VIRTUAL-IMPEDANCE CONTROL & COMPENSATION FOR GRID-CONNECTED INVERTER SYSTEMS

By

Allan R. Taylor Jr

A DISSERTATION

Submitted to
Michigan State University
in partial fulfillment of the requirements
for the degree of

Electrical Engineering –Doctor of Philosophy

2019

ABSTRACT

VIRTUAL-IMPEDANCE CONTROL & COMPENSATION
FOR GRID-CONNECTED INVERTER SYSTEMS

By

Allan R. Taylor Jr

Interface inductors are a commonly used coupling component for grid-connected DC/AC voltage-sourced power-electronics converters. They provide two main benefits: a voltage-sensing point for converter synchronization and, together with the grid impedance, they filter out the converter's voltage-switching-harmonics from influencing the grid-currents. Despite these benefits, the interface inductors add additional cost, weight, volume, and power losses to the system. Furthermore, the measurement of the synchronization voltage requires additional voltage sensors to be used, further increasing system cost and complexity. In recent years, much has been done to reduce the interface inductor's size and/or weight without compromising filtering performance, such as the use of different magnetic materials, planar windings, or higher converter-switching-frequencies. Also, voltage-sensor-less algorithms have been proposed, such as the Direct Power Control method (which is similar to the Direct Torque Control of induction machines). However, the reduction/elimination of these two items (inductors and sensors) are usually not considered together. Furthermore, many sensor-less control methods usually rely on knowledge of the source impedance, which may be difficult to estimate in grid-connected systems.

To eliminate the voltage sensors, the concept of virtual impedance can be employed to fabricate a synchronization point for the converter within software. Since the virtualized synchronization-voltage is based on information from already-available AC current sensors, the external voltage sensors can be removed from the system. In addition to synchronization,

additional virtual impedances or transfer functions can be fabricated to enhance the dynamic performance of the system, reduce computational complexity, and/or enhance the stability range. Lastly, if the AC source impedance alone is suitable to provide adequate harmonic filtering of the current (e.g. in a motor/generator connection) and the synchronization point is virtualized, the physical interfacing inductance can be completely removed from the system.

The main focus of this research work is to investigate the theory and implementation of using virtual impedances in DC/AC converter control systems. The self-synchronized inductor-less DC/AC converter system utilizing the concept of virtual-impedance is proposed. Also, a method of using only a virtual interfacing-resistance to alter the current control loop, eliminate cross-coupling terms, and reduce computational complexity is proposed. In addition, a method of virtual-impedance-compensating PLL algorithms with better transient response is proposed. Finally, the main application considered for this proposed control method is grid-connected systems, but an alternate virtual-impedance-based method for high-speed sensor-less control of permanent magnet AC machines is also proposed. A 1 kVA three-phase two-level inverter prototype has been designed to experimentally validate some of the proposed control strategies. A description of the experimental setup and experimental results are included within this report.

To my parents, Allan R Taylor and Blanca L Taylor.
Thank you for your constant support throughout my life and my academic career.

ACKNOWLEDGEMENTS

I would like to express my gratitude and thanks to my advisor, Dr. Fang Z. Peng for his guidance, his support, and for serving as my advisor, even after his departure from Michigan State University (MSU). I would also like to thank Dr. Elias Strangas, Dr. Bingsen Wang, Dr. Guoming Zhu, and Dr. Tim Hogan for their support and for serving as part of my committee.

I would like to extend special thanks to those at Kettering University (KU) who helped to make this research, as well as my Ph.D. career, possible. I would like to thank Dr. Jim MacDonald for his help enrolling me into the Faculty Doctoral Development Program at Kettering University. I would also like to thank Dr. Kevin Bai for his support and for the knowledge I gained from him during our years working in the Advanced Power Electronics Lab. I would also like to thank Dr. Mark Thompson, Dr. Karen Rogers, Dr. Dave Foster, Dr. Ken Kaiser, Dr. Ravi Warrier, Dr. Huseyin Hiziroglu, Dr. Doug Melton, Dr. Jim Gover, Dr. Mo Torfeh, and the rest of the KU ECE department faculty for their help, who have become my colleagues over the past few years. I would also like to thank Jerry Kozlowski, Amy Owens, and the rest of the KU ECE staff for their ongoing support.

Special words of thanks go to my colleagues within the MSU Power Electronics and Motor Drives Laboratory: Dr. Xiaorui Wang, Dr. Nomar Santini, Dr. Ujjwal Karki, Dr. Yaqub Mahnashi, Dr. Deepak Gunasekaran, and other lab members who have helped me over the years at MSU. I would also like to thank the MSU ECE and ERC department staff for their support.

Finally, I owe thanks to: my father, Allan Taylor, my mother, Blanca Taylor, my brother, David Robles, and especially my wife, Wallee Keating, whose love, support, and encouragement helped make my graduate studies and the completion of this dissertation possible.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

KEY TO ABBREVIATIONS

AC	Alternating Current
A2D	Analog-to-Digital
ADC	Analog-to-Digital Converter
APEL	Advanced Power Electronics Lab
ASIC	Application-Specific Integrated Circuit
CAN	Controller-Area Network
CMI	Cascaded Multi-Level Inverter
CSC	Current-Sourced Converter
DC	Direct Current
DSP	Digital Signal Processor
EEPROM	Electronically-Erasable Programmable Read-Only Memory
EV	Electric Vehicle
FPU	Floating-Point Unit
FLOP	FLoating-point OPeration
HEV	Hybrid-Electric Vehicle
HPF	High-Pass Filter
IC	Integrated Circuit
IC-PLL	Impedance Compensated Phase Locked Loop
KCL	Kirchhoff's Current Law
KVL	Kirchhoff's Voltage Law
KU	Kettering University
LPF	Low-Pass Filter

MCU	Micro-Controller Unit
MSU	Michigan State University
NPC	Neutral-Point Clamped
PCB	Printed Circuit Board
PEMDL	Power Electronics & Motor Drives Lab
PI	Proportional-Integral
PID	Proportional-Integral-Derivative
PR	Proportional-Resonant
PLL	Phase-Locked Loop
PV	Photo-Voltaic
PWM	Pulse-Width Modulation
RS-232	Recommended Standard 232
SiC	Silicon-Carbide
SOGI	Second-Order Generalized Integrator
SRF-PLL	Synchronous Reference Frame Phase-Locked Loop
TI	Texas Instruments
USB	Universal Serial Bus
VSC	Voltage-Sourced Converter

1. INTRODUCTION & BACKGROUND MATERIAL

In this chapter, the basics of pulse-width-modulated (PWM) DC/AC power-electronics converter systems are reviewed and the need for converter synchronization and harmonic filtering are discussed. Afterwards, the typical synchronization methods using a phase-locked loop (PLL) and current control methods using rotating-frame proportional-integral (PI) controllers are reviewed. Next, the stability, as well as the main benefits and issues with the conventional synchronization and harmonic-filtering methods using interface-inductors are summarized. The chapter concludes by introducing the key topic of this research, virtual impedance, as an alternate solution to converter synchronization and control. This technique can eliminate the need for external voltage sensors and enable inductor-less converter operation. In addition, the dynamic performance of a converter system can also be improved in some cases.

1.1. OVERVIEW OF GRID-CONNECTED INVERTERS

In the case of grid-connected energy systems, such as solar power generation systems or hybrid-electric-vehicle (HEV) battery chargers, a DC-AC power electronics converter and an interface inductor are typically placed between the energy system and a step-up transformer connected to the utility grid. In Figure 1.1, a typical solar three-phase inverter system is outlined. The inverter in such a three-phase system is typically a six-switch two-level voltage-sourced converter (VSC) but can be a multi-level design if higher phase-voltages and/or lower voltage-

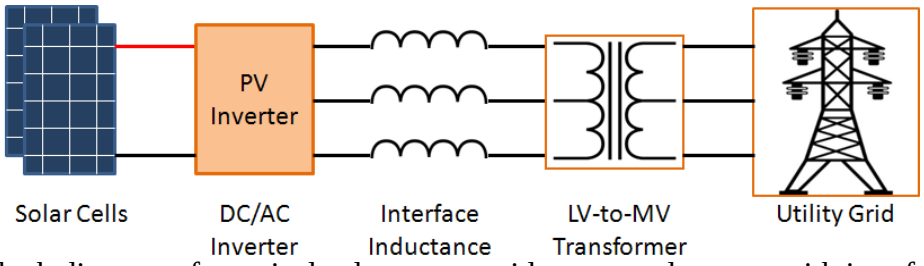


Figure 1.1: Block diagram of a typical solar power grid-connected system with interface inductor.

harmonics are needed. The transformer is used to step-up the low-voltages of the converter and energy system (typically around ~400 V) to higher utility-level voltages (around ~10 kV or so).

The interfacing inductors are placed between the converter and the transformer, primarily to block the high-frequency voltage harmonics of the converter from reaching the grid or the voltage-sensing point. In addition to the interface inductance, the leakage flux of the transformer and grid also helps to filter the line currents. This leakage flux effectively acts as an additional inductance in a series-connection with the interface inductance.

Both, a balanced three-phase and a single-phase DC/AC converter system can be modeled as an equivalent single-phase circuit with two AC voltage sources separated by an impedance; this is shown in Figure 1.2. The grid voltage, v_G, is modeled as an independent voltage source. The inverter output voltage, v_I, is modeled as a dependent voltage source; it is controlled to regulate the line current, i. The interface and transformer/grid resistances, denoted R_I and R_G, as well as the interface and transformer/grid inductances, denoted L_I and L_G, are placed in series between v_G and v_I. Applying KVL, the inverter's output voltage, v_I, can be written as following:

$$\vec{v}_I = \left(R_I + R_G\right)\vec{i} + \left(L_I + L_G\right)\frac{d\vec{i}}{dt} + \vec{v}_G. \tag{1.1}$$

The mid-point voltage in Figure 1.2, v_S, is sensed by external AC voltage sensors, to be used within the control algorithm in a phase-locked loop. This is used for synchronization of the converter system, to be discussed later. The DC-side of the inverter will be ignored; it will be considered to be a constant DC voltage value and will not influence the AC-side control.

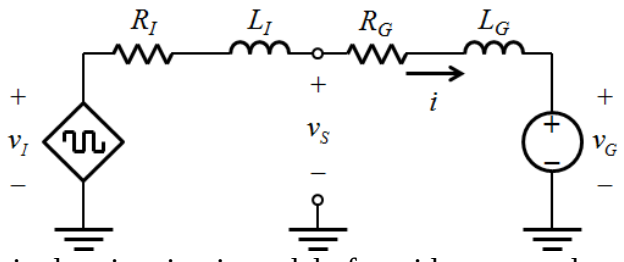


Figure 1.2: Equivalent single-wire circuit model of a grid-connected power electronics system.

1.1.1. THE NEED FOR SYNCHRONIZATION

The utility grid voltage is usually assumed to be sinusoidal – it is typically generated by the back-electro-motive force (EMF) of large rotating electrical machines driven by turbines. To enable a non-zero average power flow from the utility grid, we can deduce that the line currents must also be sinusoidal. We can argue this from examining the following average power expression; assuming $v_G = V_{Gm}\sin(\omega_v t + \phi_v)$ and $i = I_m \sin(\omega_i t + \phi_i)$, the average power, P, can be expressed as the sum of two cosine terms (after applying a trig-substitution) as shown:

$$P = \frac{1}{T}\int_0^T (v_G\, i)\, dt = \frac{\frac{1}{2}V_{Gm}I_m}{T}\int_0^T \left[\cos(\omega_v t + \phi_v - \omega_i t - \phi_i) - \cos(\omega_v t + \omega_i t + \phi_v + \phi_i)\right] dt, \qquad (1.2)$$

where ω_v and ω_i; V_{Gm} and I_m; and ϕ_v and ϕ_i are the frequencies, peak magnitudes, and phase-shift angles, respectively, of the grid voltage and current, respectively. In the case that the two frequencies are equal, i.e., $\omega_v = \omega_i = \omega$, the power expression simplifies to a non-zero value:

$$P = \frac{\frac{1}{2}V_{Gm}I_m}{T}\int_0^T \left[\cos(\phi_v - \phi_i) - \cos(2\omega t + \phi_v + \phi_i)\right] dt = \tfrac{1}{2}V_{Gm}I_m\cos(\phi_v - \phi_i). \qquad (1.3)$$

In the case that the frequencies are not equal, $\omega_v \neq \omega_i$, the integration in the power expression above will equal zero and there will be no average power transfer from the grid.

Relaxing the definition of the current waveform to be any periodic shape, we can argue based on the Fourier Series expansion of the current, i, that all other harmonics of current with frequencies not equal to the voltage frequency, ω_v, will not produce any net average power. Therefore, the line current must also be sinusoidal with the same frequency as the grid. Combining this conclusion with equation (1.1), we can determine that the inverter voltage, v_I, must also be sinusoidal for non-zero power flow. Therefore, the inverter output voltage must be synchronized to the grid frequency for proper operation.

1.1.2. THE NEED FOR INDUCTIVE FILTERING

From equation (1.1), we can solve for the line-current in the Laplace domain, yielding the following first-order differential-equation result:

$$\vec{I} = \frac{\vec{V}_I - \vec{V}_G}{s\left(L_I + L_G\right) + \left(R_I + R_G\right)} = \frac{1}{\left(L_I + L_G\right)} \frac{\vec{V}_I - \vec{V}_G}{\left(s + \frac{\left(R_I + R_G\right)}{\left(L_I + L_G\right)}\right)}. \tag{1.4}$$

The inductive impedances essentially function as a low-pass filter for the line-currents. The cut-off frequency, ω_C, of the expression for the current above is given as the following:

$$\omega_C = \frac{R_I + R_G}{L_I + L_G}. \tag{1.5}$$

Thus, any high-frequency components of the inverter voltage or grid voltage above ω_C will be attenuated in the current. The Bode magnitude plot of the current is shown in Figure 1.3 (a). The actual magnitude of the output current depends on the magnitudes of the applied voltages multiplied by the gain values given by the expression above (as shown in the Bode plot).

The inverter's "sinusoidal" output voltage is generated through the use of PWM. A partial frequency spectrum of a sine-PWM voltage waveform is shown below in Figure 1.3 (b). Assuming the fundamental grid frequency to be f_g, the PWM voltage will contain a desired low-frequency component with frequency f_g and magnitude $\approx V_{Im}$, as well as several undesired

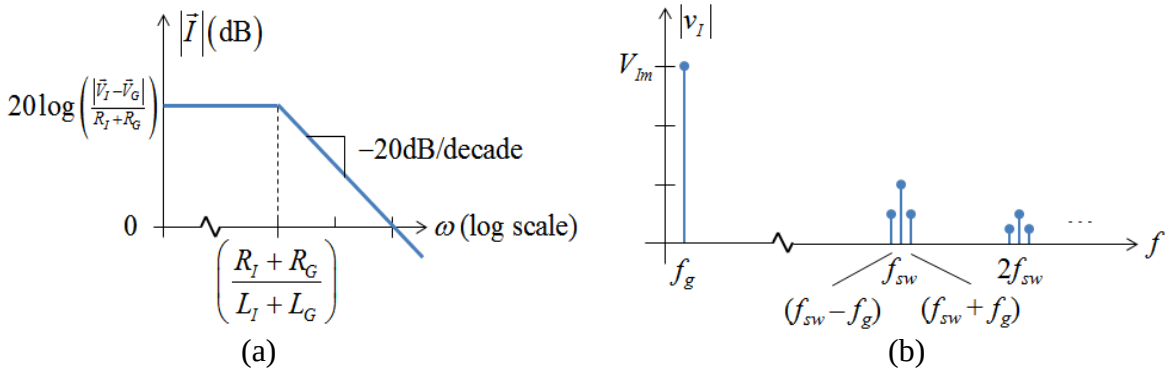


Figure 1.3: Magnitude vs. frequency responses: (a) Bode magnitude plot of grid current; (b) simplified frequency spectrum of PWM voltage waveform.

voltage harmonics at the switching frequency, f_{sw}, at the side-band frequencies, ($f_{sw} \pm f_g$), and at multiples of the switching and side-band frequencies (as well as other, smaller harmonics).

If the cut-off frequency of the current due to the impedances, ω_C, is well below the converter switching frequency, f_{sw}, then the high-frequency components of the voltages will be mostly filtered from the current. However, the magnitude of these high-frequency harmonics will not be zero; some small ripples will remain. These high-frequency current ripples are undesired, since they do not contribute to any active power transfer (as explained previously). In addition, any large *di*/*dt* along the utility lines can cause electro-magnetic interference (EMI) issues with other devices connected to the grid. The relative amount of ripple currents as compared to the desired fundamental current is typically quantified by the Total Harmonic Distortion (THD) value; one general definition of THD for current is given below, where each I_k is a k^{th} harmonic current:

$$THD_i = \frac{\sqrt{I_{2,\text{rms}}{}^2 + I_{3,\text{rms}}{}^2 + \cdots}}{I_{1,\text{rms}}{}^2} = \frac{\sqrt{\sum_2^\infty I_{k,\text{rms}}{}^2}}{I_{1,\text{rms}}{}^2}. \tag{1.6}$$

A lower THD value is generally better; a THD value of zero indicates a purely sinusoidal signal. The inductance value of the interface inductors is usually selected to limit the total harmonic distortion (THD) of the grid current to an acceptable level, usually below 5% THD or less [1]. As seen from equation (1.5), a larger inductance value will lower the cut-off frequency, providing a better filtering of harmonics.

Finally, one additional need for filtering the voltage harmonics using interfacing impedances (in addition to eliminating potential EMI issues and zero-power-producing currents) is to provide a clean voltage-sensing point for converter synchronization. Measurement of the grid voltage is traditionally used for synchronization and control of the inverter output voltage, as mentioned previously. The voltage-sensing point, v_S, is typically measured at the transformer low-voltage

(LV) side using galvanically-isolated voltage sensors. Ideally, the sensed voltage should be free of harmonics, containing only the fundamental AC voltage sinusoidal waveform. If the interfacing inductance is too small, it is possible that some of the high-frequency PWM voltage will appear at the sensing mid-point voltage, v_S. If left unchecked, this can add extra high-frequency noise into the inverter control system, possibly causing instability due to positive feedback through feed-forward-like paths within the control loop.

1.2. CONVERTER SYNCHRONIZATION USING A PLL

The synchronization of the inverter's voltage is typically done using a Phase-Locked Loop (PLL) algorithm. Some non-PLL-based converter synchronization methods exist, such as a self-synchronized synchronverter [2] and the second-order-generalized-integrator (SOGI) algorithm [3], but these will not be covered here.

A PLL is essentially a feedback control system which tries to produce an exact replica of a periodic input signal, or one of its harmonics, excluding noise. They can be hardware-based or software-based, usually implemented within a Digital Signal Processor (DSP). The input and output signals of a PLL are periodic (they can be analog or digital), where the output signal is phase-aligned with the input signal. The PLL typically consists of three main pieces [4-6]:

- **Phase Detector:** this is usually a mathematical expression which produces an error signal, ε_θ, related to the phase difference between the input and output signals. In a digital PLL, the phase detector may be as simple as an XOR gate. In an analog PLL, it may be the multiplication of two signals or another non-linear mathematical calculation.

- **Filter/Controller:** a typical PLL might have a low-pass filter and controller following the phase detector (note: some controllers have a similar form to a low-pass filter). Their purpose

is to produce an estimate of the frequency, $\hat{\omega}$, of the input signal which drives the phase-error to zero. Note that this frequency may "speed up" or "slow down" to cause the input and output signals to eventually match in phase. When the phase error (and its derivative) reaches zero, the frequency will stabilize and the PLL is said to be "locked" to the input signal.

- **Oscillator:** sometimes labeled as a VCO (voltage-controlled oscillator), this is where the "copy" of the periodic signal is generated, based on the determined frequency estimate, $\hat{\omega}$, found earlier. The oscillator could be a digital clock circuit, or it could be an integrator and a trigonometric sin(·) function. The output is usually fed back to the phase-detector block.

The general block diagram of a PLL is shown below in Figure 1.4, consisting of the three main components described above. The PLL's main purpose in a DC/AC converter is to track the frequency, phase angle, and the magnitude of the AC grid voltage. These values are used within the control algorithm to derive the inverter's output voltage and perform current regulation. In addition, the use of a PLL can help to reduce noise coming from the terminal voltage measurements, allowing clean feed-forward-like signals to be used within the control algorithm. The input to the PLL in this study will be the mid-point measurement voltage, v_S, which may contain harmonics. However, the output of the PLL (ideally) will only contain the fundamental component of the measurement voltage. The PLL can also help make the control system more immune to disturbances in the grid, as the bandwidth of a PLL is generally quite narrow.

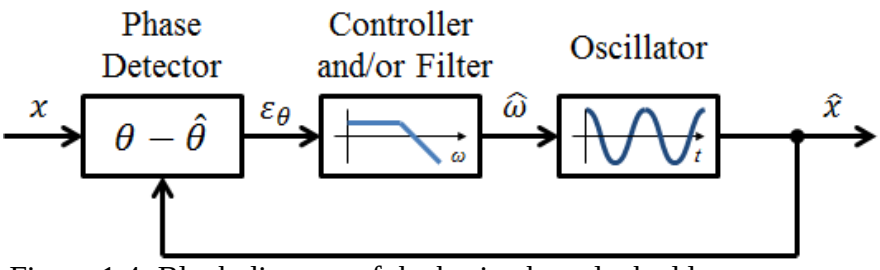


Figure 1.4: Block diagram of the basic phase-locked loop structure.

One of the most commonly used PLL topologies is the synchronous reference frame (SRF) PLL [7]. It is more commonly used in three-phase converter systems but can be implemented in single-phase systems as well with minor adjustments [8]. The SRF-PLL obtains its name from its use of the Clarke and Park coordinate transformations, used to convert a sinusoidal (rotating) three-phase system into a stationary two-phase system by means of using a rotating reference frame (hence the name). The Clarke and Park transforms, respectfully, are presented below:

$$\begin{bmatrix} v_{S\alpha} \\ v_{S\beta} \\ v_{S\gamma} \end{bmatrix} = \frac{2}{3}\begin{bmatrix} 1 & -\frac{1}{2} & -\frac{1}{2} \\ 0 & \frac{\sqrt{3}}{2} & -\frac{\sqrt{3}}{2} \\ \frac{1}{2} & \frac{1}{2} & \frac{1}{2} \end{bmatrix}\begin{bmatrix} v_{Sa} \\ v_{Sb} \\ v_{Sc} \end{bmatrix}, \tag{1.7}$$

$$\begin{bmatrix} v_{Sd} \\ v_{Sq} \end{bmatrix} = \begin{bmatrix} \cos(\theta) & \sin(\theta) \\ -\sin(\theta) & \cos(\theta) \end{bmatrix}\begin{bmatrix} v_{S\alpha} \\ v_{S\beta} \end{bmatrix}, \tag{1.8}$$

where v_{Sa}, v_{Sb}, and v_{Sc} are the three-phase stationary-frame inputs, $v_{S\alpha}$ and $v_{S\beta}$ are the two-phase stationary-frame signals, $v_{S\gamma}$ is the mean value, and v_{Sd} and v_{Sq} are the rotating-frame outputs. The angle, $\hat{\theta}$, is the estimated angle of the rotating reference frame. Note the use of the "magnitude-invariant" version of the Clarke transform here (with a ⅔ out front). Since the PLL operates in the rotating *dq*-frame, it is also sometimes referred to as the *dq*-PLL topology.

A block diagram of the basic SRF-PLL is shown in Figure 1.5 on the next page. The Park transform effectively serves as the phase detector here (and could also be considered part of the oscillator). The error signal, ε_θ, in this case (which we wish to drive to zero) is simply the *q*-axis value of the sensed voltage, v_{Sq}. The controller shown in the figure, used to drive the error signal to zero, is simply a Proportional-Integral (PI) controller. However, other controllers, such as a Proportional-Integral-Derivative (PID) controller, can also be used. From the estimated

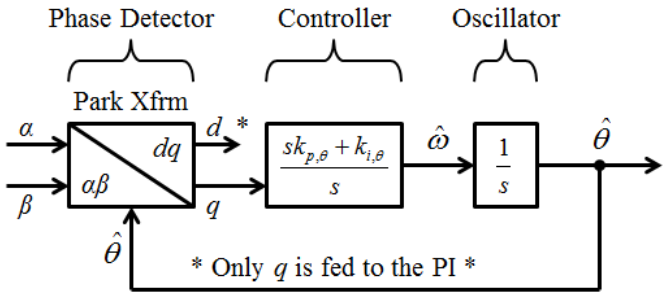


Figure 1.5: Block diagram of the Synchronous Reference Frame PLL.

frequency, $\hat{\omega}$, the oscillator-portion of the PLL generates the phase-angle estimate, $\hat{\theta}$, through an integration. The phase-angle estimate is then fed back to the Park transformation.

The inputs to the PLL, the variables v_{Sa}, v_{Sb}, and v_{Sc} in equation (1.7), are the three-phase components of the sensed mid-point voltage, v_S. Assuming the three-phase components are balanced, the $\alpha\beta$-components of the sensed mid-point voltage can be expressed as

$$\begin{bmatrix} v_{S\alpha} \\ v_{S\beta} \end{bmatrix} = \begin{bmatrix} V_{Sm}\cos(\theta_S) \\ V_{Sm}\sin(\theta_S) \end{bmatrix}, \tag{1.9}$$

where V_{Sm} and θ_S are the magnitude and phase angle of the sensed voltage, respectively. We ignore the γ-term here, assuming it to be zero. Ideally, if the angle estimate of the PLL is correct, the Park transform outputs should produce the values $v_{Sd} = V_{Sm}$ and $v_{Sq} = 0$. We assume that the voltage will be aligned to the d-axis of the rotating coordinate system (as opposed to the q-axis, as done in motor control theory).

If we examine the Park transformation output terms more closely, we can see that the error signal (the v_{Sq}-component) can be expressed as

$$v_{Sq} = -\sin(\theta)v_{S\alpha} + \cos(\theta)v_{S\beta}, \tag{1.10}$$

which, after plugging in equation (1.9) and a bit of simplification, we can express v_{Sq} as the sine of the angle-error, scaled by a constant:

$$v_{Sq} = V_{Sm}\sin(\theta_S - \theta). \tag{1.11}$$

If the angle-error is small, we can use the Taylor-series approximation of sin(x) near $x = 0$ to linearly approximate the q-axis voltage as:

$$v_{Sq} \approx V_{Sm}\left(\theta_S - \theta\right). \tag{1.12}$$

Thus, if the grid voltage (and measurement voltage) has a relatively constant magnitude, then the q-axis voltage is approximately equal to the angle error, multiplied by a constant. In a simple scenario, the voltage magnitude constant can be absorbed into the PI-controller coefficients, $k_{p,\theta}$ and $k_{i,\theta}$. A more proper way to correct for the gain is to simply divide it out. Note that the voltage magnitude can be computed using the dq-frame components:

$$V_{Sm} = \sqrt{{v_{Sd}}^2 + {v_{Sq}}^2}. \tag{1.13}$$

Thus, the normalized angle-error can be expressed as:

$$\varepsilon_\theta = \left(\theta_S - \theta\right) \approx \frac{v_{Sq}}{\sqrt{{v_{Sd}}^2 + {v_{Sq}}^2}}. \tag{1.14}$$

The overall normalized SRF-PLL block diagram is shown in Figure 1.6 below.

To tune the PI-controller of the SRF-PLL, we must first linearize the model. The sine and cosine terms inside of the Park transformation make the system nonlinear – we cannot directly apply feedback control theory to nonlinear systems. As described above in equation (1.14), the angle error is approximately being applied to the input of the PI-controller. Therefore, we can simplify the PLL feedback control diagram to remove the Park transform, hypotenuse calculation, inverse, and multiplication blocks and replace them with simply the angle-error. This

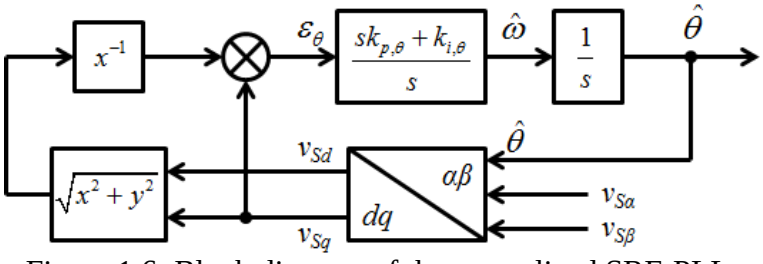


Figure 1.6: Block diagram of the normalized SRF-PLL.

is shown in Figure 1.7 below. With this linearized model, we can analyze the dynamics.

The closed-loop transfer function of the linearized SRF-PLL algorithm is given as:

$$TF_{PLL} = \frac{\hat{\theta}(s)}{\theta_S(s)} = \frac{sk_{p,\theta} + k_{i,\theta}}{s^2 + sk_{p,\theta} + k_{i,\theta}}. \tag{1.15}$$

Examining the denominator of the transfer function, we can determine that the PLL will always be stable so long as both coefficients are positive. If we compare the denominator to the standard form of a second-order system,

$$s^2 + 2\zeta\omega_n s + \omega_n{}^2 \quad \Leftrightarrow \quad s^2 + k_{p,\theta}s + k_{i,\theta}, \tag{1.16}$$

and apply the basic theory of second-order systems, we can compute the settling time, $T_{st,\theta}$, of the PLL and from it, determine an expression for the $k_{p,\theta}$ coefficient:

$$T_{st,\theta} = \frac{4}{\zeta\omega_n} = \frac{4}{\frac{1}{2}k_{p,\theta}} \quad \Rightarrow \quad k_{p,\theta} = \frac{8}{T_{st,\theta}}. \tag{1.17}$$

By examining the roots of the denominator polynomial,

$$s = -\tfrac{1}{2}k_{p,\theta} \pm \sqrt{k_{p,\theta}{}^2 - 4k_{i,\theta}}, \tag{1.18}$$

we can also obtain an expression for the $k_{i,\theta}$ coefficient to make the system critically damped, having little or no overshoot (ideally). By setting the expression under the radical equal to zero, we can solve for the $k_{i,\theta}$ coefficient:

$$0 = k_{p,\theta}{}^2 - 4k_{i,\theta} \quad \Rightarrow \quad k_{i,\theta} = \tfrac{1}{4}k_{p,\theta}{}^2. \tag{1.19}$$

Note: because the transfer function contains a first-order numerator (a zero), some overshoot may still occur even when the system has been made critically damped. Typical values for the

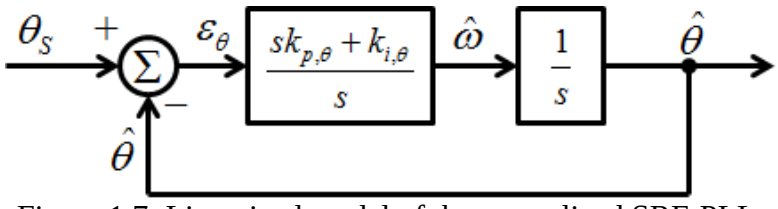


Figure 1.7: Linearized model of the normalized SRF-PLL.

settling time of the PLL system are usually in the 10's or 100's of milli-seconds. Usually, the dynamics of the PLL are made to be much slower than other system dynamics (such as the current), so that the two systems may be analyzed independently of each other.

While this PLL theory presented here proves to be a simple analysis, it is not the complete picture. It provides only an initial starting point for analyzing the converter's control system. To obtain a more accurate understanding of the stability and dynamics, the effects of the current control loop and/or the circuit impedances must also be taken into account.

1.3. LINE-CURRENT REGULATION

The control of the grid current in a three-phase DC/AC converter system is commonly done using proportional-integral (PI) controllers in each axis of the synchronous frame. The line-currents are usually measured using isolated analog Hall-effect sensors. The sampled line currents are converted to the synchronous frame using the Clarke and Park transformations, together with the observed angle-output of the PLL. During normal operation, the output voltage of the inverter is controlled to be synchronous with the grid and its fundamental magnitude and phase are adjusted to be slightly different than that of the grid. This difference in voltage shows up across the interfacing impedance and drives a current through the system; this is how the controller is able to source or sink the desired active and/or reactive power. In addition to control, the sensing of the AC line currents is also for protection purposes.

To eliminate the any effects of the transformer or grid impedances affecting the current, the sensed terminal voltage, v_S, is usually passed through the converter in a feed-forward-like path. In this sense, the control algorithm is focused solely around the interfacing impedance. To show this mathematically, recall the simplified equivalent circuit diagram (Figure 1.2) of the converter-grid system. Recall that the same line current, i, flows through both impedances. Thus,

the PLL voltage, v_S, will be influenced by not only the grid voltage and the commanded inverter voltage, but also the relative impedance values. Using the Superposition principle, we can express the PLL voltage in the phasor Laplace domain as:

$$\vec{V}_S = \frac{Z_I}{Z_I + Z_G}\vec{V}_G + \frac{Z_G}{Z_I + Z_G}\vec{V}_I, \tag{1.20}$$

where the quantities $Z_I = R_I + j\omega L_I$ and $Z_G = R_G + j\omega L_G$ are the simplified interface and grid impedances, respectively. The terminal voltage dynamics are governed by the voltage divider which is formed from the interface impedance and transformer impedance. By reworking equation (1.20) and applying Ohm's Law, we can rewrite the PLL voltage as:

$$\vec{V}_S = \left(\vec{V}_G + Z_G\vec{I}\right) = \left(\vec{V}_I - Z_I\vec{I}\right). \tag{1.21}$$

The same results can also be obtained from applying KVL. From this, we can see that the three voltages, v_I, v_S, and v_G, will be nearly equal at light loads, but will differ at heavy loading conditions. The fact that the two voltages, v_I and v_G, both influence the PLL input voltage will play an important role in the stability of the converter control algorithm, as we will later see. Also, to ensure that KVL equation (1.21) holds, we can conclude that the converter's output voltage should be controlled to be equal to the following:

$$\vec{V}_I = \vec{V}_S + Z_I\vec{I}. \tag{1.22}$$

This equation shows that the inverter voltage is nothing more than the sum of the sensed mid-point voltage and the voltage drop across the interfacing impedance. The converter output voltage does not depend upon the transformer/grid impedance or the grid voltage in this case.

Considering only the interfacing impedance, we can rewrite equation (1.22) in a space-vector notation and expand the impedance terms:

$$\vec{v}_{I,\alpha\beta} = R_I\,\vec{i}_{\alpha\beta} + L_I\,\frac{d\,\vec{i}_{\alpha\beta}}{dt} + \vec{v}_{S,\alpha\beta}; \tag{1.23}$$

the $\alpha\beta$-subscripts denote the space vectors are in the stationary frame. Next, we can apply the space-vector form of the Park transformation, moving the inverter voltages to the *dq*-frame:

$$\vec{v}_{I,dq} = \left[R_I \, \vec{i}_{\alpha\beta} + L_I \frac{d\,\vec{i}_{\alpha\beta}}{dt} + \vec{v}_{S,\alpha\beta} \right] e^{-j\hat{\theta}}. \tag{1.24}$$

Simplifying this will give a result similar to the classical result from motor control theory. Moving differential-terms from one reference-frame to another results in extra terms due to the Product Rule of Calculus, (or in this case, the Frequency Shifting property of Laplace transforms might be more applicable):

$$\vec{v}_{I,dq} = R_I \, \vec{i}_{dq} + L_I \frac{d\,\vec{i}_{dq}}{dt} + j \frac{d\hat{\theta}}{dt} L_I \vec{i}_{dq} + \vec{v}_{S,dq}. \tag{1.25}$$

The above equation can be simplified and split in to real and imaginary parts, yielding

$$\begin{cases} v_{Id} = R_I \, i_d + L_I \dfrac{d\,i_d}{dt} - \hat{\omega} L_I \, i_q + v_{Sd}, \\ v_{Iq} = R_I \, i_q + L_I \dfrac{d\,i_q}{dt} + \hat{\omega} L_I \, i_d + v_{Sq}, \end{cases} \tag{1.26}$$

where $\hat{\omega} = d\hat{\theta}/dt$ is the estimated grid-frequency. The main difference here from motor control theory is that the "back-EMF" term of the grid (v_{Sd} in this case) is assumed to be in the *d*-axis rather than the *q*-axis. At steady-state, the value of v_{Sq} should be equal to zero.

From this result, the classical synchronous-frame current control scheme is derived. This is shown in Figure 1.8 on the next page. The *dq*-frame currents are decoupled from each other using a direct calculation of the cross-linked voltages. The pseudo-feed-forward path for the sensed *d*-axis voltage is shown at the top of the figure – this voltage is directly added to the inverter's output voltage. An equivalent pseudo-feed-forward voltage is also added to the *q*-axis output to help during transient events (although the *q*-axis voltage should be zero at steady-state). We call these "pseudo" feed-forward signals since they technically are voltages fed back from

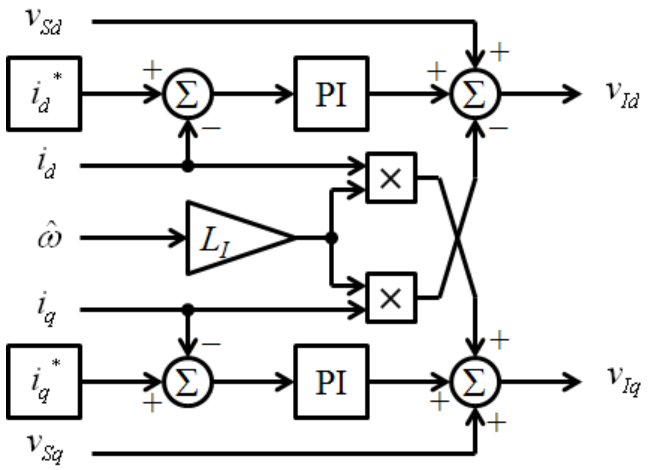


Figure 1.8: Classical *dq*-frame current controller with decoupling terms.

voltage-sensor measurements, but are not directly related to the currents in a stiff-grid condition. The only terms which are not compensated for are the resistor and inductor voltage-drop terms. These are left to the PI-controllers to generate the appropriate error voltages to cancel these remaining voltage-terms and determine the system dynamics.

The equivalent system as seen by the PI-controllers after decoupling is nothing more than a series RL-load. Therefore, the equivalent control system block diagram for both axes will be as in Figure 1.9. The closed-loop transfer function of the equivalent PI-control loop is given below:

$$TF_{DQ} = \frac{I(s)}{I^*(s)} = \frac{\frac{k_{p,c}}{L_I}\left(s + \frac{k_{i,c}}{k_{p,c}}\right)}{s^2 + s\left(\frac{R_I + k_{p,c}}{L_I}\right) + \frac{k_{i,c}}{L_I}}. \tag{1.27}$$

Since both axes will be symmetric ($L_d = L_q$), usually the same tuning parameters can be used for both PI-controllers. Examining the denominator of the transfer function, we can determine that

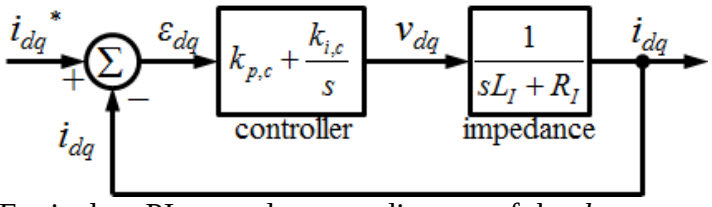


Figure 1.9: Equivalent PI-control system diagram of the *dq*-current controllers.

the control system will always be stable so long as both PI-coefficients are positive (the interface impedance values should always be real and positive). If we compare the denominator of the transfer function to the standard form of a second-order system,

$$s^2 + 2\zeta\omega_n s + \omega_n{}^2 \qquad \Leftrightarrow \qquad s^2 + \left(\frac{R_I + k_{p,c}}{L_I}\right)s + \frac{k_{i,c}}{L_I}, \tag{1.28}$$

and apply the basic dynamic results of second-order systems, we can compute the settling time, $T_{st,c}$, of the *dq*-currents and from it, determine an expression for the $k_{p,c}$ coefficient:

$$T_{st,c} = \frac{4}{\zeta\omega_n} = \frac{4}{\left(\frac{R_I + k_{p,c}}{2L_I}\right)} \qquad \Rightarrow \qquad k_{p,c} = \frac{8L_I}{T_{st,c}} - R_I. \tag{1.29}$$

By examining the roots of the denominator polynomial,

$$s = -\frac{R_I + k_{p,c}}{2L_I} \pm \sqrt{\left(\frac{R_I + k_{p,c}}{2L_I}\right)^2 - \frac{k_{i,c}}{L_I}}, \tag{1.30}$$

we can also obtain an expression for the $k_{i,c}$ coefficient to make the system critically damped (setting the radical expression equal to zero):

$$0 = \left(\frac{R_I + k_{p,c}}{2L_I}\right)^2 - \frac{k_{i,c}}{L_I} \qquad \Rightarrow \qquad k_{i,c} = \frac{\left(R_I + k_{p,c}\right)^2}{4L_I}. \tag{1.31}$$

Note: because the transfer function contains a first-order numerator (a zero), some overshoot may still occur even when the system has been made critically damped. Typical values for the settling time of the *dq*-current control system are usually in the single-milli-second range. Usually, the dynamics of the current are the fastest in the whole control system.

1.4. CONTROL-SYSTEM STABILITY ANALYSIS

To truly gain an insight into the stability of the overall inverter control system, we must consider both the PLL and the *dq*-control loops together. In addition, we must also consider the

dynamics caused by the source and interface impedances. Unfortunately, this makes the traditional transfer-function approach very difficult to use in this case, since the overall control system will be of very high order. In addition, relating the impedances, *dq*-control, and PLL together requires shifting the mathematics into different coordinate reference-frames using the Park transformation. The use of trigonometric quantities within the control loop will make the system non-linear, which voids the use of traditional feedback-control theory. The cross-linking terms which result from the use of rotating reference-frames will also pose additional difficulty, as we will essentially have two dynamic systems which interact with each other. Because of these complexity and non-linearity issues, an alternate approach is required.

As a tradeoff, rather than analyzing the stability of a non-linear system over all of the system's operating ranges, we can linearize the system at an equilibrium point and then analyze the stability of just the equilibrium point. To do this, we use an approach similar to a state-space representation; instead of working with a single higher-order differential equation, suppose that we express the dynamics of the control system as a set of ordinary 1st-order differential equations. Written in vector form, we have:

$$\frac{d\vec{x}}{dt} = \mathbf{f}\left(\vec{x}\right), \tag{1.32}$$

where $\mathbf{f}(\cdot)$ is the array of differential equations and $\vec{x}$ is an array of state variables. Suppose that a particular set of state variable values, $\vec{x}^*$, is an equilibrium point of the system. By definition, we can say that $\mathbf{f}(\vec{x}^*) = \mathbf{0}$. Now, by taking the Taylor-series expansion of the right-hand side of equation (1.32) at an equilibrium point gives us the following:

$$\frac{d\vec{x}}{dt} = \left[\mathbf{f}\left(\vec{x}^*\right) + \left.\frac{\partial \mathbf{f}}{\partial \vec{x}}\right|_{\vec{x}^*}\left(\vec{x} - \vec{x}^*\right) + \cdots\right] = \left[\mathbf{0} + \left.\frac{\partial \mathbf{f}}{\partial \vec{x}}\right|_{\vec{x}^*}\left(\vec{x} - \vec{x}^*\right) + \cdots\right]. \tag{1.33}$$

The 1st-order partial-derivative in the previous equation is denoted as the Jacobian matrix. If the components of the state vector $\vec{x}$ are (x_1, x_2, …, x_n) and the components of the vector of system-differential-equations **f**( ) are (f_1, f_2, …, f_n), then we can express the Jacobian matrix, **J**, as:

$$\mathbf{J} = \frac{\partial \mathbf{f}}{\partial \vec{x}} = \begin{bmatrix} \frac{\partial f_1}{\partial x_1} & \frac{\partial f_1}{\partial x_2} & \cdots & \frac{\partial f_1}{\partial x_n} \\ \frac{\partial f_2}{\partial x_1} & \frac{\partial f_2}{\partial x_2} & \cdots & \frac{\partial f_2}{\partial x_n} \\ \vdots & \vdots & & \vdots \\ \frac{\partial f_n}{\partial x_1} & \frac{\partial f_n}{\partial x_2} & \cdots & \frac{\partial f_n}{\partial x_n} \end{bmatrix}. \tag{1.34}$$

Next, suppose we define a vector, $\Delta\vec{x}$, which represents a small perturbation of the state variables from the equilibrium point; we can express $\Delta\vec{x} = \vec{x} - \vec{x}^*$. Taking the derivative of the state perturbation, we have:

$$\frac{d\Delta\vec{x}}{dt} = \frac{d\vec{x}}{dt} - \frac{d\vec{x}^*}{dt} = \frac{d\vec{x}}{dt} - \mathbf{0} = \mathbf{f}\left(\vec{x}\right) - \mathbf{0}, \tag{1.35}$$

since $\vec{x}^*$ is defined as an equilibrium point. If the perturbation, $\Delta\vec{x}$, is small, then only the first term of equation (1.33) is significant since the higher terms will involve powers of our displacement from the equilibrium point. Thus, assuming the Jacobian to be non-zero, the first term of equation (1.33) should be useful in telling whether the equilibrium point is stable. Rewriting equation (1.33) in terms of the perturbation vector, we have:

$$\frac{d\Delta\vec{x}}{dt} \approx \left.\frac{\partial \mathbf{f}}{\partial \vec{x}}\right|_{\vec{x}^*} \left(\vec{x} - \vec{x}^*\right) = \mathbf{J}^* \Delta\vec{x}, \tag{1.36}$$

where the matrix $\mathbf{J}^*$ is the Jacobian evaluated at the equilibrium point. The matrix $\mathbf{J}^*$ is constant, so the above equation is simply a linear differential equation which can be analyzed for stability.

According to linear dynamic system theory, a solution to a set of linear differential equations can be expressed as a superposition of terms of the form $e^{\lambda_k t}$, where $\{\lambda_k\}$ is the set of eigenvalues of the Jacobian in the system above. The eigenvalues of the Jacobian are, in general, complex numbers. We can express the eigenvalues as $\lambda_k = \sigma_k + j\omega_k$, where σ_k and ω_k are the real

and imaginary parts of the k^{th} eigenvalue, respectively. After expanding the complex exponential, each term of the homogeneous solution to the differential equation can be expressed as:

$$x_k(t) = c_k e^{\sigma t}\left\{\cos(\omega t) + j\sin(\omega t)\right\}. \tag{1.37}$$

From here we can see that only the real-part of the eigenvalues of the Jacobian is of importance. If we have $\sigma_k < 0$ for all k, then the homogeneous solution will decay with time, which is the definition of a BIBO-stable (bounded-input-bounded-output-stable) system.

1.4.1. COORDINATE SYSTEM & REFERENCE-FRAME MODELING

Now that we have established a method for determining the local stability of a system, we must begin describing the differential equations of that system. We will start with the coordinate-system definitions. In an effort to simplify the modeling equations and notation, we will introduce an additional rotating reference-frame which we will call the *xy*-frame. This will be the reference frame which is phase-aligned to the true grid voltage, v_G (rather than the sensing voltage, v_S). Furthermore, the *xy*-frame will rotate at the grid frequency, ω_G, which may be different than the estimated frequency, $\hat{\omega}$, during a transient event. This is shown in Figure 1.10.

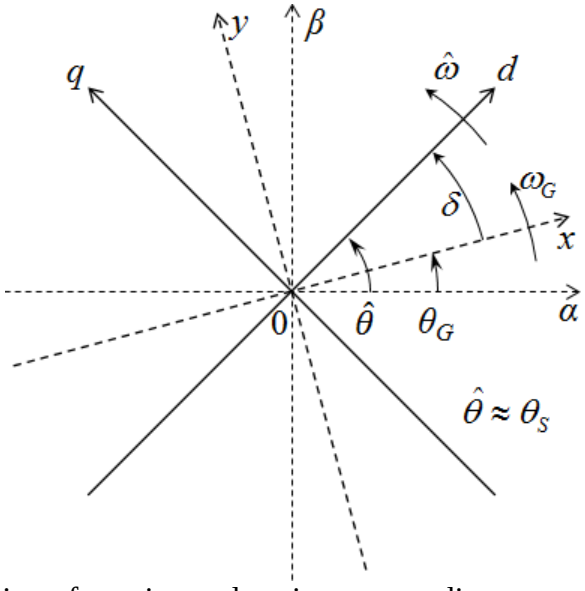


Figure 1.10: Illustration of rotating and stationary coordinate system reference-frames.

The phase angle of both rotating frames (the *xy*-frame and the *dq*-frame) are measured relative to the α-axis of the stationary-frame ($\alpha\beta$-frame). Note that the phase angle of the *dq*-frame, $\hat{\theta}$, may be different than that of the sensing voltage angle, θ_S; however the two angles will be the same at steady-state (i.e., $\varepsilon_\theta = \theta_S - \hat{\theta} \approx 0$). The angle δ represents the phase difference between the angle of the sensing-point voltage, v_S, and true grid voltage, v_G, i.e.:

$$\delta = \hat{\theta} - \theta_G. \tag{1.38}$$

The value of δ depends on the voltage drop across the grid impedance, Z_G; thus, the value of δ should be zero when there is no load current. With these definitions, we can define the grid voltage in its own synchronized reference frame as:

$$\begin{bmatrix} v_{Gx} \\ v_{Gy} \end{bmatrix} = \begin{bmatrix} \cos(\theta_G) & \sin(\theta_G) \\ -\sin(\theta_G) & \cos(\theta_G) \end{bmatrix} \begin{bmatrix} v_{G\alpha} \\ v_{G\beta} \end{bmatrix}. \tag{1.39}$$

We can also transform other quantities, such as the sensing-point voltage, v_S, the inverter voltage, v_I, and the current, i, into the *xy*-frame using the same transformation above. To convert from the synchronous grid-voltage frame (*xy*) to the synchronous sensing-voltage frame (*dq*), we can again use the Park transformation:

$$\begin{bmatrix} v_{Gd} \\ v_{Gq} \end{bmatrix} = \begin{bmatrix} \cos(\delta) & \sin(\delta) \\ -\sin(\delta) & \cos(\delta) \end{bmatrix} \begin{bmatrix} v_{Gx} \\ v_{Gy} \end{bmatrix}. \tag{1.40}$$

This may be easier to understand by examining the space-vector notation of these quantities, together with the Park transforms. We can express both transform-equations for v_G and v_S as:

$$\vec{v}_{G,dq} = \vec{v}_{G,xy} e^{-j\delta} = \vec{v}_{G,\alpha\beta} e^{-j(\theta_G+\delta)} = \vec{v}_{G,\alpha\beta} e^{-j\hat{\theta}}, \tag{1.41}$$

$$\vec{v}_{S,dq} = \vec{v}_{S,xy} e^{-j\delta} = \vec{v}_{S,\alpha\beta} e^{-j(\theta_G+\delta)} = \vec{v}_{S,\alpha\beta} e^{-j\hat{\theta}}. \tag{1.42}$$

We can also apply a similar logic to describe the AC current in any of the reference frames:

$$\vec{i}_{dq} = \vec{i}_{xy} e^{-j\delta} = \vec{i}_{\alpha\beta} e^{-j(\theta_G+\delta)} = \vec{i}_{\alpha\beta} e^{-j\hat{\theta}}. \tag{1.43}$$

1.4.2. CIRCUIT-DYNAMICS MODELING

With the coordinate systems defined, we can begin to describe the circuit parameters and their dynamics in the form of 1st-order differential equations. Starting from our KVL circuit analysis, which gave us equation (1.21), we can further describe the sensing-point voltage in the grid-oriented *xy*-frame as:

$$\vec{v}_{S,xy} = \left(\vec{v}_{G,\alpha\beta} + R_G \vec{i}_{\alpha\beta} + L_G \frac{d\vec{i}_{\alpha\beta}}{dt} \right) e^{-j\theta_G}. \tag{1.44}$$

Distributing the Park transformation to the previous equation gives us the following:

$$\vec{v}_{S,xy} = \vec{v}_{G,xy} + R_G \vec{i}_{xy} + L_G \frac{d\vec{i}_{xy}}{dt} + j\omega_G L_G \vec{i}_{xy}, \tag{1.45}$$

which we can split into real and imaginary parts. Solving for the derivative terms in both cases, we obtain the following cross-linked differential equations:

$$\begin{cases} \dfrac{di_x}{dt} = \dfrac{1}{L_G}\left[-R_G i_x + \omega_G L_G i_y + v_{Sx} - v_{Gx} \right], \\ \dfrac{di_y}{dt} = \dfrac{1}{L_G}\left[-R_G i_y - \omega_G L_G i_x + v_{Sy} - v_{Gy} \right]. \end{cases} \tag{1.46}$$

The benefit of describing the currents in the grid-oriented-frame is that the dynamics here do not depend on the PLL or any of its internal parameters.

1.4.3. PHASE-LOCKED LOOP MODELING

Next, we can describe the phase-locked loop's dynamics in terms of 1st-order differential equations. The first equation is simple to obtain; differentiating equation (1.38) gives us:

$$\frac{d\delta}{dt} = \frac{d\hat{\theta}}{dt} - \frac{d\theta_G}{dt} = \hat{\omega} - \omega_G. \tag{1.47}$$

Next, we must model the estimated frequency, $\hat{\omega}$, which is the output of the PI-controller within the PLL. To do this, we define an arbitrary quantity, γ, which represents the integral of the angle-

error. This will be used to represent the internal value of the integrator used in the PI-controller. Taking the derivative of γ, our second differential equation for the PLL system will be:

$$\frac{d\gamma}{dt} = \varepsilon_\theta = \frac{v_{Sq}}{\sqrt{{v_{Sd}}^2 + {v_{Sq}}^2}}. \tag{1.48}$$

From these definitions, we can define the estimated PLL frequency as an algebraic equation:

$$\hat{\omega} = \left(k_{p,\theta}\varepsilon_\theta + k_{i,\theta}\int \varepsilon_\theta dt\right) = \left(k_{p,\theta}\varepsilon_\theta + k_{i,\theta}\gamma\right). \tag{1.49}$$

1.4.4. CURRENT-CONTROLLER MODELING

Moving on, we will describe the PI-controllers of the current regulation loops. For these, we will take a similar approach to what was done with the PLL. We can define two quantities which are the integral of the error signal of the currents, ε_d and ε_q. These will represent the values of the integrators used in the PI-controllers for the current-regulation. Since the integration of current yields a charge, we will denote these integration variables as Q_d and Q_q, defined below:

$$\begin{cases} \dfrac{dQ_d}{dt} = \varepsilon_d = {i_d}^* - i_d, \\ \dfrac{dQ_q}{dt} = \varepsilon_q = {i_q}^* - i_q. \end{cases} \tag{1.50}$$

From these definitions, we can express the output of the PI-controllers, which, when combined with the axis-decoupling terms, form the inverter output voltages.

$$\begin{cases} v_{Id} = \varepsilon_d k_{p,c} + Q_d k_{i,c} - \hat{\omega} L_I i_q + v_{Sd}, \\ v_{Iq} = \varepsilon_q k_{p,c} + Q_q k_{i,c} + \hat{\omega} L_I i_d + v_{Sq}. \end{cases} \tag{1.51}$$

1.4.5. OVERALL CONTROL SYSTEM MODEL

To "close the loop" on the mathematical model presented in this section, we must relate the information of the inverter voltage back to the circuit to influence the current. However, the differential equations of the current are expressed in terms of only the sensing-point voltage, v_S, and the grid voltage, v_G. This was intentionally done so that the set of equations (1.46) will

remain true in later chapters. The grid impedance theoretically will never be zero and thus these equations should always hold.

To relate the inverter voltage back to the currents, we start with the KVL expression between the inverter voltage and sensing-point voltage, expressed in the *xy*-frame:

$$\vec{v}_{S,xy} = \vec{v}_{I,xy} - \left(L_I \frac{d\vec{i}_{xy}}{dt} + j\omega_G L_I \vec{i}_{xy} + R_I \vec{i}_{xy} \right). \tag{1.52}$$

Next, we can split this equation above into real and imaginary parts and plug in the set of equations (1.46) to cancel out the derivative terms. After a bit of manipulation, we arrive with the following expressions for the sensing-point voltages:

$$\begin{cases} v_{Sx} = \dfrac{1}{L_I + L_G}\left[L_G v_{Ix} + \left(L_I R_G - L_G R_I \right) i_x + L_I v_{Gx} \right], \\ v_{Sy} = \dfrac{1}{L_I + L_G}\left[L_G v_{Iy} + \left(L_I R_G - L_G R_I \right) i_y + L_I v_{Gy} \right]. \end{cases} \tag{1.53}$$

This set of equations can be interpreted as an alternate form of the voltage-divider equation (1.20), written in the *xy*-frame of reference.

The complete set of differential equations which describe the inverter system are as follows:

$$\begin{cases} (1) \quad \dfrac{di_x}{dt} = \dfrac{1}{L_G}\left[-R_G i_x + \omega_G L_G i_y + v_{Sx} - v_{Gx} \right], & \text{Circuit Dynamics} \\ (2) \quad \dfrac{di_y}{dt} = \dfrac{1}{L_G}\left[-R_G i_y - \omega_G L_G i_x + v_{Sy} - v_{Gy} \right], & \text{Circuit Dynamics} \\ (3) \quad \dfrac{d\delta}{dt} = k_{p,\theta}\varepsilon_\theta + k_{i,\theta}\gamma - \omega_G, & \text{PLL Angle Difference} \\ (4) \quad \dfrac{d\gamma}{dt} = \varepsilon_\theta = \dfrac{v_{Sq}}{\sqrt{{v_{Sd}}^2 + {v_{Sq}}^2}}, & \text{PLL-PI Integrator} \\ (5) \quad \dfrac{dQ_d}{dt} = \varepsilon_d = i_d^{\,*} - i_d, & dq\text{-PI Integrator} \\ (6) \quad \dfrac{dQ_q}{dt} = \varepsilon_q = i_q^{\,*} - i_q. & dq\text{-PI Integrator} \end{cases} \tag{1.54}$$

The complete set of auxiliary algebraic equations to accompany the previous set of differential equations are also listed below. The first set here describes the coordinate transformations between the grid and sensing-point frames:

$$\begin{cases} (1) & i_d = +i_x \cos(\delta) + i_y \sin(\delta), \\ (2) & i_q = -i_x \sin(\delta) + i_y \cos(\delta), \\ (3) & v_{Sd} = +v_{Sx} \cos(\delta) + v_{Sy} \sin(\delta), \\ (4) & v_{Sq} = -v_{Sx} \sin(\delta) + v_{Sy} \cos(\delta), \\ (5) & v_{Ix} = +v_{Id} \cos(\delta) - v_{Iq} \sin(\delta), \\ (6) & v_{Iy} = +v_{Id} \sin(\delta) + v_{Iq} \cos(\delta). \end{cases} \tag{1.55}$$

The following set of equations describes the relationships between the sensing-point, inverter, and grid voltages:

$$\begin{cases} (7) & v_{Id} = \varepsilon_d k_{p,c} + Q_d k_{i,c} - \hat{\omega} L_I i_q + v_{Sd}, \\ (8) & v_{Iq} = \varepsilon_q k_{p,c} + Q_d k_{i,c} + \hat{\omega} L_I i_d + v_{Sq}, \\ (9) & v_{Sx} = \dfrac{1}{L_I + L_G}\left[L_G v_{Ix} + (L_I R_G - L_G R_I) i_x + L_I v_{Gx} \right], \\ (10) & v_{Sy} = \dfrac{1}{L_I + L_G}\left[L_G v_{Iy} + (L_I R_G - L_G R_I) i_y + L_I v_{Gy} \right], \\ (11) & \hat{\omega} = k_{p,\theta} \varepsilon_\theta + k_{i,\theta} \gamma. \end{cases} \tag{1.56}$$

To summarize, we have found here that the dynamics of the system are 6$^{\text{th}}$-order. If additional details or features are included in the modeling, this order will increase. If the decoupling voltages can accurately cancel out the cross-coupling terms, then the system could potentially reduce to a 4$^{\text{th}}$-order system at best.

To verify the small-signal stability of the system at any equilibrium point which satisfies the non-linear model, a small-signal state-space model can be obtained by linearizing (1.54). This will result in the linearized system written in the form:

$$\frac{d\Delta\vec{x}}{dt} \approx \mathbf{A}\Delta\vec{x} + \mathbf{B}\Delta\vec{u}, \tag{1.57}$$

where the matrix **A** is the Jacobian, evaluated at the equilibrium point, $\mathbf{J}^*$, and the matrix **B** consists of the partial derivatives of the equations which relate the state-variables to the input values, evaluated at the equilibrium point:

$$\mathbf{B} = \left.\frac{\partial \mathbf{f}}{\partial \vec{u}}\right|_{\vec{x}^*} = \begin{bmatrix} \frac{\partial f_1}{\partial u_1} & \frac{\partial f_1}{\partial u_2} & \cdots & \frac{\partial f_1}{\partial u_n} \\ \frac{\partial f_2}{\partial u_1} & \frac{\partial f_2}{\partial u_2} & \cdots & \frac{\partial f_2}{\partial u_n} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{\partial f_n}{\partial u_1} & \frac{\partial f_n}{\partial u_2} & \cdots & \frac{\partial f_n}{\partial u_n} \end{bmatrix}. \tag{1.58}$$

The state-variable vector, $\Delta\vec{x}$, and input-variable vector, $\Delta\vec{u}$, can be expressed as:

$$\begin{cases} \Delta\vec{x} = \begin{bmatrix} i_x & i_y & \delta & \gamma & Q_d & Q_q \end{bmatrix}^T, \\ \Delta\vec{u} = \begin{bmatrix} i_d^* & i_q^* & v_{Gx} & v_{Gy} & \omega_G \end{bmatrix}^T. \end{cases} \tag{1.59}$$

1.5. ISSUES WITH CONVENTIONAL DC/AC SYSTEMS

To review, interface inductors are a commonly used coupling component for grid-connected DC/AC voltage-sourced power-electronics converters. Placed between the inverter output terminals and the grid-voltage connection/sensing point, they provide two main benefits to the converter system (mentioned in the previous sections and repeated again here).

- Interface inductors, together with the grid impedance, help low-pass filter the grid line-currents, improving the THD of the system and reducing potential EM-interference.
- Interface inductors help to provide a clean sinusoidal voltage-sensing point for converter synchronization by dropping the high-frequency PWM voltage harmonics of the inverter.

Due to the second benefit above, we are able to use external voltage sensors to detect the phase angle, frequency, and magnitude of the grid voltage. It is worth noting that increasing the value of the interfacing inductance, L_I, helps to further improve both of these benefits above.

Despite these benefits, the use of large interface-inductors is undesirable, since this leads to additional cost, volume, weight, and power losses (due to added resistance) within the system. In recent years, much effort has been applied to reduce the interface inductor's size and/or weight without compromising filtering performance, such as the use of different magnetic materials or using planar windings in the inductor's construction [9-10]. The transformer leakage and line leakage impedances also help in reducing current ripples, but their size is usually not a design parameter; the leakage is usually minimized during the transformer design. Note that if the transformer and grid impedances are suitably sized large enough (potentially in a weak-grid or motor/generator scenario), then the interfacing inductance could be heavily reduced. The Laplace expression for the current and its low-pass cut-off frequency, ω_C, effectively reduce to:

$$R_I \approx L_I \approx 0 \quad \Rightarrow \quad \vec{I} = \frac{\vec{V}_I - \vec{V}_G}{s\left(L_I + L_G\right) + \left(R_I + R_G\right)} \approx \frac{\left(\vec{V}_I - \vec{V}_G\right)/L_G}{\left(s + R_G/L_G\right)}; \tag{1.60}$$

$$R_I \approx L_I \approx 0 \quad \Rightarrow \quad \omega_C = \frac{R_I + R_G}{L_I + L_G} \approx \frac{R_G}{L_G}. \tag{1.61}$$

The size of the transformer leakage inductance depends on the transformer size and construction, but typical values range from 5% to 10% per-unit (pu) [11-12]. Also, the inductance of the grid varies based on loading conditions and other factors, but is also generally around a few percent pu [13-14]. It is important to note as well that the grid impedance can be either inductive for over-head cables or capacitive for underground cables [15]. A capacitive cable would not be able to provide harmonic voltage filtering as is desired. Therefore, we will limit our discussions to applications considering only over-head (inductive) cables.

Alternatively, instead of increasing inductance to obtain a lower cut-off frequency, ω_C, of the current, the switching harmonics of the inverter can be shifted up in frequency. The use of higher switching frequencies, f_{sw}, will effectively reduce the magnitude of the ripple currents (since they

will be easier to filter at higher frequencies), enabling the use of smaller inductance values. However, the disadvantage to increased switching frequencies is that the converter will exhibit additional switching power losses. In addition, the converter will exhibit stronger *di*/*dt* and *dv*/*dt* radiated emissions, potentially causing more EMI issues. Another power-electronics approach to enable smaller inductances is to reduce the amplitude of the generated voltage harmonics, rather than increasing the switching frequency. This can be achieved using multi-level converter structures, such as the neutral-point-clamped (NPC) multi-level inverter or the cascaded multi-level inverter (CMI), the latter of which is shown in Figure 1.11 below. However, this drastically complicates the system design.

In addition to the disadvantages associated with large interfacing inductances, another undesirable feature of the traditional control system is the use of dedicated voltage sensors for grid synchronization. The grid-voltage measurement requires the use of isolated (usually Hall-Effect-based) voltage sensors. The use of these discrete sensors increases the overall system cost and physical complexity (while arguably simplifies the software complexity). In very high-voltage systems, the voltage sensors may also be extremely bulky and costly. Work has also been previously done to mitigate these issues; voltage-sensor-less control algorithms for grid-

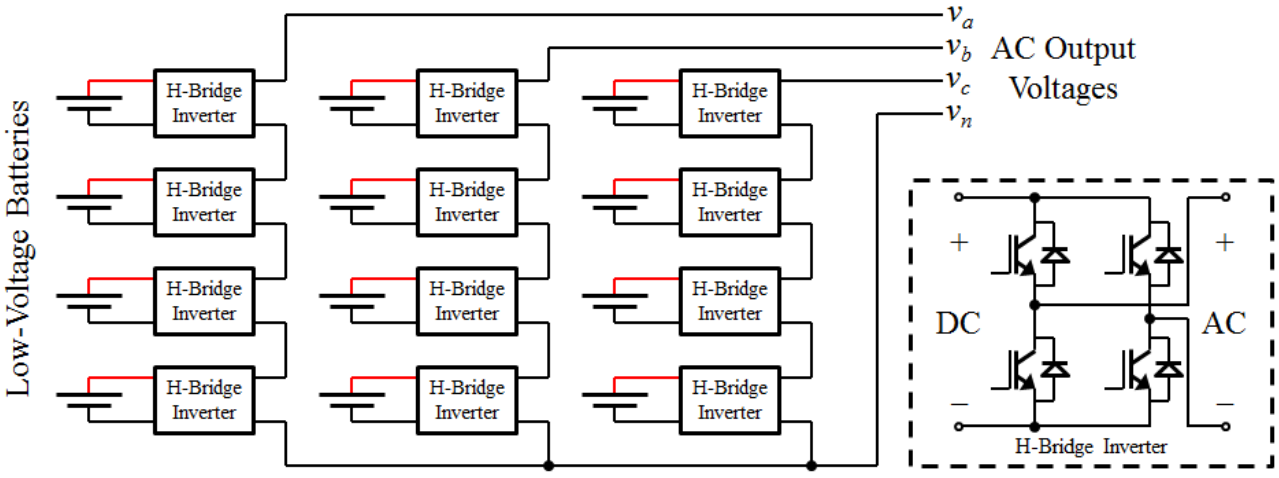


Figure 1.11: Three-phase nine-level cascaded multi-level inverter topology.

connected systems have been developed, such as the Direct Power Control algorithm proposed in [16-17] (which is similar to the Direct Torque Control algorithm of induction machines). However, these algorithms usually depend on a somewhat accurate estimation of the total impedance between the converter and grid. Our interest is to reduce Z_I and, as discussed previously, the grid impedance, Z_G, can vary depending on various operating conditions, especially in a weak-grid situation. These algorithms may have other disadvantages as well, such as a variable switching-frequency or require a very fast sampling rate and control loop.

It is important to note that these two issues (the use of large interface inductors and discrete voltage sensors) are somewhat coupled together. By reducing the size of the inductors to reduce the system size, weight, and cost, the switching harmonics present at the voltage-sensing point will be increased. The placement of analog filters on sensor lines and digital filters within software can be used to remove this noise, but at an increased cost and circuit-board area or an increased computational complexity. Also, both analog and digital filtering approaches can introduce a slight phase delay in the measured voltages.

To help reduce measurement noise, capacitors are sometimes added to the power circuit (forming an LC filter) to make the sensing voltage more "stiff", as in Figure 1.12 (a). If the transformer/grid impedance, Z_G, is small, the circuit model in Figure 1.12 (a) effectively reduces to Figure 1.12 (b). However, this is a step in the wrong direction; the interest here is to reduce R_I and L_I while taking advantage of any R_G and L_G for filtering, as mentioned previously. In this scenario of having small Z_I and large Z_G impedances, adding the interfacing capacitor, C_I, may also add additional switching losses to the converter since more harmonic voltage will be present at the mid-point node, causing increased common-mode currents to flow through the capacitor and the converter. Thus, we will not consider the LC filter scenario in this report.

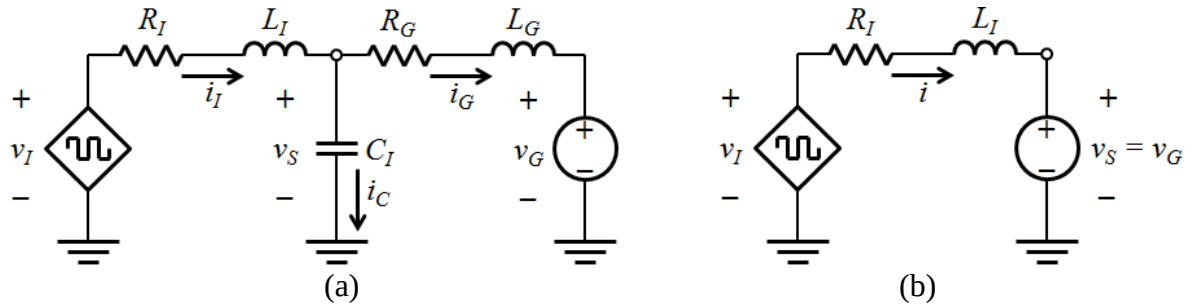


Figure 1.12: Alternate equivalent circuit models: (a) capacitively filtered voltage sensing point; (b) equivalent circuit with a "stiff" grid connection.

To summarize, the reasons for filtering the high-frequency PWM voltages generated by the inverter include the elimination of potential EMI issues, the removal of reactive-power-producing harmonic components in the current, and preventing noise from entering the control system. However there is a trade-off taking place; an increased inductance or switching frequency can help to eliminate the harmonics, but at increased cost, size, weight, or power loss. Furthermore, algorithms using discrete voltage-sensors rely on having a clean voltage-sensing point. Reducing the inductance adds noise to the measurements (since less filtering is achieved). Control methods which do not rely on grid voltage sensors instead rely on knowledge of the interfacing-impedance value.

To avoid these issues, an algorithm which could operate without either of these detrimental components (interface inductors or voltage sensors) would be ideal. However, such an algorithm would need to be able to synchronize to the grid voltage without voltage sensors and regulate the current without knowledge of the grid or interface impedances. Such an algorithm can be realized by the use of virtual impedances – impedances which are connected between the inverter and AC grid within software. These impedances allow for regulation of the current and can provide a virtual voltage-sensing point which is free of PWM switching noise, as we will soon see… If the physical voltage-sensing point is no longer needed, the voltage sensors can be removed from the system. Furthermore, if reasonable filtering of the current can still be achieved

by the grid impedance alone, i.e., if the cut-off frequency, ω_C, in equation (1.61) is still suitably low enough, then the interfacing inductors can also be removed from the system!

1.6. RESEARCH SCOPE AND CONTRIBUTIONS

The main objective of this research is to investigate alternative methods for DC/AC converter synchronization and control, which utilize virtualized impedances in software, based on the sensed AC currents. This is followed by the design and implementation of an experimental control system and DC/AC converter prototype which supports an inductor-less voltage-sensor-less operation. The key contributions of this research can be summarized as follows:

1. A self-synchronizing algorithm for PLL-based DC/AC converters, based on the concept of virtual impedances, is proposed. With this method, the AC voltage sensors can be removed and the physical interfacing inductors can be removed in specific cases.
2. A new method to reduce the computational complexity of the synchronous-frame *dq*-current control algorithm, using the concept of a virtual resistance, is proposed. A simple computational comparison is provided to quantify the complexity reduction.
3. An alternate method to implement an impedance-compensated PLL, based on fully-implemented stationary-frame virtual impedances, is proposed. The proposed method has better transient performance than the traditional partially-implemented rotating-frame virtual-impedance method.

Additional research contributions, which were not experimentally verified but have been theoretically developed within this dissertation, are also listed below. These topics are listed with the potential to be developed further as future works.

4. A more generalized expression for the quasi-static stability of a DC/AC converter's PLL synchronization loop is proposed. The analysis extends the stability criteria to the full range of converter operating points (real and reactive).

5. A combination of the self-synchronizing virtual-impedance algorithm, together with the impedance-compensation of the PLL using virtual impedance, is proposed as a new technique for the position-sensor-less control of permanent-magnet AC machines operating at high-speed ranges. A means for on-line parameter estimation can also be implemented using a low-frequency injection current, together with this technique.

1.7. REPORT ORGANIZATION

The remainder of this dissertation examines the benefits of utilizing virtualized impedances within the synchronization and control loops of a DC/AC converter. The primary focus will be on three-phase DC/AC converter systems but the concepts covered here can be extended to single-phase DC/AC converter systems.

In Chapter 2, the theoretical framework and implementation of virtual impedances is first discussed. Afterwards, a method of replacing the physical interface-impedances with virtualized interfacing-impedances is investigated. Together with this, the feasibility of removing the AC voltage sensors is also discussed. The chapter then introduces the proposed concept of using only a virtual interfacing-resistance for self-synchronization and computational simplification of the common *dq*-current control algorithm. The stability of the control system is analyzed, focusing on proper tuning of the PID controllers. Afterwards, the experimental test setup used to validate the control method is introduced. The chapter concludes with comparisons between simulated and experimental waveforms showing the dynamic performance of the algorithm.

Chapter 3 introduces the concept of utilizing virtual impedances to modify the synchronization voltage (as opposed to the inverter voltage, as discussed in Chapter 2). A newly proposed method to impedance-compensate the PLL, which has better transient response and is not affected by the PLL's own dynamics, is introduced. This method is verified through simulation results. Next, it is shown how impedance-compensation (IC) of the PLL algorithm can enhance the quasi-static stability of the PLL algorithm. Finally, a more rigorous approach to show the IC-PLL stability is also developed, following [19].

Next, Chapter 4 presents the author's concluding remarks, recommendations, and ideas for future work. A nine-level (four-module per phase) cascaded multi-level inverter (CMI) is introduced as a possible candidate to enable a true sensor-less inductor-less connection to the 208 V low-voltage utility grid. Also, the combination of impedance-compensation with a virtualized self-synchronization method is proposed for position-sensor-less high-speed (back-EMF-based) motor control applications. A low-voltage dynamometer test bench located at Kettering University is also introduced as a possible candidate to test the proposed impedance-compensated self-synchronized sensor-less permanent magnet AC motor control algorithm.

In conclusion, some details on the MATLAB Simulink simulation model, used to validate some of the developed algorithm's dynamic performance characteristics, are presented in Appendix A. Information and further details on the 1 kW experimental three-phase inverter prototype are presented in Appendix B. In Appendix C, a list of the author's scholarly works done over the past several years are presented. This list includes several journal publications, conference papers, and one patent.

2. VIRTUAL-INTERFACE-IMPEDANCE SYNCHRONIZATION METHODS

In this chapter, the theoretical framework and implementation of virtual impedances are first discussed. The limitations involved with implementing virtual-inductors are then examined. Afterwards, the concept of replacing the physical interface-impedances with virtualized interfacing-impedances and the removal of the AC voltage sensors for self-synchronization is proposed and investigated. An idea for synchronizing during the start-up process using current sensors is also outlined. The chapter concludes with the proposed concept of using only a virtual interfacing-resistance for PLL self-synchronization, which can be used to reduce the computational complexity of the commonly used *dq*-current control algorithm. The stability and dynamics of these methods are examined and confirmed by simulation and experimental results.

2.1. VIRTUAL-IMPEDANCE IMPLEMENTATION & LIMITATIONS

The concept of virtual impedance has been used in many prior applications [18-20]. However, in most of these applications, the virtual impedance is not fully implemented. The basic concept of virtualizing impedances is to directly apply the complex form of Ohm's Law within the software of the converter to change the behavior of the converter's output.

In the case of a Voltage-Sourced Converter (VSC), the output is a controlled voltage waveform, usually in the form of a PWM-voltage. The control-inputs to a VSC are typically currents, other voltages, or other control signals. To implement a virtual impedance, assuming that current is one of the sampled input signals, the converter's normal output voltage, v_I, can be augmented to become v_I' by subtracting (or adding) the voltage drop across a virtual interface impedance, Z_{VI}, calculated via Ohm's Law:

$$\vec{v}_I' = \vec{v}_I - Z_{VI}\,\vec{i}. \tag{2.1}$$

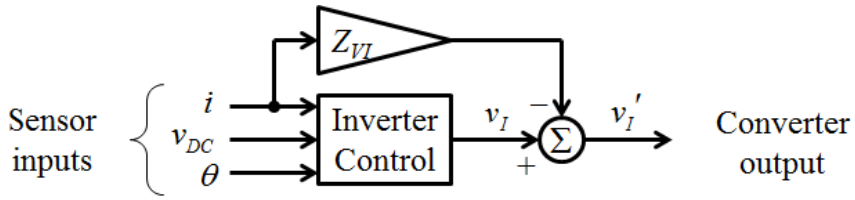


Figure 2.1: Inverter control algorithm augmented with virtual impedance calculation.

This concept is also portrayed in Figure 2.1 above. The remainder of the control system can remain unaltered, with the virtual impedance calculation occurring alongside it. A similar approach could be used within a Current-Sourced Converter (CSC), where the voltage of the converter could be measured and the converter's output current could be augmented by adding the product of a voltage and a virtual-admittance.

The implementation of these impedances is generally straightforward if done in the stationary-frame. All of the sensor inputs to the controller will ultimately originate in the stationary-frame time-domain so we can directly compute the impedance voltage-drops using the Ohm's Law expressions: $v_R = iR$, $v_L = L\ di/dt$; and $v_C = 1/C \int i\ dt$; the exception to this is the inductance. Because the sampled current-sensor inputs will undoubtedly contain sampling noise, the calculated inductor voltage will be extremely erratic if computed directly using a pure derivative. To remedy this, the sampled current can first be passed through a low-pass filter with a unity gain before taking the derivative. Combining these two operations ultimately results in a high-pass filter. We can express the approximated inductor voltage in the Laplace domain as:

$$\vec{v}_L = sL\vec{i} \approx \frac{s\,\omega_{HP}}{s+\omega_{HP}} L\vec{i}. \tag{2.2}$$

The high-pass filter's cut-off frequency, ω_{HP}, is multiplied on the numerator to preserve the low-frequency unity-gain and the value of ω_{HP} is chosen to be substantially higher than the frequency of operation. This makes the response very similar to a pure derivative at low frequencies. This is shown graphically in the Bode magnitude plots of a pure derivative and a unity-gain high-pass

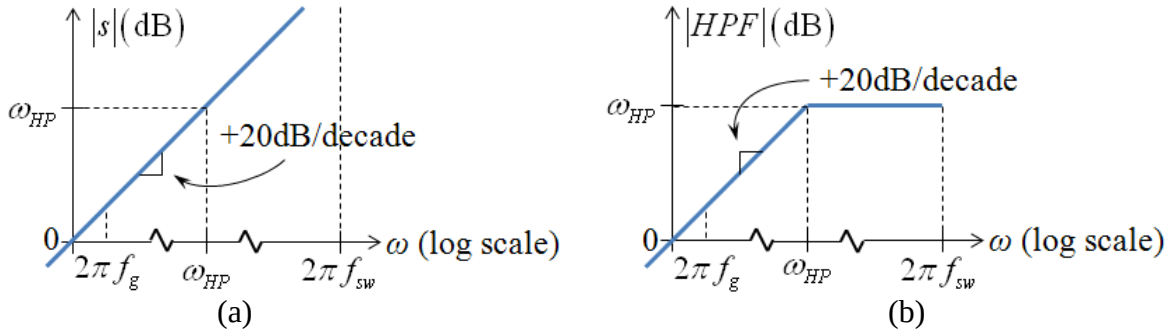


Figure 2.2: Bode magnitude plots of (a) a pure derivative and (b) a unity-gain high-pass filter.

filter, presented in Figure 2.2. In the vicinity of the grid frequency, f_g, the high-pass filter's magnitude in Figure 2.2 (b) looks nearly identical to the pure derivative's magnitude in (a). The phase plots of both transfer functions (not shown) also look the same near the grid frequency. Thus, in a low-frequency sense, the two calculations produce identical results.

The high-pass filter's cut-off frequency, ω_{HP}, must be chosen so that it is well above the desired grid operating frequency (f_g = 60 Hz typically); at least one decade above (f_{HP} = 600 Hz) is recommended so that the phase shift of the filter will not have much impact. However, care must also be taken not to set the cut-off frequency too high, as this will allow more high-frequency noise to be amplified by the derivative-like features of the filter. In practice, one should select a cut-off frequency which is below the switching frequency of the converter. This will ensure that the switching harmonics which are present in the sensed line-currents are not amplified excessively (as Figure 2.2 (a) depicts). For example, if the grid frequency is f_g = 60 Hz and the converter switching frequency is f_{sw} = 10 kHz, an acceptable high-pass-filter cut-off frequency might be around f_{HP} = 1 kHz.

Another form of virtual impedance, which is more commonly used, is the implementation within a rotating reference-frame. Using the Clarke and Park transformations to move to a reference-frame which is synchronous to the grid, a virtual-impedance can be partially implemented by examining the mathematical result of moving a derivative into a rotating-frame.

The example provided in equation (2.3) shows the results of moving an inductive voltage from the stationary-frame to a rotating-frame:

$$\vec{v}_{L,\alpha\beta} = L\frac{d\,\vec{i}_{\alpha\beta}}{dt} \quad \Rightarrow \quad \vec{v}_{L,dq} = \vec{v}_{L,\alpha\beta}e^{-j\omega t} = L\frac{d\,\vec{i}_{dq}}{dt} + j\omega L\vec{i}_{dq}. \tag{2.3}$$

Usually, only the second term, $j\omega L\vec{i}_{dq}$, at the end of equation (2.3) is directly implemented; the derivative term is excluded for the same reasons mentioned previously with derivatives in the stationary-frame. Thus, it should be noted that this second term in equation (2.3) only represents the steady-state voltage-drop across the inductance. This is the reasoning behind the claim that most virtual impedances are only "partially" implemented.

This implementation of partial-virtual-impedance is commonly used within *dq*-current regulators for motor controls or active rectifiers; it is used for decoupling the *dq*-axis currents by subtracting this steady-state voltage from the respective axes. It has also been used for impedance-compensation of PLL algorithms [19]; this will be explored in more detail within Chapter 3. The remaining derivative-voltage term is left usually to a PI-controller to cancel its influence on the current. This implementation is actually the same as what's shown previously in Figure 1.8; it has been restated below as Figure 2.3 with the partial-virtual-impedance terms highlighted in red. Partial virtual-capacitance can be implemented in a similar fashion.

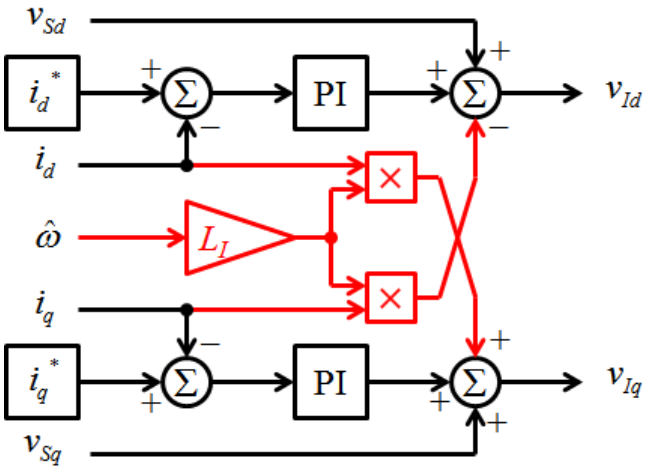


Figure 2.3: Classical *dq*-frame current controller with inductive-decoupling terms highlighted.

Finally, it should be noted that, the main limitation of using partial or fully-implemented virtual impedances, specifically virtual-inductors in a PWM-based VSC and virtual-capacitors in a PWM-based CSC, is that they cannot be used to respond to signals at frequencies equal to or higher than the converter's switching frequency. The reasoning for this is similar to the "chicken and the egg" analogy; in the case of virtual-inductors within a VSC, the voltage drop caused by the virtual-inductor is being reconstructed through the approximate PWM voltage output of the power-electronics converter. If the virtual inductor "tries to respond" to an input signal whose oscillation-period is shorter than one PWM cycle of the converter, the converter will be unable to generate this response accurately. In other words, the use of PWM only allows for different average-output-voltage values to be computed and generated once per switching period. Ideally, the highest-frequency of a signal that could be applied to the output of a PWM-based converter without being aliased should be ½ the switching frequency, based on a similar reasoning to the Nyquist-Shannon Sampling Theorem. Therefore, to summarize, virtual-inductors cannot be used to filter out PWM switching harmonics from a system (this would require some method of producing opposing PWM voltage-drops – this is what real physical inductors do).

2.2. REMOVAL OF THE INTERFACING INDUCTORS & VOLTAGE SENSORS

As discussed in Chapter 1, the interfacing-inductor provides two main benefits: a separate voltage sensing point and filtering of the PWM voltage harmonics. Using the concept of virtual impedance, the first item (the voltage sensing point) can be fabricated within software. If the PWM voltage harmonics can be safely ignored (if other inductances exists in the system), then the real impedance can be safely replaced by a virtual one! From a control-standpoint, there is little difference between a DC/AC inverter system which uses a real, physical inductor and one which uses a fully virtualized one. Even though the high-frequency content may be slightly

different (since a virtualized inductor using the high-pass-filter approximation looks like a resistor above ω_{HP}), the high-frequency harmonics do not usually play a role within the control system (it is assumed that these harmonics will be filtered out anyways). With this in mind, if we can argue that if the high-pass filter's bandwidth is high enough, then the use of a virtualized inductor will not impact the remainder of the control system; this is a reasonable assumption in most cases.

For example, if a high-pass-filter is used to compute an interface-inductor voltage and the bandwidth (the cut-off frequency) of the filter is selected to be f_{HP} = 1 kHz, the dynamic response of this inductor can be roughly 10x greater than the bandwidth of the *dq*-current PI-controllers which are tuned to a settling time of 5 ms (a reasonable speed for the currents). Using the analysis from Chapter 1, we can rewrite equation (1.3) (which is the expression for the closed-loop poles of the *dq*-current PI-control loops) to be in terms of the *dq*-current settling time, $T_{st,c}$:

$$s = -\frac{4}{T_{st,c}} \pm \sqrt{\left(\frac{4}{T_{st,c}}\right)^2 - \frac{k_{i,c}}{L_I}}. \tag{2.4}$$

If a settling time of 5 ms is selected for the currents and the controllers are critically damped, the bandwidth of the *dq*-current loops will be roughly $4/(2\times\pi\times 0.005\text{ s}) \approx 127$ Hz, nearly a decade below f_{HP}. Usually, the *dq*-currents are controlled to have the fastest dynamics in the control system. Thus, in this example, it should be acceptable to ignore the effects of the high-pass filter on the remainder of the inverter's control system.

The standard control system block diagram is shown in Figure 2.4 on the next page. Again, the only difference between the inverter output voltage, v_I, and the sensing voltage, v_S, is the voltage drop across the interfacing impedance, $Z_I = R_I + j\omega L_I$, which we will denote as v_{ZI}:

$$\vec{v}_{Z_I,\alpha\beta} = \vec{v}_{I,\alpha\beta} - \vec{v}_{S,\alpha\beta} = R_I\vec{i}_{\alpha\beta} + L_I\frac{d\vec{i}_{\alpha\beta}}{dt}. \tag{2.5}$$

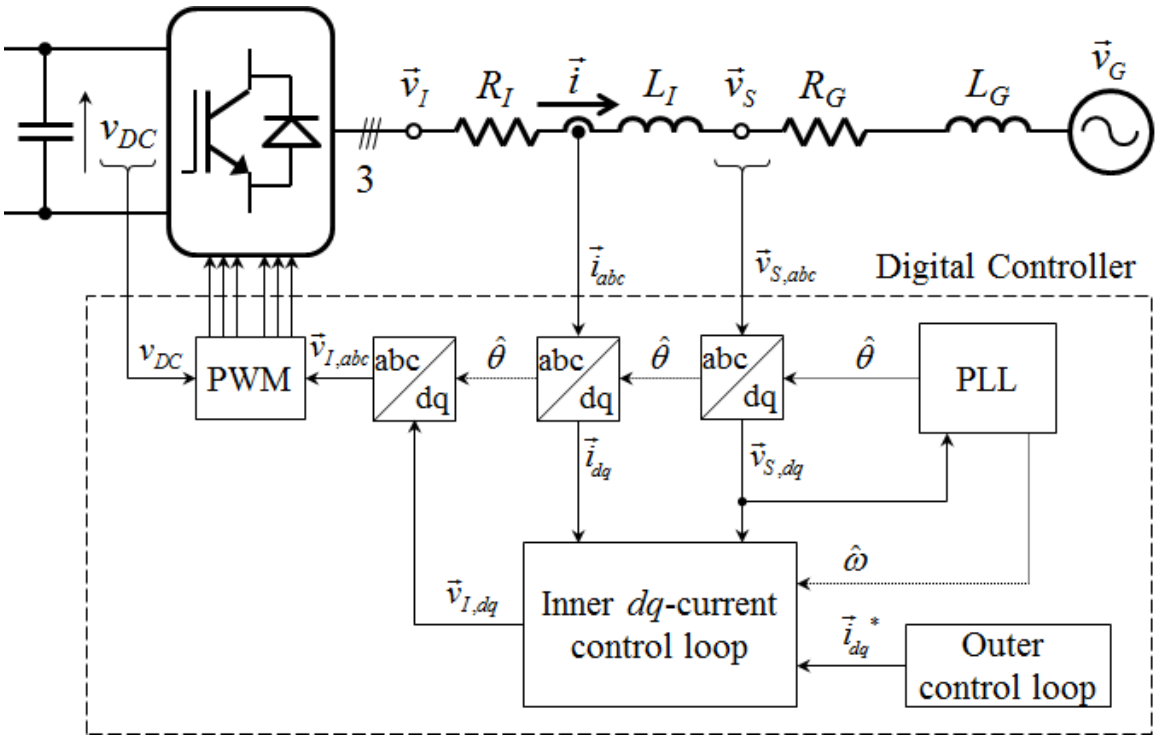


Figure 2.4: Classical inverter control block diagram with interfacing impedance.

Now, if we temporarily ignore the PWM harmonic content of the inverter voltage, v_I, and assume that the inverter could generate a pure sine-wave output voltage, then we could argue that there should be no difference between the control system above in Figure 2.4, and the modified control system shown in Figure 2.5, on the next page. In Figure 2.5, the physical interfacing impedance has been removed and the voltage-drop of this impedance has been computed within the digital controller. This virtual-impedance voltage-drop is then subtracted from the "normal" inverter output voltage, v_I, (computed by the *dq*-current controllers,) to generate an "augmented" inverter voltage, v_I', which accounts for the drop. With the inverter output voltage augmented and the physical impedance replaced by a short-circuit, the sensing voltage, v_S, remains the same as it was in Figure 2.4 and is now also equal to the inverter output: $v_S = v_I'$.

From a low-frequency standpoint, the control scheme of Figure 2.5 should work identical to that of Figure 2.4. The problem here, however, is when we relax our previous assumption and

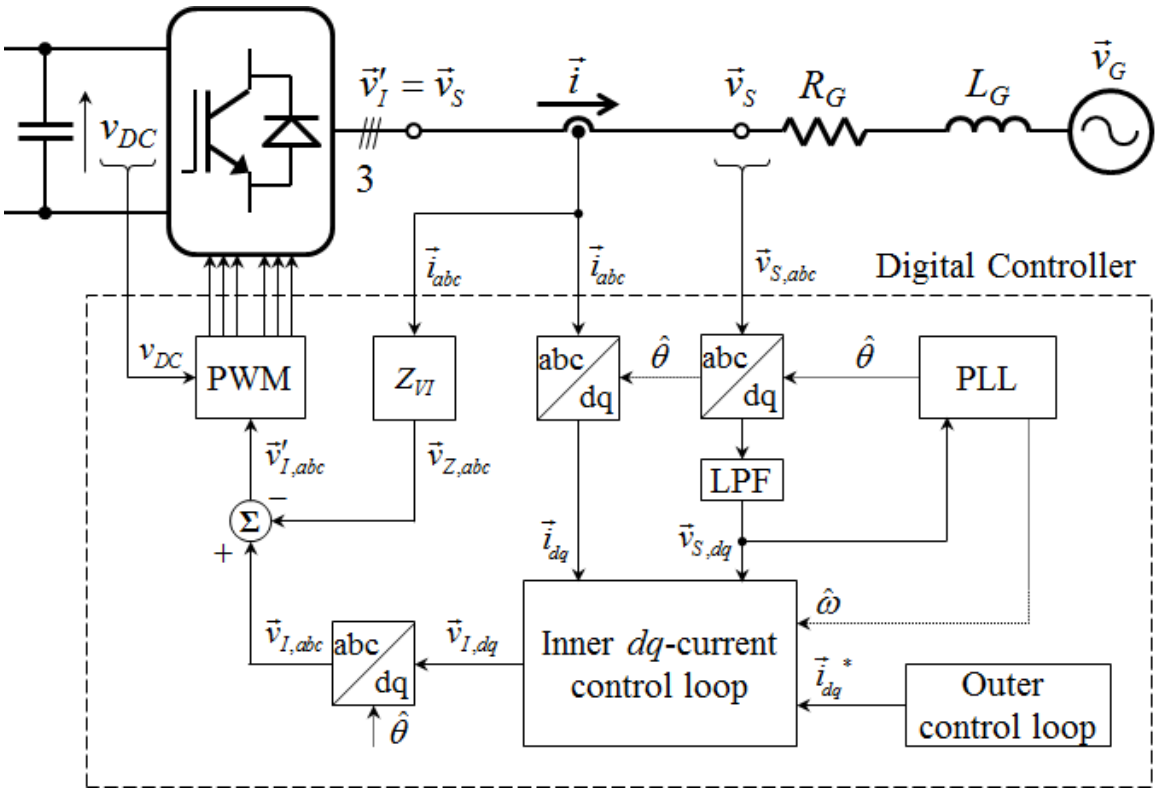


Figure 2.5: Inductor-less inverter control block diagram utilizing virtual impedance.

consider the inverter PWM harmonics. These are now being directly fed by the voltage sensors into the Clarke/Park transformations, which then in turn feeds to the PLL and the *dq*-current controllers. The PLL may be able to withstand/filter the excessive PWM harmonic content, but the *dq*-current controller cannot. Any feed-forward-like paths from the PLL to the *dq*-control will be plagued with excessive noise, requiring the use of additional low-pass filters (as in Figure 2.5) or the use of other methods to eliminate the PWM noise from the control system. However, a simpler and more convenient solution exists to solve this harmonic noise problem.

It is only the low-frequency component of the sensing voltage which is needed by the PLL and *dq*-controller (the PWM harmonics are the problem). Again, note that the inverter output voltage is now equal to the sensing voltage: $v_S = v_I'$ (since the two nodes are shorted). The harmonics are only added to the output by the PWM algorithm. The input to the PWM block will only be the low-frequency component of the inverter voltage. Thus, the desired voltage signals

(without PWM) already exist within the control algorithm! This means that we can directly bypass the PWM-to-inverter-to-sensor path and directly feed this augmented inverter voltage, v_I', within software back into the control system; this proposed control technique is shown below in Figure 2.6. Again, this works because the v_I' voltage signals in software are the same as only the low-frequency component of the sensing voltage (no PWM has been added yet).

With this control technique, the AC voltage sensors can be completely removed from the system! The only new complication that arises from this proposed control algorithm is the method of synchronization during the start-up process. Of course, there also still exists the issue of having adequate filtering inductance to smoothen the shape of the AC current – note the consistent inclusion of the grid inductor L_G. Since there is no longer any voltage-sensing on the AC side, when the converter is not operating, there is no direct way to measure the grid voltage to synchronize to it. Starting the converter without proper synchronization can cause start-up

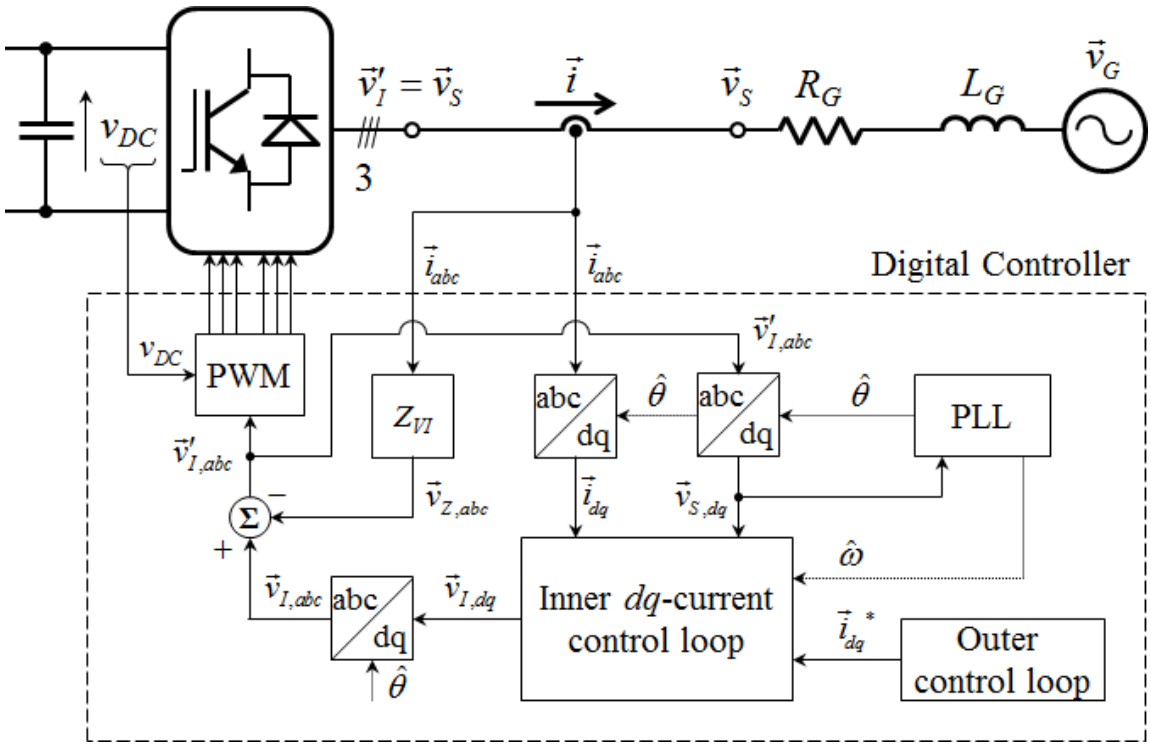


Figure 2.6: Proposed inductor-less AC voltage-sensor-less inverter control diagram using virtual interface-impedance.

transient issues, which could trigger fault protections built into the converter, grid, or both.

To solve the start-up synchronization problem, the on-board current sensors could be used to detect the frequency, phase angle, and approximate voltage magnitude of the grid. To do this, a small load could be applied to the AC-lines to generate some small AC current through the current-sensors. A separate single-phase PLL algorithm could then be used to lock onto the AC current waveform to infer the magnitude, phase, and frequency of the grid. The load could be inductive, capacitive, or resistive, however a capacitive load is recommended to avoid the bulkiness of a 60 Hz inductive load and the power losses of a resistive load. A configuration such as the one in Figure 2.7 could be used, where the capacitive load could be connected from line-to-line through a small (low-current-rated) bi-directional switch.

During the start-up process, the start-up switch, S_{SU}, could be closed, placing a start-up capacitor, C_{SU}, across two of the lines. This start-up capacitor will draw a capacitive current through two of the current sensors. After synchronization, the S_{SU} switch could be opened and the converter could start generating PWM while simultaneously enabling the self-synchronization algorithm. If the capacitor was connected across phases B and C, the current flowing through the phase-C current sensor will have the exact same phase as the phase-A line-

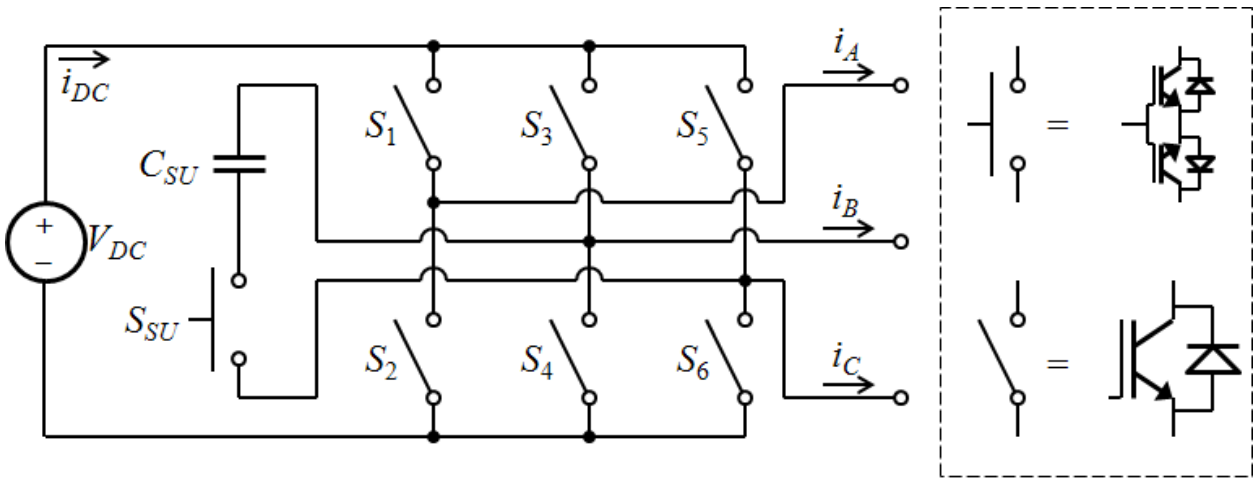


Figure 2.7: Capacitive line-line load configuration for voltage-sensor-less start-up synchronization.

to-neutral voltage (the capacitor current will have a 30° shift because the applied voltage is line-to-line and another 90° shift since the load is capacitive). Thus, a single-phase PLL could be used to lock onto the phase-C current to obtain the grid frequency and phase angle.

To detect the grid voltage magnitude, the start-up capacitor, C_{SU}, could be carefully selected to produce a reasonable current magnitude which could still be sensed. For example, suppose an inverter system was designed with bidirectional current sensors having a maximum sensing range of ±100 A (this would be adequate for a three-phase 208 V_{AC}, 20 kVA system). Assuming the microprocessor used in the converter had an Analog-to-Digital Converter (ADC) with 12-bit resolution, the per-bit resolution of the ADC channels connected to the current sensors would be equal to: 200 A / 2^{12} bits = 48.82 mA/bit. To allow a few bits of sensing range, the peak phase current, I_m, (which is equal to the capacitor current) could be limited to ~250 mA. This will cause the ADC values to vary by about ± 5 bits (which may be barely enough for the algorithm to detect and synchronize to). Working the phasor analysis, the required capacitance to produce a current of 250 mA peak on a 60 Hz grid with 208 V_{LL} RMS would be:

$$C = \frac{|I_C|}{\omega |V_C|} = \frac{0.25\ \text{A}}{(2\pi\, 60\,\text{Hz})\left(208\sqrt{2}\ \text{V}\right)} \approx 2.2\ \mu\text{F}. \tag{2.6}$$

Such a capacitor of the polyethylene-film-type can easily be found. For the synchronization method, a single-phase nonlinear-variation of the SRF-PLL algorithm, shown in Figure 2.8, can

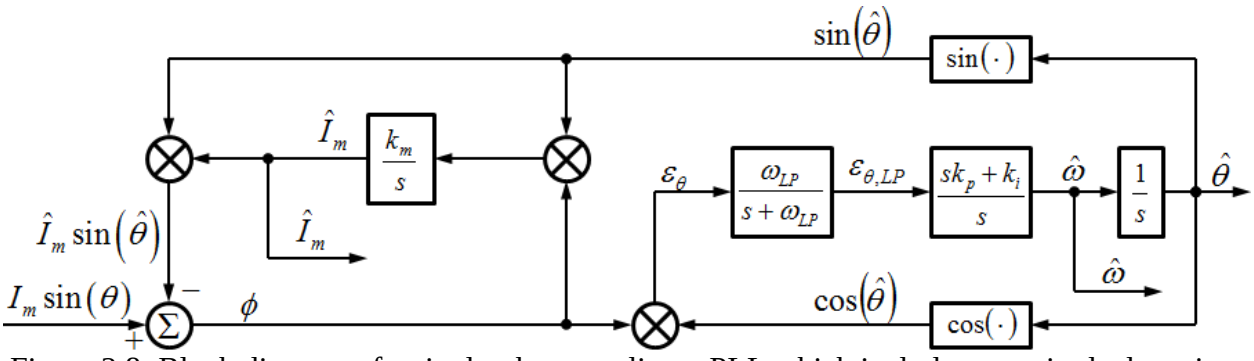


Figure 2.8: Block diagram of a single-phase nonlinear PLL which includes magnitude detection.

be used. Once the magnitude of the current is detected, the voltage magnitude could be computed by reversing equation (2.6) above.

2.3. VIRTUAL-INTERFACE-IMPEDANCE SYSTEM STABILITY ANALYSIS

To maintain similarity with the traditional control method, we will consider the case of using a virtual-impedance composed of a virtual resistance, R_{VI}, and a virtual inductance, L_{VI}. As stated in the previous section, the overall control of such a system should be practically identical to that of the system with real interfacing impedances. However, we must exercise caution when considering the stability of the system, since there is an inherent algebraic loop present in the control system shown above in Figure 2.6. The inverter's output voltage directly feeds the PLL inputs, which in turn are used in the calculation of the inverter's output voltage via the pseudo-feed-forward decoupling paths.

2.3.1. CONTROL-SYSTEM SAMPLING-DELAY MODELING

To deal with the algebraic loop, we must place single-cycle delays within the control algorithm. Luckily, such delays are conveniently already present within the digital implementation of the control algorithm, since most Analog-to-Digital Converter (ADC) hardware and PWM hardware within a Digital Signal Processor (DSP) usually require at least a few clock cycles of delay before reading or writing values. In some cases, these delays are intentionally enlarged to maintain synchronization with the PWM switching of the converter. These delays are illustrated in the block diagram shown in Figure 2.9. Thus, a total of two switching-period delays will accumulate in the control algorithm. In the traditional control

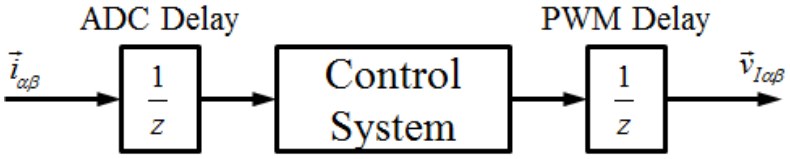


Figure 2.9: Illustration of input & output signal delays in a digital control system.

system, the sensing-point voltage, v_S, would also experience a one-cycle delay (not shown here).

Note that the delays are effectively applied in the stationary-frame of the system (the real sensor and PWM signals will be three-phase sinusoidal signals). Applying delays to sinusoidal signals will cause a slight phase-shift in the measured values. This will show up as a slight rotation of the values in the *dq*-frame (or the *xy*-frame). These delays will also affect the dynamics of the system, potentially causing instability issues.

As was done in Chapter 1, we can model all of the system dynamics as a set of 1st-order non-linear differential equations. From this set of equations, we can linearize the system at an equilibrium point by computing the Jacobian matrix, and from here, we can assess the local stability. However, an approximation must be made for these discrete-time unit-delays to move all terms to continuous-time.

One method of capturing the behavior of discrete delays in a continuous-time system is to use the Taylor-series approximation on the Laplace-transform of the unit-delay. The Laplace transform of the time-shifted unit-impulse function can be expressed using the time-shifting property of Laplace transforms, as shown below:

$$f(t) = \delta(t - t_0) \quad \Rightarrow \quad \mathfrak{L}\{f(t)\} = F(s) = e^{-st_0}; \quad t_0 \geq 0. \tag{2.7}$$

The issue here is that the Laplace transform of the unit delay cannot be easily expressed as a rational polynomial of complex frequency, *s* (due to the exponential). However, by using a first-order Taylor-series approximation, we can express the Laplace transform of the time-delay as simply a low-pass filter. The transfer function approximation is given as equation (2.8). Using the first-order Taylor series approximation of e^x near $x = 0$; we can say $e^x \approx 1 + x$, as shown:

$$F(s) = e^{-st_0} = \frac{1}{e^{+st_0}} \approx \frac{1}{1 + t_0 s}; \quad t_0 \geq 0, \tag{2.8}$$

where the time delay, t_0, in this case will be the inverter's sampling/switching period, T_{sw}.

Applying this concept to our control system, we must include additional state variables to model the time-delay low-pass filters, which were not present in our control system model developed in Chapter 1. To be consistent with our previous modeling in Chapter 1, we will first transfer the unit-delays to a rotating frame. Applying the frequency-shifting property of Laplace transforms, we obtain:

$$F(s) \approx \frac{1}{1+sT_{sw}} \qquad \Rightarrow \qquad F(s+j\omega) = \frac{1}{1+(s+j\omega)T_{sw}}. \tag{2.9}$$

As illustrated in Figure 2.9, we will need to apply this complex low-pass filter to the sampled input currents and the outputted inverter voltages. For the currents, we can introduce a new space-vector variable, $\vec{i}_{xy,LP}$. Applying the delay in the *xy*-frame gives the following:

$$\vec{i}_{xy,LP} = \frac{1}{1+(s+j\omega_G)T_{sw}}\vec{i}_{xy} \qquad \Rightarrow \qquad \frac{d\vec{i}_{xy,LP}}{dt} = \frac{1}{T_{sw}}\left[\vec{i}_{xy} - \vec{i}_{xy,LP} - j\omega_G T_{sw}\vec{i}_{xy,LP}\right]. \tag{2.10}$$

Splitting the above equation into real and imaginary parts gives us

$$\begin{cases} \dfrac{di_{x,LP}}{dt} = \dfrac{1}{T_{sw}}\left[i_x - i_{x,LP} + \omega_G T_{sw} i_{y,LP}\right], \\ \dfrac{di_{y,LP}}{dt} = \dfrac{1}{T_{sw}}\left[i_y - i_{y,LP} - \omega_G T_{sw} i_{x,LP}\right]. \end{cases} \tag{2.11}$$

For the output voltage delay, realize that because of the structure of our virtual-interface impedance system, the only difference between our augmented inverter voltage, v_I', and the virtual sensing-point voltage, v_S, is a single-cycle delay. This is illustrated in Figure 2.10 below.

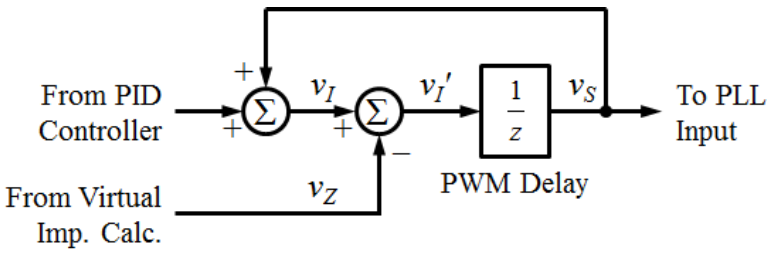


Figure 2.10: Location of PWM delay in a virtual-interface-impedance control system.

This arrangement is required to prevent an algebraic loop from forming when computing v_S within the control system. Therefore, the inverter output voltage and the sensing-point voltage are technically not equal, but should be very similar to each other. Rather than introducing new variables, we will define the PWM delay using existing variables within this particular control scheme. Modeling the delay in the *dq*-frame gives us the following relationship:

$$\vec{v}_{Sdq} = \frac{1}{1+\left(s+j\omega_G\right)T_{sw}}\vec{v}'_{Idq} \qquad \Rightarrow \qquad \frac{d\vec{v}_{Sdq}}{dt} = \frac{1}{T_{sw}}\left[\vec{v}'_{Idq} - \vec{v}_{Sdq} - j\hat{\omega}T_{sw}\vec{v}_{Sdq}\right], \tag{2.12}$$

which after splitting into real and imaginary parts gives us

$$\begin{cases} \dfrac{dv_{Sd}}{dt} = \dfrac{1}{T_{sw}}\left[v'_{Id} - v_{Sd} + \hat{\omega}T_{sw}v_{Sq}\right], \\ \dfrac{dv_{Sq}}{dt} = \dfrac{1}{T_{sw}}\left[v'_{Iq} - v_{Sq} - \hat{\omega}T_{sw}v_{Sd}\right]. \end{cases} \tag{2.13}$$

Accounting for these delays, we must also update our Park-transform definitions of moving between the grid-aligned *xy*-frame and the PLL-aligned *dq*-frame. The set of equations (1.55) will be modified to use the low-pass-filtered values instead of the actual values. This is shown below. Note that equations (5) and (6) of the set have been removed since the theoretical inverter voltage, v_I, is no longer needed in the differential equations describing the currents. Also note that the reference-frame directions of equations (3) and (4) of the set have been reversed.

$$\begin{cases} (1) \quad i_d = +i_{x,LP}\cos\left(\delta\right) + i_{y,LP}\sin\left(\delta\right), \\ (2) \quad i_q = -i_{x,LP}\sin\left(\delta\right) + i_{y,LP}\cos\left(\delta\right), \\ (3) \quad v_{Sx} = +v_{Sd}\cos\left(\delta\right) - v_{Sq}\sin\left(\delta\right), \\ (4) \quad v_{Sy} = +v_{Sd}\sin\left(\delta\right) + v_{Sq}\cos\left(\delta\right). \end{cases} \tag{2.14}$$

2.3.2. VIRTUAL INTERFACE-IMPEDANCE MODELING

In addition to the sampling delays, we must develop equations to express the virtual-interface-impedance voltages in terms of a set of 1st-order differential equations. We start from

the *s*-domain KVL expression of the virtual-impedance voltage, where the inductor voltage is calculated using a high-pass filter:

$$\vec{v}_{Z_{VI},\alpha\beta} = \left[R_{VI} + L_{VI} \frac{s\omega_{HP}}{s+\omega_{HP}} \right] \vec{i}_{\alpha\beta}, \tag{2.15}$$

where ω_{HP} is the high-pass filter cut-off frequency. We can rewrite this equation to express the voltage, $\vec{v}_{Z_{VI}}$, in the *dq*-frame by using the frequency-shifting property of Laplace transforms:

$$\vec{v}_{Z_{VI},dq} = \left[R_{VI} + L_{VI} \frac{\left(s+j\hat{\omega}\right)\omega_{HP}}{\left(s+j\hat{\omega}\right)+\omega_{HP}} \right] \vec{i}_{dq}. \tag{2.16}$$

Note that this also results in a complex low-pass filter, as seen in the previous section:

$$\vec{v}_{Z_{VI},dq} = R_{VI}\vec{i}_{dq} + L_{VI}\left(s+j\hat{\omega}\right)\left[\frac{\omega_{HP}}{s+j\hat{\omega}+\omega_{HP}} \vec{i}_{dq} \right]. \tag{2.17}$$

To accommodate this filter, we can create an additional space-vector state-variable, $\vec{i}_{dq,LP}$, to represent the stationary-frame low-pass filter of the current (to be used in the high-pass filter). Using this low-pass filter variable, we can rewrite the previous equation as:

$$\vec{v}_{Z_{VI},dq} = R_{VI}\vec{i}_{dq} + L_{VI}\left(s+j\hat{\omega}\right)\vec{i}_{dq,LP}. \tag{2.18}$$

The relationship between the low-pass filter's input, $\vec{i}_{dq}$, and output, $\vec{i}_{dq,LP}$, is expressed below:

$$\frac{d\vec{i}_{dq,LP}}{dt} = \omega_{HP}\vec{i}_{dq} - \omega_{HP}\vec{i}_{dq,LP} - j\hat{\omega}\vec{i}_{dq,LP}, \tag{2.19}$$

which, when split into real an imaginary parts, gives us

$$\begin{cases} \dfrac{di_{d,LP}}{dt} = \omega_{HP} i_d - \omega_{HP} i_{d,LP} + \hat{\omega} i_{q,LP}, \\ \dfrac{di_{q,LP}}{dt} = \omega_{HP} i_q - \omega_{HP} i_{q,LP} - \hat{\omega} i_{d,LP}. \end{cases} \tag{2.20}$$

From here, we can substitute the set of equations (2.20) into equation (2.18) to obtain the total expression for the virtual interface-inductor voltage in the *dq*-frame. Splitting into real and imaginary parts gives us the following:

$$\begin{cases} v_{Z_{VI}d} = R_{VI} i_d + L_{VI}\left[\omega_{HP} i_d - \omega_{HP} i_{d,LP} + \cancel{\hat{\omega} i_{q,LP}}\right] - \cancel{\hat{\omega} L_{VI} i_{q,LP}}, \\ v_{Z_{VI}q} = R_{VI} i_q + L_{VI}\left[\omega_{HP} i_q - \omega_{HP} i_{q,LP} - \cancel{\hat{\omega} i_{d,LP}}\right] + \cancel{\hat{\omega} L_{VI} i_{d,LP}}. \end{cases} \tag{2.21}$$

Notice that the last two terms in both equations will cancel, leaving us with:

$$\begin{cases} v_{Z_{VI}d} = R_{VI} i_d + \omega_{HP} L_{VI}\left(i_d - i_{d,LP}\right), \\ v_{Z_{VI}q} = R_{VI} i_q + \omega_{HP} L_{VI}\left(i_q - i_{q,LP}\right). \end{cases} \tag{2.22}$$

2.3.3. PID-CURRENT-CONTROLLER MODELING

Next, we will describe the modeling of two *dq*-frame PID controllers. Because the voltage-sensor-less virtual interface-impedance control algorithm requires some small amount of derivative gain in the control loop for stability (explained in a later section), we must augment the model developed in Chapter 1 to include a derivative term. As done in the previous section, to implement a derivative, we can use a high-pass filter with a high cut-off frequency.

Building from what was developed in Chapter 1, we can define two quantities which are the low-pass-filtered values of the error signal of the currents, $\varepsilon_{d,LP}$ and $\varepsilon_{q,LP}$. These will be used in the PID-controller's derivative calculation. The space-vector version of the low-pass filtered error signals can be expressed as:

$$\vec{\varepsilon}_{dq,LP} = \frac{\omega_D}{s+\omega_D}\vec{\varepsilon}_{dq} \qquad \Rightarrow \qquad \frac{d\vec{\varepsilon}_{dq,LP}}{dt} = \omega_D\left(\vec{\varepsilon}_{dq} - \vec{\varepsilon}_{dq,LP}\right), \tag{2.23}$$

where ω_D is the cut-off frequency of the low-pass filter. Rearranging and expanding this into real and imaginary parts gives:

$$\begin{cases} \dfrac{d\varepsilon_{d,LP}}{dt} = \omega_D\left(\varepsilon_d - \varepsilon_{d,LP}\right), \\ \dfrac{d\varepsilon_{q,LP}}{dt} = \omega_D\left(\varepsilon_q - \varepsilon_{q,LP}\right). \end{cases} \tag{2.24}$$

With the low-pass filter of the error defined, we can express the total derivative component of the PID-controller output as follows:

$$\frac{s\omega_D}{s+\omega_D} k_{d,c}\vec{\varepsilon}_{dq} = k_{d,c}s\vec{\varepsilon}_{dq,LP} = k_{d,c}\omega_D\left(\vec{\varepsilon}_{dq} - \vec{\varepsilon}_{dq,LP}\right). \tag{2.25}$$

With the derivative modeled, we can express the output of the PID-controllers, which, when combined with the axis-decoupling terms, form the inverter output voltages. We will use the other definitions of Q_d and Q_q from Chapter 1 as well:

$$\begin{cases} v_{Id} = \varepsilon_d k_{p,c} + Q_d k_{i,c} + \left(\varepsilon_d - \varepsilon_{d,LP}\right)\omega_D k_{d,c} - \hat{\omega}\left(L_I + L_{VI}\right)i_q + v_{Sd}, \\ v_{Iq} = \varepsilon_q k_{p,c} + Q_q k_{i,c} + \left(\varepsilon_q - \varepsilon_{q,LP}\right)\omega_D k_{d,c} + \hat{\omega}\left(L_I + L_{VI}\right)i_d + v_{Sq}. \end{cases} \tag{2.26}$$

Note that the decoupling terms have also been modified here to be based on the sum of the real interface inductance, L_I, and any virtualized interface inductance, L_{VI}. In the proposed control algorithm, the value of $L_I = 0$, so the decoupling term will reduce to only respond to L_{VI}.

2.3.4. LINEAR SYSTEM MODEL FOR VIRTUAL INTERFACE-IMPEDANCES

After defining the additional state variables in the previous sections, we can summarize all of developed mathematical relationships and provide a complete set of all the differential equations used to describe the proposed control algorithm. The other definitions from Chapter 1 regarding the PLL dynamics and the *xy*-frame current dynamics remain unchanged. The set of algebraic equations which relate various values to the state variables are summarized below. These are in addition to the four reference-frame translation equations in (2.14):

$$
\begin{cases}
(5) & v_{Id} = \varepsilon_d k_{p,c} + Q_d k_{i,c} + \left(\varepsilon_d - \varepsilon_{d,LP}\right)\omega_D k_{d,c} - \hat{\omega}\left(L_I + L_{VI}\right) i_q + v_{Sd}, \\
(6) & v_{Iq} = \varepsilon_q k_{p,c} + Q_d k_{i,c} + \left(\varepsilon_q - \varepsilon_{q,LP}\right)\omega_D k_{d,c} + \hat{\omega}\left(L_I + L_{VI}\right) i_d + v_{Sq}, \\
(7) & v_{Z_{VI}d} = R_{VI} i_d + \omega_{HP} L_{VI}\left(i_d - i_{d,LP}\right), \\
(8) & v_{Z_{VI}q} = R_{VI} i_q + \omega_{HP} L_{VI}\left(i_q - i_{q,LP}\right), \\
(9) & v'_{Id} = v_{Id} - v_{Z_{VI}d}, \\
(10) & v'_{Iq} = v_{Iq} - v_{Z_{VI}q}, \\
(11) & \hat{\omega} = k_{p,\theta}\varepsilon_\theta + k_{i,\theta}\gamma.
\end{cases}
\tag{2.27}
$$

Altogether, the complete control algorithm ends up being a 14th-order non-linear dynamic system! This will undoubtedly be very difficult to analyze by hand and will require computer-aided analysis. The set of differential equations and a short phrase indicating where each equation is used are presented in (2.28) below.

$$
\begin{cases}
(1) & \dfrac{di_x}{dt} = \dfrac{1}{L_G}\left[-R_G i_x + \omega_G L_G i_y + v_{Sx} - v_{Gx}\right], & \text{Circuit Dynamics} \\
(2) & \dfrac{di_y}{dt} = \dfrac{1}{L_G}\left[-R_G i_y - \omega_G L_G i_x + v_{Sy} - v_{Gy}\right], & \text{Circuit Dynamics} \\
(3) & \dfrac{di_{x,LP}}{dt} = \dfrac{1}{T_{sw}}\left[i_x - i_{x,LP} + \omega_G T_{sw} i_{y,LP}\right], & \text{ADC Delay} \\
(4) & \dfrac{di_{y,LP}}{dt} = \dfrac{1}{T_{sw}}\left[i_y - i_{y,LP} - \omega_G T_{sw} i_{x,LP}\right], & \text{ADC Delay} \\
(5) & \dfrac{dQ_d}{dt} = \varepsilon_d = i_d^{\,*} - i_d, & dq\text{-PID Integrator} \\
(6) & \dfrac{dQ_q}{dt} = \varepsilon_q = i_q^{\,*} - i_q, & dq\text{-PID Integrator} \\
(7) & \dfrac{di_{d,LP}}{dt} = \omega_{HP} i_d - \omega_{HP} i_{d,LP} + \hat{\omega}\, i_{q,LP}, & \text{Virtual Inductor} \\
(8) & \dfrac{di_{q,LP}}{dt} = \omega_{HP} i_q - \omega_{HP} i_{q,LP} - \hat{\omega}\, i_{d,LP}, & \text{Virtual Inductor} \\
(9) & \dfrac{d\varepsilon_{d,LP}}{dt} = \omega_D\left(\varepsilon_d - \varepsilon_{d,LP}\right), & dq\text{-PID Derivative} \\
(10) & \dfrac{d\varepsilon_{q,LP}}{dt} = \omega_D\left(\varepsilon_q - \varepsilon_{q,LP}\right), & dq\text{-PID Derivative} \\
(11) & \dfrac{dv_{Sd}}{dt} = \dfrac{1}{T_{sw}}\left[v'_{Id} - v_{Sd} + \hat{\omega} T_{sw} v_{Sq}\right], & \text{PWM Delay} \\
(12) & \dfrac{dv_{Sq}}{dt} = \dfrac{1}{T_{sw}}\left[v'_{Iq} - v_{Sq} - \hat{\omega} T_{sw} v_{Sd}\right], & \text{PWM Delay} \\
(13) & \dfrac{d\delta}{dt} = k_{p,\theta}\varepsilon_\theta + k_{i,\theta}\gamma - \omega_G, & \text{PLL Angle Difference} \\
(14) & \dfrac{d\gamma}{dt} = \varepsilon_\theta = \dfrac{v_{Sq}}{\sqrt{{v_{Sd}}^2 + {v_{Sq}}^2}}. & \text{PLL-PI Integrator}
\end{cases}
\tag{2.28}
$$

From here, the 14×14 Jacobian matrix can be computed and evaluated at an equilibrium point. This can then be used to check for local stability. While useful, this analytical procedure does not offer much insight on the limitations for tuning the PI/PID-controllers, or for selecting the values of the high-pass filter cut-off frequencies. In the next section, we will attempt to provide some rough guidelines on the tuning of the PID parameters to maintain stability.

2.3.5. CURRENT-CONTROLLER TUNING USING PADÉ APPROXIMANT

In an effort to reduce the complexity of the control system down to something which can be analyzed by hand, allowing us to gain some insight on the limitations for tuning the PID-controllers, we will make the following simplifications to the previous dynamic system model:

(a) transfer all differential equations into the *dq*-frame,

(b) assume that all of the cross-coupling terms are negligible or cancellable, and

(c) assume that pure derivatives are used for the virtual inductance and PID-controller.

The first two assumptions will allow us to remove the PLL dynamics from our analysis and completely decouple the two axes of the *dq*-frame. Thus, this will allow us to ignore 8 of the 14 differential equations from the previous page. The last assumption will allow us to further eliminate the two low-pass filters, eliminating another 2 differential equations. The resulting system will be only 4^{th}-order, which should be much easier to analyze. However, it will only tell us some information regarding the tuning of the PID-controllers.

A depiction of the simplified virtual-impedance control system is shown in Figure 2.11. The equivalent "plant" in the control system is effectively just the transformer/grid impedance, Z_G, since the real interfacing impedance, Z_I, has been removed (as previously discussed and shown in Figure 2.6); the inverter output is now directly connected to the PLL sensing point (the transformer terminals). The control system in Figure 2.11 is expressed in the *dq*-frame; the plant represents the transfer function of an RL circuit moved to the *dq*-frame without the cross-axis coupling:

$$\frac{I_D(s)}{V_D(s)} = \frac{I_Q(s)}{V_Q(s)} = \frac{1}{sL_G + R_G}, \tag{2.29}$$

where R_G and L_G are the grid's resistance and inductance (as introduced previously).

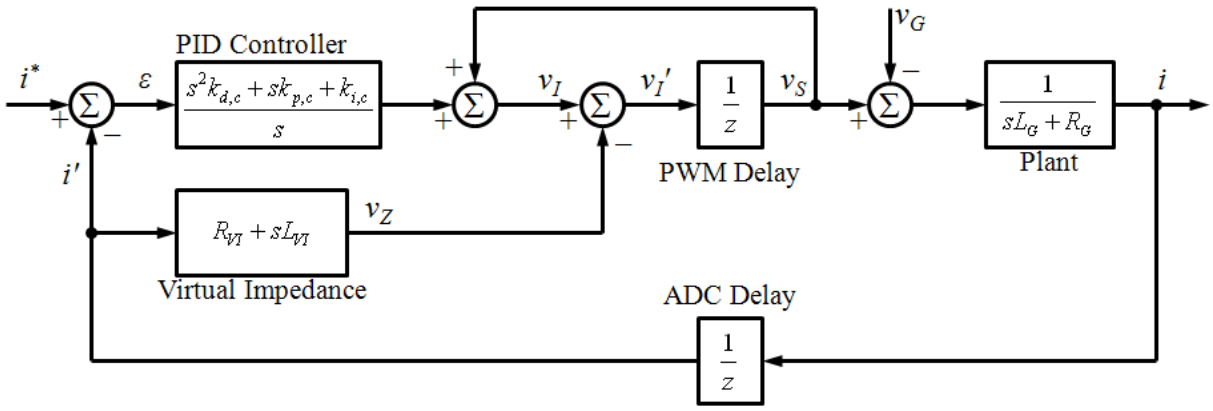


Figure 2.11: Equivalent dynamic system model of virtual-impedance control algorithm.

To improve the model somewhat, after all of the simplifications above, we can use a different approximation for the unit-time delays. Another method of capturing the behavior of discrete delays in a continuous system is to use the Padé approximant. While similar to the Taylor Series, the Padé approximant is in some cases a "better" approximation of particular non-linear functions by a rational function of a given order (i.e. a ratio of polynomials with some given order). Using a first-order Padé approximant, we can express the Laplace transform of the time-delay as the following transfer function:

$$F(s) = e^{-st_0} = \frac{e^{-st_0/2}}{e^{+st_0/2}} \approx \frac{1-\frac{1}{2}t_0 s}{1+\frac{1}{2}t_0 s}; \quad t_0 \geq 0. \tag{2.30}$$

After applying the first-order Padé approximation, the control system of Figure 2.11 can be redrawn as shown in Figure 2.12 on the next page. The positive-feedback loop at the top has been collapsed, based on its equivalent transfer function, $A(s)$, which is very similar to an integrator with a very large gain. This is expressed below in equation (2.31), where the time-delay parameter, T_{sw}, is the switching period of the inverter PWM outputs (which corresponds to the PWM duty cycle and ADC update rates within the control software);

$$A(s) = \frac{F(s)}{1-F(s)} = \frac{1-s\frac{1}{2}T_{sw}}{sT_{sw}}. \tag{2.31}$$

With the system approximated, we can collapse the system to obtain the transfer function of the current-control loop. The details are omitted here, but the resulting 4th-order transfer function will have a form as given below:

$$\frac{I(s)}{I^*(s)} = \frac{\sum_{m=0}^{M} s^m b_m}{\sum_{n=0}^{N} s^n a_n} = \frac{s^4 b_4 + s^3 b_3 + s^2 b_2 + s b_1 + b_0}{s^4 a_4 + s^3 a_3 + s^2 a_2 + s a_1 + a_0}, \tag{2.32}$$

where the denominator polynomial coefficients are:

$$\begin{cases} a_4 = \left(2L_G + L_{VI} + k_{d,c}\right) T_{sw}{}^2, \\ a_3 = \left(2R_G + R_{VI} + k_{p,c}\right) T_{sw}{}^2 + \left(L_G - L_{VI} - k_{d,c}\right) 4T_{sw}, \\ a_2 = k_{i,c} T_{sw}{}^2 + 4\left(R_G - R_{VI} - k_{p,c}\right) T_{sw} + 4\left(L_{VI} + k_{d,c}\right), \\ a_1 = 4\left(R_{VI} + k_{p,c}\right) - 4k_{i,c} T_{sw}, \\ a_0 = 4k_{i,c}. \end{cases} \tag{2.33}$$

For stability, the numerator coefficients, b_n, are of less importance and are not mentioned here.

From the denominator coefficients, we can obtain some requirements for stability, based on the Routh-Horwitz Stability Criterion. Written in terms of the polynomial coefficients, the stability of a 4th-order system is given by the following set of inequalities:

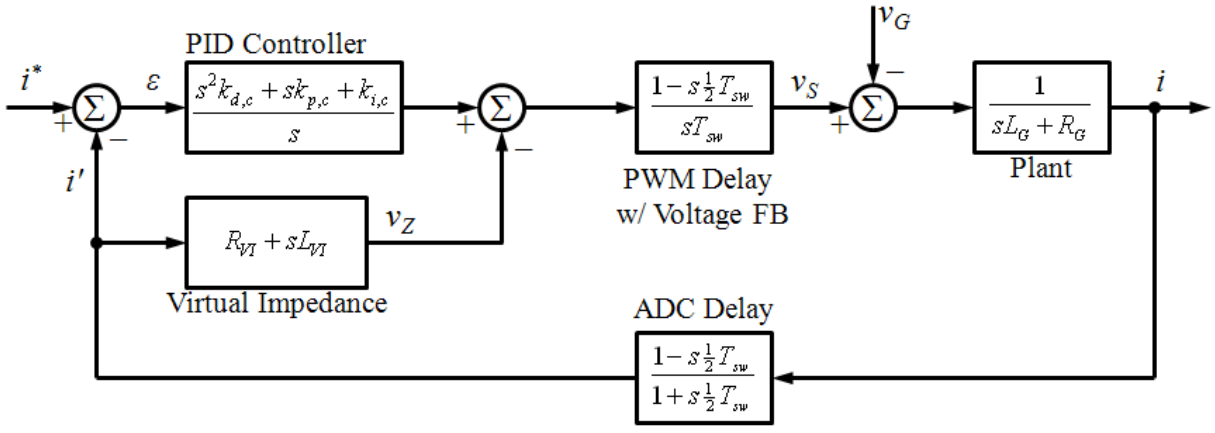


Figure 2.12: Equivalent dynamic system model of virtual-impedance control algorithm.

$$\begin{cases} a_4 > 0, \\ a_3 > 0, \\ a_2 > 0, \\ a_1 > 0, \\ a_0 > 0, \end{cases} \quad \text{and} \quad \begin{cases} a_2 a_3 > a_1 a_4, \\ a_1 a_2 a_3 > a_1^2 a_4 + a_0 a_3^2. \end{cases} \tag{2.34}$$

Assuming the device parameters, R_G, L_G, R_{VI}, and L_{VI}, to all be much smaller than $1/T_{sw}$, we can make the following observations for the stability:

From $a_3 > 0$: $$\left(L_{VI} + k_{d,c}\right) < L_G + \tfrac{1}{4} T_{sw}\left(2R_G + R_{VI} + k_{p,c}\right) \approx L_G; \tag{2.35}$$

From $a_2 > 0$: $$\left(L_{VI} + k_{d,c}\right) > \left(R_{VI} + k_{p,c} - R_G\right) T_{sw} - \tfrac{1}{4} k_{i,c} T_{sw}^2; \tag{2.36}$$

From $a_1 > 0$: $$\left(R_{VI} + k_{p,c}\right) > k_{i,c} T_{sw}. \tag{2.37}$$

Additional stability requirements can be obtained by examining the higher-order inequalities presented in (2.34). However, from the simple inequalities above, we can determine that there are upper and lower-limits to the quantity ($L_{VI} + k_{d,c}$) and a lower-limit to the quantity ($R_{VI} + k_{p,c}$).

It is interesting to note that the virtual-inductance, L_{VI}, is somewhat related to the $k_{d,c}$ derivative-gain of the controller. The two terms frequently show up together within the system transfer function. Likewise, the virtual-resistance, R_{VI}, is somewhat related to the $k_{p,c}$ proportional-gain and we could infer that a virtual-capacitance, C_{VI}, would be related to the $k_{i,c}$ integral-gain. The important thing to note here is that some small (less than ~L_S) but non-zero derivative-gain is needed for stability (but it does not matter whether it comes from L_{VI}, from $k_{d,c}$, or a little from both). This will become of more importance in the following section.

2.4. CURRENT CONTROL-LOOP SIMPLIFICATION USING VIRTUAL RESISTANCE

In a voltage-sensor-less control algorithm, the real purpose of the measurement voltage being taken after the converter's output voltage (on the other side of the interfacing impedance) is to relate the information from the current sensors to the PLL. If a current is applied in the wrong

axis (or in a different space-vector direction than estimated), a non-zero voltage will appear in the *q*-axis of the measurement point. The PLL will try to reduce this voltage to zero, realigning the system. However, for synchronization, there is no requirement that the interfacing-impedance needs to be inductive. Any type of impedance or transfer function may be used to convey the current-sensor information to the PLL. The typically-used impedance is inductive merely because of its ability to also filter the PWM voltage-harmonics from the inverter.

With the interfacing impedance becoming virtualized and the synchronization voltage measurement bypassing the PWM-stage and voltage sensors (eliminating any filtering needs), we are free to select any interfacing impedance we wish. For simplicity, the most obvious choice, which is proposed here, is to use a resistance. This will have the following two benefits:

1) The calculation of the virtual voltage-drop will be very straightforward, only consisting of three multiplication and three addition operations (if in the *abc*-frame). No integration or derivative approximations will be required, reducing computational complexity.

2) The control of the *dq*-currents can be simplified. The commonly-used decoupling terms are implemented to cancel the steady-state voltage drop of an inductance in the system. With that inductance removed, the decoupling terms are no longer needed. The *dq*-axis currents and voltages will inherently be decoupled from each other. This further reduces computational complexity of the overall control algorithm.

The regulation of the *dq*-currents can still be done using a PID-controller in each axis. The derivative-term in the controller is needed to ensure the stability; this was hinted at in the previous section and will be shown again here soon. The dynamics will also change due to the interfacing impedance now being just a resistor. The equivalent virtual-resistance control system (with the delays substituted by the 1st-order Padé approximation) is shown below in Figure 2.13.

In the figure, the positive-feedback loop of the sensing-point voltage has already been collapsed, as discussed previously and shown in equation (2.31). The transfer function of this system will also be a 4th-order system, having a form the same as in equation (2.32). The denominator coefficients are given below:

$$\begin{cases} a_4 = \left(2L_G + k_{d,c}\right){T_{sw}}^2, \\ a_3 = \left(4L_G - 4k_{d,c}\right)T_{sw} + \left(2R_G + R_{VI} + k_{p,c}\right){T_{sw}}^2, \\ a_2 = 4k_{d,c} + 4\left(R_G - R_{VI} - k_{p,c}\right)T_{sw} + k_{i,c}{T_{sw}}^2, \\ a_1 = 4\left(R_{VI} + k_{p,c}\right) - 4k_{i,c}T_{sw}, \\ a_0 = 4k_{i,c}. \end{cases} \tag{2.38}$$

From the transfer function, we can make some similar observations on the stability, as was done in the previous section. We'll assume the resistances, R_G and R_{VI}, to be larger than the grid inductance, L_G, or sampling period, T_{sw}. We'll also assume the sampling period to be slightly less than or equal to the grid inductance: $T_{sw} \leq L_G$. From these assumptions, we can make the following observations for the stability:

From $a_3 > 0$: $$k_{d,c} < \left(L_G + \tfrac{1}{4}T_{sw}\left(2R_G + R_{VI} + k_{p,c}\right)\right) \approx L_G; \tag{2.39}$$

From $a_2 > 0$: $$k_{d,c} > \left(R_{VI} + k_{p,c} - R_G\right)T_{sw} - \tfrac{1}{4}k_{i,c}{T_{sw}}^2 \approx 0; \tag{2.40}$$

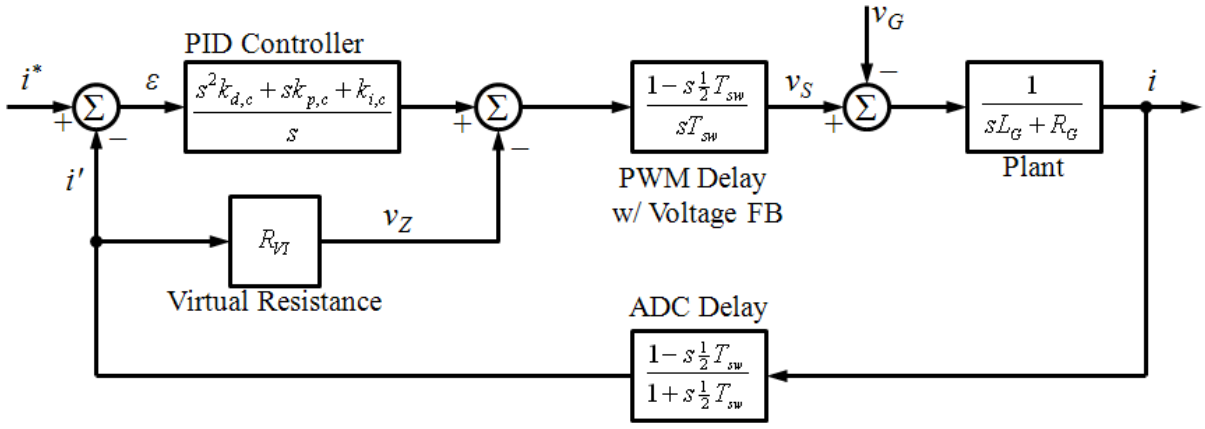


Figure 2.13: Padé-approximated dynamic system model of virtual-resistance control algorithm.

From $a_1 > 0$: $$\left(R_{VI} + k_{p,c}\right) > k_{i,c}T_{sw}. \tag{2.41}$$

Here, we reach very similar conclusions as were found in the previous section. The derivative controller-gain, $k_{d,c}$, itself now has an upper and lower limit. Also, the sum of proportional gain and virtual resistance must exceed some value, $k_{i,c}T_{sw}$. The main difference here, with $L_{VI} = 0$, is that now the derivative-gain, $k_{d,c}$, must be non-zero to ensure stability of the system.

If we can further assume that the positive-feedback-path from the inverter output-voltage to the sensing-voltage is sufficient so that the KVL equation (1.21) still holds, then we can further approximate the control system to only incorporate the PWM and ADC delays while letting the virtual-impedance function as the system plant. Equation (1.21) has been rewritten below as equation (2.42), except now Z_I has been replaced with Z_{VI}:

$$\vec{v}_S = \left(\vec{v}_G + Z_G\vec{i}\right) = \left(\vec{v}_I - \cancel{Z_I\vec{i}}\right) = \left(\vec{v}_I - Z_{VI}\vec{i}\right)/z = \vec{v}_I'/z. \tag{2.42}$$

This requires that the sampling rate of the control loop be sufficiently small, such that the phase delay between v_S and v_I' is negligible.

As mentioned previously, the use of a virtual interfacing-resistance means that the inductive decoupling-terms of the *dq*-control algorithm can be eliminated. Thus, the proposed *dq*-control system will effectively reduce to the system shown in Figure 2.14. Conveniently, with the decoupling terms removed, the estimated frequency, $\hat{\omega}$, is no longer needed by the control system. With this, a low-pass filter can potentially be removed from the PLL system as well (in some cases, the estimated frequency may be low-pass filtered to remove noise coming from the PLL voltage inputs). The addition of the sensing-point voltage, v_S, is accounted for already in the previous control system diagram (Figure 2.13), but it is included here for comparison to the traditional method.

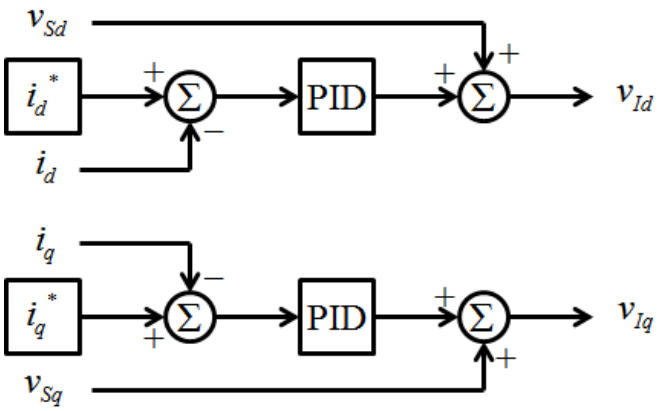


Figure 2.14: Simplified *dq*-frame current-controller without decoupling terms.

Going one step further, if the KVL equation (2.42) holds, then we can also develop an equivalent control-system model where the virtual-impedance functions as the plant. In this equivalent model, we will assume that the dynamics of the current will be tuned to be much slower than the single-cycle Padé-approximated delays. This allows us to also ignore the delay blocks, yielding a lower-order system with which we can analyze and easily estimate the dynamics. This simplified control system is shown below in Figure 2.15. The closed-loop transfer function of this system will be the following:

$$TF_{DQ} = \frac{I(s)}{I^*(s)} = \frac{s^2 k_{d,c} + s k_{p,c} + k_{i,c}}{s^2 k_{d,c} + s\left(R_{VI} + k_{p,c}\right) + k_{i,c}}. \tag{2.43}$$

The poles of this approximated system can be easily found; they depend only on the PID controller tuning and the virtual resistance value (as expected from the original proposed system description). The values of the poles are listed here:

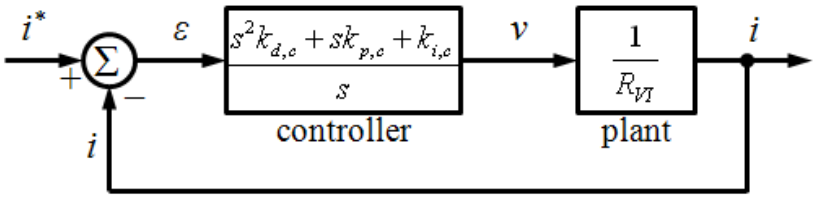


Figure 2.15: Equivalent PID-control system of the *dq*-current controllers using virtual-resistance.

$$s = \frac{R_{VI} + k_{p,c}}{2k_{d,c}} \pm \sqrt{\left(\frac{R_{VI} + k_{p,c}}{2k_{d,c}}\right)^2 - \frac{k_{i,c}}{k_{d,c}}}. \tag{2.44}$$

Further simplification can be made by setting the proportional-gain to zero: $k_{p,c} = 0$. The virtual resistance acts similarly to a proportional gain; the two parameters, R_{VI} and $k_{p,c}$, show up constantly together within the system transfer-functions within this chapter. From the previous stability analyses, there should be no issue having $k_{p,c} = 0$ so long as R_{VI} is sufficiently chosen. Next, recall that the derivative gain has an upper-bound, related to the inductance, L_G. Since the grid inductance may usually be in the milli-Henry range or smaller, we can approximate this value to be roughly equal to zero. With $k_{p,c} = 0$ and $k_{d,c} \approx 0$, the two left-half-plane zeros on the numerator of the transfer function in (2.43) will become negligibly large. Furthermore, one of the poles on the denominator will become extremely fast, making the other closed-loop pole dominate (and the transfer function will be overdamped, have a simple, almost first-order response). The control loop for the current will take the form shown in Figure 2.16 and the transfer function will reduce to:

$$k_{p,c} = 0, \quad k_{d,c} \approx 0 \qquad \Rightarrow \qquad TF_{DQ} = \frac{I(s)}{I^*(s)} \approx \frac{k_{i,c}}{sR_{VI} + k_{i,c}}. \tag{2.45}$$

Based on these results, the control of the current (if a pure integral-controller is used) will always take the form of a first-order low-pass filter, making the system have no overshoot, while being very easy to tune. The bandwidth of the current (the cut-off frequency) will be:

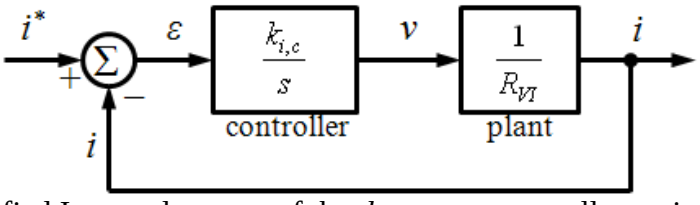


Figure 2.16: Simplified I-control system of the *dq*-current controllers using virtual-resistance.

$$\omega_{dq} = \frac{k_{i,c}}{R_{VI}}. \tag{2.46}$$

Assuming roughly four time-constants for the settling time, we also can get the expression:

$$T_{st,c} = \frac{4R_{VI}}{k_{i,c}}. \tag{2.47}$$

The limitation of this tuning approximation depends on the 2nd-order poles listed in equation (2.44). As the expression under the radical becomes closer to zero or less than zero, the system will become critically or under-damped, causing overshoot and ringing in the response of the current. The system can be maintained to have a first-order-like over-damped response so long as the following condition holds:

$$\left(R_{VI} + k_{p,c}\right)^2 > 4k_{d,c}k_{i,c}. \tag{2.48}$$

These claims are verified at the end of this chapter through MATLAB Simulink simulation and experimental results.

2.5. COMPUTATIONAL COMPLEXITY COMPARISON OF CONTROL METHODS

Moving on, as a means for comparison, the total number of mathematical calculations (add, subtract, multiply, divide, square-root, and table look-ups) required by the different control system variations can be examined. Before performing this comparison however, we must make some assumptions on the number of clock cycles required to perform each mathematical operation. A common platform for controlling a power-electronics system is to use a Digital Signal Processor (DSP) which has floating-point calculation support. The implementation of the Floating-Point Unit (FPU) is often unevenly distributed among the mathematical calculations (addition, multiplication, etc…) mentioned above. Addition and multiplication are often extremely efficient, while other operations like division and square-roots are somewhat slow

[21]. To form a basis for comparison, the number of clock cycles required to perform each mathematical operation will be assumed to have the values listed in Table 2.1. Addition, subtraction, multiplication, and table look-ups will be penalized by only 1 clock cycle. Computing saturation limits (used within PID-controllers) will be penalized 6 clock cycles, since this usually involves the use of two if-statements requiring a logical check, a branch statement, and the execution of the saturation. Division will be arbitrarily penalized by 20 clock cycles and the square-root operation will be 40 clock cycles.

Table 2.1: Assumed number of FPU clock-cycles required for basic mathematical operations.

	Add/Sub	**Multiply**	**Look-Up**	**Saturate**	**Divide**	**Sqr-Root**
# Clk Cycles	1	1	1	6	20	40

From the assumptions in Table 2.1, we can examine the total number of clock cycles required to implement some basic control-system tasks. These are presented in Table 2.2. The first several columns of the table list the frequency of each type of mathematical operation occurring in the listed task. The last column of the table lists the number of clock cycles required to implement each task, which is the product of the number of occurrences of each operation and the number of cycles required for each operation, each summed together. For example, a PI-controller

Table 2.2: Number of FPU operations and clock-cycles required for common control-system tasks.

Task	**Add/Sub**	**Multiply**	**Look-Up**	**Saturate**	**Divide**	**Sqr-Root**	**# Cycles**
Clarke Xfrm	3	4					7
Park Xfrm	2	4	2				8
Integrator	1	1					2
LPF / HPF	2	2					4
PI-Controller	2	3		2			17
PID-Controller	5	6		2			23
Inv. Hypot.	1	2			1	1	63

requires 2 additions, 3 multiplications, and 2 saturation checks. The number of clock cycles required to implement a PI-controller will therefore be: (2×1) + (3×1) + (2×6) = 17 cycles. The last row of Table 2.2 stands for "inverse hypotenuse", which is used within the PLL to normalize the angle error, described previously in equation (1.14).

With this basis for comparison, we will consider here three systems: a traditional inverter control system (using a physical inductor and voltage sensors), one with a virtualized inductor, and one using a virtualized resistor. The computational loads for each system are presented below in Table 2.3. Note that the PWM calculations and the outer control-loop, which is used to generate the *dq*-current references, i_{dq}^{*}, will be excluded. The three main columns of each system in Table 2.3, labeled as "PLL", "*dq*-con", and "Vir-Im", correspond to the implementation of the phase-locked loop, *dq*-current controllers, and any virtual impedances, respectively. The improvements or degradations to the computational times (against the classical control) are color-coded and bolded in green or red, respectively. Detailed explanations for the modification

Table 2.3: Comparison of computational-load for three different *dq*-control systems.

Task (#cyc)	**Classical System**			**Virtual Inductance**			**Virtual Resistance**		
	PLL	*dq*-con	Vir-Im	PLL	*dq*-con	Vir-Im	PLL	*dq*-con	Vir-Im
Add/Sub/Mult (×1)	1	10	0	1	10	**12**	1	**4**	**6**
Clarke Xfrm (×7)	1	2		**0**	2		**0**	2	
Park Xfrm (×8)	1	2		**0**	2		**0**	2	
Integrator (×2)	1			1			1		
LPF / HPF (×4)	1		0	1		**3**	**0**		**0**
PI-Controller (×17)	1	2		1	**0**		1	**0**	
PID-Controller (×23)		0			**2**			**2**	
Inv. Hypot. (×63)	1			1			1		
Subtotal Cycles:	**102**	**74**	**0**	**87**	**86**	**24**	**83**	**80**	**6**
Total Cycles:	**176**			**197**			**169**		

of values in the table are provided below:

- **PLL:** in both virtualized systems, the Park & Clarke transforms are not needed since no voltage sensors are used. The voltage values from the *dq*-control algorithm can directly be used. In the virtual-resistance case, the PLL no longer needs to send the frequency estimate, $\hat{\omega}$, to the *dq*-controllers for decoupling, so a low-pass filter can be removed.
- ***dq*-Con:** in both virtualized systems, a PID controller must be used to guarantee stability. (The virtual-inductor algorithm could potentially omit the derivative-term if the virtual inductance, L_{VI}, is sufficiently chosen.) The virtual-resistance algorithm sees a small improvement since the decoupling terms are no longer needed.
- **Vir-Im:** both virtualized systems experience a loss in speed due to the added calculations for the virtual-voltages. The virtual-resistance case is more efficient than the virtual-inductance case since the high-pass filters (for derivative approximation) are not needed.

The two virtualized cases also assume that the virtual-voltages are computed in the *abc*-frame.

From this comparison, we can see that the implementation of the virtual-resistance control algorithm actually saves a few clock cycles (about 4.0%) over the traditional non-virtualized control, even when considering the added computation of the virtualized components. This is because of the removal of the low-pass filter in the PLL (since $\hat{\omega}$ is no longer needed or filtered), the removal of the Park & Clarke transforms in the PLL, and the simplification of the *dq*-current control (the removal of the "$k_{p,c}$" terms and decoupling terms).

To further simplify the virtual-resistance-based control algorithm and make it even more computationally efficient, we can arbitrarily select an interfacing-resistance of $R_{VI} = 1\ \Omega$. Now, the three $V \times I$ multiplications needed to compute the virtual voltage-drops will vanish. Oddly enough, the calculation of the sensing-point voltage will now be mathematically equivalent to

simply subtracting the measured current-sensor values from the computed inverter output voltage. This will then be fed to the PWM and PLL blocks; while seemingly strange, this method does work! The result listed in Table 2.3 for a virtual-resistance system, 169 cycles, can be further reduced to 165 cycles (now a 6.2% reduction from the classical method) if a virtual resistance of 1 Ω is used and the virtualization is done within the *αβ*-frame or *dq*-frame, rather than the *abc*-frame. This eliminates 3 multiplications and 1 subtraction (a reduction of 4 floating-point operations). Implementing a virtual-resistance in either a stationary or rotating frame should be equivalent since no additional voltage terms arise when moving a resistor's voltage from the stationary-frame to a rotating-frame (as compared to the case for the voltage-drop of a virtual inductor or capacitor). Further improvement can be made if the $k_{p,c}$ proportional-gain terms are removed; an extra 4 cycles can be saved, increasing the reduction to 8.5%.

2.6. VIRTUAL-RESISTANCE SIMULATION & EXPERIMENTAL RESULTS

To validate the concepts discussed in this chapter, a MATLAB Simulink time-domain simulation model has been generated to simulate the inductor-less sensor-less control algorithm using a virtual-resistance for self-synchronization. The top-level diagram of the simulation model is shown in Figure 2.17. The inverter is simulated using MOSFETs with low on-state resistance (39 mΩ) to match the experimental prototype (introduced later in this section). The inverter is directly connected to the grid impedance, R_G + L_G, which connects to an ideal AC voltage

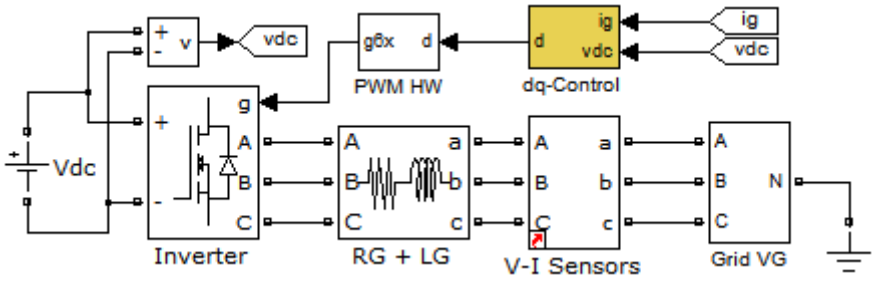


Figure 2.17: MATLAB Simulink inductor-less voltage-sensor-less virtual-resistance simulation.

source. Only the inverter DC-bus voltage and line currents are sensed and fed to the control algorithm. The true grid voltage is also sensed but not used by the controller. The inverter PWM signals are generated from the duty cycle outputs of the *dq*-controller. Inside the "PWM Hardware" block is a constant-frequency triangle carrier wave used to generate the gate-switching signals. An inverter deadband of t_{DB} = 500 ns is simulated and the inverter switching frequency is set to 10 kHz (T_{sw} = 100 µs).

These and other parameters of the simulation are summarized in Table 2.4 below. Smaller-than-usual values for the grid phase-voltage magnitude (30 V RMS) and inverter DC bus voltage magnitude (100 V) were used. This was done to make the impedances have a more significant influence on the control system since the currents were small. The total grid impedance magnitude, $|Z_G|$, was selected to be roughly 14% pu. The simulated converter was operated at a rated power output of ~1 kW.

Table 2.4: Virtual-resistance simulation electrical parameters.

Parameter	**Symbol**	**Value**	**Per-Unit**
Grid line-neutral peak voltage	V_{Gm}	$30\sqrt{2}$ V	100% pu
Inverter DC-bus voltage	V_{DC}	100 V	---
D-axis current reference	i_d^*	$10\sqrt{2}$ A	100% pu
Q-axis current reference	i_q^*	0 A	0% pu
Rated power output	S	0.9 kVA	100% pu
Per-Unit Base Impedance	Z_{base}	3 Ω	---
Grid parasitic inductance	L_G	1 mH	12.5% pu
Virtual interfacing inductance	L_{VI}	0 mH	0% pu
Grid parasitic resistance	R_G	200 mΩ	6.6% pu
Virtual interfacing resistance	R_{VI}	1 Ω	33.0% pu
Nominal grid frequency	f_g	60 Hz	---
Inverter switching frequency	f_{sw}	10 kHz	---
Inverter deadband time	t_{DB}	500 ns	---

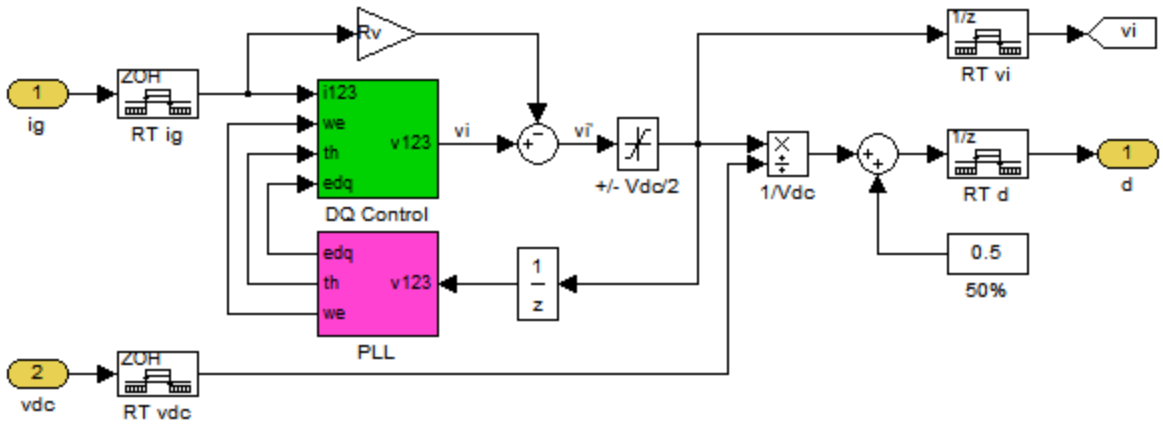


Figure 2.18: Top-level control diagram for virtual-resistance Simulink model.

The top-level diagram of the control system block is presented in Figure 2.18. The voltage drop of the virtual resistance (1 Ω in this case) is subtracted from the *dq*-controller output voltage before being sent to the PLL and output of the inverter. The PLL uses only the calculated inverter output voltage to synchronize; no external AC voltage measurements are used. A basic Sine-PWM (SPWM) algorithm is used for simplicity of the duty cycle calculations. The duty cycle outputs are sent to the "PWM Hardware" block, as mentioned previously. Additional details on the simulation model are provided within Appendix A.

The control-system parameters of the simulation are presented in Table 2.5. The *dq*-control was tuned so the current would be over-damped with a settling time of $T_{stl,c}$ = 50 ms. Meanwhile,

Table 2.5: Virtual-resistance simulation control system parameters.

Parameter	**Symbol**	**Value**
dq-control Proportional Gain	$k_{p,c}$	0
dq-control Integral Gain	$k_{i,c}$	80
dq-control Derivative Gain	$k_{d,c}$	5×10^{-4}
PLL Proportional Gain	$k_{p,\theta}$	80
PLL Integral Gain	$k_{i,\theta}$	1600
High-pass Filter Frequency	f_{HP}	1 kHz
Derivative Filter Frequency	f_D	3 kHz
Algorithm Sampling Rate	T_s	50 µs

the PLL was tuned to have a critically-damped settling time of $T_{stl,\theta}$ = 100 ms. The corresponding controller coefficients used to achieve these dynamic responses are listed in the table.

The time-domain simulation results of three different grid-disturbance cases are presented in the following figures. In each simulation output figure, the true phase-A grid voltage, v_{Ga}, the generated phase-A inverter voltage, v_{Ia}, and the phase-A grid current, i_A, are plotted vs. time. The list below describes the grid-disturbances and their corresponding figures:

- Figure 2.19 shows results of a –5 Hz frequency step-change at t = 400 ms;
- Figure 2.20 shows results of a 10% phase-angle jump (36-degrees) at t = 600 ms;
- Figure 2.21 shows results of a 10% magnitude drop in grid voltage at t = 800 ms.

In all of the simulation cases, the regulation of the current is very good during the disturbance. This is due to the very quick response of the virtual resistance (recall the virtual resistance acts similar to a proportional gain). Any change in the current immediately causes a change in the

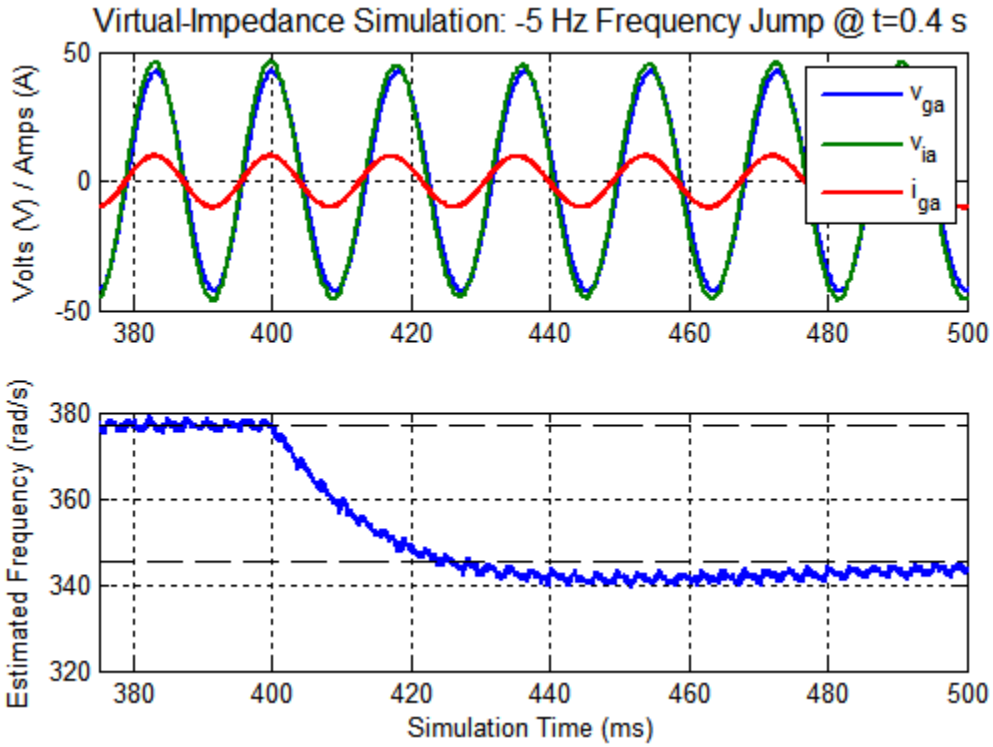


Figure 2.19: Simulink virtual-resistance simulation; 5 Hz frequency step change.

applied inverter voltage; this helps to make the control system immune to grid disturbances.

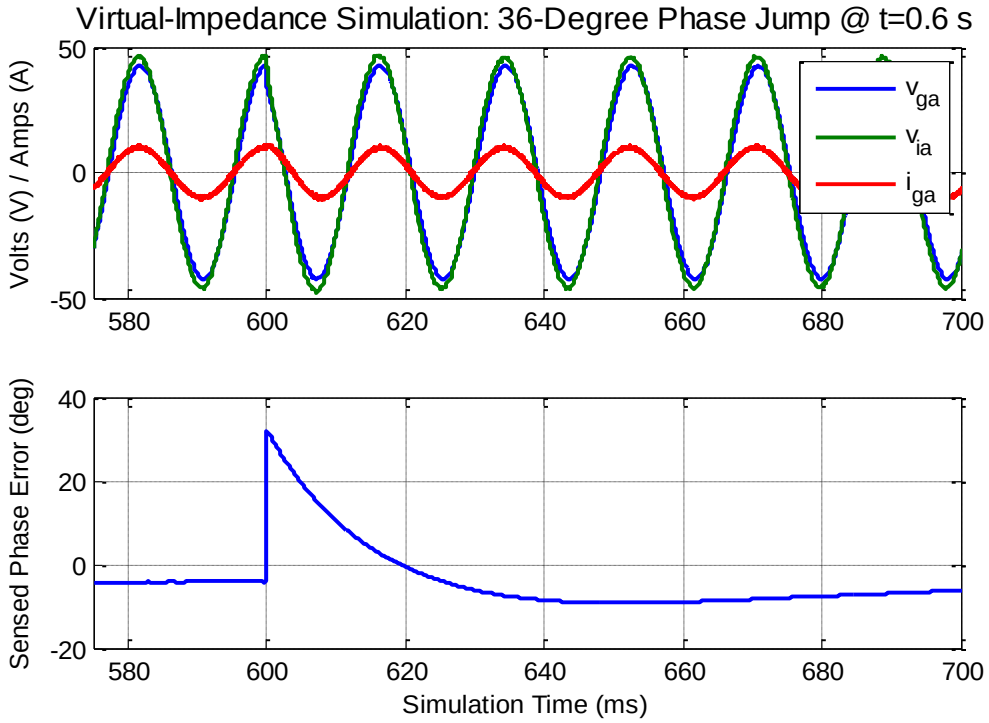


Figure 2.20: Simulink virtual-resistance simulation; 36-degree phase step change.

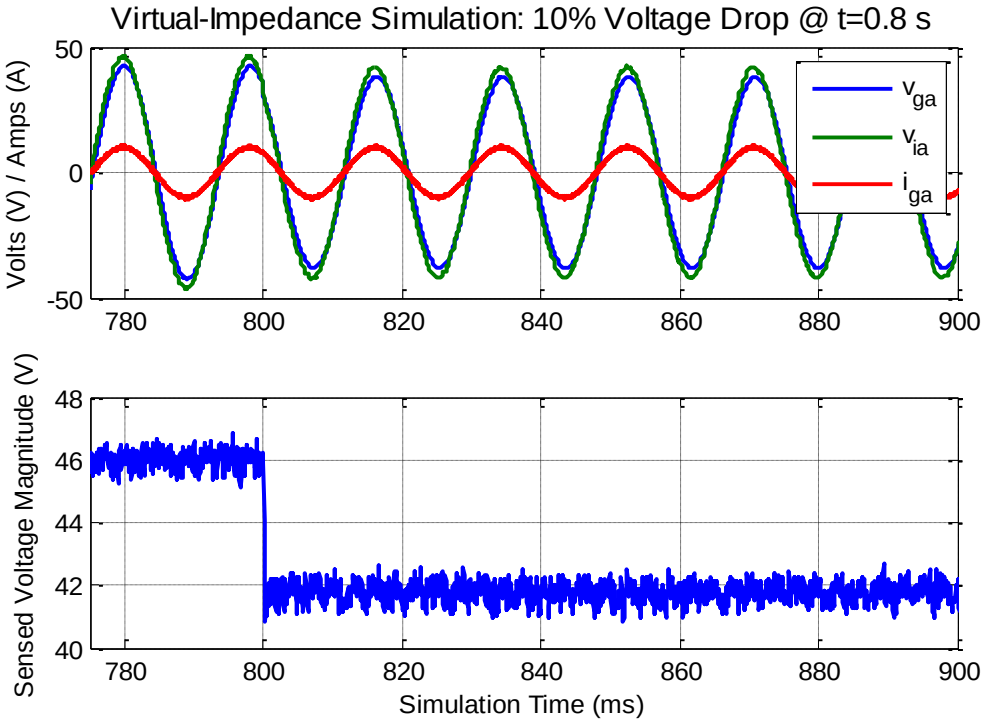


Figure 2.21: Simulink virtual-resistance simulation; 10% voltage step change.

The phase alignment of the current during the disturbances is governed by the PLL. In the frequency-jump and phase-angle-jump simulations, the PLL realigns with the grid voltage after about 100 ms, as designed. Note that the PLL is unable to regulate the phase-angle error in Figure 2.20 completely to zero. This is because of the phase-lag between the true grid voltage and the inverter connection point (which is also the sensing point) caused by Z_G. This issue can be addressed by using impedance-compensation within the PLL; this is discussed in Chapter 3.

In addition to simulation, an experimental test setup was constructed to validate the theory. The experimental setup consists of: a TDK-Lambda GEN 600-8.5 DC power supply, a SiC-based three-phase inverter, three 50A-rated 1 mH interface inductors (to simulate a non-zero grid impedance), a Staco Energy 1210B-3 variac, and a Yokogawa DL850 isolated oscilloscope. An image of the experimental test bench is shown in Figure 2.22. Here, the power supply, inverter, inductors, variac, and oscilloscope can be seen. A laptop, connected via a USB cable, was used as the man-machine interface to the inverter control software. An additional 5 V power supply was also used to power the inverter's DSP, gate driver ICs, isolated gate-drive supplies, current sensors, and voltage sensors. Additional details on the experimental test bench and the design of

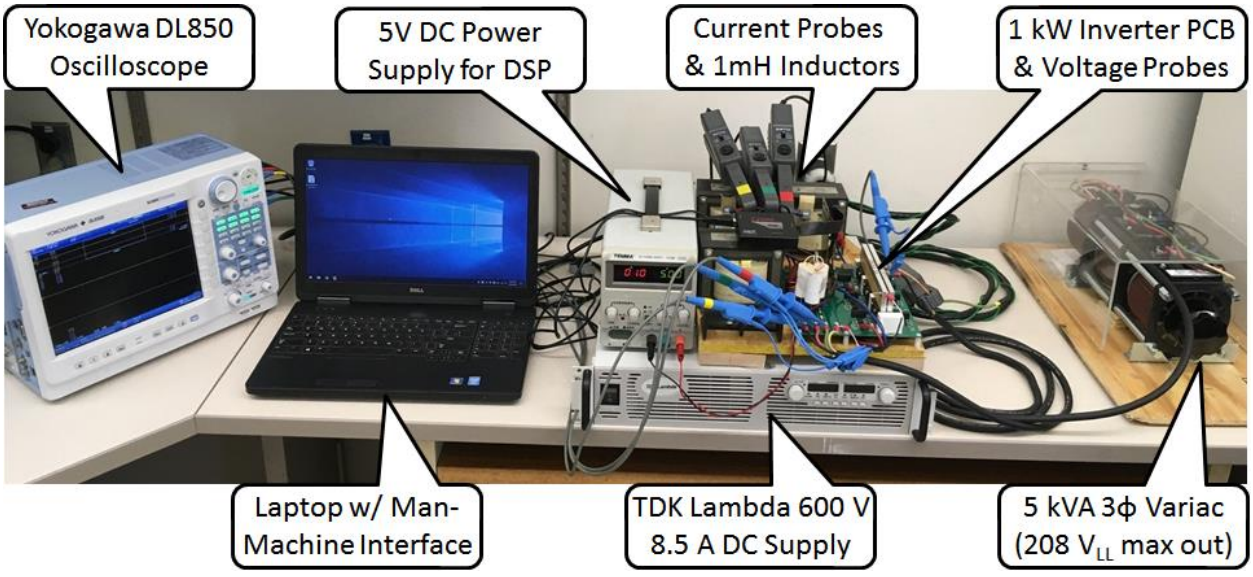


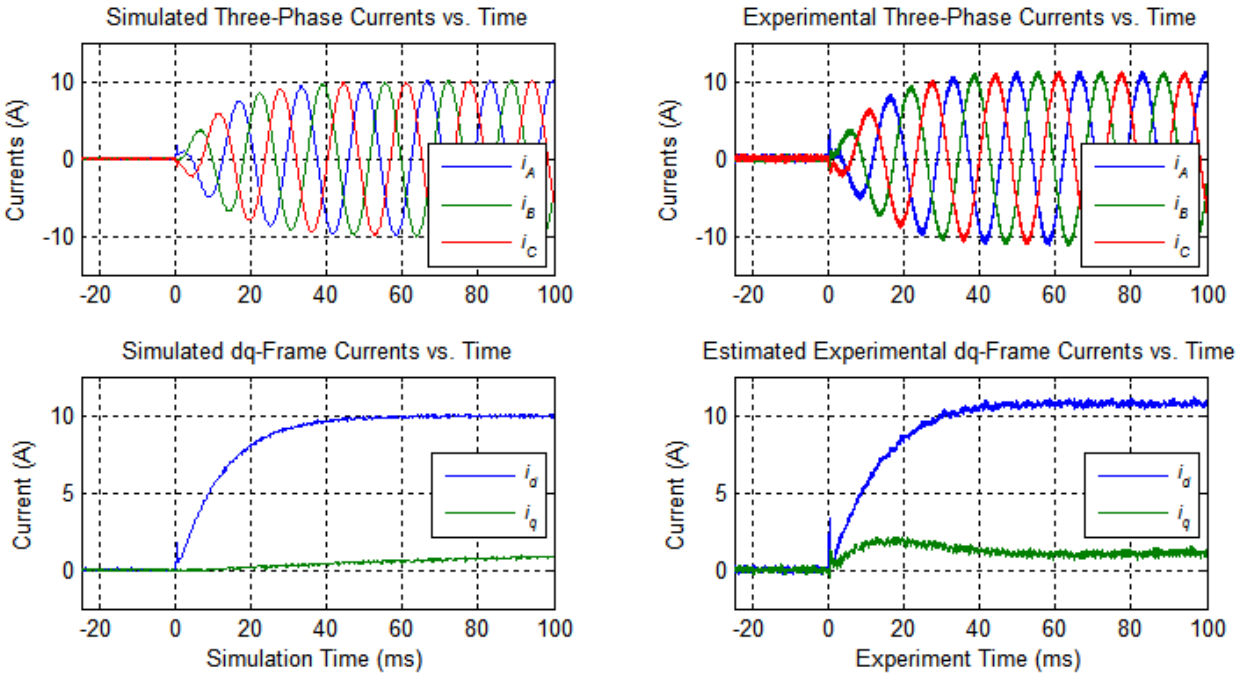
Figure 2.22: Experimental test bench for the virtual-resistance control algorithm.

the three-phase inverter are presented in Appendix B.

To confirm the theory presented in this chapter, two key experiments were performed. The first experiment was a step-change in load (the *d*-axis current) where the controller had the same tuning parameters as the previous simulation. The values in Table 2.4 and Table 2.5 (excluding $k_{p,\theta}$ and $k_{i,\theta}$) were chosen to exactly match those of the hardware test bench for this test. The PI-controller coefficients of the PLL were revised to $k_{p,\theta} = 16$ and $k_{i,\theta} = 64$ to slow down the PLL's dynamics and make $T_{stl,\theta} = 500$ ms. These simulated and experimental results are shown below in Figure 2.23. The *dq*-control of the current was tuned to set the settling time of the AC current to be ~50 ms (as previously mentioned). From equation (2.47), we had:

$$T_{st,c} = \frac{4R_{VI}}{k_{i,c}} \quad \Rightarrow \quad \frac{4(1\,\Omega)}{(80)} \approx 50 \text{ ms.} \tag{2.49}$$

The experimental results below agree very well with the simulated results and with the theory. There is a slight steady-state error in the magnitude of the experimental currents, but this is



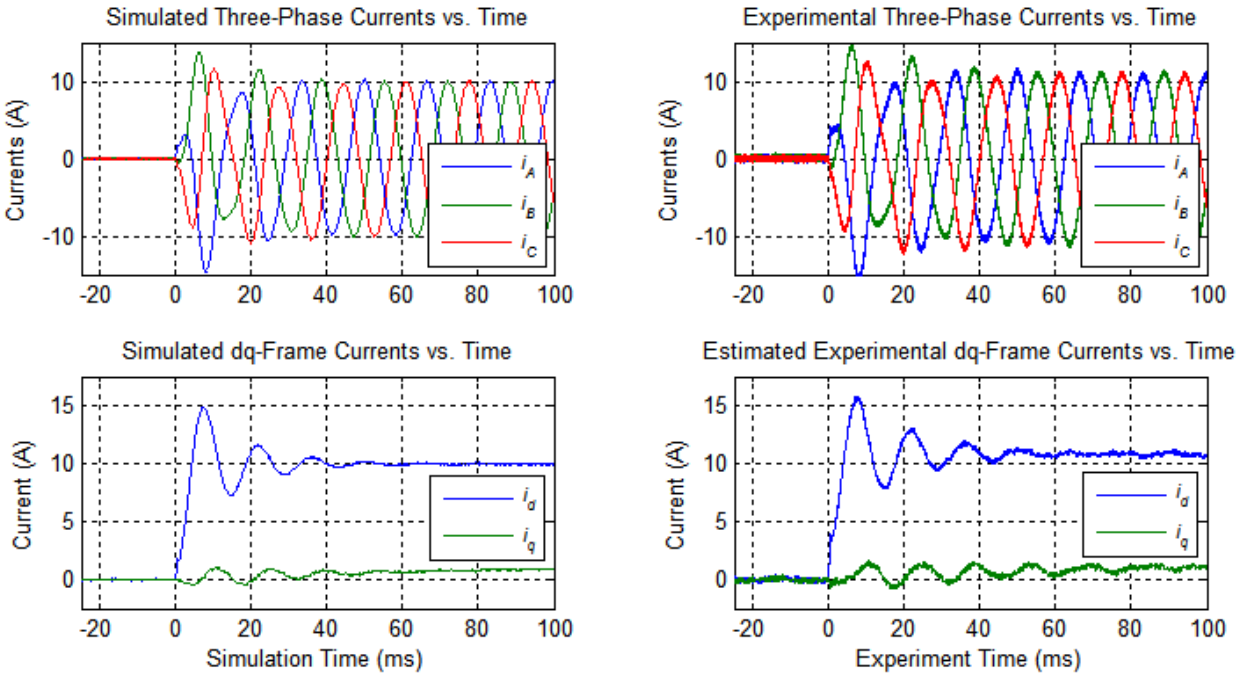
(a) (b)

Figure 2.23: Results for over-damped step-change in load current: (a) simulation, (b) experiment.

believed to be simply an ADC-sensor-gain calibration error with the current sensors on the prototype. In both plots below, we can also see the sharp spike in the current at the moment of the step-change; this is due to the derivative-component of the PID-controllers. Again, some small value of derivative-control was required to ensure the stability of the system. We can also confirm that the first-order approximation of the system dynamics should still hold, based on equation (2.48). Plugging in values gives us: 1 > 0.16, which satisfies the condition:

$$\left(R_{VI}+k_{p,c}\right)^2 > 4k_{d,c}k_{i,c} \quad \Rightarrow \quad (1+0)^2 > 4\left(5\times10^{-4}\right)(80) = 0.16\,. \tag{2.50}$$

In the second experiment, the controller tuning and virtual-resistance values were altered to yield a similar settling-time of the current, but produce an under-damped response. Specifically, the virtual resistance and integral gain were set to the following: R_{VI} = 0.1 Ω and $k_{i,c}$ = 100. As done previously, the simulated and experimental waveforms for a step-change in load (the *d*-axis current) are shown below in Figure 2.24. Using the simplified 2nd-order analysis, we can see that



(a) (b)

Figure 2.24: Results for under-damped step-change in load current: (a) simulation, (b) experiment.

the dynamics should be under-damped (the inequality below is false):

$$\left(R_{VI}+k_{p,c}\right)^2 > 4k_{d,c}k_{i,c} \quad \Rightarrow \quad (0.1+0)^2 \overset{?}{>} 4\left(5\times10^{-4}\right)(100) = 0.2\,. \tag{2.51}$$

In both Figure 2.23 and Figure 2.24, we see that the *q*-axis current was not regulated to zero; some small steady-state error began to accumulate after the step-change in the *d*-axis current. As mentioned previously, this discrepancy is due to the actual inductance of the grid affecting the phase angle of the terminal voltage (a phase-shift develops between the true grid voltage and the sensing-point voltage). Again, this issue can be addressed by impedance-compensating the PLL, to be discussed in the following chapter.

We can also compare the dynamic performance of the virtual-resistance system to the simplified 2nd-order transfer function model described by equation (2.43). However, to make this comparison fair, we must be mindful of any nonlinearities present in the real system. The PI-controllers are usually implemented to include saturation blocks; this helps to prevent integrator-wind-up and can help reduce transient current spikes by limiting the output voltage of the PI-controllers. However, the transfer function listed in (2.43) does not contain any saturation of its internal states.

To get around the saturation issue, we can instead look at a rate-limited step-response of both systems. A comparison between the full PWM-based simulation model and the simplified 2nd-order transfer function model is provided in Figure 2.25, where the true step-response is shown in (a) and the rate-limited step-response is shown in (b). The rate of the reference current was limited to ±1000 A/s in the second case.

We can see from Figure 2.25 (a) that applying a true step-input causes a large discrepancy between the realistic and simplified simulations. This is because of the derivative-term in the PID-controller. The unsaturated output from the PID-controller would normally be very large in

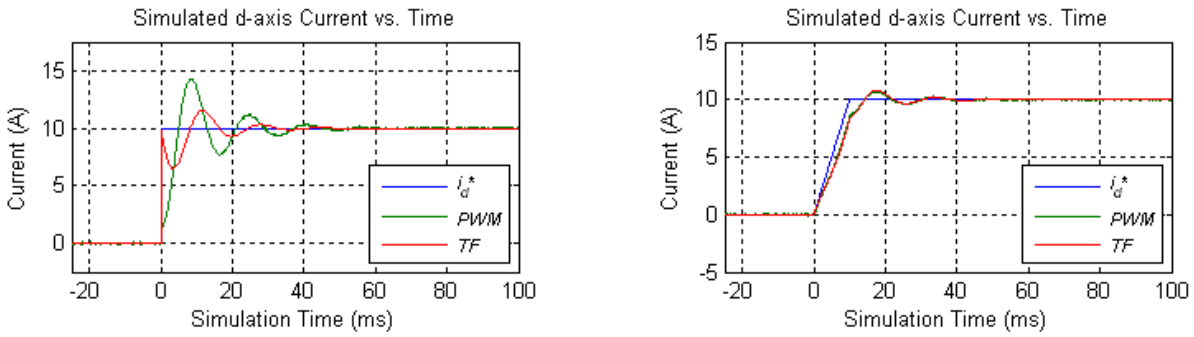


Figure 2.25: Comparison between full PWM-based simulation & simplified 2nd-order transfer function when driven by: (a) a pure step-input, (b) a 1000 A/s ramped step-input.

this case, causing a swift jump in the current. Due to the saturations added to the realistic PWM-based model, the voltage output of the PID-controller is limited to only ±25 V, causing the output current to only slightly rise before the transient response begins. This phenomenon can also be seen in the previous simulated and experimental results, shown in Figure 2.24.

In the rate-limited step-response simulation, shown in Figure 2.25 (b), the two responses are practically identical. The saturation blocks do not take effect in this case, since the slope of the error signal multiplied by the $k_{d,c}$ value is still quite small (recall that typical $k_{d,c}$ values should be roughly less than the magnitude of the grid inductance, L_G, which was around 10^{-4}). Thus, we can conclude that the 2nd-order model can provide a reasonable estimate of the response dynamics in the case of low ramp-rate transient conditions.

2.7. CHAPTER 2 SUMMARY

To conclude, virtual impedances are a useful mathematical tool to help synchronize a DC/AC power electronics converter to an AC grid. Using approximated derivatives, virtual-inductances can be used to emulate the behavior of interfacing inductors. If adequate grid-filtering is available, a virtual interfacing-impedance can be used to fully replace a physical interface impedance, simplifying the physical system and reducing the cost, size, weight, and power loss.

AC voltage sensors can also be removed from the system, being replaced by virtual voltage-sensing-points. This can further reduce system costs and complexity. Any range of virtual impedances or transfer functions may be selected to generate a virtual voltage-sensing-point. However, the most advantageous is the use of simply a virtual resistance. The computation of the virtual-resistor voltage-drops and the regulation of the *dq*-frame currents can be greatly simplified in this case, resulting in roughly 8.5% less computational load than the original system. Simulation and experimental results of a virtual-resistance self-synchronizing inverter system confirm the functionality of the algorithm and its tolerance to grid disturbances.

It should be noted that all of the virtual impedance topics presented here were applied to the inverter output voltage, without altering the voltage-synchronization point seen by the PLL. The purpose of this is to affect the perceived interfacing impedance between the inverter and voltage-sensing point. Even with the removal of the real interfacing impedance, the PLL sensing-voltage was still placed at the grid terminals, v_S. In the next chapter, we will explore some benefits of applying virtual impedances to influence the voltage-synchronization point; the purpose is to affect the total impedance between the sensing point and Thévenin-equivalent voltage source of the grid. This can impact the stability and dynamic performance of the PLL and converter, as we will soon see.

3. APPLYING VIRTUAL IMPEDANCE AT THE SYNCHRONIZATION POINT

In this chapter, some additional uses and benefits of implementing virtual impedances within DC/AC converter control algorithms are presented. First, the impedance-compensation of a PLL algorithm is discussed. This can be used to cancel out the phase-angle difference between the measurement point and the AC source, allowing for unity-power-factor at the source. A novel method is proposed and is compared with an alternate method which uses only a partially-virtualized impedance. Next, a more general approach to estimate the quasi-static stability of the SRF-PLL algorithm in a DC/AC converter system using virtual impedances is proposed. This is followed by a thorough analysis of the small-signal stability using locally linearized models. Finally, the impedance-compensated PLL algorithm, together with the virtual-resistance-based inductor-less self-synchronizing algorithm, is examined and proposed as an alternate means for implementing position-sensor-less Back-EMF-based permanent magnet (PM) motor control.

3.1. IMPEDANCE-COMPENSATED PHASE-LOCKED LOOPS

The topics presented here in this chapter differ from those in Chapter 2 by the location in which a virtual-impedance is applied within the system. As shown in Figure 3.1, we can apply a virtual impedance voltage either to the inverter output voltage or to the PLL input voltage. In

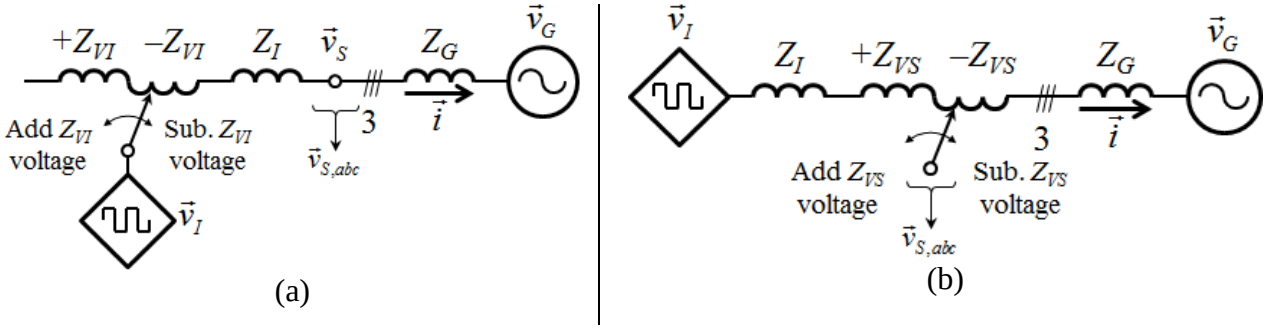
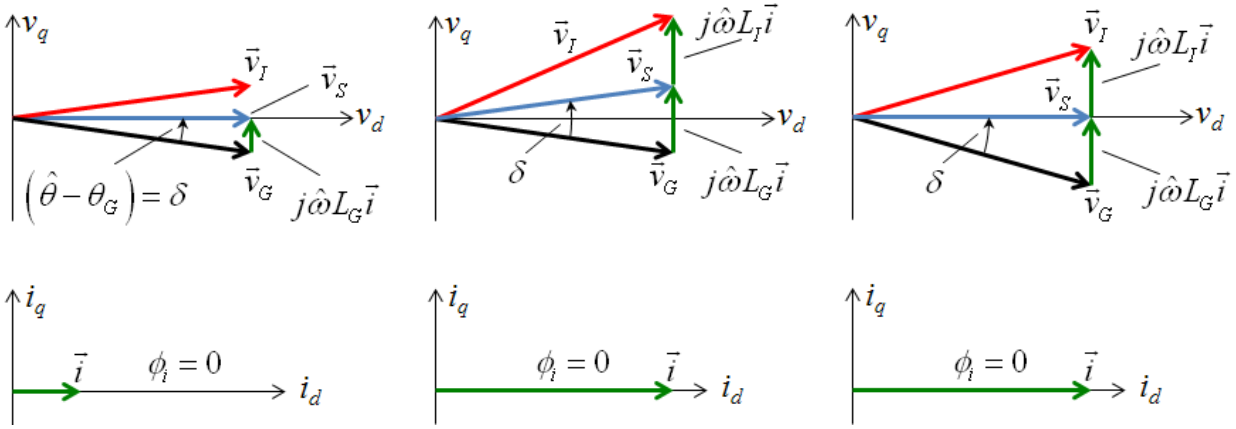


Figure 3.1: Addition or subtraction of virtual voltages from: (a) the inverter voltage; (b) the voltage sensing point.

Figure 3.1 (a), a virtual impedance applied to the inverter output voltage, denoted Z_{VI}, can shift the apparent connection point of the inverter to the grid (as seen by the controller). However, in (b), we can apply a virtual impedance to the voltage-sensing point, denoted Z_{VS}, which can shift the apparent measurement point of the PLL to be closer or further from the true grid voltage.

Shifting the apparent measurement point can be thought of as equivalently adding or cancelling out grid-side impedances. As discussed previously, along with equation (1.21), when the line currents are near zero, the PLL will synchronize to the true grid angle; the relative angle between the PLL frame and grid-frame, δ, will be small. However, at heavy loading conditions, the larger voltage drop across the grid impedance will introduce a larger phase shift at the PLL measurement point. This is illustrated in Figure 3.2 below (only inductive line impedances are considered). By applying virtual impedances at the measurement point, we can cancel out some of the voltage drop produced by the grid impedance, effectively reducing the steady-state angle-error between the PLL and the true grid angle. This can make the PLL appear to be connected to a stiffer point in the grid, moving it closer to the true Thévenin-equivalent grid voltage. In addition, this impedance cancellation (or compensation) can also alter the PLL's dynamic performance.



(a) (b) (c)

Figure 3.2: Space-vector diagram of system voltages and current (ignoring resistance) at: (a) light load; (b) load-increase transient scenario; and (c) heavy load.

To implement virtual impedances at the voltage sensing point requires modifying the PLL input voltage using impedance voltage-drops within the control software. Note that this impedance-voltage-drop can be either added or subtracted and this can be done in either the stationary-frame (fully virtualized) or the rotating-frame (usually partially virtualized). To do this, the PLL's normal sensing voltage, v_S, can be augmented to become v_S' by adding (or subtracting) the voltage-drop across a virtual impedance, Z_{VS}, calculated via Ohm's Law:

$$\vec{v}_S' = \vec{v}_S \pm Z_{VS}\,\vec{i}. \tag{3.1}$$

A block diagram illustrating the application of virtual impedance to the PLL's sensing voltage in the *dq*-frame is shown in Figure 3.3 below.

This approach was previously proposed by J. Suul, et al. in [19], using only partially implemented virtual impedances. In [19] the voltages of a virtual resistance and partial virtual

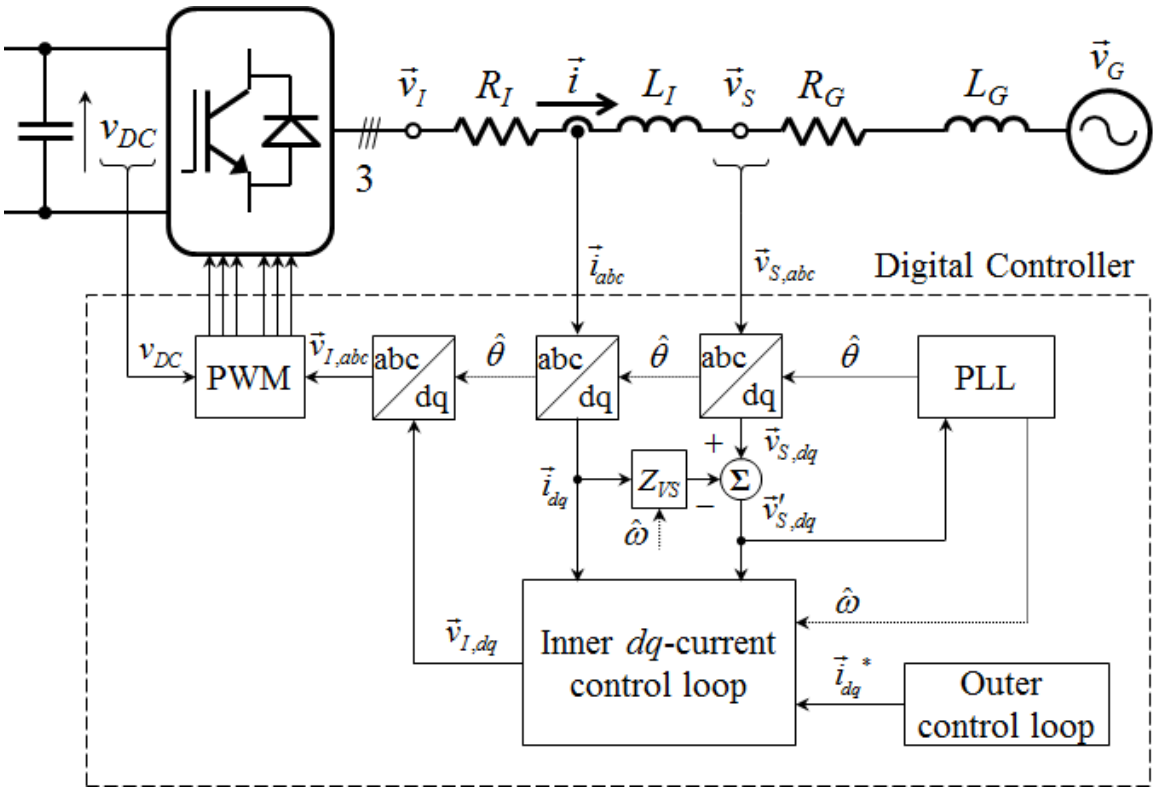


Figure 3.3: Inverter control block diagram utilizing rotating-frame virtual impedance at the voltage-sensing point.

inductance were subtracted to move the PLL voltage closer to the true grid voltage, v_G. This is expressed by the following equation:

$$\vec{v}'_{S,dq} = \vec{v}_{S,dq} - R_{VS}\vec{i}_{dq} - j\hat{\omega}L_{VS}\vec{i}_{dq}. \tag{3.2}$$

However, there are two main drawbacks with this approach. First, the virtual inductor voltage is not fully implemented. As described in Chapter 2 with equation (2.3), virtual inductors implemented in the *dq*-frame usually only include the steady-state voltage-drop term. The derivative term is ignored. This means that if the outer control loop of the inverter controller changes the *dq*-current reference values (and thus changing the actual currents), the PLL will experience a transient disturbance in its synchronization angle, until the currents again stabilize. Second, the implementation of the steady-state inductor voltage-drop relies on the estimated frequency of the PLL, $\hat{\omega}$. This not only makes analysis of the dynamics more challenging, but also means that the compensation of impedance voltages during a transient event may be slightly incorrect until the PLL's frequency estimate stabilizes.

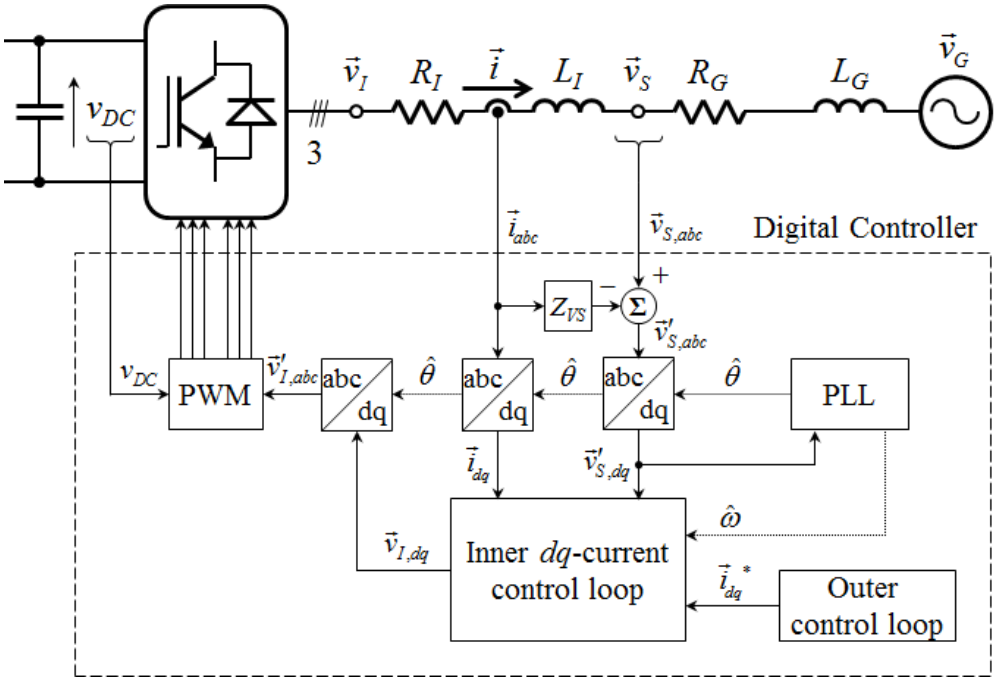


Figure 3.4: Inverter control block diagram utilizing stationary-frame virtual impedance at the voltage-sensing point.

A proposed alternative is to impedance-compensate the PLL in the stationary-frame, using the high-pass filter method described in Chapter 2. The control system block diagram for this method is shown below in Figure 3.4. With this method, the inductor voltages are fully implemented and the calculation of these inductor voltages does not rely on any inputs from the PLL, effectively decoupling the two.

As a means of comparing these two approaches, a time-domain simulation was performed in MATLAB Simulink, using similar parameters to those presented in Table 2.4. The results are shown in Figure 3.5. To emphasize the effects in a weak-grid connection, the grid inductance, L_G, was increased from 10% pu to 80 % pu. In Figure 3.5 (a) and (b), the simulation results using no impedance compensation (no comp), 50% *dq*-frame compensation (dq comp), and 50% *abc*-

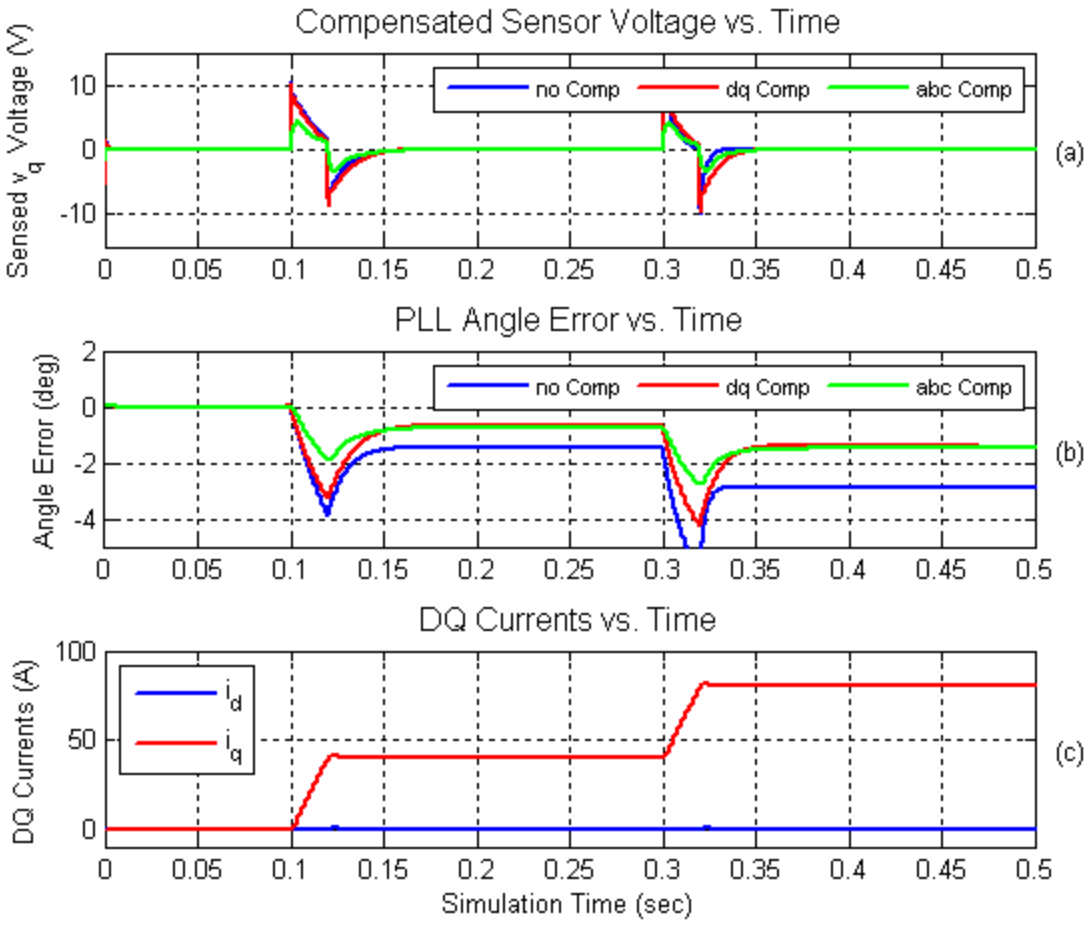


Figure 3.5: Simulink simulation comparison of impedance compensation methods.

frame compensation (abc comp) are shown. Figure 3.5 (a) shows the *q*-axis voltage as sensed by the PLL and Figure 3.5 (b) shows the PLL angle error from the true grid angle. The axis of the applied current also influences the performance of the two different impedance compensation methods. For this reason, the values of the *dq*-currents vs. time are also shown in Figure 3.5 (c). Any injected current which causes a significant change in the *q*-axis voltage will more strongly affect the PLL, causing a transient in the estimation angle. For this reason, the current applied in the simulation is all reactive, in the *q*-axis. Thus, roughly 20 kVAR of reactive power is delivered at the grid's Thévenin source. To further highlight the difference between the two compensation methods, a high ramp rate of the current was used; the current references in the simulation varied by 2000 A/s. This will cause the derivative term of the inductor voltage to be more significant, which only the proposed compensation method can address.

As shown in the figure, both the *dq*-frame and *abc*-frame impedance compensation methods are effective at eliminating some of the steady-state angle error. With no compensation, the steady-state angle error is roughly 3 degrees at full load. Both compensation methods reduce this by 50% and have a steady-state angle error of about 1.5 degrees. However, during the transient periods of the current, the *abc*-frame compensation method is affected less. We see a strong transient in the sensed *q*-axis voltage (roughly 10 V) in the uncompensated and *dq*-frame-compensated systems. The proposed *abc*-frame-compensated system has a smaller voltage spike and smaller angle-shifts during the ramps in current.

3.2. QUASI-STATIC STABILITY RANGE OF SRF-PLL FOR WEAK GRIDS

Here, we revisit the analysis of the basic SRF-PLL, presented in Chapter 1. The typical assumption is that the grid impedance, Z_G, will not affect the control of the *dq*-currents, so long as the inverter voltage is controlled to satisfy the KVL expression (1.22), restated here:

$$\vec{v}_I = \vec{v}_S + Z_I\vec{i}. \tag{1.22}$$

Restated another way, we can say that if the inverter voltage, v_I, and current, i, are controlled properly and the interface impedance, Z_I, is known, then the voltage difference between the inverter voltage and sensing-point voltage, v_S, will also be known. The grid impedance, Z_G, will not influence the voltage between inverter and sensors. It is the job of the *dq*-controllers to regulate the current and in the process, regulate this voltage difference.

In this section, we now consider the interaction between the sensing voltage, v_S, the grid impedance, Z_G, and grid voltage, v_G. Applying KVL in the other direction, we can express the sensing point voltage in terms of the grid voltage and grid impedance, as shown below:

$$\vec{v}_S = \vec{v}_G + Z_G\vec{i}. \tag{3.3}$$

The PLL, which is driven by the sensing voltage, v_S, is trying to lock to the grid voltage, v_G. However, the grid impedance, Z_G, can influence the sensing voltage; this can impact the PLL's ability to synchronize, as we'll soon see. Following a similar analysis proposed by D. Dong et al. in [22], we can modify the basic SRF-PLL control model (shown previously in Figure 1.5) to accommodate the additional voltage of the grid impedance, Z_G, which augments the sensing voltage. This is shown in Figure 3.6 below; note that here we assume the current references, i_{dq}^*, to be approximately equal to the real currents, i_{dq}. This assumption should be safe, since the

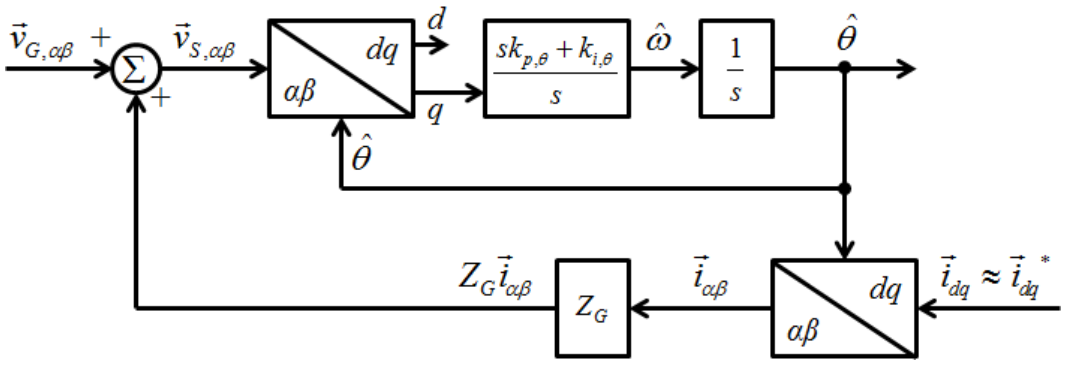


Figure 3.6: Nonlinear SRF-PLL control-system model showing the grid-impedance interaction.

dynamics of the current are usually much faster than the PLL control loop. This also means that it is the magnitude, phase, and frequency of the current (the latter of which is derived from the PLL) which influences the "$Z_G\, i$" impedance-voltage-drop term – this will be important soon.

To better understand how the grid impedance affects the PLL's synchronization, we need to simplify the previous model. Beforehand, we will first define some of the space-vector components; we will represent the grid voltage in a stationary-frame space-vector form as:

$$\vec{v}_{G,\alpha\beta} = V_{Gm}\cos\left(\theta_G\right) + jV_{Gm}\sin\left(\theta_G\right), \tag{3.4}$$

where V_{Gm} is the grid voltage peak-magnitude and θ_G is the true angle of the grid. We'll also represent the grid current in a stationary-frame space-vector form, defined as:

$$\vec{i}_{\alpha\beta} = I_m\cos\left(\hat{\theta} + \phi_i\right) + jI_m\sin\left(\hat{\theta} + \phi_i\right), \tag{3.5}$$

where I_m is the grid current peak-magnitude, $\hat{\theta}$ is the estimated angle from the PLL, and ϕ_i is the relative angle of the current, with respect to the PLL angle. The parameters of the current can also be described as follows:

$$\vec{i}_{\alpha\beta} = i_\alpha + ji_\beta = I_m\angle\left(\hat{\theta} + \phi_i\right), \quad \text{where} \quad \begin{cases} I_m = \sqrt{{i_\alpha}^2 + {i_\beta}^2} = \sqrt{{i_d}^2 + {i_q}^2}, \\ \phi_i = \arctan\left(i_q / i_d\right). \end{cases} \tag{3.6}$$

Lastly, we also need to define the magnitude and phase angle of the impedance:

$$Z_G = R_G + j\hat{\omega}L_G = \left|Z_G\right|\angle\phi_{Z_G}, \quad \text{where} \quad \begin{cases} \left|Z_G\right| = \sqrt{{R_G}^2 + \left(\hat{\omega}L_G\right)^2}, \\ \phi_{Z_G} = \arctan\left(\hat{\omega}L_G / R_G\right). \end{cases} \tag{3.7}$$

Now, if we apply the Clarke/Park transforms to the sensing voltage, v_S, and examine the expression for the q-axis voltage (which is used to obtain the angle-error),

$$v_{Sq} = -v_{S\alpha}\sin\left(\hat{\theta}\right) + v_{S\beta}\cos\left(\hat{\theta}\right), \tag{3.8}$$

and then substitute the sensing voltage with equation (3.3), we will obtain the following:

$$v_{Sq} = -\left(v_{G\alpha} + v_{Z_G\alpha}\right)\sin\left(\hat{\theta}\right) + \left(v_{G\beta} + v_{Z_G\beta}\right)\cos\left(\hat{\theta}\right), \tag{3.9}$$

where the voltage v_{ZG} is the voltage drop on the grid impedance. After some simplification of the Park transform, the resulting q-axis voltage can be expressed as:

$$v_{Sq} = V_{Gm}\sin\left(\theta_G - \hat{\theta}\right) + \left|Z_G\right| I_m \sin\left(\phi_i + \phi_{Z_G}\right). \tag{3.10}$$

Note that the time-varying components of the second terms (involving v_{ZG}) cancel, since both the current and the Park transform rotate with the same angle, $\hat{\theta}$; only the constant phase-shift terms remain. From equation (3.10), we can develop a new quasi-static PLL model, shown below in Figure 3.7, which includes the grid-impedance interaction.

From this model, we can see that there are two feedback loops which influence the PI-controller of the PLL. Following from [22], the top loop, in blue, is the traditional negative feedback loop used to synchronize to the true grid angle, θ_G, which generates a q-axis voltage labeled as v_{Sq-}. Meanwhile, the bottom loop, in red, is a positive feedback loop which aims to desynchronize the PLL from the grid and synchronize to the inverter voltage, which generates a q-axis voltage labeled v_{Sq+}. The influence of the self-synchronization loop is determined by I_m,

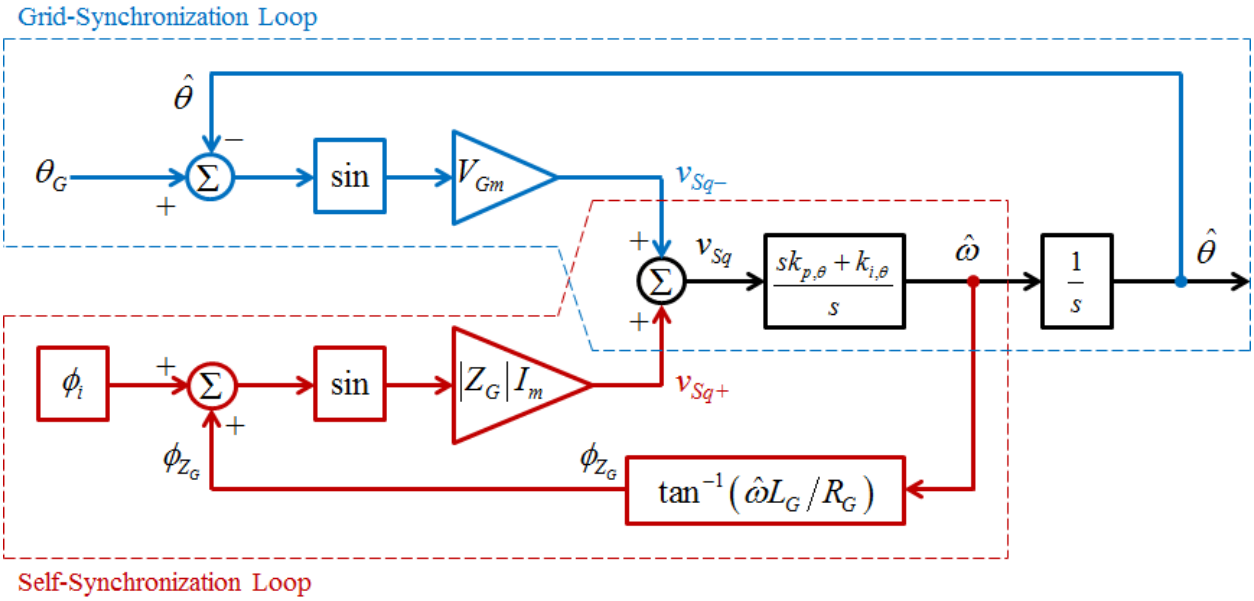


Figure 3.7: Quasi-static SRF-PLL control system including the grid-impedance interaction.

ϕ_i, $|Z_G|$, and ϕ_{ZG}. A higher injection current, larger grid impedance (a weaker grid), and a larger combined current and impedance angle (larger $\phi_i + \phi_{ZG}$) give a stronger self-synchronization feedback effect. The sign of the self-synchronization voltage, v_{Sq+}, is determined by both the power-flow direction (related to ϕ_i) and the grid's reactive components (ϕ_{ZG}). One proposed change to the analysis shown here from [22] is the inclusion of ϕ_i; this allows us to consider the PLL's stability over the full range of *dq*-operating points, providing a more generalized view.

Due to the pure integrator within the PI-controller, the PLL will always try to drive the total *q*-axis voltage to zero, $v_{Sq} = 0$. At any loaded operating point, the self-synchronization loop will generate a constant voltage which the grid-synchronization loop must counteract. This means that a finite angle error, $(\theta_G - \hat{\theta}) = \delta$, must exist to have some non-zero v_{Sq-} to cancel out the disturbance of v_{Sq+}.

With the trigonometric sine function in both loop paths (a residual of the Park transform), the grid-synchronization voltage, v_{Sq-}, is limited by the grid voltage magnitude:

$$-V_{Gm} \le v_{Sq-} \le +V_{Gm}. \tag{3.11}$$

When the disturbance, v_{Sq+}, is larger than the maximum output of v_{Sq-}, no steady-state equilibrium point, $(\theta_G - \hat{\theta})$, can be found to ensure $v_{Sq} = 0$. Therefore, the large-signal stability requirement for the PLL in a weak-grid condition can be expressed as the following:

$$\left|v_{Sq+}\right| < \left|v_{Sq-}\right| \quad \Rightarrow \quad \left|\sin\left(\phi_i + \phi_{Z_G}\right)\right| |Z_G| I_m < V_{Gm}. \tag{3.12}$$

The result of equation (3.12) shows that a larger grid voltage, a lower grid current magnitude, a smaller grid impedance magnitude, and a smaller net sum of grid-current and grid-impedance angles will result in a more stable operation of the PLL.

An existing method, discussed in [19], to enhance the PLL's quasi-static stability in a weak-grid operation is to alter the PLL's sensed voltage, v_S, by subtracting a virtual-impedance voltage

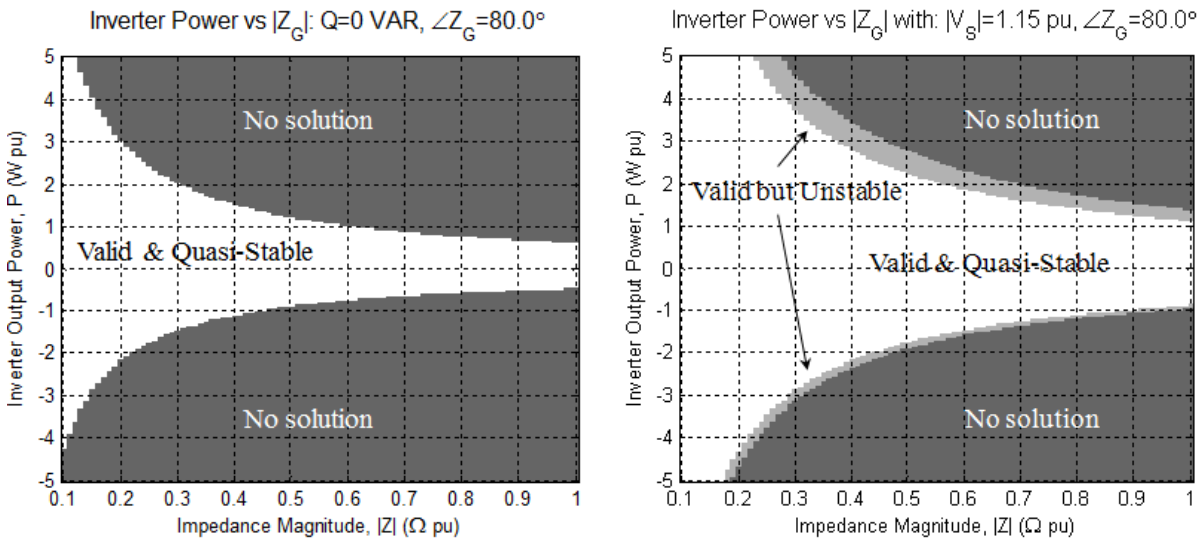
to counteract the grid impedance, Z_G. By virtually shifting sensing-point voltage closer to the true Thévenin-equivalent voltage of the grid, the PLL will behave as if it is connected to a more stiff-grid. This technique was discussed in the previous section, and agrees with the analysis above. However, the proposed alteration to this technique is to consider alternate combinations of virtual impedances to further enhance the stability in certain operating cases.

In the cases of connecting active loads and sources to the grid, such as electric vehicle (EV) battery chargers and photo-voltaic (PV) installations, only the grid-voltage magnitude, V_{Gm}, the grid impedance, Z_G, and the real power flow from the converter may be fixed values. In this simple scenario, the only other control variable may be the reactive power generation, which in turn affects the angle of the current, ϕ_i. The valid range of operating points can be found by simultaneously solving the system of nonlinear equations below, which represent the power balance between the inverter, grid impedance, and the Thévenin grid voltage, given as:

$$\begin{cases} P_I = 3\left|\vec{I}\right|^2 R_G + \Re\left\{\sqrt{3}\vec{V}_G\vec{I}^*\right\}, \\ Q_I = 3\left|\vec{I}\right|^2 X_G + \Im\left\{\sqrt{3}\vec{V}_G\vec{I}^*\right\}, \end{cases} \tag{3.13}$$

where R_G and X_G are the resistance and reactance of the grid, respectively, P_I and Q_I are the real and reactive output power of the inverter, respectively, V_G is the grid voltage, and I is the line current. The grid voltage is usually assumed to have a magnitude of 1 pu and a phase angle of 0°. The interfacing impedance is usually much smaller than the grid impedance in a weak-grid scenario and will be ignored here (Z_I could be considered to be lumped into Z_G).

As a simple example, we can consider the case where the reactive power of the inverter is set to also be 0 VAR. If we also assume the grid impedance is highly inductive (assuming a phase angle of 80°), we can generate the plot shown in Figure 3.8 (a) on the next page. From this plot, we can see the range of possible operating points of the converter. The dark shaded areas



(a) (b)

Figure 3.8: Quasi-stability & power transfer capability for DC/AC converter vs. grid impedance with: (a) no reactive power generation; (b) sensing voltage magnitude regulation via reactive power.

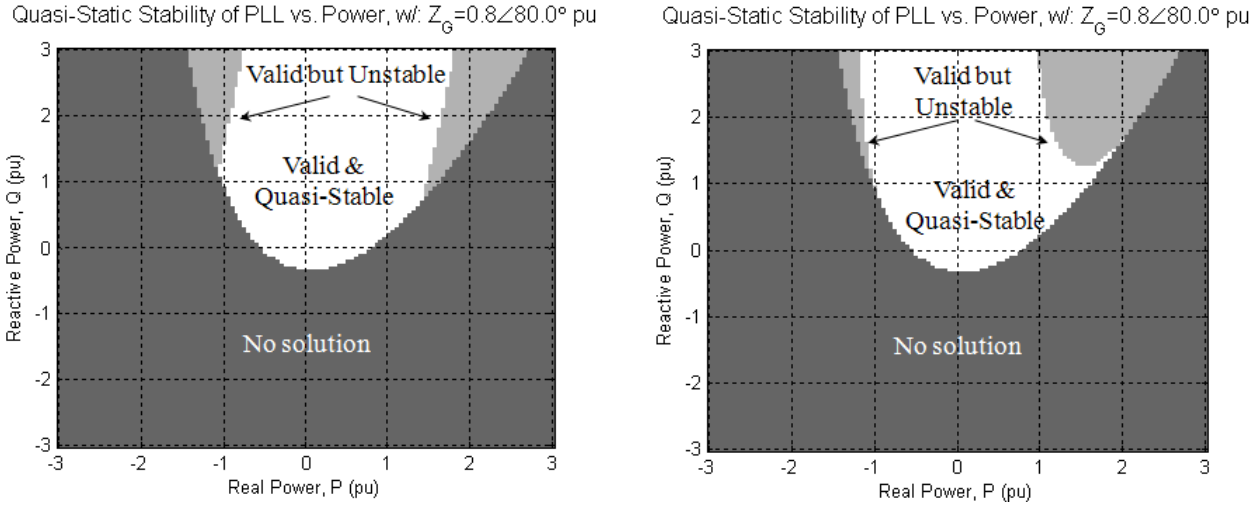
indicate operating conditions which do not have a numerical solution to the power-balance equations. The lightly shaded areas are operating points which fail the quasi-static stability requirements. The white regions represent operating conditions where a valid power solution is found, and the operating point is stable according to equation (3.12). We can see in Figure 3.8 (a) that with a highly inductive grid impedance, the power transfer capability cannot exceed roughly $P_I = 1$ pu with a grid impedance $|Z_G| > 0.6$ pu in the inverter-case. For the rectifier-case, the limit of $P_I = 1$ pu is reached around $|Z_G| > 0.45$ pu.

If we now relax these requirements by allowing the converter to generate reactive power, we can observe that more valid operating points exist, which still satisfy the power balance equations. The new system of equations used ensures the real power is balanced and the voltage magnitude at the sensing side (or inverter side) is controlled to a magnitude of 1.15 pu (avoiding over-modulation under Space-Vector PWM). The system of equations in (3.13) is revised to:

$$\begin{cases} P_I = 3\left|\vec{I}\right|^2 R_G + \Re\left\{\sqrt{3}\vec{V}_G\vec{I}^*\right\}, \\ 1.15 = \left|\vec{V}_G + Z_G\vec{I}\right|. \end{cases} \tag{3.14}$$

The resulting stable points of the relaxed system are shown in Figure 3.8 (b); from here, we can clearly see that the valid operating ranges of the converter increase by allowing reactive power.

To obtain an even more generalized view, we can hold the magnitude of the grid impedance constant and simply sweep the real and reactive power values, P_I and Q_I. We will revert back to equation set (3.13) for defining valid operating conditions. Figure 3.9 below shows the results of sweeping real and reactive power operating with a grid impedance magnitude of $|Z_G|$ = 0.8 pu and a grid impedance angle of $\angle Z_G$ = 80 degrees. The range of possible operating points maps out a sort of "parabolic" shape on the *PQ*-plane. We can see in the figure that, of this range of possible operating points, the quasi-static stability analysis shows that there are some operating points where the PLL will be unable to synchronize to the grid voltage. In Figure 3.9 (a), no impedance compensation is applied. However, in Figure 3.9 (b), a (0.5 + *j*0) pu resistive compensation is applied. The compensation allows us to shift the range of quasi-stable operating



(a) (b)

Figure 3.9: Quasi-stability vs. power transfer capability for weak-grid connection: (a) no compensation, (b) 0.5 pu resistive compensation.

points so that different operating conditions can be achieved.

As mentioned previously, the compensation impedance, Z_{VS}, need not cancel out the grid impedance entirely. By simply shifting the total phase or total magnitude of ($Z_G - Z_{VS}$), we can alter the range of quasi-stable operating points. By applying virtual impedances at the sensing-point, equation (3.12) will effectively change to:

$$\left|v_{Sq+}\right| < \left|v_{Sq-}\right| \quad \Rightarrow \quad \left|\sin\left(\phi_i + \phi_{Z_G - Z_{VS}}\right)\right| \left|Z_G - Z_{VS}\right| I_m < V_{Gm}. \tag{3.15}$$

Thus, a change in the combined phase angle or combined magnitude may be all that is needed to enhance the quasi-stability of a particular operating range. Also note from the equation above, that if we select $Z_{VS} = Z_G$, the PLL will be perfectly compensated to synchronize to the true grid voltage and all of the valid operating points will become quasi-stable.

3.3. IMPEDANCE-COMPENSATED SYSTEM STABILITY ANALYSIS

The quasi-stability analysis in the previous section provides a rough estimate of which operating points may be possible when running a DC/AC inverter system. However, the previous analysis does not take into account any small-signal behavior of the system. Furthermore, much of the control system has been omitted. The resulting quasi-static stability equations also do not rely on any of the PLL's PI coefficients or the *dq*-control's PID coefficients!

To assess the small-signal stability of the control algorithm, a complete dynamic system model can be developed, as was done in Chapters 1 and 2 previously. In this section, we will develop the necessary system of 1st-order differential equations to model an inverter system using and impedance-compensated PLL.

We will again start with the model presented in Chapter 1 and modify it to match the present control system. First, we will incorporate the single-cycle delays into the model. Assuming that a physical interfacing-impedance, Z_I, is still used, we will now have three delays within the control

system: two ADC delays for the sampling of v_S and i, and one PWM delay for outputting v_I. Building off of Chapter 2, we can introduce a new space-vector variable, $\vec{v}_{Sxy,LP}$, to represent the low-pass filtered sensing-point voltages. Applying the delay in the *xy*-frame gives the following:

$$\vec{v}_{Sxy,LP} = \frac{1}{1+(s+j\omega_G)T_{sw}}\vec{v}_{Sxy} \qquad \Rightarrow \qquad \frac{d\vec{v}_{Sxy,LP}}{dt} = \frac{1}{T_{sw}}\left[\vec{v}_{Sxy} - \vec{v}_{Sxy,LP} - j\omega_G T_{sw}\vec{v}_{Sxy,LP}\right]. \quad (3.16)$$

Splitting the above equation into real and imaginary parts gives us

$$\begin{cases} \dfrac{dv_{Sx,LP}}{dt} = \dfrac{1}{T_{sw}}\left[v_{Sx} - v_{Sx,LP} + \omega_G T_{sw} v_{Sy,LP}\right], \\ \dfrac{dv_{Sy,LP}}{dt} = \dfrac{1}{T_{sw}}\left[v_{Sy} - v_{Sy,LP} - \omega_G T_{sw} v_{Sx,LP}\right]. \end{cases} \quad (3.17)$$

The remainder of the dynamics will actually be very similar to the model developed in Chapter 2, with a few exceptions. The use of a physical interface-inductor, Z_I, means that the current control loops can revert back to using a simpler PI-controller. This will eliminate two state variables from the system. The overall system dynamics will still be 14$^{\text{th}}$-order however, due to the addition of the two differential equations above. Furthermore, we will need to redefine the inverter output voltage and the PLL error-input signal to be in terms of the augmented sensing-point voltage, v_S', rather than v_S. Also, we've introduced the space-vector quantities, $\vec{v}_{Idq,LP}$, to model the PWM delays. A summary of the modeling equations is provided below, starting with the coordinate-frame transformation equations:

$$\begin{cases} (1) \quad i_d = +i_{x,LP}\cos(\delta) + i_{y,LP}\sin(\delta), \\ (2) \quad i_q = -i_{x,LP}\sin(\delta) + i_{y,LP}\cos(\delta), \\ (3) \quad v_{Sd} = +v_{Sx,LP}\cos(\delta) + v_{Sy,LP}\sin(\delta), \\ (4) \quad v_{Sq} = -v_{Sx,LP}\sin(\delta) + v_{Sy,LP}\cos(\delta), \\ (5) \quad v_{Ix} = +v_{Id,LP}\cos(\delta) - v_{Iq,LP}\sin(\delta), \\ (6) \quad v_{Iy} = +v_{Id,LP}\sin(\delta) + v_{Iq,LP}\cos(\delta). \end{cases} \quad (3.18)$$

On the following page, the set of differential equations used to describe the system dynamics is provided. The two equations above in (3.17) are listed as equations (9) and (10) in the set:

$$
\begin{cases}
(1) & \dfrac{di_x}{dt} = \dfrac{1}{L_G}\left[-R_G i_x + \omega_G L_G i_y + v_{Sx} - v_{Gx}\right], & \text{Circuit Dynamics} \\
(2) & \dfrac{di_y}{dt} = \dfrac{1}{L_G}\left[-R_G i_y - \omega_G L_G i_x + v_{Sy} - v_{Gy}\right], & \text{Circuit Dynamics} \\
(3) & \dfrac{di_{x,LP}}{dt} = \dfrac{1}{T_{sw}}\left[i_x - i_{x,LP} + \omega_G T_{sw} i_{y,LP}\right], & \text{ADC } \vec{i} \text{ Delay} \\
(4) & \dfrac{di_{y,LP}}{dt} = \dfrac{1}{T_{sw}}\left[i_y - i_{y,LP} - \omega_G T_{sw} i_{x,LP}\right], & \text{ADC } \vec{i} \text{ Delay} \\
(5) & \dfrac{dQ_d}{dt} = \varepsilon_d = i_d^{\,*} - i_d, & dq\text{-PI Integrator} \\
(6) & \dfrac{dQ_q}{dt} = \varepsilon_q = i_q^{\,*} - i_q, & dq\text{-PI Integrator} \\
(7) & \dfrac{di_{d,LP}}{dt} = \omega_{HP} i_d - \omega_{HP} i_{d,LP} + \hat{\omega}\, i_{q,LP}, & \text{Virtual Inductor} \\
(8) & \dfrac{di_{q,LP}}{dt} = \omega_{HP} i_q - \omega_{HP} i_{q,LP} - \hat{\omega}\, i_{d,LP}, & \text{Virtual Inductor} \\
(9) & \dfrac{dv_{Sx,LP}}{dt} = \dfrac{1}{T_{sw}}\left[v_{Sx} - v_{Sx,LP} + \omega_G T_{sw} v_{Sy,LP}\right], & \text{ADC } \vec{v}_S \text{ Delay} \\
(10) & \dfrac{dv_{Sy,LP}}{dt} = \dfrac{1}{T_{sw}}\left[v_{Sx} - v_{Sx,LP} - \omega_G T_{sw} v_{Sy,LP}\right], & \text{ADC } \vec{v}_S \text{ Delay} \\
(11) & \dfrac{dv_{Id,LP}}{dt} = \dfrac{1}{T_{sw}}\left[v_{Id} - v_{Id,LP} + \hat{\omega} T_{sw} v_{Iq,LP}\right], & \text{PWM Delay} \\
(12) & \dfrac{dv_{Iq,LP}}{dt} = \dfrac{1}{T_{sw}}\left[v_{Iq} - v_{Iq,LP} - \hat{\omega} T_{sw} v_{Id,LP}\right], & \text{PWM Delay} \\
(13) & \dfrac{d\delta}{dt} = k_{p,\theta}\varepsilon_\theta + k_{i,\theta}\gamma - \omega_G, & \text{PLL Angle Difference} \\
(14) & \dfrac{d\gamma}{dt} = \varepsilon_\theta = \dfrac{v'_{Sq}}{\sqrt{{v'_{Sd}}^2 + {v'_{Sq}}^2}}. & \text{PLL-PI Integrator}
\end{cases}
\tag{3.19}
$$

Note that equation (14) in the set has been modified to use the augmented sensing-point voltage, v_S'. To be consistent with the previous chapters, equation (14) in the set is still normalized by dividing out the magnitude of the sensing-point voltage. However, in the analysis presented in

the previous section, following D. Dong et al. in [22], simply uses the *q*-axis voltage as the angle-error signal. This will introduce a gain in the PI-controller coefficients of the PLL proportional to the magnitude of the sensed voltage. To cancel out this gain and make the analysis follow that of the previous section, equation (14) of the previous set could be revised to the following:

$$\frac{d\gamma}{dt} = \varepsilon_\theta = \frac{v'_{Sq}}{k_v}, \tag{3.20}$$

where k_v is simply a constant equal to the nominal magnitude of the grid voltage (~170 V). Since this redefines the angle-error signal, equation (13) of the set will also be affected.

In addition to the set of differential equations, we also can describe the remaining algebraic equations which relate the other various quantities in the control system; these are listed below. As mentioned above, the augmented sensing-point voltage also has an effect on v_I, but this is shown in the calculation of the inverter voltages, which are equations (7) and (8) in the set:

$$\begin{cases}
(7) & v_{Id} = \varepsilon_d k_{p,c} + Q_d k_{i,c} - \hat{\omega}\left(L_I + L_{VS}\right) i_q + v'_{Sd}, \\
(8) & v_{Iq} = \varepsilon_q k_{p,c} + Q_d k_{i,c} + \hat{\omega}\left(L_I + L_{VS}\right) i_d + v'_{Sq}, \\
(9) & v_{Sx} = \dfrac{1}{L_I + L_G}\left[L_G v_{Ix} + \left(L_I R_G - L_G R_I\right) i_x + L_I v_{Gx} \right], \\
(10) & v_{Sy} = \dfrac{1}{L_I + L_G}\left[L_G v_{Iy} + \left(L_I R_G - L_G R_I\right) i_y + L_I v_{Gy} \right], \\
(11) & v_{Z_{VS}d} = R_{VS} i_d + \omega_{HP} L_{VS}\left(i_d - i_{d,LP}\right), \\
(12) & v_{Z_{VS}q} = R_{VS} i_q + \omega_{HP} L_{VS}\left(i_q - i_{q,LP}\right), \\
(13) & v'_{Sd} = v_{Sd} - v_{Z_{VS}d}, \\
(14) & v'_{Sq} = v_{Sq} - v_{Z_{VS}q}, \\
(15) & \hat{\omega} = k_{p,\theta}\varepsilon_\theta + k_{i,\theta}\gamma.
\end{cases} \tag{3.21}$$

Note that, in the *dq*-frame inverter voltage calculations, the sum of virtual sensing-point impedance and interface impedance should be used for decoupling the currents. This is due to the fact that, by impedance-compensating the PLL, the impedance voltage-drops that were removed

(subtracted) are effectively moved to the "other side" of the virtual sensing point. Thus, by decreasing the effective Z_G, we increase the total effective Z_I. This is summarized in the following equations. The use of virtual interface-impedance does not affect the sensing-point-to-grid impedance, but altering the sensing point using a virtual-sensing-impedance does affect the interface-impedance. We can define the effective interface impedance, $Z_{I,eff}$, and effective sensing-point-to-grid impedance, $Z_{G,eff}$, as follows:

$$\begin{cases} Z_{I,eff} = Z_I + Z_{VI} + Z_{VS}, \\ Z_{G,eff} = Z_G - Z_{VS}. \end{cases} \tag{3.22}$$

Another item worth noting is the reoccurrence of the voltage-divider equations for calculating the *xy*-frame sensing-point voltages, equations (9) and (10) in the set (3.21). These return since we have a physical interfacing impedance once again.

To help the reader grasp the different delays and space-vector relationships in this control system model, a MATLAB vector plot has been produced for a system running with some

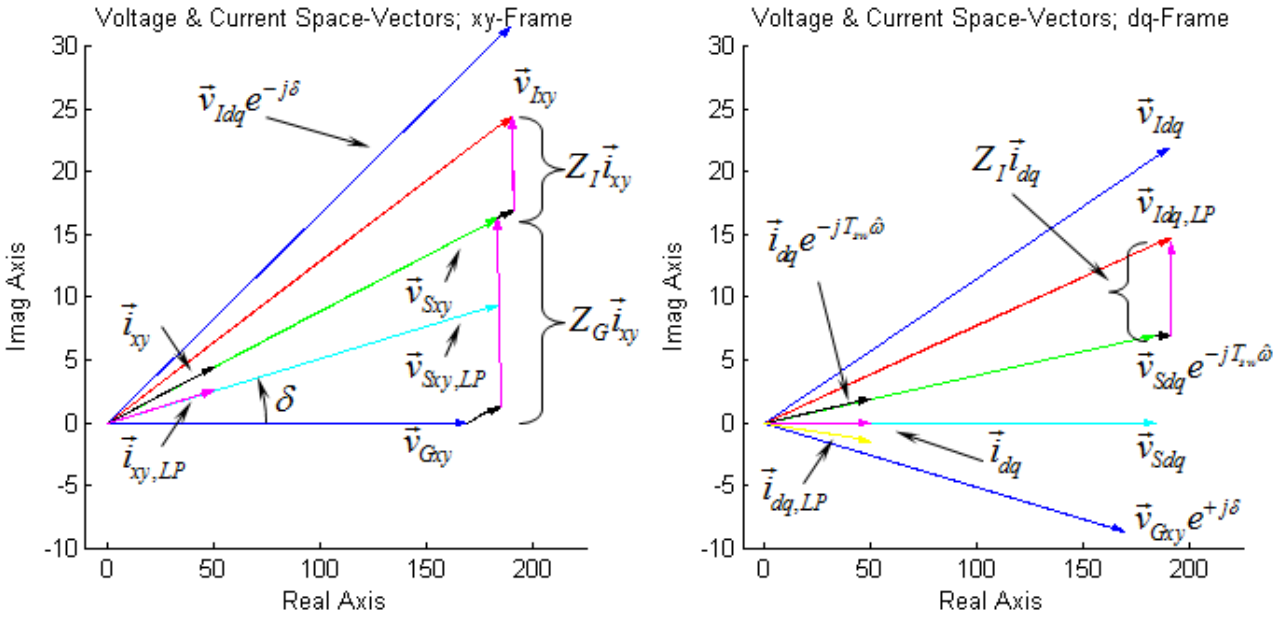


Figure 3.10: Space-vector plot of voltages & currents in the IC-PLL algorithm in: (a) the *xy*-frame, (b) the *dq*-frame.

positive *d*-axis current reference and a zero *q*-axis reference (i.e. $i_d^* > 0$ and $i_q^* = 0$). This is shown in Figure 3.10 below. The impedance voltage-drops are also shown, as well as some of the low-pass filter variables, and some other vectors which have not been previously defined.

3.4. ENHANCEMENT OF STABILITY USING IMPEDANCE-COMPENSATION

With the dynamic system model developed in the previous section, a computer-aided stability analysis was performed on the impedance-compensated PLL. To relate the results back to Section 3.2 (and the work in [22]), the PLL error signal, ε_θ, was defined to be simply v_{Sq}' / k_v, as described previously in equation (3.20). The value of the constant was set to be $k_v = 17$ (providing a gain of ~10x to the PI-controller coefficients). The Jacobian matrix was formed from the equation set (3.19) and was computed symbolically using MATLAB. The Jacobian was then evaluated at many operating points, corresponding to the P_I and Q_I values presented previously in Figure 3.9. At each operating point, the eigenvalues of the Jacobian were found numerically and the eigenvalue with the largest real-part was checked for stability (to see if it was greater or less than zero).

The plot in Figure 3.9 was regenerated with the small-signal stability information being added to the plot, based on the localized stability analysis. The results corresponding to the case

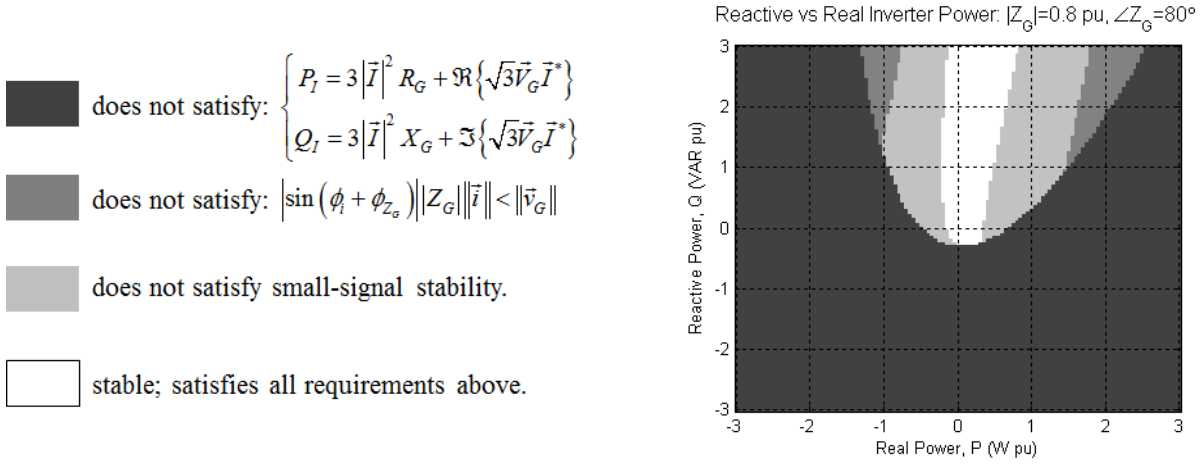


Figure 3.11: Small-signal stability vs. power transfer capability without impedance compensation.

in Figure 3.9 (a), without any impedance compensation, are presented here in Figure 3.11. From the results in Figure 3.11, we can see that the quasi-static stability developed within Section 3.2 really only provides an "upper bounds" to the actual stable operating capability of the inverter. When taking into consideration the dynamics of the current controllers, the PLL, and the time-delays introduced in the digital system implementation, the range of stable operating points is actually only a small subset of the range predicted by the quasi-static stability analysis.

To see the benefits of impedance-compensation, we can re-run the computational model used to generate Figure 3.11, this time with some amount of compensation applied. This is shown below in Figure 3.12. Here, the value of Z_{VS} is set to ¼ of the grid impedance, Z_G. We can see that, not only does the quasi-static stability range increase (as expected), but the predicted small-signal stability range has also been enlarged! Thus, we have shown that the use of impedance-compensation can improve the stability of a converter, by allowing the converter to synchronize to a more-stable point within the electrical system. Note that other parameters will also obviously affect the stability. The tunings of the PI-controllers and the selection of the high-pass filter frequencies for the derivative- approximations will also affect the range of small-signal stability.

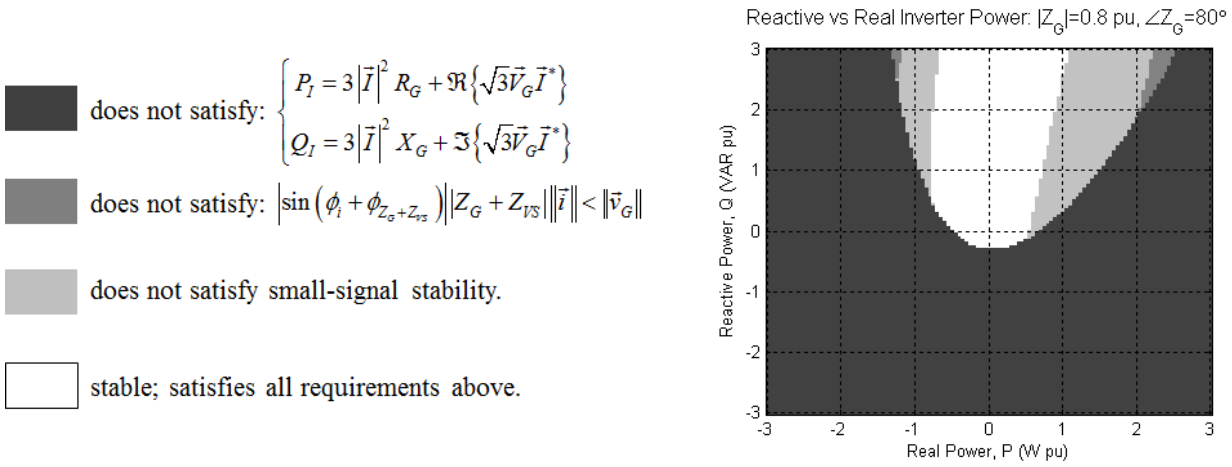


Figure 3.12: Small-signal stability vs. power transfer capability with 25% impedance compensation.

3.5. SENSOR-LESS PMAC MOTOR TERMINAL-VOLTAGE SYNCHRONIZATION

One additional application of virtual-impedances within DC/AC converters is in the control of three-phase AC machines. We have shown previously in Chapter 2 that self-synchronization of a converter to its own terminal voltage (while still regulating current) is possible. However, because the self-synchronizing algorithm aligns the PLL to the *d*-axis of the terminal voltage, some phase-angle error will exist between the terminal voltage and the Thévenin-equivalent voltage of the AC source. This has been discussed in the previous sections of this chapter. However, if we combine this self-synchronizing algorithm together with the impedance-compensation method to shift the PLL angle to the true Thévenin-equivalent-voltage angle, we can control the current from the reference-frame of the Thévenin AC source, using only the terminal voltage and current sensor information! This is the main objective in many back-EMF-based sensor-less motor control algorithms for permanent-magnet AC machines [23].

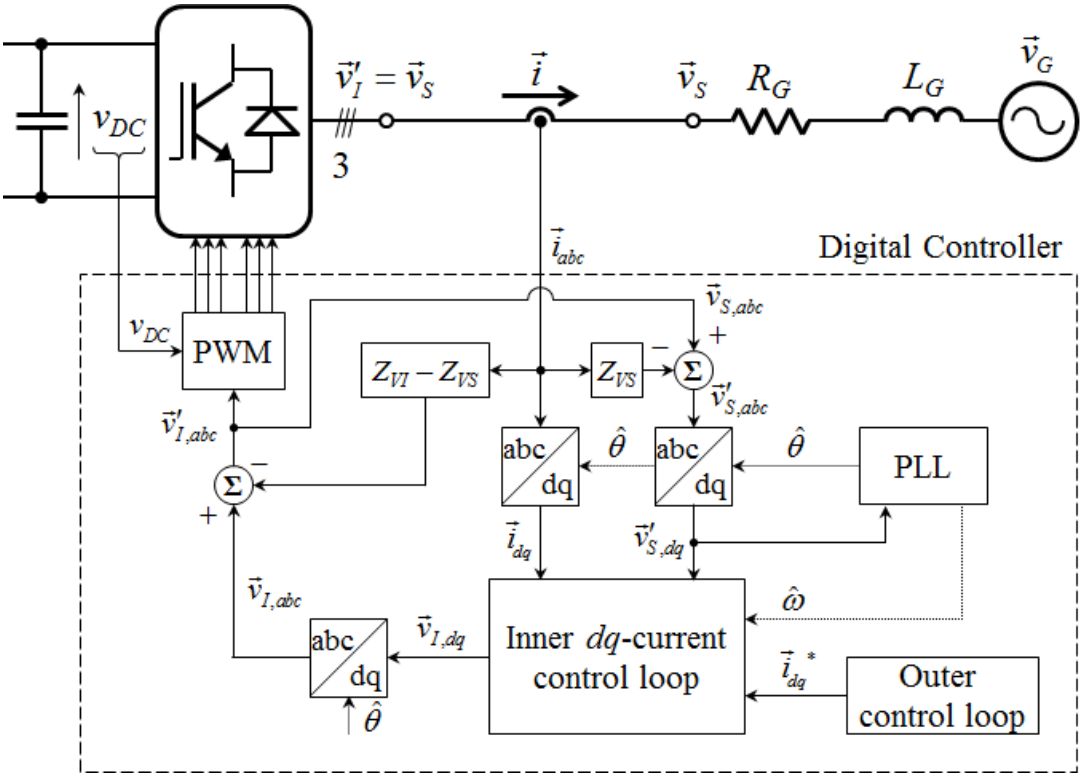


Figure 3.13: Proposed sensor-less PMAC control block diagram utilizing virtual impedance compensations.

A block diagram of the proposed control algorithm is shown in Figure 3.13. To be consistent, we will use the same variable-names as before. In this case, the grid voltage, v_G, represents the back-EMF of the machine due to the time-varying flux linkages. The impedances R_G and L_G represent the motor winding resistance and inductance, respectively. To completely cancel out the motor impedance and synchronize the PLL at the back-EMF voltage point, a virtual impedance, Z_{VS}, can be subtracted from the sensing voltage. Note that the unadjusted sensing voltage, v_S, in this case is the same as the inverter's output voltage, v_I', as in Chapter 2:

$$\vec{v}'_{S,abc} = \vec{v}_{S,abc} - Z_{VS}\vec{i}_{abc} = \vec{v}_{S,abc} - \left(R_{VS} + \frac{s\omega_{HP}}{s+\omega_{HP}} L_{VS} \right) \vec{i}_{abc}, \tag{3.23}$$

where we could select $R_{VS} = R_G$ and $L_{VS} = L_G$. To implement this, the winding impedance must be known. The winding impedances do vary with temperature and loading conditions, but are generally more predictable than the parasitic impedance of the grid. Even with some small errors in parameter values, the application of equation (3.23) should place the PLL's angle close to the true angle of the back-EMF.

Subtracting a virtual-impedance's voltage from the PLL does not eliminate the impedance from the virtual system; it simply gets shifted to the other side of the measurement point. In other words, the act of impedance-compensating the PLL effectively puts the sensing-virtual-impedance, Z_{VS}, between the perceived sensing point and the inverter output, making it appear like an interfacing-impedance. Therefore, the impedance-compensation of the PLL is really all that is required to apply an interfacing-impedance to the system for the inner *dq*-current loop to respond to. This alteration alone will work for sensor-less control of a PM machine.

However, if a different interfacing-impedance, other than Z_{VS}, is desired, we can further alter the control system to adjust the computed inverter output voltage, v_I, as previously shown in Figure 3.13. For example, if we wish to use only an interfacing resistor between the voltage

sensing point and inverter output, we can augment the inverter output voltage computed by the *dq*-current controllers, v_I, by subtracting the desired interfacing-impedance voltages and adding the undesired interfacing-impedance voltages. This can be used to effectively cancel out the Z_{VS} impedance from the interfacing location and replace it with another impedance, Z_{VI}.

As an example, the cancellation of the motor winding impedance, Z_G, and addition of a virtual interfacing resistance, R_{VI}, can be performed. With this approach, the same simplifications to the *dq*-current control algorithm can be made, reducing some of the computational complexity of the control loop. This example is shown mathematically below:

$$\vec{v}'_{I,abc} = \vec{v}_{I,abc} - \left(Z_{VI} - Z_{VS}\right)\vec{i}_{abc} = \vec{v}_{I,abc} - \left(R_{VI} - R_{VS} - \frac{s\omega_{HP}}{s+\omega_{HP}}L_{VS}\right)\vec{i}_{abc}, \tag{3.24}$$

where again, we could set $R_{VS} = R_G$ and $L_{VS} = L_G$. With this second modification to the control system, the effects of the Z_{VS} impedance (and the Z_G impedance) will be completely removed.

To help visualize the changes, we can follow the progression of applying equations (3.23) and (3.24) as illustrated in Figure 3.14. In Figure 3.14 (a), we have the original system. The sensing voltage is located at the inverter terminals. In Figure 3.14 (b), the sensing voltage is moved to the Thévenin-equivalent voltage of the machine by impedance-compensating the PLL. Note that this places the winding impedance, Z_G, between the inverter and measurement point, making it behave as if it were an interfacing-impedance. Lastly, in Figure 3.14 (c), the winding impedance is cancelled out from the inverter voltage and a virtual interfacing resistance, R_{VI}, is added, effectively placing it between the inverter and voltage sensing point. Note that if the last step is omitted, the traditional *dq*-current control algorithm with decoupling terms can be used.

In the case of parameter variation or estimation error, the virtualized sensing-point voltage will not be equal to the true back-EMF voltage; some steady-state angle error will remain. However, this can be observed during a change in the operating point. If the motor impedance is

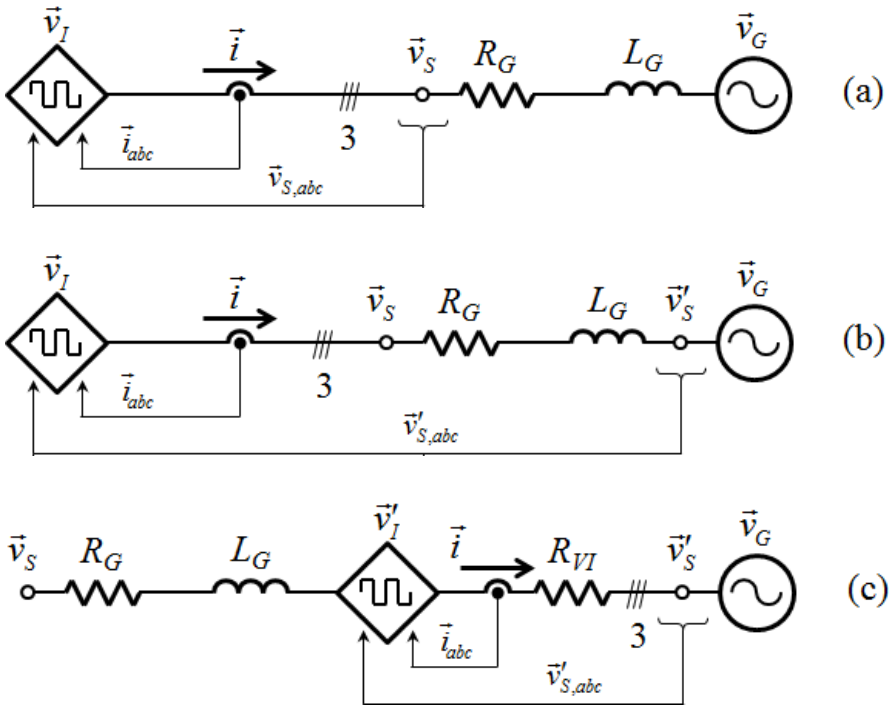


Figure 3.14: Mathematical progression of proposed sensor-less PMAC control, showing: (a) the original control system; (b) the impedance-compensated system for observing the back-EMF; (c) the resulting control system with virtual interfacing resistance.

not properly canceled out, any change in the operating-point will cause a slight change in the PLL's error signal, causing the PLL to realign itself. This is the same phenomenon described in Section 3.1 previously. These PLL error adjustments can be used to identify/update the estimated motor parameters to perfectly align the sensing point with the back-EMF of the machine.

To generate this PLL re-alignment signal, we can inject a time-varying current reference into the *d*-axis. (To follow convention, note that the PLL should be modified to eliminate *d*-axis voltage and align to the *q*-axis.) This injection current does not need to be high-frequency; it simply needs to be somewhat faster than the response time of the PLL (otherwise, the PLL will realign and the error signal will vanish). Alternatively, the PLL could be temporarily detuned during the injection phase and then retuned for normal operation. If either the resistance estimate, $\hat{R}_G$, is incorrect or if the inductance estimate, $\hat{L}_G$, is incorrect, the low-frequency injection current will cause a non-zero low-frequency voltage to appear in one or both of the machine axes. The

polarities of these voltages will change if the impedances have been over-compensated. From the observed voltages, the machine parameter estimates can be adjusted until the PLL is perfectly compensated and the angle-error is reduces to zero. The details of this "low-frequency injection" parameter estimation algorithm will be left as a future work.

3.6. CHAPTER 3 SUMMARY

In this chapter, we have outlined the theory of impedance-compensation for phase-locked loops within DC/AC converter systems. Using virtual-impedance-compensation, the effective measurement point of the PLL can be moved to another location in the circuit. This can alter the dynamics and steady-state synchronization of the PLL.

A proposed stationary-frame compensation method was introduced and shown to have better transient performance than the traditional rotating-frame compensation. This is due to using a fully implemented inductor voltage-drop, rather than only a partial implementation. Furthermore, an analysis on the quasi-static stability of the PLL, considering the grid impedance, was also given. The proposed analysis was slightly modified from an existing analysis to incorporate the phase angle of the line current. This was examined together with the power transfer capabilities of a DC/AC converter system connected in a weak-grid scenario.

Next, a full dynamic-system model was developed for the impedance-compensated converter system. The circuit dynamics, current-controller dynamics, PLL dynamics, and digital sampling delays were incorporated into the model. From the dynamic system model, the localized linear stability of the system could be assessed. Using MATLAB, the local stability at several different operating points were be obtained and plotted against the quasi-static stability ranges. From this model, it was shown that impedance-compensation of the PLL system can enhance the small-signal stability range of the overall converter system.

Lastly, an alternate method for sensor-less control of permanent magnet AC machines was presented, using the concept of virtual-impedance-compensation together with a virtual interfacing-impedance. This was introduced as a possible alternative to the typical Back-EMF observer method for position-sensor-less control of PMAC motors. The virtual-impedance-based method could be potentially much simpler mathematically than the observer-based system.

In the last chapter, the conclusions of this report are summarized. Afterwards, the author's proposed future research works are outlined.

4. CONCLUSIONS AND FUTURE WORK

In this chapter, the main contributions of this work are summarized, followed by the author's recommendations to extend this work. A method to truly test the inductor-less self-synchronization method with a direct-connection to the grid is first discussed. Afterwards, a means of testing the combined impedance-compensated PLL algorithm and virtual-resistance-based inductor-less self-synchronizing algorithm is examined and proposed as a future work.

4.1. CONCLUSIONS

In this report, some common and some novel approaches for the synchronization and current-regulation algorithms used within DC/AC power-electronics converters have been discussed. While the primary focus of this work was on grid-connected systems, the extension to electric motor drive systems is also briefly mentioned. In the following paragraphs, a summary of the work and contributions provided in each chapter are outlined.

Chapter 1 provided some background on DC/AC converters and discussed the typical synchronization and current-control strategy used in three-phase DC/AC converters. This overview was followed with the author's research scope and contributions.

In Chapter 2, the concept of virtual impedance was first reviewed. Afterwards, a novel control strategy was proposed to alter the converter's output voltage in such a way that physical interfacing inductors can be replaced by virtualized ones. In addition, it was shown that this control algorithm could be modified to allow the physical AC voltage sensors to be eliminated. Finally, some novel alterations to the previous control algorithm were proposed to utilize a virtual resistance as the interfacing impedance. With the use of a virtual interface-resistance, many parts of the *dq*-current control algorithm could be simplified, reducing the computational

complexity of the overall control system. The dynamics of the proposed control system were confirmed using simulated and experimental results.

In Chapter 3, the concept of altering the synchronization voltage using virtual impedances, referred to as "impedance-compensation", was first discussed. Next, a proposed method of impedance-compensating an SRF-PLL in the stationary-frame was shown to have better transient performance as compared to an existing method of partial impedance-compensation in the synchronous-frame. Afterwards, the large-signal stability of SRF-PLL-controlled inverter systems used in weak-grid applications was reviewed. The analysis was further generalized by the proposed inclusion of the grid current angle. Afterwards, a thorough linearized stability analysis was then conducted to show how impedance compensation can be used to improve the small-signal stability of the PLL and control system. Finally, the concept of using a virtual interfacing-impedance and an impedance-compensated PLL were proposed as a novel control strategy for the back-EMF-based position-sensor-less control of permanent-magnet AC machines at high speeds.

4.2. FUTURE WORK

As a continuation of the work presented here, the author recommends the following tasks as a means to further validate and extend these research contributions:

1. Construct a cascaded multi-level inverter (CMI) system to reduce the PWM voltage harmonics generated by the inverter system. With very low harmonics, a direct inverter-to-grid connection could be made without any interfacing inductance, and the self-synchronization algorithm could be further tested and verified.
2. Experimentally test the combined virtual-impedance + impedance-compensated PLL algorithm, proposed in Chapter 3, for the sensor-less control of a PM machine at high

speed. This experiment will provide some confirmation on the ability to cancel out a motor's winding impedance and obtain the correct phase angle of the back-EMF. If proven successful, this could be used as a simpler alternative to the commonly-used back-EMF observer method, which is mathematically more intensive.

Regarding the development of a cascaded multi-level inverter system (task #1), the CMI topology, which was briefly mentioned in Chapter 1, could be used. To operate on a typical 208 V three-phase AC grid, a four-stage nine-level CMI, as shown in Figure 1.11, could be constructed. Each module could be powered by isolated 50 V, 1 kW bi-directional DC-DC converters. In a Y-connected configuration, the same line-to-neutral voltage magnitude, ±200 V, could be generated as compared to a traditional 400 VDC two-level inverter. However, the reduced size of the voltage-steps and the use of unipolar PWM in the CMI (as opposed to bipolar PWM of the two-level inverter) will reduce the voltage ripple by a factor of 8x! Together with the use of a high switching-frequency (e.g. 50 kHz), an interfacing inductance of 200 / 8 = 25 µH would be sufficient to maintain a reasonable amount of ripple current. This estimate is based on a previously designed two-level 400 V, 10 kW, 50 kHz inverter using 200 µH inductors [24].

For the experimental testing of the sensor-less motor control algorithm (task #2), an existing 48 V PMAC motor/dynamometer test bench, located at Kettering University, could be used. A photo of the test bench is provided in Figure 4.1. The motors used in the setup are both Motenergy ME1117 4-kW three-phase 8-pole axial-flux PM machines. They are equipped with analog sin/cos position sensors; the sensors can be used to compare the estimated angle in the control algorithm with the true angle of the back-EMF waveform. The dynamometer is equipped with a Transcell TSA50 50-lb load cell to measure the torque difference between the two motors (one motor is gimbal-mounted) and a rotating mass is mounted to the intermediate shaft to

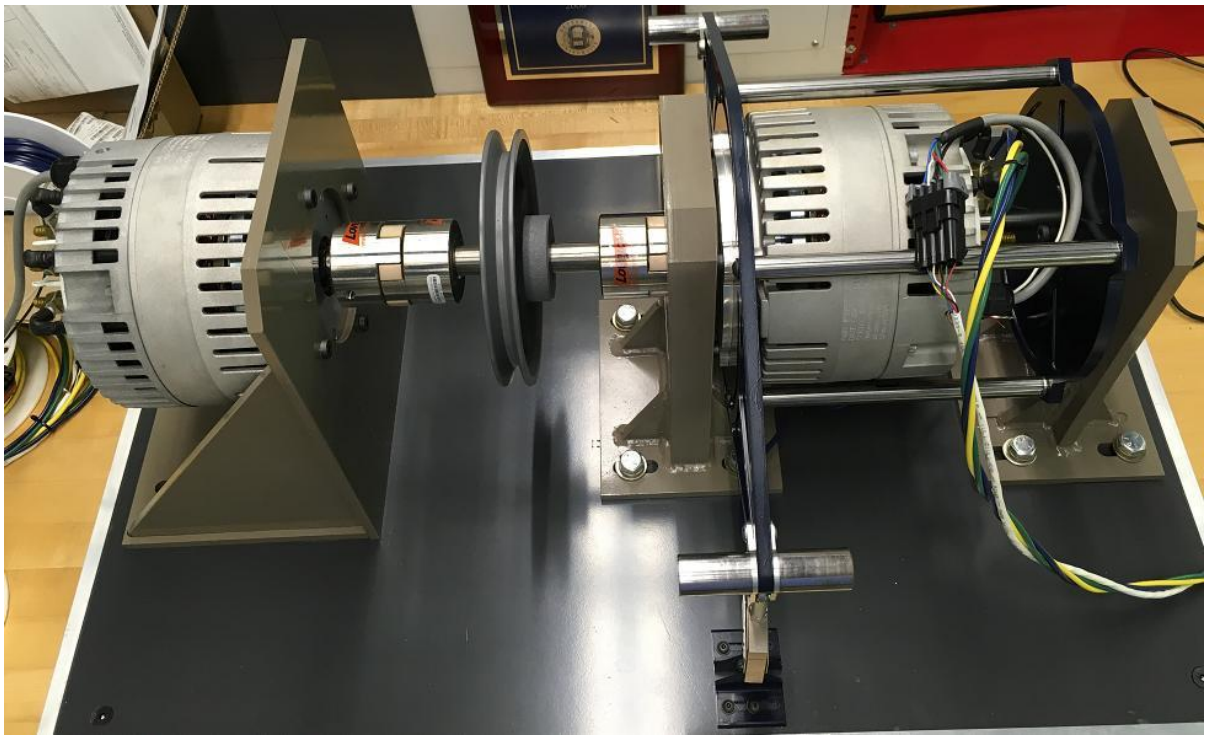


Figure 4.1: Low-voltage PMAC dynamometer test bench at Kettering University.

provide some mechanical inertia. To drive the motors, some existing 48 V inverter hardware at Kettering University can be utilized. These inverters also utilize the same F28069 TI DSP used in the experimental setup that was developed and used in Chapter 2, so much of the software written in other tasks can be reused.

APPENDICES

APPENDIX A: MATLAB SIMULINK SIMULATION MODELS

In Chapter 2, a MATLAB Simulink model of a virtual-resistance control algorithm was presented and used to compare its results with experimental data. In this appendix, additional details and information will be given regarding this simulation model.

The simulation uses the Simscape library package of MATLAB Simulink to model the DC voltage source, two-level MOSFET inverter, grid impedances, and grid AC voltage sources. The top-level diagram showing the Simscape blocks in the simulation is shown below in Figure A.1. The top-level components are simulated at the fastest sampling rate (smallest sampling time), usually 1 or ½ µs (which is much smaller than the typical calculation speed of the inverter's DSP control loop). A discrete-time solver is used for integration within the components (for inductor currents in this case).

The three AC grid voltages are generated using controlled Simscape voltage sources. This is shown in Figure A.2 on the next page. The frequency is assumed to be a given constant (from the step-function source block). The phase-angles are computed by integration of the frequency and an offset adjustment. From the angles, a trigonometric block is used to compute the shape of the signals and they are then scaled to the appropriate magnitude. Constructing the voltages this way

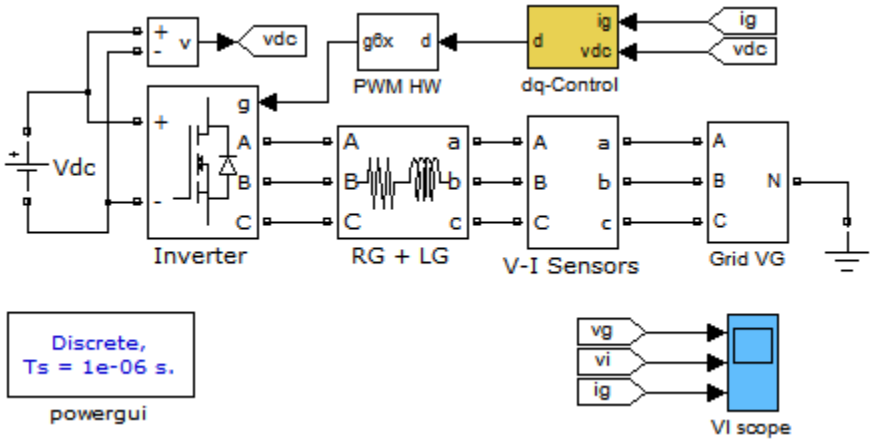


Figure A.1: Top-level diagram of the MATLAB Simulink virtual-resistance simulation.

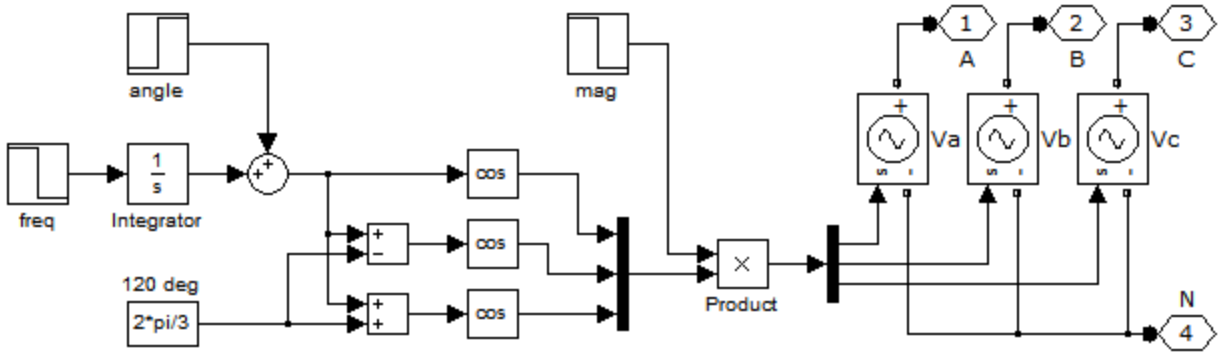


Figure A.2: MATLAB Simulink model for generating three-phase AC grid voltages.

allows for easy manipulation of the frequency, phase, or magnitude of the grid voltage while the simulation is running. With the use of step-function source blocks, each of these parameters can be adjusted at different simulation time instants. This is how the transient responses shown in Section 2.6 were generated.

To generate the PWM signals, the internals of a typical microprocessor PWM module were emulated. This is shown in Figure A.3 below. For the switching state of each phase leg, a logical "greater-than" comparison is performed between the duty-cycle values calculated by the controller and a high-frequency (e.g. 10 kHz) triangle signal which is shifted to have values from 0 to 1. From the switching state of each phase, logical inversion blocks are used to generate the equivalent low-side gating signals. The three switching signals are demuxed, inverted, and then muxed back together to yield the six different gate switching signals. From the ideal gating signals, a rising-edge deadband delay is generated by comparing each switching signal with its

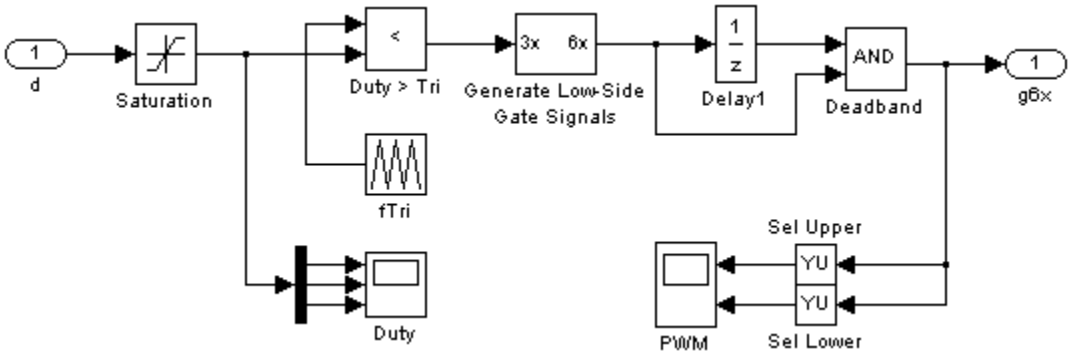


Figure A.3: MATLAB Simulink model of PWM hardware with deadband insertion.

previous value. Only when both are true is the actual gating signal set high. The deadband delay time will depend on the "1/*z*" block's time delay, which is set by the simulation time. This is also why the simulation time is usually set to around 1 µs, as it makes simulating deadband easy.

As mentioned in Chapter 2, only the grid currents and the DC-bus voltage are supplied to the *dq*-control algorithm in the sensor-less virtual-impedance control method. The *dq*-controller ultimately outputs three duty-cycle values for the three bridges of the MOSFET inverter. The top-level diagram of the *dq*-control algorithm is shown in Figure A.4. Here, rate-transition blocks are used to change the sampling time from the very-fast simulation time-step (around 1 µs) to the typical algorithm time-step (around 100 µs) used in the digital signal processor (DSP). The virtual-resistance calculation can be seen at the top of the diagram, in parallel to the *dq*-control block. Meanwhile, the PLL block is shown at the bottom, being fed by the previously calculated phase voltages. To compute the duty cycles, the basic Sine-PWM (SPWM) method is used. The outputs from the *dq*-control and virtual-resistance blocks are already in the stationary *abc*-frame. No third-harmonics have been added to these signals (they are computed using the inverse Park/Clarke transformations). To obtain the duty ratio, the algorithm simply divides by V_{DC} and then offsets by 50% to center the common-mode voltage of the PWM output.

Within the *dq*-control block, we have the basic PID controllers which regulate the currents in

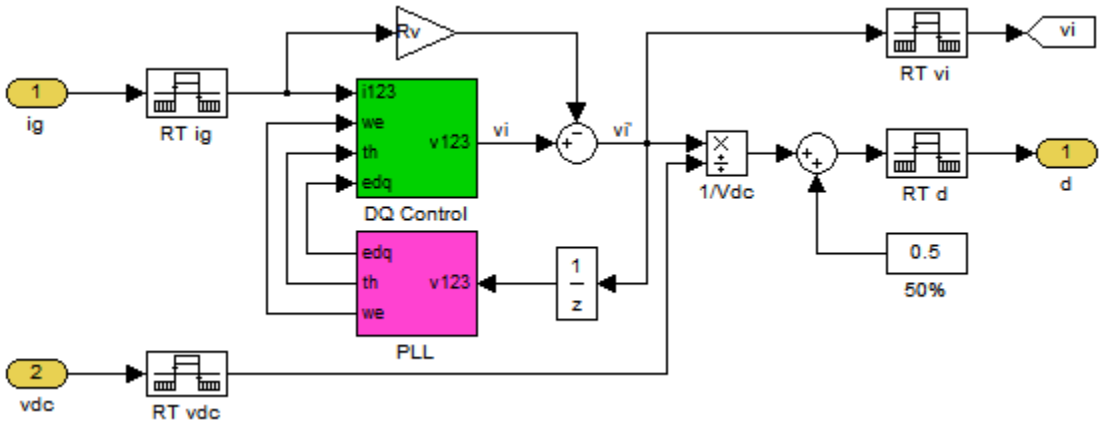


Figure A.4: Top-level control system diagram of MATLAB Simulink virtual-resistance simulation.

the *dq*-frame. This is shown in Figure A.5. The controllers are preceded by the Clarke & Park transformations and followed by the inverse Park & Clarke transformations. The outputs from the PID-controllers are the voltages used to correct any errors in the regulation of the current. These voltages are added to the pseudo-feed-forward voltage terms, which traditionally would come from the AC voltage sensors but now are fed from the PLL. These voltages are simply passed through the PLL block however – they're simply equal to the previous values of the inverter output voltages.

The partial-inductive decoupling terms can also be seen in Figure A.5. These are used when implementing virtual-inductors in the control system. For the virtual-resistance simulations, the value of $L_{VI} = 0$ so these terms resulted in a zero-output. A manual switch was also placed in-line with the PLL frequency, allowing the user to switch the frequency input to zero and eliminate the decoupling. In the cases where $L_{VI} \neq 0$, this was useful for debugging, to see what effects the decoupling terms had on the dynamics of the control system.

The PID-controllers themselves are also Simulink subsystems. The subsystem for one of the

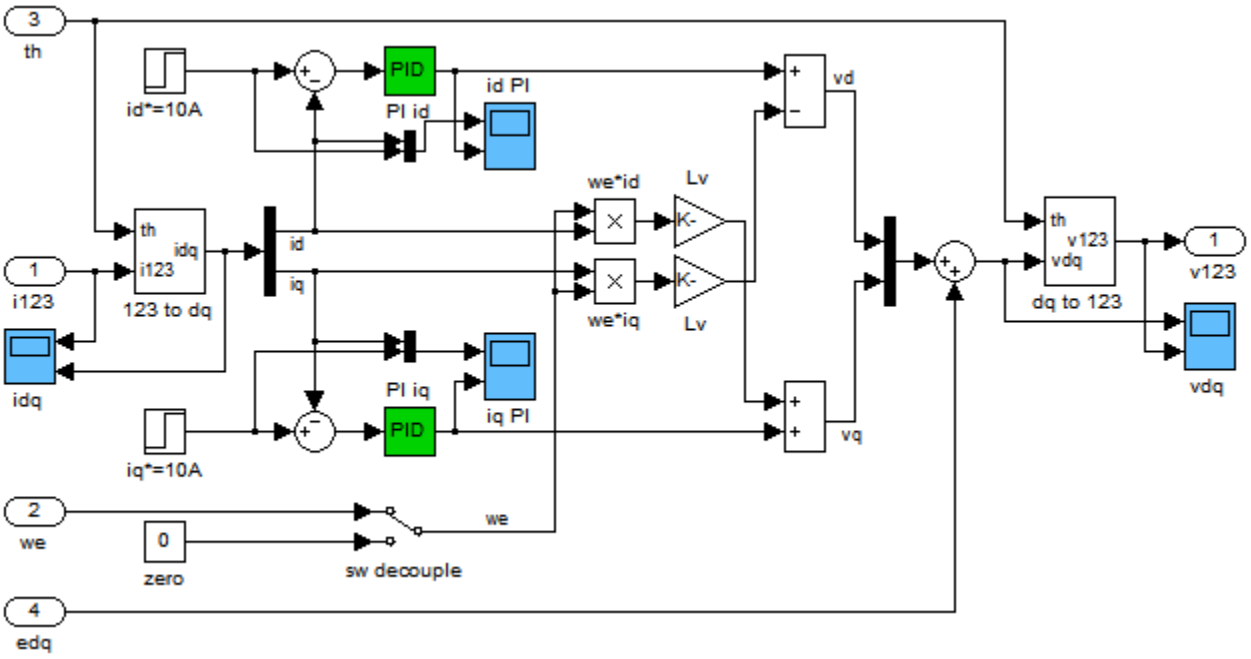


Figure A.5: MATLAB Simulink model of *dq*-controllers with decoupling terms.

controllers is shown in Figure A.6. They simply consist of the basic mathematical blocks needed to compute the output of the basic PID transfer-function, shown below:

$$C(s)=\left(sk_{d,c}+k_{p,c}+\frac{k_{i,c}}{s}\right)=\frac{s^2k_{d,c}+sk_{p,c}+k_{i,c}}{s}. \tag{A.1}$$

Saturation blocks were also used within the PID-subsystems to limit the output of the controllers to reasonable values. However, one key difference in the PID-subsystems is the implementation of the derivative term. Since it is usually desirable to avoid pure derivatives (as discussed in Chapter 2), the calculation of the derivative here actually uses a high-pass filter, similar to what was done for computing the voltage-drop of a virtual inductance. Thus, the transfer function above is effectively changed to the following:

$$C(s)=\left(\frac{s\omega_D k_{d,c}}{s+\omega_D}+k_{p,c}+\frac{k_{i,c}}{s}\right)=\frac{s^2k_{d,c}+\left(sk_{p,c}+k_{i,c}\right)\left(s+\omega_D\right)}{s\left(s+\omega_D\right)}. \tag{A.2}$$

This approach was used to help reduce high-frequency noise from propagating through the control system. Ideally, the filter cut-off frequency, ω_D, should be chosen high enough that the filter's response functions identically to a pure derivative at the frequencies of interest. In most simulations, a cut-off frequency around $f_D \approx 3$ kHz was used to approximate the derivative.

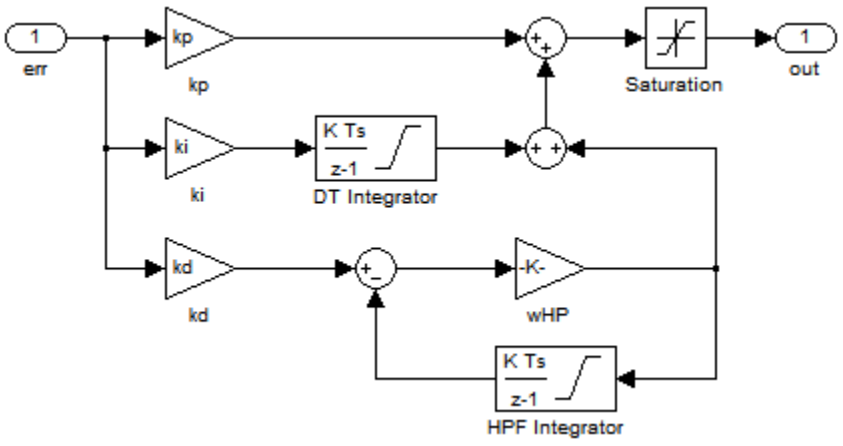


Figure A.6: MATLAB Simulink model of PID controller implemented with a high-pass filter.

Finally, the block-diagram for the Phase-Locked Loop (PLL) is presented in Figure A.7. The structure of the PLL is basically the same as was described back in Section 1.2. Additional features were added to for debugging purposes, and to see their effect on the system dynamics. These features, most of them being low-pass filters, can be enabled or disabled through the toggling of manual switches by the user. Alternate normalization methods for the PLL's error signal were also examined. In the figure, the two options to compute the angle-error are to 1) compute the arctangent of the *dq*-voltages, or 2) simply divide the *q*-axis component by the magnitude (absolute-value) of the *d*-axis component. This second method should function very similarly to the hypotenuse calculation described by equation (1.14) in Chapter 1;

$$v_{Sq} \approx 0 \quad \Rightarrow \quad \varepsilon_\theta = \left(\theta_S - \hat{\theta}\right) = \frac{v_{Sq}}{\sqrt{{v_{Sd}}^2 + {v_{Sq}}^2}} \approx \frac{v_{Sq}}{|v_{Sd}|}. \tag{A.3}$$

The angle-error can also be directly added to the estimated angle through a derivative-like control path, but this was not used in the simulations or experiments. A standard PI-controller was used to regulate the angle-error to zero.

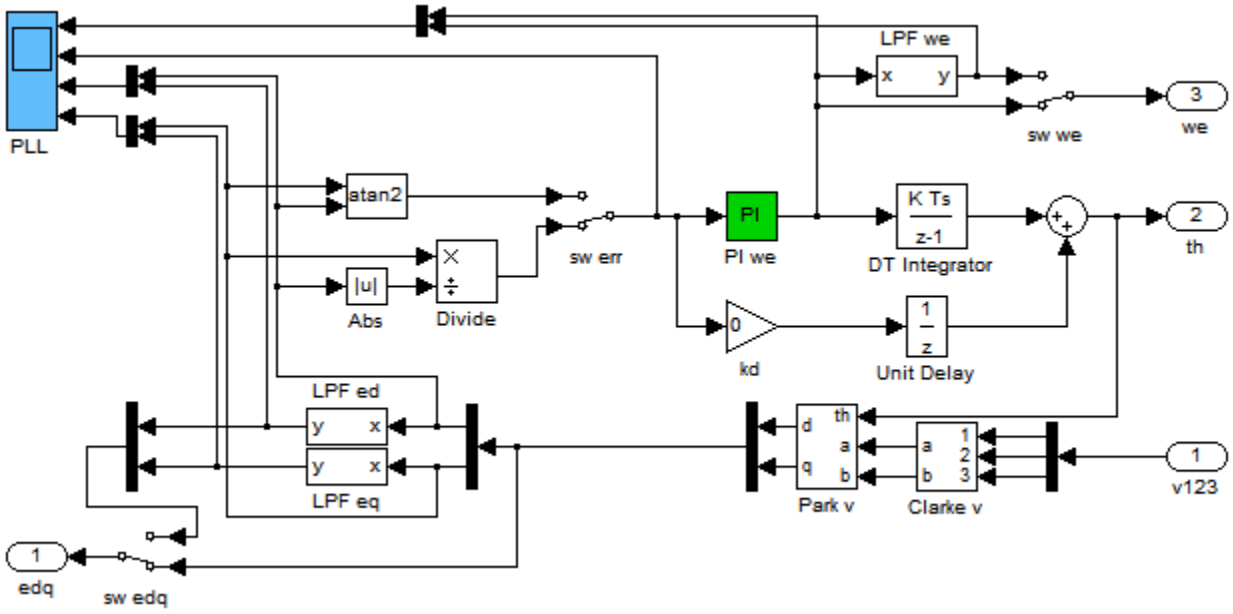


Figure A.7: MATLAB Simulink model of Phase-Locked Loop subsystem.

APPENDIX B: DETAILS ON 1 KW EXPERIMENTAL PROTOTYPE

As discussed in Chapter 2, a Silicon-Carbide (SiC) three-phase inverter was developed to validate the proposed virtual-resistance self-synchronization control algorithm. In this appendix, additional details specific to the design of the inverter will be shared.

The inverter prototype consisted of two printed circuit boards (PCBs). Both were developed using the Altium Designer PCB layout software. The first, referred to as the "power board", was designed to carry the main power-transistors and sensors. A 3D CAD model of the designed "power board" PCB is shown in Figure B.1 below. The overall dimensions of this PCB were

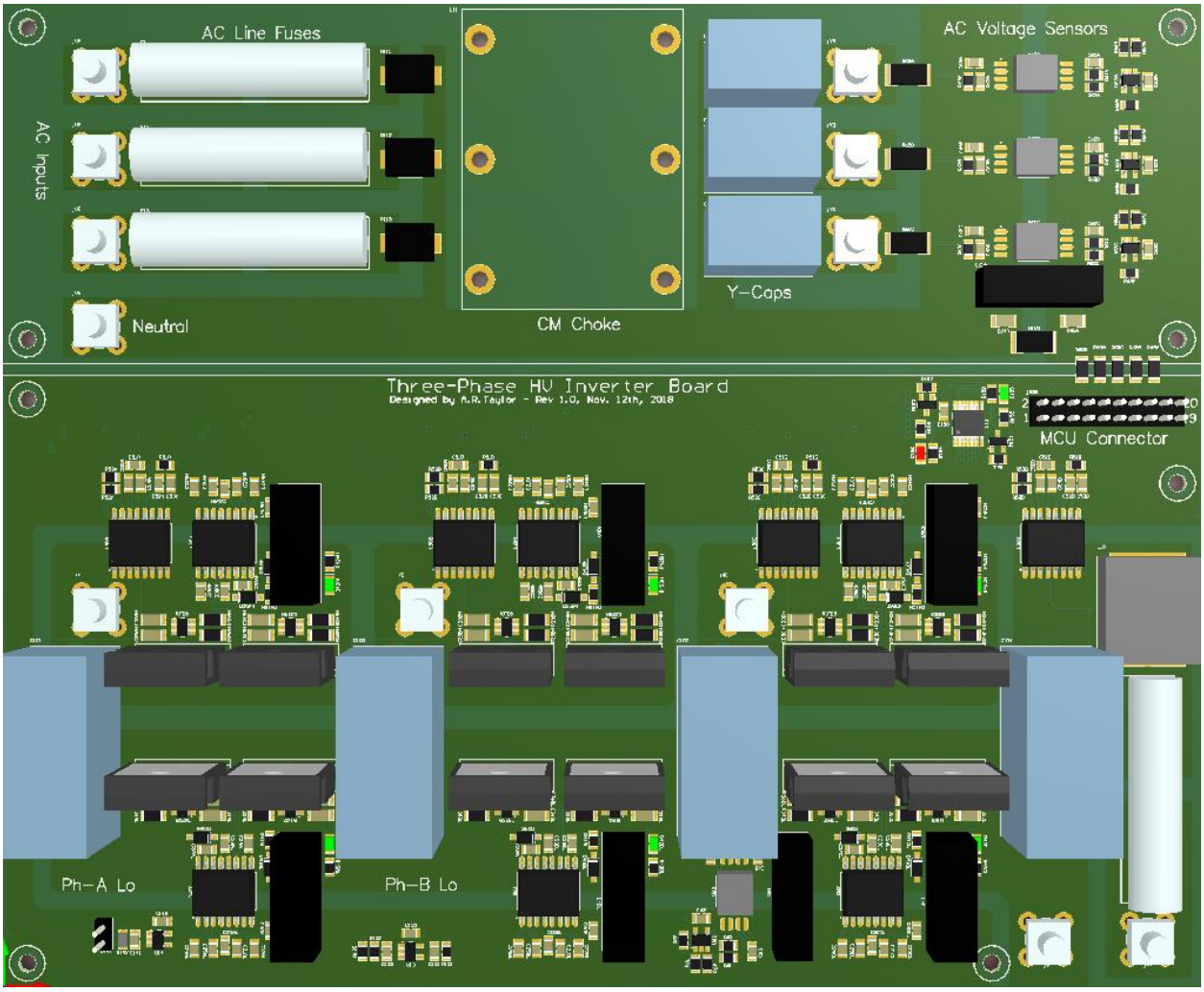


Figure B.1: PCB layout of a three-phase 1 kW 400 VDC inverter for experiments.

164 × 200 mm. The PCB is composed of four copper layers, each manufactured with a 2-oz copper thickness to help conduct heavy currents (the load currents are conducted through the board traces in some areas). Some components, whose 3D models were not obtained (such as the common-mode choke), are not shown in the CAD model, but they can be seen in the physically-populated board; this is shown in Figure B.2 below. Other components which are visible in Figure B.2 are the heatsinks for the MOSFETs, the wiring of power connections, and the second "control board" PCB, which holds the DSP and peripheral circuitry. The USB communication cable can also be seen here in the figure.

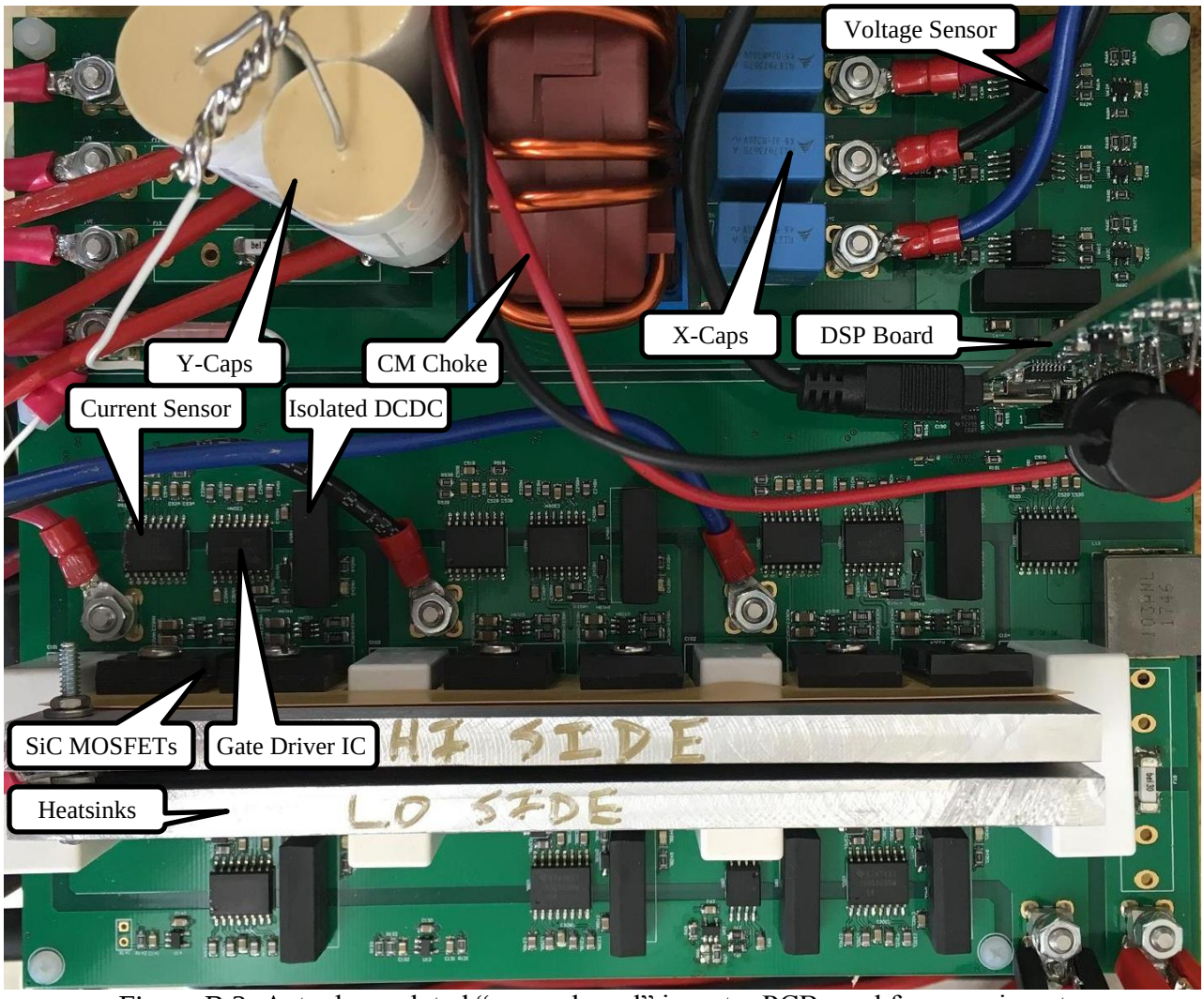


Figure B.2: Actual populated "power board" inverter PCB used for experiments.

The circuitry placed on the "power board" mainly consists of: isolated sensor and gate-drive power supplies; isolated current sensors (AC and DC); isolated voltage sensors (AC and DC); isolated gate-driver ICs; the main SiC power MOSFETs; and other passive components such as the DC-link capacitors, X-caps and Y-caps for filtering the grid connection, and the common-mode choke (used to filter out common-mode PWM voltages from the inverter).

The gate driver IC used is the ISO5500 by Texas Instruments (TI). The specific SiC-MOSFETs used were the SCT3060AL devices by ROHM Semiconductor. The cover-page of the component datasheet is shown below in Figure B.3. The inverter was designed to use two of

ROHM SEMICONDUCTOR

SCT3060AL

N-channel SiC power MOSFET Datasheet

V_{DSS}	650V
$R_{DS(on)}$ (Typ.)	60mΩ
I_D	39A
P_D	165W

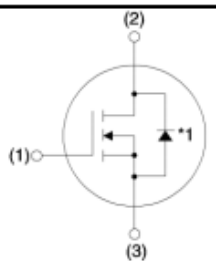
●Outline

TO-247N

●Features

1) Low on-resistance
2) Fast switching speed
3) Fast reverse recovery
4) Easy to parallel
5) Simple to drive
6) Pb-free lead plating ; RoHS compliant

●Inner circuit



●Application

・Solar inverters
・DC/DC converters
・Switch mode power supplies
・Induction heating
・Motor drives

●Packaging specifications

Type	Packing	Tube
	Reel size (mm)	-
	Tape width (mm)	-
	Basic ordering unit (pcs)	30
	Taping code	C11
	Marking	SCT3060AL

Figure B.3: Datasheet of the SCT3060AL SiC-MOSFET used in the inverter prototype.

these SiC-MOSFETs in parallel for each switching cell in the inverter. Thus, the effective $R_{DS(on)}$ resistance of the inverter's phase legs was ~30 mΩ and the rated current capacity was ~78 A. The system was intentionally over-designed to avoid potential power-loss issues or thermal issues during testing; the expected currents during testing would be no more than ~20 A.

A part of the schematics for the gate-drive circuitry is shown in Figure B.4. To enhance the gate-drive current capability, a BJT Common-Collector (push-pull) amplifier was placed between the gate-driver IC and MOSFET gates. This can be seen in the figure. The BJTs selected were the ZXTC2063E6 complementary NPN + PNP paired devices by Diodes Incorporated. Besides enhancing the gate-drive current beyond the capability of the driver output, another main benefit of using the discrete BJT amplifier circuits is that the effective gate-source loops of each MOSFET can be made extremely small in the PCB layout. The power supplies for the gate-drive

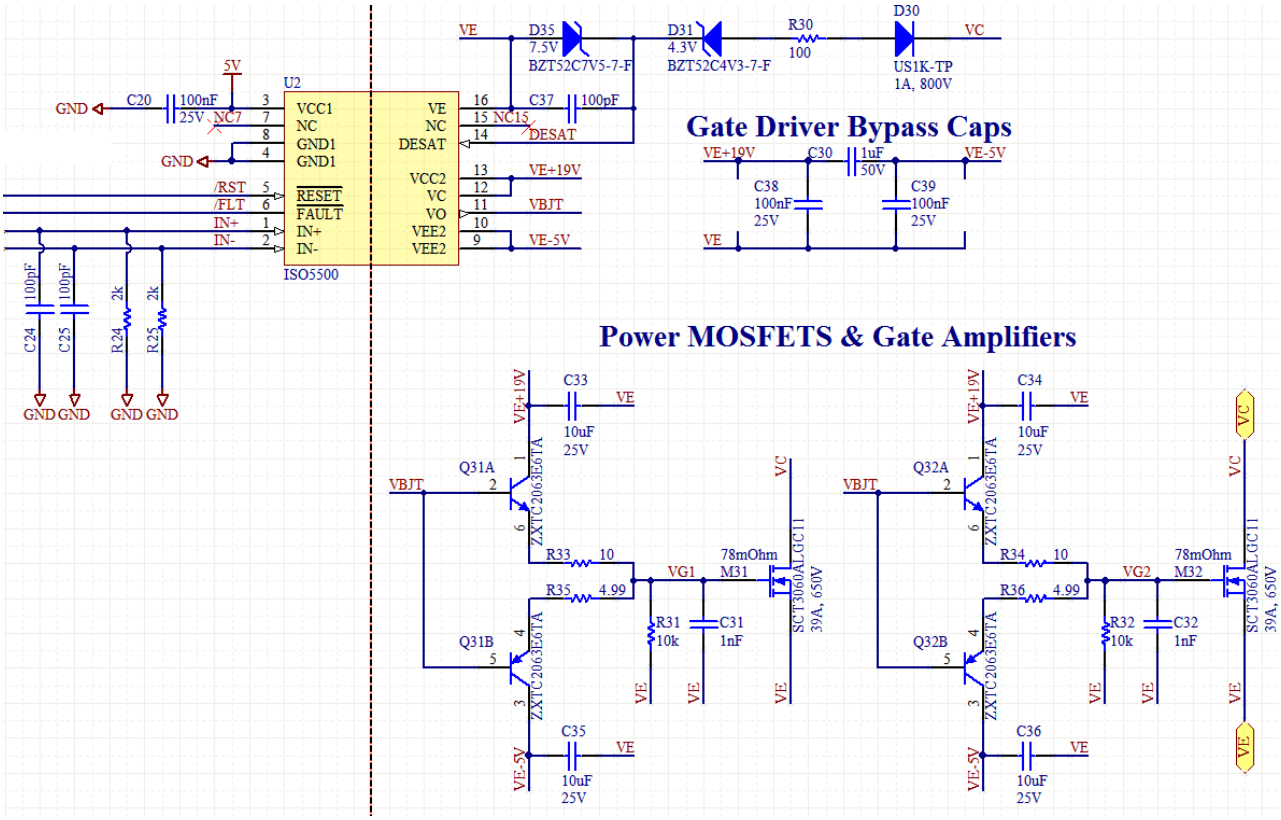
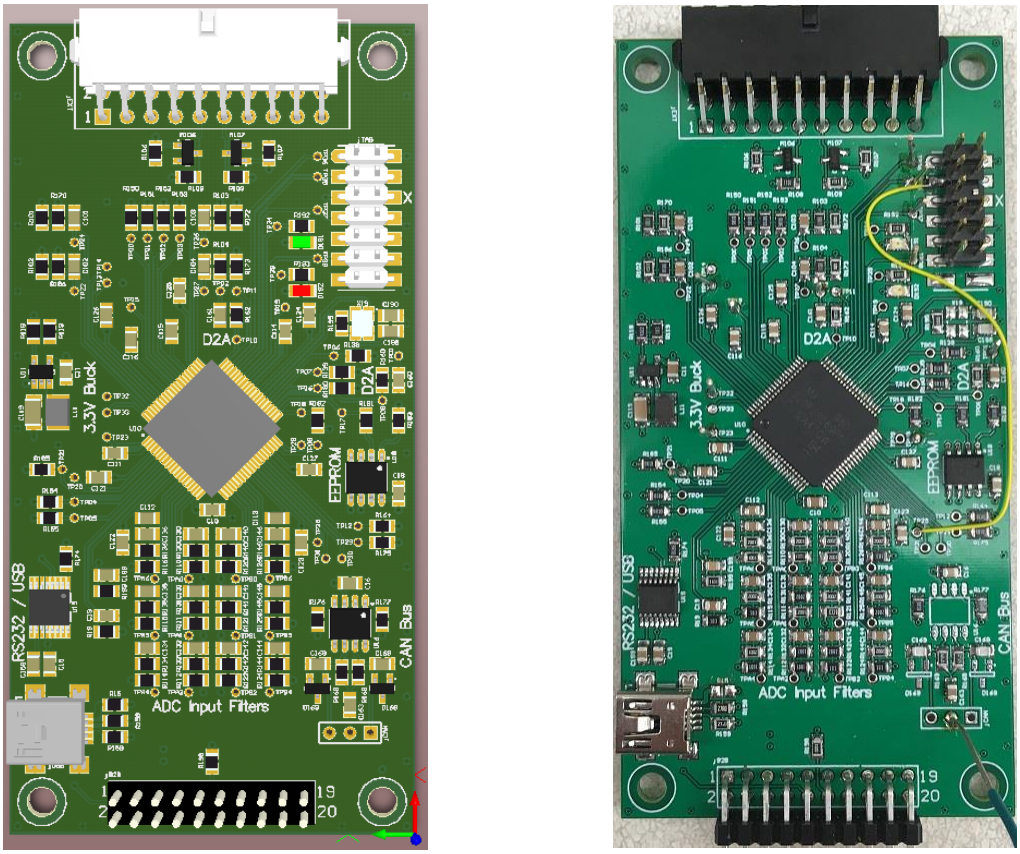


Figure B.4: Electrical schematic showing MOSFET gate driver and gate-drive amplifier circuits.

amplifiers are regulated to +19 V and –5 V; the use of a high turn-on voltage helps to ensure a low $R_{DS(on)}$ value and the use of a negative turn-off voltage helps to prevent erroneous turn-on events caused by the Miller capacitance (the C_{GD} capacitance) of the MOSFETs.

The "control board" PCB was designed in a somewhat modular fashion so that it could be used with other future projects. Like the "power board", the "control board" was also a 4-layer PCB design. The components contained on the "control board" PCB include: the main Digital Signal Processor (DSP); communication circuits and ICs for Controller-Area Network (CAN) and RS-232 (Serial) connections; a local 3.3 V power supply for the DSP; several RC filter circuits for low-pass filtering of the analog input pins; an external Electronically-Erasable Programmable Read-Only Memory (EEPROM) for storing calibration values; and several headers/connectors for connecting to a laptop, the "power board", or external devices. A 3D



(a) (b)

Figure B.5: Layout of the DSP "control board": (a) CAD model; (b) actual populated PCB.

CAD model and the actual populated “control board” PCB can be seen in Figure B.5.

The processor used on the “control board” is the TMS320F28069 floating-point DSP by TI. It features a 90 MHz rated clock-speed, a 12-bit analog-to-digital (A2D) converter, a built-in hardware floating-point unit (FPU), and several PWM-capable output pins. The FPU is exceptionally useful in speeding up code execution time, as the main control algorithm is written using 32-bit floating-point variables and mathematical operations. During testing, the DSP was actually over-clocked to 100 MHz to help speed up the control algorithm and simplify some of the calculations (since the cycle-time would change from 11.11 ns to simply 10 ns).

The main control algorithm which runs on the DSP was written in the C-programming language; it consists of over 1000 lines of code. The main control loop, which calculates the PWM duty cycles and regulates the AC currents, runs via a CPU-timer interrupt which triggers at a rate of 20 kHz, or every 50 μs (as mentioned previously in Table 2.5). At the time of writing, the algorithm requires roughly 65% of the CPU resources (requiring about 32 μs) to process the virtual-resistance control algorithm. This means that the algorithm speed could potentially be safely increased to 25 kHz (40 μs) using the same DSP and clock speed.

APPENDIX C: LIST OF SCHOLARLY WORKS

Within this section, a list of previous scholarly works developed by the author and other scholars is provided. They are sorted in chronological order from newest to oldest.

JOURNAL PAPERS:

1. **A. Taylor**, J. Lu, L. Zhu, K. Bai, M. McAmmond and A. Brown, "Comparison of SiC MOSFET-based and GaN HEMT-based high-efficiency high-power-density 7.2 kW EV battery chargers," in *IET Power Electronics*, vol. 11, no. 11, pp. 1849-1857, 18 9 2018.

2. L. Zhu, **A. R. Taylor**, G. Liu and K. Bai, "A Multiple-Phase-Shift Control for a SiC-Based EV Charger to Optimize the Light-Load Efficiency, Current Stress, and Power Quality," in *IEEE Journal of Emerging and Selected Topics in Power Electronics*, vol. 6, no. 4, pp. 2262-2272, Dec. 2018.

3. J. Lu, H. Bai, **A. Taylor**, G. Liu, A. Brown, P. M. Johnson and M. McAmmond, "A Modular-Designed Three-Phase High-Efficiency High-Power-Density EV Battery Charger Using Dual/Triple-Phase-Shift Control," in *IEEE Transactions on Power Electronics*, vol. 33, no. 9, pp. 8091-8100, Sept. 2018.

4. **A. Taylor**, G. Liu, H. Bai, A. Brown, P. M. Johnson and M. McAmmond, "Multiple-Phase-Shift Control for a Dual Active Bridge to Secure Zero-Voltage Switching and Enhance Light-Load Performance," in *IEEE Transactions on Power Electronics*, vol. 33, no. 6, pp. 4584-4588, June 2018.

5. J. Lu, G. Liu, H. Bai, A. Brown, P. M. Johnson, M. McAmmond and **A. R. Taylor**, "Applying Variable-Switching-Frequency Variable-Phase-Shift Control and E-Mode GaN HEMTs to an Indirect Matrix Converter-Based EV Battery Charger," in *IEEE Transactions on Transportation Electrification*, vol. 3, no. 3, pp. 554-564, Sept. 2017.

6. A. Traore, **A. Taylor**, M. Zohdy and F. Peng, "Modeling and Simulation of a Hybrid Energy Storage System for Residential Grid-Tied Solar Microgrid Systems," *Journal of Power and Energy Engineering*, vol. 5, no. 5, pp. 28-39, May 2017.

7. F. Yang, **A. R. Taylor**, H. Bai, B. Cheng and A. A. Khan, "Using d–q Transformation to Vary the Switching Frequency for Interior Permanent Magnet Synchronous Motor Drive Systems," in *IEEE Transactions on Transportation Electrification*, vol. 1, no. 3, pp. 277-286, Oct. 2015.

8. F. Yang, C. Jiang, **A. Taylor**, H. Bai, A. Kotrba, A. Yetkin and A. Gundogan, "Design of a High-Efficiency Minimum-Torque-Ripple 12-V/1-kW Three-Phase BLDC Motor Drive System for Diesel Engine Emission Reductions," in *IEEE Transactions on Vehicular Technology*, vol. 63, no. 7, pp. 3107-3115, Sept. 2014.

9. C. Duan, C. Jiang, **A. Taylor** and K. Bai, "Design of a zero-voltage-switching large-air-gap wireless charger with low electric stress for electric vehicles," in *IET Power Electronics*, vol. 6, no. 9, pp. 1742-1750, November 2013.

10. H. Bai, **A. Taylor,** W. Guo, G. Szatmari-Voicu, N. Wang, J. Patterson and J. Kane, "Design of an 11 kW power factor correction and 10 kW ZVS DC/DC converter for a high-efficiency battery charger in electric vehicles," in *IET Power Electronics*, vol. 5, no. 9, pp. 1714-1722, November 2012.

CONFERENCE PAPERS:

1. X. Wang, D. Gunasekaran, **A. Taylor**, W. Qian and F. Z. Peng, "Comprehensive Design and Control of Electric Powertrain Evaluation Platform for Next Generation EV/HEV Development," *2018 IEEE Transportation Electrification Conference and Expo (ITEC)*, Long Beach, CA, 2018, pp. 237-242.

2. G. Liu, H. Bai, M. McAmmond, A. Brown, P. M. Johnson, **A. Taylor** and J. Lu., "Comparison of SiC MOSFETs and GaN HEMTs based high-efficiency high-power-density 7.2kW EV battery chargers," *2017 IEEE 5th Workshop on Wide Bandgap Power Devices and Applications (WiPDA)*, Albuquerque, NM, 2017, pp. 391-397.

3. **A. Taylor**, J. Lu, H. K. Bai, A. Brown and M. McAmmond, "A model-based buck-type active filter using proportional-resonant controller and GaN HEMTs," *2017 IEEE Applied Power Electronics Conference and Exposition (APEC)*, Tampa, FL, 2017, pp. 3195-3199.

4. K. Berry, A. Traore, A. Krishna, P. Gangadhar and **A. Taylor**, "Power Systems Infrastructure of Hybrid Electric Fuel Cell Competition Go Kart," *SAE 2017 International Powertrains, Fuels & Lubricants Meeting*, SAE Technical Paper, 2017-01-2452, 2017.

5. F. Yang, **A. Taylor**, H. Bai, B. Cheng, A. Khan, Y. J. Lee and Z. Nie, "Using D-Q transformation to variable switching frequency PWM control for interior permanent magnet synchronous motor drives," *2014 IEEE Energy Conversion Congress and Exposition (ECCE)*, Pittsburgh, PA, 2014, pp. 5192-5197.

6. Fei Yang, **A. Taylor**, H. Bai, B. Cheng, A. Khan, Y. J. Lee and Z. Nie, "Application of variable switching frequency PWM control with power loss prediction in surfaced-mounted permanent magnet synchronous motor," *2014 IEEE Conference and Expo Transportation Electrification Asia-Pacific (ITEC Asia-Pacific)*, Beijing, 2014, pp. 1-4.

7. Y. Li, **A. Taylor** and K. Bai, "A hybrid observer for the full-speed-range sensorless control of interior permanent magnet motor drives," *2014 IEEE Transportation Electrification Conference and Expo (ITEC)*, Dearborn, MI, 2014, pp. 1-5.

8. **A. Taylor**, C. Jiang, K. H. Bai, A. Kotrba, A. Yetkin and A. Gundogan, "Design of a high-efficiency 12V/1kW 3-phase BLDC motor drive system for diesel engine emissions reductions," *2013 IEEE Energy Conversion Congress and Exposition*, Denver, CO, 2013, pp. 1077-1081.

9. C. Jiang, **A. Taylor**, C. Duan and K. Bai, "Extended Kalman Filter based battery state of charge (SOC) estimation for electric vehicles," *2013 IEEE Transportation Electrification Conference and Expo (ITEC)*, Detroit, MI, 2013, pp. 1-5.

10. C. Duan, C. Jiang, **A. Taylor** and K. Bai, "Design of a zero-voltage-switching large-air-gap wireless charger with low electrical stress for Plugin Hybrid Electric Vehicles," *2013 IEEE Transportation Electrification Conference and Expo (ITEC)*, Detroit, MI, 2013, pp. 1-5.

11. W. Guo, K. Bai, **A. Taylor**, J. Patterson and J. Kane, "A novel soft starting strategy of an LLC resonant DC/DC converter for plug-in hybrid electric vehicles," *2013 Twenty-Eighth Annual IEEE Applied Power Electronics Conference and Exposition (APEC)*, Long Beach, CA, 2013, pp. 2012-2015.

12. Wei Guo, H. Bai, G. Szatmari-Voicu, **A. Taylor**, J. Patterson and J. Kane, "A 10kW 97%-efficiency LLC resonant DC/DC converter with wide range of output voltage for Plug-in Hybrid Electric Vehicles," *2012 IEEE Transportation Electrification Conference and Expo (ITEC)*, Dearborn, MI, 2012, pp. 1-4.

13. **A. Taylor**, Xuntuo Wang, Hua Bai, G. Szatmari-Voicu, J. Patterson and J. Kane, "Design of A 97%-efficiency 10kW power factor correction for fast electric chargers of Plug-in Hybrid Electric Vehicles," *2012 IEEE Transportation Electrification Conference and Expo (ITEC)*, Dearborn, MI, 2012, pp. 1-6.

PATENTS:

1. **A. R. Taylor**, A. W. Brown, P. M. Johnson, "Multi-Phase-Shift Control of a Power Converter," U.S. Patent No. WO2019064259A1. Washington, DC: U.S. Patent and Trademark Office, 2019.

REFERENCES

REFERENCES

[1] J. W. Kolar and T. Friedli, "The Essence of Three-Phase PFC Rectifier Systems—Part I," in IEEE Transactions on Power Electronics, vol. 28, no. 1, pp. 176-198, Jan. 2013.

[2] A. Thomas and S. D. Smitha, “A Self-synchronized Synchronverter Technology for Integrating PV Inverters to Grid without Using a Phase Locked Loop,” Int. J. Innov. Res. Technol., vol. 2, no. 5, pp. 24–29, 2015.

[3] M. Ciobotaru, R. Teodorescu, and F. Blaabjerg, “A New Single-Phase PLL Structure Based on Second Order Generalized Integrator,” in 37th IEEE Power Electronics Specialists Conference, 2006, no. July, pp. 1–6.

[4] W. Li and J. Meiners, “Introduction to phase-locked loop system modeling,” Analog Appl., vol. 1, no. May, pp. 5–11, 2000.

[5] S. Long, "Phase Locked Loop Circuits," University of California, Santa Barbra, 2005.

[6] G. Nash, "Phase-Locked Loop Design Fundamentals," Application Note, Freescale Semiconductor, 2006.

[7] R. M. dos S. Filho, P. F. Seixas, and P. C. Cortizo, “A Comparative Study of Three-Phase and Single-Phase PLL Algorithms for Grid-Connected Systems,” in Proceedings of INDUSCON Conference, 2006, pp. 1–7.

[8] U. A. Miranda, L. G. B. Rolim, and M. Aredes, “A DQ Synchronous Reference Frame Current Control for Single-Phase Converters,” in IEEE 36th Conference on Power Electronics Specialists, 2005., 2005, vol. 2005, pp. 1377–1381.

[9] B. J. Lyons, J. G. Hayes, and M. G. Egan, “Experimental Investigation of Iron-based Amorphous Metal and 6.5% Silicon Steel for High-Current Inductors in Low-Medium Frequency DC-DC Converters,” in 2007 IEEE Industry Applications Annual Meeting, 2007, pp. 1781–1786.

[10] A. Stadler, T. Stolzke, and C. Gulden, “High frequency high current filter inductors with minimum thermal resistance,” in 2014 16th International Power Electronics and Motion Control Conference and Exposition, 2014, pp. 289–292.

[11] "Leakage inductance." Wikipedia. Wikimedia Foundation, n.d. Web. 9 Oct. 2018. https://en.wikipedia.org/wiki/Leakage_inductance.

[12] A. E. Fitzgerald, C. Kingsley, and S. D. Umans. "Chapter 2. Transformers." Electric Machinery. 6th ed. New York: McGraw-Hill, 2014. pg96. Print.

[13] J.-H. Cho, K.-Y. Choi, Y.-W. Kim, and R.-Y. Kim, "A novel P-Q variations method using a decoupled injection of reference currents for a precise estimation of grid impedance," in 2014 IEEE Energy Conversion Congress and Exposition (ECCE), 2014, pp. 5059–5064.

[14] M. Liserre, R. Teodorescu, and F. Blaabjerg, "Stability of grid-connected PV inverters with large grid impedance variation," in 2004 IEEE 35th Annual Power Electronics Specialists Conference, 2004, vol. 6, pp. 4773–4779.

[15] K. Malmedal, "Underground vs. Overhead Transmission and Distribution," 2009 presentation, NEI Electric Power Engineering, https://www.puc.nh.gov/2008IceStorm/ST&E Presentations/NEI Underground Presentation 06-09-09.pdf

[16] T. Noguchi, H. Tomiki, S. Kondo, & I. Takahashi, "Direct power control of PWM converter without power-source voltage sensors," IEEE Trans. Ind. Appl., vol. 34, no. 3, pp. 473–479, 1998.

[17] S. Hansen, M. Malinowski, F. Blaabjerg, & M. P. Kazmierkowski, "Sensorless control strategies for PWM rectifier," in 15th Annual IEEE Applied Power Electronics Conference and Exposition (APEC), 2000, vol. 2, pp. 832–838.

[18] X. Wang, Y. W. Li, F. Blaabjerg, and P. C. Loh, "Virtual-Impedance-Based Control for Voltage-Source and Current-Source Converters," IEEE Trans. Power Electron., vol. 30, no. 12, pp. 7019–7037, Dec. 2015.

[19] J. A. Suul, S. D'Arco, P. Rodríguez, and M. Molinas, "Impedance-compensated grid synchronisation for extending the stability range of weak grids with voltage source converters," IET Gener. Transm. Distrib., vol. 10, no. 6, pp. 1315–1326, Apr. 2016.

[20] J. A. Suul, M. Molinas, and P. Rodriguez, "Exploring the range of impedance conditioning by virtual inductance for grid connected voltage source converters," IEEE PES Innov. Smart Grid Technol. Conf. Eur., pp. 1–9, 2012.

[21] T.-J. Kwon and J. Draper, "Floating-point division and square root using a Taylor-series expansion algorithm," Microelectronics J., vol. 40, no. 11, pp. 1601–1605, Nov. 2009.

[22] D. Dong, B. Wen, D. Boroyevich, P. Mattavelli, and Y. Xue, "Analysis of Phase-Locked Loop Low-Frequency Stability in Three-Phase Grid-Connected Power Converters Considering Impedance Interactions," IEEE Trans. Ind. Electron., vol. 62, no. 1, pp. 310–321, Jan. 2015.

[23] T. Kim, H.-W. Lee, and M. Ehsani, "Position sensorless brushless DC motor/generator drives: review and future trends," IET Electr. Power Appl., vol. 1, no. 4, p. 557, 2007.

[24] A. Taylor, X. Wang, H. Bai, G. Szatmari-Voicu, J. Patterson, and J. Kane, "Design of a 97%-efficiency 10kW power factor correction for fast electric chargers of plug-in hybrid electric vehicles," in 2012 IEEE Transportation Electrification Conference and Expo, ITEC 2012, 2012, pp. 1–6.