

Exercise 3

A Grid-tied PV System with Grid-Following Control

M. K. Ranjram

1 Introduction

In the previous exercise we saw how a dc/dc converter enabled us to extract maximum power from a PV panel, and we also saw how the averaged model of a converter faithfully captures its key operating features and greatly simplifies system-level studies involving very different timescales (e.g. a 100kHz boost converter in a system where irradiance varies over 24 hours). In this exercise, we'll continue exploring the PV system shown in Fig. 1, now focusing on the inverter and the overall system-level operation and control.

In this exercise, we'll focus on understanding the inverter's key operational characteristics, exploring the difference between the switching and averaged models discussed in the lecture, and understanding how this inverter is controlled with sinusoidal pulse width modulation (SPWM) to deliver power from the solar PV system to the grid.

2 Implementing a Switching and Averaged SPWM Three-Phase Voltage-Sourced Converter in LTspice

For your convenience and reference, an example switching and averaged three-phase inverter modulated with sinusoidal pulse width modulation implemented in LTspice accompanies this exercise. Review this file and answer the following questions:

1. Review the gating circuitry and convert it into a block diagram.
2. Probe the gating signal $ga+$ and examine its harmonics. This can be done by clicking on the plot and the View→FFT→Okay. Set the y-axis to Linear (right-click on it). Observe the harmonic content. Where are the most significant harmonics?
3. Now plot the output voltage. Does the filter effectively neutralize the switching harmonics? Try operating with a lower switching frequency (by reducing parameter mf). When does the filter stop being filter?
4. If we wish to produce 480V line-line on the output (after the filter), what modulation index m_a should we use? Based on this, what is the absolute minimum dc voltage we can use at the input of the inverter to produce 480V line-line on the output?
5. Compare the line-line voltages produced by the averaged and switching models. Does the averaged model reasonably track the operation of the switching model?

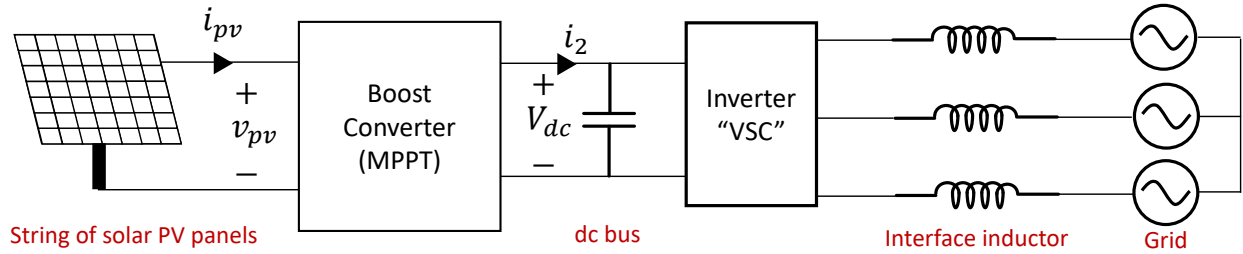


Figure 1: PV system studied in Exercises 2 and 3. The boost converter achieves MPPT of the PV strings, while the inverter (also called a “voltage-sourced converter” (VSC) or “voltage source inverter” (VSI)) delivers this power to the grid and regulates the dc bus. This exercise focuses on the boost converter and its MPPT operation.

3 Implementing Our System in Simulink and Simscape

LTspice is an effective tool for studying the operational and switching details of the three-phase inverter, SPWM, and the accuracy of our averaged model. However, while LTspice is a great tool for studying converters in detail, it is not as well-suited for system-level studies. To study our system, we’re instead going to employ Simulink and Simscape¹.

Simulink is a MATLAB toolkit which enables us to describe systems graphically in terms of block diagrams. It is a useful tool when we are interested in system-level interactions between components, such as between a converter and its controller, or between the multiple subsystems in Fig. 1.

The Simscape environment is an add-on to Simulink which allows us to describe our system in terms of the electrical equipment and electrical connections they employ. Simscape automatically converts these ‘physical’ models into the language of system-level equations used in Simulink. The alternative to using Simscape would be to determine a mathematical model of our system and then implement it as a block diagram in Simulink, which can be tedious, and gets less tractable as our systems become more complicated. Simscape provides models for common components employed in power systems and electronics, so it is well-suited for the system we are exploring in this exercise.

For your convenience, a Simulink file of our inverter system accompanies this exercise. A MATLAB script is used to initialize this simulation and is also included for your reference.

3.1 Simscape Model

The inverter system is modeled using Simscape components (in blue). We’ll start exploring this from “right to left” (from the grid connection to the dc bus).

1. Note that we connect directly to a fixed voltage source (the classic “infinite bus” model of the connecting network). This model is valid in cases where we assume that our

¹Other popular simulation tools for this purpose are “Power Systems Computer Aided Design” (PSCAD) and “Piecewise Linear Electrical Circuit Simulation” (PLECS), the latter of which can also be interfaced with Simulink. In contrast to LTspice, none of these other tools are free, so it can be advantageous to learn how to use LTspice and understand its strengths and limitations. For example, if you are primarily interested in the operation of a circuit with relatively simple control, or with a controller you will implement directly using electronics, rather than digital means, LTspice is an excellent choice.

exchanging power with the grid has little impact on it (in essence, we are so small as to not significantly impact the connecting network). This assumption is also consistent with operating as a grid-following inverter. The network is a voltage source that we will control the current into.

2. The grid and filter inductance are both modeled. Why is the filter capacitor irrelevant in this construction? Double-click on the filter inductor and observe that we are modeling its resistance and inductance within this component.
3. We implement the averaged three-phase inverter using a block provided by Simscape. This is another advantage of these toolboxes as they save us from having to construct such models manually. Nevertheless, it is important to understand how they operate and what their limitations are, which is why our previous exercises were important.
4. The dc bus is modeled by two series capacitors such that a neutral connection can be established between them. Simscape models require a special solver configuration block to be attached to them, and the system neutral is a natural place to connect this block.
5. As we discussed in lecture, the PV system is modeled as a controlled current source.
6. We provide two inputs to the physical Simscape model, a current command for the PV system and the averaged gating command for the inverter. We extract two outputs as well: the voltage at the point of common coupling (PCC) and the current into the PCC. We manipulate and manage these inputs and outputs using the Simulink environment, which requires us to use special “Simulink to Simscape” (and vice-versa) converters as you can see in the simulation.

3.2 PV Model

Now that we have a handle on the physical system model, we can turn our attention to the Simulink environment. First, we’ll discuss the PV model.

1. Simulink’s block diagram environment enables flexibility in our modeling choices. For your reference, we’ll leverage this flexibility to implement the PV model differently than previously discussed. Instead of implementing it directly as a circuit, we simply “code in” its PV characteristic using a MATLAB function. The “PVmodel” block does this, accepting an irradiance [W/m^2], temperature [$^{\circ}\text{C}$], and the voltage across the panel. Note that the modeling we performed is still necessary to parameterize this model.
2. Open the PV model block and examine the code – can you match this code to the modeling we did in Exercise 2? Note that the parameters of the PV system are implemented in the initialization script. These parameters are “passed” to the MATLAB function by defining them as parameters by clicking Function → Ports & Data Manager.

3. The MPPT tracker is also implemented using a function block. However, it is in a “triggered” subsystem. This allows us to model the fact that the MPPT is not applied in “real time” and is inherently on a slower update cycle (e.g. to capture variations in weather on the order of seconds). We also leverage the fact that the MPPT converter will be designed to regulate the commanded set point quickly relative to the changing value of the commanded power point. Thus, we don’t explicitly model this converter in the system-level simulation.
4. Open the MPPT block and review the code. Two advantages of writing out this algorithm in code are that it is easy to review and also can be directly mapped into digital control on an experimental prototype.
5. The MPPT outputs a new voltage setpoint. To avoid an “algebraic loop”, we include a delay between this output and the actual PV voltage. This highlights the convenience of employing a Simulink model, as it enables us to include discretization and memory ideas that will be important when we move to experimental prototyping.

3.3 Power Controller

The power controller sets the inverter’s gating command so that a desired active and reactive power is exchanged with the grid at the PCC. The operation of this controller was discussed in lecture, and we can review its key aspects.

1. We use Simulink-provided blocks to implement the PLL and Park (dq) transformation, again simplifying our study of the system by saving us from having to manually implement these blocks.
2. We also again employ MATLAB functions in our controllers. The relationship between the dq current into the PCC and the instantaneous power it processes was described in lecture and is employed directly as a MATLAB function block. Note again the relative ease of examining this algorithm by viewing it in code. Thus, from the measured PCC voltage and the P,Q commands, we can directly synthesize the values of the d- and q-axis currents that we need.
3. The current regulation block accepts these current references and outputs the inverter gate commands. This controller implements a discrete-time version of a popular controller called a “Proportional Integral (PI)” controller, which is widely used when the goal is to regulate setpoints or dc values. Because of the dq transformation, we turn our three-phase sinusoidal variables into quantities that are nominally dc. This allows us to use the PI controller, simplifying our design and enabling access to the rich toolset available for designing and tuning PI controllers. Note that Simulink includes discrete PI controller blocks which we can also use (and will use in the next exercise). However, the implementation in the MATLAB function is a useful reference to you as it resembles control code that would be entered into a digital controller. In particular, the PI is implemented using the Backwards Euler Method.

4. The performance of the controller is set by two numbers, K_i which is the gain on the integral component of the controller, and K_p which is the gain on the proportional component. Design of this controller is outside the scope of this exercise, but you may reach out to me if interested.
5. The controller also includes a “feed-forward” component which is the voltage drop on the filter inductor (which is between the output of the inverter, which we have direct control over, and the PCC, which we are trying to control). It is perhaps intuitive that including this information helps us to build a better controller. The details of the current controller gains and feed-forward impacts are outside the scope of this course, but it is a classic implementation and more information is available here [1] if you are interested.
6. The controller gate command is in the dq frame and must be returned to the abc domain before feeding it to the inverter. Note that this signal is saturated to prevent over-modulation of the inverter (i.e. the modulation index is always less than or equal to 1). Finally, a delay is included again to avoid an algebraic loop as discussed previously.
7. Note that while the power command is set by the bus voltage controller, the reactive power command remains controllable. This is key in ensuring our inverter can provide grid-support features. For this study, we will treat the reactive power setpoint as a defined operating condition. This can be changed while the simulation is running by changing the constant labeled ‘Q setpoint’.

3.4 Bus Voltage Controller

The bus voltage controller sets the inverter’s power command so that the bus voltage is tightly regulated to the desired value. It is simply a PI controller which acts on the measured dc voltage. Design of this controller is outside the scope of this exercise, but it should be recognized that the voltage controller should be inherently “slower” than the current controller, since that controller’s dynamics are part of the voltage control loop. Intuitively, if we are asking the power command to be changed faster than the current controller is able to regulate those commands, then those two controllers become coupled in a way that is likely to lead to instability.

4 System Performance and Trade-offs

Run the parameters script and then run the simulation. Note that the system initialization is somewhat slow (a few seconds) – this is because controllers are designed around nominal setpoints rather than for “startup” conditions. We are most interested in understanding how our inverter operates in these nominal conditions – we typically specially deal with startup separately.

Within a few seconds of simulation time, the bus voltage and the reactive power setpoint will be reached. Once it has, we can explore its performance:

1. Try changing the Q setpoint. Ideally the inverter will quickly respond to this change while not disturbing the power it's processing (equivalently, not disturbing the bus voltage). View the P,Q plots as well as the PV current/bus voltage plots. Is the converter and control resilient to changes in the commanded Q?
2. Now try changing the irradiance and temperature setpoints. We want the bus voltage to remain near its nominal value – does it? How does the change in irradiance of the PV system impact the power processed by the inverter? Does the PV's temperature have a strong impact on system operation?
3. More closely examine the impact of irradiance and temperature on the power processed by the PV system. After an irradiance change, how long does it take the MPPT to track the new maximum power point? How can we make this faster or slower? *Hint: the MPPT trigger clock and the voltage step on each perturbation can be set in the parameters script.* What affect does making MPPT faster or slower have on the bus voltage disturbance to a change in irradiance or temperature?
4. The bulk capacitor can be a dominant volume and cost contributor. For a given voltage rating, a smaller capacitance tends to be less costly (it is roughly a function of energy storage). What happens to the transient responses if you decrease the bus capacitance? Consider changes in irradiance, temperature, and reactive power setpoint. *Note: Don't forget, if the capacitance is too low we may violate our averaged-model validity! This model is valid only to the extent that ripple is negligible. Recall that exercise 2 gave us one path to connect voltage ripple with the size of this capacitor.*
5. If you are familiar with PI control tuning, you may also consider changing the response of the system by selecting new controller gains. Note that poor gain selection can lead to unstable system operation. It is a useful lesson to recognize that when this is happening, the problem may very likely be related to your specific controller gains, rather than there being a fundamental issue with your simulation file!

It is clear that while this converter does not support the grid, it certainly has other important duties it tends too – ensuring the PV system operates with an acceptable deviation of its dc bus level, shuffling the power produced by the panels to the grid, and offering the ability to provide reactive power support to the grid. This functionality comes with careful design of the inverter-PV system and by using the full current-controlling capability of our inverter. It is clear, then, that if we wish to develop an inverter that can *form* a grid we must develop an approach that is fundamentally different.

References

- [1] A. Yazdani and R. Iravani, *Voltage-Sourced Converters in Power Systems*. Hoboken, NJ, USA: John Wiley & Sons, Inc., Jan. 2010. [Online]. Available: <http://doi.wiley.com/10.1002/9780470551578>