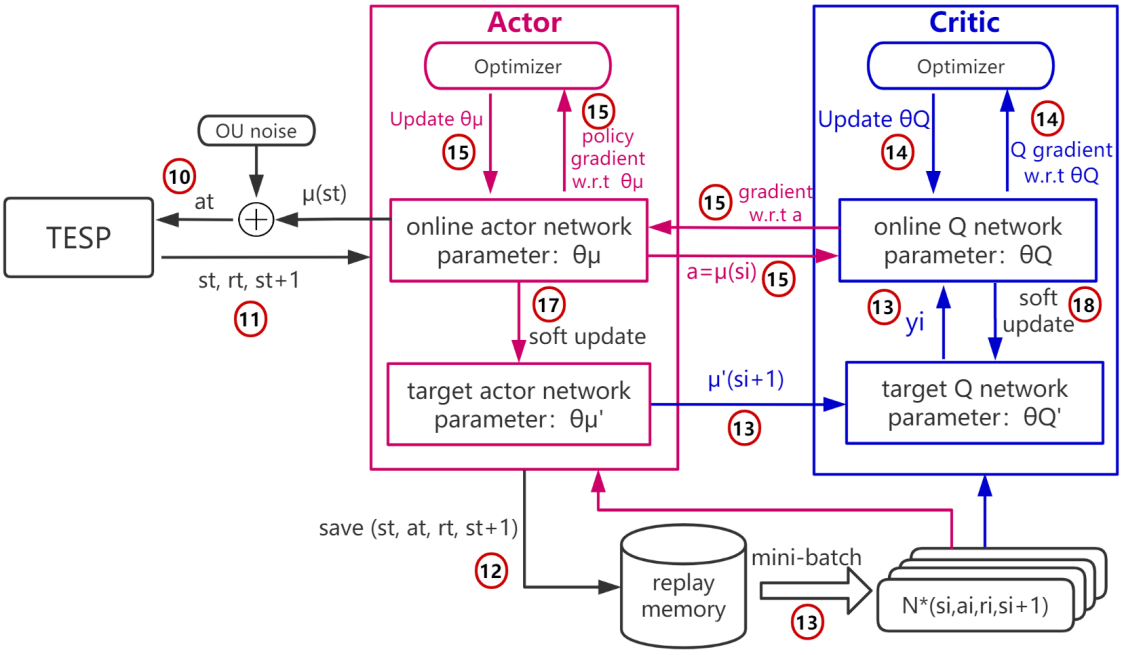


Figure 54: The structure of critic network and actor network, red numbers correspond to lines in Algorithm 1.

For the training, we initialize the actor network and the critic network with random parameters, also we use the same random parameters to initialize the target actor network and target critic network. DDPG enables the agent to explore a wide variety of actions in the beginning of learning. Specifically, after receiving the initial state s_0, the actor network explores the action space to select a control action. We add random noise to the selected action to explore the control action space to prevent converging to a local solution through Ornstein–Uhlenbeck process [95], see Algorithm 1 line 10.

During training, at each time step t, after the learning agent takes the control action a_t, it communicates this action to TESP to change the HVAC state s_t, then receives the new HVAC state s_{t+1} and the reward R_t calculated based on (4.3) as feedback from TESP. In order to improve the convergence and decrease the correlation among the training samples, we add a memory buffer for experience replay. So at every time step, the state action transition s_t, a_t, R_t, s_{t+1} is stored into the memory M. From memory M, we then randomly sample K transitions and calculate the estimated value y of each sampled transition using the target networks. The next-state Q values are calculated

with the target value network and target policy network (Fig. 54 arrow 13). Then, we minimize the mean-squared loss between the updated Q value and the original Q value (line 14). Here, we use the target networks which are constrained to change slowly. The two target networks $\theta^{\mu'}$ and $\theta^{Q'}$ will slowly track two learned networks θ^{μ} and θ^{Q} which will help improve the stability of learning. Calculation of the estimated value y through the target networks is achieved through Algorithm 1 line 13, where γ is the discounting factor indicating the The weight of the critic network is updated by minimizing the mean square error with respect to the critic network parameters using the values corresponding to the randomly selected K samples as shown in line 14 of the Algorithm 1. The policy loss is the derivative of the objective function with respect to the policy (actor network) parameters. Then the actor network is updated through the sampled policy gradient as shown in line 15 of Algorithm 1 [96]. Note that the chain rule is applied since the policy function and the actor network are both differentiable. Finally, both target networks are updated with an update rate $\tau \ll 1$ as shown in lines 17 and 18.

The actor network and the critic network of DDPG algorithm both have 2 hidden layers. The structure and different activation functions are shown in Fig. 55.

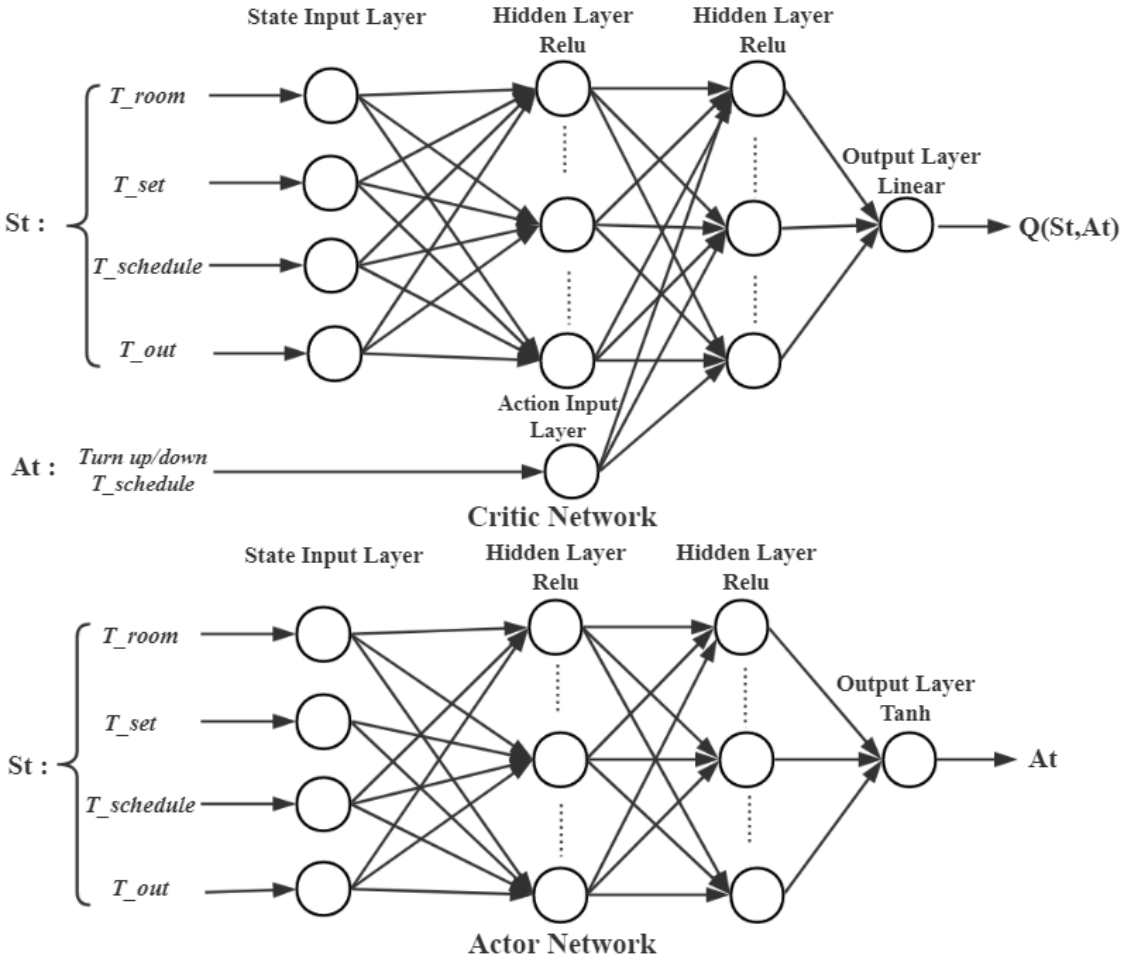


Figure 55: The structure of critic network and actor network; ReLU is a rectified linear activation unit.

4.3 Price Prediction

The optimal control strategy based on DDPG relies also on the predicted electricity price, see Fig. 53. In our approach, we utilized a multi-layer perceptron neural network with 2 hidden layers to predict the future electricity price. Through such an artificial neural network, we develop a nonlinear relationship between the input variables (e.g. temperature, system load, day of the week) and the predicted output electricity price. Fig. 56 demonstrates the topology of the utilized neural network. As listed in Fig. 56 there are up to 18 day, hour, load, temperature and price inputs connecting to the hidden layers.

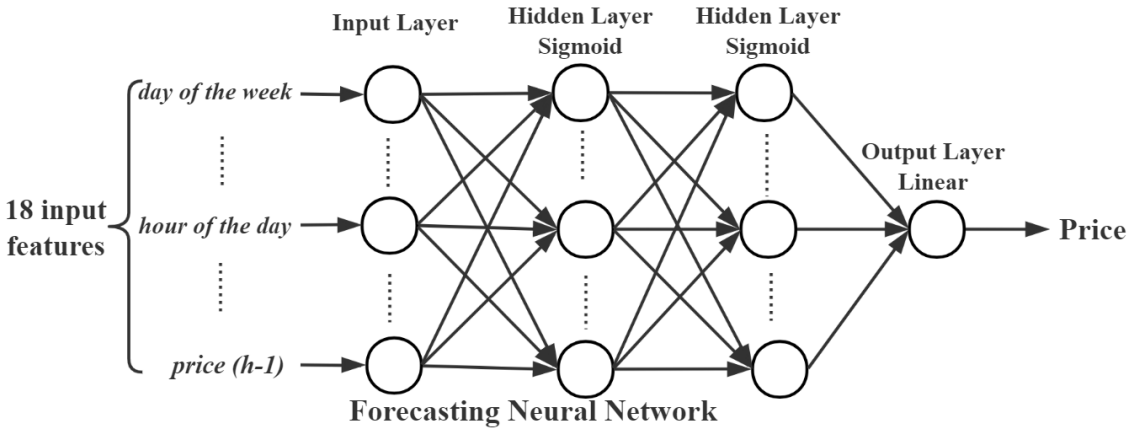


Figure 56: The neural network for price prediction

4.4 Simulations and Numerical Result

In this section, we describe the simulation scenarios and present the numerical results. We first present the performance of the proposed ANN structure for electricity price prediction and we compare it with the state-of-the-art price prediction methods such as weighted average filter [97], support vector machine (SVM)-based prediction [98], and ANN-based prediction [99]. Then, we consider different simulation scenarios in TESP to compare the proposed DDPG RL-based HVAC

control strategy with the control strategy that is already implemented in TESP in terms of electricity cost and consumer satisfaction. We represent the consumers' satisfaction by the deviation of the temperature settings from the desired temperature schedule of the HVAC systems.

4.4.1 Simulation and Performance of the Price Prediction

We generated four weeks of electricity price data using TESP. This simulation includes 306 houses with HVACs connected to a large distribution system as shown in Fig. 51. We used the first two weeks of the generated price data for training the proposed neural network and used the second two weeks of data for testing. More specifically, as shown in Table 11, up to 18 input features are used to train the proposed neural network to predict electricity price. Day of the week, hour of the day and historical price data are obtained directly from the generated TESP data. Historical weather (temperature) and the load data for price prediction training in the Pittsburgh area are obtained from the weather data in Typical Meteorological Year 3 (TMY3) format [100] and PJM website, respectively. PJM is a regional transmission organization and they provide the historical hourly load data for Duquesne Light Company on their website. Since the TESP simulation data have higher temporal resolution compared to the load data, the hourly load data is interpolated to obtain 5 minutes per sample temporal resolution.

Table 11: Input features for price prediction (h represent hour)

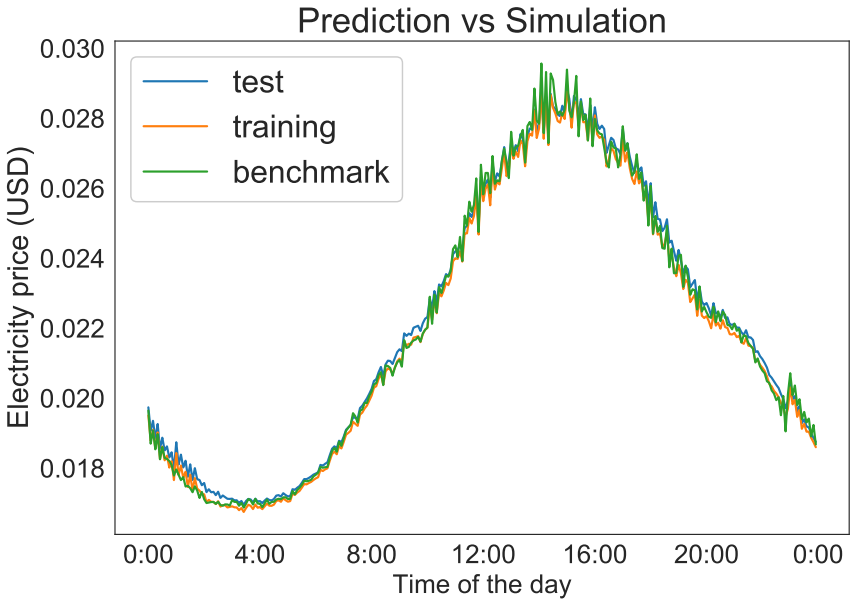
Input Features	
Day of the week	1-7
Hour of the day	1-24
Historical price	(h-1),(h-2),(h-3),(h-24),(h-25),(h-26),(h-48),(h-168)
PJM load	(h-1),(h-2),(h-3),(h-24),(h-25),(h-26)
Weather	temperature
Price distribution	mean of the distribution

4.4.1.1 Price Prediction Simulation Result

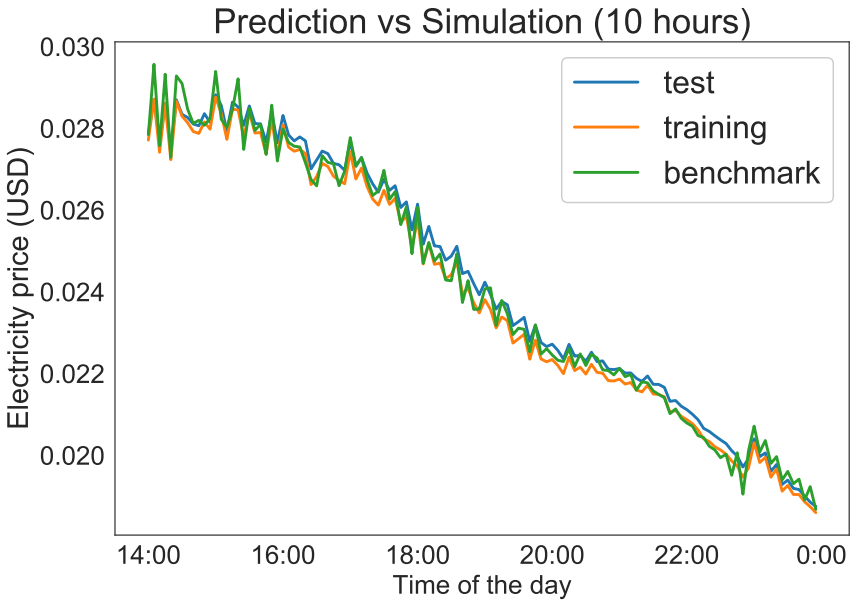
In Fig. 57 we compare the proposed neural network that is trained using all 18 inputs that are listed in Table 11 directly with the TESP simulation results. Here TESP simulation results are the benchmark. Fig. 57 (a) shows 24 hour prediction results with mean square error (MSE) of 2.12×10^{-4}), and Fig. 57 (b) shows 10 hour simulation results with MSE 2.59×10^{-4}). From these two figures, we observe that the overall trend of the predicted price is consistent with the TESP-simulated electricity price. Note that even some small fluctuations in price are also correctly predicted.

Here we also compare the proposed approach with the state-of-the-art price prediction methods. We denote the proposed approach as ANN with weather and price distribution input (ANN + weather + price distribution) and compare it with weighted average filter-based, SVM-based, ANN with weather input (ANN + weather) and ANN without weather and price distribution (ANN) methods. For this comparison, we generated TESP simulation data for the system of 306 houses as described above. Similar to the above scenario, historical weather and PJM load data are obtained from online sources.

The data was divided into 50 week-long periods, and the mean square error of predicting price of different weeks throughout the year is shown for different methods in Fig. 58. In this figure, we observe that SVM-based method is better than the weighted average filter, and ANN based methods outperform both the weighted average filter and SVM-based methods. To statistically compare the methods, we apply non-parametric one-sided rank sum test and the results are presented in Table 12. In this table, we specifically present the p-values for testing if the methods listed in the columns have a lower mean-square error in price prediction than the methods listed in the rows. A p-value lower than 0.05 means that the method listed in the column has statistically lower mean-square error compared to the method listed in the row. Similar to Fig. 58, ANN-based methods are significantly better than weighted average filter and SVM-based methods. Even though there are not statistically significant differences among ANN, ANN+W and ANN+W+P (see Table 12), adding weather and price distribution information may make the price prediction more robust, see Fig. 58. But this robustness comes with price of additional data collection.



(a) 24 hour prediction result (MSE : 2.12×10^{-4})



(b) 10 hour prediction result (MSE : 2.12×10^{-4})

Figure 57: Price prediction vs TESP Simulation data

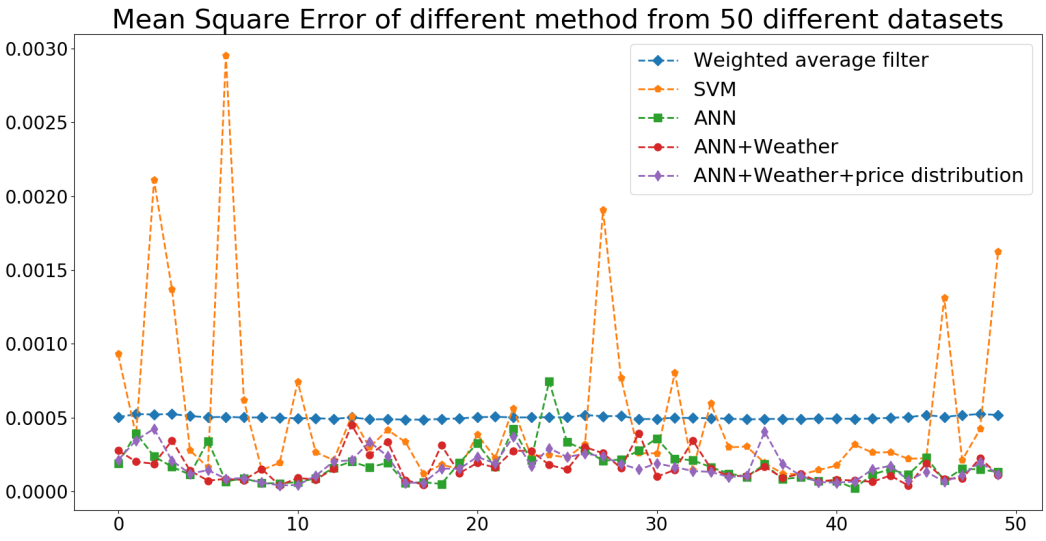


Figure 58: Error Comparison

Table 12: p value of Wilcoxon rank-sum test between the method in each row and the method in each column.

Wilcoxon Rank Sum Test of the errors in Fig. 58					
p	filter	SVM	ANN	ANN+W	ANN+W+P
filter	0.50	5.58e-5	6.75e-17	3.53e-18	3.53e-18
SVM	0.99	0.50	1.38e-7	3.31e-8	1.85e-8
ANN	1	1	0.50	0.35	0.45
ANN+W	1	1	0.65	0.50	0.58
ANN+W+P	1	1	0.55	0.42	0.50

4.4.2 Simulation and Performance of the RL Agent

4.4.2.1 Simulation scenarios

The proposed RL-based HVAC control is evaluated using TESP-simulated data on 306 houses. We specifically considered the scenarios in which HVACs are in the cooling mode. To make sure the HVACs are in cooling mode during the training, TMY3 data for Florida instead of Pittsburgh were used during the period from June to November of 2018. One generic control policy for different houses is obtained after training. The DDPG algorithm is implemented with Pytorch [101], an open source Python-based scientific computing package for machine learning. The training data comes from the simulation of 212 days in TESP.

As also mentioned above, we compare the RL-based approach with the HVAC ramp control approach that is implemented in TESP. This method (which we denote as "without RL agent" in this study) controls the HVAC using a pre-defined temperature schedule. On the other hand, the proposed RL-based method (which we denote as "with RL agent") changes the pre-defined temperature schedule based on the predicted price and DDPG-based control. We compare these two control approaches not only under normal conditions but also during a high price scenario that includes a bulk system generator outage. Test cases are illustrated in Fig. 59. Simulation configurations and key parameters of the DDPG training algorithm are listed in Table 13.

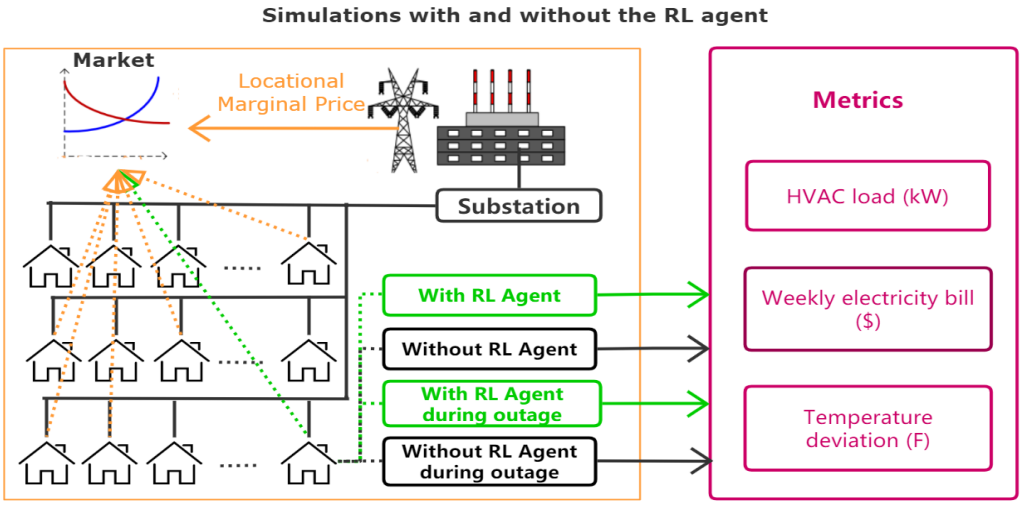


Figure 59: Test cases of the proposed RL agent

Table 13: RL Agent Training Settings

T_{test}	212 days	training data time length
Δt	5 minutes	time step
β_Q	0.000025	learning rate of the critic network
β_μ	0.00025	learning rate of the actor network
τ	0.001	model update parameter
γ	0.95	reward discount factor
α	0.1$\sim$0.5	weight factor of the electricity cost

The batch size for DDPG training is chosen to be 72. The parameter α that was introduced in (4.3) and that balances between energy cost saving and customer satisfaction is varied between 0.1 and 0.5.

4.4.2.2 Performance Metrics

In order to compare the control methods with and without RL agents, we define electricity cost saving factor (CSF) and thermal comfort improvement factor (TIF) as the performance metrics. Both are affected by α.

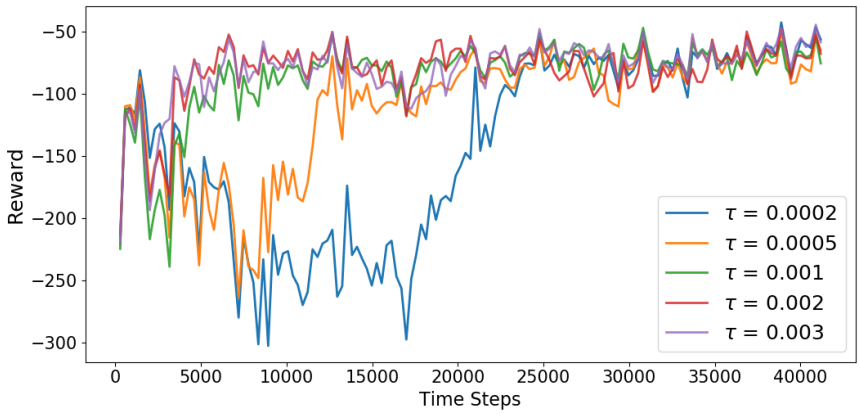
$$CSF = \frac{weeklybill_{base} - weeklybill_{RL}}{weeklybill_{base}} \times 100\% \tag{4.5}$$

$$TIF = \frac{\Delta T_{base} - \Delta T_{RL}}{\Delta T_{base}} \times 100\% \tag{4.6}$$

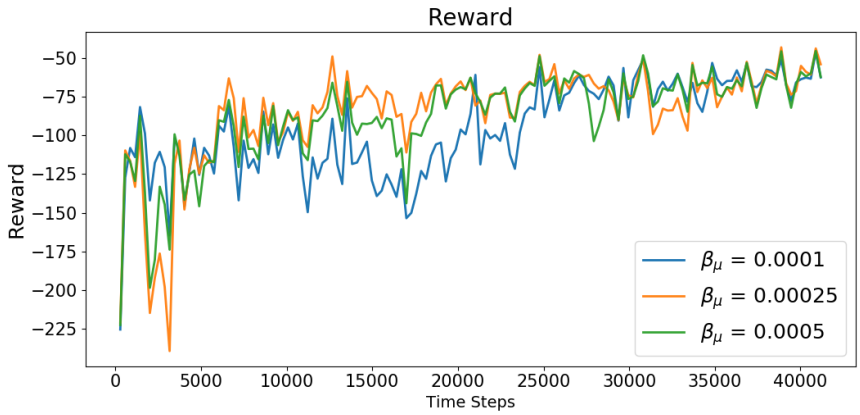
Both CSF and TIF can be greater or less than 0; a positive CSF or TIF indicates better performance with the RL agent.

4.4.2.3 Convergence of the training process

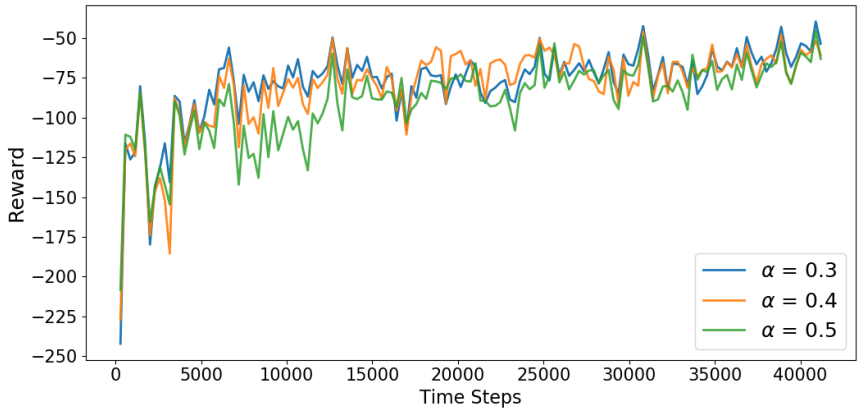
We first demonstrate the effect of the hyper-parameter selection on the convergence of the DDPG algorithm for RL-based control. Specifically, as shown in Figure 60, we plot the reward function as a function of training time steps for different hyper-parameters. Recalling that τ is the update rate for target networks, see Algorithm 1, as shown in Fig. 60(a), higher τ values lead to faster convergence. The reward converges at 25000 steps for the smallest τ value (0.0002) in our



(a) The convergence process with different τ



(b) The convergence process with different β_μ and β_Q, β_μ=10β_Q



(c) The convergence process with different α

Figure 60: Convergence of DDPG training process with different hyper parameters.

test cases. When τ is greater than 0.001, the convergence of the reward is very robust to different values of τ. Note that β_μ and β_Q are the learning rates of the actor and critic networks, respectively. Considering β_μ=10β_Q, as it is recommended in the literature to have higher learning rate for actor network than the critic network [96], in Fig. 60(b) we show the effect of β_μ on the reward function convergence. As it can be observed from this figure, the training of the algorithm is very robust to the changes in the actor and critic networks' learning rates. Finally, Fig. 60(c) shows the effect of α (the hyper-parameter that balances between electricity cost and customer satisfaction) on the reward function. As it is observed, the convergence of the reward is robust to the selection of α.

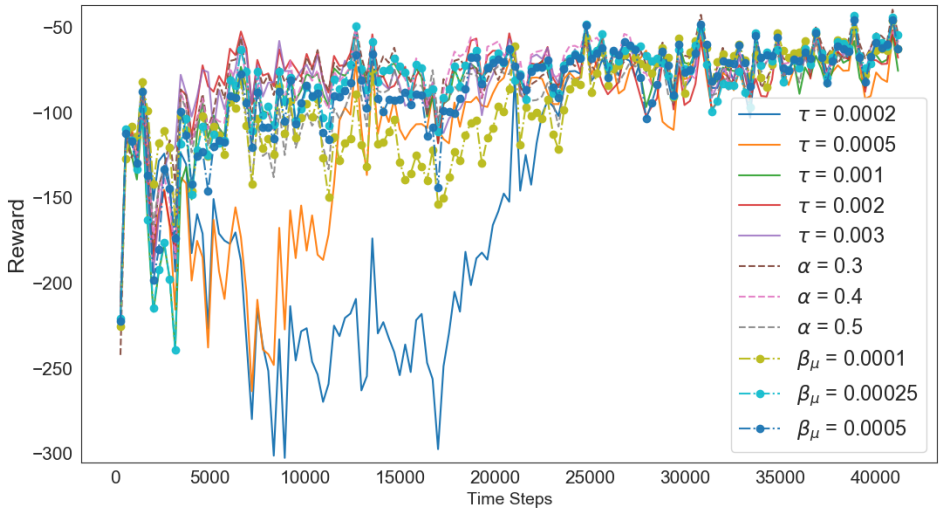


Figure 61: Convergence of DDPG training process with different hyper parameters

In summary, as shown in Fig. 61, we plot the reward function as a function of training time steps for different hyper-parameters. It can be observed that the training of the algorithm is very robust to the changes in α and β_μ (β_μ=10β_Q), and τ (when τ is greater than 0.001, which is the literature recommended value).

4.4.2.4 Performance of the DDPG RL Algorithm

Through TESP simulations as described above, we compare HVAC control with RL agent to HVAC control without RL agent. Recall here that HVAC control without RL agent uses a fixed

temperature schedule and adjusts the HVAC setting based on this fixed temperature schedule and cleared market price for electricity [31]. On the other hand, HVAC control with RL agent changes the temperature schedule and then adjusts the HVAC setting for price. For these two approaches, in Fig. 62, we plot the temperature schedules, HVAC temperatures and cleared market price for

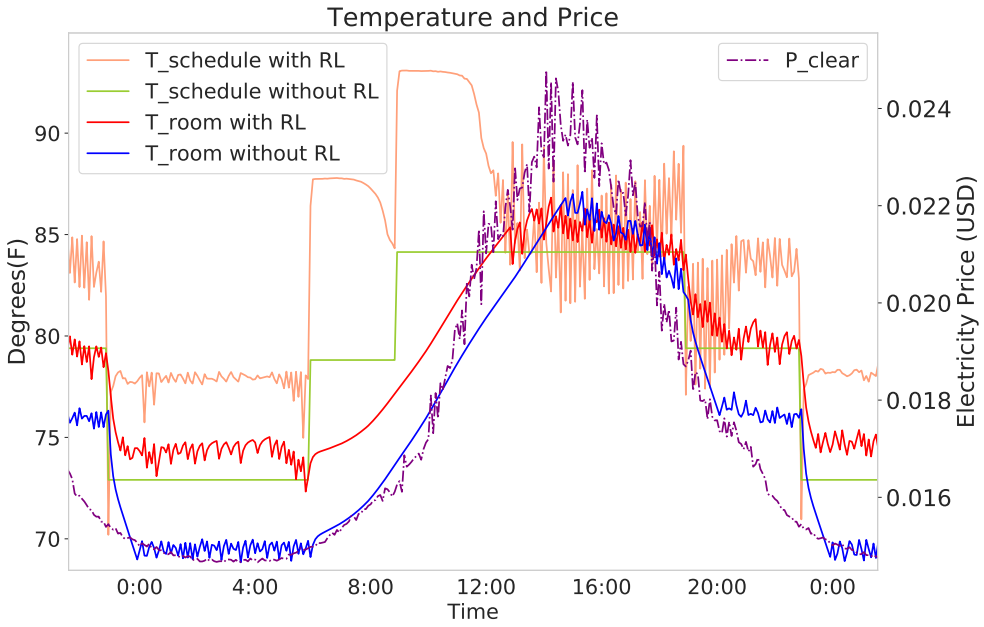


Figure 62: Room temperature with and without RL agent

electricity. The purple dashed-line demonstrates the cleared market price, green and orange lines show the fixed and changing temperature schedules, respectively, and red and blue lines are for the HVAC temperatures with and without RL-agent control, respectively. The room temperature is rising along with the increasing outside temperature from 8am to noon and it continues to rise until it triggers the HVAC to cool the room. The HVAC control with RL agent predicts the afternoon increase in the cleared market price; therefore, there is sudden drop in the temperature schedule at 12:00 with RL agent, and the temperature schedule then continues to drop below the current room temperature. As a result, the HVAC starts cooling the house a little earlier than the original control without RL agent before the price peak is reached at around 14:00. Specifically, just after 12:00, the red line starts to drop and fluctuate, before the blue line. At every time step, the RL agent controls the HVAC output to minimize the deviation of the room temperature from the original schedule

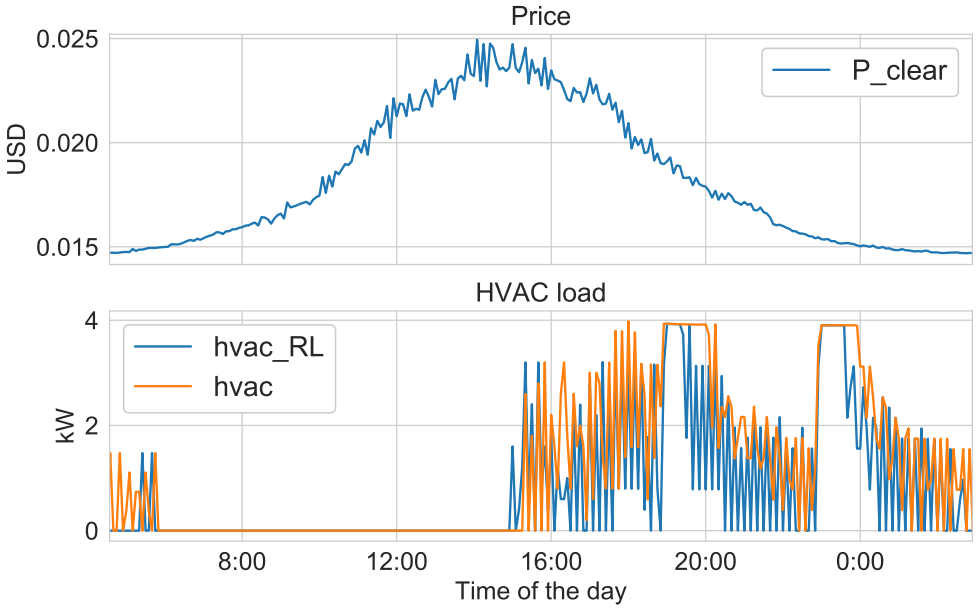
while aiming to consume more power for HVAC at relatively lower price. Fig. 63 demonstrates the energy consumption of a single HVAC controlled with (blue) and without (orange) the RL agent at different time of a day.

We observe that compared to the HVAC controlled without RL agent, HVAC controlled with RL agent consumes more power (higher HVAC load) when the price is low (Fig. 63 (b) around 6:00) and less power (lower HVAC load) when the price is high (Fig. 63 (a) around 20:00). Additionally, Fig. 64 shows the aggregated loads of 306 HVACs controlled with (blue) and without (orange) the RL agent.

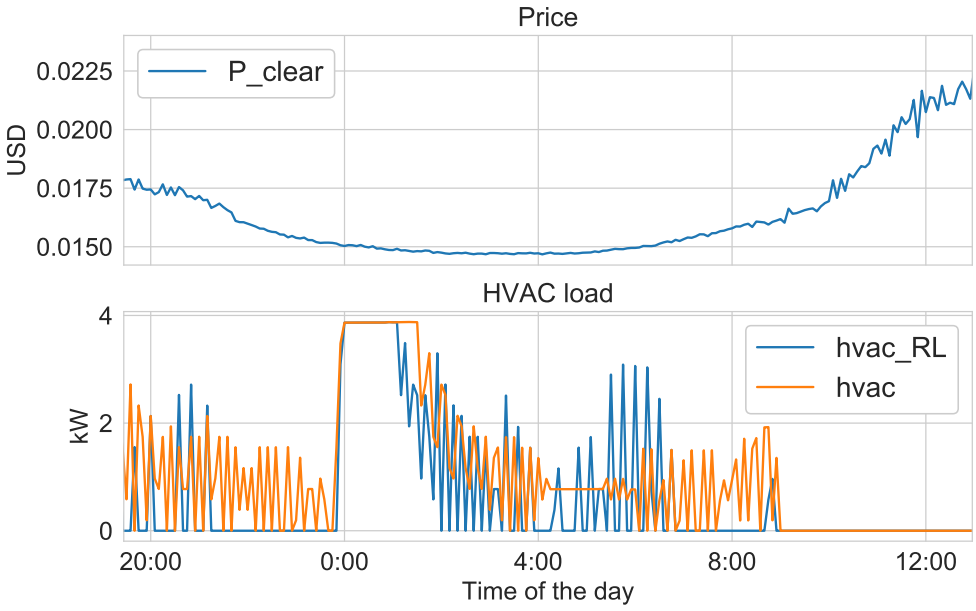
In this figure, the total HVAC load is less when the price is high around 14:00 to 18:00. The HVACs consume a little more power during the time when the price is relatively low such as 0:00 to 4:00. Similar to what we observe in Fig. 63, HVACs controlled with RL agents aim to save more energy at higher market prices.

Recall from (4.3) that α value is chosen to balance between the consumed energy cost of HVAC and the comfort level of the customers. We define the minimization of customer discomfort as the minimization of the deviation of the temperature schedule from the original schedule. Parameter α takes values between 0 and 1 and as its value increases, customers care more about the energy cost. Here we compare again the HVAC control methods with (blue) and without (orange) RL agent for different α values. More specifically, Fig. 65 is the box plot of the weekly cost of consumed power by HVAC. The green dashed lines show the weekly mean values. As can be observed from this figure the RL agent saves more money compared to the control without RL agent; saving increases as α increases. For example, the CSF is 38.5% when α is 0.5. On the other hand, Fig. 66 is the bar plot of the room temperature deviation from the desired temperature schedule.

The average temperature deviation increases with the increase of α, such that TIF ranges from 42.75% to -28.7%. Fig. 67 shows the room temperature under the control of RL agent with different α.



(a)



(b)

Figure 63: Time varying price and the HVAC load

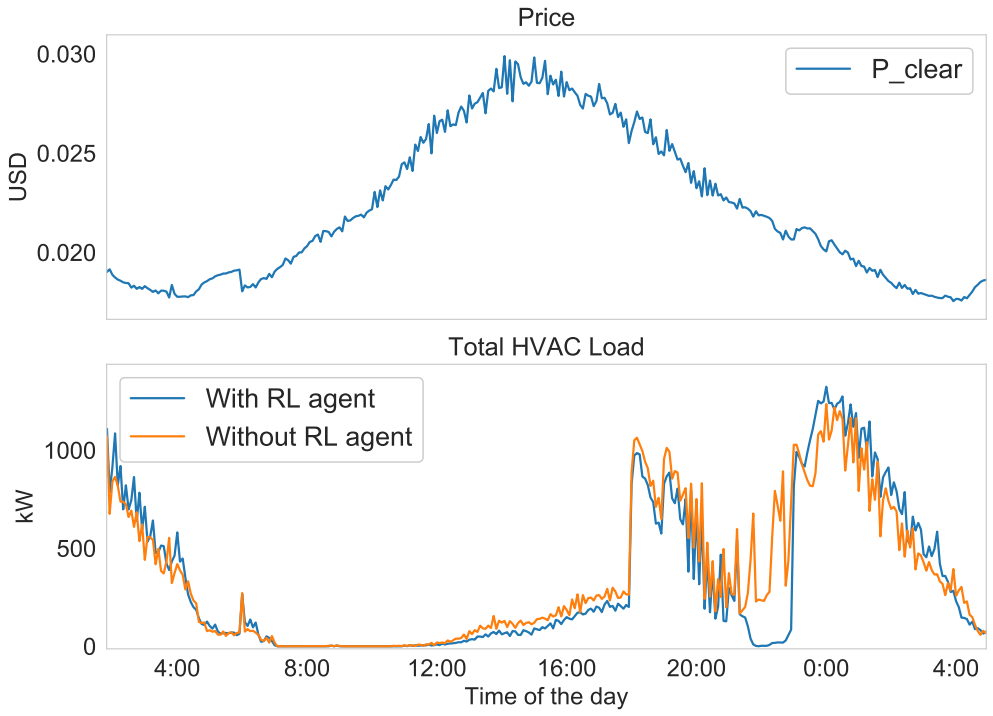


Figure 64: Aggregate responses of the HVACs

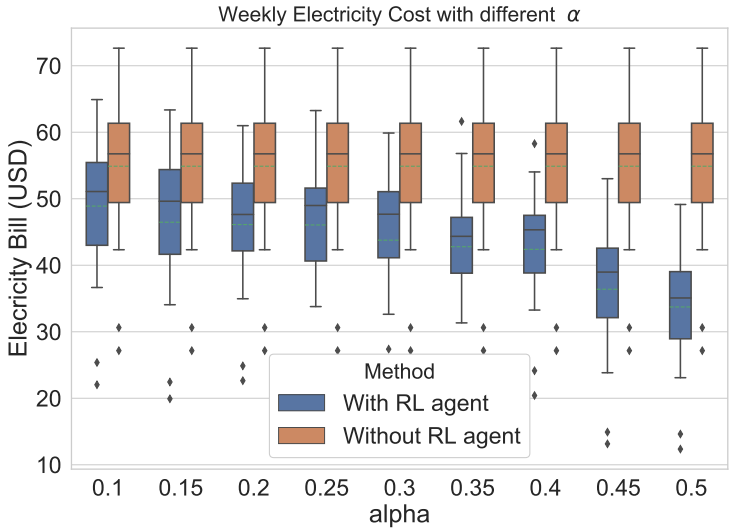


Figure 65: Comparative box plot of the weekly energy cost vs. α

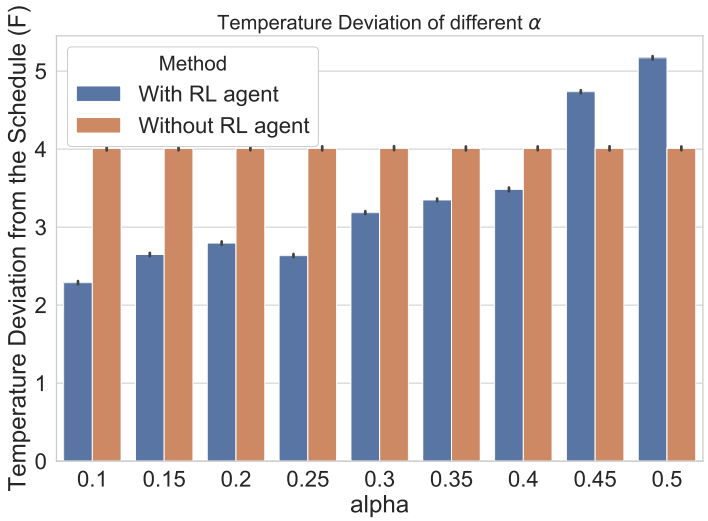


Figure 66: Comparative bar plot of the temperature deviation vs. α

When α increases the deviation of the temperature from the scheduled temperature increases. With larger α, the RL agent is very sensitive to the electricity consumption, in consequence, the RL agent tends to save more energy by sacrificing thermal comfort. For example, as shown in Fig. 66, with α = 0.45 and α = 0.5, the TIF became negative and the temperature deviation is even higher than the case that uses HVAC control without RL. Moreover, Fig. 65 and Fig. 66 also demonstrate that with certain α, the RL agent is able to reduce the energy cost and improve the occupants' comfort at the same time compared to the HVAC control without RL in the TESP, for example, see $\alpha = 0.4$.

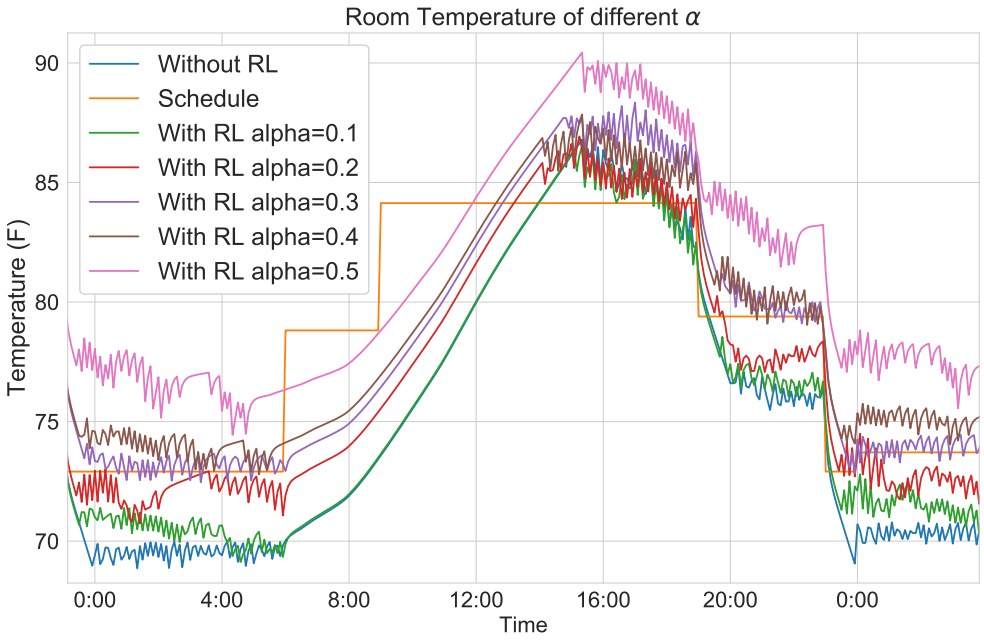


Figure 67: Room temperature of different cases.

4.4.2.5 DDPG RL Performance During a Generation Outage

In the above simulations, the clearing price is at a normal level for most of the time. To evaluate the performance of the HVAC control with RL agent during high-price events, we perform simulations with a bulk generation outage at a certain time of day. We are using the same simulation scenario with 306 HVACs as described above but now there is a generation unit outage from 12:00 to 18:00.

Due to the outage of a main generation unit and the higher cost of the back up generation unit, the Locational Marginal Price (LMP) at the substation bus becomes higher than normal during the outage, leading to a high clearing price as shown in Fig. 68.

With RL agent, the HVAC consumes less power during the outage when the electricity price is at peak. As illustrated in Fig. 68, different from the HVAC control without RL which consumes power during the price peak, the HVAC with RL agent is off beginning around 16:00 and starts to work again when the price drops.

Similar to Fig. 65 for different α values, Fig. 69 shows the box plot of weekly HVAC energy cost with generation outage.

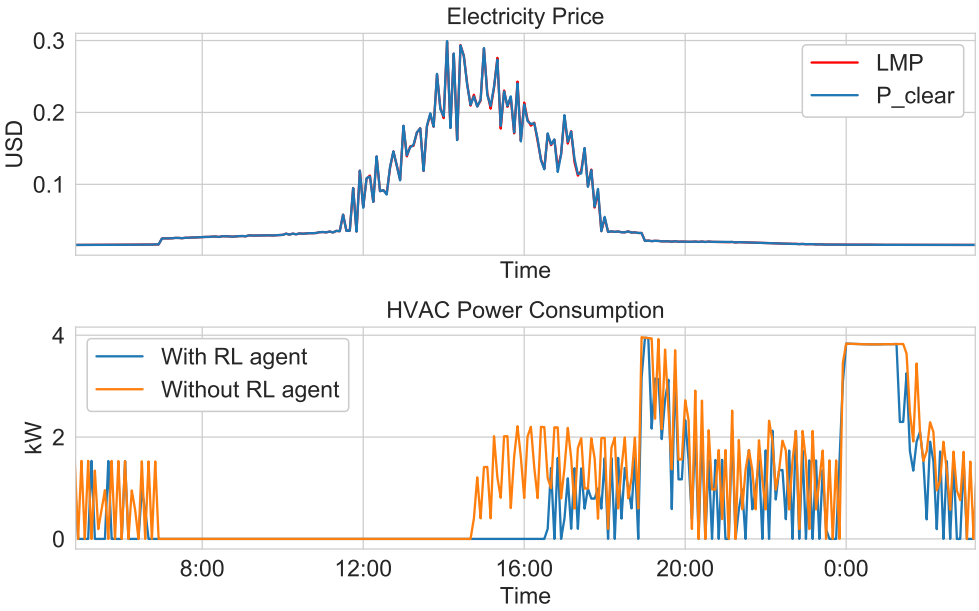


Figure 68: Electricity price and the HVAC power consumption with RL agent during generation outage

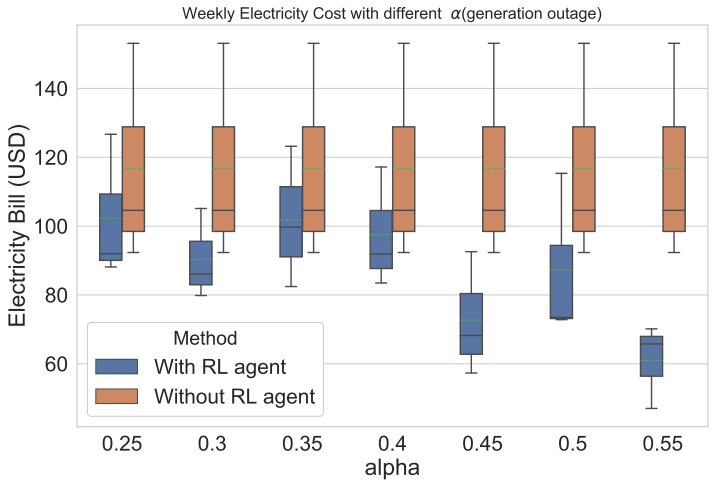


Figure 69: Weekly HVAC energy cost vs. α with generation outage.

As observed in Fig. 69, without RL agent, the energy cost of HVAC is doubled over the normal scenario as demonstrated in Fig. 65. Similar to Fig. 66, Fig. 70 shows the bar plot of room temperature deviation from the desired temperature schedule.

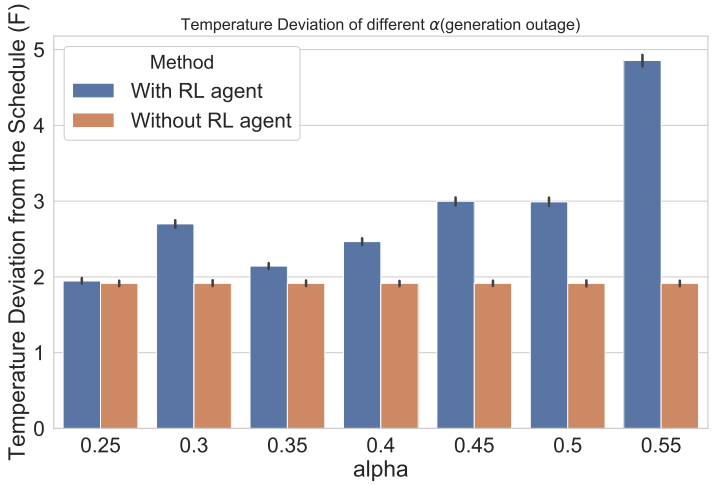


Figure 70: Average temperature deviation vs. α with generation outage.

When $\alpha = 0.25$, the thermal comfort with and without RL are almost the same in these two cases. Note that when $\alpha = 0.25$ the consumers are still able to save 12.7% of HVAC energy cost on average with the RL agent. That is, while the comfort level is preserved, there is more energy savings with HVAC control with RL. When $\alpha = 0.55$, although the average HVAC energy cost is reduced by 50%, the temperature deviation increases a lot. In general, consumers are able to save a greater extent of money with a higher α (emphasis on energy saving), reducing thermal comfort as shown in Fig. 70.

5.0 Research Summary

This dissertation investigated the machine learning applications on the simulation and control framework for electric power distribution systems with transactive agents. A reduced order model for distribution system model with independent dynamically changing end-use load was developed. Further, to reduce the simulation time of large system models with transactive elements in TESP, a reduced order model which aggregated responsive loads was developed. The two proposed reduced order models were demonstrated to be capable of reducing the simulation time while generate appropriate approximation of the full model. In addition, an RL-based approach was developed for precise control of HVAC systems that are participating in the energy market as transactive elements in TESP. The control method is implemented in TESP as a transactive agent, and is demonstrated to achieve intelligent and granular control of the HVACs to maximize the long term reward that balances between electricity cost and end-user satisfaction.

5.1 Contributions

This dissertation describes the following contributions:

1. *Reduced order model for distribution system with independent dynamic loads*:
   The algorithms presented here stochastically generates load shapes for behavioral loads and reproduces the behavior of physics-based models, saving up to 89.3% simulation time when generating the aggregate load of water heaters by stochastic simulations. The stochastic method is able to obtain the load behaviors with several parameters. It has the flexibility to generate a behavior model for any population size of loads, once the stochastic model is built for that mixture of load types. The QSTS simulation results of the test case showed that segment substitution for a model with dynamic independent loads reduces the computation time of QSTS simulation of the full model further, by 64.3% compared to the full-model benchmark. The average error is on the order of 0.002 pu voltage. The worst case error is less than 0.01 pu voltage. The old simplified method reduces the time by 95%, with sufficient accuracy for PV

studies. However, it produces larger error with a mixture of more diverse and independent load behaviors. Under these conditions, the proposed method improves the accuracy significantly compared with the old simplification method without independent load compensation.

2. *Reduced order model for transactive systems:*

   A reduced order model for the distribution system model with transactive elements in TESP was developed. The proposed reduced order model utilizes RNN with LSTMs to collectively generate bids and adjust the aggregate responsive load in every market clearing cycle. The reduced order model is integrated and tested through TESP as an aggregate responsive load (ARL) agent. The ARL agent is capable of communicating with the other agents and generating similar transactive load behaviors as the full model while retaining the bidding functions and price responsive features of the transactive loads. It is demonstrated that the model generalizes well across time and number of houses. Simulation results showed that the reduced order model is able to generate rational results similar to the full TESP model while achieving up to 92.3% of simulation time reduction with 0.03 mean absolute error. Further, the time reduction is expected to be more significant due to the nonliner growth of simulation time for the full model. The proposed ARL agent can also be used to scale up to a large model for examining bulk system impacts. Moreover, simulations could focus on a single detailed feeder while others on the same substation use the ARL agent.

3. *Reinforcement learning based controller agent for transactive HVACs:*

   a) A market price prediction model is developed through an artificial neural network (ANN). The method produces accurate results compared to the existing methods [102]. This method provides price information to the RL algorithm in this dissertation that is developed for HVAC control.

   b) A DDPG RL based model-free control algorithm is developed for the optimal deterministic decision making to adapt the schedule setting of HVACs participating in the market as transactive elements through balancing between energy cost and consumer satisfaction.

   c) The RL-based control approach is implemented and tested in HVAC systems in residential building models through the Transactive Energy Simulation Platform (TESP) [2] as a RL-agent. Through multiple experiments, the effect of control algorithm parameter selection is demonstrated, and the proposed method is compared with the transactive HVAC con-

troller that is currently implemented in TESP. Two metrics, electricity cost saving factor (CSF) and thermal comfort improvement factor (TIF), have been introduced to quantify the performance of the RL agent. It is demonstrated that the proposed control method not only saves the electricity cost but also improves the customers' comfort at the same time with more precise control.

5.2 Scope of Further Research

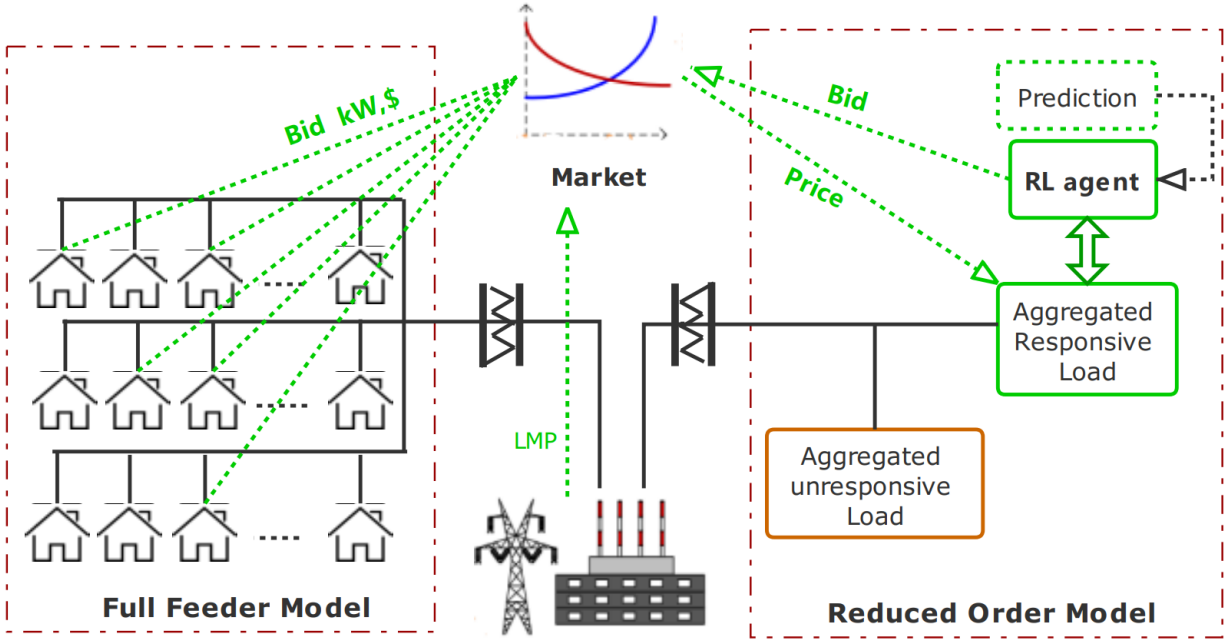


Figure 71: RL agent on aggregated responsive load in reduced order model

The RL agent can be applied on the aggregated responsive load at the substation level. In the full feeder model, each house participates in the market individually. For a large system with multiple distribution feeders, a feeder can be represented by a reduced order model as shown in Fig. 71, with only an aggregated unresponsive load and an aggregated responsive load. This will reduce the number of agents dramatically. The aggregated responsive load is able to participate in

the market by submitting a probabilistically sampled bidding price vs. quantity curve and should be able to respond to the clearing price and update the collective transactive behaviour distributions accordingly. Further, RL agents can be applied on the aggregated responsive load to adjust the bidding and response in a smart way, similar to the individual HVAC example. The feedback to the learning agent can be the approximation of certain collective metrics, e.g., average thermal comfort for large number of houses.

The legacy ramp (temperature vs price) controller can be replaced by RL-based algorithms. These algorithms can be trained off-line using past experiences, which is very useful when large amounts of data are available. The learning agent can be integrated with TESP as an RL agent for different transactive elements and different optimization objectives:

1. **Water Heaters**: Water heaters have the ability to store energy in their water tanks without impacting the comfort of the end users. This feature makes them a prime candidate for residential demand response. RL can be applied to minimize the long term cost of energy consumption.
2. **EV Charging**: Demand response approaches may reduce long-term charging/discharging costs of plug-in electric vehicles. Failing to meet the user needs would lead to negative rewards that are part of the learning process.
3. **Distributed Generation and Battery Energy Management**: RL can be applied to the control of distributed generation resources with electrical storage. The electricity consumption and the battery health can be taken into consideration as a decision making problem to find the optimal charging strategy with real time price forecasting.

5.3 Publications

Published

- Liu, Reiman, Akcakaya, McDermott, "Distribution system segmented model simplification with independent dynamically changing end-use loads," *Electric Power Systems Research*

In Review

- Liu, Akcakaya, McDermott, ”Automated Control of Transactive HVACs in Energy Distribution Systems,” *submitted to IEEE Transactions on Smart Grid*

In Preparation

- Liu, Akcakaya, McDermott, ”Reduced Order Model for Transactive Systems”.

Appendix A Game Theory Applications

A.1 Problem Formulation

α	Real time price determined by the coordinator
p	Load consumption of the consumers (kW)
K	The comfort level of the consumers
θ	weight coefficient of comfort on the consumers' utility
u_s	Seller's utility
u_c	Consumer's utility
$\sigma_s \in \Delta(A)$	mixed strategy of the seller
$\sigma_c \in \Delta(P)$	mixed strategy of the consumer
β	power generation cost $\beta(p) = c_2 p^2 + c_1 p + c_0$

For the consumers and sellers in a power system, we define a multi-time interval T=1,2,...,t_{max}, $t_{max} \geq 2$, each interval is a positive fixed duration. Denote p_{max} as the energy consumption for the consumer if the appliance is working at maximum power during the entire time period ($0 \leq p \leq p_{max}$). Let $\{\alpha \in S_\alpha, \alpha = (\alpha_1, \alpha_2, ..., \alpha_{t_{max}})\}$ and $\{p \in S_p, p = (p_1, p_2, ..., p_{t_{max}})\}$ be the decision variables for the seller and consumers respectively, α_n and P_n is the decision variables of the nth time slot for n=1,2,...,t_{max}. The consumers aim to find the optimal strategies in response to the seller's price in order to maximize the benefit by control the transactive elements such as HVACs. This optimization problem is formulated as:

$$\max_{\alpha \in S_\alpha} \frac{1}{T} \sum_{T=0}^{T-1} \{\alpha(t)p(t) - \beta(p(t)) \tag{A.1}$$

$$p \in \arg\max_{p \in S_p} \{\frac{1}{T} \sum_{T=0}^{T-1} \{\theta \cdot K(p(t)) - \alpha(t) \cdot p(t)\}\} \tag{A.2}$$

$$s.t. \qquad 0 \leq p(t) \leq p_{\max}, \alpha^{\min} \leq \alpha(t) \leq \alpha_{\max}$$

$$\begin{array}{l} p \sim X(\mu_1, {\sigma_1}^2) \\ or \\ \alpha \sim Y(\mu_2, {\sigma_2}^2) \end{array}$$

From the TESP simulation, we can obtain the probability distribution for the power consumption of the consumers p for a given electricity price α from the seller, the probability distribution for the price of the seller α for a given load p from the consumers.

A.2 Mixed Strategy of Repeat Games

A.2.1 Zero-sum Matrix Game

In Eq. (1) and (2), if $\beta(p(t))$ is equal to $\theta \cdot K(p(t))$. The problem can be turned into a zero-sum game. A mixed strategy profile (x^*, y^*) is a mixed strategy Nash equilibrium if and only if

$$(x^*)^T A y^* \geq x^T A y^*, \qquad \forall x \in X \tag{A.3}$$

$$(x^*)^T B y^* \geq (x^*)^T B y, \qquad \forall y \in Y \tag{A.4}$$

For zero-sum game, $B = -A$, so Eq. (4) becomes

$$(x^*)^T A y^* \leq (x^*)^T A y, \qquad \forall y \in Y \tag{A.5}$$

Combining the preceding,

$$x^T A y^* \leq (x^*)^T A y^* \leq (x^*)^T A y, \qquad \forall y \in Y \tag{A.6}$$

(x^*, y^*) is a saddle point of the function $x^T A y$ defined over $X \times Y$. A mixed strategy profile (x^*, y^*) is a mixed strategy Nash equilibrium if and only if

$$(x^*)^T A y^* = \inf_{y \in Y} \sup_{x \in X} x^T A y = \sup_{x \in X} \inf_{y \in Y} x^T A y \tag{A.7}$$

$(x^*)^T A y^*$ is the value of the game.

Strong Duality: P_1 wishes to choose x to minimize $x^T Ay$ ($y^T Ax$). P_2 wishes to choose y to maximize $x^T Ay$. From the point of view of P_1, assuming his strategy x is known to P_2, then P_2 will choose y to maximize $x^T Ay$,

$$\sup\{x^T Ay \,|\, y \geq 0, 1^T y = 1\} = \max_{i=1,\ldots,m} \quad (A^T x)_i$$

P_1 will choose x to minimize the worst case payoff to P_2, there is a mixed strategy for P_1 such that average loss of P_1 is at most a certain value. matter what P_2 does,

$$\begin{aligned} \min \quad & \max_{i=1,\ldots,m} \quad (A^T x)_i \\ s.t. \quad & x \geq 0, \quad \mathbf{1}^T x = 1 \end{aligned} \tag{A.8}$$

Similarly, from the point of view of P_2, assuming his strategy y is known to P_1, then P_1 will choose x to minimize $x^T Ay$,

$$\inf\{x^T Ay \,|\, x \geq 0, 1^T x = 1\} = \min_{i=1,\ldots,n} \quad (Ay)_i$$

P_2 will choose y to maximize his gain from P_1,

$$\begin{aligned} \max \quad & \min_{i=1,\ldots,n} \quad (Ay)_i \\ s.t. \quad & y \geq 0, \quad \mathbf{1}^T y = 1 \end{aligned} \tag{A.9}$$

Formulating problem (8) and (9) as LPs,

(i) x^* satisfies (8) if and only if for r^*,(x^*, r^*) solve the LP

$$\begin{aligned} \min \quad & r \\ s.t. \quad & x \geq 0, \quad \mathbf{1}^T x = 1 \\ & A^T x \leq r\mathbf{1} \end{aligned} \tag{A.10}$$

(ii) y^* satisfies (9) if and only if for s^*,(y^*, s^*) solve the LP

$$\begin{aligned} \max \quad & s \\ s.t. \quad & y \geq 0, \quad \mathbf{1}^T y = 1 \\ & Ay \geq s\mathbf{1} \end{aligned} \tag{A.11}$$

(iii) The LP's in (10)and (11) are duals of each other and the optimal objective value should be the same.

The conditions that the problem can be defined as a zero-sum game: In a zero-sum game, any advantage gained by the opponent is a lose to the player (the sum of the utilities in each entry of the payoff matrix is zero). A non-zero-sum game is a situation where one's win does not necessarily mean the other's loss, and one's loss does not necessarily mean that the other party wins. In a Non-Zero-Sum Game, all parties could gain, or all parties could lose.

The transformation between zero-sum and nonzero-sum game: Reference [103] demonstrates how one can define a transformation of a non-zero sum game into a zero sum game by introducing a passive player into a game whose payoff depends on the actions of active players. In a transformed game, each participant plays against all other players, including the passive player. The advantage of this approach is that the transformed game is zero-sum and has an equilibrium solution. The optimal strategy and the value of the new game, however, can be different from strategies that are rational in the original game. Using prisoner's dilemma as an example:

$$A = \begin{bmatrix} -1 & -10 \\ 0 & -6 \end{bmatrix} \qquad B = \begin{bmatrix} -1 & 0 \\ -10 & -6 \end{bmatrix}$$

The classic 'rational solution to this game is to cooperate (use strategy 1 with probability p=1), and the value of the game is,

$$E(x_1) = -1$$

The game is not zero sum since

$$A + B = \begin{bmatrix} -2 & -10 \\ -10 & -12 \end{bmatrix} \quad \neq \quad 0$$

Introduce a passive player, whose payoff is

$$C = -\frac{1}{2}A + B = \begin{bmatrix} -2 & -10 \\ -10 & -12 \end{bmatrix} \quad = \quad \begin{bmatrix} 1 & 5 \\ 5 & 6 \end{bmatrix}$$

The utilities of the prisoners are transformed to $\tilde{A} = A + C$ and $\tilde{B} = B + C$

$$\tilde{A} = \begin{bmatrix} 0 & -5 \\ 5 & 0 \end{bmatrix} \qquad \tilde{B} = \begin{bmatrix} 0 & 5 \\ -5 & 0 \end{bmatrix}$$

In this representation, the game becomes zero-sum, and it has solution $p_1 = p_2 = 0.5$ and the value $E(\tilde{A}) = 0$. This expected payoff to each player is higher than that of the strategy $p_1 = 1$, but it is independent of the decision of the other player to defect.

A.2.2 Nonzero-Sum Bimatrix Games

Suppose the same game is played over and over again and the final outcome is determined by averaging the outcomes of individual plays.

2 players: N={seller, consumers}

The set of pure strategy for the 2 players are:

$$A = (\alpha_1, \alpha_2, ..., \alpha_n), \qquad P = (p_1, p_2, ..., p_m)$$

Let $U_s = (u_s^{ij})$ and $U_c = (u_c^{ij})$ be two $n \times m$ matrix represent the utility of each player corresponding to each pair of actions. The bimatrix game (U_s^{ij}, U_c^{ij}) is a two person game in normal form, where the supplier and the consumer choose their strategy independently. The utility of the supplier and consumer under pure strategies are:

$$\begin{aligned} u_s^{ij} &= \alpha_i p_j - \beta(p_j) \\ u_c^{ij} &= \theta \cdot K(p_j) - \alpha_i p_j \end{aligned} \tag{A.12}$$

Let σ_s and σ_c denote the mixed strategy of the two player, σ_s^i and σ_c^j be the probability that the supplier and the consumer choose the $i-th$ price and $j-th$ load consumption respectively, where

$$\sum_{i=1}^{n} \sigma_{si} = 1, \quad \sigma_{si} \geq 0 \qquad \sum_{j=1}^{m} \sigma_{cj} = 1, \quad \sigma_{cj} \geq 0$$

The expected payoff to supplier is:

$$\sigma_s^T U_s \sigma_c = \sum\nolimits_{i=1}^{n} \sum\nolimits_{j=1}^{m} \sigma_s^i u_s^{ij} \sigma_c^j \tag{A.13}$$

Similarly the expected payoff to consumer is:

$$\sigma_s^T U_c \sigma_c = \sum\nolimits_{i=1}^{n} \sum\nolimits_{j=1}^{m} \sigma_s^i u_c^{ij} \sigma_c^j \tag{A.14}$$

First, considering an inner mixed Nash equilibrium (σ_s^*, σ_c^*), where $\sigma_{si}^* \geq 0$ for all i and $\sigma_{cj}^* \geq 0$ for all j (all pure strategies are used with positive probability).

Using the equivalent characterization of a mixed strategy Nash equilibrium (i.e., all pure strategies in the support of a Nash equilibrium strategy yields the same payoff, which is also greater than or equal to the payoffs for strategies outside the support), we have

$$u_{s1}\sigma_c^* = u_{si}\sigma_c^*, \quad i = 2, ..., n \tag{A.15}$$

$$(\sigma_s^*)^T u_{c1} = (\sigma_s^*)^T u_{ci}, \quad j = 2, ..., m \tag{A.16}$$

where u_{si} denote the rows of payoff matrix U_s and u_{ci} denote the columns of payoff matrix U_c. By Eq.(6) and Eq.(7), a system of linear equations can be solved. However, the assumption that every strategy is played with positive probability is a very restrictive assumption. Most games do not have totally mixed Nash equilibria.

Transform it into a Bi-linear Programming Problem: A mixed strategy profile (σ_s^*, σ_c^*) is a mixed Nash equilibrium of the bimatrix game (U_s, U_c) if and only if there exists a pair (r^*, s^*) such that $(\sigma_s^*, \sigma_c^*, r^*, s^*)$ is a solution to the following bi-linear programming problem:

$$\max \qquad \{{\sigma_s}^T U_s \sigma_c + {\sigma_s}^T U_c \sigma_c - r - s\} \tag{A.17}$$

$$s.t. \qquad U_s\sigma_c \leq r\mathbf{1}_n, \quad U_c^T\ \sigma_s \leq s\mathbf{1}_m, \tag{A.18}$$

$$\sum_i \sigma_{si} = 1, \qquad \sum_j \sigma_{cj} = 1, \tag{A.19}$$

$$\sigma_s \geq 0, \quad \sigma_c \geq 0, \tag{A.20}$$

where,$\mathbf{1}_n$ and $\mathbf{1}_n$ are n and m dimensional vector with all components equal to 1.

Solution of Bi-linear Programming:

$$\begin{aligned} &\max_{x,y,p,q} \quad x(A+B)y - p - q \\ &s.t. \quad xB - qe \leq 0, \qquad ex - 1 = 0\ , \quad x \geq 0, \\ &\qquad Ay - qe \leq 0, \qquad ey - 1 = 0\ , \quad y \geq 0, \end{aligned} \tag{A.21}$$

Solutions of the corresponding problem of bi-linear programming were found by computer, using the gradient method. [104]. A Difference-Index based ranking bi-linear programming approach to Solving bi-matrix games with payoffs of trapezoidal intuitionistic fuzzy numbers are introduced in [105], an enumeration method is presented in [106].

Appendix B Simulation Result

From TESP simulations, extracting the time varying price and the total HVAC load of the houses that participating in the market as shown in Fig. 74. (Values in Fig. 74 are the mean value of the simulation results from 12 month from TESP as shown in Fig. 72 and Fig. 73) Using these data to formulate time dependent the conditional distributions $(\Lambda|p)$ and $(p|\lambda)$. Specifically, $(\Lambda|p)$ is the distribution of the total load of the houses for a given price in a certain range between price λ_a and λ_b. $(\Lambda|p)$ is the distribution of possible prices for a given total load in a certain range between p_e and p_f. Then formulating a mixed strategy game.

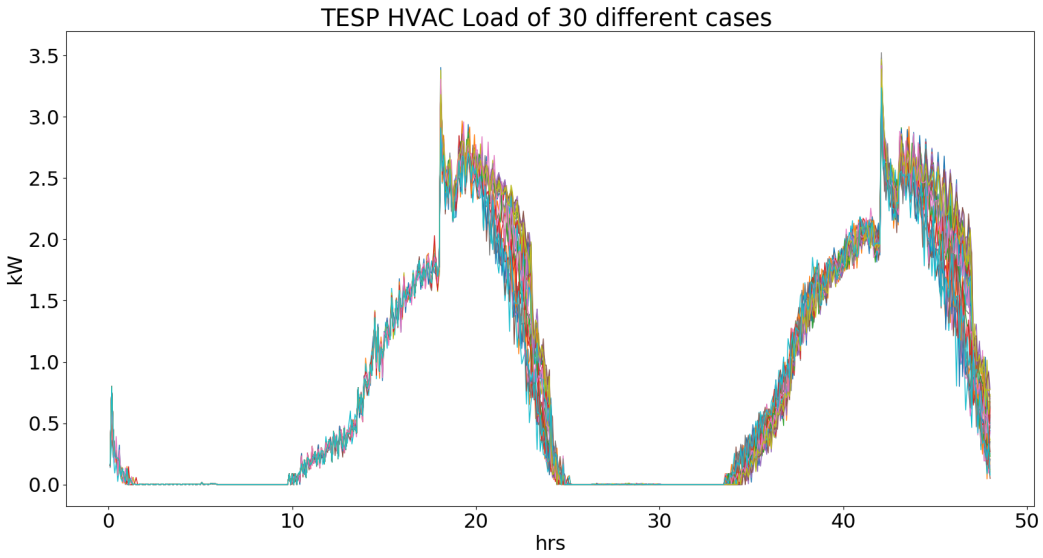


Figure 72: Load of different cases from TESP Simulation

1. From TESP simulations, the following joint distributions (p:load λ: price) are obtained. All the distributions can be turned into time dependent distributions, the number of time intervals can be changed as well:

   $f(P, \Lambda)$ (Joint distribution of cleared price and load)

   $f(P_{bid}, \Lambda)$ (Joint distribution of bidding load and cleared price)

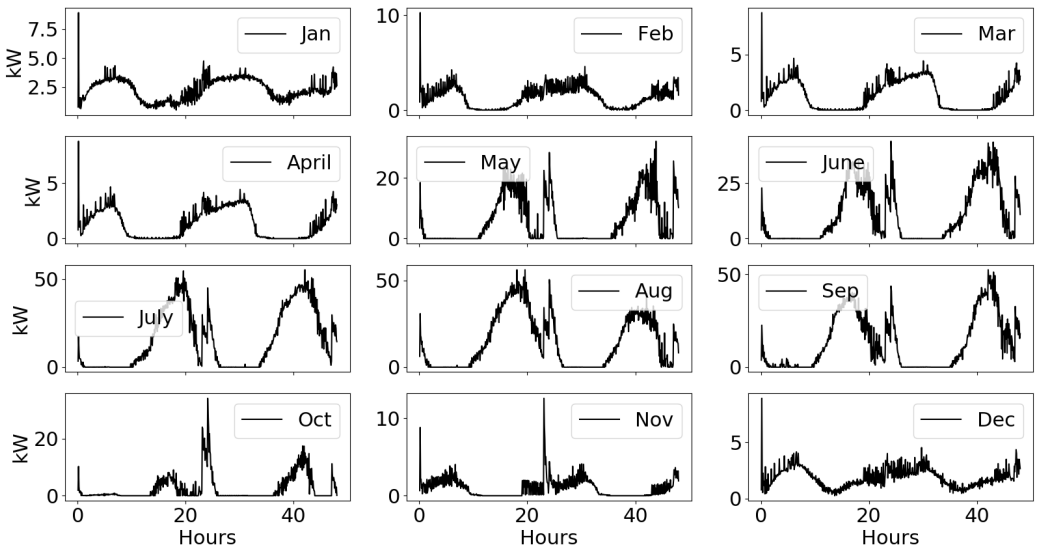
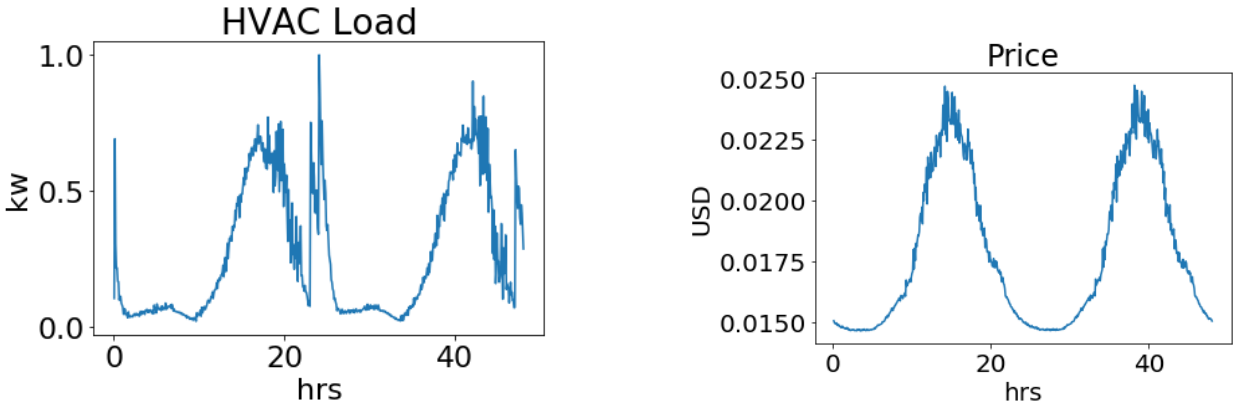


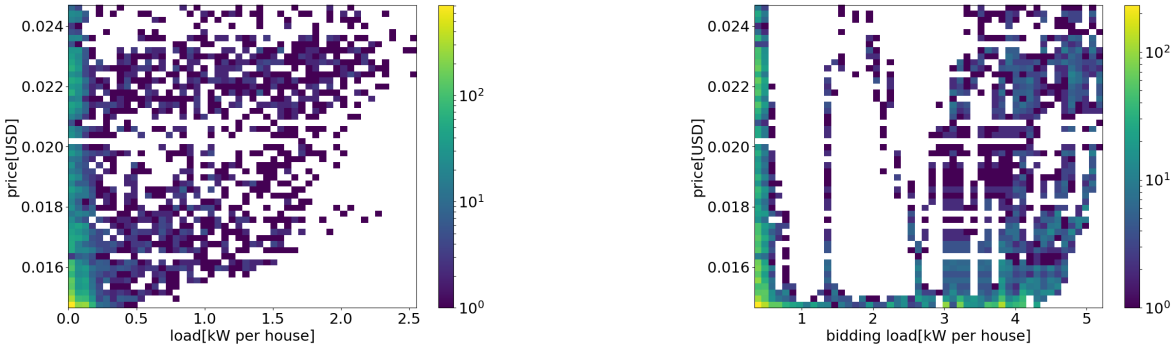
Figure 73: HVAC Load of different month from TESP Simulation



(a) transactive HVAC load of all the houses

(b) time varying price

Figure 74: TESP simulation result (mean of 12 month, load value is normalized)



(a) load vs price

(b) bidding load vs price

Figure 75: 2D histogram Load vs Price

$f(P_{bid}, P)$ (joint distribution of the bidding load and real HVAC load)

$f(P_{bid}, \Lambda_{bid})$ (joint distribution of the bidding load and bidding price).

2. Define an initial p_0 and λ_0
3. Start from time step $t, t = 1, 2, 3, ..., t_{max}$,
   a. Find the strategy set P_t, Λ_t:

      P_t: sample from $f(p|\lambda)(\lambda_{t-1}) \rightarrow p_t^1, p_t^2, ..., p_t^i$

      Λ_t: sample from $f(\lambda|p)(p_{t-1}) \rightarrow \lambda_t^1, \lambda_t^2, ..., \lambda_t^j$

   b. Calculate the payoff matrix by the utility function:

      (i) $A_{ij} = (E[p|\lambda]_{ij} - (p_{bid}|p)_{ij})^2$ minimize the difference between the bidding power and the actual power consumption, where $[E[p|\lambda]]_{ij} = E[p \sim f(p|\lambda_t^j)] = E[p|\lambda_t^j]$ is the expected value of actual power consumption for a given price λ. $[p_{bid}|p]_{ij} = X_{p_{bid}p}(p_t^i) = p_{bid}|p_t^i$ is the bidding load for for a given actual power consumption p.

      (ii) $A_{ij} = [E[\lambda|p]_{ij} - (\lambda_{bid}|\lambda)_{ij}]^2$ minimize the difference between the bidding price and the actual price, where

      $[E[\lambda|p]_{ij} = E[\lambda \sim f(\lambda|p_t^i)] = E[\lambda|p_t^i]$

      $[\lambda_{bid}|\lambda]_{ij} = X_{\lambda_{bid}\lambda}(\lambda_t^j) = \lambda_{bid}|\lambda_t^i$

      (iii) $A_{ij} = a[E[p|\lambda]_{ij} - (p_{bid}|p)_{ij}]^2 + b[E[\lambda|p]_{ij} - (\lambda_{bid}|\lambda)_{ij}]^2$ is the combination of (i) and (ii), where $a + b = 1$.

   c. Find the optimal strategy (probability vector) X_t, Y_t by solving the game:

d. Use the probability X_t, Y_t to calculate p_{t+1} and λ_{t+1}:

   Sampling from P_t with $X_t \rightarrow p_t$

   Sampling from Λ_t with $Y_t \rightarrow \lambda_t$

e. go back to (a) for next time step

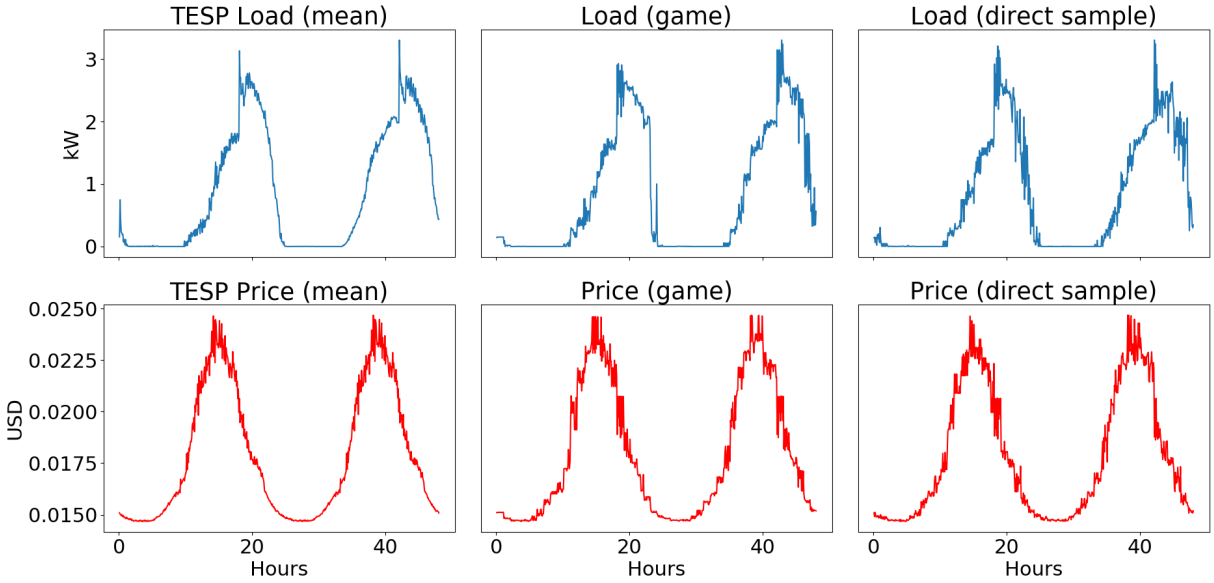


Figure 76: 1 realization of Load and Price

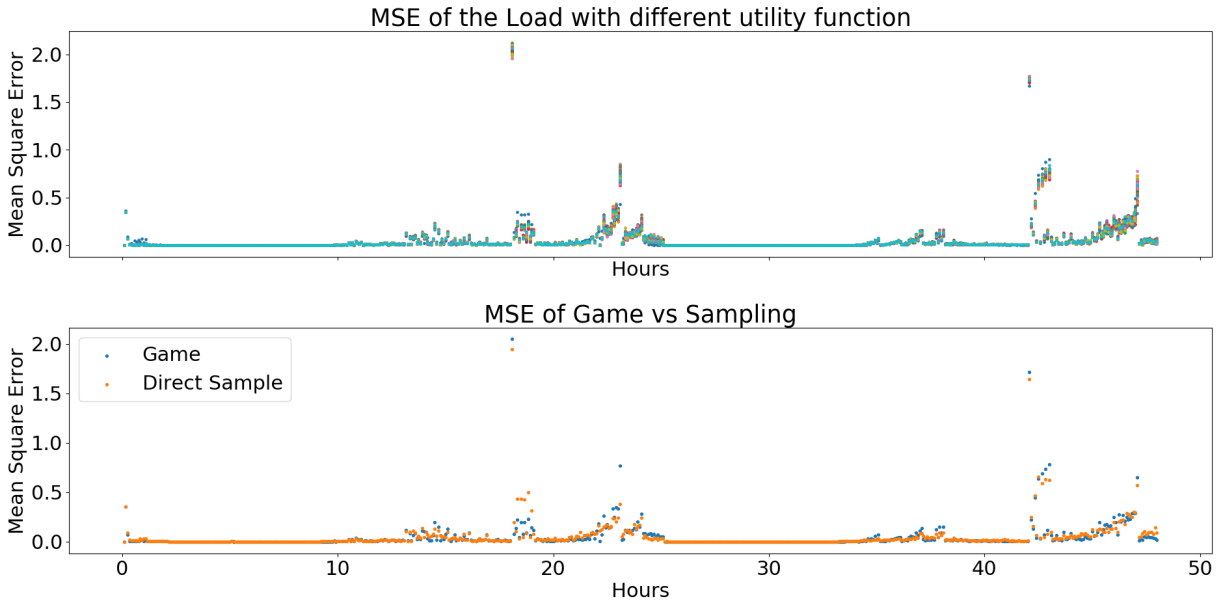


Figure 77: Mean Square Error (50 times)

Bibliography

[1] U-CPSOT Force. Final report on the august 14th blackout in the united states and canada. *Department of Energy and National Resources Canada*, 2004.

[2] Q. Huang, T. E. McDermott, Y. Tang, A. Makhmalbaf, D. J. Hammerstrom, A. R. Fisher, L. D. Marinovici, and T. Hardy. Simulation-based valuation of transactive energy systems. *IEEE Transactions on Power Systems*, 34(5):4138–4147, Sep. 2019.

[3] A. P. Reiman, T. E. McDermott, M. Akcakaya, and G. F. Reed. Electric power distribution system model simplification using segment substitution. *IEEE Transactions on Power Systems*, PP(99):1–1, 2017.

[4] B. A. Mather. Quasi-static time-series test feeder for pv integration analysis on distribution systems. In *2012 IEEE Power and Energy Society General Meeting*, pages 1–8, July 2012.

[5] K. P. Schneider, J. C. Fuller, and D. P. Chassin. Multi-state load models for distribution system analysis. *IEEE Transactions on Power Systems*, 26(4):2425–2433, Nov 2011.

[6] Jeremiah Deboever, Xiaochen Zhang, Matthew J Reno, Robert J Broderick, Santiago Grijalva, and Francis Therrien. Challenges in reducing the computational time of qsts simulations for distribution system analysis. *SAND2017-5743, Albuquerque, NM*, 2017.

[7] Z. Liu, Q. Wu, S. Huang, and H. Zhao. Transactive energy: A review of state of the art and implementation. In *2017 IEEE Manchester PowerTech*, pages 1–6, June 2017.

[8] D. P. Chassin, K. Schneider, and C. Gerkensmeyer. Gridlab-d: An open-source power systems modeling and simulation environment. In *2008 IEEE/PES Transmission and Distribution Conference and Exposition*, pages 1–5, April 2008.

[9] R. Malhame and Chee-Yee Chong. Electric load model synthesis by diffusion approximation of a high-order hybrid-state stochastic system. *IEEE Transactions on Automatic Control*, 30(9):854–860, Sep. 1985.

[10] K. Kalsi, F. Chassin, and D. Chassin. Aggregated modeling of thermostatic loads in demand response: A systems and control perspective. In *2011 50th IEEE Conference on Decision and Control and European Control Conference*, pages 15–20, Dec 2011.

[11] Stephan Koch, Johanna Mathieu, and D.S. Callaway. Modeling and control of aggregated heterogeneous thermostatically controlled loads for ancillary services. *Proc. PSCC*, pages 1–7, 01 2011.

[12] Ning Lu, David P Chassin, and Steve E Widergren. Modeling uncertainties in aggregated thermostatically controlled loads using a state queueing model. *IEEE Transactions on Power Systems*, 20(2):725–733, 2005.

[13] J. Saez-Gallego, M. Kohansal, A. Sadeghi-Mobarakeh, and J. M. Morales. Optimal price-energy demand bids for aggregate price-responsive loads. *IEEE Transactions on Smart Grid*, 9(5):5005–5013, Sep. 2018.

[14] W. Zhang, K. Kalsi, J. Fuller, M. Elizondo, and D. Chassin. Aggregate model for heterogeneous thermostatically controlled loads with demand response. In *2012 IEEE Power and Energy Society General Meeting*, pages 1–8, July 2012.

[15] Hadi Saadat. *Power system analysis*. McGraw-Hill, 1999.

[16] Thomas Ackermann, Göran Andersson, and Lennart Söder. Distributed generation: a definition1. *Electric power systems research*, 57(3):195–204, 2001.

[17] Ali Ipakchi and Farrokh Albuyeh. Grid of the future. *IEEE power and energy magazine*, 7(2):52–62, 2009.

[18] MK Deshmukh and SS Deshmukh. Modeling of hybrid renewable energy systems. *Renewable and Sustainable Energy Reviews*, 12(1):235–249, 2008.

[19] Peter Palensky and Dietmar Dietrich. Demand side management: Demand response, intelligent energy systems, and smart loads. *IEEE transactions on industrial informatics*, 7(3):381–388, 2011.

[20] Sara Deilami, Amir S Masoum, Paul S Moses, and Mohammad AS Masoum. Real-time coordination of plug-in electric vehicle charging in smart grids to minimize power losses and improve voltage profile. *IEEE Transactions on Smart Grid*, 2(3):456–467, 2011.

[21] Arijit Bhowmik, Arindam Maitra, S Mark Halpin, and Joe E Schatz. Determination of allowable penetration levels of distributed generation resources based on harmonic limit considerations. *IEEE Transactions on Power Delivery*, 18(2):619–624, 2003.

[22] Olimpo Anaya-Lara and E Acha. Modeling and analysis of custom power systems by pscad/emtdc. *IEEE transactions on power delivery*, 17(1):266–272, 2002.

[23] Mario Paolone, Carlo Alberto Nucci, Emanuel Petrache, and Farhad Rachidi. Mitigation of lightning-induced overvoltages in medium voltage distribution lines by means of periodical grounding of shielding wires and of surge arresters: Modeling and experimental validation. *IEEE Transactions on Power Delivery*, 19(1):423–431, 2004.

[24] Jing Yong, Liang Chen, and Shuangyan Chen. Modeling of home appliances for power distribution system harmonic analysis. *IEEE Transactions on Power Delivery*, 25(4):3147–3155, 2010.

[25] Johan HR Enslin and Peter JM Heskes. Harmonic interaction between a large number of distributed power inverters and the distribution network. *IEEE transactions on power electronics*, 19(6):1586–1593, 2004.

[26] Mohamed H Albadi and Ehab F El-Saadany. Demand response in electricity markets: An overview. In *2007 IEEE power engineering society general meeting*, pages 1–5. IEEE, 2007.

[27] Mohamed H Albadi and Ehab F El-Saadany. A summary of demand response in electricity markets. *Electric power systems research*, 78(11):1989–1996, 2008.

[28] D. T. Nguyen, H. T. Nguyen, and L. B. Le. Dynamic pricing design for demand response integration in power distribution networks. *IEEE Transactions on Power Systems*, 31(5):3457–3472, Sep. 2016.

[29] Koen Kok and Steve Widergren. A society of devices: Integrating intelligent distributed resources with transactive energy. *IEEE Power and Energy Magazine*, 14(3):34–45, 2016.

[30] Donald J Hammerstrom, Ron Ambrosio, Teresa A Carlon, John G DeSteese, Gale R Horst, Robert Kajfasz, Laura L Kiesling, Preston Michie, Robert G Pratt, Mark Yao, et al. Pacific northwest gridwise^{TM} testbed demonstration projects; part i. olympic peninsula project. Technical report, Pacific Northwest National Lab.(PNNL), Richland, WA (United States), 2008.

[31] Jason C Fuller, Kevin P Schneider, and David Chassin. Analysis of residential demand response and double-auction markets. In *2011 IEEE power and energy society general meeting*, pages 1–7. IEEE, 2011.

[32] Edward G Cazalet. Automated transactive energy (temix). In *Grid-Interop Forum*, 2011.

[33] Koen Kok. The powermatcher: Smart coordination for the smart electricity grid. *TNO, The Netherlands*, pages 241–250, 2013.

[34] Steven E Widergren, Donald J Hammerstrom, Qiuhua Huang, Karanjit Kalsi, Jianming Lian, Atefe Makhmalbaf, Thomas E McDermott, Deepak Sivaraman, Yingying Tang, Arun Veeramany, et al. Transactive systems simulation and valuation platform trial analysis. Technical report, Pacific Northwest National Laboratory (PNNL), Richland, WA (US), 2017.

[35] R Lincoln. Pypower: port of matpower in python. *URL: https://github. com/rwl/PY-POWER, last accessed on*, 25(12):2019, 2019.

[36] Drury B Crawley, Linda K Lawrie, Frederick C Winkelmann, Walter F Buhl, Y Joe Huang, Curtis O Pedersen, Richard K Strand, Richard J Liesen, Daniel E Fisher, Michael J Witte, et al. Energyplus: creating a new-generation building energy simulation program. *Energy and buildings*, 33(4):319–331, 2001.

[37] Selim Ciraci, Jeff Daily, Jason Fuller, Andrew Fisher, Laurentiu Marinovici, and Khushbu Agarwal. Fncs: a framework for power system and communication networks co-simulation. In *Proceedings of the symposium on theory of modeling & simulation-DEVS integrative*, page 36. Society for Computer Simulation International, 2014.

[38] J. Y. Joo and M. D. Ilić. An information exchange framework utilizing smart buildings for efficient microgrid operation. *Proceedings of the IEEE*, 104(4):858–864, April 2016.

[39] C. Ogwumike, M. Short, and F. Abugchem. Heuristic scheduling of multiple smart home appliances: Utility planning perspective. In *2016 International Conference for Students on Applied Engineering (ICSAE)*, pages 237–241, Oct 2016.

[40] L. Yu, H. Li, X. Feng, and J. Duan. Nonintrusive appliance load monitoring for smart homes: recent advances and future issues. *IEEE Instrumentation Measurement Magazine*, 19(3):56–62, June 2016.

[41] D. Forfia, M. Knight, and R. Melton. The view from the top of the mountain: Building a community of practice with the gridwise transactive energy framework. *IEEE Power and Energy Magazine*, 14(3):25–33, May 2016.

[42] H. Hao, C. D. Corbin, K. Kalsi, and R. G. Pratt. Transactive control of commercial buildings for demand response. *IEEE Transactions on Power Systems*, 32(1):774–783, Jan 2017.

[43] Abdullah Bokhari, Ali Alkan, Rasim Dogan, Marc Diaz-Aguiló, Francisco De Leon, Dariusz Czarkowski, Zivan Zabar, Leo Birenbaum, Anthony Noel, and Resk Ebrahem Uosef. Experimental determination of the zip coefficients for modern residential, commercial, and industrial loads. *IEEE Transactions on Power Delivery*, 29(3):1372–1381, 2014.

[44] K. P. Schneider and J. C. Fuller. Detailed end use load modeling for distribution system analysis. In *IEEE PES General Meeting*, pages 1–7, July 2010.

[45] R. T. Guttromson, D. P. Chassin, and S. E. Widergren. Residential energy resource models for distribution feeder simulation. In *2003 IEEE Power Engineering Society General Meeting (IEEE Cat. No.03CH37491)*, volume 1, page 113 Vol. 1, July 2003.

[46] A. Gaikwad, P. Markham, and P. Pourbeik. Implementation of the wecc composite load model for utilities using the component-based modeling approach. In *2016 IEEE/PES Transmission and Distribution Conference and Exposition (T D)*, pages 1–5, May 2016.

[47] J. A. Mueller, A. Sankara, J. W. Kimball, and B. McMillin. Hidden markov models for non-intrusive appliance load monitoring. In *2014 North American Power Symposium (NAPS)*, pages 1–6, Sept 2014.

[48] T. Zia, D. Bruckner, and A. Zaidi. A hidden markov model based procedure for identifying household electric loads. In *IECON 2011 - 37th Annual Conference of the IEEE Industrial Electronics Society*, pages 3218–3223, Nov 2011.

[49] A. Zoha, A. Gluhak, M. Nati, and M. A. Imran. Low-power appliance monitoring using factorial hidden markov models. In *2013 IEEE Eighth International Conference on Intelligent Sensors, Sensor Networks and Information Processing*, pages 527–532, April 2013.

[50] D. Bruckner and R. Velik. Behavior learning in dwelling environments with hidden markov models. *IEEE Transactions on Industrial Electronics*, 57(11):3653–3660, Nov 2010.

[51] M. Zeifman. Disaggregation of home energy display data using probabilistic approach. *IEEE Transactions on Consumer Electronics*, 58(1):23–31, February 2012.

[52] Davis Montenegro, Gustavo A Ramos, and Seddik Bacha. A-diakoptics for the multicore sequential-time simulation of microgrids within large distribution systems. *IEEE Transactions on Smart Grid*, 8(3):1211–1219, 2017.

[53] Muhammad Umer Qureshi, Santiago Grijalva, Matthew J Reno, Jeremiah Deboever, Xiaochen Zhang, and Robert J Broderick. A fast scalable quasi-static time series analysis method for pv impact studies using linear sensitivity model. *IEEE Transactions on Sustainable Energy*, 10(1):301–310, 2019.

[54] Zachary K Pecenak, Vahid R Disfani, Matthew J Reno, and Jan Kleissl. Multiphase distribution feeder reduction. *IEEE Transactions on Power Systems*, 33(2):1320–1328, 2018.

[55] R. C. Dugan and T. E. McDermott. An open source platform for collaborating on smart grid research. In *2011 IEEE Power and Energy Society General Meeting*, pages 1–7, July 2011.

[56] *EPRI Feeder J1,Distributed PV (DPV) Monitoring and Feeder Analysis*, 2013. Available: dpv.epri.com.

[57] B Liu, M Akcakaya, AP Reiman, and TE McDermott. Distribution system segmented model simplification with independent dynamically changing end-use loads. *Electric Power Systems Research*, 188:106528, 2020.

[58] David P Chassin, Jason C Fuller, and Ned Djilali. Gridlab-d: An agent-based simulation framework for smart grids. *Journal of Applied Mathematics*, 2014, 2014.

[59] Ray D Zimmerman, Carlos E Murillo-Sánchez, and Deqiang Gan. Matpower: A matlab power system simulation package. *Manual, Power Systems Engineering Research Center, Ithaca NY*, 1, 1997.

[60] RG Pratt and ZT Taylor. Development and testing of an equivalent thermal parameter model of commercial buildings from time-series end-use data. *Pacific Northwest Laboratory, Richland, WA*, 1994.

[61] Zachary C Lipton, John Berkowitz, and Charles Elkan. A critical review of recurrent neural networks for sequence learning. *arXiv preprint arXiv:1506.00019*, 2015.

[62] Sepp Hochreiter. The vanishing gradient problem during learning recurrent neural nets and problem solutions. *International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems*, 6(02):107–116, 1998.

[63] Alex Sherstinsky. Fundamentals of recurrent neural network (rnn) and long short-term memory (lstm) network. *Physica D: Nonlinear Phenomena*, 404:132306, 2020.

[64] Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural computation*, 9(8):1735–1780, 1997.

[65] T. N. Sainath, O. Vinyals, A. Senior, and H. Sak. Convolutional, long short-term memory, fully connected deep neural networks. In *2015 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 4580–4584, 2015.

[66] Nikolay Laptev, Jason Yosinski, Li Erran Li, and Slawek Smyl. Time-series extreme event forecasting with neural networks at uber. In *International Conference on Machine Learning*, volume 34, pages 1–5, 2017.

[67] Jie Chen, Guo-Qiang Zeng, Wuneng Zhou, Wei Du, and Kang-Di Lu. Wind speed forecasting using nonlinear-learning ensemble of deep learning time series prediction and extremal optimization. *Energy Conversion and Management*, 165:681 – 695, 2018.

[68] Wei Zhang, Jianming Lian, Chin-Yao Chang, and Karanjit Kalsi. Aggregated modeling and control of air conditioning loads for demand response. *IEEE transactions on power systems*, 28(4):4655–4664, 2013.

[69] Swathi Battula, Leigh Tesfatsion, and Thomas E. McDermott. An ercot test system for market design studies. *Applied Energy*, 275:115182, 2020.

[70] B. Bhattarai, M. Maharjan, S. Hanif, M. Cai, and R. Pratt. Transactive electric water heater agent: Design and performance evaluation. In *2020 IEEE Power Energy Society Innovative Smart Grid Technologies Conference (ISGT)*, pages 1–5, 2020.

[71] Stephen Wilcox and William Marion. Users manual for tmy3 data sets. 2008.

[72] Pierluigi Siano. Demand response and smart grids—a survey. *Renewable and Sustainable Energy Reviews*, 30:461 – 478, 2014.

[73] U.s. energy information administration. Accessed: 2019-11-21.

[74] James E Braun. Load control using building thermal mass. *J. Sol. Energy Eng.*, 125(3):292–301, 2003.

[75] P Centolella, M Farber-DeAnda, LA Greening, and T Kim. Estimates of the value of uninterrupted service for the mid-west independent system operator (miso). *Research Paper,*

Harvard Electricity Policy Group, Harvard Kennedy School of Government, Cambridge, Massachusetts, USA, 2010.

[76] Jianxiao Wang, Haiwang Zhong, Ziming Ma, Qing Xia, and Chongqing Kang. Review and prospect of integrated demand response in the multi-energy system. *Applied Energy*, 202:772 – 782, 2017.

[77] Albert Molderink, Vincent Bakker, Maurice GC Bosman, Johann L Hurink, and Gerard JM Smit. Management and control of domestic smart grid technology. *IEEE transactions on Smart Grid*, 1(2):109–119, 2010.

[78] Liang Yu, Tao Jiang, and Yulong Zou. Online energy management for a sustainable smart home with an hvac load and random occupancy. *IEEE Transactions on Smart Grid*, 10(2):1646–1659, 2017.

[79] Yudong Ma, Francesco Borrelli, Brandon Hencey, Brian Coffey, Sorin Bengea, and Philip Haves. Model predictive control for the operation of building cooling systems. *IEEE Transactions on control systems technology*, 20(3):796–803, 2011.

[80] Ján Drgoňa, Damien Picard, and Lieve Helsen. Cloud-based implementation of white-box model predictive control for a geotabs office building: A field test demonstration. *Journal of Process Control*, 88:63 – 77, 2020.

[81] Richard S Sutton and Andrew G Barto. *Reinforcement learning: An introduction*. MIT press, 2018.

[82] Simeng Liu and Gregor P Henze. Evaluation of reinforcement learning for optimal control of building active and passive thermal storage inventory. *Journal of solar energy engineering*, 129(2):215–225, 2007.

[83] Daniel O'Neill, Marco Levorato, Andrea Goldsmith, and Urbashi Mitra. Residential demand response using reinforcement learning. In *2010 First IEEE International Conference on Smart Grid Communications*, pages 409–414. IEEE, 2010.

[84] Shaghayegh Yousefi, Mohsen Parsa Moghaddam, and Vahid Johari Majd. Optimal real time pricing in an agent-based retail market using a comprehensive demand response model. *Energy*, 36(9):5716–5727, 2011.

[85] A. Chiş, J. Lundén, and V. Koivunen. Reinforcement learning-based plug-in electric vehicle charging with forecasted price. *IEEE Transactions on Vehicular Technology*, 66(5):3674–3684, May 2017.

[86] Lei Yang, Zoltan Nagy, Philippe Goffin, and Arno Schlueter. Reinforcement learning for optimal control of low exergy buildings. *Applied Energy*, 156:577 – 586, 2015.

[87] R. Lu, S. H. Hong, and M. Yu. Demand response for home energy management using reinforcement learning and artificial neural network. *IEEE Transactions on Smart Grid*, 10(6):6629–6639, Nov 2019.

[88] F. Ruelens, B. J. Claessens, S. Quaiyum, B. De Schutter, R. Babuška, and R. Belmans. Reinforcement learning applied to an electric water heater: From theory to practice. *IEEE Transactions on Smart Grid*, 9(4):3792–3800, July 2018.

[89] F. Ruelens, B. J. Claessens, S. Vandael, B. De Schutter, R. Babuška, and R. Belmans. Residential demand response of thermostatically controlled loads using batch reinforcement learning. *IEEE Transactions on Smart Grid*, 8(5):2149–2159, Sep. 2017.

[90] David Holmberg, Martin Burns, Steven Bushby, Avi Gopstein, Tom McDermott, Yingying Tang, Qiuhua Huang, Annabelle Pratt, Mark Ruth, Fei Ding, et al. NIST transactive energy modeling and simulation challenge phase ii final report. *NIST Special Publication*, 1900:603, 2019.

[91] José R Vázquez-Canteli and Zoltán Nagy. Reinforcement learning for demand response: A review of algorithms and modeling techniques. *Applied energy*, 235:1072–1089, 2019.

[92] Gavin A Rummery and Mahesan Niranjan. *On-line Q-learning using connectionist systems*, volume 37. University of Cambridge, Department of Engineering Cambridge, England, 1994.

[93] Tuomas Haarnoja, Aurick Zhou, Pieter Abbeel, and Sergey Levine. Soft actor-critic: Off-policy maximum entropy deep reinforcement learning with a stochastic actor. *arXiv preprint arXiv:1801.01290*, 2018.

[94] Scott Fujimoto, Herke Van Hoof, and David Meger. Addressing function approximation error in actor-critic methods. *arXiv preprint arXiv:1802.09477*, 2018.

[95] Petr Lánskỳ and Laura Sacerdote. The ornstein–uhlenbeck neuronal model with signal-dependent noise. *Physics Letters A*, 285(3-4):132–140, 2001.

[96] David Silver, Guy Lever, Nicolas Heess, Thomas Degris, Daan Wierstra, and Martin Riedmiller. Deterministic policy gradient algorithms. 2014.

[97] Amir-Hamed Mohsenian-Rad and Alberto Leon-Garcia. Optimal residential load control with price prediction in real-time electricity pricing environments. *IEEE transactions on Smart Grid*, 1(2):120–133, 2010.

[98] Xing Yan and Nurul A Chowdhury. Mid-term electricity market clearing price forecasting: A multiple svm approach. *International Journal of Electrical Power & Energy Systems*, 58:206–214, 2014.

[99] Renzhi Lu, Seung Ho Hong, and Mengmeng Yu. Demand response for home energy management using reinforcement learning and artificial neural network. *IEEE Transactions on Smart Grid*, 10(6):6629–6639, 2019.

[100] Stephen Wilcox and William Marion. *Users manual for TMY3 data sets*. National Renewable Energy Laboratory Golden, CO, 2008.

[101] Adam Paszke, Sam Gross, Soumith Chintala, Gregory Chanan, Edward Yang, Zachary DeVito, Zeming Lin, Alban Desmaison, Luca Antiga, and Adam Lerer. Automatic differentiation in pytorch. 2017.

[102] Rafał Weron. Electricity price forecasting: A review of the state-of-the-art with a look into the future. *International journal of forecasting*, 30(4):1030–1081, 2014.

[103] Roman V Belavkin. Conservation law of utility and equilibria in non-zero sum games. *arXiv preprint arXiv:1010.2439*, 2010.

[104] B.M. Mukhamediev. The solution of bilinear programming problems and finding the equilibrium situations in bimatrix games. *USSR Computational Mathematics and Mathematical Physics*, 18(2):60 – 66, 1978.

[105] Deng-Feng Li and Jie Yang. A difference-index based ranking bilinear programming approach to solving bimatrix games with payoffs of trapezoidal intuitionistic fuzzy numbers. *Journal of Applied Mathematics*, 2013:10, 2013.

[106] David Avis, Gabriel D. Rosenberg, Rahul Savani, and Bernhard von Stengel. Enumeration of nash equilibria for two-player games. *Economic Theory*, 42(1):9–37, Jan 2010.