

Program Final Report
PFR-10297

Network Optimized Distributed Energy Systems (NODES)

S. Glavaski
M. Garcia-Sanz
M. Ilic
R. Jaddivada
T. Santora

6 January 2021

Lincoln Laboratory
MASSACHUSETTS INSTITUTE OF TECHNOLOGY
LEXINGTON, MASSACHUSETTS



DISTRIBUTION STATEMENT A. Approved for public release. Distribution is unlimited.

This material is based upon work supported by the Department of Energy under Air Force Contract No. FA870215-D-0001.

This report is the result of studies performed at Lincoln Laboratory, a federally funded research and development center operated by Massachusetts Institute of Technology. This material is based upon work supported by the Department of Energy under Air Force Contract No. FA8702-15-D-0001. Any opinions, findings, conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the Department of Energy.

© 2021 Massachusetts Institute of Technology

Delivered to the U.S. Government with Unlimited Rights, as defined in DFARS Part 252.227-7013 or 7014 (Feb 2014). Notwithstanding any copyright notice, U.S. Government rights in this work are defined by DFARS 252.227-7013 or DFARS 252.227-7014 as detailed above. Use of this work other than as specifically authorized by the U.S. Government may violate any copyrights that exist in this work.

Massachusetts Institute of Technology
Lincoln Laboratory

Network Optimized Distributed Energy Systems (NODES)

S. Glavaski
Pacific Northwest National Laboratory

M. Garcia-Sanz
Case Western Reserve University (CWRU)

M. Ilic
T. Santora
Group 47

R. Jaddivada
MIT

Project Report PFR-10297
6 January 2021

DISTRIBUTION STATEMENT A. Approved for public release. Distribution is unlimited.

This material is based upon work supported by the Department of Energy under Air Force Contract No. FA8702-15-D-0001.

Lexington

Massachusetts



Final Scientific/Technical Report
Network Optimized Distributed Energy Systems (NODES)

ARPA-E Award Number: **DE-AR0000703**



Award:	DE-AR0000703
Lead Recipient:	MIT-Lincoln Laboratories
Project Title:	Synthetic Cloud-Based Regulation Reserve Distribution Management System (SECURED)
Program Director:	Dr. Sonja Glavaski, Prof. Mario Garcia-Sanz
Principal Investigator:	Prof. Marija Ilic
Research Assistant:	Dr. Rupamathi Jaddivada
Contract Administrator:	Tammy Santora
Date of Report:	January 6, 2021
Reporting Period:	September 1, 2019 – August 31, 2020

DISTRIBUTION STATEMENT A. Approved for public release. Distribution is unlimited.

This material is based upon work supported by the Department of Energy under Air Force Contract No. FA8702-15-D-0001. Any opinions, findings, conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the Department of Energy .

© 2021 Massachusetts Institute of Technology.

Delivered to the U.S. Government with Unlimited Rights, as defined in DFARS Part 252.227-7013 or 7014 (Feb 2014). Notwithstanding any copyright notice, U.S. Government rights in this work are defined by DFARS 252.227-7013 or DFARS 252.227-7014 as detailed above. Use of this work other than as specifically authorized by the U.S. Government may violate any copyrights that exist in this work.

Contents

4.1.	Introduction	13
4.2.	General Control design.....	13
4.2.1.	Information exchange at tertiary control timescales	14
4.2.2.	Information exchange at secondary control timescales.....	16
4.2.3.	Device-level control implementation	18
4.3.	Control hardware design implementation.....	20
4.4.	Multi-layered control implementation in the devices:.....	22
4.4.1.	Primary control implementation.....	22
4.4.1.1.	Electric Vehicle Control	22
4.4.1.2.	HVAC Control.....	26
4.4.2.	Secondary control design	29
4.4.3.	Tertiary control:.....	34
4.5.	Feeder-level simulation validation results:.....	35
4.5.1.	Simulation Setup	35
4.5.2.	Simulation Result	37
4.6.	References	43
4.7.	Technology Transfer Activities	44
4.8.	Follow-On Funding	46
4.9.	Appendices	48
4.9.1.	APPENDIX A: MILESTONE 2.1 DOCUMENTATION - APIs development completed.....	48
4.9.2.	APPENDIX B: MILESTONE 2.2 DOCUMENTATION – Data import platform enabled 53	
4.9.3.	APPENDIX C: MILESTONE 2.3 DOCUMENTATION - Developed device testing framework.....	55
4.9.4.	APPENDIX D: MILESTONE 2.3 DOCUMENTATION -CONTD.	58
4.9.5.	APPENDIX E: MILESTONE 4.1 DOCUMENTATION – Completed hardware construction.....	61
4.9.6.	APPENDIX F: MILESTONE 4.2 DOCUMENTATION - Completed hardware installation.....	68

Figures

Figure 1: Actual inflexible demand shown in black and its predictions over multiple rates that drive the decision making in interactive control.....	14
Figure 2: Interactive information exchange at the higher layer of the proposed approach over tertiary control or market clearing timescales	15
Figure 3: Information exchange between agents at secondary control timescales.....	17
Figure 4: Multi-layered control from the perspective of a DER	18
Figure 5: Primary control design in energy space	19
Figure 6: The multi-layered control of EV- Dark Blue, green and red arrows represent information communicated at T_p, T_s, T_t rate respectively. Light blue arrows represent information specified by the users which is either fixed or changes at much slower timescales. Furthermore, dotted lines represent ahead-of-time interactive information exchange.	23
Figure 7: Output of interest in energy space yz reaching the reference value $yzref[n]$ sent by secondary layer control.....	25
Figure 8: Electrical power consumption of EV	25
Figure 9: Current reference signals fed as control inputs for implementation through PSI API.	26
Figure 10: The multi-layered control of HVAC- Dark Blue, green and red arrows represent information communicated at T_p, T_s, T_t rate respectively. Light blue arrows represent information specified by the users which is either fixed or changes at much slower timescales. Furthermore, dotted lines represent ahead-of-time interactive information exchange.....	27
Figure 11: HVAC primary control tracking output in energy space yz , to the value yz provided by the secondary layer control.....	28
Figure 12: Averaged power consumption of HVAC with and without demand response	28
Figure 13: Evolution of temperature with the proposed energy-based control	29
Figure 14: Proposed switching logic for the HVAC	29
Figure 15: Temperature trajectories with the combined action of primary and secondary controllers.	31
Figure 16: Reference tracking with the MPC-based secondary control actions with $\mu_{reg} = 100\$/kWh$ and $\mu_e = 10\$/KWh$	31
Figure 17: Reference tracking without MPC $\mu_{reg} = 100\$/kWh$ and $\mu_e = 10\$/KWh$	32
Figure 18: Reference tracking with the MPC-based secondary control actions with $\mu_{reg} = 25\$/kWh$ and $\mu_e = 10\$/KWh$	32
Figure 19: Reference tracking with the MPC-based secondary control actions with $\mu_{reg} = 25\$/kWh$ and $\mu_e = 10\$/KWh$	33
Figure 20: Temperature profile for when control actions are obtained with lower reserve penalty cost of $\mu_{reg} = 25\$/kWh$	33
Figure 21: Real time bids for energy with HVAC load.....	34
Figure 22: Real time bids for regulation reserves with HVAC load	34
Figure 23: Day1 feeder-level inflexible demand and solar radiations in blue and orange respectively	36
Figure 24: Day2 feeder-level inflexible demand and solar radiations in blue and orange respectively	36
Figure 25: (Test 2) Reserve request outcomes for RegUp signal of 3.5665 kW.....	39
Figure 26: (Test 2) Energy and reserve capacity market outcomes while tracking RMT = 3.5665 kW	39
Figure 27: (Test 2) Closed-loop response of individual devices while tracking RMT = 3.5665 kW	40

Figure 28: (Test 6) Reserve request outcomes for RegDown signal of 1.684 kW	40
Figure 29: (Test 6) Energy and reserve capacity market outcomes while tracking RMT = -1.684 kW	41
Figure 30: (Test 6) Closed-loop response of individual devices while tracking RMT = -1.684 kW	41

List of Acronyms

PSI	Pecan Street Inc.
DER	Distributed Energy Resource
DR	Demand Response
EV	Electric Vehicle
HIL	Hardware in the Loop
HVAC	Heating, Ventilation, and Air Conditioning
IoT	Internet of Things
LLNL	Lawrence Livermore National Laboratory
MIT-LL	Massachusetts Institute of Technology – Lincoln Laboratory
NODES	Network Optimized Distributed Energy Systems
SOP	Statement of Project Objectives

1. Executive Summary

Several US states have adopted Renewable Portfolio standards, leading to large-scale penetration of solar and wind energy. As a result, there is a growing reliance on ancillary services due to the high uncertainty in net load predictions. A major operational challenge for traditional generation units participating in such ancillary services is that they have significant inertia leading to increased wear and tear. Furthermore, they are generally too slow in tracking net load decrements if operating at their economic minimum. To overcome this problem, the utilities have identified the potential of groups of small capacity controllable demand technologies as critical ancillary services providers. However, most of the existing demand response schemes and state-of-the-art methods fail to enable controllable demand units in providing ancillary services. Two major challenges are:

- Lack of provability: For acceptance as a reliable ancillary service, the aggregate of devices needs to provably adjust its consumption, even in the worst case when most devices have opted out.
- Lack of price incentives: The technology further needs to be supported by the economic signals to be viable and be adopted in practice.

The overall NODES program was motivated by the need to begin to overcome these challenges. In this report, we summarize the work done by the MIT Lincoln Laboratory (LL) project under this program. The work started as an MIT sub-contract under the Eaton project, and continued as a stand-alone MIT LL project. The objective was to develop and validate a cloud-computing solution for providing provable and robust synthetic regulation reserve services. The concept was to provide regulation service as defined in Category II of NODES FOA at an aggregate level by coordinating thousands of residential devices, including electric vehicles, water heaters, heaters and air conditioners. The proposed solution builds on the Dynamic Monitoring and Decision Systems (DyMonDS) framework initially conceptualized and demonstrated for adaptive load management (ALM) by the Principal Investigator and her collaborators, Jhi-Yong Joo and Jonathan Donadee, in particular, for energy markets in the bulk power systems. This framework suggests the distribution of a large portion of computationally intensive tasks performed simultaneously by the devices themselves, with minimal communication from and to the aggregator in terms of quantities and energy/reserve bids, respectively. This method reduces the computation and communication burden by orders of magnitude compared to the state of the art methods. Furthermore, the devices supplement their internal automation with a nonlinear control to ensure provable reserve delivery. Adopting such technology is contingent upon the device's willingness to replace its digital control logic and deploy additional sensors.

In the last fiscal year of 2019-2020, MIT has partnered with Pecan Street Inc. (PSI) to develop and validate a simple-to-implement device control with minimal need for sensor measurements. This part of the project with PSI involves the same DyMonDS framework as was pursued in the project with Eaton for ensuring transparency of price signals. For assuring reserve implementation by the devices, MIT LL project developed a device-level multi-layered energy-based control. It comprises a novel robust energy-based sliding mode control in the lower layer, which interacts with a higher layer energy-based model predictive control with linear constraints. The higher layer control performs trajectory planning by creating an envelope that the fast lower layer must follow through a computationally tractable linear programming. The sliding mode implementation in the lower layer is performed in energy space, thus requiring minimal energy and power measurements, robust to model and parameter uncertainties, while also provably implementing the regulation reserves.

The proposed control design in the lower layer further accounts for implementation aspects, including manufacturer-imposed constraints such as those on switching, service-provider-imposed

regulations to restrict the duration of demand response events. Finally, the proposed multi-layered control duly accommodates ARPA-E FOA performance metrics in every device's control design. As a result, even a single device can provably implement the regulation reserves that it has committed ahead of time.

MIT has evaluated the proposed energy-based multi-layered control design through high-fidelity simulations. It has leveraged the PSI-developed API to access the lab devices for control validation. Final hardware testbed demonstrations; however, halted due to COIVD restrictions. Nevertheless, the MIT-LL cloud computing platform-based simulations based on the proposed technology demonstrated regulation reserves' implementation for diverse scenarios while still meeting the ARPA-E FOA performance metrics.

2. Acknowledgements

This work is based upon work supported by the Department of Energy under Air Force Contract No. FA8702-15-D-0001. The MIT team would like to thank ARPA-E for their feedback at different project phases and their generous financial support. The MIT team would also like to thank the PSI team for all the help with data and hardware access provided during the project.

3. Project Goals and Accomplishments

This award allowed the MIT LL team in collaboration with PSI team to demonstrate a number of key objectives. The focus of the project was on building an IoT based demand response control solution. A number of tasks and milestones were laid out in Attachment 3, the Technical Milestones and Deliverables, at the beginning of the project. The actual performance against the stated milestones is summarized here:

No.	Milestones/Objectives	Accomplishment Summary
1.0	Control development and simulation validation	
1.1.	HVAC control validated through simulation	A robust sliding mode control of residential HVAC is proposed to overcome the difficulties pertaining to these devices' controllability and observability. The control objectives, including stability, reserve signal tracking, meeting the constraints on the temperature from a consumer standpoint, ON/OFF cycles from a manufacturer perspective, and the number of demand response events from the service provider standpoint, were all met. In addition, the ARPA-E performance metrics on response time, availability, duration were all met provably. The proposed control design method embedded in an HVAC unit was simulated under various test scenarios to validate further the performance benefits. The design method was provided in the Q2 report, which was further converted to a sliding mode control implementation for having robustness to the model and parameter uncertainties. The relevant theoretical details can be found in [1]. The proposed control design is summarized in this report.
1.2.	EV control validated through simulation	A robust sliding mode control of an EV is proposed to overcome the difficulties pertaining to the lack of battery models and limited observability of these devices. All the control objectives, such as those described in M1.1, are achieved with smooth control. The proposed control design method embedded in an EV was simulated under various test scenarios to validate the performance benefits further. The theory behind the proposed design can be found in [1]. The summary of the proposed control design is summarized briefly in this report.

1.3.	Feeder control validated through simulation	Separate centralized cloud control was developed for the DyMonDS framework. A simulation study involving both control layers for various types of flexible loads was conducted, and the results showed that the performance metrics are all met.
2.0	Lab testing environment development	
2.1.	APIs development task completed	PSI developed an API to allow the MIT LL cloud-computing resource to access in-home local 1-second data, as well as operational control/status data of the WiFi relay (hereto referred to as "BluFollower") and EVSE. 1-second level eGauge data is updated no more than every 10 seconds. Each update has the previous 10 seconds of data at a 1-second interval. Available data include whole home usage (grid), HVAC fan/blower usage, EV usage (if the home has a vehicle), solar generation, vehicle present T/F, vehicle charging, vehicle presence, and temperature data. The cloud server is provided for aggregation data (data from other homes in the testbed). The cloud server allows direct read-only data access (no API) for aggregation algorithm development. Data types are the same as local data but available at a 1-minute interval. Details have been provided in Q2 report.
2.2.	Data import platform enabled	PSI set up the GitHub code repository for pushing code to fielded units. The GitHub repository has been created and populated. The MIT team supplied their GitHub accounts to be added as contributors to the repo to host their controller code that runs on the Raspberry Pis. This repo can be found at: https://github.com/Pecan-Street/MIT_Controller Details have been provided in Q2 report.
2.3.	Developed device testing framework	Photographic evidence of test sets, data verifying simple on/off control through API from PSI cloud resources are provided in Appendix C. Available MQTT Topics and Commands for load control testing is provided in Appendix D.
3.0	HIL Simulation Validation	

3.1.	HIL simulation validation of HVAC devices accomplished	In the revised field test plan, the hardware platform now becomes the three devices in the PSI lab facility. To better understand this test environment, 1second sampled data was downloaded from PSI's database and scrutinized. This data was used to tune the parameters of the thermal models for better representation of HVAC in the lab.
3.2.	HIL simulation validation of an EV devices accomplished	For the NODES project, the probability distribution function of driver time and battery state of charge was used to generate the random simulation data. Details of the work were submitted to ARPA-E in the 2017 Q2 report.
4.0	Hardware Deployment	
4.1	Completed hardware construction	Photographic Evidence of 25 sets of WiFi Relays, Single Board Computers, and 5 sets of EV hardware are provided in Appendix E. Copies of final quality control (QC) documentation for all hardware are also provided in Appendix E.
4.2.	Completed hardware installation	Statistical information showing data reporting for 25 single-board computers with connections to control devices is provided in Appendix F and photographic evidence of the setup. Pecan Street installed EVSE controllers in the lab and a residential setting in Austin, TX, and one HVAC controller in the lab. The remaining 3 EVSE controllers and 24 HVAC controllers were installed at the PSI lab to validate request signals, using data from PSI's participant network. These controllers were used for communication testing and not to control loads.
5.0	Hardware validation and final report preparation	
5.1.	Completed the experiment with 3 devices	PSI assisted with the development of the software platform and in-home hardware to deploy MIT LL's proprietary technology. Although the team had planned to field test the system on a subset of 25 homes, due to COVID-19 restrictions, the team pivoted to a lab deployment with two EVSE loads, one HVAC load, and 27 devices connected without a load.

5.2	Completed the final report	A document that summarized the final outcomes of the project has been prepared and submitted to ARPA-E
-----	----------------------------	--

4. Project Activities, Technical Details

4.1. Introduction

This project is a continuation of the previous effort with Eaton toward utilizing the flexibility provided by tens of thousands of residential loads in providing fast regulation reserves at an aggregate level. The proposed approach involved a fast feedback control implementation at the device level, communicating at much lower rates with the coordinator in terms of the interface power quantities and associated price signals. Not only does the proposed approach drastically reduce the communication and computation needs of the coordinator, but also theoretical provability is provided. Simulation validation was performed to show the scalability and economic benefits that could be obtained. However, the major hurdle to the proposed design was its increased reliance on having high fidelity measurements of internal physical quantities that are seldom available. As a continuation of the project effort, MIT LL partnered with PSI to understand device-specific restrictions in more detail and design a robust lower layer feedback control utilizing minimal measurements.

PSI has provided MIT LL with their database access involving second-level data of 30 houses to help implement their controllers and perform validation through a simple API that is typically utilized with the Internet of Things (IoT). The team jointly developed simple-to-implement Python scripts based on MQTT protocols that can be implemented at the devices based on simple switching logic. The reference signals utilized for construction of sliding manifold as needed by the switching control, are provided by the cloud at relatively lower rates but at much faster rates than the energy market clearing timescales.

However, the final hardware validation could not be conceived due to the ongoing COVID-19 pandemic restricting the access of PSI technical staff into the Mueller community's residences. It was mutually agreed upon with ARPA-E that minimal effort to validate the control would be conceived and have thus far restricted the validation utilizing only three devices in the PSI lab facility. Thus, realistic simulations based on model fit using PSI data were demonstrated, in addition to the higher layer control aligning physical signals with the market signals while also ensuring provable delivery of regulation reserves satisfying the ARPA-E FOA performance metrics for category II of regulation reserves.

4.2. General Control design

The MIT-LL control approach emphasizes the multiple spatial and temporal layers. The proposed information exchange across the layers makes the approach scalable due to reduced computation and communication needs. However, the implementation of reserves provably is ensured through fast feedback control implemented locally at the devices. The interaction between the two algorithms is also duly considered, which ensures alignment of prices and quantities associated with energy and reserves. Figure 1 shows the temporal scales of importance for tertiary control timestep $T_t = 30 \text{ minutes}$, secondary control timestep $T_s = 5 \text{ minutes}$, $T_p = 10 \text{ seconds}$

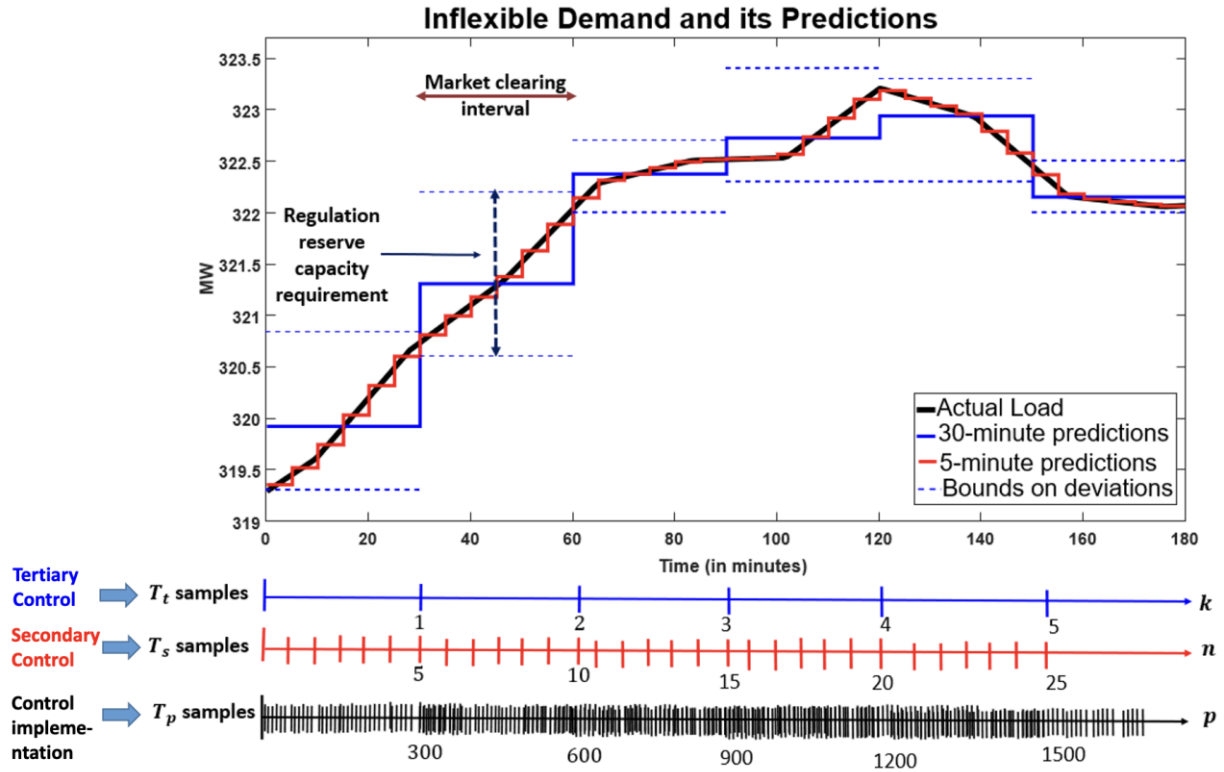


Figure 1: Actual inflexible demand shown in black and its predictions over multiple rates that drive the decision making in interactive control.

In a nutshell, the temporal separation between the algorithms is summarized as follows:

- The energy market clearing happens at much slower timescales referred to with the sampling rate of T_t seconds which typically is 1 hour. At this rate, bid functions for participation in energy and possibly providing regulation reserves are communicated. The information exchange involved in this temporal layer is explained in Section 4.2.1.
- The uncertainties lead to fast demand changes over T_s seconds which typically is of the order of minutes. This leads to the independent system operator (ISO) communicating an AGC signal representative of the fast imbalances in the network. The NODES is expected to disaggregate this aggregate AGC signal into ones that can be provided by the individual houses, also taking into consideration the reserve capacity that has been cleared at T_t rate. The information exchange involved is explained in Section 4.2.2.
- However, the devices ensure the implementation of signals communicated through a two-layered control summarized in Section 4.2.3.

The overview of the general control design approach in different layers is provided in the following sub-sections. However, the details can be referred to in the references [1-3].

4.2.1. Information exchange at tertiary control timescales

In this section, we describe the distributed multi-layered interactive control design architecture briefly to coordinate many small devices. The energy and reserve price-quantities are aligned across the spatial layers. Shown in Figure 2 towards the right of the Mueller community feeder under consideration. The far end of the distribution feeders is identified by the circles that have several households incident. Three of several houses and the information they exchange with the feeder-level coordinator referred to as NODES is shown to the left of Figure 1. The aggregator is

envisioned to interact with the independent system operator (ISO), which for the feeder in Texas is ERCOT.

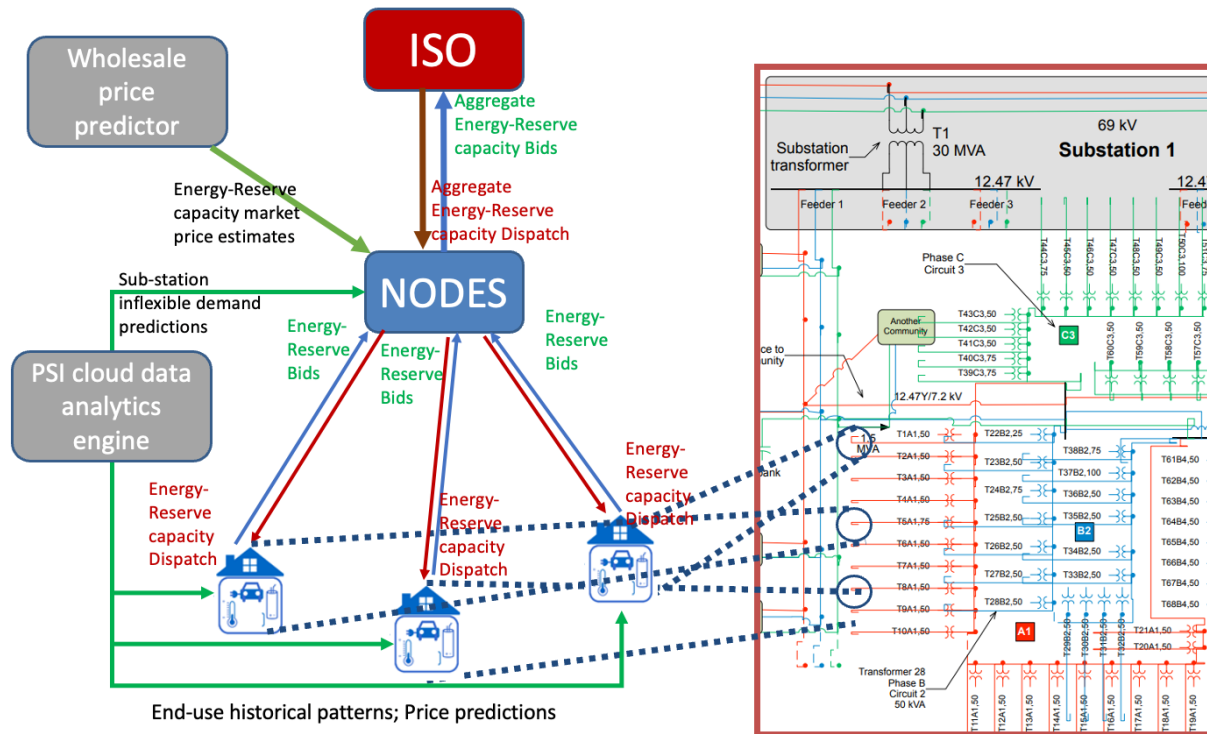


Figure 2: Interactive information exchange at the higher layer of the proposed approach over tertiary control or market clearing timescales

The steps involved in the proposed architecture are as follows:

- A wholesale price predictor based on historical data and current aggregate inflexible demand predicts prices for energy and reserve capacity provision and sends it to NODES.
- The NODES operator forwards these prices to the houses it is aggregating, or optionally an external data analytics engine can be used to keep track of historical prices to send out individual estimated prices to the houses.
- The houses, based on their local predicted end-use patterns, price predictions, and the quality of service specification of the consumers, such as those of temperature in HVAC and state of charge requirements of the EVs, are taken into consideration to compute a bid function for energy and reserve capacity. The bids comprise the information of price Vs. quantity of energy and reserve capacity curve, in addition to the respective operating-conditions-dependent limits
- The bid functions provided by the households are then aggregated by NODES by also accounting for uncontrollable inflexible feeder-level demand predictions, to submit an aggregate bid for energy and reserve capacity in the wholesale market.
- The ISO receives similar bids from numerous feeders and the conventional generators and then clears the energy and reserve quantities that are further sent to the NODES. The corresponding prices are sent to the price forecaster at the wholesale level
- The NODES performs a similar operation as that of ISO to clear the bids submitted by households and sends over the quantities to the houses and prices to the data analytics engine.

From the house's standpoint, the proposed framework requires the computation of bid functions. These functions abstract the price-willingness to pay or sell by accounting for all internal constraints. It thereby comprises two important quantities that need to be computed.

- Computation of price elasticity (Bottom-up): The houses compute these quantities by considering the provable energy and reserve quantities that the devices can deliver through the embedded automation over faster timescales while also optimizing the energy arbitrage and revenues obtained through reserve capacity provisioning. The details on these sensitivity-based bids referred to as DyMonDS bids can be referred to in [4,5].
- Computation of limits (Bottom-up): The limits on the reserves and that on physical capacity limits that can be implemented by accounting for quasi-static provably implementable droop relations by the embedded automation

The method in which the time-varying bid functions and the limits are generated for EVs and HVACs are explained briefly in Sections 4.4.1. and 4.4.2. respectively.

The two important algorithms involved in this temporal layer at the NODES level are:

- Aggregate energy and reserve bid functions (Bottom-up): The bids and the limits submitted by the houses are all aggregated to compute one single bid corresponding to the point of interconnection of the feeder with the grid.
- Disaggregation of individual house-level energy and reserve capacity signals (Top-down): These quantities are computed after-the-fact and split the energy and reserve capacity signals sent out by the ISO to the individual quantities that need to be implemented by houses.

These details have been presented in references [6,7].

4.2.2. Information exchange at secondary control timescales

To reduce the communication overhead at faster rates, we propose a uni-directional information exchange top-down from ISO to NODES, while preserving multi-directional communication links between the house-level controllers and the devices. This information exchange is shown in Fig. 3.

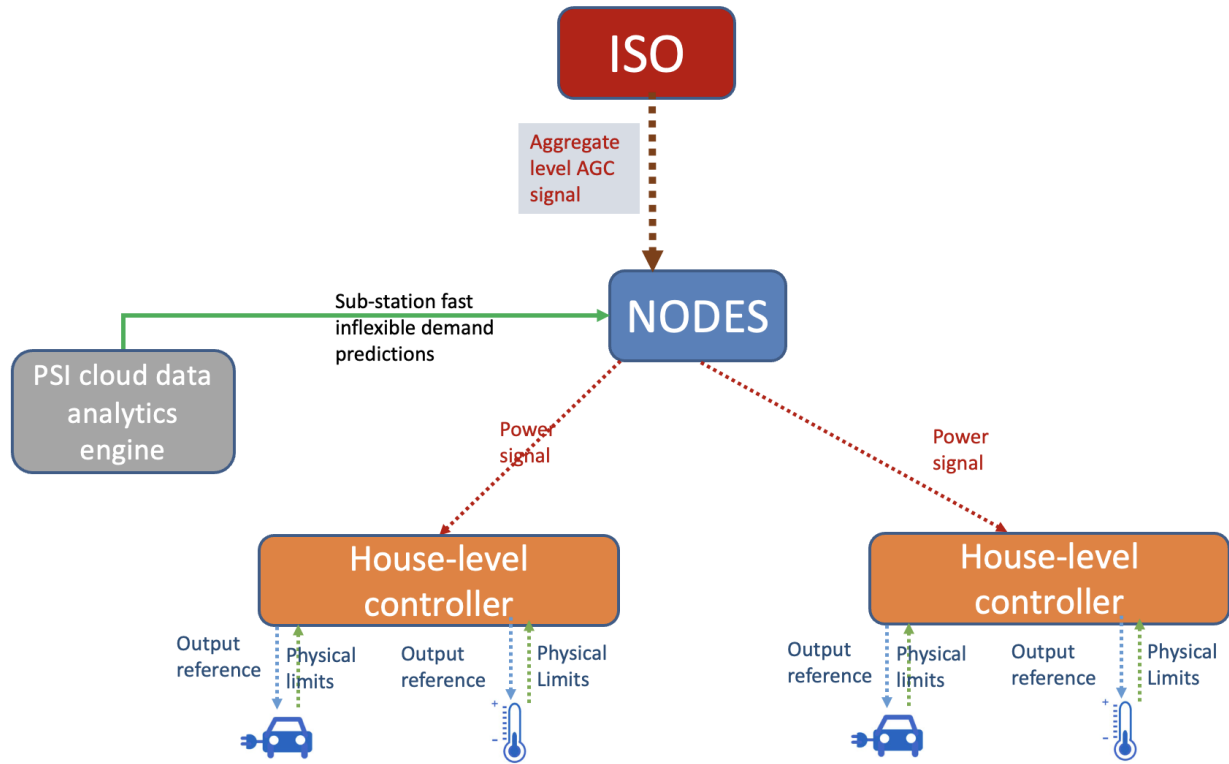


Figure 3: Information exchange between agents at secondary control timescales

The critical computation algorithms at this layer are:

- **NODES-level (top-down):** Disaggregation of the aggregate AGC signal provided by ISO into the individual signals that are to be tracked by the houses. However, this decision-making accounts for the reserve capacity dispatched in the tertiary layer, thereby guaranteeing the reserves' implementation.
- **House-level (Top-down):** The disaggregation of the AGC signal and the energy signal into the output setpoint of individual devices by accounting for the provably implementable droop relations through the devices' embedded automation.
- **Device-level (Bottom-up):** The droop coefficients, physical limits, and the measurement of averaged output variables in energy space over this timescale for use in the house-level controller.

Notice that this step involving fast information exchange only one-way between the aggregator and the houses. Each device, on the other hand, communicates with respective house-level controllers through bi-directional links. The house-level controllers can also directly accommodate device level automation altogether, avoiding the communication overhead within the households.

The setpoints computed by the house-level secondary controller to the devices are also based on the device-specific limitation that the house-level controller is cognizant of. This ensures that the device following the output setpoints does not reach control saturation over controller implementation timescales.

The details of the algorithm embedded in the devices is explained in [2,3] which is heavily dependent on the embedded automation. The implemented control in the devices for the purposed of this project is explained in detail in the next sub-section.

4.2.3. Device-level control implementation

The overall control architecture, as seen by the device, is shown in Fig. 4. In Figure 4, the primary control samples are represented by $[p]$, typically evolving at the rate of seconds. The secondary house level control signals are exchanged at the rates of minutes, identified by the sample number $[n]$. Finally, the tertiary control signals are exchanged and implemented at the rate of market-clearing timescales, typically of order 1 hour, identified by the same number $[k]$.

Figure 4 shows that the block labeled as primary control is the embedded automation that already exists on the devices. In the residential units, there is not enough flexibility to change the digital control logic embedded. Thus the internally computed signals can only be blocked.

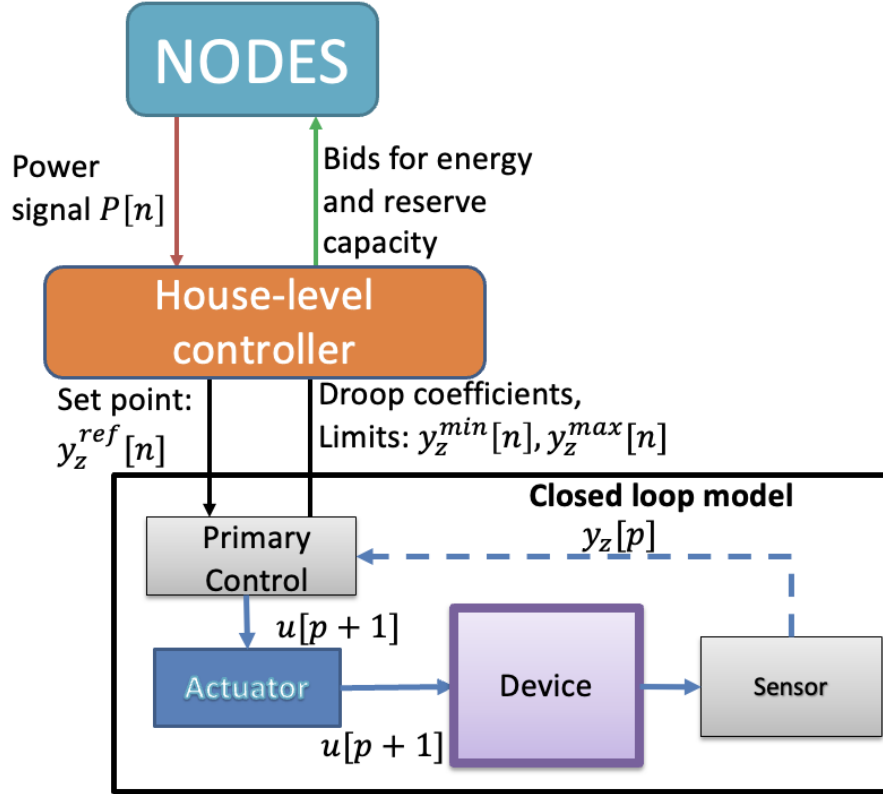


Figure 4: Multi-layered control from the perspective of a DER

We propose to utilize the dynamical model in energy space that sufficiently captures the energy/power interactions of the device with the rest of the system to control the switching cycles of the actuators [8-10]:

$$\dot{E} = -\frac{E}{\tau} + P = p \quad (1a)$$

$$\dot{p} = 4E_t - \dot{Q} \quad (1b)$$

Here, E, p, E_t respectively are the stored energy, its first derivative, and stored energy in tangent space. The quantities $P, \dot{Q}$ respectively are the instantaneous power and the reactive power quantities that can be measured at the interface. Notably, the usage of simple enough models leads to minimal sensor measurement requirements. It is shown in what follows that only energy and power variables are required for the control design implementation.

If $\dot{Q}$ is chosen as the virtual controllable input upon selecting the output variable of interest

$$y_z = \frac{E^{ref}}{\tau} - P \quad (2)$$

This quantity represents the device's overall energy imbalance when the internal quality of service (QoS) specifications are perfectly met by the primary controllers. In an ideal case, we require this quantity to be equal to zero. However, fast rate adjustments to non-zero $y_z^{ref}[n]$ are needed when reserves need to be provided by devices with minimal consequences on the comfort, abstracted through the quantity $\frac{E^{ref}}{\tau}$.

It was shown in [2,3] that provable tracking of y_z to y_z^{ref} can be achieved by virtual control. The virtual control mapping to the physical model related to the actuator can be achieved through a diffeomorphic mapping between the conventional space and energy space variables at the interfaces. The proposed energy-based control that needs to be implemented in the primary control block in Figure 4 is summarized in Figure 5. References on this approach in the context of conventional generation and power electronics control can be found in [11,12].

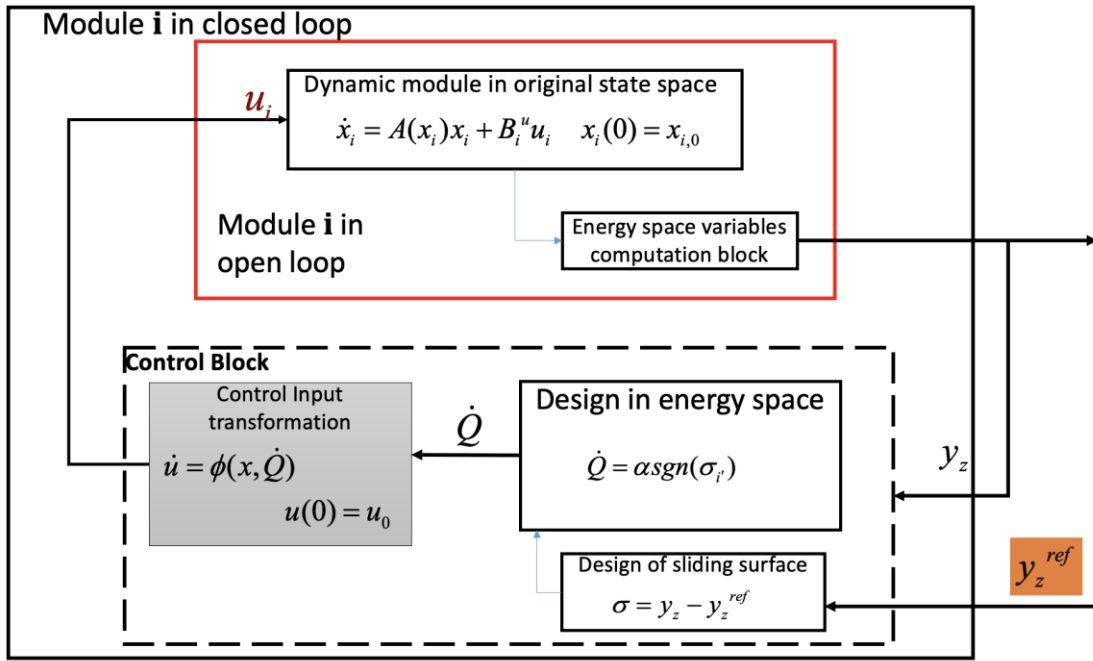


Figure 5: Primary control design in energy space

When y_z reaches y_z^{ref} , the stored energy also reaches the desired value, thus ensuring the internal variables pertaining to the quality of service specifications are also achieved under some minor assumption on the observability of the physical output variables of interest, such as temperature in the HVAC unit.

We further propose to utilize a sliding mode control implementation of the virtual control. This leads to provable tracking in finite time that can also be user-specified. Note that the linear sliding mode control in energy space translates to a nonlinear higher-order sliding mode control in conventional physical variables, leading to better tracking. We select the sliding mode control gain α such the reaching time of sliding mode is shorter than the ARPA-E performance metrics on rising time.

Furthermore, the sliding mode control threshold is chosen to ensure switching is not too frequent (not shown in the figure). The simple enough energy space models are utilized to derive static incremental relations upon reaching the sliding mode in terms of the output variables of interest in energy space, the setpoints, and the system's disturbances. This three-way relation is referred to as droop relation, analogous to the one utilized in generators. This relation's linear nature is accounted for in the secondary layer control design to compute provably achievable output setpoint values. In addition, the number of switchings permissible can be accommodated for in the secondary control in a model predictive and tractable way because of the linearity in the quasi-static droop relations made possible by the device-level feedback control.

4.3. Control hardware design implementation

The implementation of controllers at different layers as seen by the device shown in Figure 5 is summarized below:

- A raspberry PI was embedded within each of the devices where device-specific computations can be included. The computation involved made use of the sensor subscriptions at implementation timescales (of the order of seconds) through MQTT protocols.
- The fine time-granularity measurements needed for the EV and HVAC's primary control for the proposed energy-based sliding mode control and fast actuation to be performed made possible through additional control hardware and load testing hardware constructed and installed by PSI.
 - eGauges: They measure all the electrical variables, while only the real power measurements of the devices are utilized
 - SAE – J1772 interface of the EV already comes with a sensor for measuring the current charging rates, which can be accessed through its API
 - Temperature sensor: PSI has equipped a minimal cost temperature sensor within the units to be tested.
- The house-level controllers for secondary and tertiary control are implemented within the MIT-LL supercloud computing facility, which communicates to the device-level controllers through a dedicated communication link.
- The NODES and ISO modules are further simulated using dedicated processors within the MIT-LL supercloud computing facility as well.

In order to support the implementation of all the layers of the control, PSI provided support with their API/cloud services development, device testing framework and finally, the hardware construction and installation for the control algorithms to be implemented and tested. The details on these services are summarized next.

API Development/Cloud Services Development: Pecan Street developed an API to allow the LL cloud-computing resource to access in-home local 1 second data, as well as operational control/status data of the WiFi relay (hereto referred as "BluFollower") and EVSE. 1-second level eGauge data is updated no more than every 10 seconds. Each update has the previous 10 seconds of data at 1 second interval. Available data include whole home usage (grid), HVAC fan/blower usage, EV usage (if home has vehicle), solar generation, vehicle present T/F, vehicle charging, vehicle presence, and temperature data. Cloud server is provided for aggregation data (data from other homes in the test bed). The cloud server allows direct read only data access (no API) for aggregation algorithm development. Data types are the same as local data but available at a 1-minute interval.

Data Import: Pecan Street set up the GitHub code repository for pushing code to fielded units. The GitHub repository has been created and populated. The MIT team supplied their GitHub accounts to be added as contributors to the repo to host their controller code that run on the Raspberry Pis. This repo can be found at: https://github.com/Pecan-Street/MIT_Controller

Device testing framework: Photographic Evidence of test sets, data verifying simple on/off control through API from PSI cloud resources are provided in Appendix C. Available MQTT Topics and Commands for load control testing is provided in Appendix D.

Pecan Street provided a test set at the PSI lab with eGauge, electric vehicle, HVAC thermostat controller and all project hardware (including single board computer), to test and verify operation of the hardware through commands to be sent by MIT-LL algorithms and cloud computing resources.

- The EVSE controller, constructed and QC'ed by PSI in task 4.1, was installed at the PSI lab, using the lab's Nissan Leaf. The EVSE controller is capable of providing the vehicle charging status (yes/no), presence (yes/no), and to adjust the charge rate. In addition, the eGauge provides visibility into the power consumption of the electric vehicle.
- In the video available in Appendix C, we provide photographic evidence to show the vehicle at the lab responding to a set of commands to adjust the charging rate, and show the control initiate and terminate according to the control signal's variables. The eGauge console shows the charge at baseline, then changes according to the commands issued for the duration specified, then returns back to baseline.
- The HVAC controller, constructed and QC'ed by PSI in task 4.1, was installed at the PSI lab using our lab's HVAC unit. The HVAC controller is capable of turning the unit on/off and to the cool/fan/heat settings. In addition, the eGauge provides visibility into the power consumption of the HVAC compressor.
- In the video available in Appendix C, we provide photographic evidence to show the HVAC compressor responding to a command to turn off for a specific duration, and show the control initiate and terminate according to the control signal's variables. The eGauge console shows the HVAC compressor on, then turns off according to the commands issued for the duration specified, then returns back to on after the specified duration.

Hardware Construction: Photographic Evidence of 25 sets WiFi Relays, Single Board Computers, and 5 sets of EV hardware are provided in Appendix E. Copies of final quality control (QC) documentation for all hardware are also provided in Appendix E. Pecan Street constructed all hardware sets to replace the functions intended for the Eaton LCR. Three pieces of hardware were developed for the project:

- *WiFi addressable relay (BluFollower):* The WiFi addressable relay, can be used for control of HVAC loads by interrupting the thermostat control wiring. It can control on/off/heat/cool/fan HVAC settings. This enables the HVAC systems to be turned off or prevented from operating. Twenty-five (25) relays have been designed, fabricated, assembled, and quality controlled.
- *Single-board computer:* Pecan Street has selected the Raspberry Pi 3 (RPI) as the single board computer running Linux and capable of executing Python programs for algorithmic monitoring of energy data from existing eGauge devices and relay controllers.
- *Opt-out switches:* The modified SOPO approved in June 2020 removes the construction of opt-out switches.
- *EVSE Controller:* Pecan Street has developed five (5) EVSE controllers for the J1772 intercept. These provide vehicle charging/presence status and adjust charge rate. The

EVSE controller board was fabricated, assembled and quality controlled in the lab, and was assembled with the J1772 intercept.

Hardware installation: Statistical information showing data reporting in for 25 single board computers with connections to control devices is provided in Appendix F, as well as photographic evidence of the setup. Pecan Street installed EVSE controllers in the lab and in a residential setting in Austin, TX, and one HVAC controller in the lab. The remaining 3 EVSE controllers and 24 HVAC controllers were installed at the PSI lab to validate request signals, using data from PSI's participant network. These controllers were used for communication testing and not to control loads.

- The EVSE and HVACs are mapped to eGauges of 25 homes in our Austin, TX testbed for simulation purposes. The list of Data IDs is attached in Appendix G, along with the temperature sensor MAC addresses, and historic data of these Data IDs was available on the project server for MIT to access.
- Photographic evidence of the three loads are provided in Appendix F. The EVSE controller and HVAC controller in the lab are referenced above. The EVSE controller in the residential setting is shown in Appendix F. In the video available in Appendix F, we provide photographic evidence to show the vehicle responding to a set of commands to adjust the charging rate, and show the control initiate and terminate according to the control signal's variables. The eGauge console shows the charge at baseline, then changes according to the commands issued for the duration specified, then returns back to baseline.

4.4. Multi-layered control implementation in the devices:

4.4.1. Primary control implementation

4.4.1.1. Electric Vehicle Control

The electric vehicle model was studied in the previous project with Eaton, and it was identified that the state of the charge model sufficiently captures the charging and discharging cycles. In this project, the constraints imposed by the controllability conditions imposed at the device level are taken into consideration. For instance, the internal switching logic can not be changed but can only be indirectly modulated by changing the charge rate setpoints.

The EV's overall control architecture as it is implemented for the project is summarized in Figure 6. While the control design logic has been implemented for use in Raspberry Pi, the testing could not be completed due to COVID-19. MIT-LL has thus instead simulated a reasonable EV physical dynamics block to validate the proposed multi-layered control.

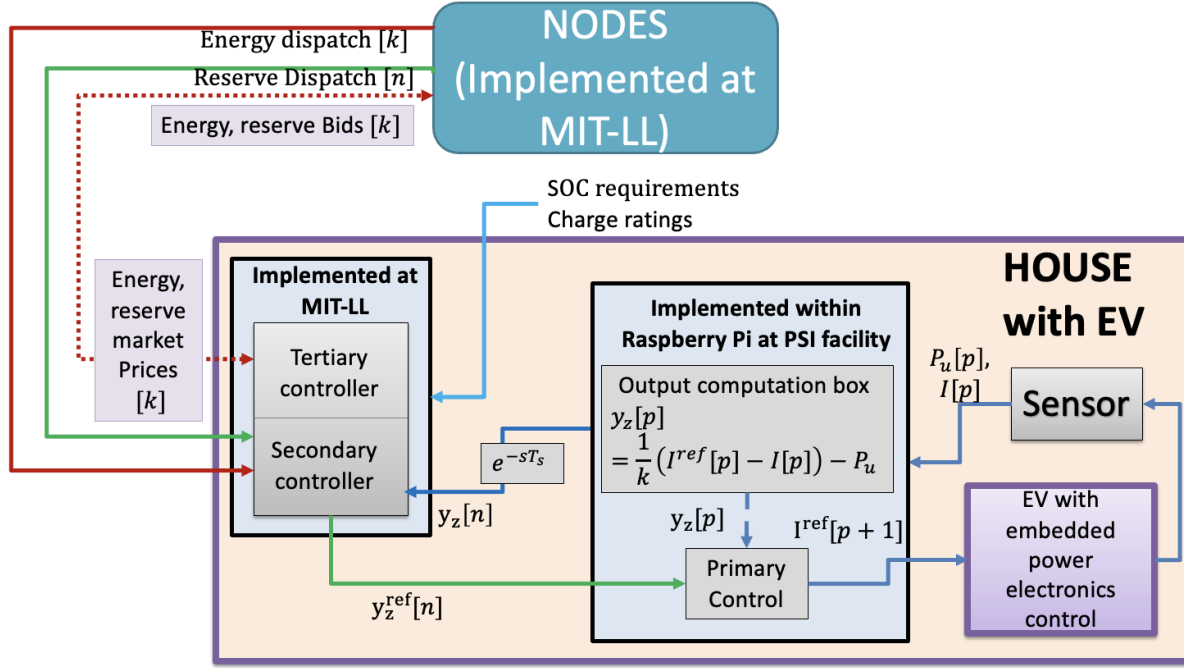


Figure 6: The multi-layered control of EV- Dark Blue, green and red arrows represent information communicated at T_p , T_s , T_t rate respectively. Light blue arrows represent information specified by the users which is either fixed or changes at much slower timescales. Furthermore, dotted lines represent ahead-of-time interactive information exchange.

The primary control design is summarized in Section 4.2.3 through Figure 6 logic implemented in Figure 5. However, the limitation is that the internal digital control design logic can not be revised. Thus, the effort has been to accommodate proposed energy-based primary control indirectly through other signals that can be passed through PSI's API.

In order to do so, it is imperative first to learn the approximate closed-loop model through measurements. This is needed to approximate the efficiency of the inverter switching required for the subsequent control design. With the unknown internal automation, it can be assumed that the closed-loop dynamical model of the filter at the interface of the EV is given as:

$$\frac{di}{dt} = -k'(i - i^{ref}) \quad (3)$$

Here, i is the current entering the EV, i^{ref} is the charge rate setpoint that can be provided through the API. The PWM switching logic is unknown, but through measurements, the rate at which the current measurements track the reference value can be estimated, which is denoted as $1/k'$. Currently, the devices charge at the maximum rate of 30 A, which can be regulated to any other value in times of need.

Conventional embedded automation requires perfect cancellation of the filter damping to attain the closed-loop model in (1). Shown in Figure 1 is the case when damping is perfect known and when it is known only with 80% accuracy.

The objective of our control implementation logic in this project is to utilize existing automation and only change i^{ref} control signal so that it results in the power dynamics to represent the general primary control design shown in Figure 5.

We design the control signal i^{ref} as follows

$$i^{ref}[p+1] = (-R_{eq}i[p] + P_u[p])\frac{1}{k'} + i[p] \quad (4)$$

Here, $i[p]$ is the current measurement sent out by sensors (These measurements can also be asynchronous). R_{eq} is the approximate damping of the filter. It must be noted that the approximate value of R_{eq} is sufficient for the desired control performance.

$P_u[p]$ is the controlled power which is incremented using the following dynamical control law:

$$P_u[p + 1] = P_u[p] + T_p \alpha \text{sign}(y_z[p] - y_z^{ref}[n]) \quad (5)$$

Here, T_p is the control implementation timestep, α is the constant that upper bounds the effects of the reactive power imbalance as dictated by the second equation of the energy space model in Eqn. 5. $y_z^{ref}[n]$ are the setpoints sent out by the secondary layer control of the EV, the computation of which will be explained towards the end of the section.

$y_z[p]$ is the measured output of interest in energy space and is expressed as

$$y_z[p] = (I^{ref} - i[p])k' - P_u[p] \quad (6)$$

Here, I^{ref} is the fixed charge rate that is set by the user ahead-of-time. This value for Nissan Leaf EV under consideration is set to 10 A. It should be noted that this value is different from the control signals $i^{ref}[p]$ in Eqn. (4).

This implementation of the aforementioned primary control given by equations (4)-(6) provably tracks any $y_z^{ref}[n]$, sent by the secondary layer control that accommodates the droop relations based on proposed automation limited by the physical capacity ratings.

Shown in Figure 7 is the example of EV control achieving the tracking of a step signal change in $y_z^{ref}[n]$ at $t = 0.5$ seconds. The resulting electrical power inputs of the EV are shown in Figure 8. Figure 8 shows the 5-minute averaged power consumption signals. Clearly, the power consumption settles at the reserve magnitude target of 0.25 kW. within 0.06 hours = 3.6 minutes, which can further be improved by increasing the value of α in Eqn. (5). The sliding mode control implementation in energy space leads to the chattering effect resulting in a reserve magnitude variability of approximately 5%. This can be reduced by selecting a threshold for the sliding mode control implementation in energy space for the sign function implementation in Eqn. (5). The control signals utilized for implementation are the current reference signals that are implemented through the API provided by PSI are shown in Figure 9.

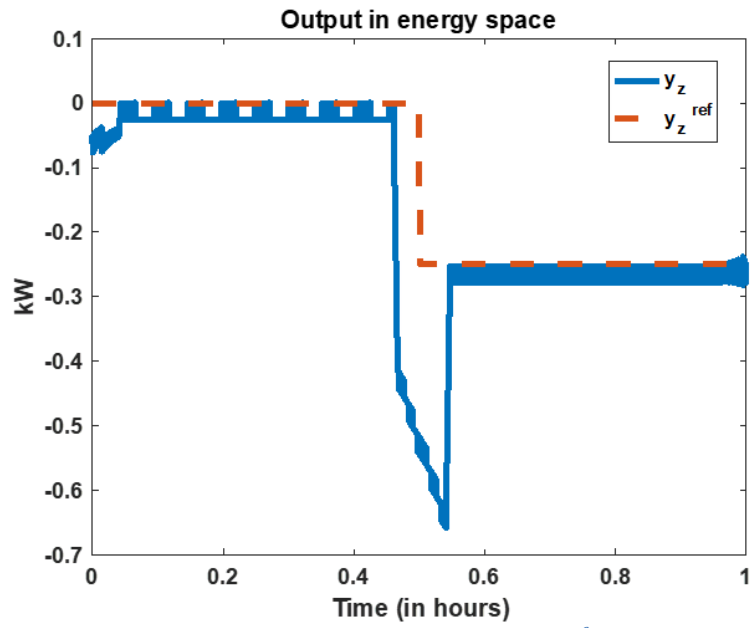


Figure 7: Output of interest in energy space y_z reaching the reference value $y_z^{ref}[n]$ sent by secondary layer control

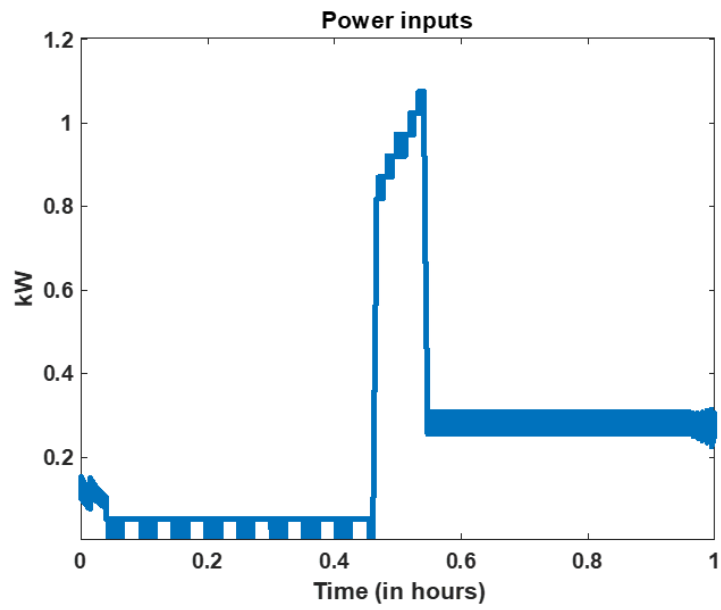


Figure 8: Electrical power consumption of EV

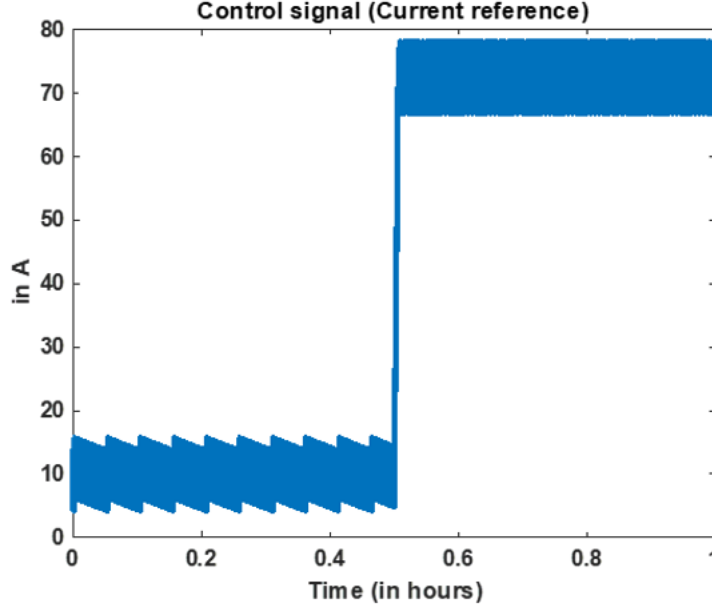


Figure 9: Current reference signals fed as control inputs for implementation through PSI API.

4.4.1.2. HVAC Control

The model of HVAC was studied in the previous project with Eaton, and it was identified that it could be sufficiently represented by the dynamics of the temperature of a single zone. The evolution of the room temperature is:

$$\frac{dT}{dt} = -\frac{1}{RC}(T - T_0) + \frac{1}{C}P^{rated}u \quad (7)$$

with the initial condition $T(0) = T^{ref}$, where T^{ref} is the consumer-set desired temperature.

$\frac{1}{R}$ is a function of the area and the heat resistance of the wall, ceiling, and window, and the heat gains from solar and people in the room. C is the energy needed to change the room temperature by 1 °F. The unit is Btu/°F. It is calculated as: $\Delta c = C_{air} * V_{house}$, where C_{air} is the heat capacity of air and the typical value of it is: 1.012 J/gK, or 0.0195 Btu/ft³°F, and V_{house} is the volume of the house.

With the unknown internal automation, it can be assumed that the closed loop dynamical model of the temperature dynamics can be written as:

For cooling case:

$$u[p] = \begin{cases} 0 & T[p] < T^{ref} - T_{db} \\ 1 & T[p] > T^{ref} + T_{db} \\ u[p-1] & \text{otherwise} \end{cases} \quad (8)$$

Here, T_{db} is the permissible temperature deviations from the reference value which has been selected for this experiment as 1.5 °F. The simplified HVAC model for cooling is explained using (7),(8). The sign of $u[p]$ can be similarly changed for the case of heating.

The thermal resistance and capacitance can be easily obtained from the first order equation of the thermal dynamics and the switching control resulting in the ON duration and OFF duration as follows. We utilize this approximate model, consider the various constraints imposed by the manufacturer provided by PSI to implement HVAC switching control.

The overall control architecture of the HVAC as it is implemented for the project is summarized in Figure 10.

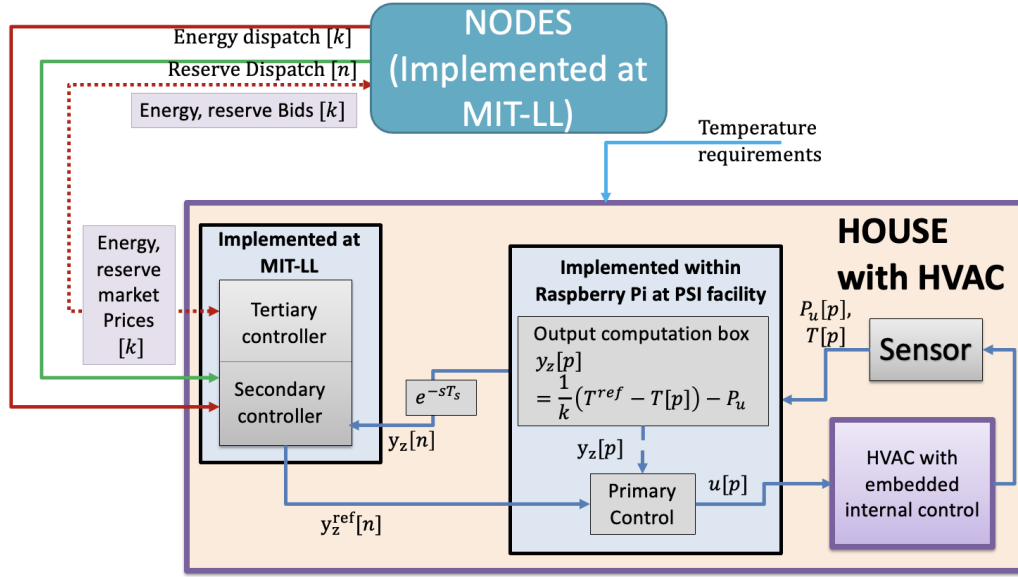


Figure 10: The multi-layered control of HVAC- Dark Blue, green and red arrows represent information communicated at T_p, T_s, T_t rate respectively. Light blue arrows represent information specified by the users which is either fixed or changes at much slower timescales. Furthermore, dotted lines represent ahead-of-time interactive information exchange

The objective of control implementation logic is to override the switching cycles based on temperature with that of an advanced switching-based logic so that it results in the power dynamics to represent the general primary control design shown in Figure 4.

The proposed logic to override the switching cycles is given by Eqn. (9)

$$u[p] = \begin{cases} 1 & y_z[p] - y_z^{ref}[n] > \frac{T_{db}}{R_{eq}} \\ -1 & y_z[p] - y_z^{ref}[n] < -\frac{T_{db}}{R_{eq}} \\ u[p-1] & \text{otherwise} \end{cases} \quad (9)$$

Here, $y_z[p]$ is the measured output of interest in energy space and is expressed as

$$y_z[p] = \frac{T^{ref} - T[p]}{R_{eq}} - P_u[p] \quad (10)$$

$P_u[p]$ is the controlled power which is incremented using the following dynamical control law:

$$P_u[p+1] = P_u[p] + T_p \alpha \text{sign}(y_z[p] - y_z^{ref}[n]) \quad (11)$$

Here, T_p is the control implementation timestep, α is the constant that upper bounds the effects of the reactive power imbalance as dictated by the second equation of energy space model in Eqn. 2b.

In order to validate the ARPA-E performance metrics for the provision of reserves, a power adjustment of 0.2 kW at 0.5 hours is initiated. The secondary control action computes the desired reference in energy space to be equal to 0.2 kW. The primary control given by Eqns. (9)-(11) then tracks this reference as shown in Fig. 11.

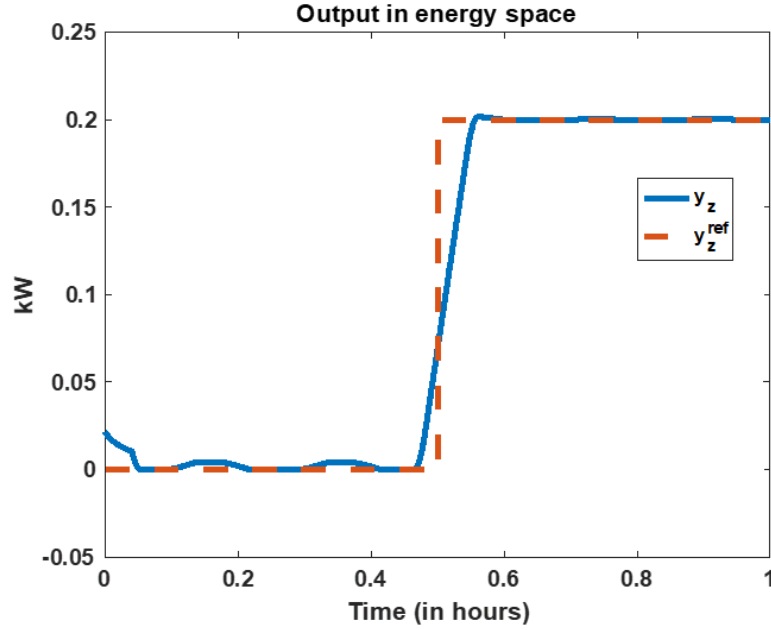


Figure 11: HVAC primary control tacking output in energy space y_z , to the value y_z^{ref} provided by the secondary layer control

Notice that the response time of less than 5 seconds is achieved because of the implementation timestep chosen as 0.36 seconds. The ramp time is clearly less than 5 minutes, which can further be improved with selection of α value in Eqn. 11. The reserve magnitude variability tolerance of less than 5% of the target is also achieved as is evident from the figure. Furthermore, the duration of 30 minutes of reserves availability has also been validated.

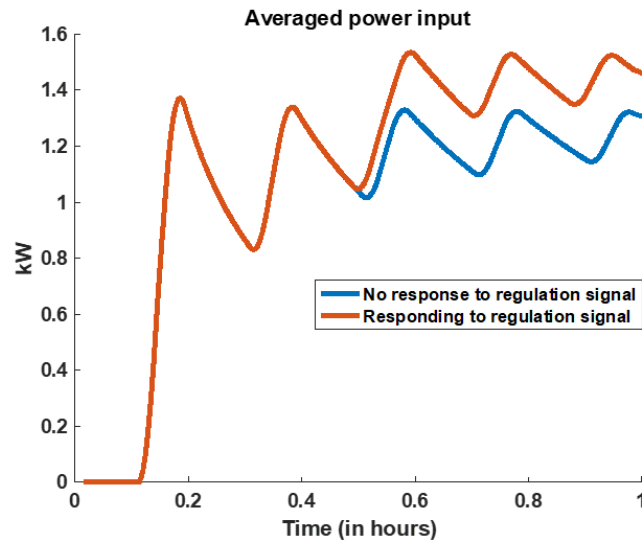


Figure 12: Averaged power consumption of HVAC with and without demand response

The resulting power adjustments are shown in Fig. 12. Here, the base signal utilized for comparison is the one when the embedded automation does not respond to the regulation signal, i.e., the step-change in $y_z^{ref}[n]$ at 0.5 seconds. After 0.5 seconds, notice that there is a difference of approximately 0.2 kW as required by the regulation reserve signal.

The temperature is also ensured to be within the permissible range of 60-65 F as shown in Figure 13. Finally, the imposition of the deadband in Eqn. (9) of the control also ensures that the switching actions are not too frequent, as shown in Figure 14.

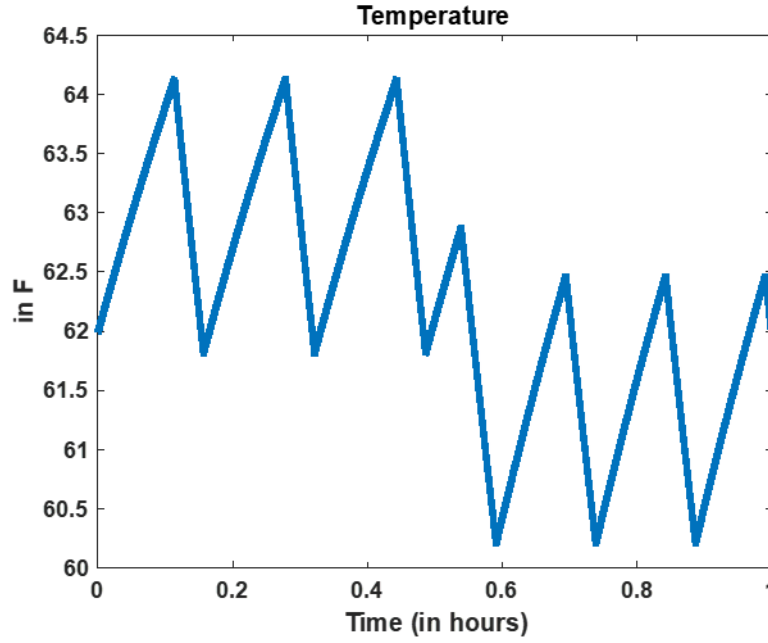


Figure 13: Evolution of temperature with the proposed energy-based control

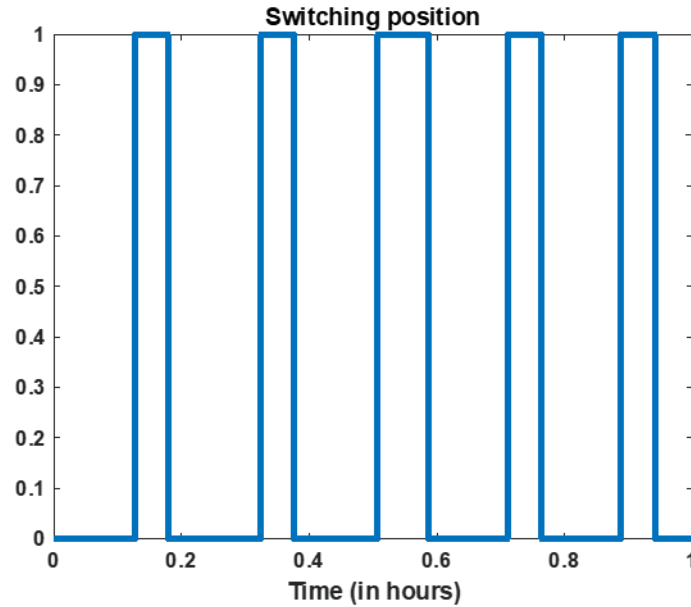


Figure 14: Proposed switching logic for the HVAC

4.4.2. Secondary control design

For either of the devices, upon applying the primary control, the closed-loop model can be utilized to obtain an incremental relation between the outputs in energy space $\Delta y_z[n]$, the power consumption $\Delta P_u[n]$ and the internal outputs of interest $\Delta y^{ref}[n]$ (For example, the current reference in EV and temperature reference in HVAC). This incremental relation is referred to as a droop relation and is provided by Eqns. 12b and 12c below. The constants α and β are operating conditions-dependent constants, which depend on internal automation. For details on these constants, refer to [1,2].

These droop relations are utilized to solve an MPC problem to obtain references for outputs in energy space $y_z^{ref}[n]$ over slower timescales for a horizon length of one or several market-clearing time intervals of length $T_t = 1 \text{ hour}$.

$$\min_{y_z^{ref}[n]} \sum_{nT_s=kT_t}^{(k+1)T_t} \mu^{reg} |\Delta P_u[n] - P^{reg}[n]| + \mu^e P_u[n] \quad (12a)$$

$$\Delta y_z[n] = \Delta y_z^{ref}[n]; \quad y_z[0] = y_{z,0}; y_z^{ref}[0] = y_{z,0}^{ref} \quad (12b)$$

$$\Delta y_z[n] = \alpha \Delta y^{ref}[n] + \beta \Delta P_u[n]; \quad y^{ref}[0] = y_0^{ref} \quad (12c)$$

$$y^{min} \leq y^{ref}[n] \leq y^{max} \quad (12d)$$

Here, μ^{reg} is the penalty of not following the regulation signal, while μ^e is the fixed energy cost being paid by the device. As a result, the objective of secondary control is to optimize the tradeoffs between energy consumption and provably supply of reserves. The first two constraints in the above formulation result from the quasi-static droop relations that can be established upon stabilization of the primary-scale dynamics. y^{min}, y^{max} are the limits on the charge rate of the EV or temperature limits in the case of HVAC. These limits are set to be equal to 0 A and 30 A for the level 2 charging of Nissan leaf EV under consideration. For HVAC, these limits are set to 58-66 F.

Problem (12) can be solved for the house with both EVs and HVACs combined. In this experiment, however, the EVs and HVACs belonged to different houses. The result of this optimization is the sequence of reference signals $y_z^{ref}[n]$ to which the primary control responds to. Notice that the secondary control MPC problem computes the reference signals that would result in the temperatures to be within pre-specified limits of 60-65 F. As a result, the primary control implementation would not lead to saturation.

First, the interactive primary and secondary controllers are simulated for a house with HVAC with permissible temperature s of 60-65 F with secondary control timestep of $T_s = 5 \text{ minutes}$ and a horizon length of 1 step to obtain secondary control actions as the regulation signal arrives. The assumed costs are $\mu^{reg} = 100\$/kWh$ and $\mu^e = 10\$/kWh$. The trajectories obtained with this approach is compared with that when the horizon length is 12 steps, amounting to 1 hour of planning.

The overlaid plots of temperature are shown in Fig. 15. Notice that the temperature crosses the limits at few time instants because of the temperature constraints in Eqn. (12d) were softened by plugging them in objective function with a penalty factor of $1e2$.

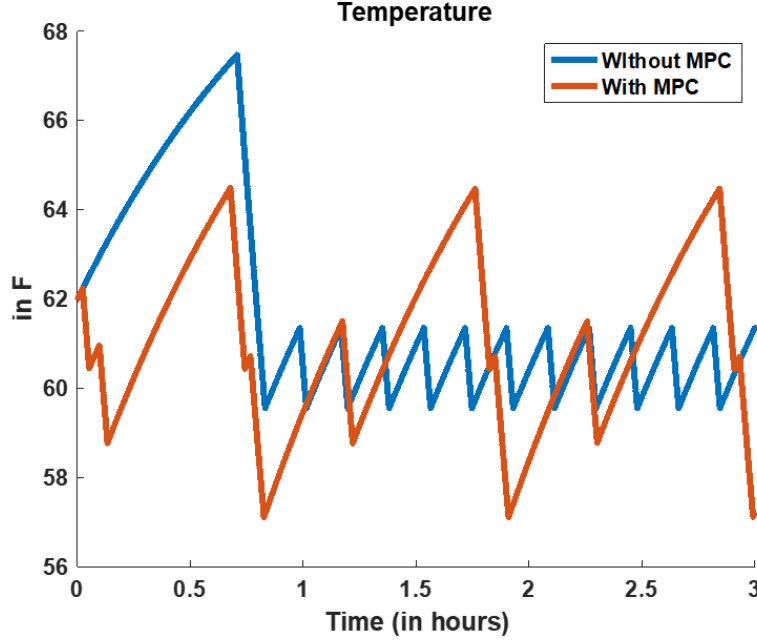


Figure 15: Temperature trajectories with the combined action of primary and secondary controllers.

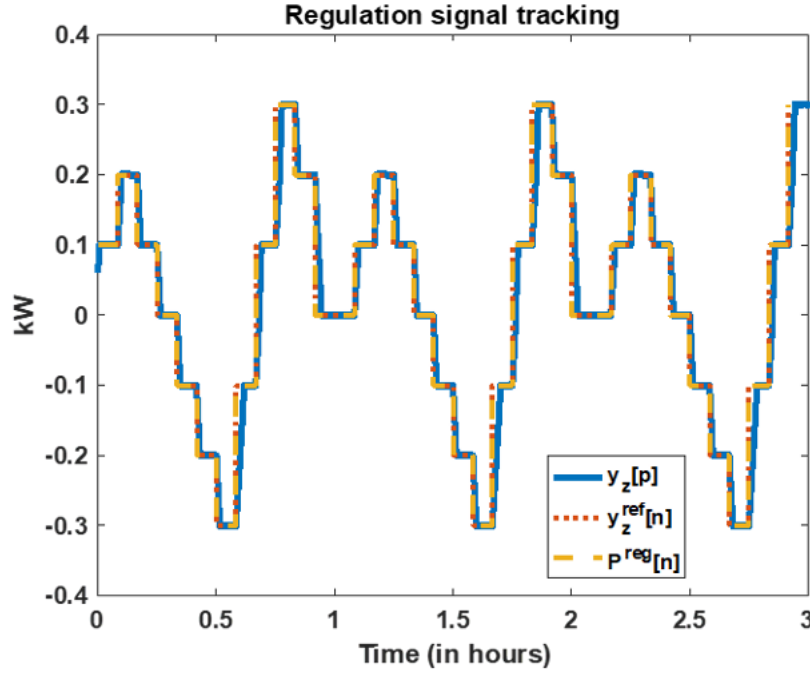


Figure 16: Reference tracking with the MPC-based secondary control actions with $\mu^{reg} = 100\$/kWh$ and $\mu^e = 10\$/KWh$

Fig. 16 and 17 show the result of the control action for the case with and without MPC. In these plots, the primary control action results in fast-changing values of y_z shown in blue perfectly chasing the reference signal $y_z^{ref}[n]$ in red computed every 5 minutes. Overlaid is also the regulation signal, which needs to be the same as the secondary control action y_z^{ref} objective function when compared to the comfort constraints (12c) that are softened and utilized. Notice from Figure 16 that the case with MPC results in perfect tracking of the regulation signal while also ensuring temperatures are within permissible limits.

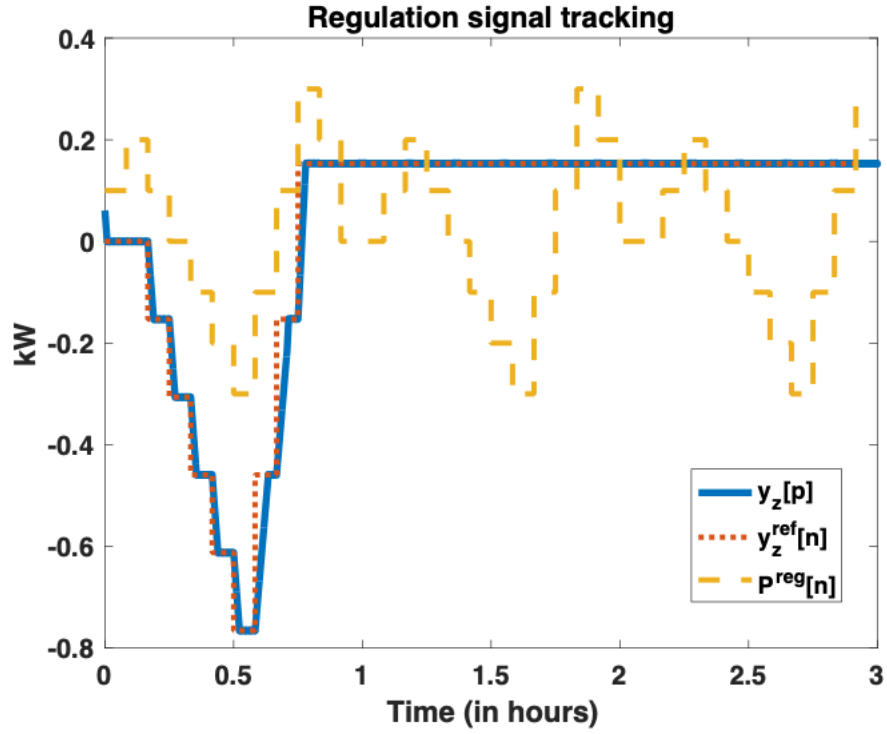


Figure 17: Reference tracking without MPC $\mu^{reg} = 100\$/kWh$ and $\mu^e = 10\$/KWh$

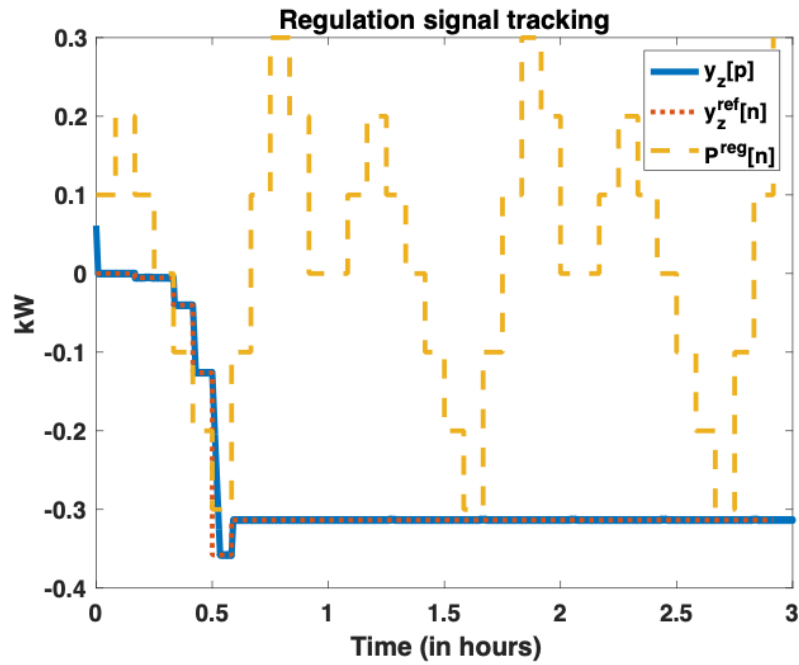


Figure 18: Reference tracking with the MPC-based secondary control actions with $\mu^{reg} = 25\$/kWh$ and $\mu^e = 10\$/KWh$

Shown in Fig. 18 and 19 is the case when reserve penalty cost is four times lower, where the MPC actions still result in much better tracking compared to the case without MPC respectively. Compared to Fig. 15, the temperature profile is now better when provision of reserves is made less important as shown in Fig. 20.

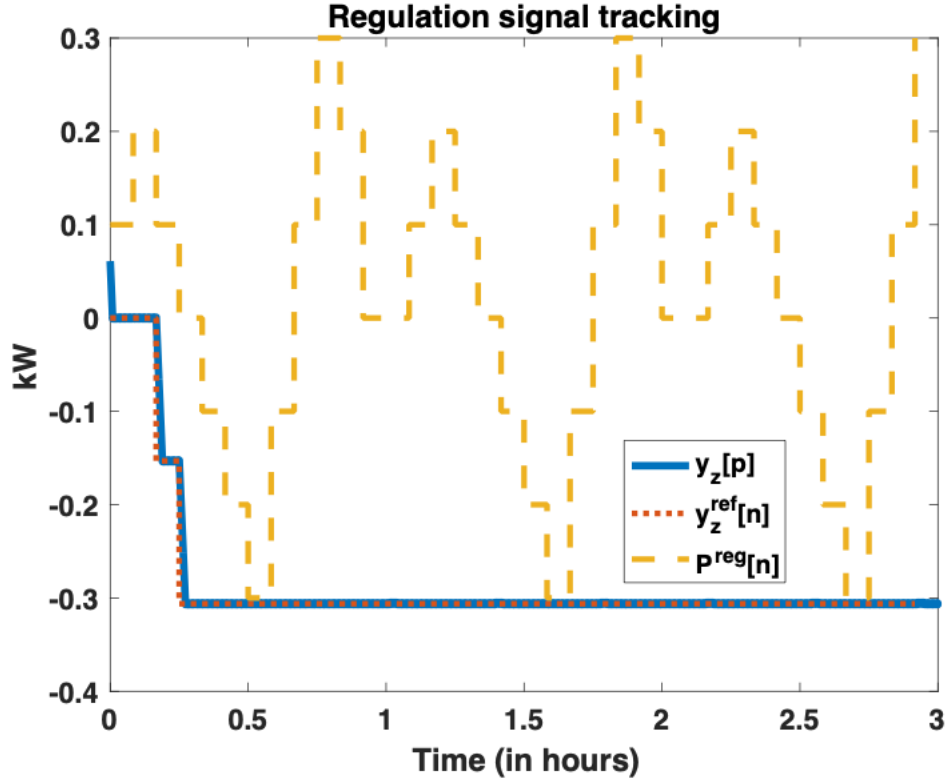


Figure 19: Reference tracking with the MPC-based secondary control actions with $\mu^{reg} = 25\$/kWh$ and $\mu^e = 10\$/KWh$

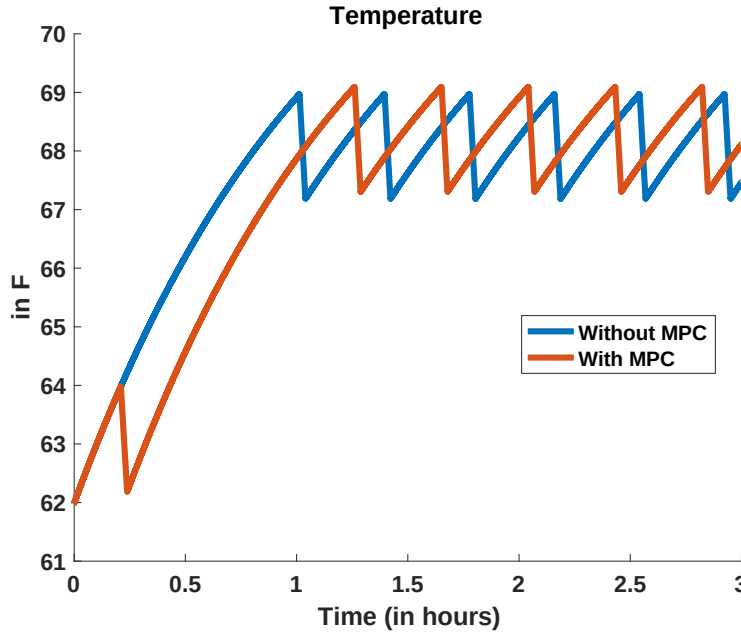


Figure 20: Temperature profile for when control actions are obtained with lower reserve penalty cost of $\mu^{reg} = 25\$/kWh$. Similar conclusions have also been obtained with EV secondary control. Details with comprehensive simulation results can be referred to in [2].

The problem (12) can also be solved further to obtain the limits on reserve supply ahead of time, by considering the objective function as $B[k]$ with an additional constraint $|\Delta P_u[n]| \leq B[k]$. The resulting limits $B^{\min}[k]$, $B^{\max}[k]$ are accounted for in the tertiary control to maximize profits of the house-level controller as explained next.

4.4.3. Tertiary control:

The estimated energy and reserve prices are provided by the price forecaster module implemented on MIT-LL. For details, see [4]. The objective is to maximize the energy arbitrage and the revenues obtained by the provision of reserve capacity over market timescales. However, the total reserve capacity that can be provided by the devices is a result of the embedded automation utilized in the secondary layer.

$$\min_{P_u[k], B[k]} \sum_{kT_t=0}^H \hat{\lambda}_e[k] P_u[k] + \hat{\lambda}_r[k] B[k] \quad (13a)$$

$$0 \leq P_u[k] \pm B[k] \leq P^{max} \quad (13b)$$

$$B^{min}[k] \leq B[k] \leq B^{max}[k] \quad (13c)$$

$$e[k+1] = e[k] + T_t(P_u[k] \pm B[k]) \quad (13d)$$

$$E^{min} \leq e[k']; e[k] \leq E^{max}; \quad (13e)$$

The third constraint models the state of charge evolution of the EV. The second and fourth constraint model the limits on the power and energy that can be injected. The minimum state of charge constraint is the one that dictates the driver's requirements to fulfill his/her driving scheduled by a pre-specified time sample k' . The constraints (13d)-(13e) can be avoided if there is no EV present in the house. Due to COVID-19, the actual hardware testing could not be completed. We have thus taken the measurements of PSI to fit this model to obtain parameters that can be utilized to test the control design through simulations.

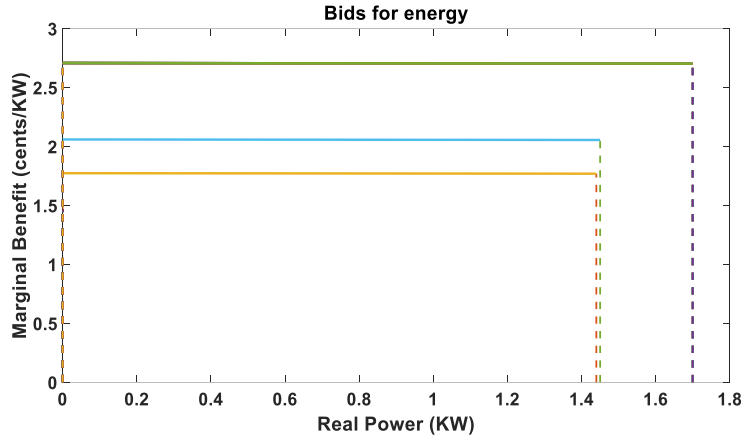


Figure 21: Real time bids for energy with HVAC load

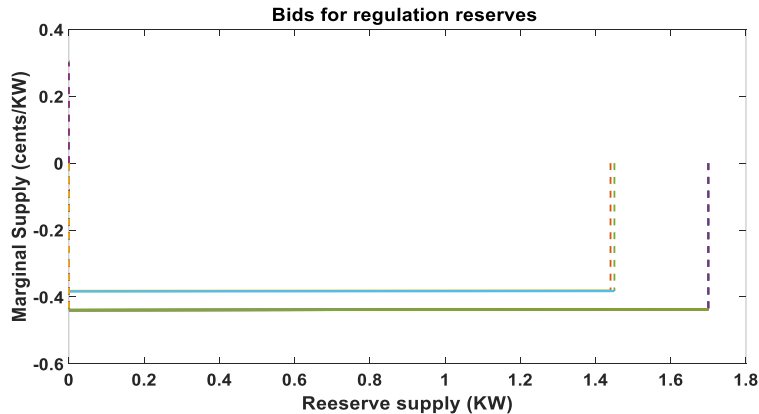


Figure 22: Real time bids for regulation reserves with HVAC load

The DyMonDS based time-varying bids for the energy and regulation reserve markets are as shown in Figure 21 and Figure 22 for three consecutive time instants.

4.5. Feeder-level simulation validation results:

Twenty test scenarios using two types of loads (i.e., HVACs and electric vehicles) were simulated to demonstrate the superior performance of the proposed control design. The simulation results validated that the control performance meets the metrics defined in the milestone. Monte-Carlo randomness is considered in conducting this milestone task: the 20 test scenarios are randomly selected from 72 different possible combinations using Monte-Carlo method

4.5.1. Simulation Setup

This section describes the simulation setup in terms of the feeder system used, the composition of controllable load types, configuration parameters of individual loads, and all required input data needed for simulations.

System Under Study

All simulations conducted in this task are based on the Pecan Street Mueller Community distribution system. This system contains 1 feeder load with 25 homes incident. PSI lab facility is also at the same location with the following controllable units:

- 2 controllable electric vehicles (EVs)
- 1 controllable HVAC

EVs have a rated power of 3 kW, battery size of approximately 30 kWh with close to 90% charging and discharging efficiency. The two EVs are at different initial SOC values at different times of the day under consideration in the simulation study. Controllable HVAC unit is assumed to have a permissible temperature deviation of HVAC was 67-73 F.

Figure 23 and 24 shows the plots of net inflexible demand over the two days in the month of July under consideration. Overlaid is also the solar power, which provides reasoning for the negative net demand for a few time instants.

The demand response signal references for the aggregate of the controllable units is then given by:

$$DR_Pload_ref(nT_s) = DR_Pload_act[t_{start}] + RMT(kT_t)$$

In the above equation, $DR_Pload_ref(nT_s)$ is the time varying total load reference for the controllable water heaters and electric vehicles. $DR_Pload_act(t_{start})$ is the total load of the controllable devices at the starting of the demand response. This value sets the baseline to evaluate the reserve contribution of the devices.

$RMT(kT_t)$ is the difference between the total controllable devices' load and the baseline power, or the so-called reserve magnitude target (RMT) as defined in the NODES FOA, which is found every market interval of $T_t = 1 \text{ hour}$

$$RMT(kT_t) = 7\% \text{ of } \max_{nT_s \in [kT_t, (k+1)T_t]} |PS_{net}(nT_s)|$$

In the above equation, $PS_{net}(nT_s)$ is the fine granular net load of the Pecan street feeder under consideration. $PS_{net}(kT_t)$ is the scheduled feeder load over market timescales.

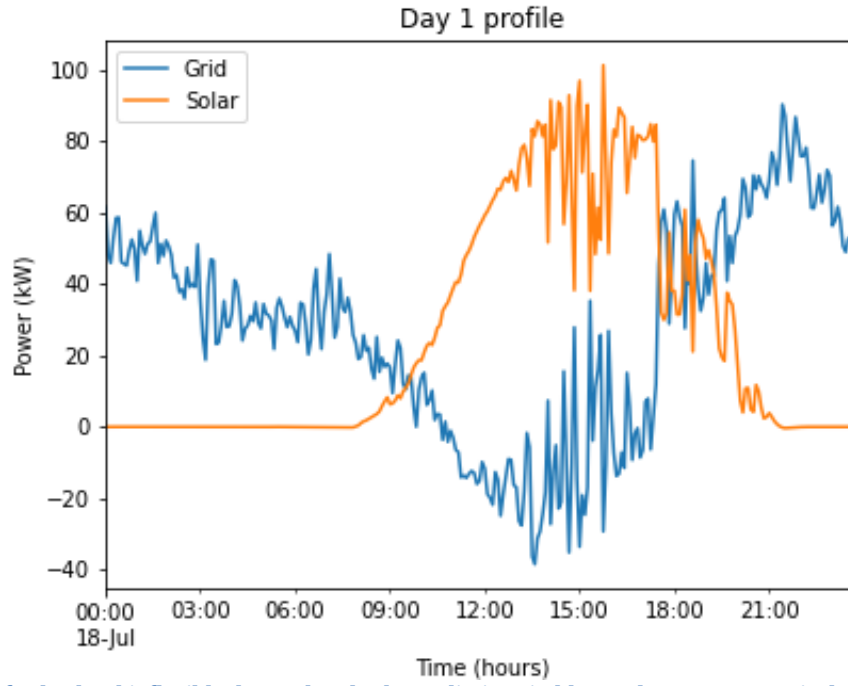


Figure 23: Day1 feeder-level inflexible demand and solar radiations in blue and orange respectively

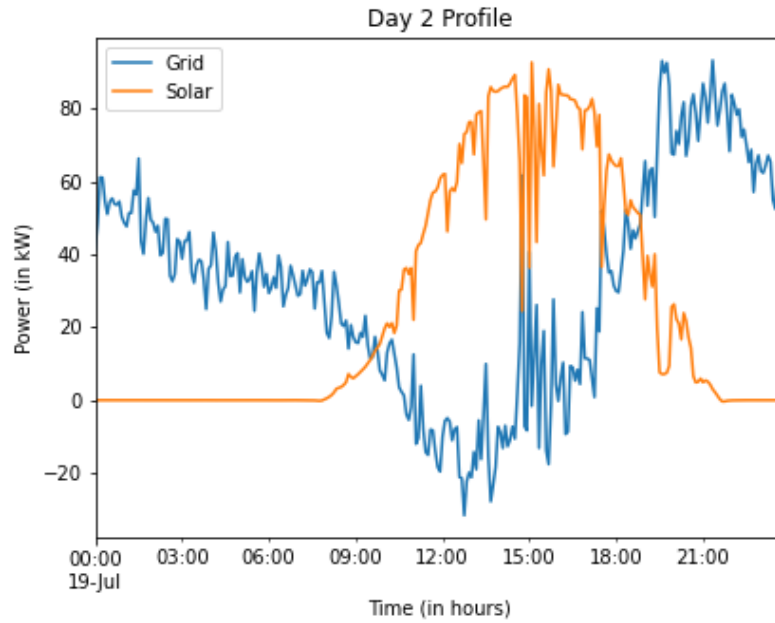


Figure 24: Day2 feeder-level inflexible demand and solar radiations in blue and orange respectively

Test Scenarios

Many different variations are considered in setting up the test scenarios to assure an comprehensive evaluation of the developed control under most conceivable situations in the real world. These variations are summarized in Table 1. The simulated day and the time of the day result in different net loading scenario and thereby result in different reserve targets. Furthermore, the EV would

have different initial state of charge and the HVAC would have a different initial and ambient temperature.

Table 1: Variations in test scenario definition

Simulated Day	Time of the Day	Regulation Reserve Type
Day 1	2AM 9AM	Step Response (up reserve) Step Response (down reserve)
Day 2	2PM 8PM	AGC Tracking

The step response regulation-reserve-type scenario is considered to validate performance metrics as defined in NODES FOA. But in reality, the AGC signal is time-varying within the market-clearing interval, i.e. it varies over secondary control timescale T_s . A realistic AGC signal is produced by scaling down the AGC signal utilized in CAISO. This information was provided in the previous project phase by LLNL.

Out of the 72 possible combinations, 20 tests are randomly selected using Monte-Carlo method and are listed in Table 2.

Table 2: List of evaluated test scenarios

Test #	Simulated Day	Time of the Day	Regulation Reserve Type
1	Day 1	8:00PM	Step response (up reserve)
2	Day 1	2:00AM	Step response (up reserve)
3	Day 1	9:00AM	Step response (up reserve)
4	Day 1	8:00PM	Step response (down reserve)
5	Day 1	2:00PM	Step response (up reserve)
6	Day 1	9:00AM	Step response (down reserve)
7	Day 2	2:00PM	Step response (down reserve)
8	Day 2	2:00PM	Step response (up reserve)
9	Day 2	2:00AM	Step response (up reserve)
10	Day 2	2:00AM	Step response (up reserve)
11	Day 2	8:00PM	AGC tracking
12	Day 2	2:00AM	AGC tracking
13	Day 1	2:00AM	AGC tracking
14	Day 1	2:00PM	AGC tracking
15	Day 1	8:00PM	AGC tracking
16	Day 1	2:00AM	AGC tracking
17	Day 2	2:00PM	AGC tracking
18	Day 2	8:00PM	AGC tracking
19	Day 2	2:00AM	AGC tracking
20	Day 2	9:00AM	AGC tracking

4.5.2. Simulation Result

One important objective of the task is to validate the control to meet the performance metrics as defined in this project's proposal.

Perf Targets: Initial Response Time < 5 s, Reserve Magnitude Variability Tolerance (RMTV) < 5%, Ramp Time < 5 m, Duration > 60 m

As the performance metrics are defined based on a step response behavior, only group 1 (the first ten simulation cases) are evaluated against the target metrics. Table 3 summarizes the simulation setup of group 1 cases. Group 2 AGC tracking cases are further validated for meeting NODES performance metrics in [3].

Table 3: Summary of group 1 simulation setups

Test Number	Day Number	Direction	Time of the day	Max. Hourly net Load (KW)	Reserve target (KW)
1	1	up	08:00 PM	70.365	4.925
2	1	Up	02:00 AM	50.951	3.5665
3	1	Up	09:00 AM	24.064	1.684
4	1	Down	08:00 PM	70.365	4.925
5	1	Up	02:00 PM	27.755	1.942
6	1	Down	09:00 AM	24.064	1.684
7	2	Down	02:00 PM	61.686	4.318
8	2	Up	02:00 PM	61.686	4.318
9	2	Up	02:00 AM	49.758	3.483
10	2	Down	02:00 AM	49.758	3.483

The device controllers are implemented on the Raspberry PI boards, and thus the response time is equal to the sensor delay, and there is not much communication delay. The response time is thus less than 1s.

The secondary control has been implemented with a timestep of 5 minutes to satisfy NODES FOA's ramp time requirements. However, the primary control is implemented to chase the secondary control reference signals even faster by increasing the sliding mode control gain α in Eqn. (5) and (11).

The RMTV is adjusted by selecting the temperature deadband in HVAC control. Currently, the temperature deadband is set to meet comfort constraints and to also avoid excessive switching cycling that results in wear and tear. There thus is a tradeoff between RMTV and device-specific constraints. More the number of HVAC units, the tradeoff effects can be minimized. For the limited number of units under consideration in the simulation study, the RMTV is noted in the tables. Notice that for certain time instants, the RMTV metric imposed by NODES FOA is violated

The DERs are dispatched for providing regulation reserves over secondary control timescales within the limits that have already been agreed upon by the DERs considering its internal requirements over market-clearing tertiary control timescales. Since the tertiary control time step is chosen as 1 hour, guaranteed participation of reserve provision is ensured.

The simulation results corresponding to the randomly selected case is plotted in Figure 25 to Figure 30. Each simulated case is presented with three figures: (a) the power response of the devices; (b) the market dispatch quantities for energy and reserve capacity to each of the device aggregates every hour; and (c) the actual power output from a few individual water heaters and EVs, along with their internal state evolution.

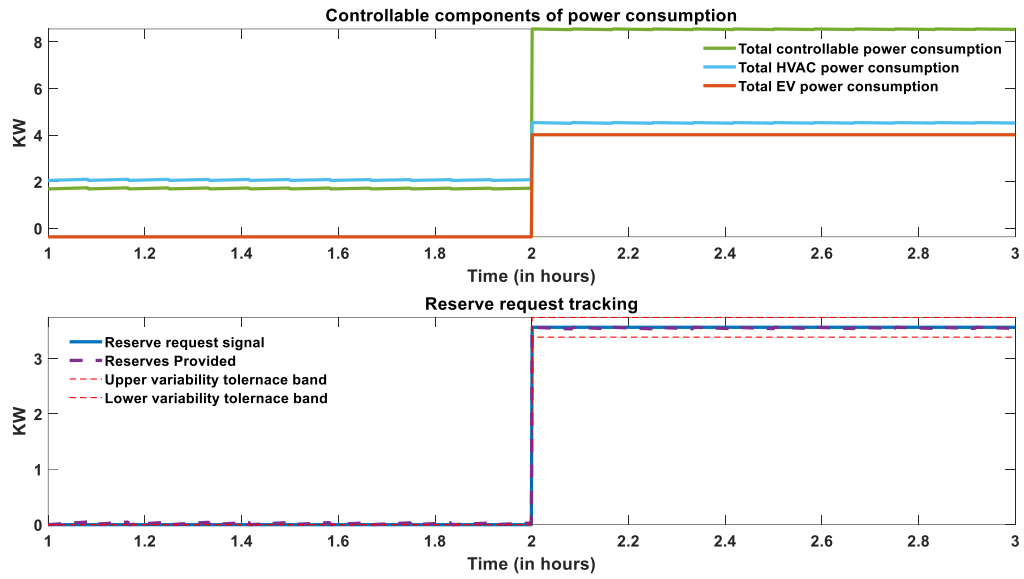


Figure 25: (Test 2) Reserve request outcomes for RegUp signal of 3.5665 kW

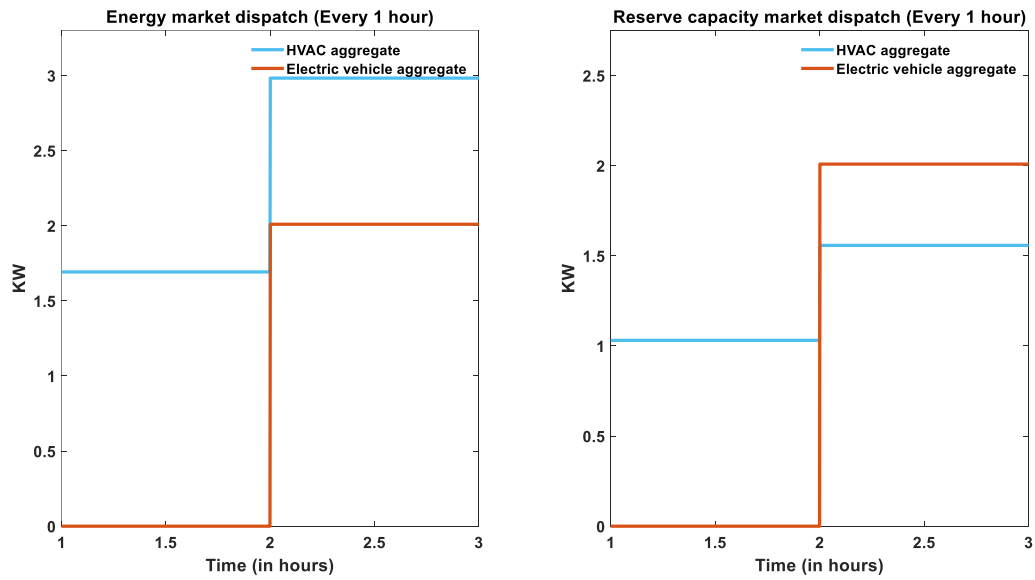


Figure 26: (Test 2) Energy and reserve capacity market outcomes while tracking RMT = 3.5665 kW

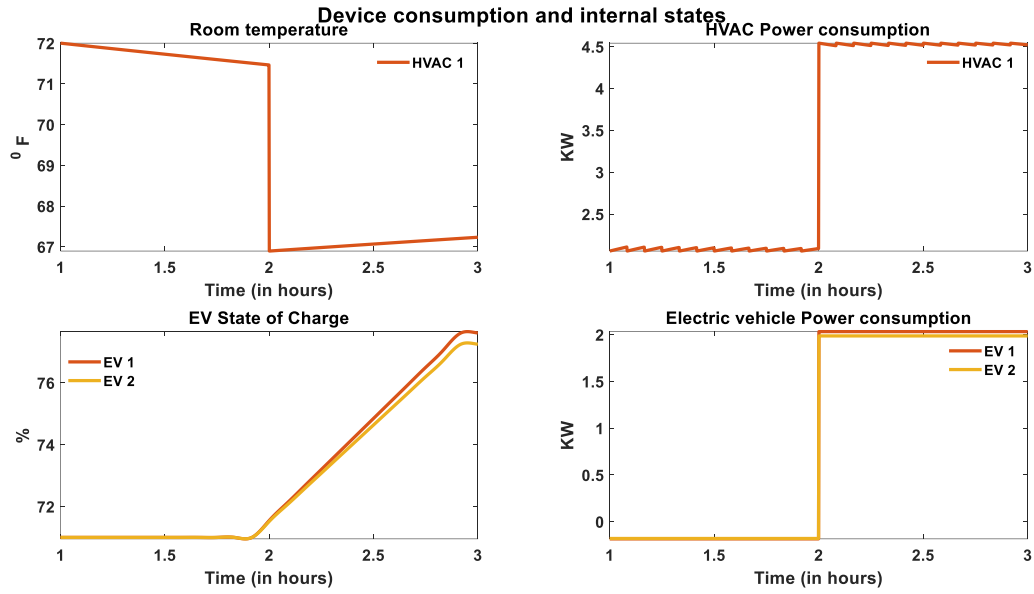


Figure 27: (Test 2) Closed-loop response of individual devices while tracking RMT = 3.5665 kW

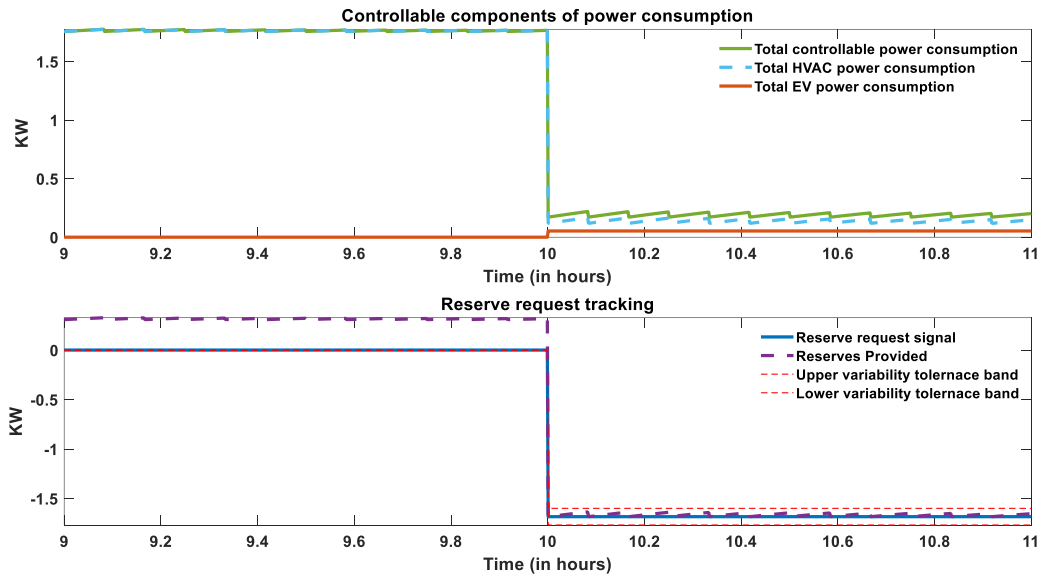


Figure 28: (Test 6) Reserve request outcomes for RegDown signal of 1.684 kW

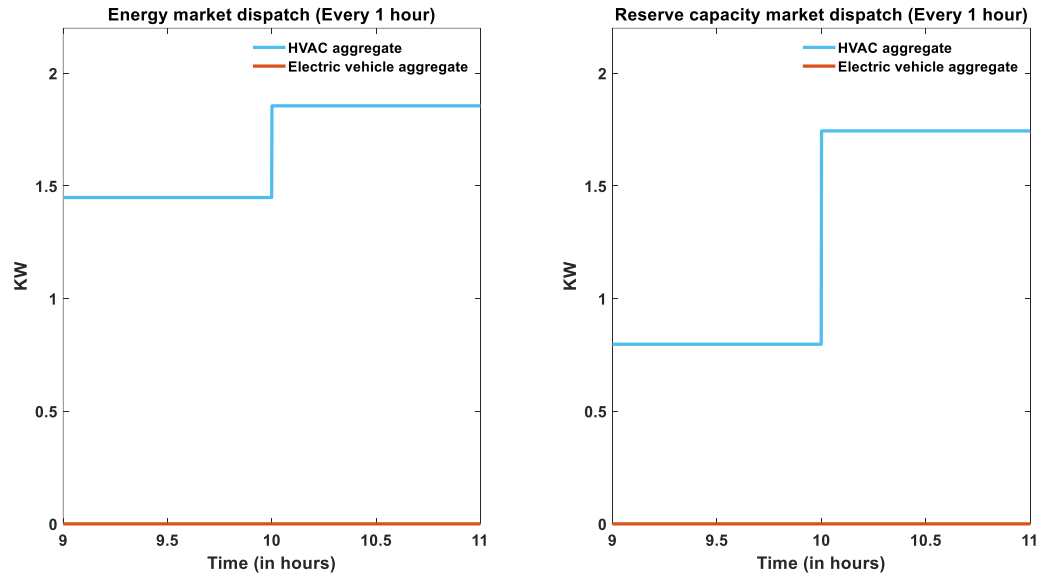


Figure 29: (Test 6) Energy and reserve capacity market outcomes while tracking RMT = -1.684 kW

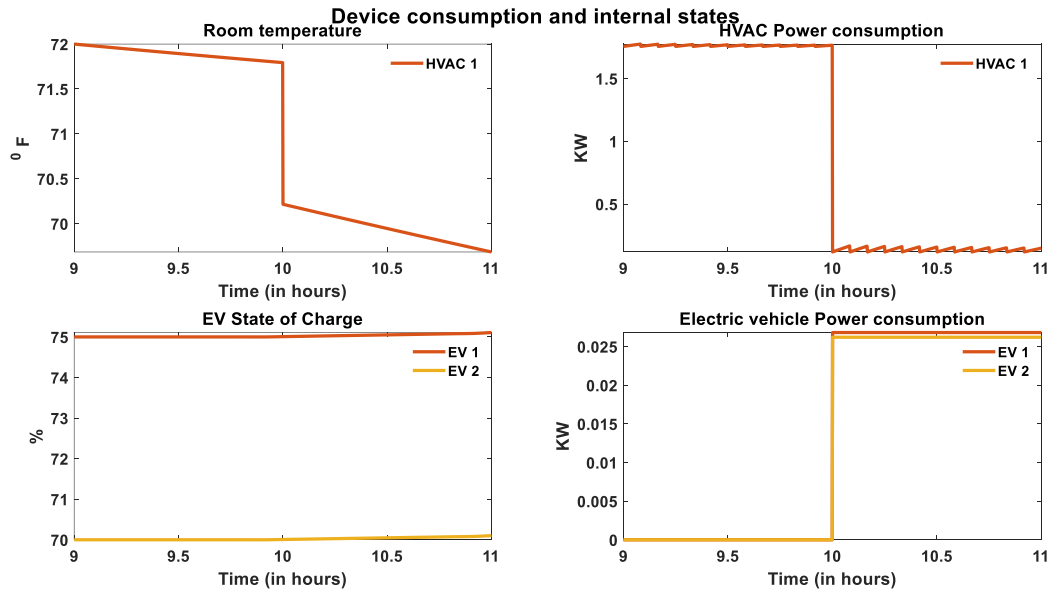


Figure 30: (Test 6) Closed-loop response of individual devices while tracking RMT = -1.684 kW

The overall outcomes of all the group-1 tests conducted are tabulated below in Table 4 case by case.

Table 4: Summary of group 1 simulation results

Test Number	Response Time (in s)	Ramp Time (in s)	RMTV (in % of RMT)	Duration
1	< 1 s	7.2s.	3.44	> 1 hour
2	< 1 s	7.2 s	0.67	> 1 hour
3	< 1 s	3.6 s	1.39	> 1 hour
4	< 1 s	7.2s	3.77	> 1 hour
5	< 1 s	3.6 s	1.34	> 1 hour

6	< 1 s	7.2 s	1.85	> 1 hour
7	< 1 s	3.6 s	0.86	> 1 hour
8	< 1 s	7.2 s	0.53	> 1 hour
9	< 1 s	7.2 s	0.69	> 1 hour
10	< 1 s	3.6 s	1.17	> 1 hour

Our proposed approach only guarantees provable tracking performance. If there are insufficient devices at the market-clearing stage, the minimum storage requirements can be identified. However, the minimum number of devices required to track a given RMT is a subject of future research.

4.6. References

- [1] Marija D. Ilic, Rupamathi Jaddivada "Interactive energy-based provable control of residential HVAC units with limited sensing and actuation", EESG Working Paper R-2020-20
- [2] Marija D. Ilic, Rupamathi Jaddivada "Interactive energy-based cognizant control of Electric vehicle units with limited sensing and actuation", EESG Working Paper R-2020-21
- [3] Marija D. Ilic, Rupamathi Jaddivada "Multi-layered end-to-end interactive control of large number of distributed energy resources", EESG Working Paper R-2020-22
- [4] Ilic, M.D., Joo, J.Y., Andrews, B.W., Raghunathan, B., Benitez, D. and Maus, F., Robert Bosch GmbH, 2012. Adaptive load management: a system for incorporating customer electrical demand information for demand and supply side energy management. US Patent Application 12/895,780.
- [5] Joo, J.Y., Gu, Y., Xie, L., Donadee, J. and Ilić, M., 2013. Look-Ahead Model-Predictive Generation and Demand Dispatch for Managing Uncertainties. In *Engineering IT-Enabled Sustainable Electricity Services* (pp. 247-259). Springer, Boston, MA.
- [6] Jaddivada, Rupamathi, and Ilic, Marija. "A distribution management system for implementing synthetic regulation reserve control." 49th North American Power Symposium, 2017 IEEE. IEEE, 2017.
- [7] Ilic, M. and Jaddivada, R., 2019, September. Toward technically feasible and economically efficient integration of distributed energy resources. In *2019 57th Annual Allerton Conference on Communication, Control, and Computing (Allerton)* (pp. 796-803). IEEE.
- [8] Ilić, M.D. and Jaddivada, R., 2018. Multi-layered interactive energy space modeling for near-optimal electrification of terrestrial, shipboard and aircraft systems. *Annual Reviews in Control*, 45, pp.52-75.
- [9] Ilic, M.D. and Jaddivada, R., 2018, December. Fundamental modeling and conditions for realizable and efficient energy systems. In *2018 IEEE Conference on Decision and Control (CDC)* (pp. 5694-5701). IEEE.
- [10] Ilic, M.D., Jaddivada, R. and Korpas, M., 2020. Interactive protocols for distributed energy resource management systems (DERMS). *IET Generation, Transmission & Distribution*, 14(11), pp.2065-2081.
- [11] Miao, X., Ilić, M., Smith, C., Overlin, M. and Wiechens, R., 2020, October. Toward Distributed Control for Reconfigurable Robust Microgrids. In *2020 IEEE Energy Conversion Congress and Exposition (ECCE)* (pp. 4634-4641). IEEE.
- [12] Miao, X., 2019. Toward Distributed Control for Autonomous Electrical Energy Systems (AEESs)

4.7. Technology Transfer Activities

A. Journal Articles/ Book chapters

- i. Jaddivada, R., 2020. *A unified modeling for control of reactive power dynamics in electrical energy systems*, MIT EECS PhD Thesis
- ii. Ilic, M. and Jaddivada, R., 2019. New Energy Space Modeling for Optimization and Control in Electric Energy Systems. *Optimization and Engineering*.
- iii. Millner, A.R., Smith, C.L., Jaddivada, R. and Ilic, M.D., 2019. Component Standards for Stable Microgrids. *IEEE Transactions on Power Systems*, 34(2), pp.852-863.
- iv. Ilic, M., Jaddivada, R., Miao, X. and Popli, N., 2019. Toward Multi-Layered MPC for Complex Electric Energy Systems. In *Handbook of Model Predictive Control* (pp. 625-663). Birkhäuser, Cham.
- v. Ilić, M. D., & Jaddivada, R. (2018). Multi-layered interactive energy space modeling for near-optimal electrification of terrestrial, shipboard and aircraft systems. *Annual Reviews in Control*.
- vi. Marija D. Ilic, Rupamathi Jaddivada "Interactive energy-based provable control of residential HVAC units with limited sensing and actuation", EESG Working Paper R-2020-20 (Under Preparation for submission to IEEE Transactions on Power Systems). **To Appear**
- vii. Marija D. Ilic, Rupamathi Jaddivada "Interactive energy-based cognizant control of Electric vehicle units with limited sensing and actuation", EESG Working Paper R-2020-21 (Under Preparation for submission to IEEE Transactions on Power Systems). **To Appear**
- viii. Marija D. Ilic, Rupamathi Jaddivada "Multi-layered end-to-end interactive control of large number of distributed energy resources", EESG Working Paper R-2020-22 (Under Preparation for submission to IEEE Transactions on Automatica). **To Appear**
- ix. M. D. Ili'c, N. Popli, R. Jaddivada, X. Miao, "Harnessing Flexibilities of Heterogeneous Generation and Demand Technologies in System Operation", MIT-EECS Working Paper R-WP-1-2018, (Under Preparation for submission to IEEE Transactions on Power Systems). **To Appear**

B. Papers

- i. Ilic, M. and Jaddivada, R., 2019. Exergy/energy dynamics-based integrative modeling and control for difficult hybrid aircraft missions. In *AIAA Propulsion and Energy 2019 Forum* (p. 4501).
- ii. Jaddivada, R., Davuluri, S., Korpaas, M. and Ilic, M., 2019, June. Recursive Algorithm for Resource Allocation in Radial Network Systems. In *2019 8th Mediterranean Conference on Embedded Computing (MECO)* (pp. 1-4). IEEE.
- iii. Lauer, M., Jaddivada, R. and Ilic, M., 2019, June. Household Energy Prediction: Methods and Applications for Smarter Grid Design. In *2019 8th Mediterranean Conference on Embedded Computing (MECO)* (pp. 1-4). IEEE.
- iv. Lauer, M., Jaddivada, R. and Ilić, M., 2019, May. Secure Blockchain-Enabled DyMonDS Design. In *Proceedings of the International Conference on Omni-Layer Intelligent Systems* (pp. 191-198). ACM.
- v. Ilic, M.D. and Jaddivada, R., 2018, December. Fundamental Modeling and Conditions for Realizable and Efficient Energy Systems. In *2018 IEEE Conference on Decision and Control (CDC)* (pp. 5694-5701). IEEE.

- vi. Ilić, M.D., Jaddivada, R. and Miao, X., 2018, October. Rapid Automated Assessment of Microgrid Performance Software System (RAMPS). In *2018 IEEE PES Innovative Smart Grid Technologies Conference Europe (ISGT-Europe)* (pp. 1-6). IEEE.
- vii. Ilić, M.D., Jaddivada, R. and Miao, X., 2018, October. Scalable electric power system simulator. In *2018 IEEE PES Innovative Smart Grid Technologies Conference Europe (ISGT-Europe)* (pp. 1-6). IEEE.
- viii. Jaddivada, Rupamathi, and Ilic, Marija. "A distribution management system for implementing synthetic regulation reserve control." 49th North American Power Symposium, 2017 IEEE. IEEE, 2017.

C. Status Reports

D. Media Reports

E. Invention Disclosures

F. Patent Applications

G. Licensed Technologies

H. Networks/Collaborations Fostered

I. Websites Featuring Project Work Results

J. Other Products (e.g., Databases, Physical Collections, Audio/Video, Software, Models, Educational Aids or Curricula, Equipment or Instruments)

K. Awards, Prizes, and Recognition

4.8. Follow-On Funding

No additional funding is committed or received at this point.

4.9. Appendices

4.9.1. APPENDIX A: MILESTONE 2.1 DOCUMENTATION - APIs development completed

High Level Summary

This document demonstrates the deliverables produced by Pecan Street Inc (PSI) towards completion of Task 2: Lab testing environment and development. Contained within at a high level are the following:

- The GitHub repository for the MIT team to host and push their controller code to fielded units. It also contains example python clients for publishing and subscribing to the MQTT broker https://github.com/Pecan-Street/MIT_Controller
- The cloud-hosted server that the MIT team has been given direct ssh access to:
 - Running a PostgreSQL database that the MIT team has been given direct read access to containing the data being published
 - Running an MQTT broker that the MIT has been given credentials for connecting to. This acts as the API for sending HVAC and EVSE commands as well as receiving status information.
 - The cloud-hosted server also listens for specific published MQTT data topics containing the data being published from the Raspberry Pis, and aggregates and stores that data in the local PostgreSQL database. The MIT team may also listen to those topics from any computer with an internet connection for rapid access and response to the published data
- A Raspberry Pi / LCR that is:
 - Publishing whole home usage (grid), HVAC fan/blower usage, EV usage (if home has vehicle), solar generation, vehicle present T/F, vehicle charging, vehicle presence, and temperature data to the MQTT broker on the cloud server
 - Listening to its own command topics from the MQTT broker on the cloud server and reflecting that it has received the different types of commands for EVSE and HVAC control
 - Storing a limited amount of its own data locally in a sqlite database

Cloud Server

PSI has provisioned a cloud hosted Debian Linux server from Linode.com. The IP address is **45.79.13.55** and the hostname is **li1112-55.members.linode.com**. PSI has provided the MIT team with credentials to SSH directly into this host.

Database Access

A PostgreSQL database is hosted containing aggregated data from all homes on the cloud server for the direct read-only access for aggregation algorithm development. PSI has delivered credentials for connecting to the database to the MIT team. Note that these are not the same credentials as used to SSH to the cloud server. The data types in this database are the same as those collected from the homes, stored at 1 minute intervals. The MIT team may SSH to the cloud server to access the database or may connect from their local workstation or laptop to the PostgreSQL database by using the cloud server's hostname and the database credentials provided. The client used to

connect to the database may be a desktop application like PgAdmin, a command-line client like psql, or library access via a programming language like Python3 using a module like psycopg2. A list of clients for many platforms can be found here:

https://wiki.postgresql.org/wiki/PostgreSQL_Clients

To connect using psql after SSH'ing into the cloud server, issue the following command:

```
psql -h 127.0.0.1 -p 5432 -d postgres -U mit_user
```

...then enter the matching database password when prompted. The "public" schema contains the databases with the aggregated data.

If connecting from a remote computer with psql issue command:

```
psql -h 45.79.13.55 -d postgres -p 5432 -U mit_user
```

These same connection parameters can be used in a desktop GUI or Python script to connect to the database. An example of connecting from a remote laptop to the database is seen here:

```
smock@Steves-PCI-MBP ~ % psql -h 45.79.13.55 -d postgres -p 5432 -U mit_user
Password for user mit_user:
psql (12.1, server 9.6.17)
SSL connection (protocol: TLSv1.3, cipher: TLS_AES_256_GCM_SHA384, bits: 256, compression: off)
Type "help" for help.

postgres=> \l

          List of databases
  Name      | Owner   | Encoding | Collate | Ctype   | Access privileges
-----+-----+-----+-----+-----+-----
 postgres   | postgres | UTF8     | en_US.UTF-8 | en_US.UTF-8 | 
 public     | postgres | UTF8     | en_US.UTF-8 | en_US.UTF-8 | 
 template0  | postgres | UTF8     | en_US.UTF-8 | en_US.UTF-8 | =c/postgres +
            |          |          |          |          | postgres=CTc/postgres
 template1  | postgres | UTF8     | en_US.UTF-8 | en_US.UTF-8 | =c/postgres +
            |          |          |          |          | postgres=CTc/postgres
(4 rows)

postgres=> \c postgres
psql (12.1, server 9.6.17)
SSL connection (protocol: TLSv1.3, cipher: TLS_AES_256_GCM_SHA384, bits: 256, compression: off)
You are now connected to database "postgres" as user "mit_user".
postgres=> \dt

          List of relations
 Schema |      Name      | Type  | Owner
-----+-----+-----+-----
 public | energy_usage   | table | esha
 public | energy_usage_test | table | esha
 public | evse           | table | esha
 public | evse_control   | table | esha
 public | hvac           | table | esha
 public | hvac_control   | table | esha
 public | indoor_temperature_data | table | esha
(7 rows)

postgres=>
```

Figure 1: Connecting from a remote laptop to the cloud server's PostgreSQL database containing aggregated status data.

API via MQTT Publish and Subscribe

The cloud server is running an MQTT broker that the MIT team can use to listen to the broadcast in-home 1 second data as well as operational/control status data of WiFi relay, opt- out button and EVSE data. The in-home Raspberry Pi / LCR will collect and broadcast the 1 second data to this MQTT broker on a specific topic to each RPi/LCR based on the last 4 characters of its MAC address. PSI provided the MIT team a document describing all of the topics and data types necessary to both publish commands to the EVSE and HVAC systems, and

to listen for 1 second data being published from the homes. The general topic design is as follows:

For subscribing to published data the topics generally look like:

`MIT/DATA/{MAC}/{SYSTEM}/{TYPE}`

For publishing commands the topics look like:

`MIT/CONTROL/{MAC}/{SYSTEM}`

where {MAC} would be replaced the last 4 characters of the Raspberry Pi's MAC address and

{system} would be replaced with the system that you would like to get information from, such as hvac, temp, evse, or energy. You can add wildcards to the topics to listen to groups. "+" is a wildcard for a single level of a topic, and "#" will add all subtopics below. Examples of subscribing to topics include:

- **mit/data/280a/temp**: Listens to temperature data being sent from the Raspberry Pi with MAC address "280a"
- **mit/data/+temp** : Listens to temperature data from ALL of the participants' Raspberry Pis. The "+" is used as the wildcard for the single level of the MAC address.
- **mit/data/#**: Listens to ALL data being emitted by ALL of the participants' Raspberry Pis. This could be a large volume of messages, so be careful!
- **mit/data/280a/hvac/status** : Listens for the HVAC status information for Raspberry Pi with MAC address "280a"
- **mit/data/280a/evse/status**: Listens for the EVSE status information for Raspberry Pi with MAC address "280a"
- **mit/data/280a/energy/1s_data**: Listens for the energy usage status information for Raspberry Pi with MAC address "280a"

The same wildcard examples shown for the temp system can be applied to the hvac, evse, and energy system topics. The topics to *publish* commands to are:

- **mit/control/280a/hvac**
- **mit/control/280a/evse**

Publishing MQTT messages does not support wildcards, so messages must be sent individually to each Raspberry Pi's MAC address.

PSI provided credentials and connection information for the MQTT broker. The hostname is **li1112-55.members.linode.com**, the username is **mit**, and the port is **8883**. The password was sent to the MIT team and is omitted here for security reasons. SSL is enabled on this port to encrypt communications.

MQTT publisher and subscriber clients also exist as desktop applications, command-line utilities, and software libraries. Here are examples of using the Mosquitto command line clients to publish and subscribe is as follows:

```
mosquitto_pub -h li1112-55.members.linode.com -u mit -P '{YOURPASS}' -t  
mit/control/280a/hvac -m '{"mac":"432a","command": "idle","duration":  
10}'
```

```
mosquitto_sub -h li1112-55.members.linode.com -u mit -P  
'iVogzJp72$^V#wajTRQu' -t 'mit/data/+/energy/1s_data' -v -d
```

Replace {YOURPASS} with your password.

Examples of Python3 code that both publishes commands to the broker to be sent to a specific Raspberry Pi (for example, to set the HVAC *idle* on the Raspberry Pi with MAC address *280a* for *5 minutes*) as well as listens to specific topics to receive status data being published by the Raspberry Pis. The default topic in the example script is "**mit/data/#**", which will listen to all status data from all households and print out all messages being sent out.

There are a number of desktop applications that can be used to both publish and subscribe to the MQTT broker. The following shows configuring the MQTTBox app, and then sending and receiving messages.

The screenshot shows the MQTTBox application window with the title bar 'MQTTBox'. The main content area is titled 'MQTT CLIENT SETTINGS' and includes a 'Menu' button and a 'Client Settings Help' link. The settings are organized into a grid of fields and checkboxes:

- MQTT Client Name:** Text field containing 'MIT'.
- MQTT Client Id:** Text field containing '9e989a40-e169-4093-a04f-3cac9c' with a refresh icon.
- Append timestamp to MQTT client id?:** Checkbox checked 'Yes'.
- Broker is MQTT v3.1.1 compliant?:** Checkbox checked 'Yes'.
- Protocol:** Dropdown menu showing 'mqtt/s / tls'.
- Host:** Text field containing 'li1112-55.members.linode.com'.
- Clean Session?:** Checkbox checked 'Yes'.
- Auto connect on app launch?:** Checkbox checked 'Yes'.
- SSL / TLS Version:** Dropdown menu showing 'Auto'.
- SSL / TLS Certificate Type:** Dropdown menu showing 'CA signed server certificate'.
- Username:** Text field containing 'mit'.
- Password:** Password field with masked characters.
- Reschedule Pings?:** Checkbox checked 'Yes'.
- Queue outgoing QoS zero messages?:** Checkbox checked 'Yes'.
- Reconnect Period (milliseconds):** Text field containing '1000'.
- Connect Timeout (milliseconds):** Text field containing '30000'.
- KeepAlive (seconds):** Text field containing '10'.
- Will - Topic:** Text field containing 'Will - Topic'.
- Will - QoS:** Dropdown menu showing '2 - Exactly Once'.
- Will - Retain:** Checkbox unchecked 'No'.
- Will - Payload:** Empty text field.

At the bottom of the settings grid are two buttons: 'Save' and 'Delete'.

Figure 2: Configuration of MQTT desktop client MQTTBox to publish/subscribe to the PSI MQTT broker.

Menu

←

Connected

Add publisher

Add subscriber

⚙

MIT - mqtt://li1112-55.members.linode.com

Topic to publish

mit/control/280a/hvac

QoS

2 - Exactly Once

Retain

☐

Payload Type

Strings / JSON / XML / Characters

e.g: {'hello':'world'}

Payload

{"mac":"280a","command":"idle","duration":10}

Publish

{"mac":"280a","command":"idle","duration":10}
topic:mit/control/280a/hvac, **qos:**2, **retain:**false

mit/#

{"mac": "280a", "command": "idle", "duration": 5}

qos : 2, **retain :** false, **cmd :** publish, **dup :** false, **topic :** mit/control/280a/hvac, **msgid :** 13, **length :** 74, **Raw payload :** 123341099799345832345056489734443234991111091099711010034583234105100108101344432341001171149711610511111034583253125

{"mac":"280a","command":"idle","duration":10}

qos : 2, **retain :** false, **cmd :** publish, **dup :** false, **topic :** mit/control/280a/hvac, **msgid :** 12, **length :** 72, **Raw payload :** 12334109979934583450564897344434991111091099711010034583234105100108101344434100117114971161051111103458324948125

Figure 3: MQTT desktop application MQTTBox showing publishing a message (left) to topic mit/control/280a/hvac to go idle for 10 minutes, and subscribing to all messages (left, orange) and showing a few message commands have been received.

4.9.2. APPENDIX B: MILESTONE 2.2 DOCUMENTATION – Data import platform enabled

Pecan Street set up the GitHub code repository for pushing code to fielded units.

The GitHub repository has been created and populated. The MIT team will supply their GitHub accounts to be added as contributors to the repo that will host their controller code that will run on the Raspberry Pis. This repo can be found at:

https://github.com/Pecan-Street/MIT_Controller

There is a README file at the root of the repo that has installation and configuration instructions.

In order to provide a well-defined and testable interface between this code and the rest of the code controlling the Raspberry Pi, an abstract interface class has been developed to define the contract (`AbstractController.py`). In addition, an example of an implementation of this interface (`ConcreteControllerExample.py`) and a stub-class (`MITController.py`) that the MIT team can use as a starting point for their development have been provided.

Also provided in this repository are client examples of how to publish and subscribe to the MQTT broker for sending commands to an HVAC system or EVSE system and receiving data published by the Raspberry Pis. These can be found in the `mqtt_client_examples` directory.

Finally, there are unit tests to ensure quality is maintained that are run with each commit.

github.com/PecanStreet/MIT_Controller

PecanStreet / MIT_Controller

Private

Unwatch releases

1

Unstar

1

Fork

0

<> Code

Issues 0

Pull requests 0

Actions

Projects 0

Security

Insights

Code for MIT Project

14 commits

2 branches

0 packages

0 releases

Branch: master

New pull request

Create new file

Upload files

Find file

Clone or download

SMock123

Merge pull request #1 from PecanStreet/feature/examples

Latest commit 882d59d 34 minutes ago

.github/workflows	Create pythonapp.yml	6 days ago
config	Added a subscriber example.	1 hour ago
mqtt_client_examples	Added a subscriber example.	1 hour ago
tests	Fixing some formatting issues.	6 days ago
.gitignore	Adding a .gitignore	4 days ago
AbstractController.py	Adding initial interface class for the controller code, an example im...	12 days ago
ConcreteControllerExample.py	Adding initial interface class for the controller code, an example im...	12 days ago
MITController.py	Fixing some formatting issues.	6 days ago
README.md	Adding example config.txt.	2 hours ago
requirements.txt	Adding example config.txt.	2 hours ago

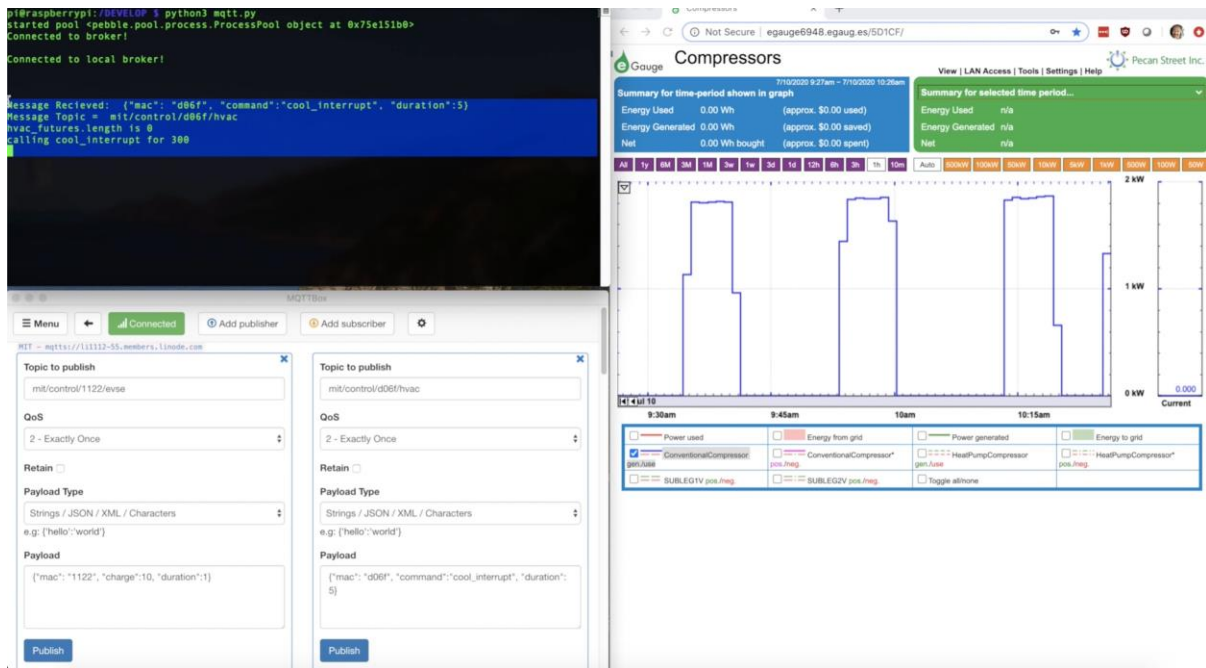
4.9.3. APPENDIX C: MILESTONE 2.3 DOCUMENTATION - Developed device testing framework



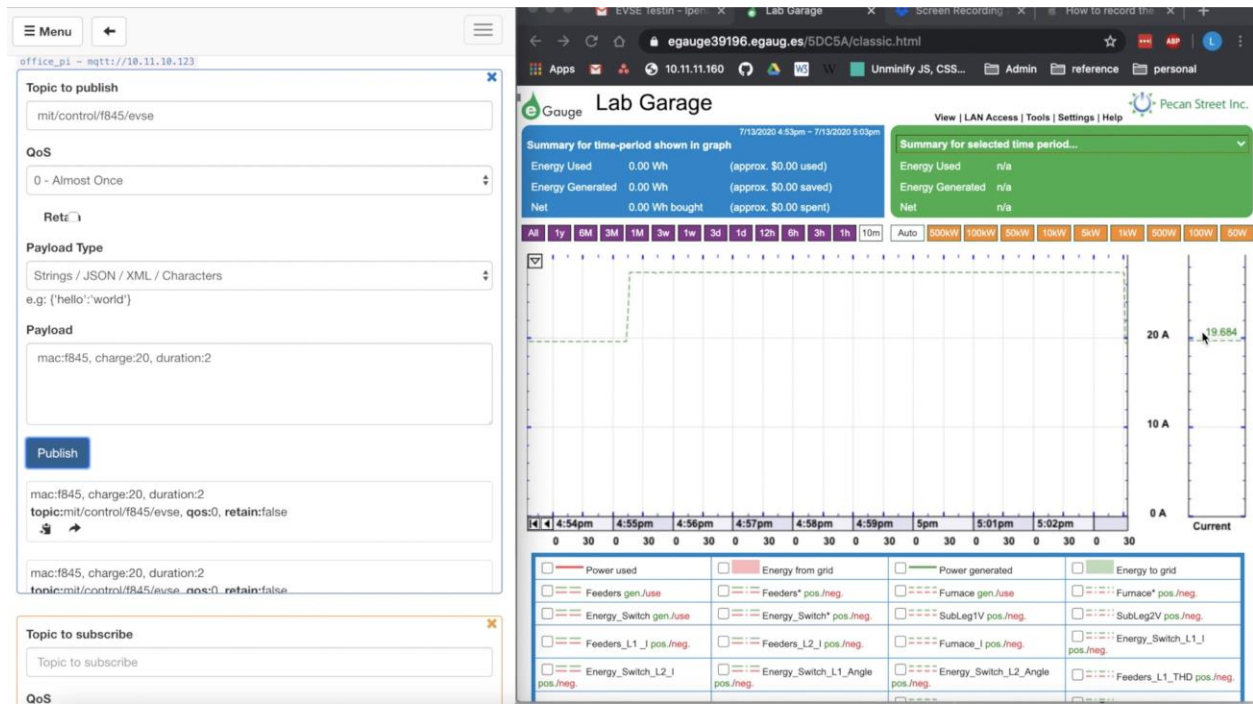
Photographic evidence of the HVAC test set at the PSI lab, including HVAC follower board, single board computer, and thermostat



Photographic evidence of the EVSE test set at the PSI lab, including EVSE follower and J1772 connector



Data verifying simple on/off control for the HVAC compressor at the PSI lab through the API from PSI cloud resources. Please watch the full video to view the HVAC compressor turn off in real time [here](#).



Data verifying simple on/off control (adjustment of the charging rate) for the EV at the PSI lab through the API from PSI cloud resources. Please watch the full video to view the EV adjust its charging rate in real time [here](#).

4.9.4. APPENDIX D: MILESTONE 2.3 DOCUMENTATION -CONTD.

Available MQTT Topics and Commands

Commands may be sent to HVAC and EVSE systems by sending a JSON message to the two control topics on the cloud broker.

EVSE Commands:

EVSE commands consist of setting a charge in Amps for a duration of time specified in minutes to a specific Raspberry Pi as identified by the last 4 characters of its MAC address.

The EVSE command topic is of the form:

```
mit/control/{MAC}/evse
```

For example, if you wanted to send a command to the Raspberry Pi with the MAC address '0cf8' to set the charge to 10A for 5 minutes, you would send the following JSON message:

```
{"mac": "0cf8", "charge":10, "duration":5}
```

to topic

```
mit/control/0cf8/evse
```

HVAC Commands:

HVAC commands similarly consist of sending a JSON message to a topic with a MAC address, a command, and a duration specified in minutes.

The HVAC topic takes the form:

```
mit/control/{MAC}/hvac
```

For example, if you want to send a 'cool_interrupt' (turn off the HVAC cooling compressor) to a Raspberry Pi with the MAC address 'd06f' for a duration of 5 minutes, you would send the following JSON message:

```
{"mac": "d06f", "command":"cool_interrupt", "duration":5}
```

to the topic

```
mit/control/d06f/hvac
```

Available commands for HVAC are:

- cool : turn on the HVAC cooling compressor
- cool_interrupt: suspend running the compressor
- heat: turn on the HVAC heater
- heat_interrupt: suspend running the heater
- fan: turn on the fan
- fan_interrupt: suspend running the fan
- idle: release any control on the HVAC system

Success / Error Topics:

You can listen to the topic:

```
mit/success/#
```

and

```
mit/error/#
```

to receive notifications when there are successful commands executed and there are errors such as if an attempted command violates one of the command rules. An example of a successful command response on the success topic looks like:

```
{ "mac": "d06f", "command": "cool_interrupt", "duration": 5, "epoch":  
1594846118 }
```

An example of a failed command and the message sent on the error topic is:

```
{"mac": "d06f", "system": "hvac", "message": "ERROR: Event  
failed rules validation: Duration of current requested event is not between 5  
and 15 minutes inclusive. {\"mac\": \"d06f\", \"command\":  
\"cool_interrupt\", \"duration\": 150}\", \"epoch\": 1594848145}
```

The message returned shows that the command failed because it violated the duration rules for HVAC curtailment. A cool_interrupt of 150 minutes is not allowed.

4.9.5. APPENDIX E: MILESTONE 4.1 DOCUMENTATION – Completed hardware construction

1) *WiFi Addressable Relay (BluFollower):*

The blufollower is designed to do multiple control/monitoring actions. First, the design monitors the existing control lines through an AC opto-isolator. Typical HVAC control lines are 24VAC though there are some proprietary systems that use different AC/DC voltages. The follower system is compatible with the most common thermostat types, 3 and 5 wire systems, which are the majority of the residential thermostats in the USA.

The 24VAC is input to an opto-isolator which is used to provide a signal through a pullup resistor to one of the digital inputs of the Raspberry PI. This provides overvoltage protection and galvanic isolation for safety. The RPI can then provide status of the residential thermostat - if it's calling for cool/heat/fan, the RPI can monitor those lines and provide that input to the server for either algorithm development or decisions on actions. For instance, if the algorithms call to prevent operation for the next half hour and the thermostat is calling for cool for 5 minutes straight, a decision may be made to accommodate occupant comfort and allow that signal to pass through.

A two level circuit has been developed for control signals. The first level is to interrupt the control signals from the existing HVAC thermostat. Once the signals have been interrupted, a second set of relays can be activated to provide a new control signal to the HVAC system calling for cool/heat/fan. The relay that interrupts the existing control signal also provides power to the relays providing a new control signal. This prevents the physical condition of the new signal being tied to the existing control signal, effectively eliminating the condition where two outputs can be tied together, which typically leads to device damage.

In addition to the monitor/control functionality, the board implements a watchdog timer. If the RPI locks up or fails for any reason, the circuit will default control to the existing thermostat.

The circuit consists of a resettable timer: a pulse every second from the RPI digital output will reset the timer. If more than a second has gone by, the power to the entire relay interrupt/control section is removed. The relays are designed so that the power off, normally closed state passes the HVAC thermostat wiring through the board without interruption.

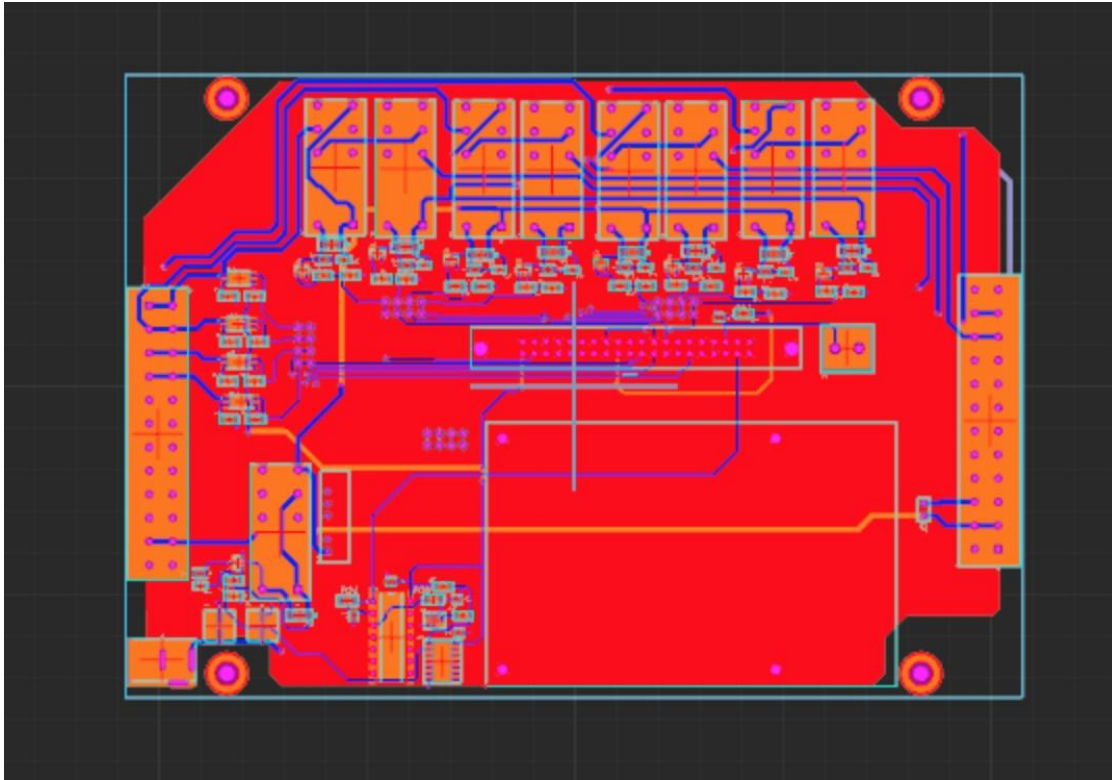


Figure 1: BluFollower PCB Layout

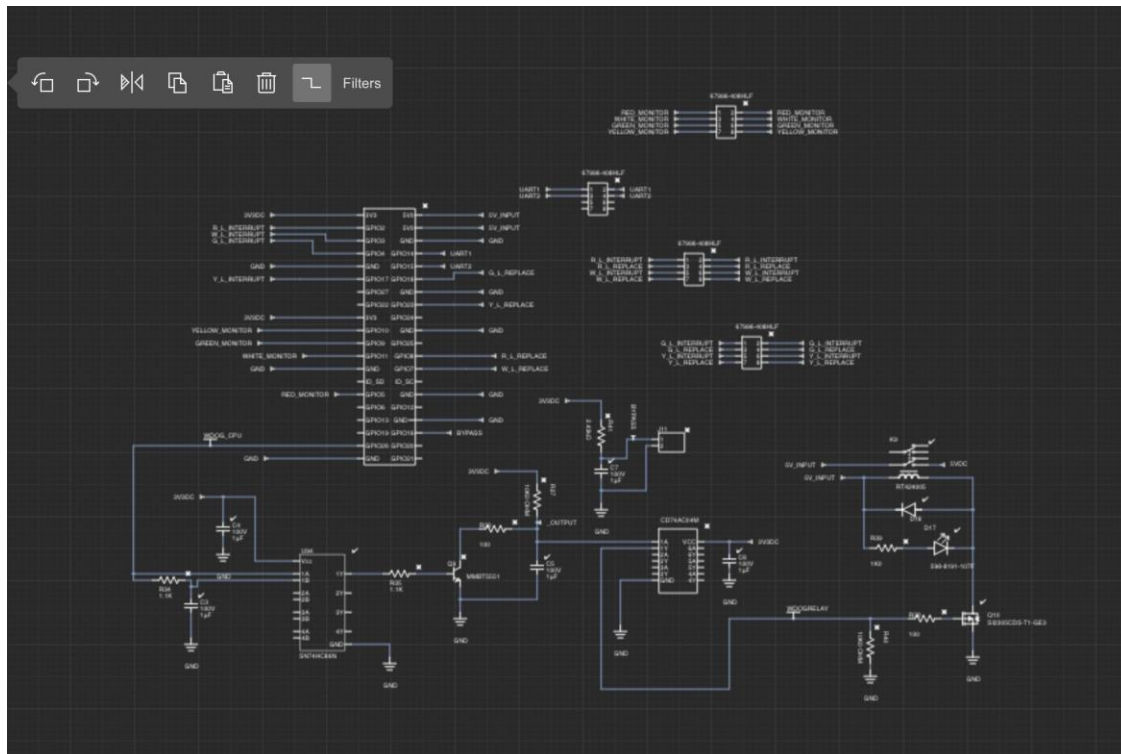


Figure 2: Portion of BluFollower Schematic

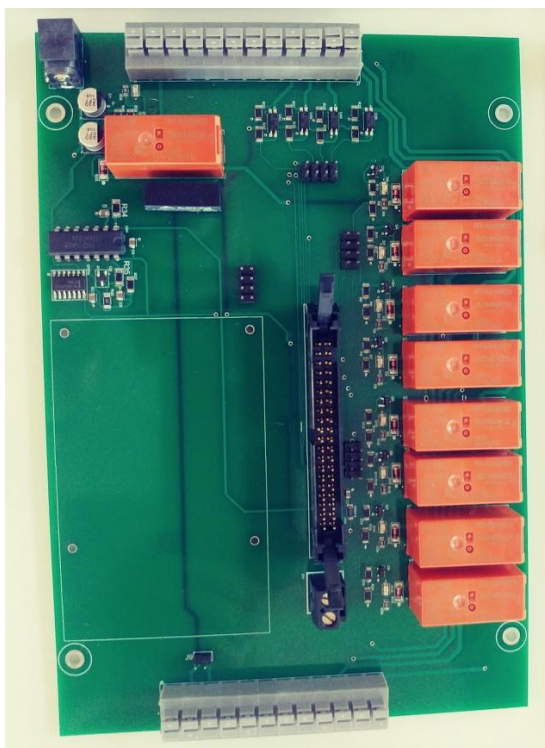


Figure 3: Photographic Evidence of PCB



Figure 4: Photographic evidence of 25 sets of WiFi Relays and single board computers

Production and Quality Control Log												
Product Name: MIT LL HVAC Controller												
ID/Count	Assembly date	Assembly Initials	QC Status	Quality Control Round 1			Quality Control Round 2			Quality Control Round 3		
				QC date	QC Initials	QC Notes	QC date	QC Initials	QC Notes	QC date	QC Initials	QC Notes
1	5/29	RAS	Passed QC	6/2	AT	Fails QC: Board is not turning on all 6 LEDs (LED 2, 3 and 5 - counted from right to left)	6/2	RAS	recheck k3 k4.	6/3	AT	Passed QC
2	5/29	RAS	Passed QC	5/29	AT	Passed QC						
3	5/29	RAS	Passed QC	5/29	AT	Passed QC						
4	5/29	RAS	Passed QC	6/2	AT	Fails QC: Yellow input stays true whether connected or not connected	6/2	RAS	recheck k3 k4	6/3	AT	Passed QC
5	5/29	RAS	Passed QC	5/29	AT	Passed QC						
6	5/29	RAS	Passed QC	5/29	AT	Passed QC						
7	5/29	RAS	Passed QC	5/29	AT	Passed QC						
8	5/29	RAS	Passed QC	6/2	AT	Fails QC: Board is not turning on all 6 LEDs (LED 2 and 4 don't turn on)	6/2	RAS	recheck k3 k4.	6/3	AT	Passed QC
9	5/29	RAS	Passed QC	6/3	AT	Fails QC: Board is not turning on all 6 LEDs (LED 6 doesn't turn on)	6/2	RAS	recheck k3 & k4, replace 1 LED	6/3	AT	Passed QC
10	06/01	RAS	Passed QC	6/1	AT	Passed QC						
11	06/01	RAS	Passed QC	6/3	AT	Fails QC: LED 8 is the only LED on, and when board is connected to Pi it causes the whole Pi computer to restart	6/2	RAS	recheck k3 k4, replace c3 u94 u101	6/3	AT	Passed QC
12	06/01	RAS	Passed QC	6/2	AT	Fails QC: Yellow input stays true whether connected or not connected. LEDs 4 and 5 do not turn on	6/2	RAS	recheck k3 k4.	6/3	AT	Passed QC
13	06/01	RAS	Passed QC	6/1	AT	Passed QC						
14	06/01	RAS	Passed QC	6/1	AT	Passed QC						
15	06/01	RAS	Passed QC	6/2	AT	Fails QC: LED 4 and 6 don't turn on	6/2	RAS	recheck k3 k4	6/3	AT	Passed QC
16	06/01	RAS	Passed QC	6/1	AT	Passed QC						
17	06/01	RAS	Passed QC	6/2	AT	Fails QC: LED 5 doesn't turn on. Red input stays true whether connected or not connected	6/2	RAS	recheck k3 k4	6/3	AT	Passed QC
18	06/01	RAS	Passed QC	6/1	AT	Passed QC						
19	06/01	RAS	Passed QC	6/1	AT	Passed QC						
20	06/01	RAS	Passed QC	6/1	AT	Passed QC						
21	06/01	RAS	Passed QC	6/3	AT	Fails QC: LED 4 doesn't turn on	6/2	RAS	recheck k3 k4, replace 1 LED, U72 shorted and removed, electro caps	6/3	AT	Passed QC
22	06/01	RAS	Passed QC	6/1	AT	Fails QC: LED 1 and 4 don't turn on	6/2	RAS	recheck k3 k4.	6/3	AT	Passed QC
23	06/01	RAS	Passed QC	6/1	AT	Passed QC						
24	06/01	RAS	Passed QC	6/1	AT	Passed QC						
25	06/01	RAS	Passed QC	6/2	AT	Missing	6/2	RAS	recheck K3 K4	6/3	AT	Passed QC
26	06/01	RAS	Passed QC	6/2	AT	Missing	6/2	RAS	recheck K3 K4	6/3	AT	Passed QC

Figure 5: Photographic evidence of final quality control document of the HVAC controller (also provided as excel file to MIT LL)

2) Single Board Computer (Raspberry Pi 3):

Pecan Street selected the Raspberry Pi 3 (RPI) as the Single Board Computer for collecting data and control of the relays for HVAC and EVSE. An RPI has been set up in the Pecan Street lab and configured as it would be in a home (refer to Figure 6). Software has been developed and tested in the lab that collects electrical circuit usage from the eGauge, indoor temperature, and EVSE status and both saves the data in a local database and broadcasts that data out to the MQTT broker on the cloud server for clients to monitor.



Figure 6: Raspberry Pi test setup in the PSI lab.

The RPI also runs a web server that is used to remotely configure settings like the mapping of circuits and a set of eGauge configuration options.

```
~ — mit@li1112-55: ~ — -zsh    ~ — pi@raspberrypi: ~ — ssh pi@10.11.10.181 ...    ~ — -zsh ...
[root@raspberrypi:/SCRIPTS/egauge_data# python3 test_1s_mqtt.py
connecting
looping
start_time: 1582827081.3038323
Connected to broker!

http://egauge6483.egaug.es/cgi-bin/egauge?inst
Results:
air1: 72
air2: 71
car1: 2
grid: -1177
solar: 5266
{"epoch": 1582827082, "air1": "72", "air2": "71", "car1": "2", "grid": "-1177", "solar": "5266"}
```

Figure 7: Communication validation between the RPI and electricity data. Communication with BluFollower and EVSE controller will occur once hardware components are completed.



Figure 8: Photographic evidence of 25 single board computers, Raspberry Pi 3s

3) EVSE Controller:

Pecan Street's EVSE controller intercepts the charging between the vehicle and the home charging station in order to manage the vehicle's charging. As a result, the EVSE controller treats the vehicle as a load that is capable of providing DR services. The controller's input is connected to the charger's pilot bus, while the output is connected to the vehicle. Based on the signals received via WiFi from the broker, the EVSE controller can change the charging current by modifying its Pulse Width Modulation (PWM) signal, compliant with J1772.

Following the hardware design and pilot board construction for the EVSE, the technology was evaluated and validated in the lab. After successful tests on the software development, Pecan Street constructed five additional boards to be deployed in the field. Due to COVID-19 restrictions, two of these boards were deployed in the lab and in one residential setting, while the remaining three were installed, but not connected to a load, to be used for testing commands and communication. All hardware has been ordered, fabricated, and is available at the lab for assembly. Therefore, MIT LL, ARPA-E, and PSI decided not to move forward with the override functionality.

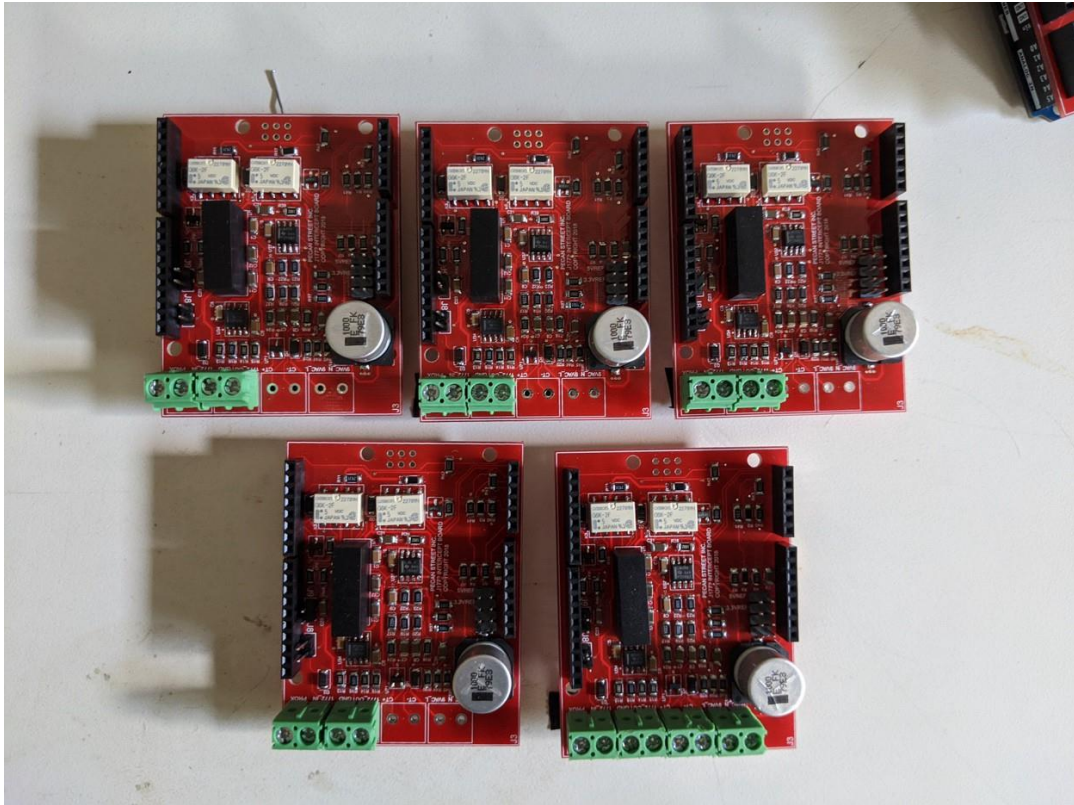


Figure 9: Photographic evidence of five sets of EVSE boards



Figure 10: Photographic evidence of five sets of fully assembled EVSE Controller board connected to J1772 compatible connector

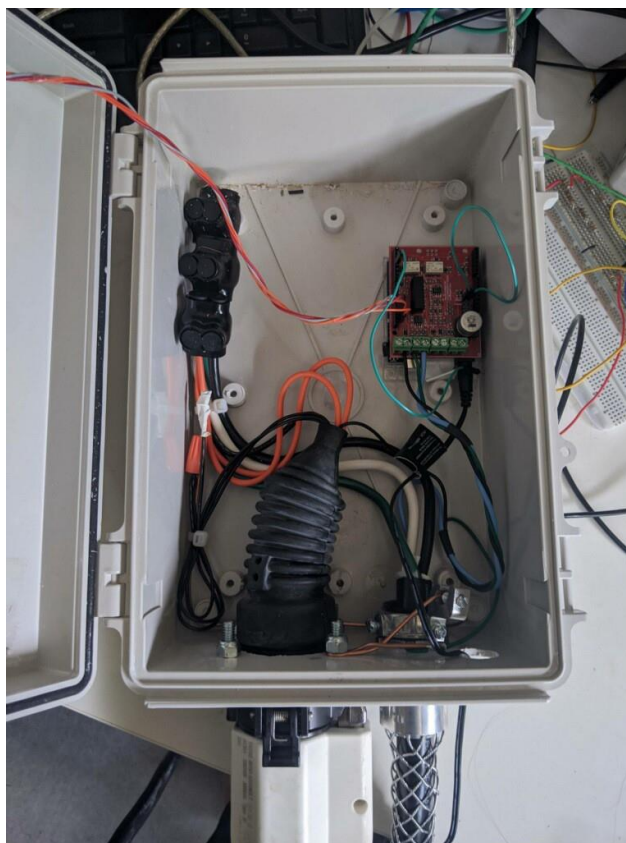
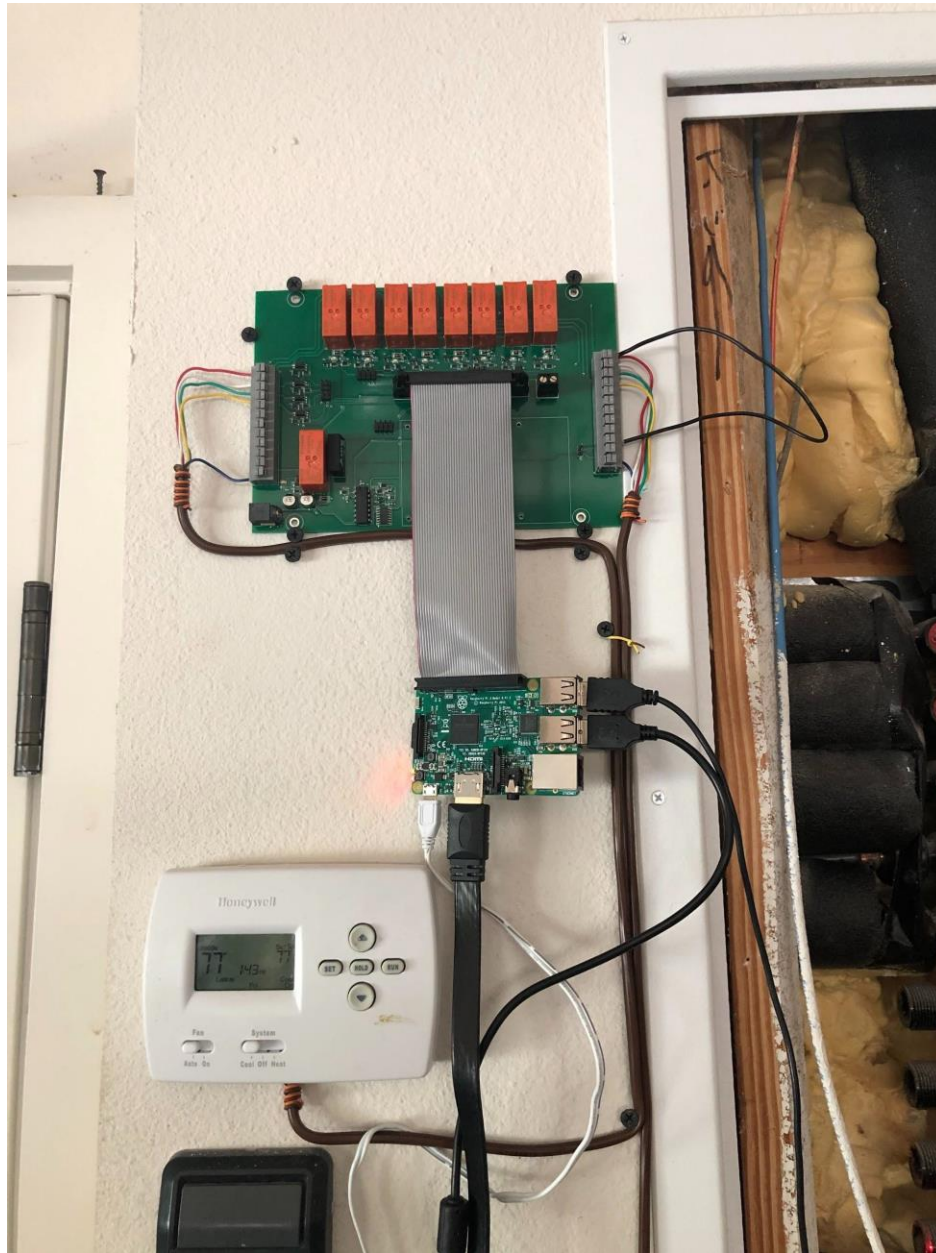


Figure 11: Residential vehicle charger showing conductors to EVSE controller

Product Name: MIT LL EVSE Controller												
				Quality Control Round 1			Quality Control Round 2			Quality Control Round 3		
Count	Assembly date	Assembly Initials	QC Status	QC date	QC Initials	QC Notes	QC date	QC Initials	QC Notes	QC date	QC Initials	QC Notes
1	5/29/2020	AT	Passed QC	6/1	SH	QC completed with resistor test set.	6/2/2020	AT	QC completed with resistor test set.	6/10	AT	Working code (tested with car) Received MQTT messages Passed QC
2	5/29/2020	AT	Passed QC	6/1	SH	QC completed with resistor test set.	6/2/2020	AT	QC completed with resistor test set.	6/10	AT	Working code (tested with car) Received MQTT messages Passed QC
3	5/29/2020	AT	Passed QC	6/1	SH	QC completed with resistor test set.	6/2/2020	AT	QC completed with resistor test set.	6/10	AT	Working code (tested with car) Received MQTT messages Passed QC
4	5/29/2020	AT	Passed QC	6/1	SH	QC completed with resistor test set.	6/2/2020	AT	QC completed with resistor test set.	6/10	AT	Working code (tested with car) Received MQTT messages Passed QC
5	5/29/2020	AT	Passed QC	6/1	SH	Failed QC	6/2/2020	AT	QC completed with resistor test set.	6/10	AT	Working code (tested with car) Received MQTT messages Passed QC

Figure 12: Photographic evidence of final quality control document of the EVSE controller

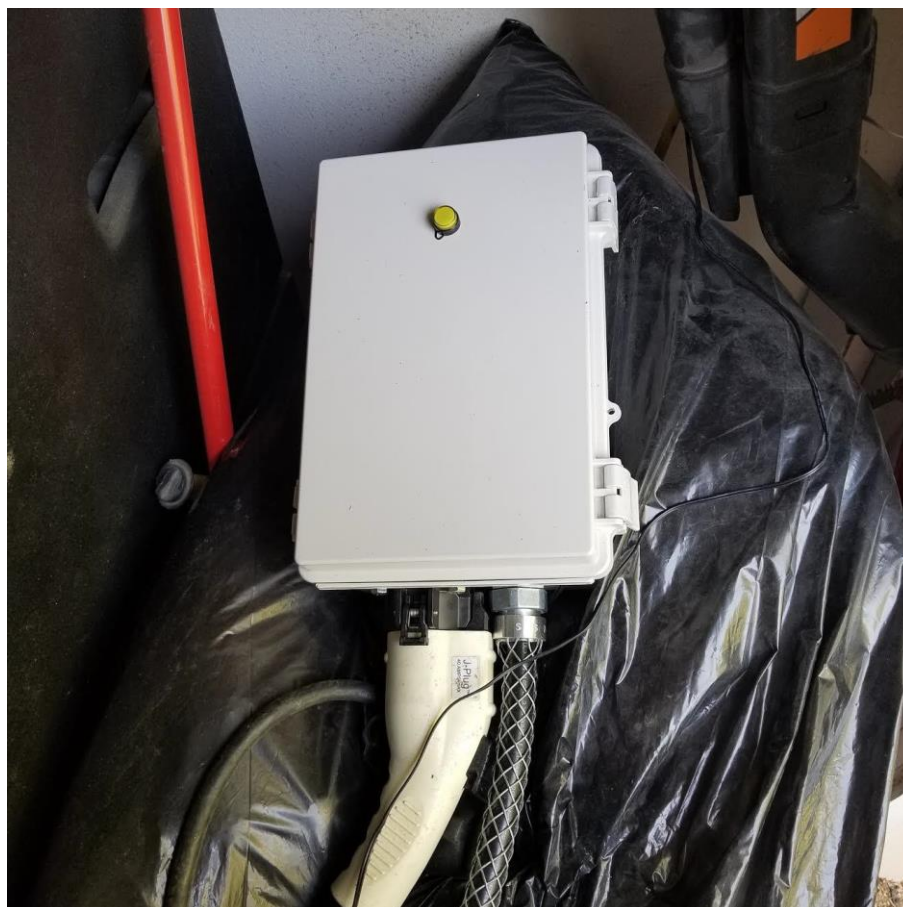
4.9.6. APPENDIX F: MILESTONE 4.2 DOCUMENTATION - Completed hardware installation



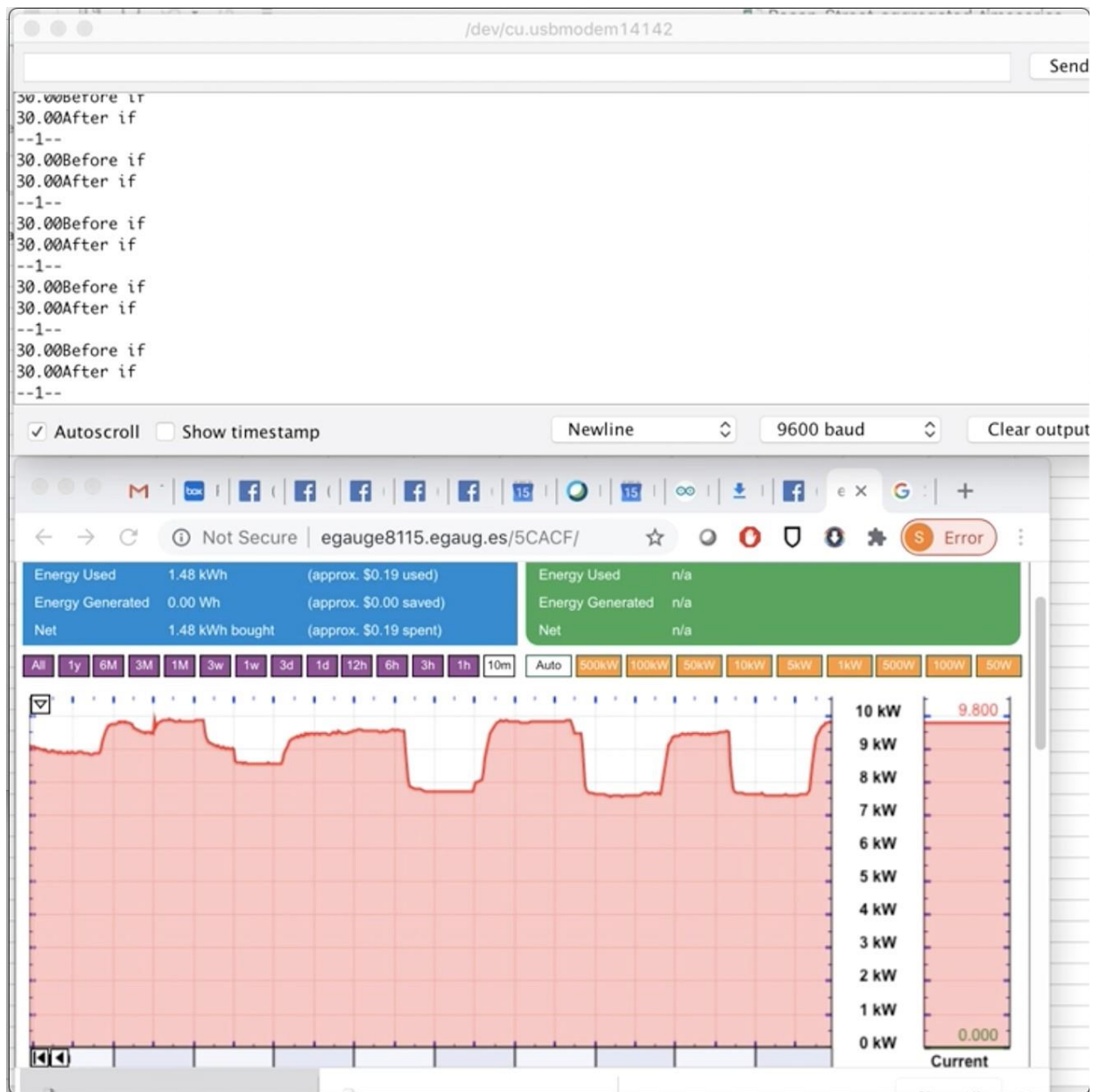
Load 1 in the PSI lab - connected to the HVAC load



Load 2 in the PSI lab - connected to the EV load



Load 3 in a residential setting - connected to the EV load



Data verifying simple on/off control (adjustment of the charging rate) for the EV in a residential setting (load 3) through the API from PSI cloud resources. Please watch the full video to view the EV adjust its charging rate in real time [here](#).



Photographic evidence of 24 HVAC and 3 EVSE follower boards connected in the PSI lab for communication testing, using data from PSI's participant network.

8.7. APPENDIX G: MILESTONE 4.2 DOCUMENTATION CONTD.

Data IDs

The list of Data IDs in the following spreadsheet are all homes in the Mueller neighborhood in Austin, TX on the same transformer phase. Historic data for these homes can be found in the home directory in the 1s_data directory. Temperature data for thirteen of the homes is available, and also available on the project server in the same directory.

These Data IDs are also mapped to the HVAC and EVSE followers in the lab that are being used for communication validation, and not connected to actual loads.

Pecan Street Inc. - MIT SECURED Project 2019-2020							
HVAC & EVSE Followers: With and Without Loads							
Count	Load	Type	Location	Data ID	Follower MAC Address	Indoor Temp Data Available?	Temp Sensor MAC Address
1	With load	EVSE	GARAGE	2695	97bd	N/A	
2	With load	EVSE	HOME	6498	0cf8	N/A	
3	With load	HVAC	GARAGE	2695	d06f	Yes	d06f
1	Without load	HVAC	LAB	9971	8b8a	Yes	600194755FB4
2	Without load	HVAC	LAB	2925	12e9	Yes	84F3EBB3344D
3	Without load	HVAC	LAB	974	97bd	Yes	84F3EBB3360C
4	Without load	HVAC	LAB	6558	31af	Yes	84F3EB8E5617
5	Without load	HVAC	LAB	171	f52c	Yes	84F3EB8E527E
6	Without load	HVAC	LAB	7541	600c	Yes	84F3EB8E5525
7	Without load	HVAC	LAB	1283	18a5	Yes	84F3EB8E5773
8	Without load	HVAC	LAB	3538	4e0f	Yes	84F3EB8E533C
9	Without load	HVAC	LAB	2094	f03e	Yes	5CCF7FF1A181
10	Without load	HVAC	LAB	3310	63fe	Yes	84F3EB8E581F
11, 12	Without load	HVAC & EVSE	LAB	1642	f613	Yes	5CCF7FF1A1B2
13	Without load	HVAC	LAB	2818	bc1e	Yes	84F3EBB21F62
14	Without load	HVAC	LAB	3719	90b9	Yes	84F3EB8E5355
15	Without load	HVAC	LAB	2442	5156	No	
16, 17	Without load	HVAC & EVSE	LAB	7850	78fc	No	
18	Without load	HVAC	LAB	661	de7a	No	
19,20	Without load	HVAC & EVSE	LAB	1169	17fb	No	
21	Without load	HVAC	LAB	3849	5029	No	
22	Without load	HVAC	LAB	7429	eed6	No	
23	Without load	HVAC	LAB	153	95be	No	
24	Without load	HVAC	LAB	3506	6462	No	
25	Without load	HVAC	LAB	8995	b75d	No	
26	Without load	HVAC	LAB	1354	79b3	No	
27	Without load	HVAC	LAB	7660	cf83	No	
28	Without load	HVAC	LAB	1103	37b4	No	

