Comparison of Model Predictive Control and Robust Optimization for the Optimal Power Flow Problem

May 9, 2024

Carla Becker, Hannah Davalos, Jean-Luc Lupien, John Schafer, Keyi Kang Yao

Abstract

This project investigates the multi-period Optimal Power Flow problem in a microgrid setting, solved with both robust optimization and model predictive control. Using the IEEE 39-bus test system, we use real-world solar generation, wind generation, and building load data to model power generation and consumption dynamics in an idealized microgrid. Outfitted with battery storage at all renewable energy generation sites, we seek to maximize the use of generated renewable energy and simultaneously minimize the need for diesel generation to close any gap between generation and demand. This report outlines our process to obtain real-world data and load it into the test bed, our struggles with, and eventual solution to, handling a non-radial grid, and a comparison of our ultimate solution methods –robust optimization and model predictive control– using full-information optimization as a baseline. We specifically compare two rather different days from our dataset: one with the prototypical, Bell-curve-like solar generation that peaks at midday and a second with sporadic solar generation that likely resulted from a stormy or cloudy day. Ultimately, we find that robust optimization, while safe, is too conservative, and that model predictive control obtains similar performance to full-information optimization, while ensuring stability.

1 Introduction

1.1 Motivation and Background

The increasing reliance on renewable energy sources such as wind and solar power is critical for achieving sustainability in energy production [1, 2]. However, the large-scale integration of renewable generation into existing power grids presents significant challenges due to their inherent intermittency [3]. Unlike conventional power sources, which can be controlled to match supply with demand, renewable outputs are highly dependent on environmental conditions, leading to potential supply-demand imbalances and grid instability. To address these challenges, optimization techniques are pivotal, allowing for dynamic adjustment of grid operations [4]. More specifically, solving the optimal power flow (OPF) problem in real-time is crucial to the successful exploitation of renewable energy.

This project explores various optimization strategies aimed at minimizing reliance on conventional diesel generation while satisfying power flow constraints. We consider the multi-period OPF over a 24-hour period, using a model based on the IEEE 39-bus test system with added wind, solar,

and battery storage assets. Real-world data will inform a comprehensive comparative analysis of two approaches: 1) robust optimization, which ensures grid security by preparing for worst-case scenarios,but which may lead to under utilization of renewable resources and 2) model predictive control (MPC), which uses predictive models to forecast future renewable generation and load demand over a time horizon, allowing for proactive adjustments. As a baseline for comparison, we perform full-information optimization which assumes perfect knowledge of future conditions, illustrating potential gains from highly accurate forecasting. This research aims to evaluate these models to determine their efficacy in real-world settings, guiding future strategies for grid management in an era increasingly dominated by renewable energy, enhancing grid stability, and maximizing the utilization of renewable resources.

1.2 Relevant Literature

Multiple methods for solving the optimal power flow (OPF) problem in real-time exist, including online convex optimization [5], distributed feedback control [6], feedback optimization [7], deep reinforcement learning [8], supervised learning [9], robust optimization [10], and model predictive control [11].

In online convex optimization, an agent makes sequential decisions and suffers an environment-determined loss. Under certain assumptions, online algorithms are guaranteed to be indistinguishable from the optimal decision sequence over long time periods. The advantage of these methods are that they have mathematically proven performance guarantees. The disadvantage is the requirement of an inversion step. More traditional control frameworks, like the distributed feedback controller, perform linearization of the power flow equations.

However, this project will fundamentally rely on a robust optimization formulation to cope with the uncertainties of solar and wind power production. In [12] slack variables are introduced that allow the variability of solar resources to not completely halt the grid system. In [13], the authors formulate an OPF that uses a Gaussian probability density function to describe the uncertainty of Renewable Energy Sources (RES). In [14], the authors model RES uncertainty as a distributionally robust ellipsoid constructed from forecast errors from actual historical data. This approach will be similar to the one employed in this project's formulation of a robust OPF.

Finally, Model Predictive Control is another method often used to solve the OPF problem and can provide promising results by constantly iterating and improving itself because of its ability to observe and then course correct, even in the face of uncertain future states. For example, [15] formulates an MPC model that maximizes the output from renewable energy generation by managing the charge and discharge cycles of the battery storage system to relax network capacity, in a situation and formulation quite similar to our own.

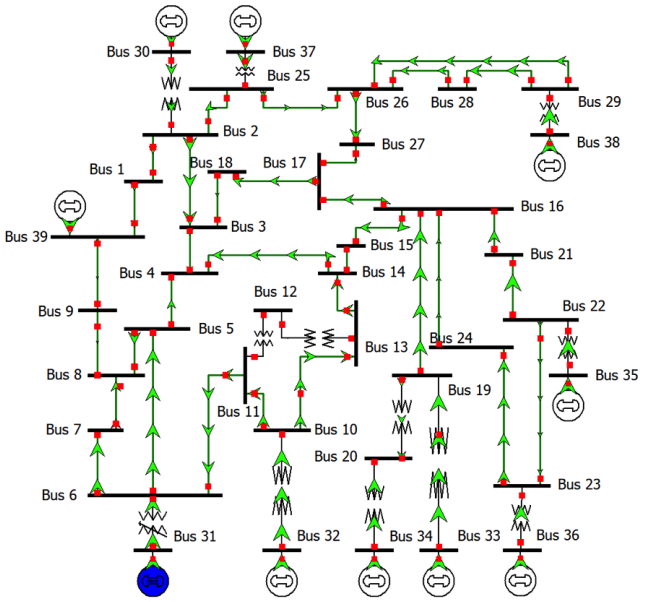
1.3 Focus of this Study

In this study, we consider a 24-hour period and find the optimal power flow for operation of a fully renewable microgrid that has solar, wind and battery storage assets. The objective is to minimize diesel generation while ensuring all loads are adequately supplied and all physical constraints are met. To accomplish this, we perform both robust optimization and model predictive control. To compare the performance of the two methods, we consider the curtailment of renewable power and deem more curtailment to be worse performance.

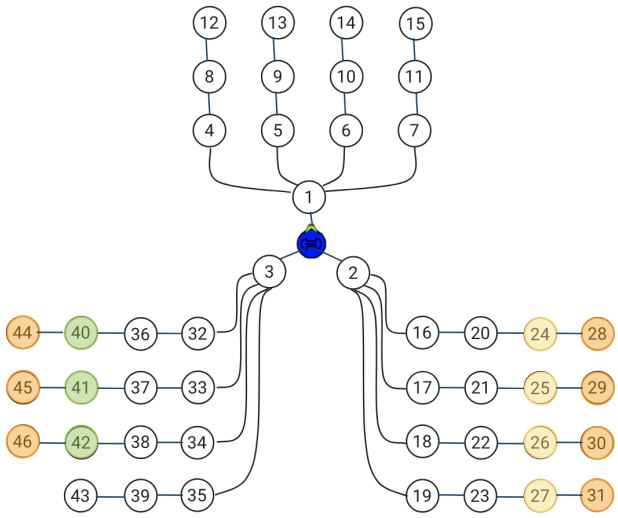
2 Technical Description

2.1 Network Model

In this study, we considered 3 different grid networks, all derived from the IEEE 39-bus test system [16]. In our first network iteration, we considered the meshed network in Figure 1a. We input wind generation data at buses 30, 37, and 38, solar generation data at buses 41, 42, 43, 44, and 45, and building load data at all other nodes, except for bus 31 which we treated as a diesel generation node. In this first iteration, we also extended the network by 8 nodes so that each renewable energy source had its own node for battery storage. Using the framework from HW3 [17], we constructed the relevant adjacency matrix and parent vector for the meshed network. Additionally, we used the line resistance and reactance values provided in the IEEE 39-bus technical note. For connections between renewable nodes and their respective battery storage nodes, we used resistance and reactance values 10 times smaller than the smallest values seen in the standard network to represent that the battery storage is onsite (impedance is proportional to line length).



(a) IEEE 39-bus mesh network [16]. Blue node is a diesel generator.



(b) 47-node radial network. Blue node is a diesel generator, green nodes are wind generators, yellow nodes are solar generators, orange nodes are battery storage, and white nodes are loads.

Figure 1: Two of the three networks we considered.

Unfortunately, formulating the meshed network problem with the framework from HW3 was intractable and led to "optimal" results that set all power flow to zero. To pivot from this issue, we constructed a radial version of the extended 47-node network, pictured in Figure 1b. In this network, we used a similar range of line resistance and reactance values to model physical distances between loads and generators. Perhaps we had more debugging to do, but results from this network were also lackluster, recommending zero power flow everywhere as the optimal.

Instead of spending more time with the radial network, we decided to abandon modeling the network connectivity using the adjacency matrix and parent node vector and solve for power on individual lines by using the voltages at the nodes on either side of the line and line impedance

to calculate the corresponding power via Ohm's Law. The details of this relaxation are further described in section §2.3.

2.2 Input Data

As mentioned in the previous section, we input real-world data into the network to model wind and solar generation and consumer demand. The wind generation data we used comes from the Australia Energy Market Operator [18]. They provided 5-minute resolution data for 22 wind farms. The solar generation and consumer demand data we used comes from the University of California, San Diego [19]. The dataset contains 15-minute resolution generation data from 67 panel arrays and load data from 27 buildings. We processed the data by downsampling the wind data (to be 15-minute resolution), splitting the three datasets into daily episodes, and normalizing the power generation/consumption by the daily maxima. For network configurations using the IEEE 39-bus network, normalized, downsampled data from three of wind farms was input at buses 30, 37, and 38 and normalized solar generation data was input at nodes 41, 42, 43, 44, and 45. The remaining 30 nodes were loaded with data from one of each of the 27 buildings, with three nodes receiving repeat data.

2.3 Formulation of Optimal Power Flow Problem

In developing our formulation, we initially adapted the HW3 constraints to also account for time by indexing our decision variables by time. However, we found that since some of our nodes had multiple parents, the optimal values of some of our decision variables were driven to 0. To combat this issue, we instead developed a formulation with decision variables and constraints that allowed us to use a non-radial network.

2.3.1 Decision Variables

The following is a list of the decision variables whose values we will determine when we solve our optimization problem:

1. $p_i(t)$, $q_i(t)$: active and reactive diesel power generated at generator node i at time t
2. $p_i^r(t)$, $q_i^r(t)$: active and reactive power generated at renewable node i at time t
3. $\mathbf{W}_{ij}(t)$: squared voltage between nodes i and j at time t

2.3.2 Optimization Parameters

The following is a list of the parameters that will be used in our optimization problem:

1. T: set of time points in the time horizon we are considering
2. $\mathcal{D}$: set of diesel nodes
3. $\mathcal{G}$: set of generator nodes
4. $\mathcal{R}$: set of renewable nodes
5. $\mathcal{B}$: set of buses

6. $\mathcal{L}$: set of lines
7. $l_i^P(t)$, $l_i^Q(t)$: active and reactive demand of node i at time t
8. $\underline{p_i}$: minimum active power at generator node i
9. $\overline{p}_i$, $\underline{p_i}$: maximum and minimum active power at node i
10. $\overline{q}_i$, $\underline{q_i}$: maximum and minimum reactive power at generator node i
11. $\overline{r}_i(t)$, $\underline{r_i}(t)$: maximum and minimum active power at renewable node i at time t
12. $\overline{v}$, $\underline{v}$: maximum and minimum nodal voltage, which we set to be 1.05 and 0.95, respectively.
13. $\overline{s}_{ij}$: maximum apparent power through line (i, j)
14. γ_{ij}^*: impedance of line (i, j)

2.3.3 Objective Function

Our objective function minimizes the total amount of diesel used over the time horizon.

$$\min \sum_{t \in T} \sum_{i \in \mathcal{D}} p_i(t)$$

2.3.4 Constraints

This formulation is an adapted form of the second-order cone relaxation (SOCR) of the alternating current (AC) optimal power flow problem. This relaxation was first introduced in [20] and subsequently used to great effect in a number of optimal power flow applications such as in unit commitment, grid reconfiguration, and expansion planning [**?**, 1, 21]. The advantage of the SOCR formulation lies in its efficient solving by commercial solvers. For this particular project, it also has the advantage of applicability to meshed grids. Because of our consideration of the meshed IEEE 39-bus system over a 24-hour period, the solving efficiency is crucial to obtaining timely solutions.

$$\underline{p_i} \leq p_i(t) \leq \overline{p_i} \ , \qquad \forall i \in \mathcal{G} \quad (1)$$

$$\underline{q_i} \leq q_i(t) \leq \overline{q_i} \ , \qquad \forall i \in \mathcal{G} \quad (2)$$

$$\underline{r_i(t)} \leq p_i^r(t) \leq \overline{r_i(t)} \ , \qquad \forall i \in \mathcal{R} \quad (3)$$

$$\underline{q_i} \leq q_i^r(t) \leq \overline{q_i} \ , \qquad \forall i \in \mathcal{R} \quad (4)$$

$$\underline{v^2} \leq \mathbf{W}_{ii}(t) \leq \overline{v}^2 \ , \qquad \forall i \in \mathcal{B} \quad (5)$$

$$\left\| (\mathbf{W}_{ii}(t) - \mathbf{W}_{ii}(t)) \gamma_{ij}^* \right\| \leq \overline{s}_{ij} \ , \qquad \forall ij \in \mathcal{L} \quad (6)$$

$$\left\| \begin{matrix} 2\mathbf{W}_{ij}(t) \\ \mathbf{W}_{ii}(t) - \mathbf{W}_{jj}(t) \end{matrix} \right\| \leq \mathbf{W}_{ii}(t) + \mathbf{W}_{jj}(t) \ , \qquad \forall ij \in \mathcal{L} \quad (7)$$

$$p_i(t) + p_i^r(t) + l_i^P(t) + j\left[q_i(t) + q_i^r(t) + l_i^Q(t)\right] = \sum_{j \in \mathcal{N}} (\mathbf{W}_{ii}(t) - \mathbf{W}_{ij}(t)) \gamma_{ij}^* \qquad \forall ij \in \mathcal{L} \quad (8)$$

The following is a brief description of each constraint:

1. Bounds on active power for diesel generator node $i \in \mathcal{G}$ at time t
2. Bounds on reactive power flow for diesel generator node $i \in \mathcal{G}$ at time t
3. Bounds on active power for renewable node $i \in \mathcal{R}$ at time t
4. Bounds on reactive power flow for renewable node $i \in \mathcal{R}$ at time t
5. Represents the voltage limits
6. Represents the norm of the power flowing through line (i, j)
7. Represents the SOCP relaxation condition on the voltages
8. Represents the line flow constraints for line (i, j)

It is important to note that we are able to have a non-radial network as a direct consequence of constraints (7) and (8).

2.4 Battery Energy Storage System Decision Variables and Constraints

For clarity, the additional Decision Variable, Parameters and Constraints for the Battery Energy Storage System (BESS) to be included in the OPF problem are as follows.

Decision Variables

The decision variables for each BESS node i at time t are defined as follows:

1. $\text{SoC}_i(t)$: State of charge at BESS node i at time t.
2. $P_{out,i}(t)$: Power output from BESS node i at time t.
3. $P_{in,i}(t)$: Power input to BESS node i at time t.

Parameters

1. η_{in}: charging efficiency.
2. η_{out}: discharging efficiency.
3. $P_{in,\max}$: maximum power charging.
4. $P_{out,\max}$: maximum power discharging.
5. $\text{SoC}_{\min}(t)$: minimum state of charge (0).
6. $\text{SoC}_{\max}(t)$: maximum state of charge (100).
7. Starting SoC: initial state of charge (10).
8. Cycle: boolean indicator to determine if the state of charge must return to Starting SoC at the end of the period.

Constraints

The BESS must adhere to both static and dynamic constraints. The static constraints are constraints that hold for all value of $t \in T$. The dynamic constraints model how the battery SOC changes over time, as a function of the battery SOC in the previous time period and the power into the node, and the efficiency.

The BESS nodes must adhere to the following static constraints:

$$P_{\text{out},i}(t) \geq 0 \tag{9}$$
$$P_{\text{in},i}(t) \geq 0 \tag{10}$$
$$\text{SoC}_i(t) \geq 0 \tag{11}$$
$$P_{\text{out},i}(t) \leq \text{Max Out} \tag{12}$$
$$P_{\text{in},i}(t) \leq \text{Max In} \tag{13}$$
$$\text{SoC}_i(t) \leq \text{Max Charge} \tag{14}$$
$$\text{SoC}_i(t=0) = \text{Starting SoC} \tag{15}$$
$$P_{\text{in},i}(t=0) = 0 \tag{16}$$
$$P_{\text{out},i}(t=0) = 0 \tag{17}$$
$$\text{SoC}_i(t=\text{end}) = \text{Starting } SoC \quad (\text{if cycle} = \text{True}). \tag{18}$$

The BESS nodes must adhere to the following dynamic constraint:

$$\text{SoC}_i(t) = \text{SoC}_i(t-1) + \left(P_{\text{in},i}(t) \cdot \eta_{\text{in}} - \frac{P_{\text{out},i}(t)}{\eta_{\text{out}}} \right) \tag{19}$$

This constraint encodes the energy dynamics of the BESS where the $SoC(t)$ of the next time step is contingent on the previous $SoC(t-1)$, the power in or out as well as the given charging and discharging efficiencies respectively.

2.5 Robust Optimization and Handling Stochasticity in Input Data

With the inclusion of renewables, namely wind and solar energy, we introduce uncertainty to our problem. Wind and solar energy output is highly dependent on weather conditions, which vary over time. We want our optimization problem to remain convex and specifically want to account for the worst-case so we consider these constraints from a robust optimization lens. Thus, we model the stochasticity introduced by solar and wind through second-order cone constraints so our problem remains convex.

Let $W_a(t)$ and $W_c(t)$ be random variables, such that $W_a(t)$ represents the uncertain power capacity of solar installation a at time t and $W_c(t)$ represents the uncertain power capacity of wind turbine c at time t. We define $a \in A$ as our set of solar nodes, where $A \subset \mathcal{R}$, and $c \in C$ as our set of wind nodes such that $C \subset \mathcal{R}$. To estimate the values of $W_a(t)$ we used real data. Specifically, we used data on solar generation for 67 panels across 27 buildings and took the average at the specific time t, denoted by $\bar{a}(t)$. To estimate the wind generation, we took the average generation across 22 turbines at time t, denoted by $\bar{c}(t)$. We similarly considered the standard deviations at each time t for both wind and solar, denoted by $S_a(t)$ and $S_c(t)$, respectively. This allowed us to form a range of possible values for $W_a(t)$ and $W_c(t)$, centered at $\bar{a}(t)$ and $\bar{c}(t)$. To get sensible values for grid-scale solar installation we scaled our values accordingly.

For solar installation a, we have $W_a(t) \in [\bar{a}(t) - S_a(t), \bar{a}(t) + S_a(t)]$. Similarly, we have $W_c(t) \in [\bar{c}(t) - S_c(t), \bar{c}(t) + S_c(t)]$ for wind turbine c. Our decision variables are the percentage power output of each PV installation $a \in A$ at time $t \in T$ denoted by $\sigma_a(t)$, and the percentage power output, $\sigma_c(t)$, of each wind turbine $c \in C$ at time $t \in T$.

Our second-order cone constraints are:

$$\begin{bmatrix} -\bar{a}(t) & 1 \end{bmatrix} \begin{bmatrix} \sigma_a(t) \\ s_a(t) \end{bmatrix} \left\| \begin{bmatrix} S_a(t) & 0 \\ 0 & 0 \end{bmatrix}^T \begin{bmatrix} \sigma_a(t) \\ s_a(t) \end{bmatrix} \right\|_2 \leq 0 \tag{20}$$

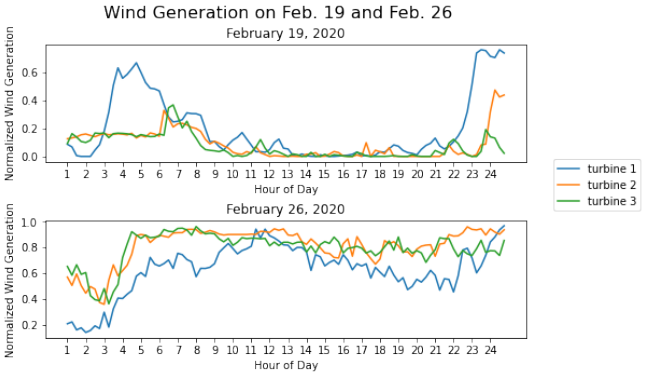
$$\begin{bmatrix} -\bar{c}(t) & 1 \end{bmatrix} \begin{bmatrix} \sigma_c(t) \\ s_c(t) \end{bmatrix} \left\| \begin{bmatrix} S_c(t) & 0 \\ 0 & 0 \end{bmatrix}^T \begin{bmatrix} \sigma_c(t) \\ s_c(t) \end{bmatrix} \right\|_2 \leq 0 \tag{21}$$

The multi-period optimal power flow problem that we consider in this paper is constituted by combining constraints $1-19$. When solving the full-information case, the bounds on maximum and minimum renewable generation is known and the robust constraints are not considered. When solving the robust case, constraints $20 - 21$ are added to the problem while actual renewable generation is considered to be unknown.

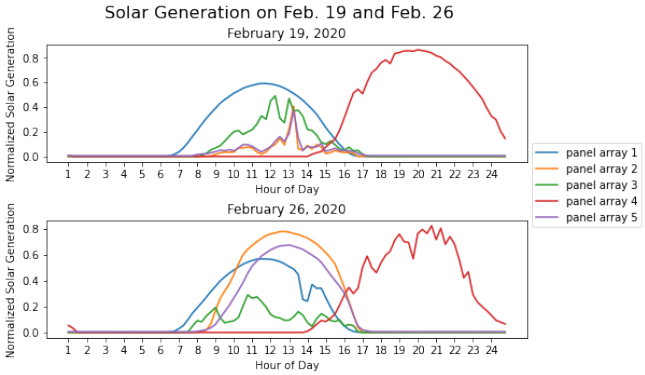
3 Results and Discussion

3.1 Full Information and Robust Optimization

With the formulation of our distinct optimization methods complete, two sample days from the historical data set were chosen to exemplify the performance of the different solution methods. Both days had relatively similar demand profiles that have a slightly convex shape 3. However, the first day, February 26th 2020 possessed a relatively stereotypical RES generation, ramping up until a peak at mid day, with a subsequent drop off towards the end of daylight hours 2b. Meanwhile, February 19th had sporadic RES generation with many more local minimums and maximums 2b. Furthermore, the renewable generation was not enough to satisfy demand without the use of diesel generation. The difference between the two days elucidate the inherent distinctions between full information and robust optimization.



(a) Wind generation data on Feb. 19^{th} and 26^{th} for the 3 nodes that are wind generation nodes.



(b) Solar generation data on Feb. 19^{th} and 26^{th} for the 5 nodes that are wind generation nodes.

Figure 2

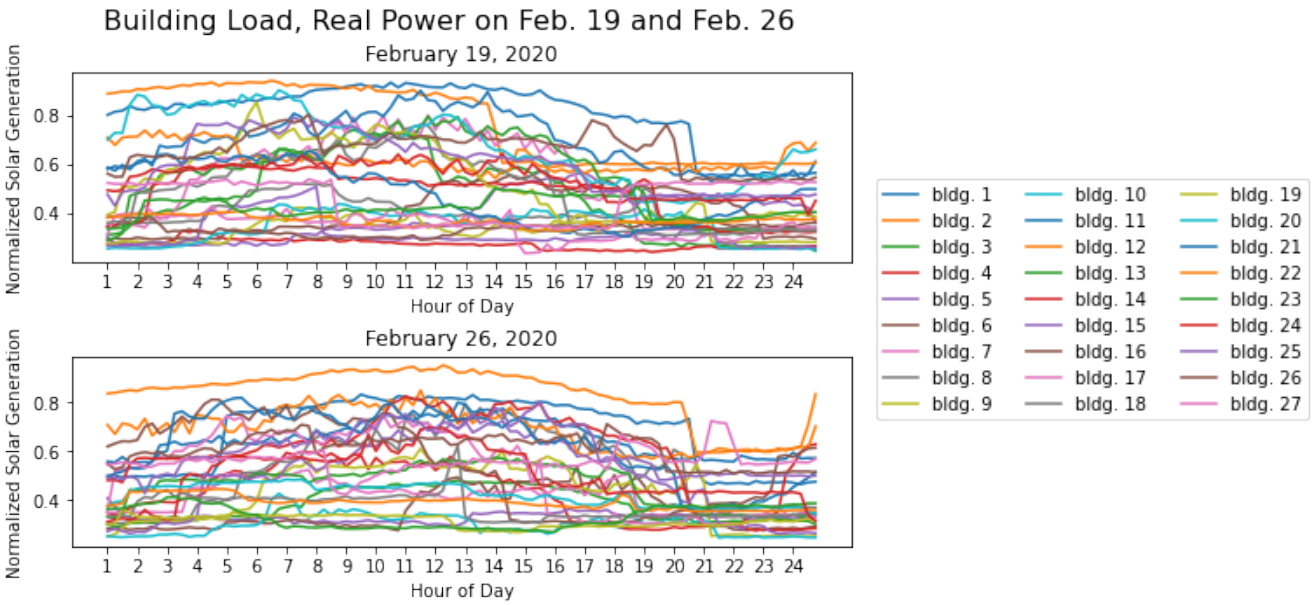
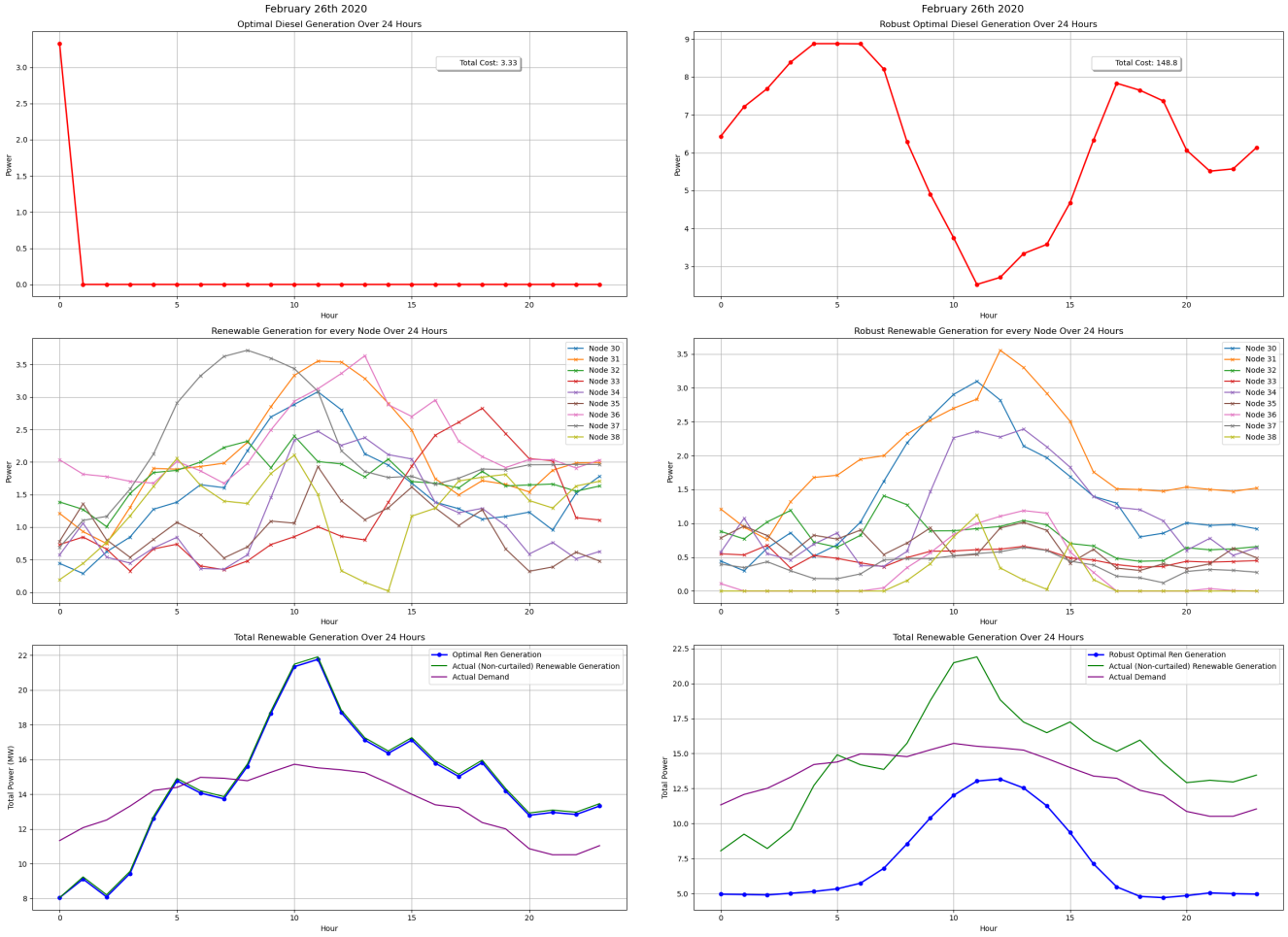


Figure 3: Building load data for the 30 consumer nodes in the network. Three nodes are copies so that the 27 buildings worth of raw data cover all 30 nodes.



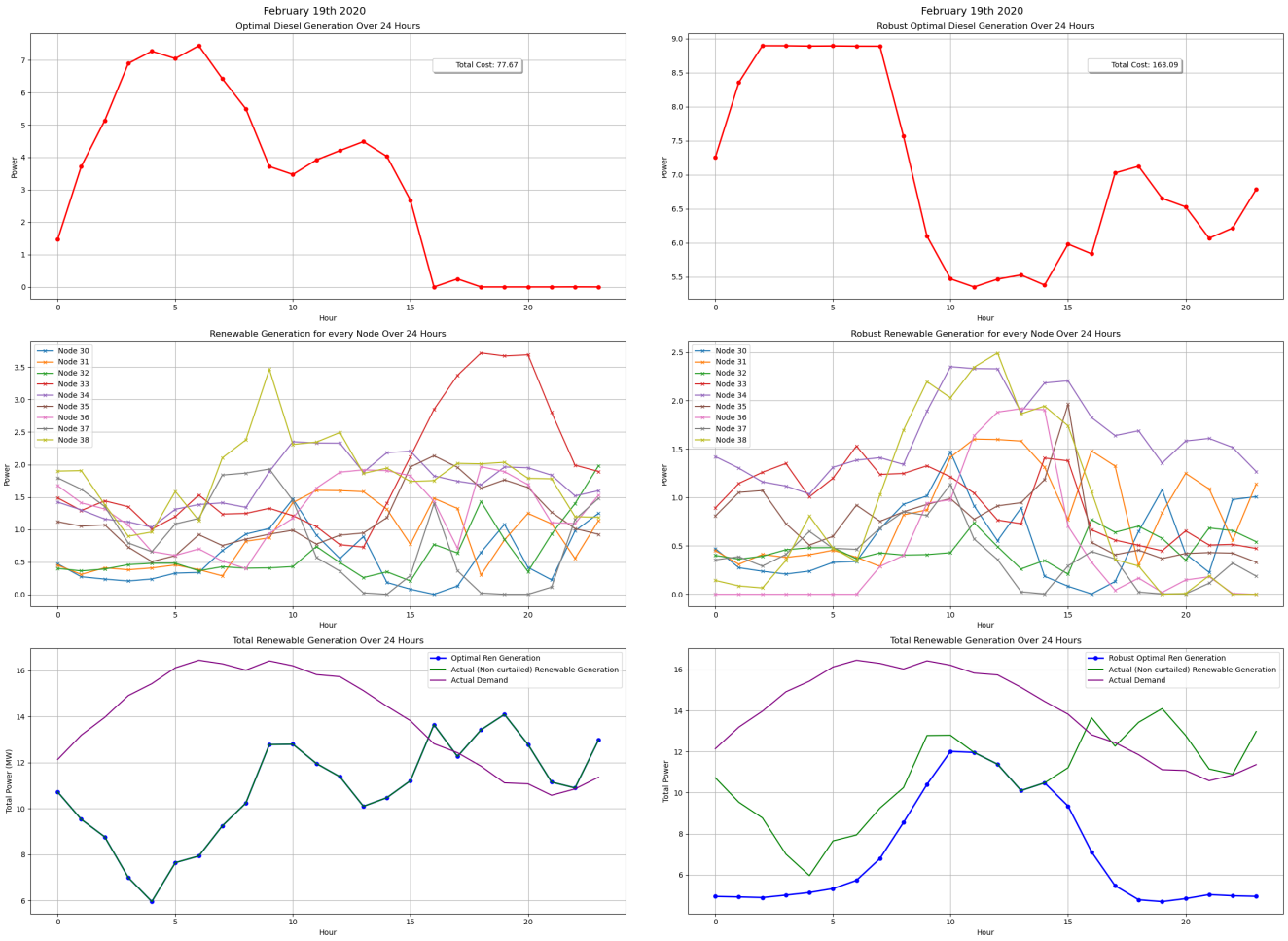
(a) Febuary 26th Full Information

(b) Febuary 26th Robust

Figure 4: Febuary 26th Results

Unsurprisingly with full information, the utilized RES generation is *very* close to the actual renewable generation. In fact this as good of a result that could be obtained with the given parameters. The slight curtailment of the RES at the midday peak is directly tied to the requirement that each BESS return to the starting SoC of 10%. As seen in Figure 6a, Battery 38 actually is required to discharge its excess energy to the grid in order to achieve an SoC of 10% by end of day. The excess power from this battery node causes the slight curtailment observed in 4a.

Meanwhile the differences are stark when robust constraints are implemented. 4b illustrates that the utilized renewable energy sources (RES) are significantly lower than the available resources. The net used RES also follows a standard Gaussian like curve owing to the mean and standard deviation constraints imposed as described earlier. Consequently the relative cost of the diesel node is much higher. Furthermore, as RES is greatly curtailed, the use of the BESS is also greatly reduced as seen in 6b. The addition of robust constraints greatly constrain the efficiency of the micro grid which becomes acutely obvious when RES is at the higher end of its potential output.



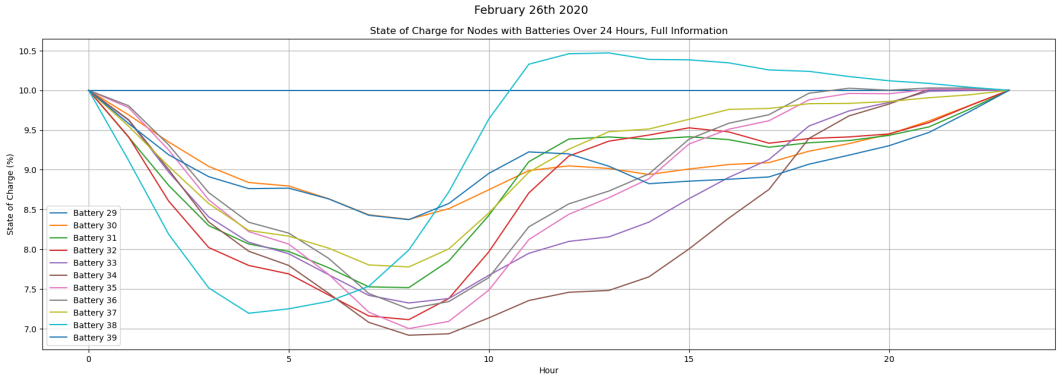
(a) Febuary 19th Full Information

(b) Febuary 19th Robust

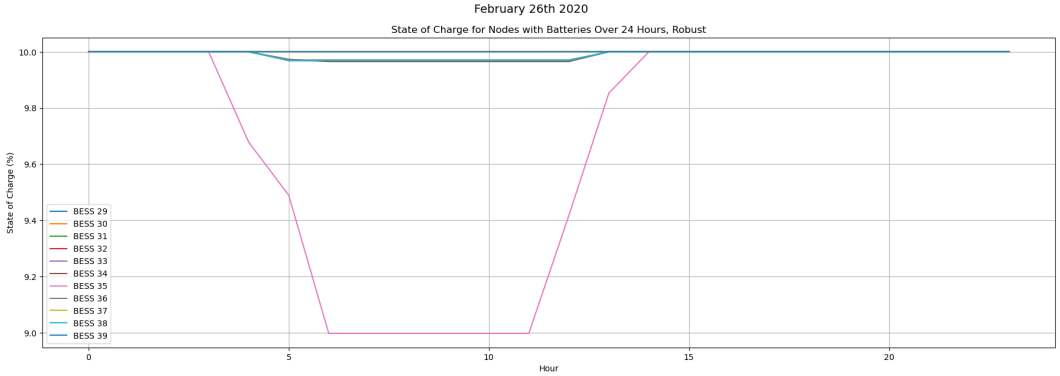
Figure 5: Febuary 19th Results

February 19th is a slightly different story owing to the more sporadic solar production on that day. Again5a shows the full utilization of the RES, although the total cost is higher owing to the generally lower magnitude of RES generation on this day.

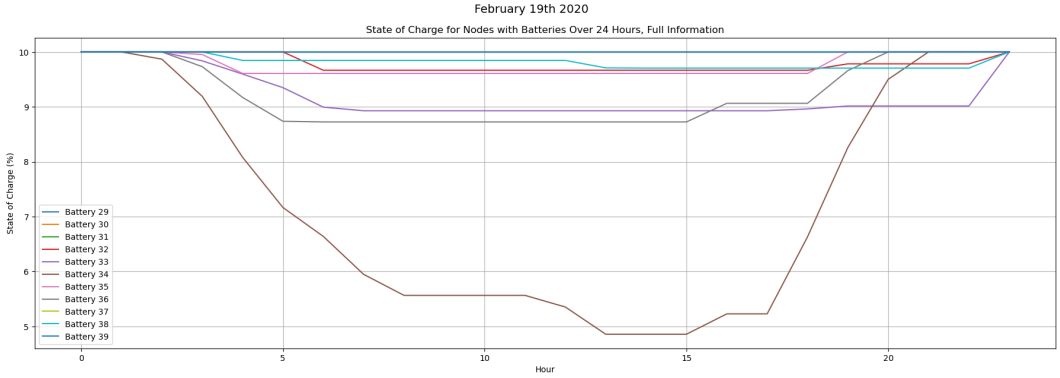
In the robust case, there is curtailment of the RES but relatively speaking, not nearly as much as in 4b (as the overall generation is much less). In fact at around 13-14 hours, RES generation actually dips below what would be generally expected at that point in the day. It is interesting to note in 6b that the BESS are utilized much more as compared to 6c and 6b. In aggregate its difficult to explain as to why BESS is utilized more in the full information vs robust case and vice versa. It is inherently related to the specific individual node demands and how they are spatially related to the BESS vs the singular diesel node.



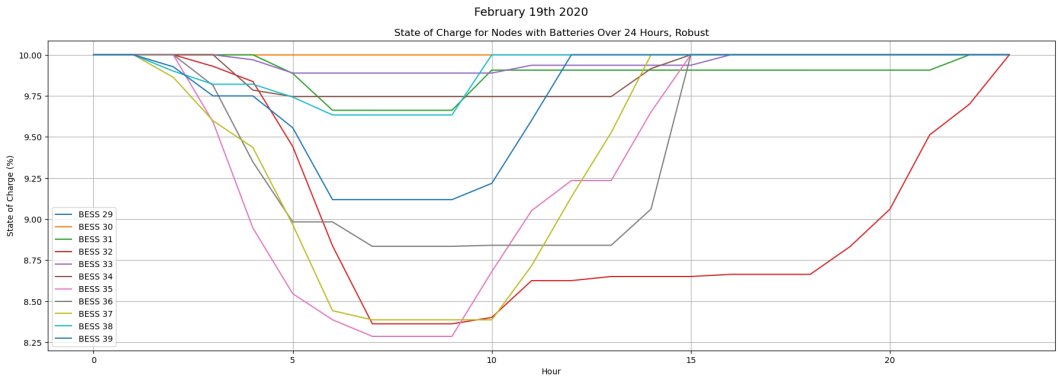
(a) February 26th Full Information



(b) February 26th Robust



(c) February 19th Full Information

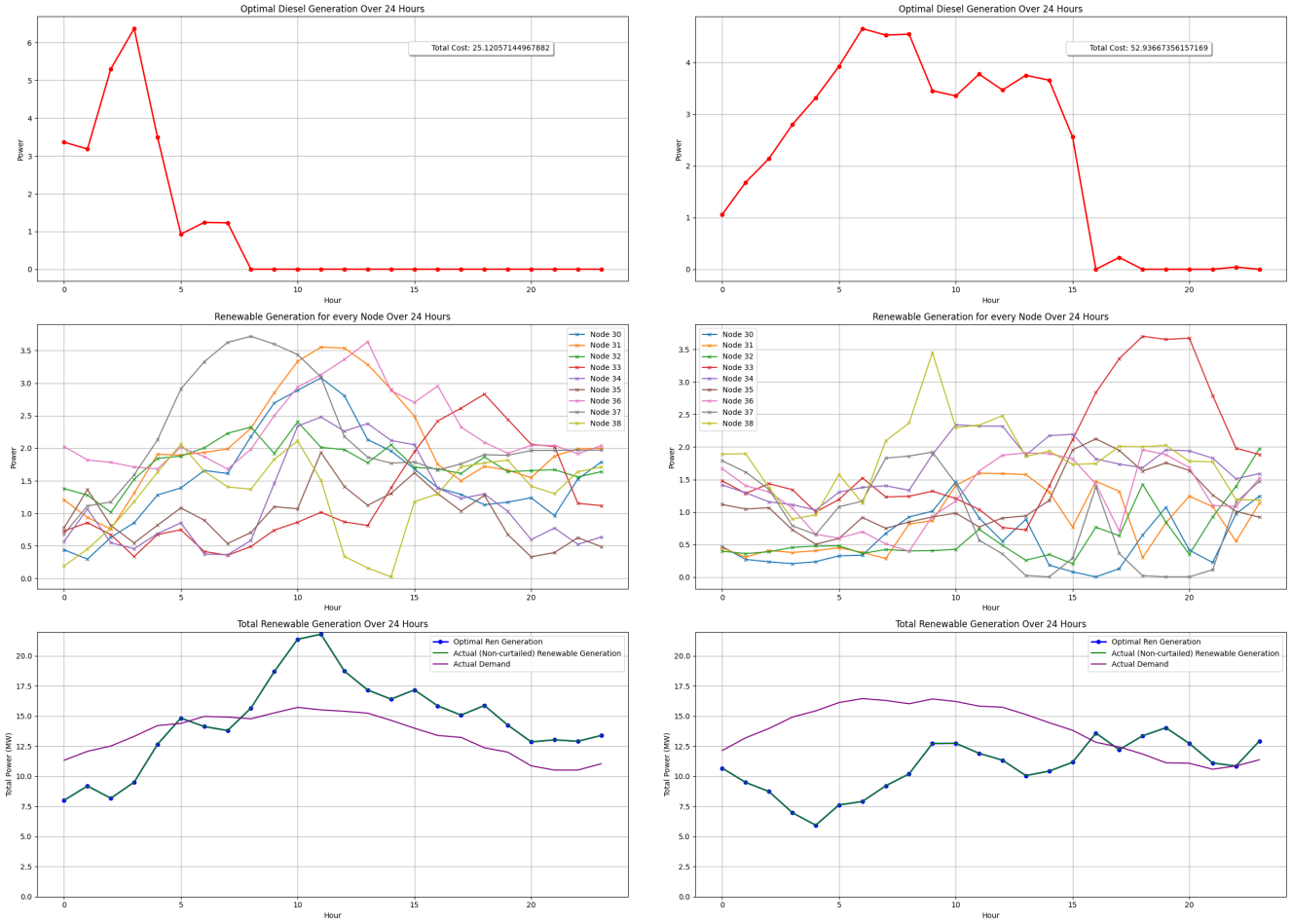


(d) February 19th Robust

Figure 6: BESS Dynamics

3.2 Model Predictive Control Implementation

In addition to the Full Information and Robust solutions to the OPF problem for February 26th and 19th, a Model Predictive Control (MPC) algorithm was utilized to balance the consideration for realistic and safe operating characteristics under uncertainty (robustness) while exploiting the available RES generation at every given point. This was achieved using a 24-hour rolling horizon with real-time information and robust prediction. In this way, the algorithm has access to real-time renewable generation (rather than a prediction) but uses the worst-case (robust) generation for the remainder of the 24-hour planning horizon. The first decision in the sequence of 24 hours is then implemented and the process is repeated for every hour. The results of this approach are presented in the Figures 7a and 7b.



(a) MPC Solution for Feb 26th

(b) MPC Solution for Feb 19th

Figure 7: MPC Results on Trial Datasests

We can observe that the MPC formulation eliminates many of the issues present in the robust formulation. Notably, the MPC demonstrates an improved use of renewable generation and a better battery usage through time. The MPC vastly outperforms the robust optimization in total diesel generation. However, the MPC notably does not consider full-information of the future and should produce good results even in situations where actual renewable generation is low.

4 Summary and Conclusions

In this project, we explored two methods for computing the optimal flow of power in a microgrid over a time horizon of 24 hours: robust optimization and model predictive control. We tried three different test beds with each of these methods: a version of the IEEE 39-bus test system extended to 47 nodes to include battery storage, a radialized version of the 47-node extended network, and the standard IEEE 39-bus system with battery storage located at the same nodes as their respective renewable generators. In order to handle the meshed network that is the standard IEEE 39-bus system, we performed an SOCP relaxation, which allowed us to ignore assignments of parent and children nodes and treat all flows as two-way flows. We used full-information optimization to benchmark both the robust optimization and model predictive control results. As we used a time horizon of 24 hours, we sought to compare the two methods on two quite different days: one day where the solar and wind generation was fairly consistent and a second where storms or clouds caused the generation to be more chaotic. In both instances, we found that the robust optimization significantly curtailed renewable generation in favor of diesel generation, which would be safe when implemented in a real-world power grid, but would not fully capitalize on the cleaner energy available. The model predictive control on the other hand, performed as well, or better than, the baseline full-information optimization case. Thus, we believe that if any method were to be used for real-time optimal power flow, it would be model predictive control.

References

[1] F. Badal, S. Sarker, and S. Das, "A survey on control issues in renewable energy integration and microgrid," *Protection and Control of Modern Power Systems*, vol. 4, no. 1, 2019.

[2] J. A. Taylor, *Convex optimization of power systems*. Cambridge University Press, 2015.

[3] F. Zohrizadeh, C. Josz, M. Jin, R. Madani, J. Lavaei, and S. Sojoudi, "A survey on conic relaxations of optimal power flow problem," *European Journal of Operational Research*, vol. 287, no. 2, pp. 391–409, 2020.

[4] S. H. Low, "Convex relaxation of optimal power flow—part I: Formulations and equivalence," *IEEE Transactions on Control of Network Systems*, vol. 1, no. 1, pp. 15–27, 2014.

[5] E. Hazan and S. Kale, "Online submodular minimization." *Journal of Machine Learning Research*, vol. 13, no. 10, 2012.

[6] X. Yi, X. Li, L. Xie, and K. Johansson, "Distributed online convex optimization with time-varying coupled inequality constraints," *IEEE Transaction on Signal Processing*, vol. 68, pp. 731–746, 2020.

[7] M. Picallo, L. Ortmann, S. Bolognani, and F. Dörfler, "Adaptive real-time grid operation via online feedback optimization with sensitivity estimation," *Electric Power Systems Research*, vol. 212, p. 108405, 2022.

[8] Z. Yan and Y. Xu, "Real-time optimal power flow: A lagrangian based deep reinforcement learning approach," *IEEE Transactions on Power Systems*, vol. 35, no. 4, pp. 3270–3273, 2020.

[9] P. Siano, C. Cecati, H. Yu, and J. Kolbusz, "Real time operation of smart grids via fcn networks and optimal power flow," *IEEE Transactions on Industrial Informatics*, vol. 8, no. 4, pp. 944–952, 2012.

[10] Y. Guo, K. Baker, E. Dall'Anese, Z. Hu, and T. Summers, "Stochastic optimal power flow based on data-driven distributionally robust optimization," in *2018 Annual American Control Conference (ACC)*, 2018, pp. 3840–3846.

[11] D. Erazo-Caicedo, E. Mojica-Nava, and J. Revelo-Fuelagan, "Model predictive control for optimal power flow in grid-connected unbalanced microgrids," *Electric Power Systems Research*, vol. 209, p. 108000, 2022.

[12] T. Samakpong, W. Ongsakul, and J. Polprasert, "Robust optimization-based ac optimal power flow considering wind and solar power uncertainty," in *2014 International Conference and Utility Exhibition on Green Energy for Sustainable Development (ICUE)*. Pattaya, Thailand: IEEE, 2014, pp. 1–7.

[13] A. Ramadan, M. Ebeed, S. Kamel, and L. Nasrat, "Chapter 5 - optimal power flow for distribution systems with uncertainty," in *Uncertainties in Modern Power Systems*, A. F. Zobaa and S. H. A. Aleem, Eds. Academic Press, 2021, pp. 145–162. [Online]. Available: https://www.sciencedirect.com/science/article/pii/B9780128204917000050

[14] A. Arab and J. E. Tate, "Distributionally robust optimal power flow via ellipsoidal approximation," *IEEE Transactions on Power Systems*, vol. 38, no. 5, pp. 4826–4839, 2023.

[15] P. Zeng and Z. Wu, "Model predictive control for energy storage systems in a network with high penetration of renewable energy and limited export capacity," *IEEE*, p. 1, 2014. [Online]. Available: https://ieeexplore.ieee.org/document/7038359

[16] R. D. Zimmerman, C. E. Murillo-Sánchez, and R. J. Thomas, "Matpower: Steady-state operations, planning, and analysis tools for power systems research and education," *IEEE Transactions on Power Systems*, vol. 26, no. 1, pp. 12–19, 2011.

[17] S. Moura and D. B. Arnold, "Hw3: Optimal economic dispatch in distribution feeders with renewables," 2024.

[18] J. Dowell, "Australian electricity market operator (aemo) 5 minute wind power data," *University of Strathclyde, doi*, vol. 10, 2015.

[19] S. Silwal, C. Mullican, Y.-A. Chen, A. Ghosh, J. Dilliott, and J. Kleissl, "Open-source multi-year power generation, consumption, and storage data in a microgrid," *Journal of Renewable and Sustainable Energy*, vol. 13, no. 2, 2021.

[20] R. Jabr, "Radial distribution load flow using conic programming," *IEEE Transactions on Power Systems*, vol. 21, no. 3, pp. 1458,1459, 2006.

[21] Y. Chen, Y. Li, J. Xiang, and X. Shen, "An optimal power flow formulation with socp relaxation in radial network," in *2018 IEEE 14th International Conference on Control and Automation (ICCA)*, 2018, pp. 921–926.

Appendix 1

```
import numpy as np
import cvxpy as cp

from case import *


class MRunResult:

    def __init__(self, diesel, renew, sot, value):
        self.diesel = diesel
        self.renew = renew
        self.sot = sot
        self.value = value


def runMOPF(case, genData, T=1, init_sot=0.1, cycle=True, robust=False, verb=Fals
    # load data from case
    n = case.N
    Y = case.adj
    Sij = case.smax
    v_lim = case.vlim
    lines = case.getLines()

    renew = genData.renew
    avgs = genData.avgs
    stds = genData.stds
    batteries = genData.batteries
    diesel = genData.diesel
    Load_data = genData.Load_data
    renew_lim = genData.renew_lim

    #Voltage matrix
    V = [cp.Variable((n,n), hermitian=True) for _ in range(T)]

    # power transfer variables
    pij = cp.Variable((len(lines), T))
    pji = cp.Variable((len(lines), T))
    qij = cp.Variable((len(lines), T))
    qji = cp.Variable((len(lines), T))

    # total generation variable
    pi_g = cp.Variable((n, T))
    qi_g = cp.Variable((n, T))
    renew_real = cp.Variable((len(renew), T))
    renew_react = cp.Variable((len(renew), T))
```

```
diesel_real = cp.Variable((len(diesel), T))
diesel_react = cp.Variable((len(diesel), T))
sot = cp.Variable((n, T))
pow_out = cp.Variable((n, T))
pow_in = cp.Variable((n, T))

max_out = 2
max_in = 3
max_charge = 100
eta_in = 0.95
eta_out = 0.95

# Define constraints
constraints=[]

constraints += [pow_out >= 0, pow_in >= 0, sot >= 0]
constraints += [pow_out <= max_out, pow_in <= max_in, sot <= max_charge]

#Constraints on active and reactive generation (min-max)
constraints += [diesel_real >= 0, diesel_real <= 50]
constraints += [diesel_react >= -50, diesel_react <= 50]
constraints += [renew_react >= -50, renew_react <= 50]
for t in range(T):
    for i in range(renew_lim.shape[0]):
        constraints += [renew_real[i, t] <= renew_lim[i, t]]
constraints += [renew_real >= 0]


# Calculate the sum of all inbound power flows to each bus
for t in range(T):

    if robust:
        max_gen = np.sum(avgs[:,t])-np.linalg.norm(stds[:,t])
        if max_gen > 0:
            constraints += [cp.sum(renew_real[:,t])<=max_gen]
        else:
            constraints += [cp,sum(renew_real[:,t])<=0]
    if cycle:
        constraints += [sot[:,-1] == init_sot]

    for i in range(n) :
        psum = 0
        qsum = 0
        for line in range(len(lines)):
            start, end = lines[line]
            if start == i:
                psum += pij[line, t]
                qsum += qij[line, t]
```

```
            elif end == i:
                psum += pji[line, t]
                qsum += qji[line, t]

        # Sum pij = pi
        if t>0 and i in batteries:
            constraints += [sot[i, t] == sot[i, t-1] + (pow_in[i, t]*eta_in-p
        else:
            constraints += [sot[i, t] == init_sot]
            constraints += [pow_in[i, t]==0, pow_out[i, t] == 0]
        gensum = 0
        rgensum = 0
        if i in batteries:
            gensum += -pow_in[i, t]+pow_out[i, t]
        if i in renew:
            gensum += renew_real[renew.index(i), t]
            rgensum += renew_react[renew.index(i), t]
        if i in diesel:
            gensum += diesel_real[diesel.index(i), t]
            rgensum += diesel_react[diesel.index(i), t]
        constraints += [psum == gensum-Load_data[i,0, t]]
        # Sum qij = qi
        constraints += [qsum == rgensum-Load_data[i,1, t]]



        # Voltage limits
        constraints+=[cp.real(V[t][i,i])>= 0.95**2, cp.real(V[t][i,i]) <= 1.0
        constraints += [cp.imag(V[t][i,i]) == 0]

    # Power flow equations (sparse representation)
    for line in range(len(lines)):
        i, j = lines[line]

        #Powerflow
        constraints+=[pij[line, t] + 1j*qij[line, t]==(V[t][i,i]-V[t][i,j])*n
        constraints+=[pji[line, t] + 1j*qji[line, t]==(V[t][j,j]-V[t][j,i])*n

        if not Sij[i,j] == 0:
        #Apparent power capacity S_bar
            constraints+=[cp.square(pij[line, t])+cp.square(qij[line, t])<=cp
            constraints+=[cp.square(pji[line, t])+cp.square(qji[line, t])<=cp

    for line in range(len(lines)):
        i, j = lines[line]
        constraints+=[cp.norm(cp.hstack([2*V[t][i,j],(V[t][i,i]-V[t][j,j])]))
        constraints+=[cp.norm(cp.hstack([2*V[t][j,i],(V[t][j,j]-V[t][i,i])]))

# Define costs
Costs = cp.sum(diesel_real)
```

```
    prob = cp.Problem(cp.Minimize(Costs), constraints)
    prob.solve(verbose=verb)
    try:
        ans = MRunResult(diesel_real.value, renew_real.value, sot.value, prob.val
    except TypeError:
        return None
    return ans


    case39 = loadCase("cases/case39.json")

    diesel = [29]
batteries = [29, 30, 31, 32, 33, 34, 35, 36, 37, 38]
renew = [30, 31, 32, 33, 34, 35, 36, 37, 38]

class gens:

    def __init__(self, diesel, batteries, renew, avgs, stds, Load_data, renew_lim
        self.renew = renew
        self.diesel = diesel
        self.batteries = batteries
        self.avgs = avgs
        self.stds = stds
        self.Load_data = Load_data
        self.renew_lim = renew_lim

import os
import matplotlib.pyplot as plt
pv_directory   = 'PV Generation Data'
load_directory = 'Building Load Data'
wind_directory = 'Wind Generation Data'

# ALREADY GENERATED, takes 5 minutes to generate again so keep commented out
#dp.generate_json_from_pv_data(pv_directory)
#dp.generate_json_from_bldg_data(load_directory)
#dp.generate_json_from_wind_data(wind_directory)

with open(os.path.join(pv_directory, 'pv_data.json'), 'r') as json_file:
    pv_dict = json.load(json_file)

for key in pv_dict.keys(): # get data from first key only (CAPTL_WF)
    solar_data = np.array(pv_dict[key])
    break

# for key, value in pv_dict.items():
    # print(key, value['0'])
    # break

with open(os.path.join(load_directory, 'real_data.json'), 'r') as json_file:
```

```
    real_load_dict = json.load(json_file)

with open(os.path.join(load_directory, 'reactive_data.json'), 'r') as json_file:
    reactive_load_dict = json.load(json_file)

with open(os.path.join(wind_directory, 'wind_data.json'), 'r') as json_file:
    wind_dict = json.load(json_file)

real_load_dict['GeiselLibrary0']['0']
def toNDct(dict):
    ans = np.zeros([len(dict.keys()), 427, 96])
    for b, bld in enumerate(dict):
        for d, day in enumerate(dict[bld]):
            ans[b, d, :] = dict[bld][day]
    return ans

real_load = toNDct(real_load_dict)
react_load = toNDct(reactive_load_dict)
wind = toNDct(wind_dict)
solar = toNDct(pv_dict)

Load_data = np.zeros((39, 2, 24))

day_index = -4 # Feb 29, 2020 is last day (-1), so use -4 for Feb. 26 and -8 for
Load_data[:27,:] = np.stack((real_load[:,day_index,0::4], react_load[:,day_index,


#Factor of 2.123 produces ~100% renewable penetration (Energy of renewables= ener
#Limits for Feb 26
renew_lim = (wind[:9,day_index,0::4]+solar[:9,day_index,0::4])*2.123


avgs = np.mean(wind[:8,:,0::4]+solar[:8,:,0::4], axis=1)*2.123
stds = np.std(wind[:8,:,0::4]+solar[:8,:,0::4], axis=1)*2.123

genDat = gens(diesel, batteries, renew, avgs, stds, Load_data, renew_lim)

#NON-ROBUST
#SoC Initial=10%
res = runMOPF(case39, genDat, T=24, init_sot=10)

#NON-ROBUST
import matplotlib.patches as mpatches

#ROBUST
res_robust = runMOPF(case39, genDat, T=24, init_sot=10, robust=True)
print('Robust Cost', np.round(res_robust.value,2))
```