Energy Management System MPC Discussion

For this project, I am going to take a look at a system from a recent research paper titled, "Simulation of energy management system using model predictive control in AC/DC microgrid". This study covers a type of system I'm interested in better understanding, and may also consider in my own research I hope to do here in this program. It combines a photovoltaic system with a battery storage system, supercapacitor, and a grid tie in order to establish a local microgrid. This could be a small scale residential system, but this could also apply to other scenarios, for instance a larger-scale PV generation system with a colocated load (hydrogen power, datacenter, cryptominer, etc). I was very interested to explore how MPC in this scenario can be used to help find an optimal balance of power usage from the grid, from the PV system, and when to do load curtailment/demand response, as well as the optimal times to charge the battery in this system.

Let's start by looking at the manipulated variables:

$$u(k) = [P\ grid\ (k), P\ batt\ (k), P\ load\ (k)]^T$$

is the vector of the control inputs, representing the amount of power input from the grid, from the battery, and the amount of power the system is drawing as a load, respectively. The following linear model represents the state of the system:

$$x(k + 1) = Ax(k) + Bu(k) + B_d d(k)$$

Where the state of charge (SOC) of the battery and supercapacitor are the state variables:

$$x(k) = [SOC\ batt\ (k), SOC\ SC\ (k)]^T$$

Interestingly, here, the solar panel power produced itself is actually modelled with ML-based predictions (another interesting example of hybrid modeling), and treated in the model as measured disturbances:

$$d(k) = P_{gen}(k):\ Disturbance\ (PV\ generation\ power)$$

For the input constraints, we have the state of charge as a percentage for both the battery and the supercapacitor:

$$20\% \le SOC_{batt}(k) \le 80\%$$

$$20\% \le SOC_{SC}(k) \le 80\%$$

And for the inputs, the power amounts (in kW):

$$-30 \le P_{load}(k) \le -5$$

$$-20 \le P_{grid}(k) \le 30$$

$$-10 \le P_{bat}(k) \le 5$$

The cost function used in the research paper is quite complex, but basically, it assigns weights for each of the control inputs through S_i, α_i, λ_i.

The full formulation for the MPC Optimization problem could be described as follows:

$$min\, J = \sum_{k=1}^{N_p} [S_1 (SOC\, batt\, (t + k) - SOC\, batt, ref)^2 + S_2 (SOC\, SC\, (t + k) - SOC\, SC, ref)^2]$$

$$+ \sum_{k=1}^{N_u} [\alpha_1 P^2 grid\, (t + k) + \lambda_1 \Delta P^2\, grid\, (t + k) + \alpha_2 P^2\, batt\, (t + k) + \lambda_2 \Delta P^2\, batt\, (t + k) \ldots$$

$$\ldots + \alpha_3 P^2\, load(t + k) + \lambda_3 \Delta P^2\, load(t + k)]$$

The goal being to minimize over the prediction horizon N_p and the control horizon N_u. The first term penalizes deviation of the state of charge of both the battery and supercapacitor from reference values, in order to keep the stored energy levels of both near a desired operating point. The second term penalized grid power, battery power, and load power respectively and their changes (ΔP) penalize the rate of change and ensure smooth transitions in power usage. The paper notes that the supercapacitor power is not controlled directly by the MPC (though it is used in the model) but a separate balance equation at the DC bus. That makes sense considering this is a hybrid energy storage system (HESS), where the battery provides high energy density storage, and the SC provides high power density storage, helping to extend the battery's lifespan.

One of the challenges in solving this optimization problem would be finding all the weights mentioned above. The study mentions that the weight values were discovered through extensive empirical testing, in order to establish a priority among the stated goals. It seems like in this research, they sought a balance between competing objectives, and it took some trial and error using MATLAB MPC design rules as a guide.

Hybrid Modeling Discussion

Based on the hybrid modeling discussion we had in class, as well as my experience and exposure so far with data science and machine learning, I see a couple of areas where there are limitations, as well as opportunities for improvement.

The first one that comes to mind is the amount and quality of data required to do machine learning. For certain chemical, or power generation, processes, there may be plenty of empirical data available both from research and from real-world applications. In that case, it's a matter of tracking down the data and acquiring access to it (which could be challenging, depending on whether or not the creators of the data have made it available to interested parties). Even if you can get the data, it may or may not be in the best shape, requiring further time with data engineering and preprocessing to be useful.

One idea to help mitigate this data problem is transfer learning. This is a technique we learned about in the Advanced Machine Learning class for processes such as image classification, where we may want to classify some images of our own but may not have access to vast high quality image data to train our model. In that case, we can apply transfer learning, using an available pre-trained neural network that is trained on a similar dataset (i.e., chemical process data), and fine tune it on a smaller dataset for our specific application.

Another key limitation discussed in the hybrid modeling seminar is the so-called "vanishing gradient" problem. It appears that neural networks trained with backpropagation can miss the influence of process variables that form a longer-term dependency, due to the small gradients making it difficult for the network to update weights effectively for this past data. One idea to address this is the use of long short term memory (LSTM) networks, that allow the network to better retain information over long time steps through memory cells and gating mechanisms. Another promising idea, as discussed in our lecture, was to incorporate attention mechanisms. Basically, borrow the multi-headed attention mechanism from the transformer architecture popular in the construction of large language models. This allows the model to focus on specific parts of the input data sequence, enabling the capture of relevant past data and longer term dependencies.