MPC-FMU Integration Model Version 2

Fred Fan
06/09/2024

The version 2 model has exactly the same setup as the version 1 model, which includes the main function in Julia (MPC_S V2.jl), the Julia Passive Model (PassiveModel.jl), the Julia MPC function (MPCAlgorithms_LBNL.jl), and the updated python script (FMU_Simulation_2). The updated FMU (Stanford_Hybrid_System_2.fmu) from Weiping is also uploaded into the same folder.

I haven't gotten a chance to update the Passive Model in Julia (especially the PCM SOC part that we talked about in the meeting on 06/04/2024), but I did manage to fix the FMU issue. The updated FMU from Weiping provided me with options to also include the PCM hot and cold temperatures as the outputs from FMU. Reading from the Github updates, it seems like the issue only exists when we try to call the FMU in iterations (in MPC, which technically is the only option), but the issue doesn't exist when we call the FMU only once but for the full simulation.
https://github.com/modelon-community/PyFMI/pull/109

Therefore, I just initialized the FMU at every iteration, either with the initial M_States declared at the beginning of the loop (global current_m_states = DataFrame(Indoor_Temp = [294.15], PCM_Hot_Temp = [273.15+48.9], PCM_Cold_Temp = [273.15+10], PCM_Hot_SOC = [1], PCM_Cold_SOC = [0.983103]), or with the M_States passed from the last iteration of the FMU. Basically we manually update the states for the FMU by initializing the FMU all the time.

The model is able to run smoothly in this manner. However, it does run into infeasible situations in the MPC. The issue is that the MPC would output a command (which in MPC's mind according to the reduced order model in Julia, executing such a command would maintain the indoor temperature within the set point boundaries), and the FMU will take such a command and simulate (the Modelica has a different physical model to the reduced order model in Julia, or rather worse, the Modelica has a different understanding of what that command actually does), and the output indoor temperature is now outside of the set point boundaries, which will make the next iteration of the MPC model infeasible (since in the MPC, the indoor temperature at all timesteps must be within the set point boundaries).

I can use an example to better explain this issue. For example, the MPC senses that although the indoor temperature is within the set point boundaries at the moment, it is on the lower end. Therefore, the MPC would like to discharge the PCM hot storage to increase the indoor temperature, which means that the output command is 3 (discharge PCM). However, Modelica would take the command=3 as input, and proceed to discharge the PCM cold storage instead, which will make the final indoor temperature lower than the lower set point, which will make the next iteration of MPC infeasible.

Therefore, the key is to understand what this means exactly: “3 = discharge TES (based on requests from zone)”. We have to make sure that the reduced order model that Julia MPC is using to make decisions is very similar to the Modelica model, with the aligned understanding of what a certain command would mean.

But in general, with the new format of FMU (always manually initialized), the MPC-FMU integration structure can run smoothly without the indoor temperature constraint. With the version 2 code, if you lower the set point temperature lower bound to 65 (from 68) Fahrenheit in the main code, it is able to run for 5 iterations without running into infeasible situations.

Feel free to reach out to me with any questions.