Microgrid Demand Response: A Comparison of Simulated and Real Results

Luis Fernando Enriquez-Contreras, A S M Jahid Hasan, Jubair Yusuf, Jacqueline Garrido, Sadrul Ula
College of Engineering - Center for Environmental Research and Technology
University of California, Riverside
Riverside, California, U.S.A.
lenri001@ucr.edu, ahasa006@ucr.edu , jyusu001@ucr.edu , jgarr023@ucr.edu, sula@cert.ucr.edu

***Abstract*—Microgrids can be used in demand response (DR) and islanding operations. This paper explores a real-world microgrid implementation performed at the University of California, Riverside. The intended purpose is to see the feasibility of demand response and islanding in a real microgrid. An algorithm is used to generate "target data" for the Battery Energy Storage (BES) Inverter to follow. The other purpose is to see the accuracy when transitioning from a simulated model to a real-life implementation. The results from this implementation show it is feasible to respond to demand response scenarios with the current setup. However, more research is needed to successfully implement islanding responses due to infrequent power spikes that briefly draw more power than the microgrid can produce.**

***Index Terms*—demand response, real-world application, microgrids, islanding**

I. Introduction

According to the U.S. Department of Energy (DOE), 575 microgrid systems in the U.S. are registered in its database [1].As microgrids become commonplace, sophisticated control algorithms are being developed to address the increasing demand for microgrid control systems. The applications of microgrids are almost unfathomable in the possibilities it opens to the electrical grid, communities, and the individual. One major problem is the intermittency of renewable energy sources such as solar and wind power. Energy storage solutions like the BES address by storing and releasing power at desired times. The electrical grid and power companies benefit from the increased load-shedding capabilities a BES microgrid can offer during critical peak hours since microgrids can be programmed to be self-reliant on power during critical times. Microgrids also benefit communities and individuals, since it catalyzes the possibility for them to mitigate their reliance on third-party energy providers, or in some cases, obtain energy independence altogether. Microgrids can assist communities when the power grid fails, as it can run as an isolated system. Real-world microgrid implementation is an area of research where simulation models are improved upon and implemented. The transition from a microgrid simulated model to a full-scale implementation requires knowledge of modeling, programming, and power configuration. This task's complexity makes it very difficult for a communication, programming, or power engineering group to execute solely. [2] is an example of microgrid that implemented demand response, that managed to island its microgrid from 11:00 to 13:00. Many microgrids have components from different manufacturers that are not designed to operate in synchrony as is the case for the microgrid in this paper. The components have no complex control system and only have simple register control. The goal is to see the accuracy of simulated microgrid models when applied to a real system.

This validation's main contribution is documenting the difficulties and challenges involved in transitioning from a simulated model to real-life implementation. This paper focuses on the problems and outcomes of the microgrid's full-size components and describes how software is implemented in a full-scale microgrid. The validation results are compared to the simulated model to analyze the differences and similarities between simulated and real-life data. The code for the simulated model and the modifications needed for real-life implementation are further discussed in this paper.

A. Literature Review

[3] is a real-word demonstration of a DC microgrid that combines solar and BES to power a load bank in rugged rural setting. [4] and [5] demonstrate the importance of focusing on the power perspective of a microgrid, and not only the energy perspective, since paper [4] saw a strong correlation between self-consumption rates (SCR) and the power capacity of a BES instead of energy capacity and [5] is advocating for an adequate BES policy, rather than net metering. Both papers [4] [5] implement real microgrids, yet their main focus is not on-demand response but rather on SCR. [6] implements a full-scale microgrid and utilizes it for demand response. However, it does not focus on the software aspect of the implementation, nor does it compare the noise in real-life results to the simulated model seen in Hardware in the Loop (HIL) models [7] [8]. In [7], a HIL microgrid involving a three-phase synchronous generator is tested and compared to purely simulated model. In [8], a BES and diesel generator are optimized during every interval in a HIL microgrid. Both papers compare the results of simulated and real results through visual graphics, but [8] also uses a power violation table to describe the amount of energy, the amount of power, and the duration of instances where power was consumed from the main grid.

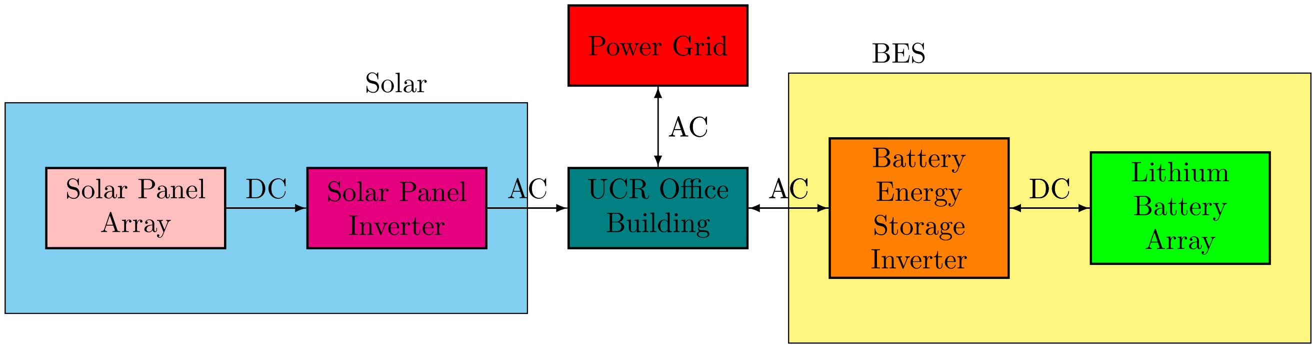


Fig. 1: Power System Setup for Office Building Microgrid

II. Methodology

A. Purpose

The primary purpose of this work is to execute and analyze our algorithm in a real-life microgrid and demonstrate the steps needed for actual implementation. The validation uses previously recorded data from the day before to create an optimized model. .This model produces an optimal load for the microgrid given the initial conditions. Load in [9] can be positive or negative, meaning that the microgrid system can consume or produce power. In other words, the microgrid can have a negative load by having excess power or a positive load by having a deficit of power which would be the typical "standard power consumption" of a building. The power the BES inverter needs to produce or consume in order to obtain an optimal demand response is calculated by subtracting the previous day's net load from the algorithm's optimized load produced. The microgrid attempts to implement that load in real life using this desired power data. The microgrid data is optimized for a single day in 15-minute intervals throughout the day. During each interval, the dataset assigns a value that the power output of the microgrid should reach. The microcontroller's only control for this validation is the ability to lower and raise the power in the solar and BES inverter. The validation runs for one day with a mostly hands-off approach letting the algorithm decide the power delivered or consumed from the inverters. The load can vary significantly due to the intermittent loads of the HVACs and level 2 EV chargers. The algorithm does not control the microgrid directly; a subprogram analyzes the data and executes the control. The algorithm is running atop another software and has no lower-level control.

For example, suppose the data set gives a value of 50 kilowatts at 10:15 AM. In that case, the microgrid controller must provide that power from the solar panels and the BES. However, real-life conditions can inhibit microgrids from reaching those goals, such as lack of sufficient power capacity of the microgrid, variability in the real-life load, and the malfunction of microgrid components hampering operations. Nevertheless, the algorithm is a common approach in multiple academic papers on microgrid simulations. This was intended to be as hands-off as possible and addresses problems faced when porting a simulated microgrid algorithm to real-life implementation.

Vehicle to Grid (V2G) allows a Battery Electric Vehicle (BEV) to power a building using it batteries as a BES. Paper [10] managed to implement V2G in this microgrid, but the control for that experiment was completely manual. V2G is usually bidirectional, so the V2G "charger" and the BEV are more akin to a BES than a regular EV charger. While this validation only has regular and unrestricted level 2 EV charging, the behavior and characteristics of the BES and building will be used in future experiments to implement an automated V2G setup.

B. Physical Setup

This work implements demand response at the University of California, Riverside microgrid. Our setup consists of a 100 kVA solar inverter, a 500 kWh BES, and an inverter to charge and discharge the BES, as seen in Figure 1. The admin building's load is considered the "regular load" for this microgrid, and is the main office space for the graduate researchers and staff at College of Engineering, Center for Environmental Research & Technology, University of California, Riverside (CE-CERT). Figure 1 illustrates the power connections in the microgrid, and the admin building is the main connection for all the microgrid components and the only part that consumes power. Power from the power grid, solar, and battery inverter is connected to the main breaker panel. The solar inverter is uni-directional for it can only deliver power to the building, while the power grid and BES inverter are bidirectional since they both can supply the building with or consume power. The battery trailer consumes power by charging the batteries. The power grid can "consume" excess solar, but this is not recommended since it leads to high demand charges and low utilization rates of the solar power produced. The setup is to supply energy and reduce the demand for the 20,000-square-foot admin building. The communication layer consists of a GNU/Linux microcontroller that uses Python to instruct the BES and solar inverter, as seen in Figure 2. The controller uses Modbus RS485 to instruct the BES inverter and uses

Modbus TCP to instruct the solar inverter. The data gathered is then sent to the laboratory server for storage. The goal for this validation is to implement the simulation model previously published in [9].

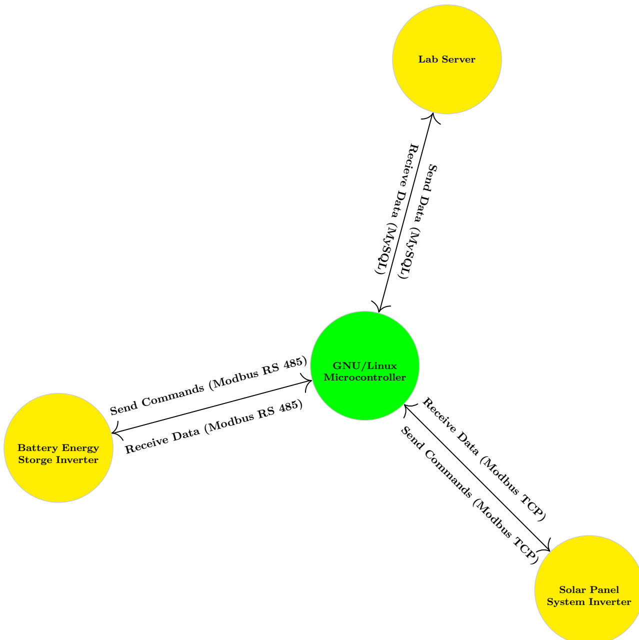


Fig. 2: Communication System Setup for Office Building Microgrid

C. Algorithm Setup

The algorithm is designed to minimize the demand charges of $E^{Battery}$ (E^B), P^{Grid} (P^G)considering $P^{Battery}$ (P^B), P^{Solar} (P^S), and P^{Load} (P^L). This optimization is described in Algorithm 1 and discussed further in detail by the authors in [9]. The rate structure is a flat energy charge with a time independent monthly demand charge.

D. Software Setup

The software's primary goal is to interpret data from the algorithm and implement it in the microgrid. This is done in two steps: first, analyze the data, determine which steps and procedures are needed; second, implement those procedures via register control. The first task is a generic workflow of basic behaviors of the microgrid components. Since the microgrid only has two controllable components, the battery inverter and the solar inverter, the system is limited in actual actions it can implement with no mechanism currently in place to control the load. The limited actions include: curtailing, which lowers the amount of power produced or increases the amount of power produced from the solar inverters; BES discharging, which provides three-phase power to the building; or BES charging, which uses three-phase power from either the solar inverter or the grid to charge the batteries. The action needed for both components depends on the value and the state of charge available in the battery and the power capacity

Algorithm 1: Optimize Net Load

Read solar values P^S for electric meter
Get net load P^G
Convert to 15 minute data
Add P^G and P_S to get P^L
$P_t^B \geq 0$ = Charging
$P_t^B < 0$ = Discharging
Optimize P^G for P^B, P^S, and P^L

$$\min_{P^G, P^B, P^S, P^L} \quad f(P^G) \tag{1}$$

subject to

$$E_{t+1}^B = E_t^B + P_t^B \Delta t \tag{2}$$

$$E^{B_{min}} \leq E_t^B \leq E^{B_{max}} \tag{3}$$

$$-100 \leq P^B \leq 100 \tag{4}$$

$$P^L = P^G + P^S + P^B \tag{5}$$

of the solar inverter. The solar inverter data is determined by prior data since there is no system currently in the setup to measure solar capacity directly. The meters only measure the power outputs of the inverters and not their power capacity. The software controls power output, so while a meter may measure 50 kW, the solar inverter may have the capacity to produce 100 kW since the solar inverter can be curtailed to 50 kW. The opposite is also true; a command may be sent to the solar inverter for 90 kW, but if the power capacity is only 60 kW, the solar inverter will only output 60 kW. The power capacity for the solar inverter depends on solar irradiance. The real capacity of the BES mainly depends on its energy capacity, and to a lesser degree, on its power capacity. The microcontroller prioritizes solar power first and then uses the BES if needed; the reduction of the peaks is the ultimate goal, but having the battery for power production during the night is also necessary.

The state machine for the control code of the BES inverter is seen in Figure 3, and the code layout is described in Algorithm 2. After the algorithm generates the dataset, it is input as a dataframe. The code determines if the value for power is within the appropriate limits for this inverter; if it is, the inverter is set to that value and records data from the inverter while it waits for a new time interval to repeat the process. It should be noted that negative values charge the batteries for the BES inverter, and positive values power the microgrid, which is opposite to the convention in the algorithm.After the specific component action is decided, the microcontroller unit (MCU) sends the actual command to the components. This involves actual register control of the components. Modbus TCP is used to control the solar inverter and Modbus RS-485 controls the battery inverter. Therefore, the MCU must know the I.P. and RS-485 addresses of all the components assigned to the local intranet system. The MCU communicates with the solar inverter via our local intranet while it controls the

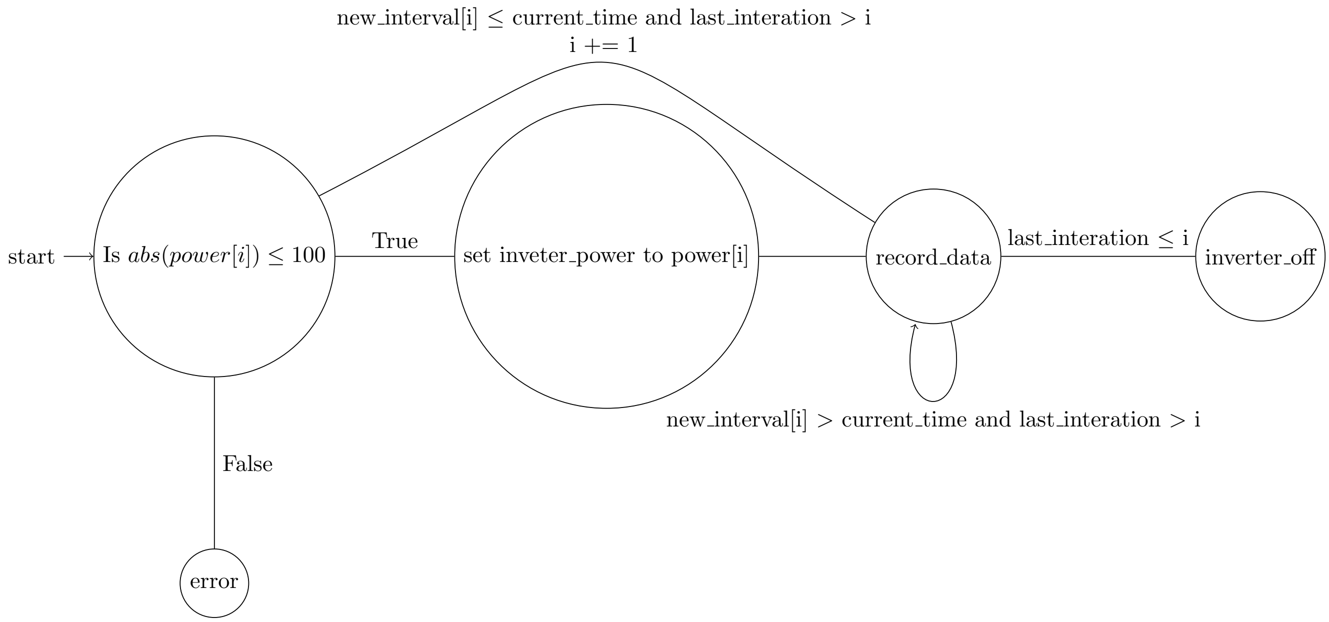


Fig. 3: Battery Inverter State Machine

BES inverter via RS 485 Modbus protocol. It comes equipped with COM ports and Wi-Fi to do all the controls. Python is used as its register control and is where the component actions are executed. It also runs on a GNU/Linux O.S., which helps in the goal of maintaining our system's independence from proprietary software.

Algorithm 2: Microgrid Control Software

```
df ← dataset
output ← []
count ← 0
delay ← 15 minutes
new_interval ← current_time
while data_length > count do
    inverter_power ← df.power[count]
    while new_interval > current_time do
        output ← inverter_get_data
    end
    count += 1
    new_interval += delay
end
```

III. Validation

The first step in implementing the microgrid validation is to gather the previous day's data and run the optimization algorithm. The behavior of the solar and building load data does not vary significantly from one day to another, so the prior day's data is used with the assumption that the next day's behavior will be similar. Riverside's weather is extremely dry during June, and the sky was sunny throughout the entire week of the experiment. Data from the building and solar inverter creates a target netload for the building. Next, the previous day's netload is subtracted from the solar data to get the building load; this is added to the target netload to obtain the target power output of the inverter in 15-minute intervals, as seen in Figure 4. The solar inverter, building load, and the net load are the parameters for the optimization, and output is the BES inverter which is the behavior the inverter should follow to produce the desired optimized load.

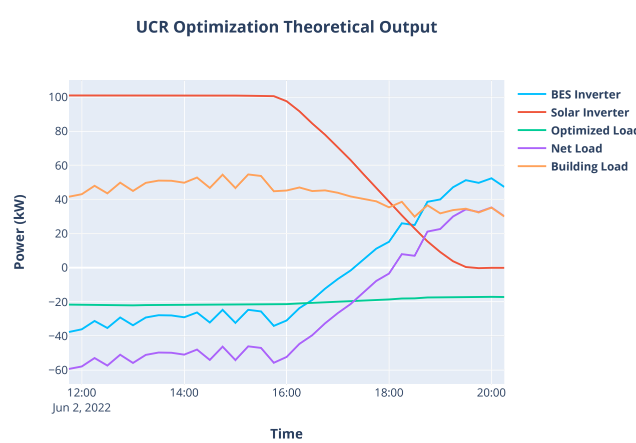
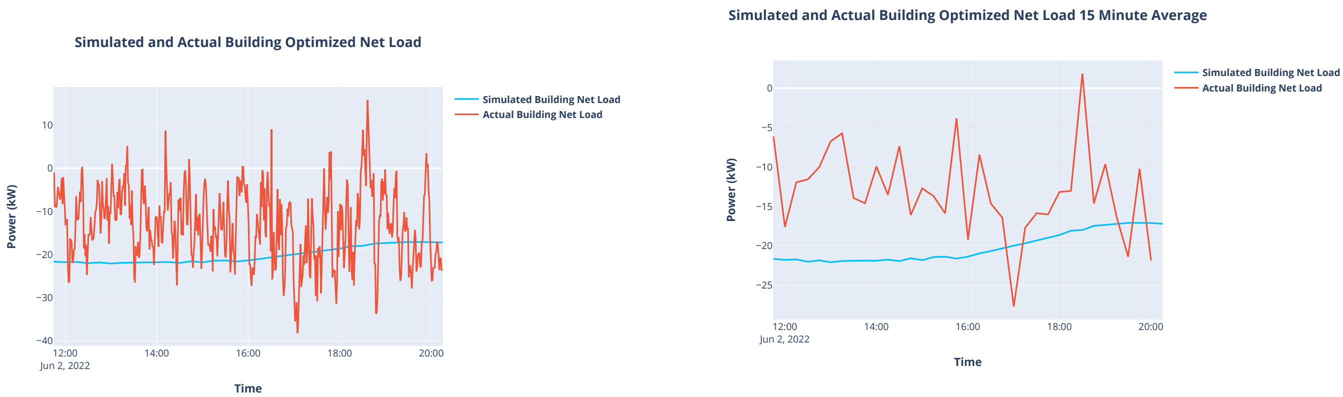


Fig. 4: UCR Microgrid Building Algorithm Validation

The BES inverter parameter in Figure 4 commands the software mentioned previously to manage the inverter throughout the validation. The validation began at 11:45 and ended at 20:15, lasting 8 hours and 30 minutes. The main goal was to



(a) UCR Microgrid Building Theoretical and Actual Load Data

(b) UCR Microgrid Building Theoretical and Actual Load 15 Minute Average Resample

Fig. 5: UCR Microgrid Building Theoretical and Actual Load

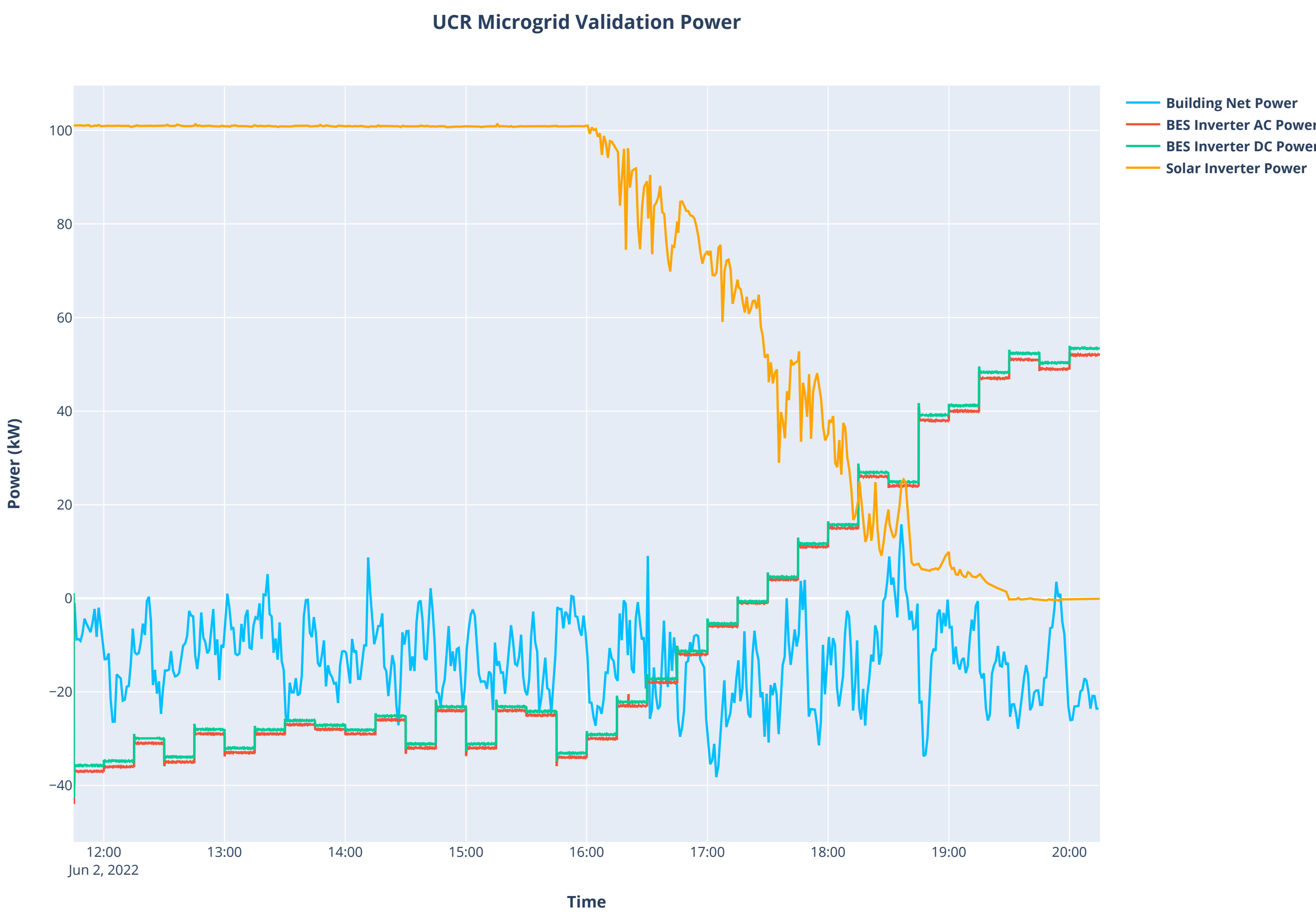


Fig. 6: UCR Microgrid Inverter, Solar, and Building Power

see how the BES can achieve near-zero import/export from the grid. From 11:45 to 17:15, the BES inverter utilized the excess solar power to charge the batteries; after 17:30, the BES inverter started to discharge the batteries and send power to the building to compensate for the declining solar power output. This transition started from a gradual decline in charging that compensated for the decreasing amount of excess solar power to discharging when there was no more excess solar power and, later no solar power. Besides the BES being controlled by the algorithm, everything else in this microgrid was allowed to operate under normal conditions. The HVAC units ran as usual, E.V. chargers were utilized throughout the day, and it was a normal Thursday for researchers and staff at this university building. While in past experiments, this research group could control and curtail the solar inverter [11], this was not necessary for this validation since the algorithm required the most excess solar possible, so the solar inverter operated as usual with no curtailment or control from the algorithm.

IV. Results

Transitioning from a simulated optimization model to a real-world implementation introduces the variability of real-world loads and renewable energy production. While the algorithm predicted a smooth curve net-load at around -20 kW for the building, the real output for the straddled nearer to zero and even exceeded it a couple of times, as seen in Figures 5a and 5b. It should be noted that the inverter was remarkably accurate in following the algorithm data commands other than a jump at the beginning of the inverter operation. Nevertheless, the degree of success in this basic validation is promising; throughout the validation, the 15-minute intervals of data barely exceeded three to four kilowatts, significantly less than the $\approx$ 25 kW during previous days. Running this algorithm for an entire month can significantly lower the demand charges.

Given that this algorithm was designed for demand response and reducing demand charges for microgrid facilities, this was close to the intended result. However, if islanding is desired, this approach, as done in prior validations, would not be achievable since the microgrid's netload exceeded zero kW on multiple occasions, which would not meet the load balancing requirements needed to run in islanded mode. Figure 5a shows high-resolution data that has numerous occasions of power failure from the microgrid perspective. Figure 5b averages the data in a 15-minute interval, which is the perspective of an electric utility company. In this case, the power exceeded zero only one time, and it was only $\approx$ 2kW. Short duration power spikes do not have a tremendous effect on demand response applications compared to islanding applications. Improvements to the microgrid setup and software are needed before reaching this degree of success.

In Figure 6, the inverter is seen as 15-minute "steps." Every 15 minutes, the value is changed to either a higher or lower value, dependent on the needs of the algorithm. At first, the inverter is only charging, but as solar power decreases, the charge rate decreases. This slowly transitions from charging, to slow charging, to slow discharging, to fully discharging. This successfully mitigates the sudden peaks that usually occur during a brief period.

V. Conclusion

The results from this short validation demonstrate the possibility lowering the max demand of the net load, thus reducing the facility's demand costs, by utilizing a BES system in combination with solar power. The algorithm proposed and implemented achieved the goal of lowering the max demand, thereby reducing the stress on the grid. However, further improvements need to be made to increase demand response accuracy and reliability. Load prediction is needed to optimize data for the algorithm more accurately; optimization using shorter time intervals is also needed to reduce possible power surges. This approach is unsuitable for islanding since the net load would exceed zero on multiple occasions, and instantaneous load matching is needed for islanding. Future improvements on this microgrid system will integrate load control, and a faster reaction time to the system that should enable full capability of automated islanding.

References

[1] "U.S.Department of Energy Combined Heat and Powerand Microgrid Installation Databases," https://doe.icfwebservices.com/microgrid, accessed: 2010-06-26.

[2] M. Shahidehpour, Z. Li, W. Gong, S. Bahramirad, and M. Lopata, "A hybrid ac\/dc nanogrid: The keating hall installation at the illinois institute of technology." *IEEE Electrification Magazine*, vol. 5, no. 2, pp. 36–46, 2017.

[3] P. Torres, T. Costa, L. Araújo, J. Filho, S. Williamson, and W. Macêdo, "Solar photovoltaic-based dc microgrid testing under real-world operating conditions," in *The 10th International Conference on Power Electronics, Machines and Drives (PEMD 2020)*, vol. 2020, 2020, pp. 431–436.

[4] M. Khemir, M. Rojas, R. Popova, T. Feizi, J. F. Heinekamp, and K. Strunz, "Real-world application of sustainable mobility in urban microgrids: Reduction of emissions and operational experience from germany," *IEEE Transactions on Industry Applications*, 2021.

[5] K. P. Satsangi, D. B. Das, G. S. Babu, and A. Saxena, "Real time performance of solar photovoltaic microgrid in india focusing on self-consumption in institutional buildings," *Energy for Sustainable Development*, vol. 52, pp. 40–51, 2019.

[6] G. Gust, T. Brandt, S. Mashayekh, M. Heleno, N. DeForest, M. Stadler, and D. Neumann, "Strategies for microgrid operation under real-world conditions," *European Journal of Operational Research*, vol. 292, no. 1, pp. 339–352, 2021.

[7] E. Fonkwe, J. Kirtley, M. Almeida, and D. Medjo, "Model identification of dynamic microgrids and controller optimization with high fidelity hardware-in-the-loop platform," in *2016 IEEE 17th Workshop on Control and Modeling for Power Electronics (COMPEL)*. IEEE, 2016, pp. 1–8.

[8] C. M. Rangel, D. Mascarella, and G. Joos, "Real-time implementation & evaluation of grid-connected microgrid energy management systems," in *2016 IEEE Electrical Power and Energy Conference (EPEC)*. IEEE, 2016, pp. 1–6.

[9] A. J. Hasan, L. F. Enriquez-Contreras, J. Yusuf, and S. Ula, "A comprehensive building load optimization method from utility rate structure perspective with renewables and energy storage," in *2021 International Conference on Smart Energy Systems and Technologies (SEST)*. IEEE, 2021, pp. 1–6.

[10] A. S. M. J. Hasan, L. F. Enriquez-Contreras, J. Yusuf, M. J. Barth, and S. Ula, "Demonstration of microgrid resiliency with v2g operation," in *2021 IEEE Transportation Electrification Conference & Expo (ITEC)*, 2021, pp. 243–248.

[11] R. B. Faruque, M. F. Scudder, and S. Ula, "Real-world implementation of advanced pv curtailment and reactive power control using non-smart pv inverter: A case study," in *2020 IEEE Power & Energy Society General Meeting (PESGM)*. IEEE, 2020, pp. 1–5.