

EE232 Final Project Report

Yidi Wang

June 12, 2019

1 Summary

Introduction

Nowadays, the use of photovoltaic (PV) is widespread because of the high demand of CO_2 emission reduction and renewable energy. However, the power generation from PV is not stable because people cannot control the light intensity from the sun and the ambient temperature. Therefore, how to match the intermittent energy production with the dynamic demand is a major challenge for the use of PV. The solution is to connect a energy storage unit to the network. This integration is currently limited by two constraints: regulations and power flow management. In this paper, the authors focus on the power flow management problem, which is about how to optimize the use of storage, the use of the PV source, and to match the local production with the local consumption. They proposed an optimal power management mechanism for grid connected PV systems with storage.

System Modeling

Solving the optimal power flow management problem need three stages: forecasting stages, predictive optimization stage and load command stage. The problem is solved in the predictive optimization stage, using the prediction from the priori knowledge of future from the first stage. They choose dynamic programming (DP) as the tool to solve the optimization problem.

In the system modeling, they considered the grid as a perfect power sink or source. The network contains three major components: PV generator, batteries and converter. The PV generator is modeled by a linear power source according to the ambient temperature and the irradiance level.

They considered three features of flat plate lead acid batteries. The first one is state of charge (SOC). The second is the aging of the battery, which is indicated by state of health (SOH). They choose an aging coefficient $Z = 3.10^{-4}$ for lead acid technology. The third one is voltage, which has been considered linear in charge and discharge as a function of SOC.

The last considered component is converter, which is modeled according to their efficiency as a function of the input normalized power, where losses are assumed to be a quadratic function.

Objective

The objective is to find the optimal sequence of SOC transition from the initial time to the final time with the lowest cash flow (CF) value. The transition equation is given by:

$$\text{Min}(\text{CF}) = \text{Min} \sum_{t=t_0}^T [\text{CR}(t) + \text{CP}(t)]$$

where "CR" stands for the cash received and "CP" stands for cash paid.

For the evaluation, they deployed the design onto a microcontroller and tested in real-time operation using the RTLab experimental bench. The results are compared with the ruled-based strategy in the same condition. In the simulation over 24 hours, the day-ahead predictive optimization can provide around 13% of gain on the electricity bill.

2 Critical Review

The results shown in the paper is not very good. The predicted scheduling is not close to what really happened. The system modeling in the paper is built on the ideal scenarios. As shown in Fig.16 in the paper, the curves are not overlapped at all except the initial state. The PV generator is expected to charge the battery when the loads and the grid are not drawing much power. However, in reality, as the evaluation part shows, the PV power is not optimized as it is mainly used to feed the grid instead of charging the battery. The results of ruled-based implementation shows that the battery is almost fully charged.

The given experiment results over 24 hours are not sufficient. In their expectation, the final SOC and the initial SOC should be similar over a day. They started the experiment when the battery is half-charged. They should at least continue the experiment for another day to see the SOC behavior if the SOC of the last day is almost full.

If one more experiment with a initial SOC being 100%, I can infer that the battery behavior in the predictive curve would discharge first, then stay at the half-charged state, while the ruled-based curve would always stays at a almost fully-charged state.

3 Improvement

For the first issue I mention in the above paragraph, that the PV generator is always charging the battery instead of feeding the grid, there are lots of uncertainty in the prediction. In the modeling, the power load of the battery is only constrained within a power range in which the battery can operate normally.

The naive solution should be adding constraints onto the transmission lines from PV generator to the energy storage unit to limit the power that the battery can draw from the generator, forcing more power flows to the grid. But the power flow to the battery is still hard to control and it cannot be dynamically adopted to the PV power generation rate.

Maybe a transmission line that can dynamically change its capacity according to the demand is needed. But it is hard to be achieved. Or equivalently, we can divide the capacity storage unit to smaller energy quantums with some capacity limits on the transmission lines to the batteries, and that is a distributed battery network. By independently switching on/off the connections to the batteries, we can schedule and control the power to the batteries more smartly.

If distributed batteries are used in the grid, we then need a monitor to keep track of a sequence of $SOC[1, \dots, n]$. At the end of every time slice, the algorithm will choose the connections to batteries to be switched on or off in the next time slice. The amount of needed power to the batteries is predicted based on the priori knowledge. The switching decisions can be made by a greedy algorithm. The battery with lower SOC will have a higher priority to be chosen.

In this way, we can have a better control over the power flow into the storage units. In the system architecture, we need to add a module inside predictive layer, where a battery monitor and

a greedy algorithm is implemented. The monitor will supervise the SOC of batteries, sort them in a ascending order, and the algorithm will decide which battery connections to the network should be switched.