

Exercise 4

Parallel-connected Grid-Forming Inverters

M. K. Ranjram

1 Introduction

In the previous exercise we saw how a three-phase inverter could be controlled to transfer power from solar panels to the grid. We controlled the inverter as a current source in order to transfer a desired amount of real and reactive power with the grid, and we chose the real power setpoint in order to regulate the dc bus at the input of the inverter. Fundamental to this operation is the existence of a grid which we can synchronize to, and which establishes the voltage that we feed current into.

In this exercise, we'll turn our attention to grid-forming inverters. Here, the control freedom of the inverter will be used to regulate sinusoidal voltages of a desired frequency and magnitude at the point of common coupling (PCC). In other words, we will control our inverter to behave as a voltage source. This may at first seem straight-forward since the inverter produces ac voltages at its terminals which we control directly by defining the gating signals to its switches. However, our interest is in regulating voltage at the PCC, which is after the filter we use in the inverter, and therefore is not a voltage we control directly.

Our approach will be to employ an inner current control loop which regulates the currents out of the inverter according to an outer voltage control loop. Note that we also used this kind of inner/outer control structure in the previous exercise. There, the inner control loop also regulated the currents out of the inverter, but the outer voltage loop was used to control the dc bus voltage. Here, this outer loop will be used to control the voltage at the PCC. Note that we are using the full control capability of our inverter to regulate the PCC voltage. This means that some other part of our system must regulate the dc bus of the inverter. Furthermore, as mentioned in lecture, our inverter must have dispatchable power. If the input energy into the system was only provided by solar panels, we could not have this capability unless we purposefully operated off peak power at all times. An example way to achieve this dc bus and have dispatchable power would be for a battery energy storage system (BESS) to be connected to the bus, as described in lecture. In this exercise, we will assume that the dc bus of the inverter is a dc voltage source – in other words, we'll assume some other converter, like a BESS, is regulating this bus and providing dispatchable power.

2 Implementing a Grid-Forming Inverter in Simulink and Simscape

For your reference and convenience, an example grid-forming inverter implemented in Simulink accompanies this exercise¹. You'll notice that the implementation is different than in the

¹This simulation is adapted from the Simulink demo model entitled “Islanded Operation of an Inverter-based Microgrid Using Droop Control Technique”.

previous exercise. This is because this simulation employs the “Specialized Power Systems” blocks defined in Simscape. You can view these components by clicking Library Browser → Simscape → Electrical → Specialized Power Systems. In the previous exercise, the inverter was defined directly in the electrical environment of Simscape. The advantage of using the Specialized Power Systems blocks is that they naturally accommodate many common elements in power systems, such as the two-level three-phase inverter we have been studying, and convenient components such as defined power loads. It also produces signals that can be directly processed in Simulink. Note that this simulation is initialized via the ‘parameters’ script that accompanies the Simulink file. Double-click on the subsystem named “IBR 1 with droop control” and answer the following questions.

1. We now include a filter capacitor at the output of our inverter. Why do you think this capacitor is important, given that in the previous exercise we could omit it?
2. The inverter control is now done on a per-unit basis. This is so that we can use the same inverter subsystem to define other inverters in our overall system, even if those other inverters are rated for different power levels. Review the Simulink diagrams and write the formulas for P_{base} , V_{base} , Z_{base} and I_{base} in terms of the nominal power of the inverter P_{nom} and the nominal voltage at the PCC V_{nom} . *Note: I_{base} is computed inside the Measurements subsystem block.*
3. Open the Droop Control subsystem. Write out the formulas for the droop and voltage commands in terms of the reference frequency, nominal voltage, measured power, and measured reactive power. Note that the droop gains are entered as percentages in the initialization script. Draw the droop laws on a plot.
4. We saw in lecture that the droop slopes for each inverter should be scaled relative to their power rating to ensure correct power sharing. How do we implement this in this simulation?
5. The control is still done in the dq-frame, but this frame is established internal to the converter. Write down how the angle reference, ρ , is computed (this computation happens in the droop control subsystem).
6. Open the Measurements subsystem. How is the conversion to per-unit quantities implemented? Write out the corresponding formulas.
7. Open the Voltage Regulators subsystem. Write out the corresponding control laws. Note that the PI controllers employ “anti-windup”. This is a protective method that can help ensure good controller response in cases where the control action can saturate. When saturation happens, integration will cause the controller to “wind up” - creating memory around this saturated condition which you then have to remove once the saturation ends. Anti-windup prevents this by happening by removing the integral action once the output saturates.
8. Open the Current Regulator subsystem. You will notice that this control architecture is identical to the implementation in exercise 3. Why do you think this is the case?

9. Open the Vref Generation block. How is the averaged inverter gating command generated from the dq voltage command? Note that unlike our previous exercise, we do not implement an inverse dq block (dq-abc). This is because we are internally generating the PCC frequency, so we can just feed-forward this value when synthesizing our gating commands.
10. What signals are contained in the output port ‘meas’?

The inverter is parameterized by setting its filter values, its nominal power, and its control gains. As before, choosing the control gains is an essential part of ensuring that the system behaves well, but the selection of these gains is outside the scope of this exercise. For your reference, especially if you are familiar with PI controllers, you may try changing the gains and seeing the effect it has on the response of the system.

Our two-inverter system is created by placing two of these inverter blocks in parallel and connecting them to a set of loads. These loads represent the power draw of the larger system the inverters connect two. There is a fixed load and two dynamic loads which allow us to step-up and step-down the power demanded by the system. Open these blocks and examine how they work. You’ll notice that the second inverter is connected through an R-L impedance, this “tie line” accounts for the additional impedance that may exist between the two inverters. Note also that none of these loads establish the voltage of the system – we are relying on our inverters to do this.

3 System Performance and Trade-offs

Run the parameters script and then run the Simulink simulation.

1. Open the “Powers” scope. There are three events that happen at 5, 10, and 20 seconds. What are these events?
2. What is the difference between how the powers are split between the two inverters before and after 10 seconds? What is causing this difference? Does it match with your expectation?
3. Open the “droop commands” scope. What happens to the frequency of the two inverters after 10 seconds? At the end of the simulation, what do they settle to? Do you think it is a coincidence that they settle to this value, or are the controllers doing something special to enforce this?
4. What would we have to add to this system to make the inverters operate at their nominal set points again? *Hint: we talked about this in lecture.*
5. Why is it important that the two inverters settle to the same frequency? What would happen if they settled at different frequencies?
6. Try adding a “phase error” to the second inverter by introducing a fixed offset in the commanded value of ρ . Does this change the operation before droop control is enabled? Does the system behave after droop control is enabled, even with this inherent timing mismatch?

7. Try different converter parameter initializations, load power initializations, or droop control parameters. How do these affect the system's operation?

Inverters are essential for interfacing renewable energy resources such as wind and solar to the larger power system. Presently, many of these inverters operate in a “grid-following” manner, their control relying on a tightly regulated grid voltage. However, as these “inverter-based resources” become more widely deployed, and displace synchronous generators, they need to participate in supporting the grid. We can achieve this kind of functionality by employing “grid-forming” controls. This involves controlling the inverter as a voltage source, but also requires that they can operate in parallel autonomously. We have seen in this exercise a simple and popular method for achieving this capability – droop control. Interestingly, the core power electronic circuit used, the three-phase inverter, is the same as what can be used for grid-following controls. The key difference is in how the system is controlled.