Digital Control in Power Electronics

Digital Control in Power Electronics
Simone Buso and Paolo Mattavelli
www.morganclaypool.com

ISBN-10: 1598291122 paperback
ISBN-13: 9781598291124 paperback

ISBN-10: 1598291130 ebook
ISBN-13: 9781598291131 ebook

DOI10.2200/S00047ED1V01Y200609PEL002

A lecture in the Morgan & Claypool Synthesis Series
LECTURES ON POWER ELECTRONICS #2

Lecture #2
Series Editor: Jerry Hudgins, University of Nebraska-Lincoln

Series ISSN: 1930-9525 print
Series ISSN: 1930-9533 electronic

First Edition
10 9 8 7 6 5 4 3 2 1

Printed in the United States of America

Digital Control in Power Electronics

Simone Buso
Department of Information Engineering
University of Padova, Italy

Paolo Mattavelli
Department of Electrical, Mechanical and
Management Engineering
University of Udine, Italy

LECTURES ON POWER ELECTRONICS #2

MORGAN & CLAYPOOL PUBLISHERS

ABSTRACT

This book presents the reader, whether an electrical engineering student in power electronics or a design engineer, some typical power converter control problems and their basic digital solutions, based on the most widespread digital control techniques. The presentation is focused on different applications of the same power converter topology, the half-bridge voltage source inverter, considered both in its single- and three-phase implementation. This is chosen as the case study because, besides being simple and well known, it allows the discussion of a significant spectrum of the more frequently encountered digital control applications in power electronics, from digital pulse width modulation (DPWM) and space vector modulation (SVM), to inverter output current and voltage control. The book aims to serve two purposes: to give a basic, introductory knowledge of the digital control techniques applied to power converters, and to raise the interest for discrete time control theory, stimulating new developments in its application to switching power converters.

KEYWORDS

Digital control in power electronics, Discrete time control theory, Half-bridge voltage source converters, Power converters, Power electronics

Contents

CHAPTER 1

Introduction: Digital Control Application to Power Electronic Circuits

Power electronics and discrete time system theory have been closely related to each other from the very beginning. This statement may seem surprising at first, but, if one thinks of switch mode power supplies as *variable structure periodic* systems, whose state is determined by *logic signals*, the connection becomes immediately clearer. A proof of this may also be found in the first, fundamental technical papers dealing with the analysis and modeling of pulse width modulated power supplies or peak current mode controlled dc–dc converters: they often provide a mathematical representation of both the switching converters and the related control circuits, resembling or identical to that of *sampled data* dynamic systems.

This fundamental contiguousness of the two apparently far areas of engineering is probably the strongest, more basic motivation for the considerable amount of research that, over the years, has been dedicated to the application of digital control to power electronic circuits. From the original, basic idea of implementing current or voltage controllers for switching converters using digital signal processors or microcontrollers, which represents the foundation of all current industrial applications, the research focus has moved to more sophisticated approaches, where the design of custom integrated digital controllers is no longer presented like an academic curiosity, but is rather perceived like a sound, viable solution for the next generation of high-performance power supplies.

If we consider the acceleration in the scientific production related to these topics in the more recent years, we can easily anticipate, for a not too far ahead future, the creation of energy processing circuits, where power devices and control logic can be built on the same semiconductor die. From this standpoint, the distance we see today between the tools and the design methodology of power electronics engineers and those of analog and/or digital integrated circuit designers can be expected to significantly reduce in the next few years.

We have to admit that, in this complex scenario, the purpose of this book is very simple. We just would like to introduce the reader to basic control problems in power electronic circuits and to illustrate the more classical, widely applied digital solutions to those problems. We hope this will serve two purposes: first, to give a basic, introductory knowledge of the digital control techniques applied to power converters, and second, to raise the interest for discrete time control theory, hopefully stimulating new developments in its application to power converters.

1.1 MODERN POWER ELECTRONICS

Classical power electronics may be considered, under several points of view, a mature discipline. The technology and engineering of discrete component based switch mode power supplies are nowadays fully developed industry application areas, where one does not expect to see any outstanding innovation, at least in the near future. Symmetrically, at the present time, the research fields concerning power converter topologies and the related conventional, analog control strategies seem to have been thoroughly explored.

On the other hand, we can identify some very promising research fields where the future of power electronics is likely to be found. For example, a considerable opportunity for innovation can be expected in the field of large bandgap semiconductor devices, in particular if we consider the semiconductor technologies based on silicon carbide, SiC, gallium arsenide, GaAs, and gallium nitride, GaN. These could, in the near future, prove to be practically usable not only for ultra-high-frequency amplification of radio signals, but also for power conversion, opening the door to high-frequency (multi-MHz) and/or high-temperature power converter circuits and, consequently, to a very significant leap in the achievable power densities.

The rush for higher and higher power densities motivates research also in other directions. Among these, we would like to mention three that, in our vision, are going to play a very significant role. The first is the integration in a single device of magnetic and capacitive passive components, which may allow the implementation of minimum volume, quasi monolithic, converters. The second is related to the analysis and mitigation of electromagnetic interference (EMI), which is likely to become fundamental for the design of compact, high frequency, converters, where critical autosusceptibility problems can be expected. The third one is the development of technologies and design tools allowing the integration of control circuits and power devices on the same semiconductor chip, according to the so-called *smart power* concept. These research areas represent good examples of what, in our vision, can be considered modern power electronics.

From this standpoint, the application of digital control techniques to switch mode power supplies can play a very significant role. Indeed, the integration of complex control functions, such as those that are likely to be required by the next generation power supplies,

is a problem that can realistically be tackled only with the powerful tools of digital control design.

1.2 WHY DIGITAL CONTROL

The application of digital control techniques to switch mode power supplies has always been considered very interesting, mainly because of the several advantages a digital controller shows, when compared to an analog one.

Surely, the most relevant one is the possibility it offers for implementing sophisticated control laws, taking care of nonlinearities, parameter variations or construction tolerances by means of self-analysis and autotuning strategies, very difficult or impossible to implement analogically.

Another very important advantage is the flexibility inherent in any digital controller, which allows the designer to modify the control strategy, or even to totally reprogram it, without the need for significant hardware modifications. Also very important are the higher tolerance to signal noise and the complete absence of ageing effects or thermal drifts.

In addition, we must consider that, nowadays, a large variety of electronic devices, from home appliances to industrial instrumentation, require the presence of some form of man to machine interface (MMI). Its implementation is almost impossible without having some kind of embedded microprocessor. The utilization of the computational power, which thus becomes available, also for lower level control tasks is almost unavoidable.

For these reasons, the application of digital controllers has been increasingly spreading and has become the only effective solution for a whole lot of industrial power supply production areas. To give an example, adjustable speed drives (ASDs) and uninterruptible power supplies (UPSs) are nowadays fully controlled by digital means.

The increasing availability of low-cost, high-performance, microcontrollers and digital signal processors stimulates the diffusion of digital controllers also in areas where the cost of the control circuitry is a truly critical issue, like that of power supplies for portable equipment, battery chargers, electronic welders and several others.

However, a significant increase of digital control applications in these very competing markets is not likely to take place until new implementation methods, different from the traditional microcontroller or DSP unit application, prove their viability. From this standpoint, the research efforts towards digital control applications need to be focused on the design of custom integrated circuits, more than on algorithm design and implementation. Issues such as occupied area minimization, scalability, power consumption minimization and limit cycle containment play a key role. The power electronics engineer is, in this case, deeply involved in the solution of digital integrated circuit design problems, a role that will be more and more common in the future.

1.3 TRENDS AND PERSPECTIVES

From the above discussion, it will be no surprise if we say that we consider the increasing diffusion of digital control in power electronics virtually unstoppable. The advantages of the digital control circuits, as we have briefly outlined in the previous section, are so evident that, in the end, all the currently available analog integrated control solutions are going to be replaced by new ones, embedding some form of digital signal processing core. Indeed, it is immediate to recognize that the digital control features perfectly match the needs of present and, even more, future, highly integrated, power converters. The point is only how long this process is going to take. We can try to outline the future development of digital controllers distinguishing the different application areas.

The medium-to high-power applications, such as electrical drives, test power supplies, uninterruptible power supplies, renewable energy source interfaces, are likely to be developed according to the same basic hardware organization for a long time to come. The application of microcontroller units or digital signal processors in this area is likely to remain very intensive. The evolution trend will probably be represented by the increasing integration of higher level functions, e.g., those concerning communication protocols for local area networks or field buses, man to machine interfaces, remote diagnostic capabilities, that currently require the adoption of different signal processing units, with low-level control functions.

As far as the low power applications are concerned, as we mentioned in the previous section, we cannot, at the moment, describe an established market for digital controllers. However, the application of digital control in this field is the object of an intensive research. In the near future, new control solutions can be anticipated, which will replace analog controllers with equivalent digital solutions, in a way that can be considered almost transparent to the user. Successively, the complete integration of power and control circuitry is likely to determine a radical change in the way low power converters are designed.

1.4 WHAT IS IN THIS BOOK

As mentioned above, in front of the complex and exciting perspectives for the application of digital control to power converters, we decided to aim this book at giving the reader a basic and introductory knowledge of some typical power converter control problems and their digital solutions. Referring to the above discussion, we decided to dedicate the largest part of our presentation to topics that can be considered the current state of the art for industrial applications of digitally controlled power supplies.

The book is consequently proposed to power electronics students, or designers, who would like to have an overview of the most widespread digital control techniques. It is not intended to provide an exhaustive description of all the possible solutions for any considered problem, nor

to describe the more recent research advances related to any of them. This choice has allowed us to keep the presentation of the selected materials relatively agile and to give it an immediate, practical usefulness.

Accordingly, what the reader should know to take full advantage of the contents that are presented here is relatively little: a basic knowledge of some power electronic circuits (essentially half-bridge and full-bridge voltage source inverters) and the fundamental mathematical tools that are commonly employed in modeling continuous and discrete time dynamic system (Laplace transform and Z transform, for starters) will perfectly do.

As the reader will realize, if he or she will have the patience to follow us, the book is conceived to explain the different concepts essentially by means of examples. To limit the risk of being confusing, proposing several different topologies, we decided to take into account a single, relatively simple test case and develop its analysis all along the text. Doing so, the contents we have included allowed us to present, organically and without too many context changes, a significant amount of control techniques and related implementation details.

In summary, the book is organized as follows. Chapter 2 describes the considered test case, a voltage source inverter, and the first control problem, i.e., the implementation of a current control loop, discussing in the first place its analog, i.e., continuous time, solutions. Chapter 3 is dedicated to digital control solutions for the same problem: in the beginning we present a relatively simple one, i.e., the discretization of continuous time controllers. In the following, other fully digital solutions, like those based on discrete time state feedback and pole placement, are presented. Chapter 4 is dedicated to the extension to three phase systems of the solutions presented for the single-phase inverter. In this chapter we discuss space vector modulation (SVM) and rotating reference frame current controllers, like those based on Park's transformation. Finally, Chapter 5 presents the implementation of external control loops, wrapped around the current controller, which is typically known as a multiloop controller organization. The design of an output voltage controller, as is needed in uninterruptible power supplies, is considered first. Both large bandwidth control strategies and narrow bandwidth ones, based on the repetitive control concept, are analyzed. After that, and in conclusion, two other significant examples of multiloop converter control, which we may find in controlled rectifiers and active power filters, are considered and briefly discussed.

CHAPTER 2

The Test Case: a Single-Phase Voltage Source Inverter

The aim of this chapter is to introduce the test case we will be dealing with in the following sections. As mentioned in the introduction, it would be extremely difficult to describe the numerous applications of digital control to switch mode power supplies, since this is currently employed in very wide variety of cases. In order not to confuse the reader with a puzzle of several different circuit topologies and related controllers, what we intend to do is to consider just a single, simple application example, where the basics of the more commonly employed digital control strategies can be effectively explained. Of course, the concepts we are going to illustrate, referring to our test case, can find a successful application also to other converter topologies.

The content of this chapter is made up, in the first place, by an introductory, but fairly complete, description of the power converter we will be discussing throughout this book, i.e., the half-bridge voltage source inverter. Secondly, the principles of its more commonly adopted low-level control strategy, namely pulse width modulation (PWM), will be explained, at first in the continuous time domain, and then in the discrete time domain. The issues related to PWM control modeling are fundamental for the correct formulation of a switch mode power supply (SMPS) digital, or even analog, control problem, so this part of the chapter can be considered essential to the understanding of everything that follows. The final part of the chapter is instead dedicated to a summary of the more conventional analog control strategies, which will serve as a reference for all the following developments.

2.1 THE VOLTAGE SOURCE INVERTER

The considered test case is shown in Fig. 2.1. As can be seen, the power converter we want to take into consideration is a single-phase voltage source inverter (VSI). The VSI has a conventional topological structure, which is known as a half bridge. We will now analyze the power converter's organization in some detail.

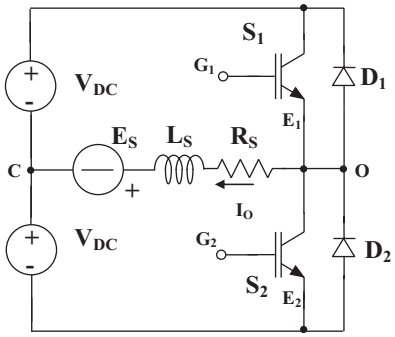

FIGURE 2.1: Half-bridge voltage source inverter.

2.1.1 Fundamental Components

The ideal voltage sources V_{DC} at the input are, in practice, approximately implemented by means of suitably sized capacitors, fed by a primary energy source. They are normally large enough to store a considerable amount of energy and their purpose is to deliver it to the load, rapidly enough not to cause the circulation of substantial high-frequency currents through the primary source. This, in turn, can be represented by any real dc voltage source, from batteries to line-fed rectifiers, depending on the particular application. However, for our discussion, modeling the energy source as an ideal voltage source does not represent any limitation.

The power switches are represented by the conventional IGBT symbol, but it is possible to find implementations with very different switch technologies, such as, for instance, power MOSFETs or, for very high power application, thyristors. As can be seen, each switch is paralleled to a *free-wheeling* diode, whose purpose is to make the switch bidirectional, at least as far as the current flow is concerned. This interesting property makes the VSI of Fig. 2.1 a four-quadrant converter, with the capability of both delivering and absorbing power.

Again, in order to simplify the treatment of our control problems and without any loss of generality, we will assume that the switch plus diode couple behaves like an ideal switch, i.e., one whose voltage is zero in the "on" state and whose current is zero in the "off" state. Moreover, we will assume that the change from the "on" state to the "off" state and vice versa takes place in a null amount of time.

In our simple example, the load will be described as the series connection of a resistor R_S, an inductor L_S, and a voltage source E_S, which can be either dc or ac. We will learn to control the current across the load using several different strategies. It is worth mentioning that, with this particular structure, the load model is capable of representing various different applications of the VSI, including electrical drives, voltage-controlled current sources, and controlled rectifiers.

The role and meaning of the different components, in particular of the voltage E_S, will be different in each case, but the structure will be exactly the same.

2.1.2 Required Additional Electronics: Driving and Sensing

Several components are needed to allow the proper operation of the VSI that were not described in the previous section. First of all, the power switches need to be driven by a suitable control circuit, allowing the controlled commutation of the device from the "on" to the "off" state and vice versa. Depending on the particular switch technology, the driving circuitry will have different implementations. For example, in the case of MOSFET or IGBT switches the driving action consists in the charging and discharging of the device input capacitance, which is, in fact, a power consuming operation. To take care of that, suitable drivers must be adopted, whose input is represented by the logic signals determining the desired state of the switch and output is the power signal required to bring the switch into that state. A typical complication in the operation of drivers is represented by the floating control terminals of the high-side switch (G_1 and E_1 in Fig. 2.1). Controlling the current between those terminals and, simultaneously, that between the same terminals of the low-side switch (G_2 and E_2 in Fig. 2.1) requires the adoption of isolated driving circuits or the generation of floating power supplies, e.g., based on bootstrap capacitors.

We will not discuss further the operation of these circuits and simply assume that the logic state of the control signal is instantaneously turned into a proper switch state. An exception to this will be the discussion of dead-times, presented in the following. Of course, the interested reader can find more details regarding state-of-the-art switch drivers in technical manuals or datasheets, easily available on the world wide web, such as for example [1].

In addition to drivers, the controlled operation of the converter requires the measurements of several electrical variables. Typically, the input voltage of the inverter circuit, V_{DC}, its output current, i.e., the current flowing through the load, I_O, and, sometimes, the voltage E_S are measured and used in the control circuit. The acquisition of those signals requires suitable signal conditioning circuits, analog in nature, that can range from simple resistive voltage dividers and/or current shunts, possibly combined to passive filters, to more sophisticated solutions, for example those employing operational amplifiers, to implement active filters and signal scaling, or Hall sensors, to measure currents without interfering with the power circuit.

In our discussion we will simply assume that the required control signals are processed by suitable conditioning circuits that, in general, will apply some scaling and filtering to each electrical variable. The frequency response of these acquisition filters and the scaling factors implied by sensors and conditioning circuits will be properly taken into account in the controller design example we will present in the following chapters.

2.1.3 Principle of Operation

The principle of operation of the half-bridge inverter of Fig. 2.1 is the following. Closing the high-side switch S_1 imposes a voltage across the load (i.e., V_{OC} in the figure) equal to $+V_{DC}$. In contrast, closing the low-side switch S_2 imposes a voltage $-V_{DC}$ across the load. If a suitable control circuit regulates the *average* voltage across the load (see Section 2.1.4 for a rigorous definition of the average load voltage) between these two extremes, it is clearly possible to make the state variable I_O follow any desired trajectory, provided that this is consistent with the physical limitations imposed by the topology. The main limitation is obvious: the voltage across the load cannot exceed $\pm V_{DC}$. Other limitations can be seen, giving just a little closer look to the circuit. Considering, as an example, the particular case where E_S and R_S are both equal to zero, the current I_O will be limited in its variations, according to the following equation:

$$\left|\frac{dI_O}{dt}\right| \leq \frac{V_{DC}}{L_S}. \tag{2.1}$$

In practice, the maximum current absolute value will be limited as well, mainly because of the limited current handling capability of the active devices. This limitation, different from the previous ones, is not inherent to the circuit topology and will need to be enforced by a current controller, in order to prevent accidental damage to the switches, for example in the case of a short circuit in the load. What should be clear by now is that any controller trying to impose voltages, currents, or current rates of change beyond the above-described limits will not be successful: the limit violation will simply result in what is called *inverter saturation*. It is worth adding that, in our following discussion, we will consider linear models of the VSI, capable of describing its dynamic behavior in a small-signal approximation. Events like inverter saturation, typical of large signal inverter operation, will not be correctly modeled. In order to further clarify these concepts, the derivation of a small-signal linear model for the VSI inverter of Fig. 2.1 is presented in Aside 1.

In the most general case, the VSI controller is organized hierarchically. In the lowest level a controller determines the state of each of the two switches, and in doing so, the average load voltage. This level is called the *modulator* level. The strategy according to which the state of the switches is changed along time is called the *modulation law*. The input to the modulator is the set-point for the average load voltage, normally provided by a higher level control loop. A direct control of the average load voltage is also possible: in this case the VSI is said to operate under open loop conditions. However, this is not a commonly adopted mode of operation, since no control of load current is provided.

Because of that, in the large majority of cases, a current controller can be found immediately above the modulator level. This is responsible for providing the set-point to the modulator.

Similarly, the current controller set-point can be provided by a further external control loop or directly by the user. In the latter case, the VSI is said to operate in *current mode*, meaning that the control circuit has turned a voltage source topology into a controlled current source. We will deal with further external control loops in one of the following chapters; for now, we will focus on the modulator and current control levels.

Indeed, the main purpose of this chapter is exactly to explain how these two basic controller levels are organized and how the current regulators can be properly designed.

2.1.4 Dead-Times

Before we move to describe the modulator level one final remark is needed to complete the explanation of the VSI operation. The issue we want to address here is known as the *switching dead-time*. It is evident from Fig. 2.1 that under no condition the simultaneous conduction of both switches should be allowed. This would indeed result into a short circuit across the input voltage sources, leading to an uncontrolled current circulation through the switches and, very likely, to inverter fatal damage. Any modulator, whatever its implementation and modulation law, should be protected against this event. In the ideal switch hypothesis of Section 2.1.1, the occurrence of switch cross conduction can be easily prevented by imposing, under any circumstances, logically complementary gate signals to the two switches. Unfortunately, in real-life cases, this is not a sufficient condition to avoid cross conduction. It should be known from basic power electronics knowledge that real switch commutations require a finite amount of time and that the commutation time is a complex function of several variables such as commutated current and voltage, gate drive current, temperature, and so on. It is therefore impossible to rely on complementary logic gate signals to protect the inverter. An effective protection against switch cross conduction is implemented by introducing commutation dead-times, i.e., suitable delays before the switch turn-on signal is applied to the gate.

The effect of dead-times is shown in Fig. 2.2 in the hypothesis that a positive current I_O is flowing through the load. The figure assumes that a *period of observation* can be defined, whose duration is T_S, where switches S_1 and S_2 are meant to be on for times t_{ON1} and t_{ON2}, respectively, and where the load current is assumed to be constant (i.e., the load time constant L_S/R_S is assumed to be much longer than the observation period T_S). The existence of such an observation period guarantees that the definition of average load voltage is well posed. By that we simply mean the weighted average over time of the instantaneous load voltage in the period of observation.

To avoid cross conduction the modulator delays S_1 turn on by a time t_{dead}, applying the V_{GE1} and V_{GE2} command signals to the switches. The duration t_{dead} is long enough to allow the safe turn-off of switch S_2 before switch S_1 is commanded to turn on, considering propagation delays through the driving circuitry, inherent switch turn-off delays, and suitable safety margins.

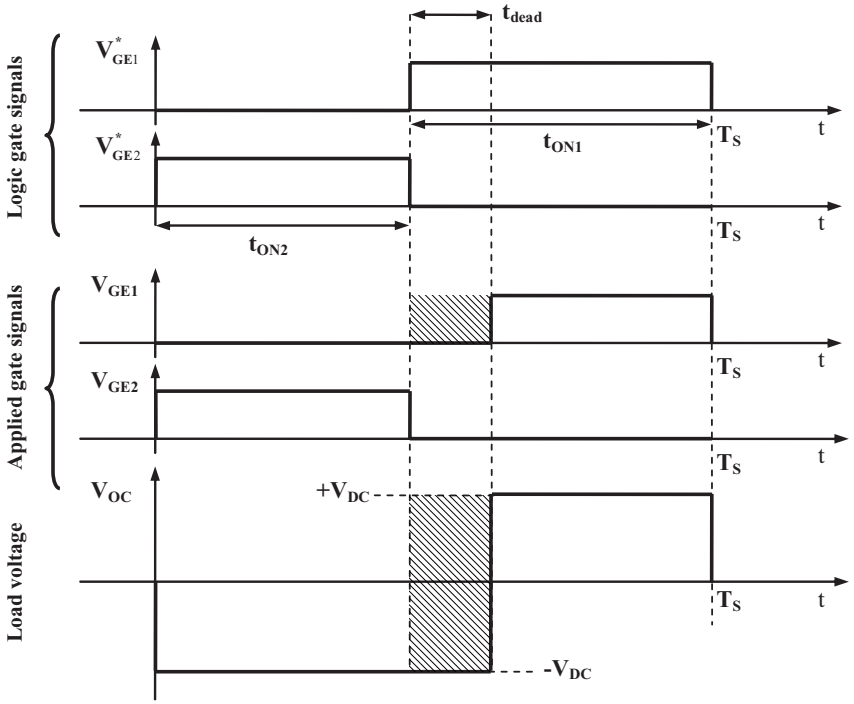

FIGURE 2.2: Dead-times effect: when a positive current I_O flows through the load, the actual on time for switch S_1 is shorter than the desired one. Consequently, the average voltage across the load is different from the desired one.

At the time of writing (2006), the typically required dead-time duration for 600 V, 40 A IGBTs was well below 1 μs. Of course, the dead-time required duration is a direct function of the switch power rating.

Considering Fig. 2.2, it is important to note that the effect of the dead-time application is the creation of a time interval where both switches are in the off state and the load current flows through the free-wheeling diodes. Because of that, a difference is produced between the desired duration of the switch S_1 on time and the actual one, which turns into an error in the voltage across the load. It is as well important to note that the opposite commutation, i.e., where S_1 is turned off and S_2 is turned on, does not determine any such voltage error. However, we must point out that, if the load current polarity were reversed, the dead-time induced load voltage error would take place exactly during this commutation.

The above discussion reveals that, because of dead-times, no matter what the modulator implementation is, an error on the load voltage will always be generated. This error ΔV_{OC}, whose entity is a direct function of the dead-time duration and whose polarity depends on the

load current sign according to the relation

$$\Delta V_{OC} = -2V_{DC}\frac{t_{dead}}{T_S}\text{sign}(I_O), \tag{2.2}$$

will have to be compensated by the current controller. Failure to do so will unavoidably determine a tracking error on the trajectory the load current has to follow (i.e., current waveform distortion). We will later see how some current controllers are inherently immune to dead-time induced distortion, while others are not.

We cannot end this discussion of dead-times without adding that, motivated by the considerations above, several studies have been presented that deal with their *compensation*. Both off-line, or feed-forward, techniques and closed-loop arrangements have been proposed to mitigate the problem. The interested reader can find very detailed discussions of these topics in technical papers such as, for instance, [2] and [3].

2.2 LOW-LEVEL CONTROL OF THE VOLTAGE SOURCE INVERTER: PWM MODULATION

The definition of a suitable modulation law represents the first step in any converter control design. Several modulation techniques have been developed for switch mode power supplies: the most successful, for the VSI case, is undoubtedly the pulse width modulation (PWM). Compared to other approaches, such as pulse density modulation or pulse frequency modulation, the PWM offers significant advantages, for instance in terms of ease of implementation, constant frequency inverter operation, immediate demodulation by means of simple low-pass filters. The analog implementation of PWM, also known as *naturally sampled* PWM, is indeed extremely easy, requiring, in principle, only the generation of a suitable carrier (typically a triangular or sawtooth waveform) and the use of an analog comparator. A simple PWM circuit is shown in Fig. 2.3.

2.2.1 Analog PWM: the Naturally Sampled Implementation

Considering the circuit and what has been explained in Section 2.1, it is easy to see that, as a result of the analog comparator and driving circuitry operation, a square-wave voltage V_{OC} will be applied to the load, with constant frequency $f_S = 1/T_S$, T_S being the period of the carrier signal $c(t)$, and variable duty-cycle d. This is implicitly defined, again from Fig. 2.3, as the ratio between the time duration of the $+V_{DC}$ voltage application period and the duration of the whole modulation period, T_S. Finally, Fig. 2.3 allows us to see the relation between duty-cycle and the average value (in the modulation period) of the load voltage, which is calculated in Aside 1.

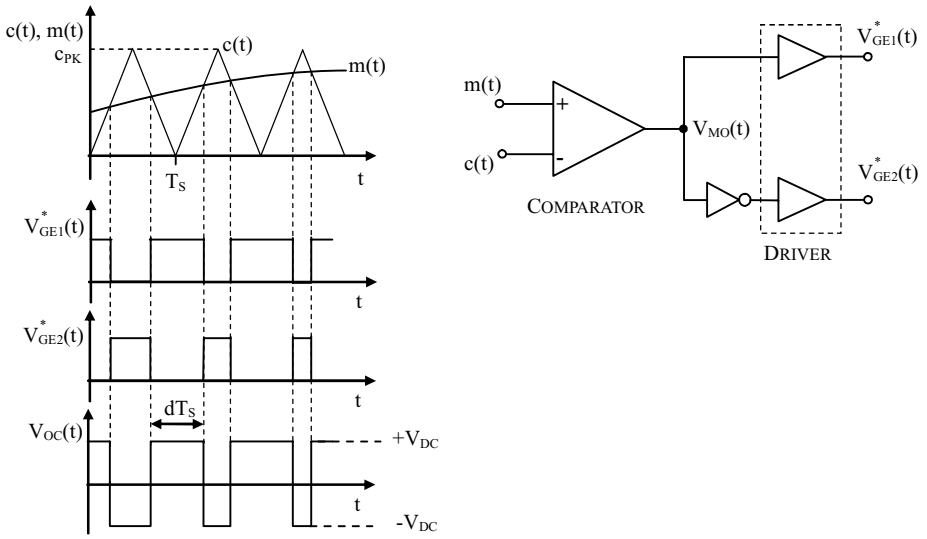

FIGURE 2.3: Analog implementation of a PWM modulator. The analog comparator determines the state of the switches by comparing the carrier signal $c(t)$ and the modulating signal $m(t)$. The figure shows the logic state of each switch and the resulting inverter voltage. No dead-time is considered.

It is now interesting to explicitly relate the signal $m(t)$ to the resulting PWM duty-cycle. Simple calculations show that, in each modulation period, where a constant m is assumed, the following equation holds:

$$\frac{m}{d\,T_S} = \frac{c_{PK}}{T_S} \quad \Leftrightarrow \quad d = \frac{m}{c_{pk}}. \tag{2.3}$$

If we now assume that the modulating signal changes slowly along time, with respect to the carrier signal, i.e., the upper limit of the $m(t)$ bandwidth is well below $1/T_S$, we can still consider the result (2.3) correct. This means that, in the hypothesis of a limited bandwidth $m(t)$, the information carried by this signal is transferred by the PWM process to the duty-cycle, which will change slowly along time following the $m(t)$ evolution. The duty-cycle, in turn, is transferred to the load voltage waveform by the power converter. The slow variations of the load voltage average value will therefore copy those of the signal $m(t)$.

The simplified discussion above may be replaced by a more mathematically sound approach, which an interested reader can find in power electronics textbooks such as [4], [5], and [6]. However, this approach would basically show that the frequency content, i.e., the spectrum, of the modulating signal $m(t)$ is shifted along frequency by the PWM process, and is replicated around all integer multiples of the carrier frequency. This implies that, as long as the spectrum of the signal $m(t)$ has a limited bandwidth with an upper limit well below the carrier frequency,

signal demodulation, i.e., the reconstruction of the signal $m(t)$ spectrum from the signal $V_{OC}(t)$, with associated power amplification, can be easily achieved by low-pass filtering $V_{OC}(t)$. In the case of power converters, like the one we are considering here, the low-pass filter is actually represented by the load itself.

Referring again to Fig. 2.1 and to Aside 1, it is possible to see that the transfer function between the inverter voltage V_{OC} and load current I_O indeed presents a single-pole low-pass filter frequency response. The pole is located at an angular frequency that is equal to the ratio between the load resistance R_S and the load inductance L_S. Because of that, we can assume that, if the load *time constant*, L_S/R_S, is designed to be much higher than the modulation period T_S, the load current I_O average in the modulation period will precisely follow the trajectory determined by the signal $m(t)$. This is the situation described in Fig. 2.4. It is worth noting that, while the average current is suitably sinusoidal, the instantaneous current waveform is characterized by a residual switching noise, the current *ripple*. This is a side effect determined by the nonideal filtering of high-order modulation harmonics, given by the load low-pass characteristics.

Aside 1. VSI State Space Model

The VSI represented in Fig. 2.1 can be described in the state space by the following equations:

$$\begin{cases} \dot{x} = Ax + Bu \\ y = Cx + Du \end{cases}, \qquad (A1.1)$$

where $x = [I_O]$ is the state vector, $u = [V_{OC}, E_S]^T$ is the input vector, and $y = [I_O]$ is the output variable. In this very simple case, the state and output vectors have unity size, but, in the general case, higher sizes can be required to correctly model the converter and its load. Direct circuit inspection yields

$$A = [-R_S/L_S], \quad B = [1/L_S, -1/L_S], \quad C = [1], \quad D = [0, 0]. \qquad (A1.2)$$

Based on this model and using Laplace transformation, the transfer function between the inverter voltage V_{OC} and the output current I_O, $G_{I_O V_{OC}}$ can be found to be

$$G_{I_O V_{OC}}(s) = C \cdot (sI - A)^{-1} \cdot B_{11} = \frac{1}{R_S} \cdot \frac{1}{1 + s\dfrac{L_S}{R_S}}. \qquad (A1.3)$$

The transfer function (A1.3) relates variations of the inverter voltage V_{OC} to the consequent variations of the output current I_O. The relation has been derived under no restrictive hypothesis, meaning that it has a general validity. In particular, (A1.3) can be used to relate variations of the *average* values of V_{OC} and I_O, where by average of any given variable v we

mean the following quantity:

$$\overline{v}(t) = \frac{1}{T_S}\int_{t}^{t+T_s} v(\tau)d\tau, \qquad (A1.4)$$

where T_S is our observation and averaging interval. In the particular case of PWM control, the definition (A1.4) is well posed once the averaging period T_S is taken equal to the modulation period.

Considering now the input variable V_{OC}, we can immediately calculate its average value as a function of the PWM duty-cycle. This turns out to be equal to

$$\begin{aligned}\overline{V}_{OC}(t) &= \frac{1}{T_s}\int_{t}^{t+T_s} V_{OC}(\tau)\mathrm{d}\tau \\ &= \frac{1}{T_S}(T_S \cdot V_{DC} \cdot d(t) - V_{DC}(1-d(t)) \cdot T_S) = V_{DC}(2d(t)-1), \qquad (A1.5)\end{aligned}$$

where $d(t)$ is the duty-cycle, as defined in Section 2.2. We can now easily calculate the relation between variations of the duty-cycle d and variation of $\overline{V}_{OC}$. Perturbation of (A1.5) yields

$$\frac{\partial \overline{V}_{OC}}{\partial d} = 2V_{DC}, \qquad (A1.6)$$

where V_{DC} is assumed to be constant. In the assumption of small perturbations around any given operating point, the transfer function between duty-cycle and load current can be obtained substituting (A1.6) into (A1.3). We find

$$G(s) = \frac{\tilde{I}_O}{\tilde{d}}(s) = \frac{2V_{DC}}{R_S} \cdot \frac{1}{1 + s\dfrac{L_S}{R_S}}, \qquad (A1.7)$$

where $\tilde{I}_O$ and $\tilde{d}$ represent small perturbations of the variables I_O and d around any selected operating point. The result (A1.7) can be used in the design of current regulators.

In general, we will see how the removal of such switching noise from the control signals, that is essential for the proper operation of any digital controller, is fairly easy to achieve, even without using further low-pass filters in the control loop.

In the following sections, we will see how a current controller can be designed. The purpose of the current controller will be to automatically generate the signal $m(t)$ based on the desired load current trajectory, which will be designated as the *current reference* signal.

Before we move to digital PWM and current control design, there is a final issue to consider, related to the dynamic response of the PWM modulator [7–11]. Considering the circuit in Fig. 2.3., it is possible to see that a sudden change in the modulating signal amplitude always

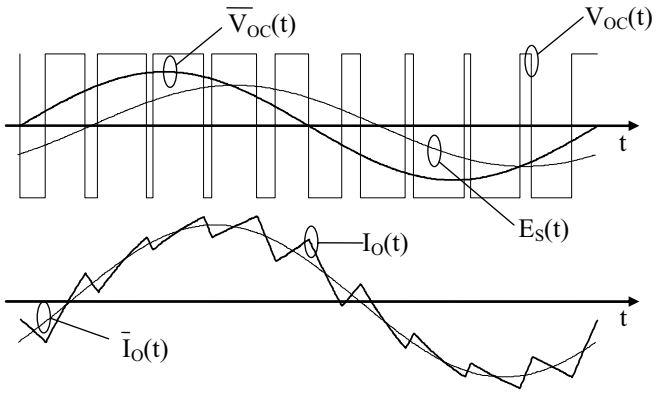

FIGURE 2.4: Example of PWM application to the VSI of Fig. 2.1. The instantaneous load voltage $V_{OC}(t)$ is demodulated by the low-pass filter action of the inverter load. The resulting load current $I_O(t)$ has an average value, $\overline{I}_O(t)$, whose waveform is determined by the instantaneous voltage average value $\overline{V}_{OC}(t)$ (and by the load voltage E_S, here assumed to be sinusoidal).

implies an immediate, i.e., within the current modulation period, adjustment of the resulting duty-cycle. This means that the analog implementation of PWM guarantees the minimum delay between modulating signal and duty-cycle. This intuitive representation of the modulator operation can be actually corroborated by a more formal mathematical analysis. Indeed, the derivation of an equivalent modulator transfer function, in magnitude and phase, has been studied and obtained since the early 1980s. The modulator transfer function has been determined using small-signal approximations [7], where the modulating signal $m(t)$ is decomposed into a dc component M and a small-signal perturbation $\tilde{m}$ (i.e., $m(t) = M + \tilde{m}$). Under these assumptions, in [7], the author demonstrates that the phase lag of the naturally sampled modulator is actually zero, concluding that the analog PWM modulator delay can always be considered negligible. Quite differently, we will see in the following section how the discrete time or digital implementations of the pulse width modulator [8], which necessarily imply the introduction of sample-and-hold effects, determine an appreciable, not at all negligible, delay effect.

2.2.2 Digital PWM: the Uniformly Sampled Implementation

The basic principles described in Section 2.2.1 apply also to the digital implementation of the PWM modulator. In the more direct implementation, also known as "uniformly sampled PWM," each analog block is replaced by a digital one. The analog comparator function is replaced by a digital comparator, the carrier generator is replaced by a binary counter, and so forth. We can see the typical hardware organization of a digital PWM, of the type we can find inside several microcontrollers and digital signal processors, either as a dedicated peripheral unit or as a special programmable function of the general purpose timer, in Fig. 2.5.

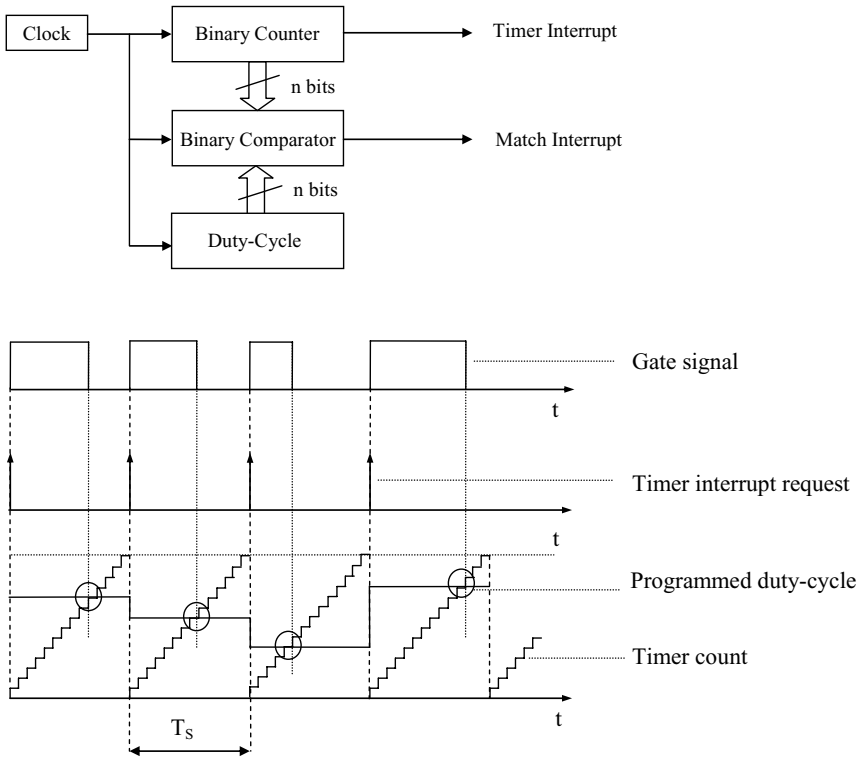

FIGURE 2.5: Simplified organization of a digital pulse width modulator. The binary comparator triggers an interrupt request for the microprocessor any time the binary counter value is equal to the programmed duty-cycle (match condition). At the beginning of the counting period, the gate signal is set to high and goes low at the match condition occurrence.

The principle of operation is straightforward: the counter is incremented at every clock pulse; any time the binary counter value is equal to the programmed duty-cycle (match condition), the binary comparator triggers an interrupt to the microprocessor and, at the same time, sets the gate signal low. The gate signal is set high at the beginning of each counting (i.e., modulation) period, where another interrupt is typically generated for synchronization purposes. The counter and comparator have a given number of bits, n, which is often 16, but can be as low as 8, in case a very simple microcontroller is used. Actually, depending on the ratio between the durations of the modulation period and the counter clock period, a lower number of bits, N_e, could be available to represent the duty-cycle. The parameter N_e is also important to determine the duty-cycle quantization step, which can have a significant impact on the generation of limit cycles, as we will explain in the following chapters. For now it is enough to say that, with this type of modulator, the number N_e of bits needed to represent the duty-cycle is given by the

following relation:

$$N_e = \text{floor}\left[\frac{\log_{10}\left(\frac{f_{clock}}{f_S}\right)}{\log_{10} 2}\right] + 1, \tag{2.4}$$

where f_{clock} is the modulator clock frequency, $f_S = 1/T_S$ is the desired modulation frequency, and the *floor* function calculates the integer part of its argument. Typical maximum values for f_{clock} are in the few tens of MHz range, while modulation frequencies can be as high as a few hundreds of kHz. Therefore, when the desired modulation period is short, the number of bits, N_e, given by (2.4) will be much lower than the number bits, n, available in the comparator and counter circuits, unless a very high clock frequency is possible.

Fig. 2.5. allows us to discuss another interesting issue about digital PWM, that is the dynamic response delay of the modulator. In the considered case, it is immediate to see that the modulating signal update is performed only at the beginning of each modulation period. We can model this mode of operation as a *sample and hold* effect. We can observe that, if we neglect the digital counter and binary comparator operation assuming infinite resolution, the digital modulator works exactly as an analog one, where the modulating signal $m(t)$ is sampled at the beginning of each modulation period and the sampled value kept constant for the whole period.

It is now evident that, because of the sample and hold effect, the response of the modulator to any disturbance, e.g., to one requiring a step change in the programmed duty-cycle value, can take place only during the modulation period *following* the one where the disturbance actually takes place. Note that this delay effect amounts to a dramatic difference with respect to the analog modulator implementation, where the response could take place already during the *current* modulation period, i.e., with negligible delay. Therefore, even if our signal processing were fully analog, without any calculation or sampling delay, passing from an analog to a digital PWM implementation would imply an increase in the system response delay. We will see how this simple fact implies a significant reduction of the system's phase margin with respect to the analog case, which often compels the designer to adopt a more conservative regulator design and to accept a lower closed loop system bandwidth.

Since these issues can be considered fundamental for all the following discussions, from the intuitive considerations reported above, we can now move to a precise small-signal Laplace-domain analysis, which might be very useful for a clear understanding of control limitations and delay effects implied by the uniformly sampled PWM.

An equivalent model of the uniformly sampled PWM process is represented in Fig. 2.6(a). As can be seen, the schematic diagram adopts the typical continuous time model of a sampled

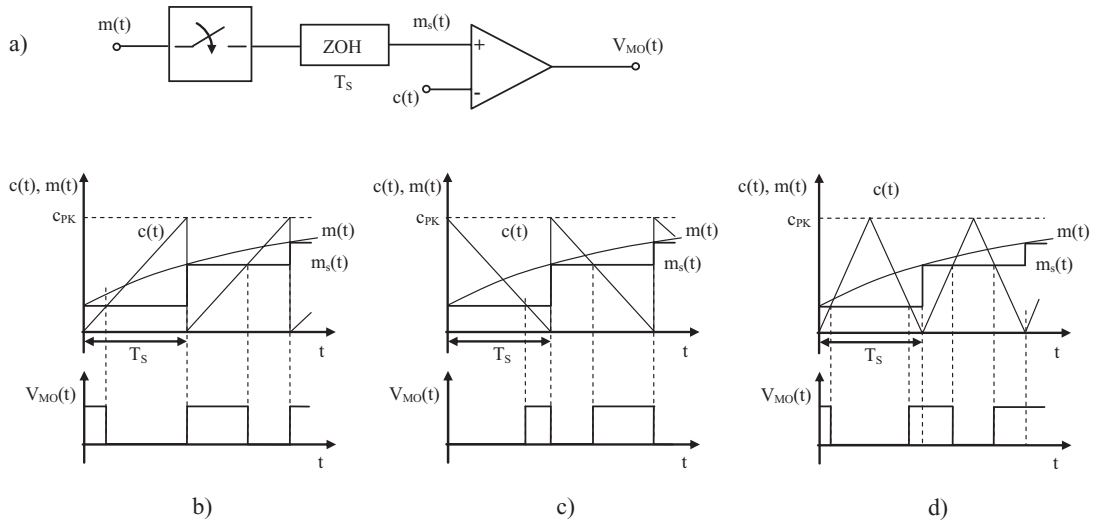

FIGURE 2.6: Uniformly sampled PWM with single update mode: (a) general block diagram, (b) trailing-edge modulation, (c) leading-edge modulation, (d) triangular carrier modulation.

data system, where an ideal sampler is followed by a zero-order hold (ZOH). The quantization effect that is associated, in the physical implementation of the modulator of Fig. 2.5., with the digital counter and binary comparator operation, is neglected, being irrelevant from the dynamic response delay standpoint. Accordingly, in the model of Fig. 2.6(a), after the modulating signal $m(t)$ is processed by the ZOH, the PWM waveform is generated by an ideal analog comparator, which compares the ZOH output signal $m_s(t)$ and the carrier waveform $c(t)$.

Depending on $c(t)$, several different uniformly sampled pulse-width modulators can be obtained. For example, in Fig. 2.6(b) a trailing-edge modulation is depicted, where the update of the modulating signal is performed at the beginning of the modulation period. Note that this is an exactly equivalent representation of the modulator organization of Fig. 2.5. In a small-signal approximation, it is possible to find that the transfer function between the modulating signal $m(t)$ and the output of the comparator $V_{MO}(t)$ is given by [7]

$$\text{PWM}(s) = \frac{V_{MO}(s)}{M(s)} = \frac{e^{-s\,DT_S}}{c_{PK}}, \tag{2.5}$$

where $V_{MO}(s)$ and $M(s)$ represent the Laplace transforms of $V_{MO}(t)$ and $m(t)$, respectively. Therefore, the uniformly sampled modulator presents a delay whose value is proportional to the steady-state duty-cycle D.

In more general terms, the delay introduced by the PWM modulator represents the time distance between the modulating signal $m(t)$ sampling instant and the instant when the output pulse is completely determined (i.e., when $m_s(t)$ intersects $c(t)$ in Fig. 2.6). The result (2.5)

has been extended also to other types of modulator organizations (trailing edge, triangular carrier, etc.) [8]. For example, for the leading-edge modulation represented in Fig. 2.6(c), the small-signal modulator transfer function turns out to be

$$\mathrm{PWM}(s) = \frac{V_{\mathrm{MO}}(s)}{M(s)} = \frac{\mathrm{e}^{-s(1-D)T_S}}{c_{\mathrm{PK}}}, \tag{2.6}$$

while, for the triangular carrier modulation, where the sampling of the modulating signal is done in the middle of the switch *on* period (Fig. 2.6(d)), it is

$$\mathrm{PWM}(s) = \frac{V_{\mathrm{MO}}(s)}{M(s)} = \frac{1}{2c_{\mathrm{PK}}}\left(\mathrm{e}^{-s(1-D)\frac{T_S}{2}} + \mathrm{e}^{-s(1+D)\frac{T_S}{2}}\right). \tag{2.7}$$

Finally, the case of the triangular carrier modulator, where the sampling of the modulating signal is done in the middle of the switch *off* period, can be simply derived from (2.7) substituting D with D', being $D' = 1 - D$.

2.2.3 Single Update and Double Update PWM Mode

To partially compensate for the increased delay of the uniformly sampled PWM, the double update mode of operation is often available in several microcontrollers and DSPs. In this mode, the duty-cycle update is allowed at the beginning and at the half of the modulation period. Consequently, in each modulation period, the match condition between counter and duty-cycle registers is checked twice, at first during the run-up phase, then during the run-down phase.

In the occurrence of a match, the state of the gate signal is toggled. As can be seen in Fig. 2.7, the result of this mode of operation is a stream of gate pulses that are symmetrically allocated within the modulation period, at least in the absence of any perturbation. Interrupt

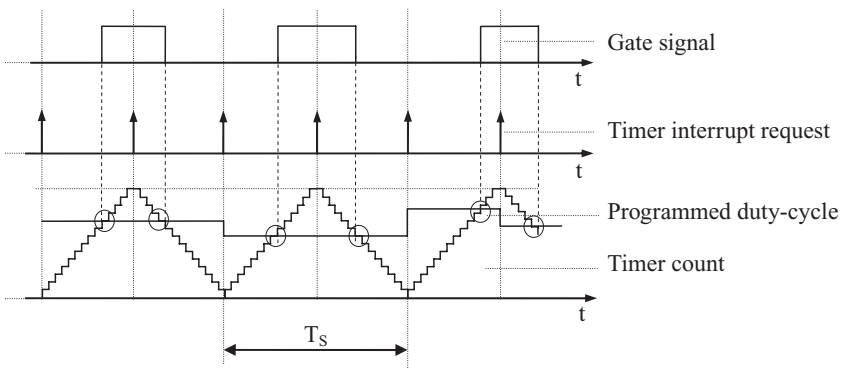

FIGURE 2.7: Double update mode of operation for a digital pulse width modulator. Duty-cycle update is allowed at the beginning and at a half of the modulation period. Note that the gate pulses are now symmetrically allocated within the modulation period (in steady state).

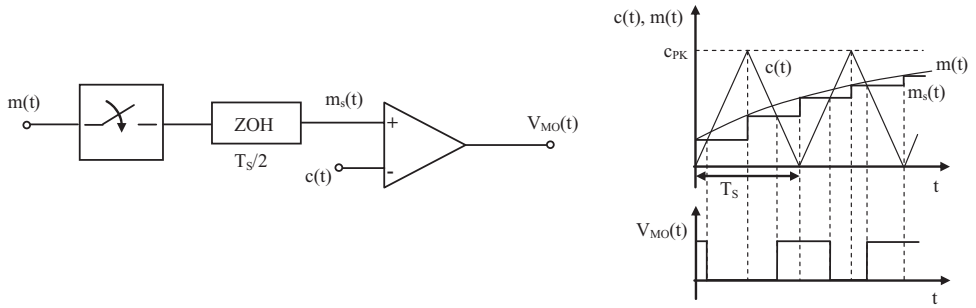

FIGURE 2.8: Model of the uniformly sampled PWM with double update.

requests are generated by the timer at the beginning and at the half of the modulation period, to allow proper synchronization with other control functions, e.g., with the sampling process.

It is also evident from Fig. 2.7 that, in the occurrence of a perturbation, the modulator response delay is reduced, with respect to the single update case because, now, the duty-cycle update can be performed at the occurrence of each half period interrupt request. In this case though, an asymmetric pulse is generated, but symmetry is restored immediately afterward, so that its temporary loss is of little consequence.

Maybe less evident is the drawback of this operating mode: given the number of bits, N_e, needed to represent the duty-cycle and the clock frequency f_{clock}, the switching period has to be doubled to contain both the run-up and run-down phases. Of course, it is possible to maintain the same modulation frequency of the single update case, but, in order to do that, either the clock frequency needs to be doubled or the number of bits needs to be reduced by 1.

Following the reasoning reported in the previous section, we can derive an exact, continuous time equivalent model also of the digital PWM with double duty-cycle update. A representation of this model is shown in Fig. 2.8. Simple calculations show that the small-signal modulator transfer function is, in this case, given by [8]

$$\mathrm{PWM}(s) = \frac{V_{\mathrm{MO}}(s)}{M(s)} = \frac{1}{2c_{\mathrm{PK}}}\left(\mathrm{e}^{-sD\frac{T_s}{2}} + \mathrm{e}^{-s(1-D)\frac{T_s}{2}}\right). \tag{2.8}$$

It is interesting to compare the modulator phase lag for the single and double update modes of operation. In (2.7), we find $\arg(\mathrm{PWM}(j\omega)) = -\omega T_s/2$, while, in (2.8), $\arg(\mathrm{PWM}(j\omega)) = -\omega T_s/4$, so that, as it could be expected, the modulator phase lag is reduced by one half in the double update mode. This property can give significant benefits, in terms of the achievable speed of response, for any controller built on top of the digital modulator.

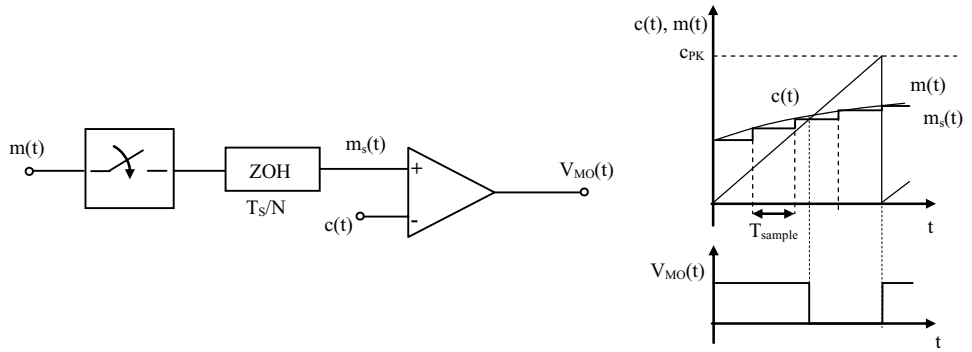

FIGURE 2.9: Multi-sampled PWM.

2.2.4 Minimization of Modulator Delay: a Motivation for Multisampling

In the more recent studies concerning digital control of power converters the key role played by the modulator delay in limiting the achievable control bandwidth has been very well clarified. A different approach has been suggested, which exploits the possibility of sampling control variables, and consequently adjusting the duty-cycle, several times (e.g., 4, 8, 16 times) within the modulation period. The purpose of this is to reduce the PWM response delay and increase the system phase margin, extending the benefits seen for the double update in comparison with the single update mode.

In order to evaluate the modulator phase lag, let us consider the system shown in Fig. 2.9: the modulating signal is sampled N times during the switching period, so that the sampling time is now $T_{\text{sample}} = T_S/N$; moreover, in order to fully exploit the advantages of the multiple-sampling technique, the control algorithm updates the control signal $m(t)$ at each sampling event. In the multisampled case, the PWM is modeled with an equivalent system similar to that shown in Fig. 2.6, with the only difference that the input signal $m_s(t)$ is now a sequence of variable amplitude pulses, updated with frequency $f_{\text{sample}} = N \cdot f_S$. Accordingly, the hold time of the ZOH is now $T_{\text{hold}} = T_{\text{sample}} = T_S/N$. It can be shown that the low-frequency, small-signal behavior of the multisampled digital PWM is again that of a pure delay,

$$\text{PWM}(s) = \frac{1}{c_{\text{PK}}} e^{-st_d}, \tag{2.9}$$

but the equivalent delay time is now given by

$$t_d = DT_S - \frac{\text{floor}(ND)}{N} T_S, \tag{2.10}$$

where floor(ND) denotes the greatest integer which does not exceed $N \cdot D$. Equations (2.9) and (2.10) can be derived analytically with methods similar to those used in [7], for the uniformly

sampled modulator, and applying a small-signal approximation. The first term $D \cdot T_S$ in (2.10) is the same delay as found in (2.5), and does not depend on the multisampling factor N. The second term takes into account the multiple sampling effect, which is primarily that of reducing the equivalent delay time, and thus the total phase lag introduced by the PWM. Moreover, from (2.10) we can infer that, as N tends to infinity, the equivalent delay time tends to zero. The result is obvious, since when N is high the multisampled PWM approaches the naturally sampled modulator, where the phase lag is known to be zero.

The main drawback of such an approach is represented by the need for proper filtering of the switching noise from the control signals, that is, instead, straightforward with the single or double update mode. Filtering the control signals may impair the system phase margin, reducing the advantage of the multisampled strategy. We will come back to this issue in Chapter 3, where we will open the discussion of digital controllers. For now, it will be enough to say that some research is in progress around the world to find means to get the needed filtering without worsening the system stability margin, for example using sophisticated estimation techniques. One last remark about multisampling refers to the hardware required for the implementation. This is significantly different from what can be considered the standard PWM organization, available with off the shelf microcontrollers and DSPs, and calls for other solutions, e.g., the use of hardware programmable digital control circuits, like those based on field programmable gate arrays (FPGAs).

2.3 ANALOG CONTROL APPROACHES

We begin here to deal with the control problem this book is all about. In order to better appreciate the merits and limitations of the digital approach, we will now briefly discuss two possible analog implementations of a current control loop: the PI linear controller and the nonlinear hysteresis controller. We refer to our test case, as represented in Fig. 2.1, but in order to make some explicit calculations, we will take into account the parameters listed in Table 2.1.

In this example we suppose that the purpose of the VSI is to deliver a given amount of output power P_O to the load, which is represented by the voltage source E_S. The resistor R_S may represent the lossy elements of the load and of the inverter inductor. What we are discussing can be thought as the typical ac motor drive application, where a sinusoidal current of suitable amplitude and given frequency, f_O, must be generated on each motor phase. Consequently, we have also taken into account the presence of a current transducer, whose gain, G_{TI}, is given in Table 2.1, and that may be in practice implemented by a Hall sensor.

For the controller implementation, we can assume that one of the average current mode control integrated circuits, available on the market, is used. This will generally include all the needed functions, from error amplification and loop compensation to PWM modulation. Of course, to keep the discussion simple, the presence of additional signal scale factors, for example due to internal voltage dividers, is not taken into account. Also, the PWM parameters reported

TABLE 2.1: Half-Bridge Inverter Parameters

Rated output power, P_O	1000 (W)
Phase inductance, L_S	1.5 (mH)
Phase resistance, R_S	1 (Ω)
Phase load voltage, E_S	100 (V_{RMS})
Load frequency, f_O	125 (Hz)
DC link voltage, V_{DC}	250 (V)
Switching frequency, f_S	50 (kHz)
PWM carrier peak, c_{PK}	4 (V)
Current transducer gain, G_{TI}	0.1 (V/A)

in Table 2.1, although realistic, do not necessarily represent those of any particular integrated controller.

2.3.1 Linear Current Control: PI Solution

Fig. 2.10 shows the control loop block diagram, where all the components are represented by their respective transfer functions or gains. In particular, the controller block is represented by the typical proportional integral regulator structure, whose parameters K_P and K_I will be determined in the following. The output of the regulator represents the modulating signal that drives the pulse width modulator. This has been modeled as the cascade combination of two separate blocks: the first one is the modulator static gain, as given by (2.3), the second one is actually a first-order Padé approximation of its delay, considered equal to a half of the duration of the modulation period.

This choice deserves some clarification, since we have previously assessed the delay effect of an analog PWM to be negligible. The point is that, for reasons that will be fully motivated

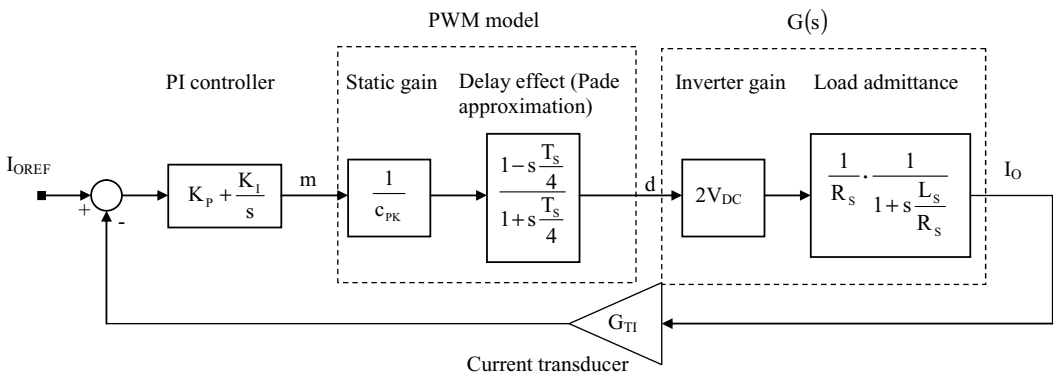

FIGURE 2.10: Control loop block diagram.

in Chapter 3, we are here considering the modulator as if it was *digitally* implemented, i.e., characterized by the sample and hold delay that we have previously described. From Section 2.2, we know that the equivalent model of the digital modulator can be given by (2.5), (2.6), or, possibly, (2.7). The proper characterization of these models is a little complicated. For this reason, in Fig. 2.10, we consider the response delay of the digital PWM to be, on average, equal to a half of the modulation period and we model this average delay with its first-order Padé approximation. In Chapter 3, we will clearly account for this approximation and show that this is actually not penalizing.

Considering now the inverter and load models, we see that they are exactly based on the analysis presented in Aside 1. Finally, to fully replicate a typical implementation, a transducer gain is taken into account. Additional filters, which are normally adopted to clean the transducer signal from residual switching noise, are instead not taken into account, in favor of a more essential presentation. Their transfer functions can be easily cascaded to the transducer block gain if needed.

Given the block diagram of Fig. 2.10, the design of the PI compensator is straightforward. However, for the sake of completeness, we present the simple design procedure in Aside 2. Once the proper K_P and K_I values are determined, we still may want to check the system dynamic behavior and verify if a stable closed loop controller with the desired speed of response has been obtained.

In order to do that, before developing any converter prototype, it is very convenient to use one of the several dynamic system software simulators available. The simulation of the VSI depicted in Fig. 2.1, together with its current controller, gives the results described by Fig. 2.11. In particular, Fig. 2.11(a) shows the response of the closed loop system to a step

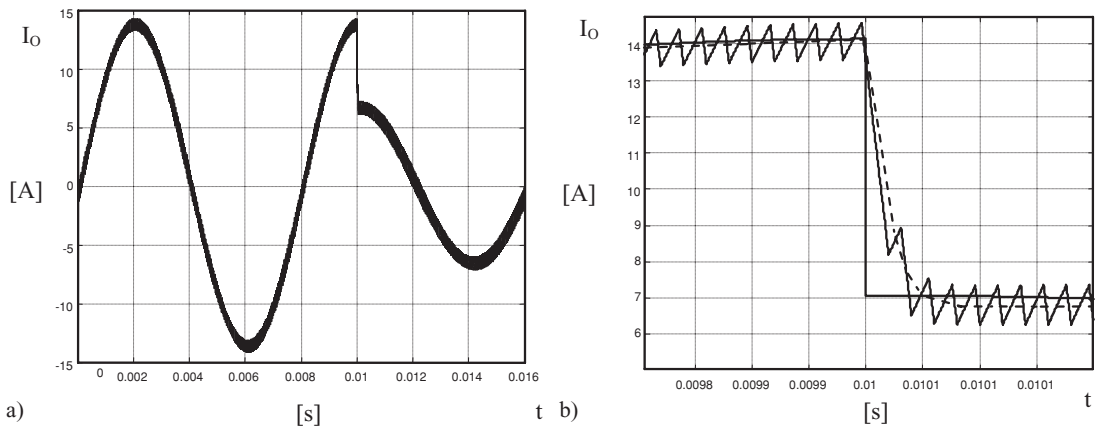

FIGURE 2.11: Simulation of the VSI depicted in Fig. 2.1 with the controller designed according to the procedure reported in Aside 2. The depicted variable is the VSI output current I_O. (a) Controller response to a step reference amplitude change. (b) Details of the previous figure.

change in the I_{OREF} current reference amplitude. It is possible to see that the closed loop plant is properly controlled, with a sufficiently high phase margin not to incur in oscillations after the transient. Fig. 2.11(b) shows the details of the transient response: the controller reaches the new steady-state condition in three modulation periods, exhibiting no overshoots.

It is worth noting that an anti wind-up action is included in the PI controller to prevent deep saturation of the integral controller during transients. One closing remark in Fig. 2.11(b) is due: an appreciable, albeit relatively small, steady-state tracking error between the reference signal (continuous line) and the instantaneous current average value (i.e., once the current ripple is filtered, dashed line), is visible both before and after the transient. This represents the residual tracking error of the current controller. As any other controller including an integral action, our PI is able to guarantee zero steady-state tracking error only for dc signals. In the case of an ac reference signal, as that of Fig. 2.11(b), a residual error will always be found, whose amplitude depends on the closed loop system gain and phase at the particular reference signal frequency.

Aside 2. Design of the Analog PI Current Controller

At first, we want to determine the open loop gain for the block diagram of Fig. 2.10. This is given by the cascade connection of all blocks. We find

$$G_{\mathrm{OL}}(s) = \left(K_{\mathrm{P}} + \frac{K_{\mathrm{I}}}{s}\right) \frac{2V_{\mathrm{DC}}}{c_{\mathrm{PK}}} \frac{1 - s\dfrac{T_{\mathrm{S}}}{4}}{1 + s\dfrac{T_{\mathrm{S}}}{4}} \frac{G_{\mathrm{TI}}}{R_{\mathrm{S}}} \frac{1}{1 + s\dfrac{L_{\mathrm{S}}}{R_{\mathrm{S}}}}. \qquad (A2.1)$$

The regulator design is typically driven by specifications concerning the required closed loop *speed of response* or, equivalently, the maximum allowed *tracking error* with respect to the reference signal. These specifications can be turned into equivalent specifications for the closed loop bandwidth and phase margin. To give an example, we suppose that, for our current controller, a closed loop bandwidth, f_{CL}, equal to about one sixth of the switching frequency f_{S} is required, to be achieved with, at least, a 60° phase margin, phm.

We therefore have to determine the parameters K_{P} and K_{I} so as to guarantee the compliance to these requirements.

To rapidly get an estimation of the searched values, we suppose that we can approximate the open loop gain at the crossover angular frequency, i.e., at $\omega = \omega_{\mathrm{CL}} = 2\pi f_{\mathrm{CL}}$, with the following expression:

$$G_{\mathrm{OL}}(j\omega_{\mathrm{CL}}) \cong K_{\mathrm{P}} \frac{2V_{\mathrm{DC}}}{c_{\mathrm{PK}}} \frac{1 - j\omega_{\mathrm{CL}}\dfrac{T_{\mathrm{S}}}{4}}{1 + j\omega_{\mathrm{CL}}\dfrac{T_{\mathrm{S}}}{4}} \frac{G_{\mathrm{TI}}}{R_{\mathrm{S}}} \frac{1}{1 + j\omega_{\mathrm{CL}}\dfrac{L_{\mathrm{S}}}{R_{\mathrm{S}}}}, \qquad (A2.2)$$

which, in principle, will be a good approximation as long as $K_I \ll \omega_{CL} K_P$ (to be verified later). Imposing now the magnitude of (A2.2) to be equal to one at the desired crossover frequency, we get

$$K_P = \frac{c_{PK}}{2V_{DC}} \frac{R_S}{G_{TI}} \sqrt{1 + \left(\omega_{CL} \frac{L_S}{R_S}\right)^2}. \tag{A2.3}$$

The parameter K_I can then be calculated considering the open loop phase margin and imposing that to be equal to phm. We find from (A2.1)

$$-180^\circ + phm = -90^\circ - 2\tan^{-1}\left(\omega_{CL}\frac{T_S}{4}\right) - \tan^{-1}\left(\omega_{CL}\frac{L_S}{R_S}\right) + \tan^{-1}\left(\omega_{CL}\frac{K_P}{K_I}\right), \tag{A2.4}$$

which yields

$$K_I = \frac{\omega_{CL} K_P}{\tan\left(-90^\circ + phm + 2\tan^{-1}\left(\omega_{CL}\frac{T_S}{4}\right) + \tan^{-1}\left(\omega_{CL}\frac{L_S}{R_S}\right)\right)}. \tag{A2.5}$$

Note that (A2.5) is exact; only the K_P value is obtained through an approximation. Considering the parameters listed in Table 2.1 and $\omega_{CL} = 2\pi f_S/6 \cong 52.4$ krad s^{-1}, we can immediately find the following values:

$$\begin{aligned} K_P &= 6.284 \\ K_I &= 1.802 \times 10^4 \text{ (rad s}^{-1}\text{)}. \end{aligned}$$

It is easy to verify that the condition $K_I \ll \omega_{CL} K_P$ is reasonably met by this solution. Nevertheless, in order to explicitly evaluate the quality of the approximated solution, we can compare the values above with the solutions of the *exact* design equations. We practically need to solve the following system of equations:

$$\begin{cases} \dfrac{K_I}{K_P} = \dfrac{\omega_{CL}}{\tan\left(-90^\circ + phm + 2\tan^{-1}\left(\omega_{CL}\dfrac{T_S}{4}\right) + \tan^{-1}\left(\omega_{CL}\dfrac{L_S}{R_S}\right)\right)} \\ K_P = \dfrac{c_{PK}}{2V_{DC}} \dfrac{R_S}{G_{TI}} \sqrt{\dfrac{1 + \left(\omega_{CL}\dfrac{L_S}{R_S}\right)^2}{1 + \left(\dfrac{1}{\omega_{CL}}\dfrac{K_I}{K_P}\right)^2}} \end{cases} \tag{A2.6}$$

The solution yields $K_P = 6.274$, $K_I = 1.8 \times 10^4$ (rad s^{-1}).

As can be seen, the exact values are very close to those found by the approximated procedure above. This happens in the large majority of practical cases, so that (A2.3) and (A2.5) can be very often directly used.

As a final check of the design, we now present the Bode plot of the open loop gain, where the desired crossover frequency and phase margin can be read.

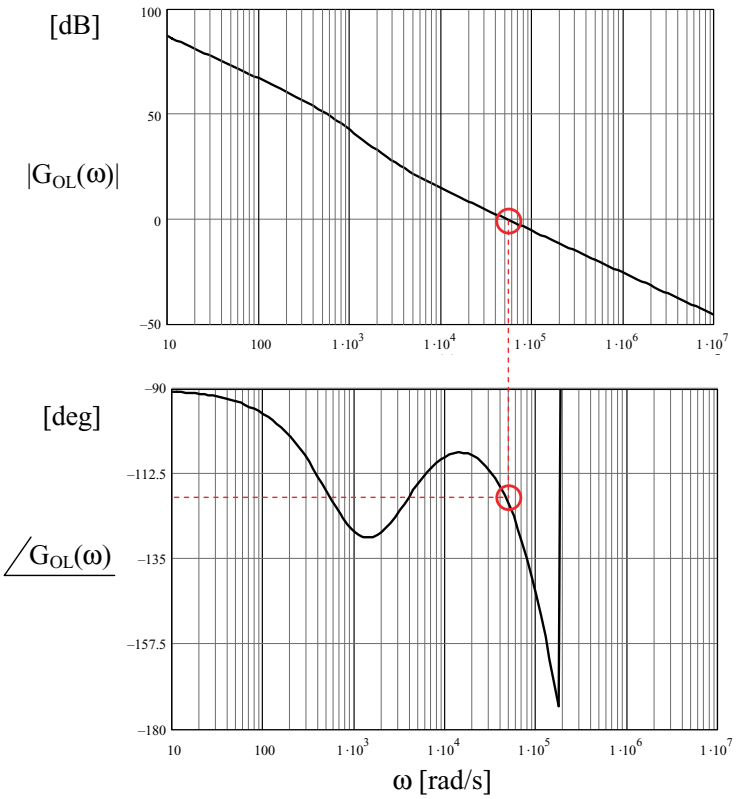

FIGURE A2.1: Bode plot of the open loop gain.

An interesting advantage of the PI current controller usage is the automatic compensation of dead-time induced current distortion. Referring to our brief discussion in Section 2.1.4, it is possible to see how, from the current controller standpoint, the dead-time effect can be equivalently seen as a disturbance signal that sums with the average inverter output voltage, generated by an ideal (i.e., with no dead-times) pulse width modulator. If the dead-time duration can be considered constant, as is often the case, the disturbance signal is very close to a square

wave, whose amplitude is directly proportional to the dc link voltage and to the dead-time duration and inversely proportional to the switching period duration (2.2). Compared to the output current signal, this square wave has the same frequency and *opposite* phase. We know that the PI controller guarantees a significantly higher than unity open loop gain at the current reference frequency (see the Bode plot in Aside 2), which is typically maintained for at least a decade above. As a result, the controller will reject the disturbance quite effectively: only minor crossover effects, due to an incomplete compensation of the higher order harmonics of the square wave, will be observable on the output current waveform.

2.3.2 Nonlinear Current Control: Hysteresis Control

The PI controller discussed above is not the only possible solution to provide the VSI of Fig. 2.1 with a closed loop current control. Other approaches are viable, among which the hysteresis current controller is the most successful. Even if we are not going to develop this topic in detail, we still would like to briefly describe the principles of this type of analog current controller, just not to give to the reader the wrong feeling that analog current control only amounts to PI regulators and PWM.

It is important to underline from the start that the hysteresis controller is a particular type of bang-bang nonlinear control and, as such, the dynamic response it is able to guarantee is extremely fast; actually it is the fastest possible for any VSI with given dc link voltage and output inductance. The basic reason for this is that the hysteresis controller does not require any modulator: the state of the converter switches is determined directly by comparing the instantaneous converter current with its reference. A typical hysteresis current controller is depicted in Fig. 2.12.

As can be seen, an analog comparator is fed by the instantaneous current error, and its output directly drives the converter switches. Thanks to the VSI topology and to the fact that the dc link V_{DC} voltage will always be higher than the output voltage E_S peak value, the current derivative sign will be positive any time the high-side switch is closed and negative any time the low-side switch is closed. This guarantees that the controller organization of Fig. 2.12 will maintain the converter output current always close to its reference. Under the limit condition of zero hysteresis bandwidth, the current error can be forced to zero as well: unfortunately this condition implies an infinite frequency for the switch commutations, which is, of course, not practical. In real-life implementations, the hysteresis bandwidth is kept sufficiently small to minimize the tracking error without implying too high switching frequencies. As a consequence, also the compensation of dead-time induced current distortion will be very good.

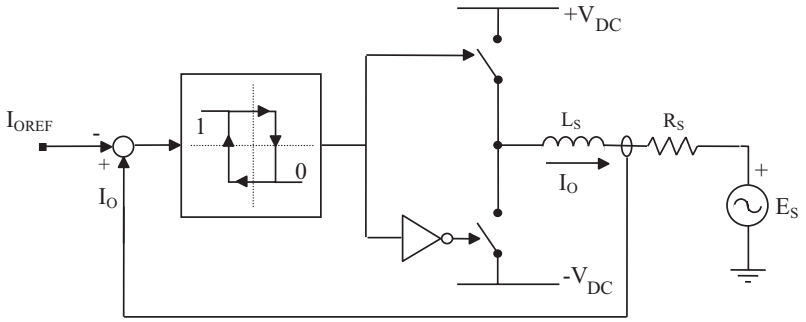

FIGURE 2.12: Hysteresis current control hardware organization.

What is even more important, in the case of any transient, which may bring the instantaneous current outside the hysteresis band, the controller will almost immediately close the right switch to bring the current back inside the band, thus minimizing the response delay and tracking error. Clearly, there is no linear controller that can be faster than this.

Nevertheless, the hysteresis current controller is not ubiquitously used in power electronics. That is because, despite its speed of response and high-quality reference tracking capabilities, this type of controller does have some drawbacks as well. The main is represented by a variable switching frequency. Indeed, any time the current reference is not constant the converter switching frequency will vary along the current reference period. The same holds in case the output voltage E_S is variable. The range of frequency variation can be very large, thus making the proper filtering of the high-frequency components of voltages and currents quite expensive. Moreover, in the VSI applications like controlled rectifiers or active filters, the injection of a variable frequency noise into the utility grid is not recommended, because unpredictable resonances with other connected loads could be triggered. To solve this and other problems a considerable research activity has been developed in the last few years. Different control solutions, which try to keep the benefits of the hysteresis controller and, for example, get a fixed switching frequency out of it, have been proposed. We are not going to deal with this advanced topics. However, the interested reader can find much useful information in technical papers such as [12] or [13].

REFERENCES

[1] ISOSMART™ Half Bridge Driver Chipset, IXBD4410/4411 Data sheet and Application note, © 2004, IXYS website.

[2] N. Urasaki, T. Senjyu, K. Uezato and T. Funabashi, "An adaptive dead-time compensation strategy for voltage source inverter fed motor drives," *IEEE Trans. Power Electron*, Vol. 20, No. 5, pp. 1150–1160, 2005.doi.org/10.1109/TPEL.2005.854046

[3] A. R. Munoz and T. A. Lipo, "On-line dead-time compensation technique for open-loop PWM-VSI drives," *IEEE Trans. Power Electron*., Vol. 14, No. 4, pp. 683–689, 1999.

[4] N. Mohan, T. Undeland and W. Robbins, *Power Electronics: Converters, Applications and Design*. New York: Wiley, 2003.

[5] J. Kassakian, G. Verghese and M. Schlecht, *Principles of Power Electronics*. Reading, MA: Addison-Wesley, 1991.

[6] R. W. Erickson and D. Maksimovic, *Fundamentals of Power Electronics*, 2nd edition. Berlin: Springer, 2001.

[7] R. D. Middlebrook, "Predicting modulator phase lag in PWM converter feedback loop," Adv. Switched-Mode Power Convers., Vol 1, pp. 245–250, 1981.

[8] D. M. Van de Sype, K. De Gusseme, A. P. Van den Bossche and J. A. Melkebeek, "Small-signal Laplace-domain analysis of uniformly-sampled pulse-width modulators," *In 2004 Power Electronics Specialists Conference (PESC)*, 20–25 June, pp. 4292–4298.

[9] D. M. Van de Sype, K. DeGusseme, A. R. Van den Bossche and J. A. Melkebeek, "Small Signal Z-domain Analysis of Digitally Controlled Converters," *IEEE Trans. on Power Electron.*, Vol. 21, No. 1, pp. 470–478, 2006.

[10] G. C. Verghese, M. E. Elbuluk and J. G. Kassakian, "A general approach to sampled-data modeling for power electronic circuits," *IEEE Trans. Power Electron*., Vol. 1, pp. 76–89, 1986.

[11] G. R. Walker, "Digitally-implemented naturally sampled PWM suitable for multi-level converter control," *IEEE Trans. Power Electron*., Vol. 18, No. 6, pp. 1322–1329, 2003.doi.org/10.1109/TPEL.2003.818831

[12] Q. Yao and D. G. Holmes, "A simple, novel method for variable-hysteresis-band current control of a three phase inverter with constant switching frequency," in *Conf. Rec. IEEE-IAS Annual Meeting*, Toronto, ON, Canada, Oct. 1993, pp. 1122–1129.

[13] S. Buso, S. Fasolo, L. Malesani and P. Mattavelli, "A dead-beat adaptive hysteresis current control," *IEEE Trans. Indust. Appl.*, Vol. 36, No. 4, pp. 1174–1180, 2000.doi.org/10.1109/28.855976

CHAPTER 3

Digital Current Mode Control

In this chapter we begin the discussion of digital control techniques for switching power converters. In the previous chapter, we have introduced the topology and operation of the half-bridge VSI and designed an analog PI current controller for this switching converter. Referring to that discussion, the first part of this chapter is dedicated to the derivation of a digital PI current controller resembling, as closely as possible, its analog counterpart. We will see how, by using proper *discretization* techniques, the continuous time design can be turned into a discrete time design, preserving, as much as possible, the closed loop properties of the former. It is important to underline from the beginning that the continuous time design followed by some discretization procedure is not the only design strategy we can adopt. *Discrete time design* is also possible, although its application is somewhat less common: as we will explain, its typical implementations rely on the use of *state feedback* and *pole placement* techniques. The second part of the chapter will describe in detail a remarkable example of discrete time design and, in doing so, it will also show how the synthesis of regulators that have no analog counterpart whatsoever can be implemented. This is the case of the *predictive* or *dead-beat* current controller.

3.1 REQUIREMENTS OF THE DIGITAL CONTROLLER

The first step in the design of a digital controller is always the implementation of a suitable *data acquisition* path. While signal acquisition organization is somehow *implicit* in analog control design, because both the plant and the controller operate in the *continuous* time domain, digital control requires particular care in signal conditioning and analog to digital conversion implementation. The reason for this is ultimately that, while the control signals are taken from a plant that operates in the continuous time domain, the operation of the controller takes place in the *discrete* time domain. Therefore, signals have to be converted from the continuous to the discrete time domain and, of course, the other way round. It is very important to be aware of the fact that not *every* implementation of this conversion process leads to a satisfactory controller performance. We will see how the control of conversion noise and the avoidance of *aliasing* phenomena play a critical role.

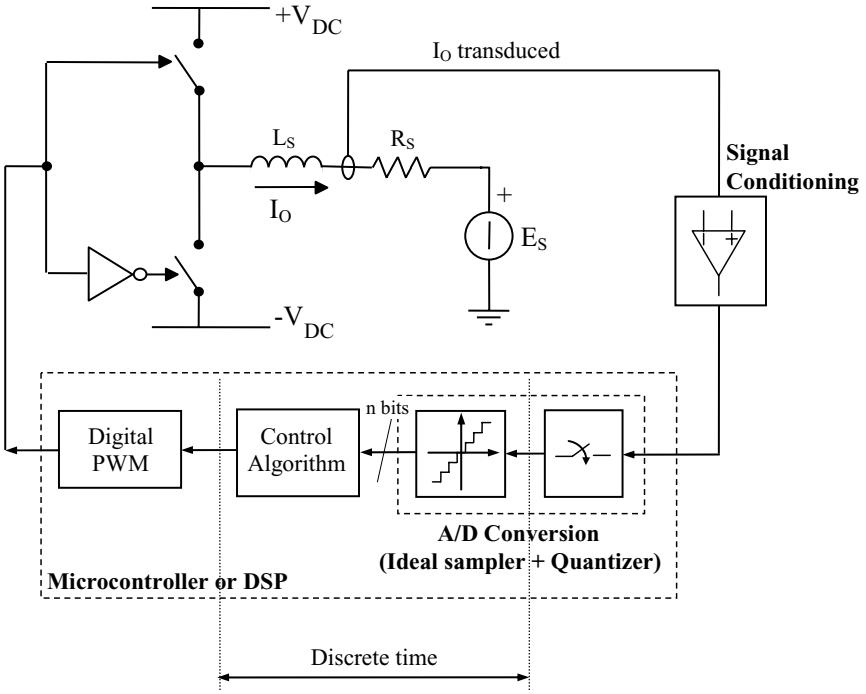

FIGURE 3.1: Typical organization of a digital current controller.

3.1.1 Signal Conditioning and Sampling

The typical organization of a digital current controller for the considered VSI is depicted in Fig. 3.1. Compared to Fig. 2.1, the power converter is represented here in a more compact form, using ideal switches and just a schematic representation of the driving circuitry, as these details are not essential for the following discussion. As can be seen, we assume that the digital controller is developed using a microcontroller (μC) or digital signal processor (DSP) unit, with suitable built-in peripherals. Although this is not the only available option for the successful implementation of a digital controller, it is by far more commonly encountered. Because of this, we will not discuss other possibilities, such as the use of custom digital circuits or field programmable gate arrays (FPGAs). Almost every μC and several low-cost DSP units, typically identified as motion control DSPs or industrial application DSPs, include the peripheral circuits required by the setup of Fig. 3.1. These are basically represented by an analog to digital converter (ADC) and a PWM unit. The data acquisition path for our current controller is very simple, being represented by the cascade connection of a current sensor, a properly designed signal conditioning electronic circuit, and the ADC. It is worth adding some comments about the conditioning circuit, with respect to its general features described in Section 2.1.2, in order to relate its function more precisely to the ADC operation. From this point of view, the

conditioning circuit has to guarantee that (i) the sensor signal is amplified so as to fully exploit the input voltage range of the ADC, and (ii) the signal is filtered so as to avoid *aliasing* effects.

The full exploitation of the ADC input voltage range is a key factor to reduce the quantization effects that may undermine control stability and/or reduce the quality of the regulation. The reason for this is that the number of *effective* bits, N_e, that are used for the internal representation of the input signal samples is maximum when the input voltage range is fully exploited. We can actually see that this number is given by the following relation,

$$N_e = n - \text{floor}\left(\frac{\log_{10}\frac{\text{FSR}}{S_{PP}}}{\log_{10} 2}\right), \tag{3.1}$$

where S_{PP} is the peak-to-peak amplitude (in Volts) of the transduced input signal, FSR is the ADC full scale range (in Volts), and n is the ADC bit number. A little complication we typically find when designing the conditioning circuit is related to the sign of the input signal. It is quite common for the transduced current signal to be bipolar (i.e., to have both positive and negative sign), while the lower bound of the ADC voltage range is almost always zero. To take care of that, the conditioning circuit has to offset the input signal by a half of the ADC FSR. This operation associates the lower half of the ADC range with the negative values of the input signal, and the upper half with the positive values. These simple considerations are normally enough to properly design the gain of the conditioning amplifier in the frequency band of interest. Given the expected peak-to-peak amplitude of the VSI output current and considering a suitable safety margin for the detection of overcurrent conditions, due to load transients or faults, it is immediately possible to determine the gain required to exploit the ADC full scale range.

The aliasing phenomenon is a consequence of the violation of Shannon's theorem, which defines the limitations for the exact reconstruction of a uniformly sampled signal [1]. The theorem shows that there is an upper bound for the sampled signal bandwidth, beyond which perfect reconstruction, even by means of ideal interpolation filters, becomes impossible and aliasing phenomena appear. The limit frequency is called the Nyquist frequency and is proved to be equal to a half of the sampling frequency, f_C. In general, we will have to limit the frequency spectrum of the sampled signal by filtering, so as to make it negligible above the Nyquist frequency. This condition will determine the bandwidth and roll-off of the conditioning amplifier. A very intuitive graphical representation of the aliasing phenomenon is given in Fig. 3.2.

Another interesting issue, related to signal acquisition in digital control, is the definition of a suitable ADC model. From Fig. 3.1 we can see that the analog to digital conversion process can be mathematically modeled as the cascade connection of an ideal sampler and an n-bit

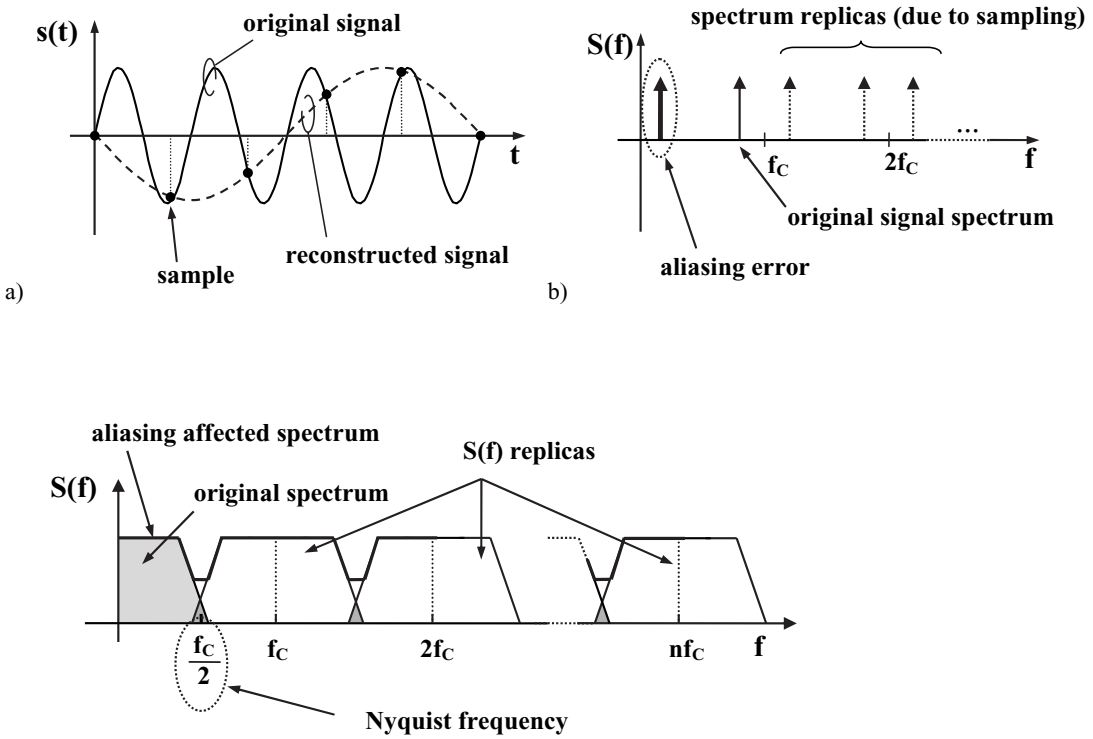

FIGURE 3.2: (a) Effect of a too low sampling frequency on the reconstructed signal (aliasing). (b) Interpretation of the aliasing effect of (a) in the frequency domain. Note how a low-frequency spectrum component is generated because of aliasing. (c) A more general situation: a distorted, aliasing-affected, spectrum is reconstructed because of the partial overlap of spectrum replicas.

uniform quantizer. The former is defined as a sampler whose output is a stream of *null duration* pulses, each having an amplitude equal to that of the input signal at the sampling instant. Its function is to model the actual sampling process, i.e., the transformation of the time variable from the continuous domain to the discrete domain, where time only exists as integer multiples of a fundamental unit, the sampling period. The latter is taken into account to model the loss of information implied by what can be interpreted as a *coding* procedure, where a continuous amplitude signal, i.e., a signal whose instantaneous level can vary with continuity in a given range of values, is transformed into a discrete amplitude signal, i.e., a digital signal, whose instantaneous level can only assume a finite number of values in the same given range. Because the possible discrete values can be interpreted as integer multiples of a fundamental unit, the quantization step Q, or, equivalently, the least significant bit (LSB), the quantizer is called "uniform." Nonuniform quantizers can sometimes be encountered, but very rarely in the kind

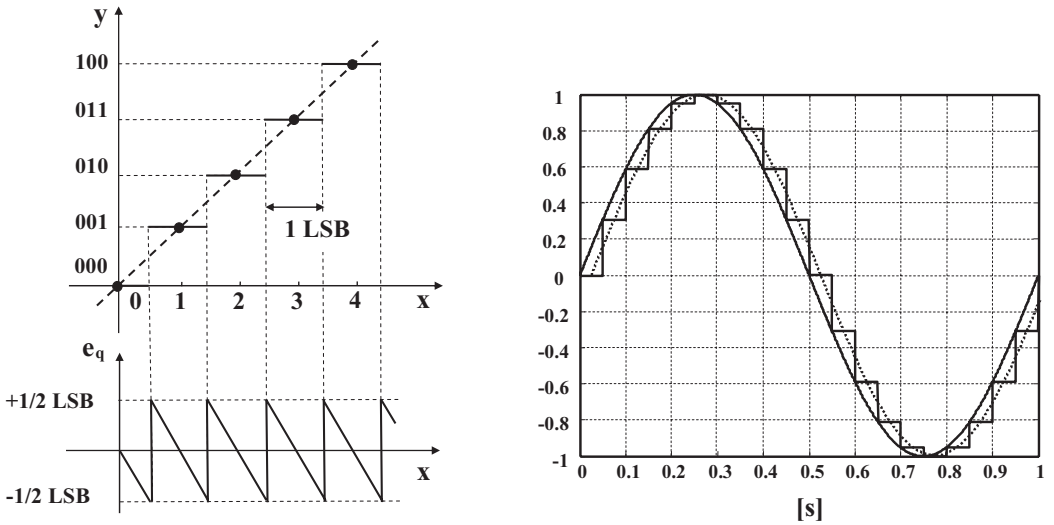

FIGURE 3.3: (a) Uniform quantizer transcharacteristic and quantization error e_q. (b) Sample and hold delay effect: compare the input signal (continuous line) and the reconstructed output signal, i.e., the fundamental harmonic component of the sampled signal (dotted line).

of application we are interested in. For this reason, we will only discuss the uniform quantizer case. The typical transcharacteristic diagram for a uniform quantizer is shown in Fig. 3.3(a). As can be seen, a typical quantization noise e_q can be defined that is added to the signal as a result of analog to digital conversion. This can be interpreted as the loss of some of the information associated with the input signal, inherent to the analog to digital conversion and unavoidable. We will further discuss this phenomenon in one the following paragraphs. As far as the dynamic behavior of the ADC is concerned, it should be evident that both the quantizer and the ideal sampler are essentially instantaneous functions, which do not contribute to the dynamics of the system.

Fig. 3.1 reveals another interesting point about the digital current controller organization, which is related to the digital PWM. This component processes the output of the control algorithm, a discrete time signal, and turns it into a continuous time signal, the state of the switches. This function, which represents the inverse of the sampling process and allows the controller to actuate the system under control, is known as *interpolation.* It is now evident that, from the digital control theory's standpoint, the DPWM is the part of our control system where interpolation takes place.

For reasons that will become clear in the following, it is often important to develop a *continuous time equivalent* model of the controller, i.e., of everything that is included between

the sampler and the interpolator. In other words, we often are interested in a mathematical description of the digital controller as it is "seen" from the external, continuous time world's standpoint. This problem can be solved by considering what is known as a zero-order hold (ZOH) approximation of the interpolation process. Neglecting the presence of the control algorithm, we can describe this model simply by considering that, in order to reconstruct the continuous time signal from the discrete time input samples, each sample value is held constant for the entire duration of the sampling period. It is actually possible to use different interpolation models [2], but, for the problems of our interest, this is normally a good enough model. We will see in the following how this approach is related to the DPWM equivalent continuous time models presented in Chapter 2.

However, it is immediate to recognize in this function a typical dynamic effect: anytime a signal is sampled and converted again into a continuous time signal by the interpolator, which we have now modeled as a simple holder, we cannot reconstruct exactly the original signal, but we have to face a delay effect that is directly proportional to the sampling period. An example of this effect is shown in Fig. 3.3(b). We will come back to this issue in Section 3.2.2, when we discuss the digital controller design technique based on *discretization*.

3.1.2 Synchronization Between Sampling and PWM

The general considerations presented in the previous section have to be extended considering the particular nature of the system we want to control. As defined in Chapter 2, the VSI is controlled at the lowest level by a PWM modulator. This determines the presence, on each electrical variable, of the typical high-frequency noise known as ripple. It is fundamental to clarify how this is taken care of in the sampling process.

It is evident that, in order not to violate Shannon's theorem, the sampling process should proceed at a very high frequency, so high that the spectrum of the sampled signal might be considered negligible at the Nyquist frequency, even if a significant ripple is observable. This would require a sampling frequency at least one order of magnitude higher than the switching frequency. Unfortunately, hardware limitations do not allow the sampling frequency to become too high: we must keep in mind that our controller implementation will be based on standard microcontroller or DSP hardware.

When we discuss the adoption of multisampling strategies, we will see how they require a nonconventional hardware organization; for example, the use of FPGA circuits. In a typical case, instead, since the duty-cycle update is allowed at most twice per modulation period, in the double update mode of operation of the digital PWM, the sampling frequency cannot get higher than twice the switching frequency. Of course, in order to push the bandwidth of the closed loop plant as high as possible, we are normally not interested in sampling frequencies lower than the allowed maximum, at least for the current controller. When, in one of the following chapters,

we discuss the application of digital control to external loops, we will see how sometimes lower sampling frequencies can offer some advantages. However, in the case of a current controller, the sampling frequency should be maximized. The reason for this is quite obvious: by doing so the inherent sample and hold delay can be minimized and, consequently, the closed loop plant bandwidth can be maximized.

In conclusion, in a typical case, the sampling frequency will be set equal to either the switching frequency, or, if this is consistent with the available digital PWM implementation, two times the switching frequency. But if this is what we do, the Shannon's theorem conditions will always be violated!

This is one of the key issues in digital control applications to power electronic circuits: the typically recommended high ratio between sampling frequency and sampled signal bandwidth will never be possible. Nevertheless, we will shortly see how this is normally advantageous, rather than detrimental, for the controller effectiveness. The reason for this lies in *synchronization*.

If the sampling and switching processes are suitably synchronized, the effect of aliasing is the automatic reconstruction of the average value of the sampled signal, which is *exactly* what has to be controlled. This means that the violation of the Shannon's theorem conditions does not actually limit the controller performance, but it even helps to reduce the controller complexity. The need for low-pass filters to eliminate the ripple from the sampled signal is, in fact, removed. This effect is schematically shown in Fig. 3.4.

We can see that synchronization allows the reconstruction of the average signal value anytime the sampling takes place in the middle of the switch-on period or in the middle of

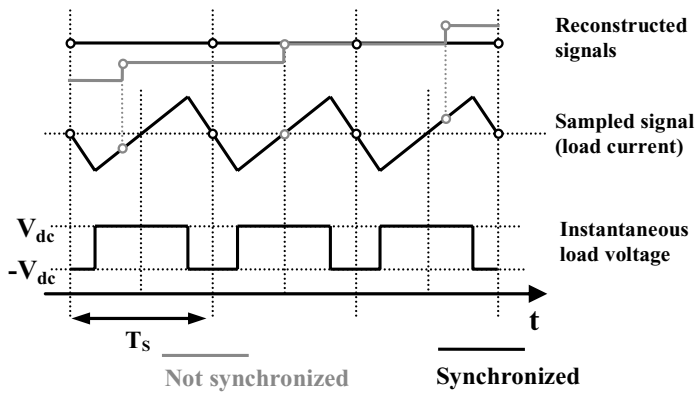

FIGURE 3.4: Example of synchronized sampling and switching processes. In case the sampling takes place always at the beginning (or in the middle) of the modulation period, the average current value is automatically obtained. If the sampling frequency is lower than the switching one, an aliased, low-frequency component appears on the reconstructed signal.

the switch-off period (or both, if double update mode is possible). Instead, if the switching and sampling frequencies are different, low-frequency aliased components will be created in the reconstructed signal. Please note that, even if the sampling and switching frequencies are set equal, there still can be a zero frequency error in the reconstruction of the average sampled signal, in case the sampling instants are not coincident with the beginning and/or the half of the modulation period. This is generally a minor problem, since the current regulator will often be driven by an external loop (see Chapter 5) that, typically including an integral action, will compensate for any steady-state (or very low frequency) error in the current trajectory.

To minimize aliasing effects and reconstruction errors, practically all of the microcontrollers and DSPs designed for power converters control allow virtually perfect synchronization of the sampling and switching processes. In most cases, the ADC operation is synchronized by the processor hardware with the modulator. Typically, analog to digital conversion of the control variables is started by a signal that also clocks the beginning of the modulation period and can be retriggered at a half of the modulation period, if needed.

3.1.3 Quantization Noise and Arithmetic Noise

Quantization of variables and finite arithmetic precision are two among the most critical issues in digital control. Even if a detailed discussion of these issues is far beyond the scope of this book, we feel that it is mandatory to recall at least some basic information about both of them. The interested reader can deepen his or her knowledge of both issues referring to very good digital control and digital signal processing textbooks such as [1–3].

As we briefly discussed in Section 3.1.1 quantization takes place anytime the amplitude values of a sampled signal are coded using a finite set of symbols. While the original signal's instantaneous amplitude can assume an infinite number of values in a given range, the sampled and coded signal's amplitude can only take one out of a finite number of possible values. The typical implementation of analog to digital conversion in microcontrollers and DSPs associates a *binary* code with the amplitude values of the sampled signal. In the case of the uniform quantizer, the rule to associate a binary code N with any given signal sample x is very simple, and can be mathematically expressed as

$$\begin{cases} \left(N-\dfrac{1}{2}\right)\cdot Q < x < \left(N+\dfrac{1}{2}\right)\cdot Q \quad \Rightarrow \quad x_{\mathrm{q}} = N \\ Q = \dfrac{\mathrm{FSR}}{2^n} = \mathrm{LSB}, \end{cases} \tag{3.2}$$

where n represents the ADC bit number and, as was previously described, if FSR represents the full scale range, in volts, of the ADC, then Q is the ADC quantization step, equal to one least significant bit (LSB). Please note that (3.2) simply translates the transcharacteristic of the

uniform quantizer depicted in Fig. 3.3(a) into a mathematical form. From (2.3) we see that Q represents the minimum variation of input signal x that *always* causes the variation of at least one bit in the binary code associated with x_q, the coded signal. Therefore, any variation of signal x smaller than Q is not always able to determine some effect on x_{q}. This simple observation shows us that the quantization process actually implies the loss of some of the information associated with the original signal x. It is a common approach to model this effect as an additive noise, superimposed to the signal. In order to simplify the mathematical characterization of the quantization noise, the stochastic process associated with it is assumed not to be correlated to signal x, which is (obviously) hardly the case, uniform in probability density and with a statistical power equal to

$$\sigma_q^2 = \frac{\mathrm{LSB}^2}{12}. \tag{3.3}$$

It is then possible to derive a very useful relation that expresses the *maximum* signal to noise ratio (SNR) of an ADC as a function of its number of bits. This is given by

$$\mathrm{SNR} = 10 \cdot \log_{10}\left(\frac{12}{8} \cdot 2^{2n}\right) = 6.02 \cdot n + 1.76\,(\mathrm{dB}). \tag{3.4}$$

We will not elaborate the statistical modeling of the quantization noise any further. Equation (3.4) is a very useful tool to estimate the number of bits one needs, in order to get a desired SNR for a given conversion process. For example, if one needs at least a 50 dB SNR, (3.4) shows that the number of bits should be higher than 8. Please note that this model does not take into account any other source of noise besides quantization, like, for example, those associated with the signal conditioning circuitry or with the power converter. Consequently, the *actual* signal to noise ratio will always be lower than what is estimated by using (3.4).

There are at least two other major forms of quantization that always take place in the implementation of a digital control algorithm: (i) arithmetic quantization and (ii) output quantization. As far as the former is concerned, we can say that what we call arithmetic quantization is nothing but an effect of the finite precision that characterizes the arithmetic and logic unit used to compute the control algorithm. The finite precision determines the need for truncation (or rounding) of the controller coefficients' binary representations, so as to fit them to the number of bits available to the programmer for variables and constants. In addition, it may determine the need for truncation (or rounding) after multiplications. In general, the effect of coefficient and multiplication result truncation (or rounding) is a distortion of the controller's frequency response, i.e., the shift of the system poles, that can have some impact on the achievable performance. Both truncation and rounding effects can be modeled again as a type of quantization and so as an equivalent noise, of arithmetic nature, added to the signal. Although extremely

interesting, predicting the amplification of arithmetic noise within a closed loop control algorithm by pencil and paper calculations is a really tough job. To check the control algorithm operation to this level of detail, the only viable option is its complete, low-level simulation, based on a model that includes the emulation of the adopted controller arithmetic unit.

It should be clear by now that, in case a floating-point representation of constants and variables within a control algorithm were employed, none of the above-discussed arithmetic quantization effects could be observed. It is important to say, though, that the availability of floating-point processors in the field of digital control industrial applications is very rare. At the time of writing (2006), only state-of-the-art DSP units, designed for high-performance real-time signal processing, can rely on a floating-point arithmetic unit. However, the cost of such DSP units is well beyond the maximum affordable for a typical industrial control application. Therefore, at least for the near future, industrial engineers, designing digital regulators for switching converters, will have to face the problems generated by fixed-point arithmetic units. Fortunately, the availability of low-cost 16- or even 32-bit microcontrollers and DSPs is increasing every day. The occurrence of severe arithmetic quantization problems is therefore rarer and rarer, being confined to extremely demanding applications or to applications where the use of 8-bit microcontrollers is the only viable option and the emulation of a higher precision arithmetic is out of the question for memory or timing constraints. It is basically for this reason that we will not take arithmetic quantization into account in the following discussion of digital control implementation. In practice, our results will be determined by assuming infinite precision arithmetic, considering it to be well approximated by modern 16-bit digital controllers.

Output quantization, instead, is related to the truncation (or rounding) operation inherent in the digital to analog conversion that brings the control algorithm output variable back from the digital to the continuous time domain. In our application case, this function is actually inherent in the digital PWM (DPWM) process. The reduction of the control variable output (in our case the desired duty-cycle) bit number, needed to write it into the PWM duty-cycle register, represents again a quantization noise source. Note that unless a very high clock to modulation frequency ratio is available (see Section 2.2.2), the effective number of bits that might be used to represent the duty-cycle is always much smaller than the typical variable bit number (16 or 32). Therefore, output quantization is unavoidable. The most unpleasing effect of output quantization may be the occurrence of a peculiar type of instability, specific to digital control loops, that is known as limit cycle oscillation (LCO).

To open just a brief discussion of LCOs, we would like to show, in the first place, how a limit cycle can be generated in a very simple situation. The case is depicted in Fig. 3.5. We denote by variable d the duty-cycle of a switching converter, like the one considered in our discussion, whose desired set-point is the particular value we need to apply to bring the converter to the steady state. Variable x may be associated, for example, with the converter

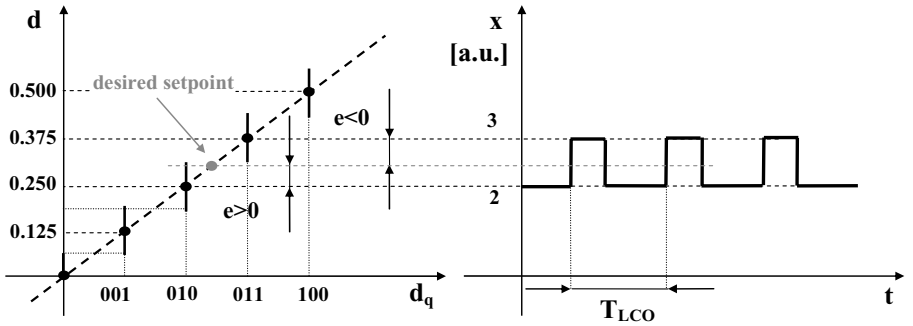

FIGURE 3.5: Example of limit cycle occurrence. The desired set-point for the output control variable d is not one of the possible output values. Consequently, the system oscillates, with period T_{LCO}, between the two closest outputs. Here we assume that the system includes at least one integral action in the transfer function from the input to the output.

average output current. Unfortunately, as we see from Fig. 3.5, the desired set-point for d is not any one of the possible outputs, because of output quantization.

As a result, we will in any case apply either a bigger than needed duty-cycle, causing the current increase beyond the steady-state level, or a lower than needed duty-cycle, causing the current decrease below the steady-state value. This happens because the converter output current is, to a first approximation, proportional to the integral of the inverter average output voltage, which is in turn proportional to the duty-cycle. Commutations between the two states are determined by the current controller, which reacts to the current error buildup by changing the duty-cycle.

This results in a persistent oscillation, i.e., a limit cycle, of the control variables, which is not due to any system instability but only to the presence of the output quantization. Of course, the amplitude and frequency of the limit cycle are largely dependent on several controller and converter parameters like, for example, controller bandwidth, open loop system time constants and open loop system static gain. Please note that in the cases like the half-bridge converter considered here, where the input to output converter transfer function presents a low-pass behavior, well approximated by an integral action, this type of limit cycle is practically unavoidable.

Within the general digital control theory, limit cycles have been extensively studied, with different degrees of detail and complexity. In power electronics and, more precisely, in the area of dc–dc converter applications, several fundamental papers on quantization resolution and limit cycling have been published, like, for example [4, 5] and others cited therein. Without entering too much into this fairly complex topic, we would now like to review the fundamental conditions for the elimination of limit cycles. It is worth clarifying, right from the start, that the conditions reported hereafter are necessary, but not sufficient, for the elimination of limit

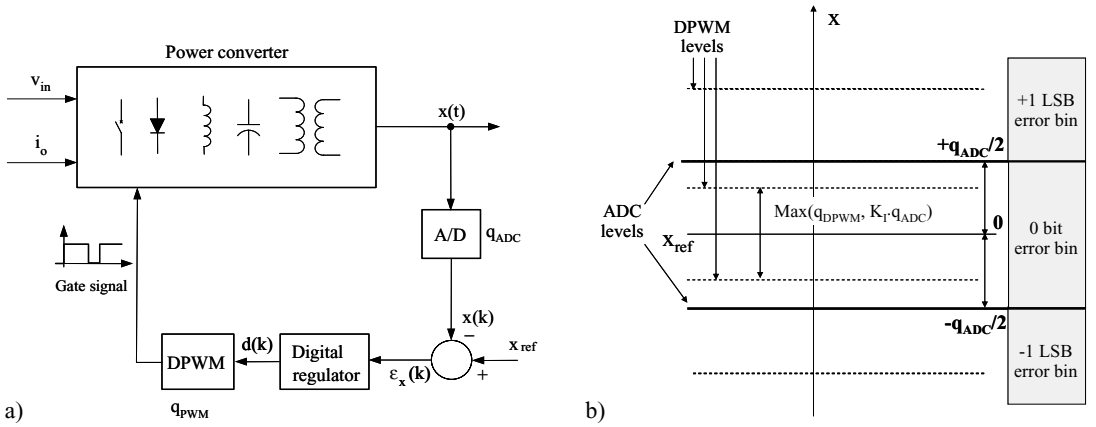

FIGURE 3.6: (a) Digitally controlled power converter with ADC and DPWM quantization; (b) quantization of state variable $x(t)$ and effects of DPWM quantization.

cycle oscillations. Therefore, the actual presence and amplitude of LCOs are usually checked by means of time-domain simulations. This may be a time-consuming investigation, since the presence of LCOs strongly depends on the converter operating point, e.g., on the load current and input voltage levels. In some cases, the system does not show LCOs, except for a very small set of output current values. In addition, a limit cycle can sometimes be triggered only by some particular transients, having a very particular amplitude. It is therefore not so easy to ensure the actual elimination of LCOs.

However, in order to review the fundamental conditions for the elimination of LCOs, let us consider the digitally controlled power converter shown in Fig. 3.6(a), where we assume that the dominant quantization effects derive from the ADC and the DPWM, while the rounding effects in the control algorithm are neglected. As a matter of fact, the fixed-point arithmetic and the coefficient round-off may play a relevant role in the accuracy of the controller frequency response definition and in the amplification of quantization noise. Nevertheless, a practical design approach is often based on the assumption of infinite controller resolution and on the verification a posteriori by means of time-domain simulations and experiments.

The *first* condition is to ensure that the variation of one DPWM level, i.e., 1 LSB of the duty-cycle digital representation, here denoted as q_{DPWM}, does not give a variation of the controlled output variable $x(t)$, in steady-state conditions, greater that the quantization level of $x(t)$, here denoted as q_{ADC}. Thus, if we define as $G(s)$ the transfer function between the duty-cycle, d, and the controlled variable, $x(t)$, the first necessary condition for the elimination of LCOs is

$$q_{\mathrm{DPWM}}\, G_{\mathrm{dc}} < q_{\mathrm{ADC}}, \tag{3.5}$$

where G_{dc} is the steady-state gain (i.e., $G_{\mathrm{dc}} = G(j0)$). The condition (3.5) indicates that the effect on variable x of the DPWM quantization step, determined by the converter steady state gain G_{dc}, must be smaller than the ADC quantization step. It is worth noting that this reasoning applies to the control of dc quantities, while the analysis, and even the interpretation, of limit cycles in the presence of time-varying references (as in dc/ac converters) may be slightly different.

The *second* condition is the presence of an integral action in the controller. This condition has been formally demonstrated in [5]. However, its motivation can be explained considering that, if only a proportional term (or a proportional derivative term) is included in the adopted controller, a minimum quantized error on the controlled variable $x(t)$ determines a variation on the average converter output voltage that is equal to $G_{\mathrm{dc}} \cdot K_{\mathrm{P}} \cdot q_{\mathrm{ADC}}$ (even considering the quantization of the DPWM to be infinite). Since $G_{\mathrm{dc}} \cdot K_{\mathrm{P}}$ is usually much greater than 1, this variation is much greater than q_{ADC} and, consequently, condition (3.5) is not satisfied. Therefore, in order to comply with (3.5), a lower amplification of the minimum quantized error on the input variable must be ensured. This always happens when an integral action is included in the control algorithm. In that case, the integral gain *induces* a smaller quantization effect on the DPWM, since the minimum variation of the duty-cycle, due to the minimum quantized error on $x(t)$, is now equal to $K_{\mathrm{I}} \cdot q_{\mathrm{ADC}}$, with K_I normally much smaller than K_{P}. This guarantees that (3.5) is typically satisfied. Of course, in addition to that, the following condition has to be satisfied as well,

$$K_{\mathrm{I}} G_{\mathrm{dc}} < 1, \tag{3.6}$$

which actually imposes an upper limit to K_{I}. The simultaneous verification of conditions (3.5), where we can now define the DPWM resolution as the *maximum* between its *physical,* hardware quantization and what we have called the *induced* quantization, determined by the integral term, and (3.6), is necessary to make the elimination of LCOs theoretically possible. A schematic representation of these considerations is also given in Fig. 3.5(b). However, even if the two conditions above are satisfied, limit cycle oscillations may still be present, essentially because of the effect of the quantizer nonlinearity on the feedback loop.

This possible instability may be analyzed using *describing function techniques*, including the ADC quantization and possibly the DPWM's one. Thus, the *third* condition for LCO elimination is that the closed loop system is stable from the describing function's standpoint. Unfortunately, the describing function approach is a valid approximation only in the case of limit cycle oscillations that are well approximated by sinusoidal waveforms.

In conclusion, we can say that the analytical prediction of the occurrence of limit cycles, of their amplitude, and their frequency is a very complicated problem. In any case, the use of simulation is highly recommended, since the compliance with the above three conditions, as we explained, does not guarantee the absence of LCOs. However, it is important to underline

that, even if a limit cycle is detected, a proper design of the controller and the signal acquisition path can generally bring its amplitude and frequency to practically acceptable levels.

3.2 BASIC DIGITAL CURRENT CONTROL IMPLEMENTATIONS

In this section, we present the basic implementations of the digital current controller for the VSI depicted in Fig. 3.1. We will discuss different control algorithms and the related design criteria, with the intention of highlighting the merits and the limitations of each solution. The discussion will refer to an ideal controller implementation, where the above-mentioned quantization effects can be considered negligible. Instead, we will focus our attention on the performance allowed by the different solutions and on the impact of the digital controller implementation on the dynamic response of the converter, considering, in particular, figures of merit like the response delay to step changes in the current reference, or the residual tracking error in the presence of sinusoidal reference current signals. Throughout the discussion, we will refer to the converter parameters that we have already taken into account in Section 2.3.1, where we presented the analog controller implementation, and that are reported in Table 2.1.

3.2.1 The Proportional Integral Controller: Overview

The first digital controller we discuss is the proportional integral, or PI, controller. In the last part of Chapter 2, we have described in detail a possible analog implementation of this solution. We now move to a digital implementation observing that, in general, it can be quite convenient to derive a digital controller from an existing analog design. This procedure, which is called *controller discretization,* has the advantage of requiring only a minimal knowledge of digital control theory to be successfully applied. All that is needed is a satisfactory analog controller design and the application of one of the several possible discretization methods to turn the analog controller into a digital one. As we will see in the following, although generally satisfactory, the application of this method implies some loss of precision, as compared to a direct digital design, mainly due to the approximations involved in the discretization process itself and in the equivalent continuous time representation of delays.

Referring to Fig. 3.7, we can see the block diagram of the control loop. As can be seen, it replicates the organization of the block diagram of Fig. 2.10, with the remarkable difference that some of the blocks are now *discrete time* blocks. In particular, we can see how the controller and modulator blocks are now inside the digital domain, the shaded area, that represents a microcontroller or DSP board. The inverter and transducer models are instead exactly equal to those of Fig. 2.10 and, as such, continuous time models. The link between the two time domains is represented by the ideal sampler at the input of the controller and by the digital pulse width modulator, which generates the controller output and, as we have explained, inherently

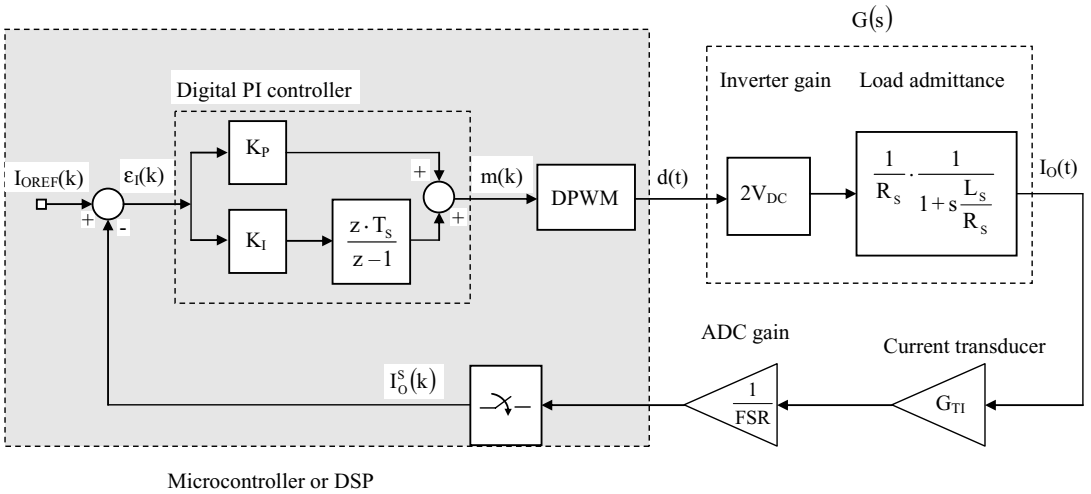

FIGURE 3.7: Block diagram of the digital current control loop with PI regulator.

implements the interpolator, or holder, function. All these characteristics imply that we are actually dealing with a *sampled data* dynamic system.

For the reasons we previously explained talking about synchronization, we assume that the controller operation is clocked by the DPWM, i.e., a new iteration of the control algorithm is started as soon as a modulation period begins. We also assume, for simplicity, that the single update mode of operation is adopted, so that, during each modulation period, a single new value of the controller output is computed. The computation is based on the current sample, acquired at the start of the period and indicated by $I_O^S(k)$. Since the controller operation proceeds at time steps that are multiple of T_S, the modulation and sampling period, in all the controller signals we simply denote with k the instant $k \cdot T_S$ from the origin of time. Accordingly, we say that, at the kth modulation period, the output of the controller, i.e., the digital modulating signal, is $m(k)$. Please note that, although we keep identifying the output of the controller by m, as in the analog case, this must no longer be considered an analog signal, but rather a sequence of binary codes, i.e., a quantized discrete time signal. Of course, the same holds for each of the other controller internal signals, like I_{OREF} and I_O^S.

It is worth noting that, in order to make Fig. 3.7 more realistic, we will modify the static gains of the modulator and of the feedback path with respect to the analog design example of Fig. 2.10. Indeed, in a digital implementation, the modulator static gain is represented by the numerical scale factor that turns the binary code $m(k)$ in the corresponding duty-cycle $d(t)$. In general, this depends on the way variables are normalized in the control algorithm. It is possible to verify that, as soon as the normalization of variables is such that $m(k)$ is coded as a fractional binary number, i.e., the maximum binary value of m is made equivalent to unity, the

modulator static gain is also unity, i.e., $m(k)$ directly represents the duty-cycle, without further scale factors. The fractional normalization hypothesis also explains the presence of the ADC gain at the input of the digital controller, meaning that a full scale input value of the ADC is normalized to unity as well. Under these assumptions and without loss of generality, we will assume the DPWM static gain to be equal to unity. If a different normalization criterion is adopted, the modulator static gain will have to be adjusted accordingly.

3.2.2 Simplified Dynamic Model of Delays

As briefly outlined above, the application of discretization techniques requires the designer to determine an equivalent continuous time model of his or her sampled data system, to use it in the design of a continuous time controller stabilizing the feedback loop and, finally, to turn the continuous time controller into an equivalent discrete time one. Therefore, first of all, we need to discuss the derivation of an equivalent, continuous time model for the system represented in Fig. 3.7.

The typical textbook approach [2, 3] to sampled data dynamic systems control normally requires us to properly model, in the continuous time domain, the discrete time system included between the ideal sampler located at the controller input and the output interpolator. As we have explained in Section 3.1.1, the typical way to do this is by considering a suitable model of the interpolator, e.g., some kind of holder, and, after that finding an equivalent continuous time representation for the cascade connection of the ideal sampler and the holder, which is called a *sample and hold*. Please note that this method, schematically illustrated by Fig. 3.8, is actually what we have already used in Chapter 2, modeling the different types of DPWM. Once the sample and hold is modeled, the designer can operate the controller synthesis in the continuous time domain, assuming that once converted back into a discrete time equivalent and inserted between the sampler and the interpolator in the original sampled data system, the controller will maintain the closed loop properties determined by the continuous time design.

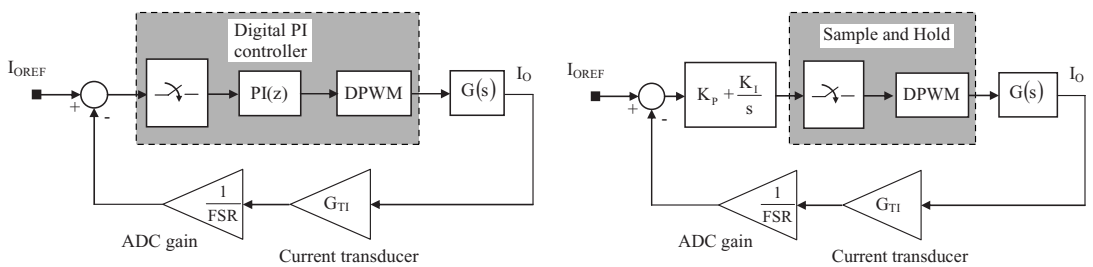

FIGURE 3.8: Procedure to define the continuous time equivalent of the digital current control loop.

This is what we have to do with the sampled data system of Fig. 3.7, with a significant difference. The difference lies in the fact that, in this case, the function of the interpolator is inherent to the DPWM, because that is the block where the conversion from the digital to the analog domain takes place. This means that once the holder effect is properly modeled in the DPWM, the conversion of the sampled data system into an equivalent, continuous time one will be complete. This may seem a minor detail, but in this difference lies the key for the correct interpretation of the system in Fig. 3.7 as a sampled data system. In Chapter 2, we have described several continuous time equivalent models for the DPWM. Considering, for example, model (2.7), after minor rearrangements and assuming, as we explained above, $c_{PK} = 1$, we get the following expression,

$$\mathrm{DPWM}(s) = \frac{1}{2}\left(\mathrm{e}^{-s(1-D)\frac{T_S}{2}} + \mathrm{e}^{-s(1+D)\frac{T_S}{2}}\right) = \mathrm{e}^{-s\frac{T_S}{2}}\cos\left(\omega\frac{T_S}{2}D\right) \cong \mathrm{e}^{-s\frac{T_S}{2}}, \qquad (3.7)$$

which, as can be seen, shows the equivalence of the considered DPWM to a half modulation period delay, cascaded to a frequency-dependent again. Considering the typical current controller bandwidth to be limited well below the modulation frequency, $1/T_S$, the gain term can actually be approximated by unity, independently of the duty-cycle D, so that the last part of (3.7) holds. In the above assumptions, (3.7) shows that we can quite accurately model the DPWM as a pure, half modulation period delay. Please note that this exactly coincides with the continuous time model of the zero-order hold usually adopted in a sampled data controller design. Of course, if a different DPWM model were considered, the result (3.7) would represent a coarser approximation, but could still be used as a simplified representation of the holder delay effect. Considering now the first-order Padé approximation of (3.7), a rational, continuous time transfer function can be obtained,

$$\mathrm{e}^{-s\frac{T_S}{2}} \cong \frac{1 - s\frac{T_S}{4}}{1 + s\frac{T_S}{4}}, \qquad (3.8)$$

where T_S is, of course, the sampling period. The usefulness of (3.8) is that a rational transfer function is clearly easier to deal with than the exponential function. We have actually already met (3.8) in Chapter 2, Fig. 2.10, where it was used, basically under the same assumptions, to approximately model the DPWM delay in an analog regulator design example.

We are now ready to consider the continuous time equivalent of our sampled data system. This is shown in Fig. 3.9. As can be seen, we have obtained exactly the same model of Fig. 2.10, with the only difference that the static gain of the modulator is now considered equal to 1 and that there is an additional gain in the feedback path. To simplify the following developments of this result, we assume $\mathrm{FSR} = c_{PK}$, so that the open loop static gain of Fig. 3.9 and that

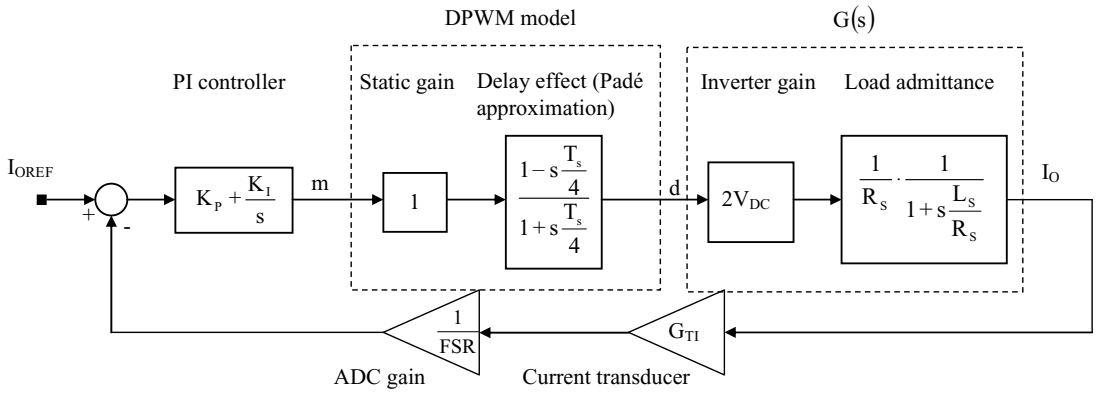

FIGURE 3.9: Block diagram of the continuous time equivalent of the digital current control loop.

of Fig. 2.10 are identical. Of course, in general, the two loop gains will have a different dc value, which will require some straightforward adjustment of the controller parameters. Under our assumption instead, the analog PI controller we have designed in Chapter 2 represents a satisfactory stabilizing controller also for the loop of Fig. 3.9.

Therefore, we are now ready to take the last step toward the design of the digital PI current controller. All we have to do is to apply a suitable discretization method to the analog controller we already possess. The way this can be done is the subject of next section.

3.2.3 The Proportional Integral Controller: Discretization Strategies

According to digital control theory, the application of any discretization method always implies a loss of performance with respect to a purely analog control implementation. This is also true for our case. Indeed, if a analog current controller were designed for the system of Fig. 3.7, since the delay effect of the analog PWM is negligible, the controller bandwidth could be higher than that we can achieve once a digital modulator, which presents a higher delay, is used.

In Chapter 2 we have chosen to design the analog PI controller considering a digital PWM modulator and modeling its delay exactly as in Fig. 3.9. That choice, together with the "educated" choice of the ADC FSR value that was done in the previous section, allowed us to find a controller that, although not ideal for the analog implementation, is now ready for discretization without further adjustments. From a textbook's standpoint, this offers two advantages: to keep the presentation more compact and to allow, in the end, the comparison of two virtually identical controllers, analog and digital, and thus putting into evidence the impact of discretization on the final performance. However, note that in the general case the analog design would have to be started from scratch, based on the equivalent model of Fig. 3.8.

There are actually several possible discretization strategies, some based on the invariance of the dynamic response to particular signals (steps, ramps, etc.) and the others based on numerical

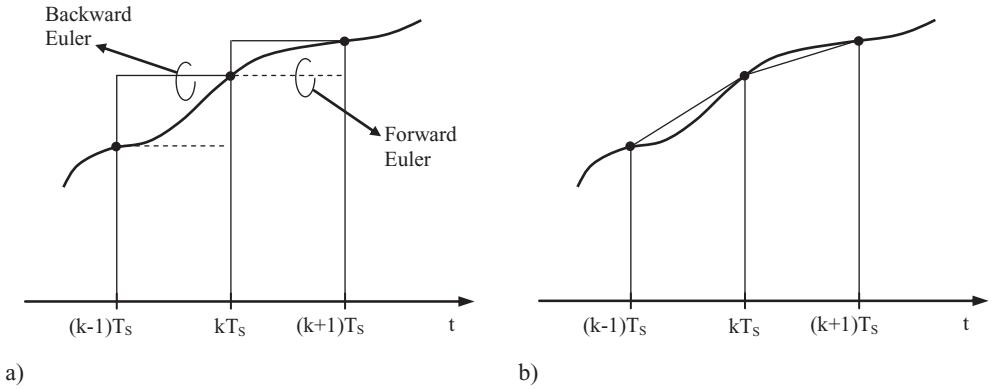

FIGURE 3.10: (a) Euler integration method (forward and backward). (b) Trapezoidal integration method.

integration methods. The latter are those we will consider now. The basic concept behind them is very simple: we want to replace the continuous time computation of integrals with some form of numerical approximation. The two basic methods that can be applied for this purpose are known as the Euler integration and the trapezoidal integration method. The principle is illustrated in Fig. 3.10.

As can be seen, the area under the curve is approximated as the sum of rectangular or trapezoidal areas. The Euler integration method can actually be implemented in two ways, known as *forward* and *backward* Euler integration, the meaning being obvious from Fig. 3.10(a). Writing the rule to calculate the area as a recursive function of the signal samples, applying Z-transform to this area function, and imposing the equivalence with the Laplace transform integral operator, gives a direct transformation from the Laplace transform independent variable s to the Z-transform independent variable z.

Table 3.1 shows the transformations that are obtained for the two discretization methods, where the two possible versions of the Euler integration method are considered. These are called *Z-forms*. The practical meaning of each Z-form is as follows: the substitution of the s variable in the controller transfer function with the indicated function of the z variable determines

TABLE 3.1: Discretization Methods

METHOD	*Z*-FORM	3% DISTORTION LIMIT
Backward Euler	$s = \frac{z-1}{z \cdot T_S}$	$\frac{f_S}{f} > 20$
Forward Euler	$s = \frac{z-1}{T_S}$	$\frac{f_S}{f} > 20$
Trapezoidal (Tustin)	$s = \frac{2}{T_S}\frac{z-1}{z+1}$	$\frac{f_S}{f} > 10$

Aside 3. Discretization of the PI Current Controller

In Aside 2, we have determined the proportional and integral gains of an analog PI current controller. These were $K_P = 6.274$ and $K_I = 1.8 \times 10^4$ (rad s^{-1}). The corresponding controller transfer function is given by

$$\mathrm{PI}(s) = K_I \frac{1 + s \cdot \dfrac{K_P}{K_I}}{s}. \qquad (A3.1)$$

We proceed now to the controller discretization, considering, at first, the Euler integration method in the *backward* version. Substituting the s variable with the expression indicated in the first row of Table 3.1, we find

$$\mathrm{PI}(z) = K_I \frac{1 + \dfrac{z-1}{z \cdot T_S} \cdot \dfrac{K_P}{K_I}}{\dfrac{z-1}{z \cdot T_S}} = \frac{(K_P + K_I \cdot T_S) \cdot z - K_P}{z-1} = K_P + K_I \cdot T_S \cdot \frac{z}{z-1}. \qquad (A3.2)$$

As can be seen, we have obtained a new rational transfer function that can be simplified to give the discrete time implementation of the PI controller. The block diagram corresponding to the last expression in (A3.2) is shown in Fig. A3.1, which represents the parallel realization of the discrete time regulator, followed by a possible, very simple, model of the calculation delay.

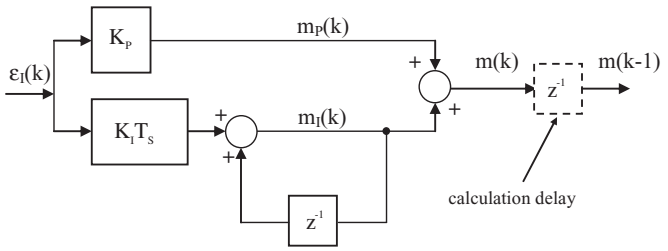

FIGURE A3.1: Block diagram representation of the digital PI controller.

Recalling the basic Z-transform properties, we can immediately write down the control algorithm that may be used to implement the PI regulator in our microcontroller or DSP unit. This is as follows,

$$\begin{cases} m_I(k) = K_I \cdot T_S \cdot \varepsilon_I(k) + m_I(k-1) \\ m(k) = m_P(k) + m_I(k) = K_P \cdot \varepsilon_I(k) + m_I(k), \end{cases} \qquad (A3.3)$$

where $\varepsilon_I(k)$ represents the current error at instant kT_S. Please note that Fig. A3.1 actually represents a more detailed description of the digital PI controller depicted also in Fig. 3.7.

Similarly, we can apply the trapezoidal integration based Z-form, also known as Tustin transform. Following the same procedure above, it is easy to derive the control algorithm that translates the discretized PI controller. We find

$$\begin{cases} m_{\mathrm{I}}(k) = K_{\mathrm{I}} \cdot T_{\mathrm{S}} \cdot \dfrac{\varepsilon_{\mathrm{I}}(k) + \varepsilon_{\mathrm{I}}(k-1)}{2} + m_{\mathrm{I}}(k-1) \\ m(k) = m_{\mathrm{P}}(k) + m_{\mathrm{I}}(k) = K_{\mathrm{P}} \cdot \varepsilon_{\mathrm{I}}(k) + m_{\mathrm{I}}(k). \end{cases} \tag{A3.4}$$

As can be seen, the structure of (A3.4) is similar to that of (A3.3); the only difference being determined by the computation of the integral part that is not based on a single current error value, but rather on the moving average of the two most recent current error samples.

This difference is responsible for the lower frequency response distortion of the Tustin transform. It is worth noting that the proportional and integral gains for the two different versions of the discretized PI controller are exactly the same. As can be seen, in both cases we find that the proportional gain for the digital controller is exactly equal to that of the analog controller, while the digital integral gain can be obtained simply by multiplying the continuous time integral gain and the sampling period. Please note that also the application of prewarping does not change much the values of the controller gains; especially when a relatively high ratio between the sampling frequency and the desired crossover frequency is possible. This is also confirmed by the Bode plots, shown in Fig. A3.2, that refer to each of the different PI controllers we have considered so far, i.e., the original continuous time one and of each of the three discretized versions (Euler, Tustin, and prewarped).

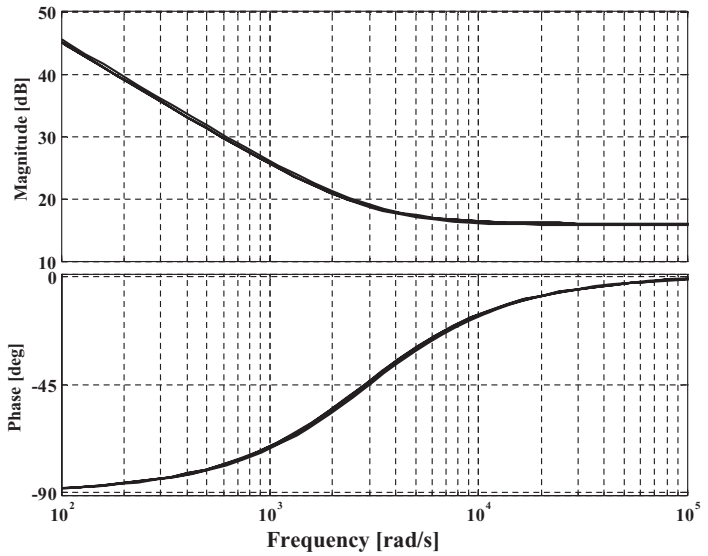

FIGURE A3.2: Bode plots of the different PI realizations.

As can be seen, with our design parameters and sampling frequency, the plots are practically undistinguishable.

In summary, we have seen that, given a suitably designed analog PI regulator, the application of any of the considered discretization strategies simply requires the computation of the digital PI gains, as in the following,

$$\begin{cases} K_{\text{I\_dig}} = K_{\text{I}} \cdot T_{\text{S}} \\ K_{\text{P\_dig}} = K_{\text{P}}, \end{cases} \qquad \textit{(A3.5)}$$

and the implementation of the proper control algorithm (A3.3) or (A3.4).

The last issue we need to discuss is the role of the calculation delay model that appears in Fig. A3.1 (dotted z^{-1} block). If the unit delay block is added to the controller block diagram, it becomes possible to evaluate the effect of the calculation delay on the control performance and the closed loop system stability. This can be done using any kind of system modeling and simulation software. Of course, the duration of the calculation delay is, in this case, supposed to be equal to one sampling period, as a worst-case approximation. More important, the design of the original analog PI controller was performed *neglecting* the calculation delay, so it is likely that its inclusion in the digital controller model, at the time of verification, will significantly affect the dynamic performance. To compensate that, the analog design should be corrected considering an equivalent control loop delay equal to $(3/2)T_{\text{S}}$ in (3.8).

the transformation of the continuous time controller into an *equivalent* discrete time one, the equivalence being in the sense of the integral approximation explained above.

Since the numerical integration methods imply a certain degree of approximation, if we compare the frequency response of the controller before and after discretization, some degree of distortion, also known as frequency warping effect, can always be observed. Table 3.1 also shows the condition that has to be satisfied to make the distortion lower than 3% at a given frequency f. The condition is expressed as a limit for the ratio between the sampling frequency $f_{\text{S}} = 1/T_{\text{S}}$ and the frequency of interest, f. As can be seen, the trapezoidal integration method, which generates the so-called Tustin Z-form, is more precise than the Euler method, and as such guarantees a smaller distortion at each frequency or, equivalently, a higher 3% distortion limit, which is as high as one tenth of the sampling frequency. Ideally, it is also possible to prewarp the controller transfer function so as to compensate the frequency distortion induced by the discretization method and get an exact phase and amplitude match of the continuous time and discrete time controllers at *one* given frequency, which is normally the desired crossover frequency.

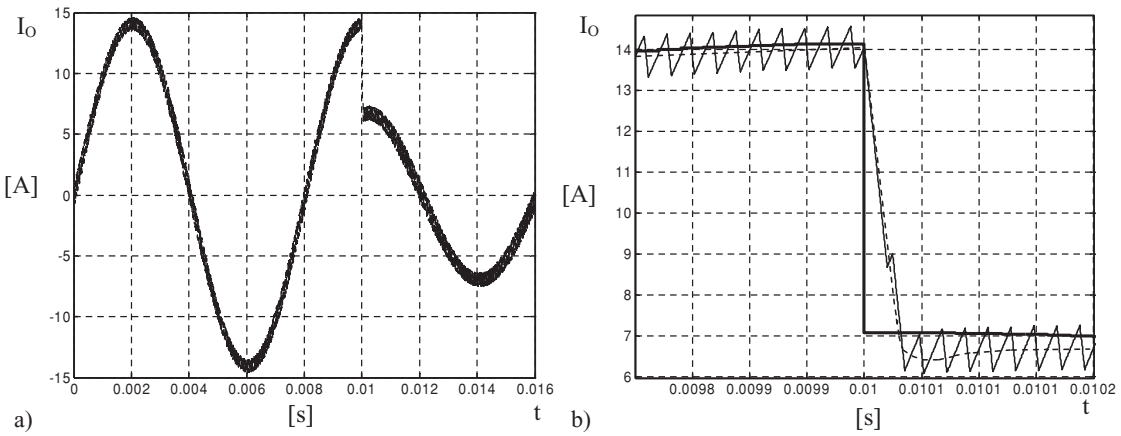

FIGURE 3.11: Simulation of the VSI with the controller designed according to the procedure reported in Aside 3. The depicted variable is the VSI output current I_O. (a) Controller response to a step reference amplitude change. (b) Details of the previous figure. It is possible to see that no calculation delay effect has been included in the simulation.

However, this method implies some more involved calculations and is therefore easily applicable only if we can use some calculation software implementing the discretization techniques. As we show in Aside 3, in the typical application case, the difference in the controller frequency response we can get is usually small, so that the application of discretization methods more complex that the Euler one is seldom motivated, at least for the PI controller.

To conclude the discussion of discretization techniques, we now present the results of the simulation of our VSI with the digital controller obtained by following the procedure outlined in Aside 3 and implementing the algorithm described by (A3.3). These are shown in Fig. 3.11.

It is interesting to compare these results with those reported in Fig. 2.11. As can be seen, there is very little difference in the achieved performance. Watching very carefully, it is however possible to note a slight increase in the phase shift between the output current and its reference, a consequence of the slightly lower bandwidth achieved by the digital controller (a frequency warping effect).

3.2.4 Effects of the Computation Delay

In the above discussion, we have shown how the delay effect associated with the DPWM operation can be taken care of. An additional complication we have to deal with is represented by the fact that the block diagram of Fig. 3.7 actually hides a second, independent source of delay: this is the control algorithm computation delay, i.e., the time required by the processor to compute a new m value, given the input variable sample. Although digital signal processors and microcontrollers are getting faster and faster, in practice the computation time of a digital

current controller always represents a significant fraction of the modulation period, ranging typically from 10% to 40% of it. A direct consequence of this hardware limitation is that, in general, we cannot compute the input to the modulator during the same modulation period when it has to be applied. In other words, the modulator input, in any given modulation period, must have been computed during the previous control algorithm iteration. Dynamically, this means that the control algorithm actually determines an additional one modulation period delay.

One could consider this analysis to be somewhat pessimistic, because powerful microcontrollers and DSPs are available today, which allow the computation of a PID routine in much less than a microsecond. However, it is important to keep in mind that, in industrial applications, the cost factor is fundamental: cost optimization normally requires the use of the minimum hardware that can fulfill a given task. The availability of hardware resources in excess, with respect to what is strictly needed, simply identifies a poor system design, where little attention has been paid to the cost factor. Therefore, the digital control designer will struggle to fit his or her control routine to a minimum complexity microcontroller much more often than he or she will experience the opposite situation, where a high-speed DSP will be available just for the implementation of a digital PI or PID controller.

The conventional approach to tackle the problem consists in assuming that a whole control period is dedicated to computations, as shown in Aside 3, Fig. A3.1. In this case, in order to get from the digital controller a satisfactory performance the calculation delay effect has to be included from the beginning in the analog controller design. Practically, this can be done by increasing the delay effect represented by the Padé approximation of Fig. 2.10 and Fig. 3.9 by T_S. After that, the procedure described in Aside 3 for the controller synthesis through discretization can be reapplied. It is important to underline once more that, if the analog controller is not redesigned and a significant calculation delay is associated with the implemented algorithm, the achieved performance can be much less than satisfactory. An example of this situation is shown in Fig. 3.12(a), where a calculation delay equal to one modulation period is considered. Note how the step response tends to be underdamped. In the other case instead, as is shown in Fig. 3.12(b), the dynamic response of the redesigned controller is smoother, but a significant reduction of its speed can be observed. Please note that the result has been obtained by reducing the crossover frequency to $f_S/15$, while keeping the same phase margin of the original design. The previous example shows that when the maximum performance is required, this conventional approach may be excessively conservative. Penalizing the controller bandwidth to cope with the computation delay, the synthesis procedure will unavoidably lead to a worse performance, with respect to conventional analog controllers. This is the reason why, in some cases, a different modeling of the digital controller can be considered that takes into

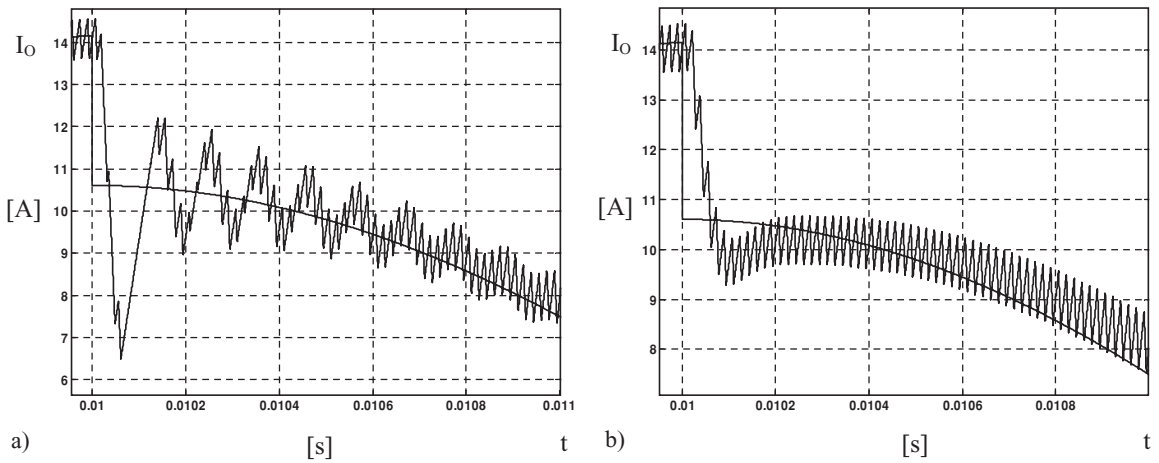

FIGURE 3.12: Simulation of the VSI with the digital PI controller including the calculation delay. (a) Details of the controller response to a step reference amplitude change without redesign: undershoot and oscillating response. (b) Details of controller response with redesign: reduced undershoot, reduced speed of response, increase of phase shift.

account the exact duration of the computation delay and so, by using modified *Z*-transform, exactly models the duty-cycle update instant within the modulation period. In this way, the penalization of the digital controller with respect to the analog one can be minimized and a significant performance improvement, with respect to the case of Fig. 3.12(b), can be achieved. This will be the subject of Section 3.2.6.

3.2.5 Derivation of a Discrete Time Domain Converter Dynamic Model

What we have described so far is a very simple digital controller design approach. It is based on the transformation of the sampled data system into a continuous time equivalent, which is used to design the regulator with the well-known continuous time design techniques. The symmetrical approach is also possible. In this case, the sampled data system is transformed into a discrete time equivalent, which can be used to design the controller directly in the discrete time domain. We will now present a short review of this strategy.

Discrete time models for power electronic circuits have been widely discussed in the past (see, for example, [6–8]). The detailed and precise discrete time converter model is generally based on the integration of the linear and time-invariant state space equations, associated with each switch configuration (i.e., turn-on and turn-off). Then, the state variable time evolutions, obtained separately for each topological or switch state, are linked to one another exploiting the continuity of the state variable, i.e., imposing the final state of one configuration to be the

initial state of the next. This approach, which requires the use of exponential matrixes, leads to a general discrete time state space model and precisely represents the system dynamic behavior in the discrete time domain. Therefore, in principle, it represents a very good modeling approach for digitally controlled power electronic circuits. Nevertheless, it is not very commonly used, mainly for the following two reasons: (i) the obtained discrete time model depends on the particular type of modulator adopted, as the sequence of state variable integrations, one for each topological state, depends on the modulator mode of operation (leading edge, trailing edge, etc.); (ii) the exponential matrix computation is relatively complex and, therefore, not always practical for the design of power electronic circuit controllers.

A more direct, equivalent, approach to discrete time converter modeling is described in Figs. 3.13 and 3.14(a), where the PWM modulator is represented using the frequency domain model, PWM(s), derived in the previous chapter, $G(s)$, the converter transfer function, is obtained from the continuous time converter small signal model, and $x^s(t)$ is the sampled output variable, which has to be controlled by the digital algorithm. To account for the time required by the AD conversion and by the control algorithm computation in the DSP (or μC), a time delay T_d is cascaded to the controller transfer function Reg(z). More explicitly, in a uniformly sampled PWM, time T_d represents the delay between the output variable sampling and the duty-cycle update instants. When this is equal to one modulation period, a simple z^{-1} block could be substituted in the control loop.

Aside 4. PI Current Controller with Integral Anti-wind-up

In Aside 3, we have completed the design of a digital PI current controller. This Aside is dedicated to a typical implementation issue, i.e., the control of the integral part wind-up. This phenomenon can take place any time the PI controller input signal, i.e., the regulation error, is different from zero for relatively large amounts of time. This typically happens in the presence of large reference amplitude variations or other transients, causing inverter saturation. The problem is determined by the fact that, if we do not take any countermeasure, the integral part of the controller will be accumulating the integral of the error for the entire transient duration. Therefore, when the new set-point is reached, the integral controller will be very far from the steady state and a transient will be generated on the controller variable, which typically has the form of an overshoot. It is fundamental to underline that this overshoot is not related to the small signal stability of the system. Even if the phase margin is high enough, the transient will always be generated, as it is just due to the way the integral controller reacts to converter saturation. An example of this problem is shown in Fig. A4.1(a).

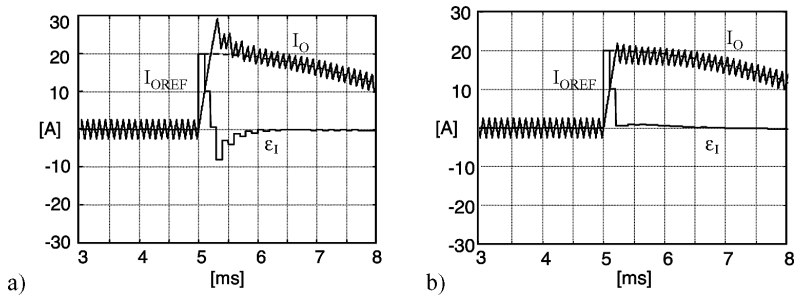

FIGURE A4.1: Dynamic behavior of the PI controller during saturation: (a) no anti-wind-up; (b) anti-wind-up.

The solution to this problem is based on the dynamic limitation of the integral controller output during transients. Transients can be detected monitoring the output of the controller proportional part: in a basic implementation, any time this is higher than a given limit, the output of the integral part of the controller can be set to zero. Integration is resumed only when the regulated variable is again close to its set-point, i.e., when the output of the proportional part gets below the specified limit. More sophisticated implementations of this concept are also possible, where the limitation of the integral part is done gradually, for example keeping the sum of the proportional and integral outputs in any case lower than or equal to a predefined limit. In this case, shown in Fig. A4.2, at each control iteration, a new limit for the integral part is computed and, if needed, the integral output is clamped.

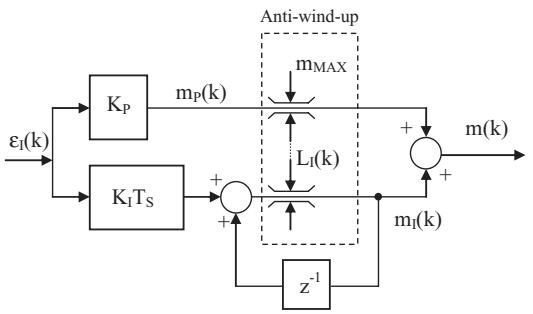

FIGURE A4.2: Block diagram representation of the digital PI controller with anti-wind-up action.

This implementation, of course, requires a slightly higher computational effort, which amounts to the determination of the following quantity, where m_{MAX} is the controller output limit:

$$|L_I(k)| = m_{MAX} - |K_p \varepsilon_I(k)|. \qquad (A4.1)$$

However, the result can be quite effective, as shown in Fig. A4.1(b). Please note that similar provisions can be as well adopted to limit the state variables of any other type of digital regulator.

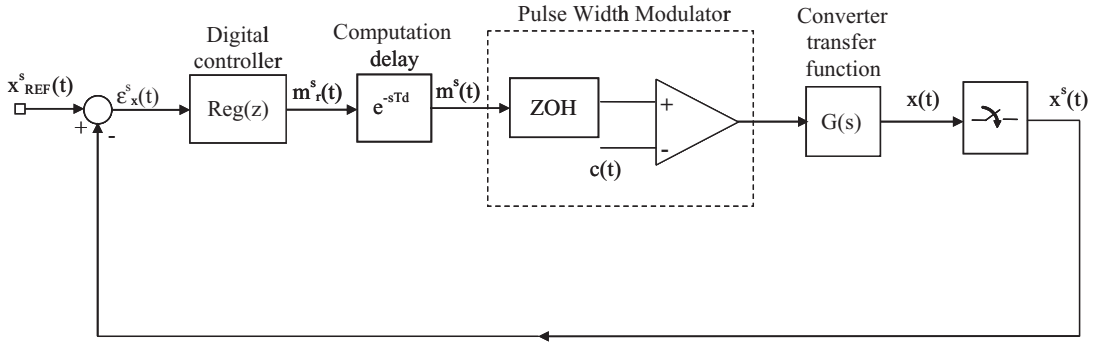

FIGURE 3.13: Model of the control loop for digitally controlled converters.

Before continuing with this discussion, we have to clarify two key points, fundamental to establish a correct relationship between our modeling approach and standard digital control theory. First of all, the zero-order hold (ZOH) function that, when cascaded to an ideal sampler, models the conversion from sampled time variables into continuous time variables is, in our case, internal to the PWM model and, therefore, does not appear right after the sampler. As a consequence, recalling that an ideal sampler has, by itself, a gain equal to $1/T_S$ [1], if we want to correctly represent the transfer function between the sampled time input variable and the continuous time output variable of the modulator, a gain equal to T_S has to be added to the modulator transfer function PWM(s). Having clarified this, the discrete time transfer function $G_T(z)$, which exactly represents the discrete time state variable dynamic equations is given by

$$G_T(z) = Z[e^{-s\,T_d}\,T_S\,\mathrm{PWM}(s)G(s)]. \tag{3.9}$$

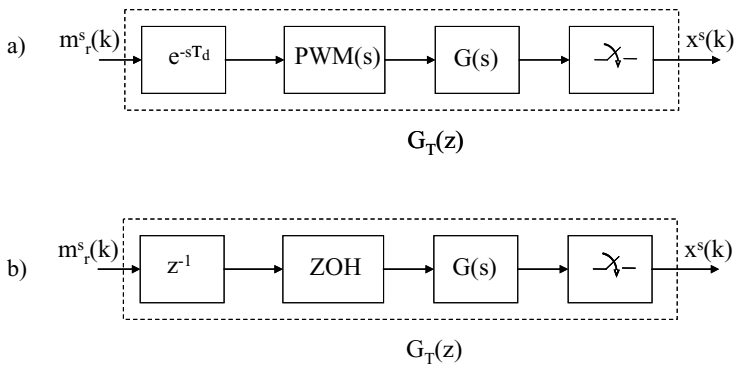

FIGURE 3.14: Equivalent dynamic model of computation delay, the PWM transfer function, the converter, and the sampler: (a) general form, (b) simplified representation, where the PWM is approximated as a zero-order hold (ZOH) and the control delay is equal to one modulation period.

This z-domain approach is very powerful: indeed it is capable of correctly quantifying the difference in the converter dynamics determined by the different uniformly sampled modulator implementations (trailing edge, leading edge, triangular carrier modulation, etc.), as it takes into account the exact duty-cycle update instant. Nevertheless, there are two strong motivations to simplify the discretization process and the evaluation of (3.9) in the case of triangular carrier modulation: (i) the control delay is usually equal to one modulation period, and a simple z^{-1} block can be used to represent it; (ii) the PWM modulation transfer function $T_S \cdot \mathrm{PWM}(s)$ looks very much like that of a zero-order hold, as (3.7) clearly shows. Therefore, an approximated, but more intuitive, ZOH discretization method can be used to obtain the open loop discrete time transfer function. This is given by

$$G_T(z) = z^{-1} Z[H(s)G(s)], \tag{3.10}$$

where $H(s) = (1 - e^{-s T_S})/s$ is the ZOH transfer function. Moreover, assuming that $G(s)$ is well approximated by a pure integrator, as is the case of our current control example with $R_S = 0$, i.e., $G(s) = 2\ V_{DC}/s\ L_S$, and assuming $T_d = T_S$, there is no difference between (3.9) and (3.10). In fact, rewriting (3.10) we find that

$$G_T(z) = z^{-1} Z\left[\frac{1 - e^{-s T_S}}{s}\ \frac{2\ V_{DC}}{s\ L_S}\right] = \frac{2\ V_{DC}}{L_S} z^{-1}(1 - z^{-1}) Z\left[\frac{1}{s^2}\right] = \frac{2\ V_{DC}\ T_S}{L_S} \frac{1}{z(z-1)}, \tag{3.11}$$

while, rewriting (3.9), we get

$$G_T(z) = Z\left[e^{-s T_S}\ T_S\ \mathrm{PWM}(s)\ \frac{2\ V_{DC}}{s\ L_S}\right] = \frac{2\ V_{DC}\ T_S}{L_S} z^{-1} Z\left[\frac{\mathrm{PWM}(s)}{s}\right] = \frac{2\ V_{DC}\ T_S}{L_S} \frac{1}{z(z-1)}, \tag{3.12}$$

where $\mathrm{PWM}(s)$ is the transfer function given by (2.7), with $c_{PK} = 1$. The equivalence between the two approaches is easily justified if we consider that the output current variation only depends on the integral of the inverter voltage, i.e., only on the average voltage value generated by the PWM, and not on the particular allocation of the PWM pulse within the modulation period.

Following the same reasoning, the extension of the z-domain small signal model derivation to the case of the multisampled system, described in Section 2.2.4, is straightforward,

$$G_T(z) = Z[e^{-s T_d} \frac{T_S}{N} \mathrm{PWM}(s)G(s)], \tag{3.13}$$

where the Z-transform is taken with a sampling period equal to T_S/N. Model (3.12) can be used for the direct discrete time design of the current controller, simply deciding the closed loop poles allocation.

3.2.6 Minimization of the Computation Delay

As previously described, one of the most important factors that limit the dynamic performance of digitally controlled power converters is the computational delay between the sampling instant and the duty-cycle update instant. We have previously described how, in order to avoid antialiasing filters, a common practice is to sample the inductor current in the middle of either the turn-on or the turn-off times, thus ensuring that its average value is automatically acquired. However, this provision usually introduces a delay in the control loop, strongly limiting the achievable bandwidth. The control delay can be reduced by half in the double update mode, where the input variables are sampled both in the middle of turn-on and turn-off time and the duty-cycle is updated twice in one switching period.

However, the increase of computational power of DSPs, microcontrollers, and FPGAs, which are now able to complete the control algorithm computation in smaller and smaller fractions of the switching period, makes possible the further reduction of the control delay. This can be obtained shifting the current sampling instant toward the duty-cycle update instant, leaving just enough time for the ADC to generate the new input sample and to the processor for the control algorithm calculation.

The situation under investigation is depicted in Fig. 3.15, where T_d is, once again, the time required by AD conversion and calculations. Time T_C is instead available for other noncritical functions or external control loops. As can be seen, since $T_d \ll T_S$, with T_S being the modulation period, the sampling of the state variable $x(t)$, i.e., in our case of the inductor current $I_O(t)$, is delayed with respect to conventional controller organizations and shifted toward the PWM update instant, as shown in Fig. 3.15. From the controller's standpoint, this implies a reduction of the feedback loop delay.

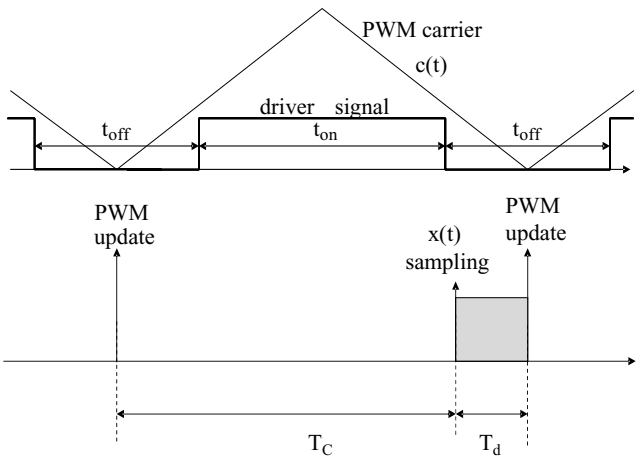

FIGURE 3.15: Sampling of variable $x(t)$ shifted toward the PWM update.

In order to quantify the effectiveness of this reduction, an accurate discrete time model is needed. To this purpose, we can consider the block diagram of Fig. 3.14(a), and replace the PWM block with a zero-order hold (ZOH), which, as we have seen, represents a very good approximation, especially in the case of triangular carrier waveform. Now, if the control delay T_d is a submultiple of the sampling period T_S, the continuous system is easily convertible into a discrete time model using conventional Z-transform and considering T_d as the sampling period. In our case, the delay T_d is a generic fraction of sampling period T_S and therefore, modified Z-transform has to be used to correctly model the system. The basics of modified Z-transform are briefly recalled hereafter. Let us define

$$p = 1 - \frac{T_d}{T_S} \tag{3.14}$$

where $0 \leq p \leq 1$. If $g(t)$ is the impulse response of $G(s)$, we denote, as we did before, the Z-transform of the ideally sampled version of $g(t)$ (i.e., $Z[L^{-1}[G(s)](k\,T_S)]$) simply as $Z[G(s)]$, with L^{-1} being the inverse of Laplace transform. Consequently, the discrete time model of the continuous system of Fig. 3.7 can be expressed as

$$Z\left[\underbrace{H(s)\,G(s)}_{G_1(s)}\,\mathrm{e}^{-s\,p\,T_S}\right] = \sum_{k=0}^{\infty} z^{-k} g_1(k\,T_S - T_d) = Z_m\,[G_1(s)] = G_1(z,\,p) \tag{3.15}$$

where $g_1(t)$ is the impulse response of $G_1(s)$, and $G_1(z,\,p)$ (or $Z_m\,[G_1(s)]$) is the modified Z-transform of $G_1(s)$. In the particular case of the zero-order hold, $H(s) = (1 - \mathrm{e}^{-s\,TS})/s$ and (3.15) becomes

$$G_T(z,\,p) = Z\left[\underbrace{\frac{1 - \mathrm{e}^{-s\,T_S}}{s}}_{H(s)}\,G(s)\mathrm{e}^{-s\,p\,T_S}\right] = \frac{z-1}{z}\,Z\left[\frac{G(s)}{s}\,\mathrm{e}^{-s\;p\;T_S}\right] = \frac{z-1}{z}\,Z_m\left[\frac{G(s)}{s}\right]. \tag{3.16}$$

The modified Z-Transform maintains the properties of the conventional Z-transform, since it is simply defined as the Z-transform of a delayed signal; see (3.15). The results of the modified Z-transform application to particular cases of interest are usually available in lookup tables [9].

In our example case, the discrete time transfer function between the modulating signal $M(z)$, input of the DPWM, and the *delayed* inductor current $I_O(z)$ can be written as

$$\frac{I_O(z)}{M(z)} = \frac{2\,V_{DC}\,T_S}{L_S} \cdot \frac{z\,p - (p-1)}{z(z-1)}. \tag{3.17}$$

TABLE 3.2: Achievable Current Loop Bandwidth BW_i Versus p[a]

p	0	0.5	0.8
BW_i	$f_s/13.4$	$f_s/9$	$f_s/6.2$

[a] Phase margin is 50° and the current regulator is purely proportional.

It may be interesting to observe that in the usual case, where $p = 0$, i.e., the sampling and computation delay amounts to one full modulation period, (3.17) reduces to

$$\frac{I_O(z)}{M(z)} = \frac{2\,V_{DC}\,T_S}{L_S} \cdot \frac{1}{z(z-1)}, \tag{3.18}$$

which, as can be verified, is equal to (3.11) and (3.12). In order to quantify the advantages of exactly modeling the delay, i.e., of considering $p > 0$, let us take into account, as a benchmark parameter, the achievable current loop bandwidth. We assume, for simplicity, that the current regulator is purely proportional and that the phase margin is kept constant, for example to +50°. To this purpose, we look for the frequency where (3.17) shows a −130° phase rotation, and we define that as the achievable current loop bandwidth (BW_i). The results are reported in Table 3.2. Please note how, simply by shifting the sampling instant toward the duty-cycle update instant, a significant improvement in the achievable current loop bandwidth can be obtained. It is also possible to note that only with $p = 0$ (sampling in the middle of turn-off time) or $p = 0.5$ (sampling in the middle of turn-on time) the sampled current is the average inductor current, while, for other values of p, some kind of algorithm is needed for the compensation of the current ripple, possibly accounting for dead-time effects as well. For this reason, the application of the concept here described to current control is fairly complicated, while it can be much more convenient for the control of other system variables, where the switching ripple is smaller. This can be the case, for example, of output voltage control in an Uninterruptible Power Supply, a particular application of the VSI we will discuss in Chapter 5.

3.2.7 The Predictive Controller

We now move to a totally different control approach, describing the predictive, or *dead-beat*, current control implementation [10, 11]. The basic organization of this controller is shown in Fig. 3.16: it closely resembles the one shown in Fig. 3.7 with two major differences, the presence of an additional input to the controller and the absence of the delay block modeling the sample and hold process. The motivations for these differences will be given shortly.

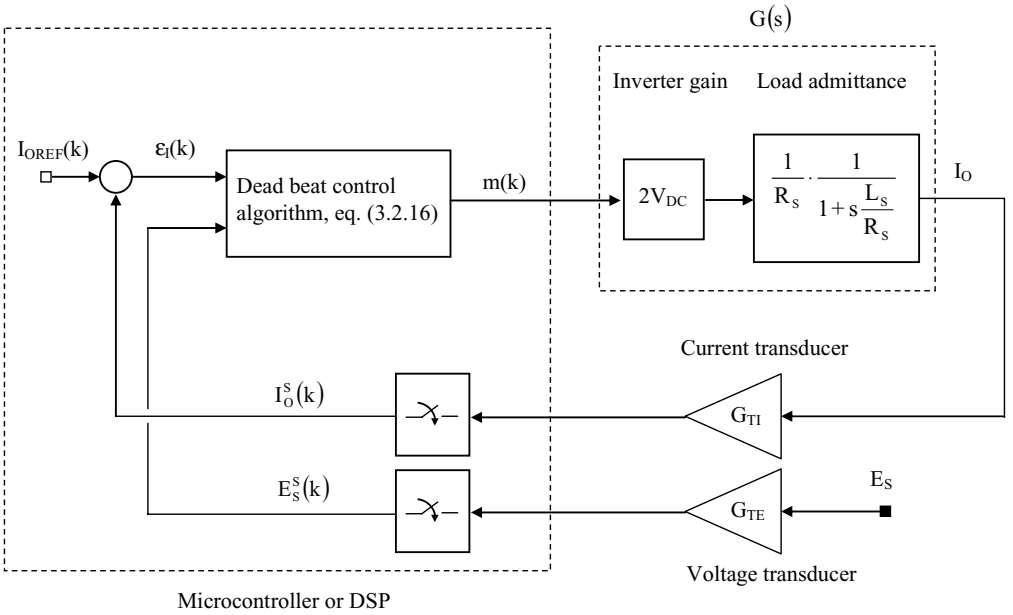

FIGURE 3.16: Block diagram of the dead-beat current control loop.

In principle, the dead-beat control strategy we are going to discuss is nothing but a particular application case of discrete time dynamic state feedback and direct pole allocation, and, as such, its formulation for our VSI model, as derived in Aside 1, can be obtained applying standard digital control theory. However, this *theoretical* approach is not what we are going to follow here. Instead, we will present a different derivation, completely equivalent to the theoretical one, but closer to the physical converter and modulator operation. We will discuss the equivalence of the two approaches in Aside 5, but we feel that the physical one is somewhat easier to explain and better puts into evidence the merits and limitations of the predictive controller. For this reason, we chose to begin our discussion exactly from the *physical* approach.

3.2.7.1 Derivation of the Predictive Controller

The reasoning behind the physical approach to predictive current control is quite simple and can be explained referring to Fig. 3.17, which represents an average model of the VSI and its load. At any given control iteration, we want to find the average inverter output voltage, $\overline{V}_{OC}$, that can make the average inductor current, $\overline{I}_O$, reach its reference by the end of the modulation period *following* the one when all the computations are performed. In other words, at instant $k \cdot T_S$ we perform the computation of the $\overline{V}_{OC}$ value that, once generated by the inverter, during the modulation period from $(k+1) \cdot T_S$ to $(k+2) \cdot T_S$, will make the average current equal to its reference at instant $(k+2) \cdot T_S$. Please note that, doing so, the computation and modulation

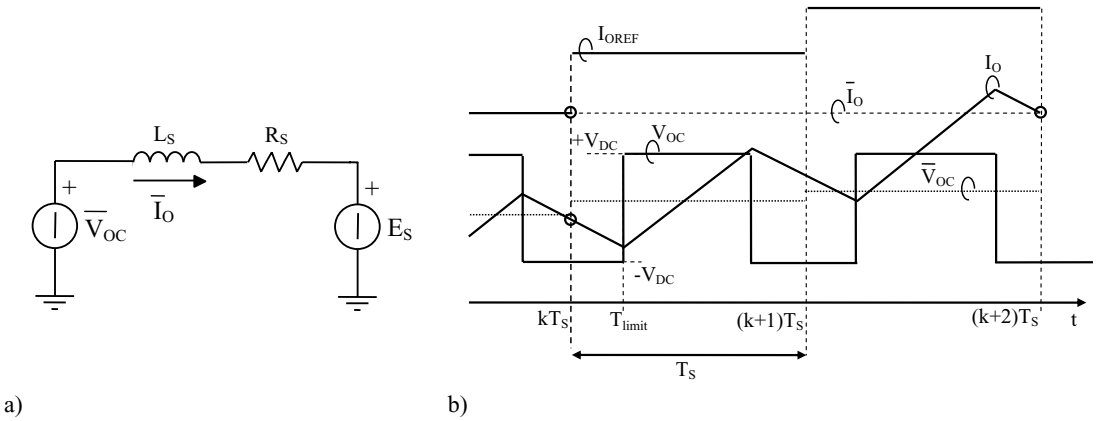

FIGURE 3.17: (a) Average equivalent circuit for the VSI of Fig. 3.1. (b) Average and instantaneous idealized waveforms for the inverter of Fig. 3.1.

delays are inherently taken care of, and the controller dynamic response, as we are going to show, turns out to be equivalent to a pure two modulation period delay.

Following this reasoning, the control equation can be easily derived. Examining Fig. 3.17(a) the average inductor current at the instant $(k+1)T_S$ is given by

$$\overline{I}_O(k+1) = \overline{I}_O(k) + \frac{T_S}{L_S} \cdot \left[\overline{V}_{OC}(k) - E_S(k)\right], \qquad (3.19)$$

where resistance R_S has been considered negligible, as it is often the case. Equation (3.19) simply expresses the physical fact that the current variation in an inductor is proportional to the integral of the applied inductor voltage. This voltage integral has been computed exploiting, once again, the Euler numerical integration rule. In other words, we are here considering again a zero-order hold discrete time equivalent of the dynamic system represented in Fig. 3.17(a). Please note that, in (3.19) and according to Fig. 3.17(b), $\overline{V}_{OC}(k)$ indicates the average inverter voltage to be generated in the modulation period *following* the sampling instant $k \cdot T_S$, when all calculations are performed.

In principle, from (3.19) it would be possible to compute the $\overline{V}_{OC}(k)$ value required to make $\overline{I}_O(k+1)$ equal to $I_{OREF}(k)$, thus achieving a one cycle delay dynamic response for the closed loop controlled system. In practice, since the computation of voltage $\overline{V}_{OC}(k)$ value occupies part of the modulation period, it is not possible to guarantee that, in all cases, the calculations will be over before the output voltage has to change its state from negative to positive, instant indicated by T_{limit} in Fig. 3.17(b). Please note that, in all cases when the average voltage to be applied is strongly positive, instant T_{limit} will be very close to instant $k \cdot T_S$, thus leaving very little time for computations.

The simplest way to solve these timing problems is to move the objective of the computation one step forward, i.e., instead of computing $\overline{V}_{OC}(k)$, we will now find the expression of $\overline{V}_{OC}(k+1)$. This will give us a whole modulation period to complete the calculations. Please note that we followed exactly the same approach when we modeled the calculation delay for the digital PI controller, considering it to be equal to one modulation period. Also similarly to the case of the digital PI, more sophisticated modeling approaches, taking into account the exact computation delay and duty-cycle update instant allocation within the modulation period are indeed possible, but we will not consider them here. Therefore, rewriting (3.19) one step forward, we get

$$\begin{aligned}\overline{I}_O(k+2) &= \overline{I}_O(k+1) + \frac{T_S}{L_S}\cdot\left[\overline{V}_{OC}(k+1) - E_S(k+1)\right] \\ &= \overline{I}_O(k) + \frac{T_S}{L_S}\cdot\left[\overline{V}_{OC}(k+1) + \overline{V}_{OC}(k) - E_S(k+1) - E_S(k)\right], \quad (3.20)\end{aligned}$$

where $\overline{I}_O(k+1)$ has been replaced by its expression (3.19). Assuming now that the phase voltage E_S is a slowly varying signal, as it is often the case, whose bandwidth is much lower than the modulation and sampling frequency, it is possible to consider $E_S(k+1) \cong E_S(k)$, thus obtaining the following dead-beat control equation,

$$\overline{V}_{OC}(k+1) = -\overline{V}_{OC}(k) + \frac{L_S}{T_S}\cdot\left[\overline{I}_O(k+2) - \overline{I}_O(k)\right] + 2\cdot E_S(k), \quad (3.21)$$

where $\overline{I}_O(k+2)$ can be replaced by $I_{OREF}(k)$, the desired set-point. Equation (3.21) can be used to determine the duty-cycle, for the modulation period starting at instant $(k+1)\cdot T_S$, that will make the inductor current reach its reference at instant $(k+2)\cdot T_S$, with a two modulation period delay. If this holds, and indeed it does, application of (3.21) makes the closed loop system dynamic response equivalent to a pure delay, i.e., guarantees a dead-beat control action.

As it is possible to observe, the application of (3.21) requires the phase voltage E_S to be measured every sampling period, so that, differently from the PI current controller, the predictive controller, at least in this basic implementation, requires the sensing and analog to digital conversion not only of the regulated variable, i.e., the output current, but also of the phase voltage.

Another, less evident, point regarding (3.21) is that, in general, the set-point for the average inverter output voltage it provides us with will have to be correctly scaled down, so as to fit it to the digital pulse width modulator. The fitting is normally accomplished *normalizing* the output of the controller to the inverter voltage gain. In addition to this, the control equation has to be modified also to properly account for the transducer gains of both current and voltage sensors. It is easy to verify that an equivalent control equation, taking into account the transducer

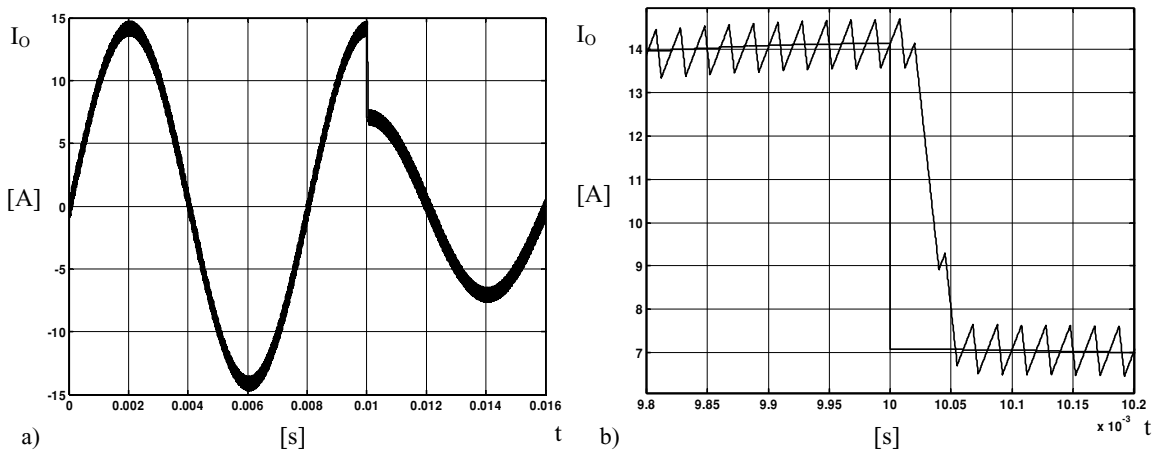

FIGURE 3.18: Simulation of the VSI with the predictive controller. The depicted variable is the VSI output current I_O. (a) Controller response to a step reference amplitude change. (b) Details of the previous figure.

gains and voltage normalization, is as follows,

$$m(k+1) = -m(k) + \frac{L_S}{T_S} \cdot \frac{1}{2 \cdot G_{TI} \cdot V_{DC}} [I^S_{OREF}(k) - I^S_O(k)] + 2 \cdot \frac{1}{2 \cdot G_{TE} \cdot V_{DC}} E^S_S(k), \tag{3.22}$$

where $m(k)$ is the modulating signal input of the digital PWM and all variables are now *internal* variables, properly scaled down to fit to the microprocessor arithmetic unit. Please note that (3.22) also assumes that the modulating signal is bipolar, ranging between plus and minus one-half of the modulator full scale input. As we did for the digital PI, this is assumed to be equal to unity, without any loss of generality. Under these assumptions, (3.22) can be turned into a control algorithm, to be programmed in the microcontroller or DSP unit.

An example of the predictive controller dynamic performance is shown in Fig. 3.18. In particular, Fig. 3.18(a) shows the reference tracking capability of the controller in the presence of a step change in the current reference. A more detailed view of the transient is shown in Fig. 3.18(b). It is interesting to compare this plot with Figs. 3.11 and 3.12. As can be seen, in spite of the calculation delay, the dynamic response is faster than that obtained with the PI controller and reference tracking is resumed almost exactly after only two modulation periods from the first controller intervention.

A simple improvement of the presented control strategy makes it possible to derive an estimation equation that allows us to save the measurement of the phase voltage E_S. As in the control equation's case, the estimation equation can be derived by simple physical considerations, basically referring to (3.19). Indeed, rewriting (3.19) one step backward we get

$$\overline{I}_O(k) - \overline{I}_O(k-1) = \frac{T_S}{L_S} \cdot [\overline{V}_{OC}(k-1) - E_S(k-1)], \tag{3.23}$$

from which we can extract an estimation of $E_S(k-1)$. Simple manipulations of (3.23) yield

$$\hat{E}_S(k-1) = \overline{V}_{OC}(k-1) - \frac{L_S}{T_S} \cdot [\overline{I}_O(k) - \overline{I}_O(k-1)]. \tag{3.24}$$

Equation (3.24) represents the basic estimation equation for the predictive control of the VSI output current. It is typically possible to improve the quality of the estimation by using some form of interpolation or filtering that can remove possible estimator instabilities.

3.2.7.2 Robustness of the Predictive Controller

The predictive controller derivation assumes that Eq. (3.19) is a valid model of the VSI and its load. Although this is a generally solid assumption, in certain conditions the validity of (3.19) can be impaired. There can be at least two different reasons for this to happen, namely model mismatches and parameter uncertainties.

Aside 5. Derivation of the Predictive Controller Through Dynamic State Feedback

The VSI represented in Fig. 3.1 can be described in the state space by the following set of equations,

$$\begin{cases} \dot{x} = Ax + Bu \\ y = Cx + Du, \end{cases} \tag{A5.1}$$

which, as we recall from the discussion reported in Aside 1, can be used to relate average inverter electrical variables. In this case $x = [\overline{I}_O]$ is the state vector, $u = [\overline{V}_{OC}, E_S]^T$ is the input vector, $y = [\overline{I}_O]$ is the output variable, and the state matrixes are

$$A = [-R_S/L_S], \quad B = [1/L_S, -1/L_S], \quad C = [1], \quad D = [0, 0]. \tag{A5.2}$$

It is possible to derive a zero-order hold discrete time equivalent of (A5.1) considering the following system,

$$\begin{cases} x(k+1) = \Phi x(k) + \Gamma u(k) \\ y(k) = Cx(k) + Du(k), \end{cases} \tag{A5.3}$$

where, by definition, $\Phi = e^{A \cdot T_S}$ and $\Gamma = (\Phi - I) \cdot A^{-1} \cdot B$. Computation of Φ and Γ yields

$$\begin{aligned} \Phi &= e^{A \cdot T_S} = e^{-\frac{R_S}{L_S} \cdot T_S} \xrightarrow{R_S \to 0} 1 \\ \Gamma &= \begin{bmatrix} -\dfrac{e^{-\frac{R_S}{L_S} \cdot T_S} - 1}{R_S} & \dfrac{e^{-\frac{R_S}{L_S} \cdot T_S} - 1}{R_S} \end{bmatrix} \xrightarrow{R_S \to 0} \begin{bmatrix} \dfrac{T_S}{L_S} & -\dfrac{T_S}{L_S} \end{bmatrix}, \end{aligned} \tag{A5.4}$$

where both matrixes have been calculated for the limit condition where the R_S value is negligible. Of course, this approximation is not strictly necessary to perform the following calculations and could be avoided. However, since we want to compare the results provided

by the theoretical approach with those provided by the physical approach, we need to operate under the same conditions, which motivates the assumption of a negligible R_S value.

Given (A5.3), we can derive the predictive controller as a particular case of state feedback and pole placement. In order to show that we may consider again (A5.3), rewriting the state equations explicitly. We get the following result:

$$\Sigma : \begin{cases} \overline{I}_O(k+1) = \overline{I}_O(k) + \dfrac{T_S}{L_S} \cdot \overline{V}_{OC}(k) - \dfrac{T_S}{L_S} \cdot E_S(k) \\ y(k) = \overline{I}_O(k). \end{cases} \qquad (A5.5)$$

Please note that the first equation in (A5.5) is exactly equal to (3.19). It is easy to verify that (A5.5) is exactly equivalent to the part of the following block diagram indicated by Σ. As can be seen, the block diagram includes the feedback controller as well. This schematic representation puts into evidence some interesting features of the considered discrete time system. In the first place, the diagram reveals how the E_S input can be considered an exogenous disturbance, whose compensation can be obtained by adding a suitable signal, ideally the $\overline{E}_S$ signal itself, to the control input $\overline{V}_{OC}$. As we will see in the following, the computation delay will make it impossible to get a perfect compensation of the disturbance, and, consequently, only partial compensation will be achieved. In addition, the block diagram shows how the feedback controller is itself a *dynamic system*. Differently from what is often done in state feedback applications, we are using here *dynamic state feedback* instead of a simple *static* feedback.

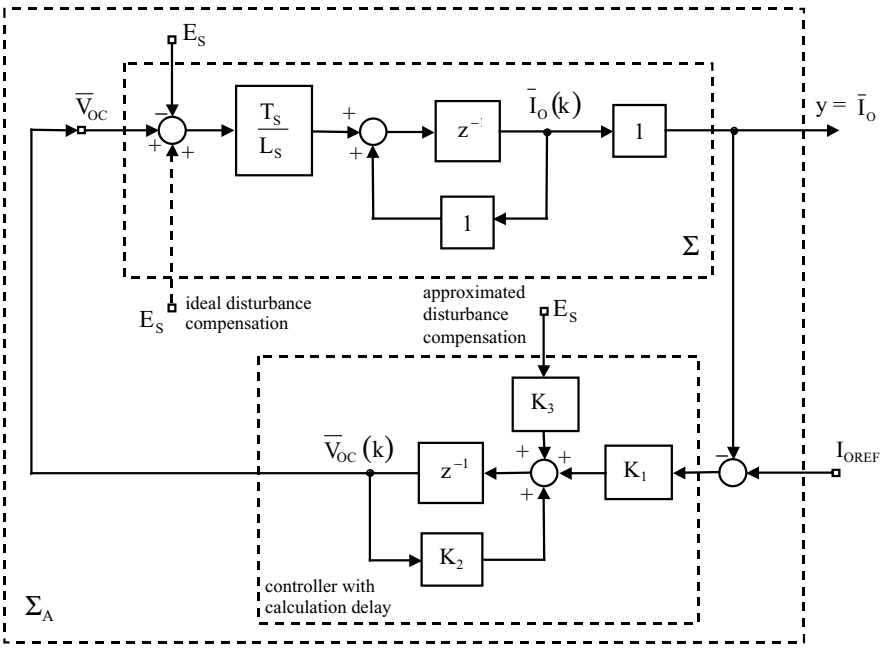

FIGURE A5.1: Block diagram of the discrete time models for the VSI and the dead-beat controller.

This is done to make the modeling of the computation delay more direct. In this case, referring to the diagram, we can represent the controller by means of the following state equation,

$$\overline{V}_{OC}(k+1) = K_2 \cdot \overline{V}_{OC}(k) + K_1 \cdot [I_{OREF}(k) - \overline{I}_O(k)], \qquad (A5.6)$$

where, of course, the identity $\overline{I}_O = y$ has been used. The interconnection of Σ and the controller feedback generates a new, augmented, dynamic system, indicated by Σ_A. This is described by the following equations,

$$\Sigma_A : \begin{cases} \overline{I}_O(k+1) = \overline{I}_O(k) + \dfrac{T_S}{L_S} \cdot \overline{V}_{OC}(k) - \dfrac{T_S}{L_S} \cdot E_S(k) \\ \overline{V}_{OC}(k+1) = K_2 \cdot \overline{V}_{OC}(k) + K_1 \cdot [I_{OREF}(k) - \overline{I}_O(k)] + K_3 \cdot E_S(k) \\ y(k) = [1\ 0] \cdot \begin{bmatrix} \overline{I}_O(k) \\ \overline{V}_{OC}(k) \end{bmatrix}, \end{cases} \qquad (A5.7)$$

which correspond to the state vector augmentation to $x_A = [\overline{I}_O\ \overline{V}_{OC}]^T$, to the new input vector $u_A = [E_S\ \ I_{OREF}]^T$ and to the *approximated* compensation of the exogenous disturbance, governed by gain K_3. The corresponding state matrixes are as follows:

$$\Phi_A = \begin{bmatrix} 1 & \dfrac{T_S}{L_S} \\ -K_1 & K_2 \end{bmatrix}, \quad \Gamma_A = [\Gamma_{A1} | \Gamma_{A2}] = \left[\begin{array}{c|c} -\dfrac{T_S}{L_S} & 0 \\ K_3 & K_1 \end{array}\right], \quad C_A = [1 \quad 0], \quad D_A = [0 \quad 0]. \qquad (A5.8)$$

Based on (A5.8) it is possible to determine parameters K_1, K_2, and K_3 to get the desired pole allocation and disturbance compensation. It is easy to verify that choosing

$$K_1 = \frac{L_S}{T_S}, \quad K_2 = -1 \qquad (A5.9)$$

the eigenvalues of Σ_A move to the origin of the complex plane. As is well known, this is a sufficient condition to achieve a dead-beat closed loop response from the controlled system. Alternatively, the position of poles on the complex plane can be chosen to achieve a different closed loop behavior, for example one equivalent to that of a continuous time, first-order, stable system, characterized by any desired time constant. Indeed, with the direct discrete time design of the regulator, the designer has, in principle, complete freedom in choosing the preferred pole allocation. It is possible to demonstrate that the allocation of poles in the origin of the complex plane makes the closed loop system behavior very peculiar, bearing no similarity with any continuous time system's one. That is because the position of poles in the continuous time domain corresponding to the origin of the complex plane in the discrete time domain is minus infinity, which is not physically realizable, of course. In the discrete time domain instead, the allocation in the origin is perfectly realizable and determines the typical dead-beat closed loop behavior, i.e., the step response of the close loop system becomes equal

to a linear combination of different order pure delays. In order to verify this property with our example, we now compute the closed loop transfer function between input I_{OREF} and output $\overline{I}_O$. Applying standard state feedback theorems and after simple calculations, we find

$$\frac{\overline{I}_O}{I_{OREF}}(z) = C_A \cdot (zI - \Phi_A)^{-1} \cdot \Gamma_{A2} = \frac{1}{z^2}, \qquad (A5.10)$$

which corresponds, as expected, to a dynamic response equivalent to a pure two modulation period delay. Similarly, we can compute the closed loop transfer function from the disturbance to the output. We find

$$\frac{\overline{I}_O}{E_S}(z) = C_A \cdot (zI - \Phi_A)^{-1} \cdot \Gamma_{A1} = \frac{1}{z^2} \cdot \frac{T_S}{L_S} \cdot (-z - 1 + K_3). \qquad (A5.11)$$

As can be seen, there is no value of K_3 that can guarantee a zero transfer function from disturbance to output. This is due to the fact that the compensation term of the controller equation (A5.7) is one step delayed with respect to the control output and, as such, is only approximated. Under these conditions, the best we can do is to minimize the transfer function (A5.11). It is easy to verify that the choice $K_3 = 2$ achieves this minimization. Rewriting (A5.11) in the time domain and imposing $K_3 = 2$ we find

$$\overline{I}_O(k) = \frac{T_S}{L_S} \cdot [-E_S(k-1) + E_S(k-2)], \qquad (A5.12)$$

which, under the assumption of slowly varying E_S, guarantees the minimum disturbance effect of the output. Having determined the controller parameters K_1, K_2, and K_3, we are now ready to explicitly write the control equation, which turns out to be

$$\overline{V}_{OC}(k+1) = -\overline{V}_{OC}(k) + \frac{L_S}{T_S} \cdot [I_{OREF}(k) - \overline{I}_O(k)] + 2 \cdot E_S(k). \qquad (A5.13)$$

As can be seen, (A5.13) is equal to (3.21).

To complete this theoretical discussion of the predictive controller, we need to add a final remark about the phase voltage estimation. The derivation presented here refers to the basic predictive controller implementation, where the phase voltage E_S is assumed to be measured. If we want to consider the use of an output voltage estimator, additional care must be taken. The estimation equation can be directly obtained from the state variable $\overline{I}_O$ update equation (A5.5). However, using the estimated voltage $\hat{E}_S$ instead of the measured one in the control equation determines an increase of the order of the system, because $\hat{E}_S$ is a function of input and state variable values. As a general rule, this makes the dead-beat properties and stability of the controlled system more sensitive to model and parameter mismatches, reducing its robustness.

An example of the former type is represented by not properly modeled circuit components (like R_S) that, assumed to be negligible, turn out to be comparable with the other circuit components. The effect of model mismatches is normally very serious: since the control equation is based on a given system model, any deviation of the physical system from the model makes the controller interaction with the physical system unpredictable in its effects. Minor mismatches determine deviations from the expected dynamic performance, major ones can even determine unstable or lightly damped closed loop responses.

Parameter uncertainties, instead, are typically determined by construction tolerances or parameter value drifts, such as those due to varying physical or environmental operating conditions (current, temperature). Their effect on the dynamic performance of the closed loop system can be serious, ranging from different extents of performance degradation to system instability.

The formal analysis of model mismatches goes beyond the scope of our discussion, requiring a solid background in system identification theory. Instead, we can briefly discuss the effect of parameter uncertainty and provide an estimation of the predictive controller robustness to parameter variations. Considering (3.22) we see that several parameters contribute to the definition of the algorithm coefficients, each of them being a potential source of mismatch. To give an example of the analysis procedure we can apply to estimate the sensitivity of the controller to the mismatch, we begin by referring, for simplicity, to (3.21), where the only parameter we need to take into account is inductor L_S. Of course, transducer gain or dc link voltage variations can be treated similarly. We can easily model errors or variations on parameter L_S distinguishing the value used in the VSI model from that used in the predictive controller equation. In order to do that we can rewrite (3.21) as follows,

$$\overline{V}_{OC}(k+1) = -\overline{V}_{OC}(k) + \frac{L_S \pm \Delta L_S}{T_S} \cdot [\overline{I}_O(k+2) - \overline{I}_O(k)] + 2 \cdot E_S(k), \qquad (3.25)$$

where parameter L_S has been replaced by $L_S \pm \Delta L_S$, thus putting into evidence the possible presence of an error, ΔL_S, implicitly defined as a positive quantity. The analysis of the impact of ΔL_S on the system's stability requires the computation of the system's eigenvalues. Referring to the procedure outlined in Aside 5, we can immediately find the state matrix corresponding to (3.19) and (3.25). This turns out to be

$$\Phi'_A = \begin{bmatrix} 1 & \frac{T_S}{L_S} \\ -\frac{L_S \pm \Delta L_S}{T_S} & -1 \end{bmatrix}. \qquad (3.26)$$

It is now immediate to find the eigenvalues of matrix Φ'_A. These are given by the following expression:

$$\lambda_{1,2} = \sqrt{\pm \frac{\Delta L_S}{L_S}} \quad \Rightarrow \quad |\lambda_{1,2}| = \sqrt{\frac{\Delta L_S}{L_S}}. \qquad (3.27)$$

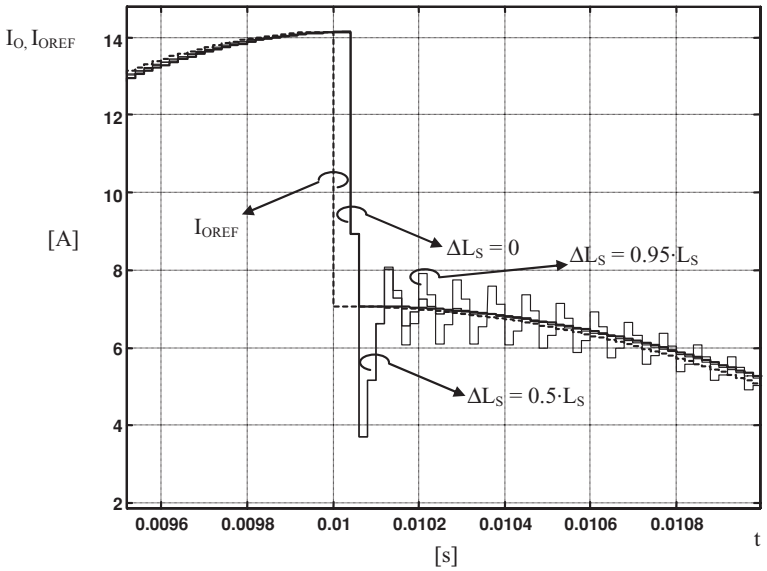

FIGURE 3.19: Simulation of the VSI with the predictive controller and different level of mismatch on parameter L_S. The figure shows the response to a step reference change of the sampled VSI output current I_O.

From (3.27) we see that the magnitude of the closed loop system's eigenvalues is limited to the square root of the relative error on L_S. This means that unless a higher than 100% error is made on the estimation of L_S or, equivalently, unless a 100% variation of L_S takes place, due to changes in the operating conditions, the predictive controller will keep the system stable. Please note that, interestingly, this result is independent of the sampling frequency. Of course, even if instability requires bigger than unity eigenvalues, the good reference tracking properties of the predictive controller are likely to get lost, even for smaller than unity values of the relative error. We can visualize the effect of ΔL_S considering Fig. 3.19. Please note that, differently from previous figures, Fig. 3.19 shows the sampled current and its reference to better put into evidence the effect of the parameter mismatch. We can immediately see how the presence of a mismatch determines an oscillatory response. Please note that the undershoot is not affected by the amount of mismatch, while the damping factor is. With a 95% mismatch the response is lightly damped and barely acceptable for practical applications.

As it might be expected, the robustness of the predictive controller to mismatches gets worse if the estimation of the phase voltage is used instead of its measurement. The analytical investigation of this case is a little more involved than the previous one, but still manageable with pencil and paper calculations. The procedure consists in writing the system (3.19), controller (3.21), and estimator (3.24) equations, either solving them using Z-transform to find

the reference to output transfer function, or, equivalently, arranging them to get the state matrix, and, finally, examining the characteristic polynomial of the system. Following the former procedure, we get

$$\lambda(z) = z^3 \pm 3 \cdot \frac{\Delta L_S}{L_S} \cdot z \pm 2 \cdot \frac{\Delta L_S}{L_S}. \tag{3.28}$$

It is now possible to plot the zeros of the characteristic polynomial, i.e., the closed loop system eigenvalues, as functions of ΔL_S. Considering only negative signs in (3.28), we find the results presented in Fig. 3.20. The figure shows that only a 20% error is allowed before system instability occurs. It is worth noting that this result is independent of the switching frequency since (3.28) is only a function of the mismatch error ΔL_S. Moreover, it is interesting to note that the unstable pole is at half of the sampling frequency, since it lies on the real axis. Instead, considering positive signs in (3.28) a similar situation can be found, where the minimum variation required to induce instability is somewhat higher than the previous one. Fig. 3.20, therefore, represents the worst-case condition.

The results we have just found seem to completely undermine the practical applicability of the dead-beat current controller, especially when the output voltage estimation is considered. Fortunately, this is hardly the case. The reason is that it is possible, with some modifications of the controller structure, as those suggested in [10, 11], to strongly improve the controller robustness, making it perfectly apt to practical applications.

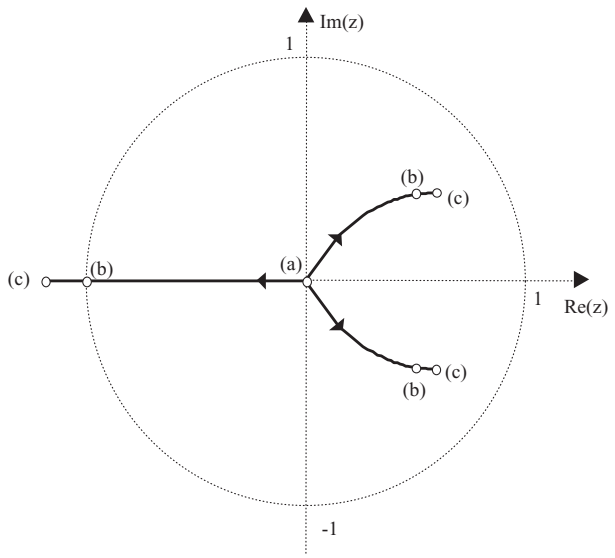

FIGURE 3.20: Plot of the closed loop system eigenvalues as functions of the parameter L_S mismatch. (a) $\Delta L_S = 0$. (b) $\Delta L_S = 0.2 \cdot L_S$. (c) $\Delta L_S = 0.3 \cdot L_S$.

3.2.7.3 Effects of Converter Dead-Times

Converter dead-times are another nonideal characteristic of the VSI that is not taken into account by the model which the predictive controller is based on. In a certain sense, their presence can be considered a particular case of model mismatch. We know from Section 2.1.4 that the presence of dead-times implies a systematic error on the average voltage generated by the inverter. The error has an amplitude that depends directly on the ratio of dead-time duration and modulation period and a sign that depends on the load current sign. As we did in Section 2.3.1, we can model the dead-time effect as a square-wave disturbance having a relatively small amplitude (roughly a few percent of the dc link voltage) and opposite phase with respect to the load current. We can consider this disturbance as an undesired component that is summed, at the system input, to the average voltage requested by the current control algorithm.

As such, the disturbance should be, at least partially, rejected by the current controller. The effectiveness of the input disturbance rejection capability depends on the low-frequency gain the controller is able to determine for the closed loop system. And this is where the dead-beat controller shows another weak point. We have seen how the dead-beat action tends to get from the closed loop plant a dynamic response that is close to a pure delay. Unfortunately, this implies a very poor rejection capability for any input disturbance. To clarify this point we can again refer to Aside 5, Fig. A5.1, and compute the closed loop transfer function from the exogenous disturbance E_S to the output $\overline{I}_O$. Indeed, this is the transfer function experienced by the dead-time induced voltage disturbance as well. Simple calculations yield

$$\frac{\overline{I}_O}{E_S}(z) = \frac{1}{z^2} \cdot \frac{T_S}{L_S} \cdot (z+1), \tag{3.29}$$

which means that the output is proportional to the sum of two differently delayed input samples. In terms of disturbance rejection the result is rather disappointing. Plotting the frequency response of (3.29) we find that it is practically flat from zero up to the Nyquist frequency; i.e., there is no rejection of the average inverter voltage disturbance.

Consequently, we cannot expect the dead-beat controller to compensate for the dead-time effect. This means that unless some external, additional compensation strategy is adopted, a certain amount of current distortion is likely to be encountered. A typical example is shown in Fig. 3.21, where the sampled output current and its reference are shown. We can clearly see the double effect of uncompensated dead-times: (i) a systematic amplitude error and (ii) a crossover distortion phenomenon. The reason for the former is an obvious consequence of the negligible rejection capability of the dead-beat controller. The latter instead is due to the sign inversion of the voltage disturbance, taking place at the moment of current zero crossing, which the controller tries to compensate for. We can also note how, because of the current ripple amplitude, the sign of the voltage disturbance is not stable around the zero crossing instant,

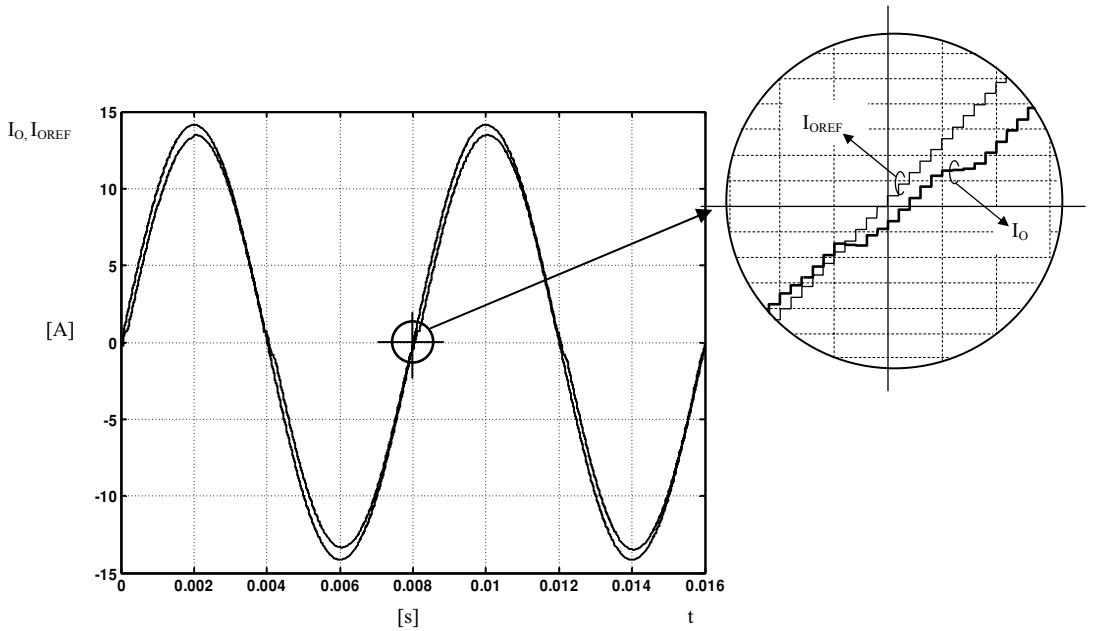

FIGURE 3.21: Simulation of the VSI with the predictive controller and dead-times. The dead-time effects are as follows: (i) crossover distortion, visible in the inset; (ii) steady-state error between the amplitude of the converter current and its reference. The sampled output current is shown instead of the instantaneous one to eliminate the ripple, which can mask the error. Dead-times are set to 5% of the modulation period, just to magnify the effect.

which makes the transient duration longer. It is worth noting that Fig. 3.21 is obtained assuming that the duration of the dead-time is about 5% of the modulation period. This is an exaggerated value, which is used in the simulation on purpose, just to magnify the current distortion. In practical situations, dead-times are in the range between 1% and 2% of the modulation period and the overall effect on the converter output current is accordingly smaller.

There are different possible methods to compensate the dead-time induced output current distortion, which are sometimes also used in conjunction with regulators, like the PI controller, which present a significant low-frequency rejection capability. That is because such regulators are nonetheless exposed to crossover distortion that can be unacceptable for some applications, like, for example, high-quality electrical drives. In the case of the dead-beat controller some form of compensation is instead mandatory. Compensation methods can be divided into (i) closed loop or on-line and (ii) open loop or off-line. The best performance is offered by closed loop dead-time compensation, which requires, however, the measurement of the actual inverter average output voltage. Its comparison with the voltage set-point provided to the modulator

gives sign and amplitude of the dead-time induced average voltage error, which can therefore be compensated with minimum delay simply by summing to the set-point for the following modulation period the opposite of the measured error. The need for measuring the typically high output inverter voltage requires particular care. The estimation of the inverter average output voltage is normally done by measuring the duration of the voltage-high and voltage-low parts of the modulation period, i.e., by computing the actual, effective output duty-cycle.

However, much more often, off-line compensation strategies are used. These offer a lower quality compensation, but can be completely embedded in the modulation routine programmed in the microcontroller (or DSP), requiring no measure. The off-line compensation of dead-times is based on a worst-case estimation of the dead-time duration and on the knowledge of the sign of the output current, which is normally inferred from the reference signal (not from the measured output current, to avoid any complication due to the high-frequency ripple). Given both of these data, it is possible to add to the output voltage set-point a compensation term that balances the dead-time induced error. The method normally requires some tuning, in order to avoid under- or overcompensation effects. The results are normally quite satisfactory, unless a very high precision is required by the application, allowing us to eliminate the amplitude error and to strongly attenuate the crossover distortion phenomenon.

3.2.7.4 Comparison with PI Controller

A final remark is needed to summarize the main features of the dead-beat predictive current controller and to compare its performance with that of the PI controller. The predictive controller is capable of a very fast dynamic response, the best among digital current controllers and clearly superior to that achievable by any digital PI controller. Therefore, it is very well suited to those applications of VSIs where the capability of tracking rapidly variable current reference signals is required. Examples of these applications can be the active power filters and the high-performance adjustable speed drives. On the other hand, the predictive controller, at least in its basic implementation, requires the measurement of the load voltage, which generally complicates the hardware needed for its implementation well beyond what is required by a PI controller. We have also seen how estimation techniques can be employed to avoid the voltage measurement, but we need to point out that (i) the estimation makes the controller more sensitive to model and parameter mismatches, and (ii) the dynamic performance is worsened, although it usually remains superior to that of a conventional PI regulator. Finally, we have seen how the compensation of dead-times is practically mandatory for the dead-beat controller, which has no inherent integral action, while it may not be required by the PI, unless very low distortion current waveforms are required by the application. Moreover, the sensitivity of the predictive controller to measurement noise is surely higher than that of the PI controller, which calls for particular care in the design of the signal conditioning circuitry.

REFERENCES

[1] A. V. Oppenheim, R. W. Schafer and J. R. Buck, *Discrete Time Signal Processing*, 2nd edition. Englewood Cliffs, NJ: Prentice-Hall, 1999.

[2] K. Ogata, *Discrete Time Control Systems*, Englewood Cliffs, NJ: Prentice-Hall, 1987.

[3] K. J. Astrom and B. Wittenmark, *Computer-Controlled Systems: Theory and Design*, 3rd edition. Englewood Cliffs, NJ: Prentice-Hall, 1997.

[4] A. V. Peterchev and S. R. Sanders, "Quantization resolution and limit cycling in digitally controlled PWM converters," *IEEE Trans. Power Electron.*, Vol. 18, No. 1, pp. 301–308, Jan. 2003.doi.org/10.1109/TPEL.2002.807092

[5] H. Peng, D. Maksimovic, A. Prodic and E. Alarcon, "Modeling of quantization effects in digitally controlled dc–dc converters," *IEEE PESC'04*, Aachen, Germany, 2004.

[6] D. M. Van de Sype, K. DeGusseme, A. R. Van den Bossche and J. A. Melkebeek, "Small Signal Z-domain Analysis of Digitally Controlled Converters," *IEEE Trans. on Power Electron.*, Vol. 21, No. 1, pp. 470–478, 2006.

[7] G. C. Verghese, M. E. Elbuluk and J. G. Kassakian, "A general approach to sampled-data modeling for power electronic circuits," *IEEE Trans. Power Electron.*, Vol. 1, pp. 76–89, April 1986.

[8] J. G. Kassakian, M. F. Schlecht and G. C. Verghese, *Principle of Power Electronics.* Reading, MA: Addison-Wesley, 1991.

[9] B. K. Kuo, *Digital Control Systems*. Champaign, IL: SRL Publishing Company, 1977.

[10] L. Malesani, P. Mattavelli and S. Buso, "Robust dead-beat current control for PWM rectifiers and active filters," *IEEE Trans. Indust. Appl.*, Vol. 35, No. 3, pp. 613–620, May/June 1999.doi.org/10.1109/28.767012

[11] G. H. Bode, P. C. Loh, M. J. Newman and D. G. Holmes, "An improved robust predictive current regulation algorithm," *IEEE Trans. Indust. Appl.*, Vol. 41, No. 6, pp. 1720–1733, Nov./Dec. 2005.doi.org/10.1109/TIA.2005.858324

CHAPTER 4

Extension to Three-Phase Inverters

In this chapter we present the possible means for the application to three-phase inverters of what we have just seen about digital current control of single-phase VSIs. When the three-phase converter is characterized by four wires, i.e., three phases plus neutral, the application is straightforward, since a four-wire three-phase system is totally equivalent to three independent single-phase systems. Of course, this particular situation does not deserve any further discussion. In contrast, we need to apply a little more caution when we are dealing with a three-phase system with an insulated neutral, i.e., with a three-wire, three-phase system. The objective of this chapter to give the basic knowledge needed to extend the control principles we have previously described for these kinds of systems. Two fundamental tools are required to design an efficient three-phase current controller: (i) $\alpha\beta$ transformation and (ii) space vector modulation (SVM).

In the first part of this chapter, we are going to illustrate the principles of both. Next, we will show how, under certain assumptions, the three-phase system dynamic model can be transformed into an equivalent two-phase system, with independent components. We will see how, in this particular case, the controller design for the two-phase system is identical to that of a single-phase one.

In the final part of this chapter we will discuss a particular kind of two-phase controller, known as rotating reference frame controller, presenting the merits and limitations of this solution.

4.1 THE $\alpha\beta$ TRANSFORMATION

The $\alpha\beta$ transformation represents a very useful tool for the analysis and modeling of three-phase electrical systems. In general, a three-phase linear electric system can be properly described in mathematical terms only by writing a set of tridimensional dynamic equations (integral and/or differential), providing a self-consistent mathematical model for each phase. In some cases though, the existence of physical constraints makes the three models not independent from each other. In these circumstances the order of the mathematical model can be reduced without any loss of information. We will see a remarkable example of this in the following.

Supposing that it is physically meaningful to reduce the order of the mathematical model from three to two dimensions, the $\alpha\beta$ transformation represents the most commonly used relation to perform the reduction of order. To explain the way it works we can consider a tridimensional vector $\vec{x}_{abc} = [x_a \quad x_b \quad x_c]^T$ that can represent any triplet of the system's electrical variables (voltages or currents). We can now consider the linear transformation, $T_{\alpha\beta\gamma}$,

$$\begin{bmatrix} x_\alpha \\ x_\beta \\ x_\gamma \end{bmatrix} = T_{\alpha\beta\gamma} \begin{bmatrix} x_a \\ x_b \\ x_c \end{bmatrix} = \sqrt{\frac{2}{3}} \begin{bmatrix} 1 & -1/2 & -1/2 \\ 0 & \sqrt{3}/2 & -\sqrt{3}/2 \\ 1/\sqrt{2} & 1/\sqrt{2} & 1/\sqrt{2} \end{bmatrix} \begin{bmatrix} x_a \\ x_b \\ x_c \end{bmatrix}, \tag{4.1}$$

which, in geometrical terms, represents a change from the set of reference axes denoted as *abc* to the equivalent one indicated as $\alpha\beta\gamma$. This change of reference axes takes place because the standard R^3 orthonormal base B_{abc},

$$B_{abc} = \{[1 \quad 0 \quad 0]^T, \quad [0 \quad 1 \quad 0]^T, \quad [0 \quad 0 \quad 1]^T\}, \tag{4.2}$$

is replaced by the new base $B_{\alpha\beta\gamma}$,

$$B_{\alpha\beta\gamma} = \sqrt{2/3}\{[1 \quad -1/2 \quad -1/2]^T, \quad [0 \quad \sqrt{3}/2 \quad -\sqrt{3}/2]^T, \quad [1/\sqrt{2} \quad 1/\sqrt{2} \quad 1/\sqrt{2}]^T\} \tag{4.3}$$

The $B_{\alpha\beta\gamma}$ base is once again orthonormal, i.e., its vectors have unity norm and are orthogonal to one another thanks to the presence of the $\sqrt{2/3}$ coefficient, which also appears in (4.1). This coefficient is sometimes omitted when the maintenance of the base orthonormality is not considered essential. However, orthonormality implies that (i) the inverse of the $T_{\alpha\beta\gamma}$ transformation is equal to the matrix transposed, i.e., $T_{\alpha\beta\gamma}^{-1} = T_{\alpha\beta\gamma}^T$ and (ii) the computation of electrical powers is independent from the transformation of coordinates, i.e., $\left\langle \vec{e}_{abc}, \vec{i}_{abc} \right\rangle = \left\langle \vec{e}_{\alpha\beta\gamma}, \vec{i}_{\alpha\beta\gamma} \right\rangle$, where the "$\langle\rangle$" operator represents the scalar product of vectors, $\vec{e}$ is a voltage vector and $\vec{i}$ is a current vector. The latter property justifies the fact that (4.1) is sometimes called the "power invariant" transformation. The geometrical interpretation of (4.1) is shown in Fig. 4.1(a).

The $T_{\alpha\beta\gamma}$ transformation has an additional interesting property, which becomes clear when we take into account the condition

$$x_a + x_b + x_c = 0 \quad \Rightarrow \quad x_\gamma = 0, \tag{4.4}$$

whose meaning is to operate the restriction of the tridimensional space to a plane π Fig. 4.1(a) Examining (4.3) and (4.4) we can see how the first two components of the base $B_{\alpha\beta\gamma}$ lie on π, while the third is orthogonal to π. This means that the first two components of $B_{\alpha\beta\gamma}$ actually represent an orthonormal base for plane π, while the third component has no projection on π. This observation is fundamental for our conclusion: every time the constraint (4.4) is

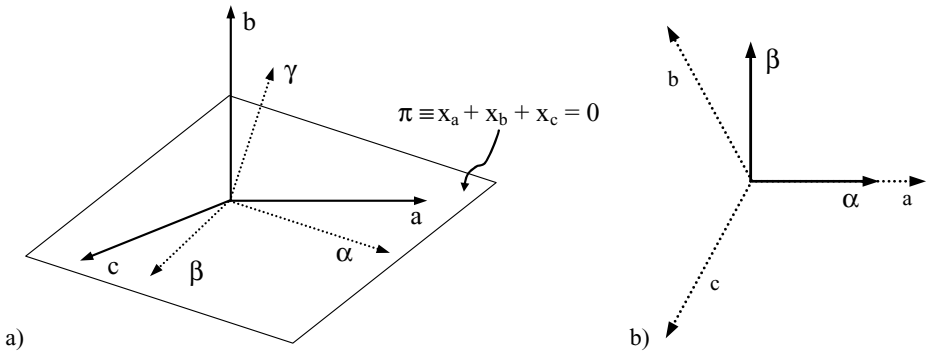

FIGURE 4.1: (a) Graphical representation of the $T_{\alpha\beta\gamma}$ coordinate transformation.

meaningful for a tridimensional system, the coordinate transformation $T_{\alpha\beta\gamma}$ allows us to describe the same system in a bidimensional space without any loss of information. This holds because any vector complying with (4.4) is actually lying on the plane π and, as such, can be expressed as a linear combination of base vectors defined for π. We can therefore define the so-called $\alpha\beta$ transformation as follows:

$$\begin{bmatrix} x_\alpha \\ x_\beta \end{bmatrix} = T_{\alpha\beta} \begin{bmatrix} x_a \\ x_b \\ x_c \end{bmatrix} = \sqrt{\frac{2}{3}} \begin{bmatrix} 1 & -1/2 & -1/2 \\ 0 & \sqrt{3}/2 & -\sqrt{3}/2 \end{bmatrix} \begin{bmatrix} x_a \\ x_b \\ x_c \end{bmatrix}, \tag{4.5}$$

and its inverse as

$$\begin{bmatrix} x_a \\ x_b \\ x_c \end{bmatrix} = T^T_{\alpha\beta\gamma} \begin{bmatrix} x_\alpha \\ x_\beta \\ 0 \end{bmatrix} = \sqrt{\frac{2}{3}} \begin{bmatrix} 1 & 0 \\ -1/2 & \sqrt{3}/2 \\ -1/2 & -\sqrt{3}/2 \end{bmatrix} \begin{bmatrix} x_\alpha \\ x_\beta \end{bmatrix} = T^T_{\alpha\beta} \begin{bmatrix} x_\alpha \\ x_\beta \end{bmatrix}. \tag{4.6}$$

Equation (4.6) is easily obtained exploiting the base orthonormality and considering the transposed matrix of (4.5). In geometrical terms (4.5) simply determines the projection of any vector $\vec{x}_{abc} = [x_a \quad x_b \quad x_c]^T$ on the plane π. We need to underline once more that this is physically meaningful only if the γ component of the vector $\vec{x}_{abc} = [x_a \quad x_b \quad x_c]^T$ is zero. The γ component, as can be easily verified, is nothing but the *arithmetic average* of the three vector component values, also known as the *common mode* vector component. When this is not zero, the application of the $\alpha\beta$ transformation implies the loss of the information associated with the common mode. It is also interesting to note that the projection on the plane π of the base B_{abc} determines three 120° angled axes, as shown in Fig. 4.1(b), which makes the matrix in (4.5) easy to remember.

It is very useful to visualize the effect of the application of $T_{\alpha\beta}$ to some particular cases. We begin presenting the case of sinusoidal voltage signals. If we consider a triplet of symmetric

sinusoidal signals like

$$\begin{aligned} e_a &= U_M \sin(\omega t), \\ e_b &= U_M \sin(\omega t - 2\pi/3), \\ e_c &= U_M \sin(\omega t + 2\pi/3), \end{aligned} \tag{4.7}$$

it is easy to verify that

$$\begin{aligned} e_\alpha &= \sqrt{\frac{3}{2}} U_M \sin(\omega t), \\ e_\beta &= -\sqrt{\frac{3}{2}} U_M \cos(\omega t). \end{aligned} \tag{4.8}$$

It is possible to see that the space vector $\vec{e}_{abc}$, associated with (4.7), satisfies the constraint (4.4) and that, as such, it can be described without loss of information in the $\alpha\beta$ reference frame. In that frame, the vector $\vec{e}_{\alpha\beta}$, can be interpreted as a $\sqrt{\frac{3}{2}} U_M$ amplitude rotating vector, the angular rotation speed being equal to ω.

4.2 SPACE VECTOR MODULATION

Space vector modulation (SVM) is a frequently used method to implement PWM in three-phase switching converters, with an insulated neutral. It allows us not only to simplify the control organization, but also to maximize the exploitation of the converter hardware, inherently realizing a third harmonic injection mechanism. The basic principles behind SVM can be explained referring to the idealized three-phase voltage-source inverter of Fig. 4.2 As can be seen, the structure is a straightforward extension of the single-phase one we have been considering so far. Its characteristics and modes of operation are analyzed in detail in every power electronics textbook (such as [1] and [2]), so we won't spend many words on it. However,

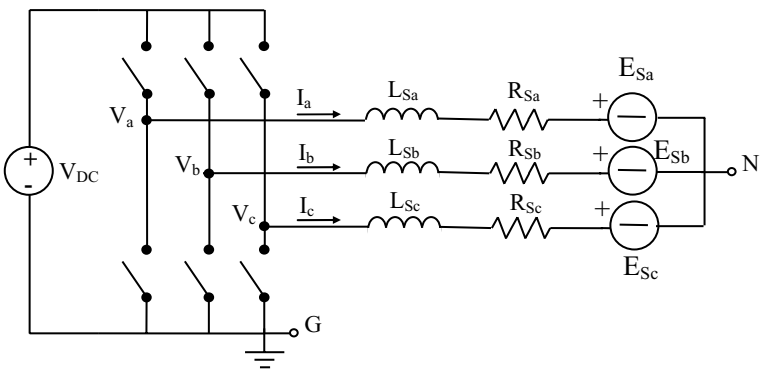

FIGURE 4.2: Three-phase VSI simplified schematic.

three fundamental characteristics, essential to the understanding of what follows, have to be underlined: (i) the converter has an insulated neutral, i.e., the circuit node indicated by N in Fig. 4.2 is floating, (ii) there is a single input dc voltage source, which makes the phase voltages V_a, V_b, V_c, referred to node G, unipolar, and (iii) the load is generally symmetrical and balanced, i.e., all impedances have the same values and E_{Sa}, E_{Sb}, and E_{Sc} are symmetrical and balanced sinusoidal voltages.

The application of SVM requires the instantaneous inverter output voltage, represented by the vector $\vec{V}_{abc} = [V_a \quad V_b \quad V_c]^T$ to be *projected* on the $\alpha\beta$ reference frame, as defined in the previous section. From Fig. 4.2 it is immediately recognizable that, at any instant each inverter phase voltage can be either zero or equal to the dc link voltage V_{DC}. Therefore, the inverter output voltage vector can assume, at any instant, only one out of eight different values. The possible output voltage vector values and their *projections* on the plane π are shown in Fig. 4.3. As can be seen, there are two different possibilities to impose a zero phase-to-phase voltage on the load. This property can be exploited in the implementation of SVM, for example, to minimize the number of switch commutations.

The idea behind SVM is quite simple [3, 4]. A desired output voltage vector, represented in the $\alpha\beta$ reference frame, is obtained from the superposition of the inverter output vectors, so that, on average, at the end of any modulation period a voltage equal to the desired one will have been generated. The procedure can be explained referring to Fig. 4.4. The desired vector, $\vec{V}^*_{\alpha\beta}$, is projected on the two closest inverter output state vectors, i.e., $\vec{V}_{100}$ and $\vec{V}_{110}$, in the example of Fig 4.4. Of course, the position of $\vec{V}^*_{\alpha\beta}$ considered in this example is arbitrary; however, exactly the same reasoning can be applied to different vector locations. The length of each projection, V_1 and V_2, determines the fraction δ of the modulation period that will be occupied by each output vector, according to the following relation:

$$\delta_1 = \frac{|V_1|}{\left|\vec{V}_{100}\right|} \quad \delta_2 = \frac{|V_2|}{\left|\vec{V}_{110}\right|}. \tag{4.9}$$

The application of the zero voltage vector for a fraction δ_3 of the modulation period is normally required to satisfy the following condition:

$$\delta_1 + \delta_2 + \delta_3 = 1, \tag{4.10}$$

which simply expresses the fact that the modulation period must be fully occupied by output voltage vectors. Following this procedure, the average inverter output voltage $\overline{V}_{\mathrm{o}}$ will be given by

$$\overline{V}_{\mathrm{o}} = \delta_1 V_{100} + \delta_2 V_{110} + \delta_3 V_{111} = V_1 + V_2 = \vec{V}^*_{\alpha\beta}, \tag{4.11}$$

as expected. Please note that (i) the zero vector can be either $\vec{V}_{111}$ or, equivalently, $\vec{V}_{000}$, (ii) the order of application of the inverter output vectors is arbitrary and can be used as a degree of

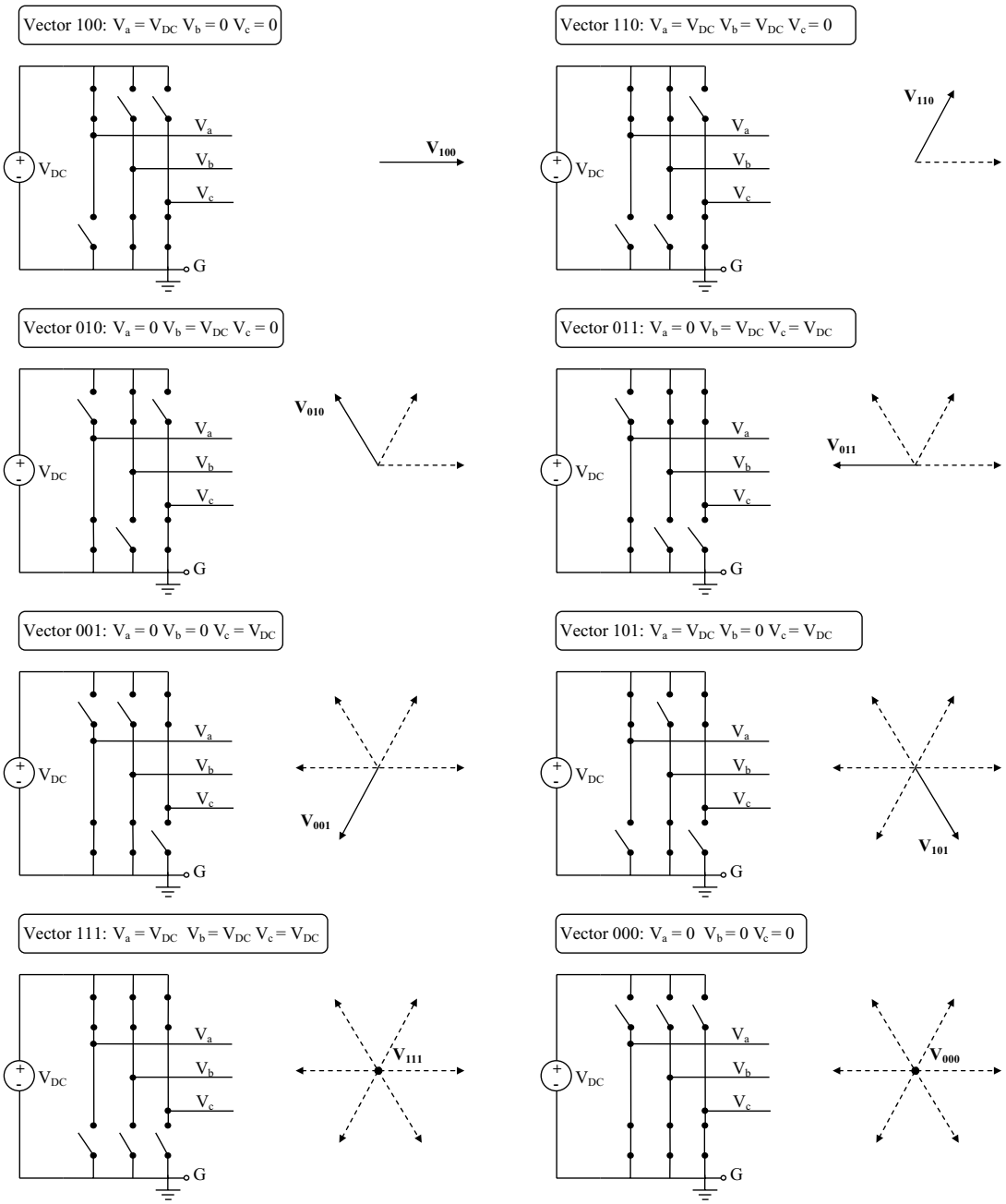

FIGURE 4.3: Three-phase inverter output voltage vectors and their projection on the plane π.

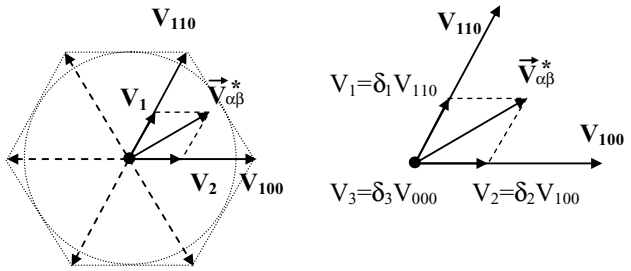

FIGURE 4.4: Generation of the voltage reference vector by superposition of inverter output vectors.

freedom in the implementation of SVM (see Aside 6), and (iii) the commutation from V_1 to V_2 always requires the commutation of a single inverter phase, no matter what sector of the hexagon the reference vector is lying on.

The implementation of the above-described procedure requires some amount of computation. In any modulation period, given the α and β components of the voltage reference vector $\vec{V}^*_{\alpha\beta}$ one has to (i) locate the two closest inverter output vectors, i.e., the hexagon sector where $\vec{V}^*_{\alpha\beta}$ is lying on, (ii) determine the amplitude of V_1 and V_2, and (iii) calculate the values of δ_1, δ_2, δ_3, using (4.9) and (4.10). Of course, the simplest way to perform these computations is by using a microcontroller or DSP. This is the reason why SVM is almost always associated with digital control. In Aside 6, we will further discuss some implementation issues of SVM.

We have seen in the previous section that the projection on the $\alpha\beta$ reference frame of a triplet of symmetrical, sinusoidal, phase voltages is a constant amplitude rotating vector. Therefore, every time our three-phase VSI has to generate a triplet of sinusoidal phase voltages, which happens very frequently, the SVM procedure will have to synthesize the rotating reference vector corresponding to it. This will determine a period-by-period adjustment of the output vectors and of the δ_1, δ_2, δ_3 values. It can be interesting to identify the locus of the constant amplitude rotating reference vectors that can be generated by the inverter without distortion. This is represented by the circle *inscribed* in the vector hexagon (Fig. 4.4). It is easy to verify that every vector that lies inside the circle generates a valid δ_1, δ_2, δ_3 triplet. Instead, a vector that lies partially outside the circle cannot be generated by the inverter, because the sum of the corresponding δ_1, δ_2, δ_3 becomes greater than unity. This situation is called inverter *saturation* and generally causes output voltage distortion.

If we consider (4.5) and Fig. 4.4, it is easy to calculate the amplitude U_{MMAX} of the voltage triplet (4.7), which corresponds to a rotating vector having an amplitude equal to the radius of the inscribed circle. We find that

$$\sqrt{\frac{3}{2}}U_{\mathrm{MMAX}} = \sqrt{\frac{2}{3}}V_{\mathrm{DC}}\frac{\sqrt{3}}{2} \quad \Leftrightarrow \quad U_{\mathrm{MMAX}} = \frac{2}{\sqrt{3}}\frac{V_{\mathrm{DC}}}{2} \cong 1.15\frac{V_{\mathrm{DC}}}{2}, \qquad (4.12)$$

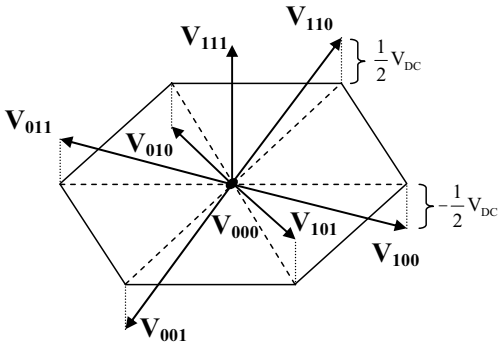

FIGURE 4.5: Tridimensional view of the space vector hexagon.

which shows a very interesting fact: the application of SVM increases the range of the possible sinusoidal output voltages by 15% with respect to what could be expected looking at the schematic of Fig. 4.2. The reason why this happens is that the inverter output voltage vectors of Fig. 4.3 *do not* comply with the constraint (4.4), as it is easy to verify. Consequently, what is used to synthesize the desired output voltage vector $\vec{V}^*_{\alpha\beta}$ is not the superposition of vectors lying on the plane π as Fig. 4.4 might suggest. A more realistic representation of the inverter output vectors, which puts into evidence their γ component, is shown in Fig. 4.5.

Aside 6. Implementation of Space Vector Modulation

We now consider a possible implementation algorithm for space vector modulation, which can be directly programmed into a microcontroller or a digital signal processor. The first issue in SVM implementation is the identification of the hexagon sector where the reference vector is lying. This can be done by implementing once again a base change from the $\alpha\beta$ reference frame to a new set of three different reference frames. Fig. A6.1 shows the considered set.

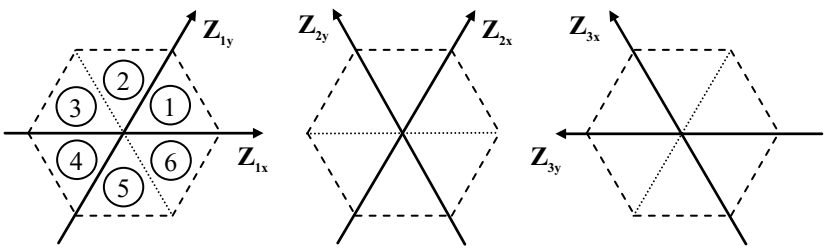

FIGURE A6.1: Set of three bidimensional reference frames.

As can be seen, each frame refers to a particular couple of hexagon sectors. The method we propose simply requires the projection of the inverter output voltage reference vector $\vec{V}^*_{\alpha\beta}$ onto each one of the three hexagon reference frames. This is easily implemented with the following set of reference base change matrixes:

$$M_1 = \begin{bmatrix} 1 & -\frac{1}{\sqrt{3}} \\ 0 & \frac{2}{\sqrt{3}} \end{bmatrix} \quad M_2 = \begin{bmatrix} 1 & \frac{1}{\sqrt{3}} \\ -1 & \frac{1}{\sqrt{3}} \end{bmatrix} \quad M_3 = \begin{bmatrix} 0 & \frac{2}{\sqrt{3}} \\ -1 & -\frac{1}{\sqrt{3}} \end{bmatrix}, \qquad (A6.1)$$

which map the orthogonal set of axes α and β onto the three, nonorthogonal sets Z. It is interesting to note that the algorithm required to implement the three projections is quite simple. Here we propose a possible operation sequence that gives the six Z components:

$$\begin{aligned} tmp &= \frac{V^*_\beta}{\sqrt{3}}; && \text{save to a temporary register} \\ Z_{1x} &= V^*_\alpha - tmp; && Z_{1x} \text{ found} \\ Z_{2y} &= -Z_{1x}; && Z_{2y} \text{ found} \\ Z_{1y} &= 2tmp; && Z_{1y} \text{ found} \\ Z_{3x} &= Z_{1y}; && Z_{3x} \text{ found} \\ Z_{2x} &= V^*_\alpha + tmp; && Z_{2x} \text{ found} \\ Z_{3y} &= -Z_{2x}; && Z_{3y} \text{ found} \end{aligned}$$

As can be seen, the sequence implies the execution of only one multiplication. Once the $Z_{ix}\,Z_{iy}$ components are known, it is simple to determine the hexagon sector by checking their sign. The procedure outlined in the flow chart of Fig. A6.2 accomplishes this task. The sequence of sign checks can be efficiently implemented with logic operations in the modulator routine. In the end, with a few lines of code we have determined (i) the position of the reference vector in the hexagon and (ii) the lengths of its projections on the two adjacent output voltage vectors (represented by one of the three computed $Z_{ix}\,Z_{iy}$ couples). We are therefore ready to program the PWM modulator to generate such vectors, plus one of the two possible zero vectors. There are only two final issues that need to be taken care of: the sequence of vector generation and the possible occurrence of saturation.

As far as the former issue is concerned, we present two examples of possible generation sequences in Fig. A6.3. Depending on the controlled system characteristics, one can be more advantageous than the other. As far as the latter is concerned, there is not a single straightforward way to cope with saturation.

All strategies imply the acceptance of some degree of distortion of the output voltage. Once saturation is detected, which is easily done (the output vector durations, summed together, exceed the duration of the modulation period), some strategies reduce proportionally each

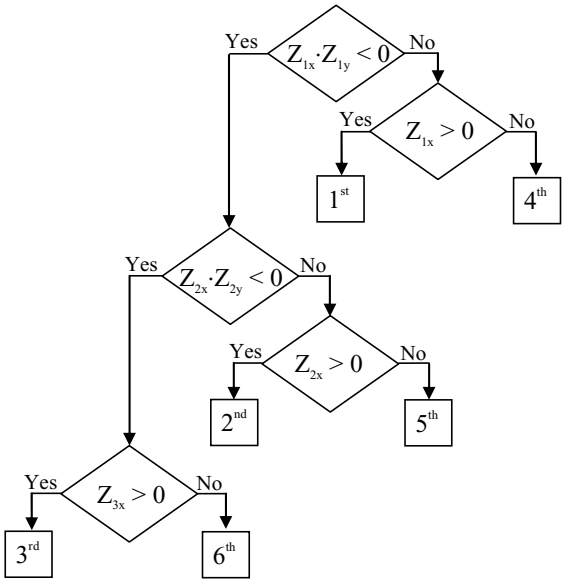

FIGURE A6.2: Flow chart of the sector identification algorithm.

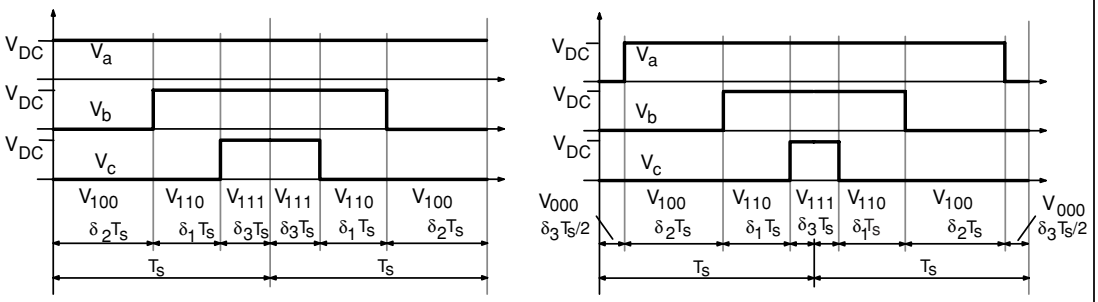

FIGURE A6.3: Two different application sequences for the same output voltage vectors. The sequence on the left implies a minimization of the number of switchings. The sequence on the right implies the minimization of the current ripple amplitude (voltage pulses have even symmetry). Note that each strategy develops in two adjacent modulation periods.

duration until a sum equal to the modulation period duration is obtained. Other strategies consider the reduction of only one component (the shorter of the two) so as to get their sum again to be equal to the duration of the modulation period. The latter strategy, of course, implies the loss not only of the correct vector amplitude, but also of its phase. Another issue of some interest concerning saturation is the automatic change from linear to six-step modulation [5], which can be necessary in heavy saturation conditions. It is easy to verify that this is inherently achieved by the second saturation strategy we have just described.

Therefore, any time one of the inverter output voltage vectors is generated, a nonzero γ component is produced on the load, which, being orthogonal to π is not visible in the vector decomposition of Fig. 4.4. Referring to Fig. 4.2, this means that SVM implies a particular modulation of the voltage between nodes N and G, V_{NG}. This is due to the common mode component of the inverter output voltage vectors. Indeed, it is easy to demonstrate that, in the case of a symmetrical load structure, almost always encountered in practice, V_{NG} is instantaneously and exactly equal to the γ component of the inverter output voltage. The most important implication of this fact is that the phase to neutral voltage of the load will always be insensitive to any common mode component of the inverter output voltage, i.e., one can freely add common mode components to the $\vec{V}_{abc}$ vector, without perturbing the load voltage.

This is exactly what SVM implicitly does. Its effect, from the inverter's standpoint, can be proved to be very similar to that of the third harmonic injection, sometimes employed in analog three-phase PWM implementations. An increase by 15% of the voltage–amplitude range that corresponds to a linear converter operation, i.e., to the absence of any saturation phenomenon, is obtained, as (4.12) clearly demonstrates.

This remark concludes our essential presentation of SVM. We are well aware that several other interesting issues could be addressed, but we feel like what we have presented is more than enough to allow us to discuss the following digital control application examples. The interested reader can find very useful additional information about SVM in the fundamental papers [3] and [4] and in several others that, in more recent times, have contributed to the development of PWM strategies for multiphase converters.

4.2.1 Space Vector Modulation Based Controllers

The typical organization of a three-phase VSI controller based on SVM is shown in Fig. 4.6. As can be seen, the controller takes advantage of the application of $\alpha\beta$ transformations to operate on two sampled variables instead of three. This not only simplifies the control algorithm, but also allows to directly generate the reference voltage components for the SVM in the $\alpha\beta$ reference frame. From those components, a suitable modulation procedure, like the one outlined in Aside 6, will be able to determine the phase duty cycles, managing inverter saturation if needed.

One could wonder whether the application of $\alpha\beta$ transformations to the controller input signals, in general, modifies the transfer function or state space model the controller design is based on. Clearly, if this is the case, passing from the three-phase system to the electrically equivalent two-phase one implies the need for a complete controller redesign. Luckily, this is hardly the case. Under the assumption of balanced and symmetrical load, we can indeed demonstrate that the design of the $\alpha-$ or β-axis controller is exactly identical to that of a single-phase current controller operating on one of the three inverter phases. In order to show this, we need to define the continuous time state space model of the inverter and its load. It is

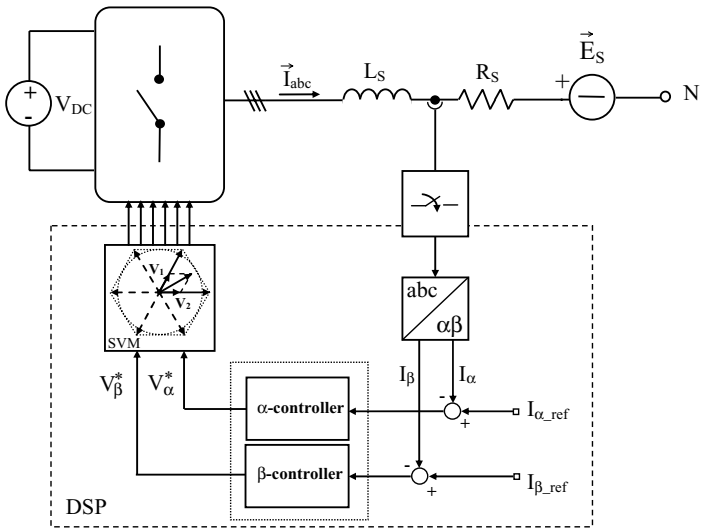

FIGURE 4.6: Organization of a three-phase digital current controller based on SVM.

easy to verify that this is given by

$$\frac{\mathrm{d}}{\mathrm{dt}}\begin{bmatrix} I_a \\ I_b \\ I_c \end{bmatrix} = -\frac{R_S}{L_S}\begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}\begin{bmatrix} I_a \\ I_b \\ I_c \end{bmatrix} + \frac{1}{3L_S}\begin{bmatrix} 2 & -1 & -1 \\ -1 & 2 & -1 \\ -1 & -1 & 2 \end{bmatrix}\begin{bmatrix} V_a \\ V_b \\ V_c \end{bmatrix} - \frac{1}{L_S}\begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}\begin{bmatrix} E_{\mathrm{Sa}} \\ E_{\mathrm{Sb}} \\ E_{\mathrm{Sc}} \end{bmatrix} \tag{4.13}$$

where the instantaneous neutral to ground voltage expression $V_{\mathrm{NG}} = \frac{1}{3}(V_a + V_b + V_c)$ has been used. Now, if we apply to the different vectors in (4.13) the $T_{\alpha\beta}$ transformation, i.e., we replace each vector $\vec{x}_{abc}$ with $T_{\alpha\beta}^T \vec{x}_{\alpha\beta}$, after some rearrangement, we get

$$\frac{\mathrm{d}}{\mathrm{dt}}\vec{I}_{\alpha\beta} = -\frac{R_S}{L_S} T_{\alpha\beta} I_3 T_{\alpha\beta}^T \vec{I}_{\alpha\beta} + \frac{1}{3L_S} T_{\alpha\beta}\begin{bmatrix} 2 & -1 & -1 \\ -1 & 2 & -1 \\ -1 & -1 & 2 \end{bmatrix} T_{\alpha\beta}^T \vec{V}_{\alpha\beta} - \frac{1}{L_S} T_{\alpha\beta} I_3 T_{\alpha\beta}^T \vec{E}_{S\alpha\beta}, \tag{4.14}$$

where I_3 is the 3×3 identity matrix. Simplifying the matrix products, we find the following result:

$$\frac{\mathrm{d}}{\mathrm{dt}}\vec{I}_{\alpha\beta} = -\frac{R_S}{L_S} I_2 \vec{I}_{\alpha\beta} + \frac{1}{L_S} I_2 \cdot \vec{V}_{\alpha\beta} - \frac{1}{L_S} I_2 \vec{E}_{S\alpha\beta}, \tag{4.15}$$

where I_2 represents the 2×2 identity matrix. Please note that the contribution of V_{NG} to the system dynamics, known as phase interference, has been cancelled by the application of the $T_{\alpha\beta}$ transformation, as expected. Equation (4.15) shows that the equations for the two axes are now fully decoupled, i.e., totally independent from each other. In addition, the structure and parameters of the two-axis system are identical to that of the original three-phase system. Consequently, under the assumption of symmetrical and balanced load, it is not necessary to have any model adjustment and the design of the current regulator for the α and β axes can be done exactly as on a single-phase inverter.

This very important result implies that everything we have said about PI and predictive digital current control in the previous chapter can be immediately used also in three-phase inverters. The only additional elements we have to take into account are the implementation of a suitable SVM algorithm and of the $\alpha\beta$ transformation.

4.3 THE ROTATING REFERENCE FRAME CURRENT CONTROLLER

Once the three-phase inverter of Fig. 4.2 has been proved to be completely equivalent to a couple of independent single-phase inverters, other questions may be asked. Indeed, one could wonder whether the mapping of the system in the $\alpha\beta$ reference frame could be somehow exploited to *improve* the current controller dynamic performance.

While this is not possible for the dead-beat controller, which already provides the best possible dynamic response among digital current regulators, in the case of the PI current controller the answer to the above question is affirmative. The implementation of the so-called *rotating reference frame* controller indeed allows a significant improvement of the reference tracking capabilities of the PI regulator. This section is therefore dedicated to the illustration of the basic principles behind this solution.

The first concept we have to introduce is that of Park's transformation, a very well-known tool for electrical machine designers.

4.3.1 Park's Transformation

The idea behind Park's transformation is quite simple. Instead of mapping the three-phase inverter and its load onto a fixed two-axis reference frame, this transformation maps it onto a two-axis synchronous rotating reference frame. This practically means moving from a static coordinate transformation to a dynamic one, i.e., to a linear transformation whose matrix has time varying coefficients.

Before entering into the mathematical details, we may refer to Fig. 4.7 to get an idea of Park's transformation's meaning. The transformation defines a new set of reference axes, called

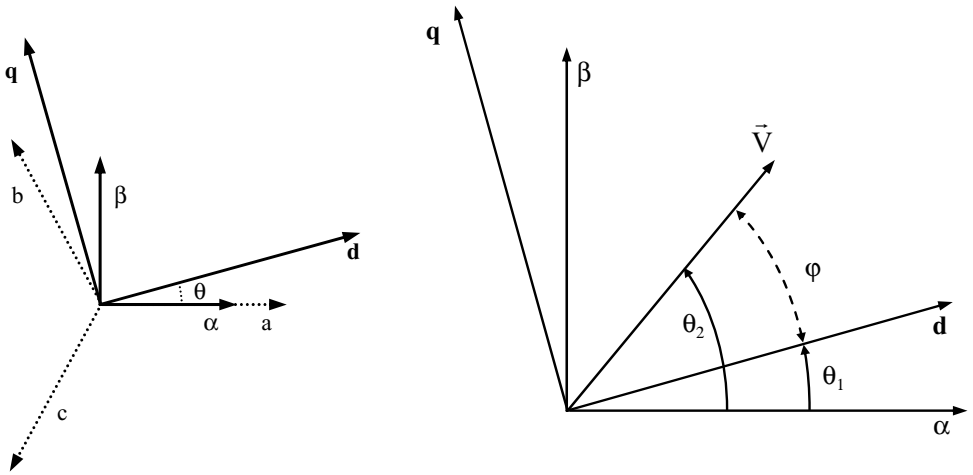

FIGURE 4.7: Vector diagrams for Park's transformation.

d and q, which rotate around the static $\alpha\beta$ reference frame at a constant angular frequency ω. Referring to Fig. 4.7, this means that $\theta = \omega t$.

We have seen in section 4.1 that the application of the $\alpha\beta$ transformation to a triplet of symmetrical and balanced sinusoidal signals (4.7) turns them into a couple of 90° shifted sinusoidal signals (4.8), whose geometrical interpretation can be that of the rotating vector, $\vec{V}$. The rotating vector angular speed equals the angular frequency of the original voltage triplet, which we can consider the *fundamental* frequency of our three-phase system. Now, if the angular speed of the rotating vector equals ω, what happens is that, in the dq reference frame, the vector $\vec{V}$ is not moving at all! Referring again to Fig. 4.7, what we have just seen implies that angles θ_1 and θ_2 will both increase with angular frequency ω, while angle φ will be *constant* and so will be the lengths of vector $\vec{V}$ projections on the d and q axes.

The advantage of using Park's transformation is represented exactly by the fact that sinusoidal signals with angular frequency ω will be seen as *constant* signals in the dq reference frame. We have seen how a PI controller, especially a digital PI controller, can be affected by a nonnegligible tracking error with respect to sinusoidal reference signals, which is due to the limited closed loop gain at the frequency of interest. In contrast, a PI controller can guarantee zero tracking error on constant signals, thanks to the built-in integral action. Therefore, if a PI controller is implemented in the dq reference frame, without any additional provision, its tracking error with respect to sinusoidal signals having angular frequency equal to ω, i.e., to the frequency of Park's transformation, will become equal to zero. As we will see in the following, this principle is exploited in the implementation of the so-called synchronous frame current controllers, where the Park's transformation angular speed is chosen exactly equal to the three-phase system fundamental frequency.

We can now show the mathematical formulation of Park's transformation. Considering Fig. 4.7, it is easy to demonstrate that this is given by the following matrix:

$$\begin{bmatrix} x_d \\ x_q \end{bmatrix} = T_{dq} \begin{bmatrix} x_\alpha \\ x_\beta \end{bmatrix} = \begin{bmatrix} \cos\theta & \sin\theta \\ -\sin\theta & \cos\theta \end{bmatrix} \begin{bmatrix} x_\alpha \\ x_\beta \end{bmatrix}, \tag{4.16}$$

where $\theta = \omega t$. Please note that, using complex phasorial representation of the vectors, (4.16) can be very simply expressed as

$$\vec{x}_{dq} = x_d + \mathrm{j}x_q = \left(x_\alpha + \mathrm{j}x_\beta\right)\left(\cos\theta - \mathrm{j}\sin\theta\right) = \vec{x}_{\alpha\beta}\mathrm{e}^{-\mathrm{j}\theta}. \tag{4.17}$$

It is easy to show that T_{dq} is associated with another orthonormal base of the R^2 space, so that its inverse can be immediately found:

$$\begin{bmatrix} x_\alpha \\ x_\beta \end{bmatrix} = T_{dq}^T \begin{bmatrix} x_d \\ x_q \end{bmatrix} = \begin{bmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{bmatrix} \begin{bmatrix} x_d \\ x_q \end{bmatrix}, \tag{4.18}$$

which, using the complex phasorial notation, can be simply written as $\vec{x}_{\alpha\beta} = \vec{x}_{dq}\mathrm{e}^{+\mathrm{j}\theta}$.

As we did in the previous section, we can as well investigate the transformation of the system state equations, determined by the application of Park's transformation. To do this, all we need to do is consider equation (4.15) and use (4.18) on the left- and right-hand sides of it. Almost immediately we find the following result:

$$\frac{\mathrm{d}}{\mathrm{dt}}\vec{I}_{dq} = \begin{bmatrix} -\frac{R_S}{L_S} & +\omega \\ -\omega & -\frac{R_S}{L_S} \end{bmatrix} \vec{I}_{dq} + \frac{1}{L_S} I_2 \vec{V}_{dq} - \frac{1}{L_S} I_2 \vec{E}_{Sdq}, \tag{4.19}$$

which shows a very interesting fact. The two system dynamic equations are now complicated by the cross-coupling of the two axes, i.e., they are no longer independent from each other. This is the reason why, in control schemes such as the one of Fig. 4.8, decoupling feed-forward paths are sometimes included. These make the system dynamics totally identical to those of the original one.

To complete this brief discussion of Park's transformation we need to say that, in addition to what we have seen so far, it is also possible to implement the so-called *inverse sequence* Park's transformation. This is nothing but the transformation we have just presented, which we may now identify as the *direct sequence* Park's transformation, where the direction of the dq axes rotation is assumed to be *inverted*. It is immediate to verify that the implementation of the inverse-sequence transformation simply amounts to swapping the roles of (4.16) and (4.18).

One could wonder why the inverse sequence transformation is ever required, since we have shown that the direct sequence transformation is capable of mapping all the $\alpha\beta$ space without any loss of information. The reason is that, so far, we have considered balanced and symmetrical

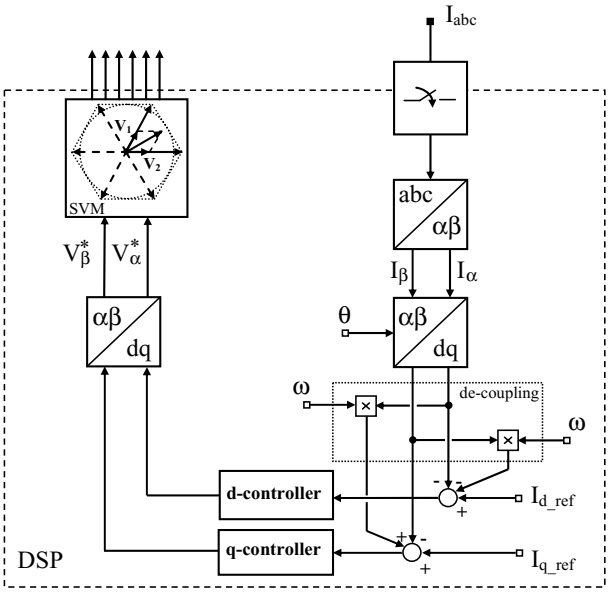

FIGURE 4.8: Organization of a three-phase digital current controller in the *dq* reference frame.

three-phase systems, but, more generally, impedance unbalances and/or unsymmetric voltage sources can be found. In this case, a three-phase system can be shown to be equivalent to the superposition of a direct sequence system and an inverse sequence system, both of them symmetrical and balanced and so both properly describable in the $\alpha\beta$ reference. If we neglect the so-called *omopolar* components, the superposition of *both* the direct and the inverse sequence two-phase systems is exactly equivalent to the original three-phase system, while none of them is by itself. Of course, in the case of zero or negligible unbalance/unsymmetry, the inverse sequence components will be accordingly zero or negligible, which motivates, in the majority of practical cases, the use of (4.16) and (4.18) alone.

Finally, it is important to underline that, because the elements of T_{dq} and T_{dq}^{T} are not time invariant, the application of Park's transformation, differently from the $\alpha\beta$ transformation, affects the system dynamics. This means that any controller, designed in the *dq* reference frame, is actually equivalent to a stationary frame controller that *does not maintain* the same frequency response. To keep the discussion reasonably simple, we refer, for the moment, to analog current regulators. In the end, we will see how to adapt our conclusions to digital current regulators.

4.3.2 Design of a Rotating Reference Frame PI Current Controller

For the reasons previously explained, we are very interested in PI controllers, which, once implemented in the rotating reference frame, can offer zero steady-state tracking error [6] for sinusoidal signals whose angular frequency is equal to ω. In some applications, where the phase

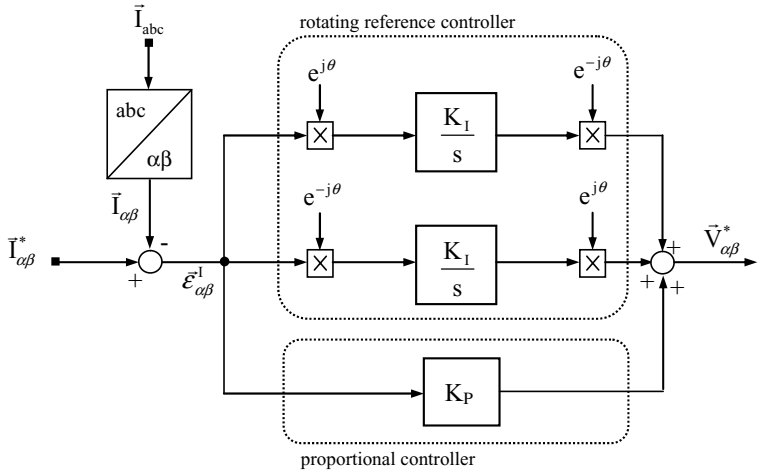

FIGURE 4.9: Organization of the rotating reference controller in the continuous time domain.

error between the current reference signal and the inverter phase current must be as small as possible, this indeed represents the optimal solution. In order to properly design the PI controller in the *dq* reference frame, we need to understand what is the corresponding stationary frame controller. We begin by presenting a suitable model of the rotating reference frame current controller. This is shown in Fig. 4.9.

There are several important issues related to Fig. 4.9. First of all, as can be seen, the schematic is drawn in vector form, i.e., the indicated quantities represent bidimensional currents and voltages. Accordingly, Park's transformation is simply represented by multiplication with the complex phasors $e^{\dot{x}j\theta}$, where, as usual, $\theta = \omega t$. Secondly, both direct and inverse sequence transformation are taken into account, so as to make the schematic representative of as many practical cases as possible.

Even more importantly, the PI controller has been decomposed into a parallel structure, as it is always possible to do. Once this is done it is immediate to realize that, since the proportional gain is time invariant, the Park's transformation operators that would apply to it can be eliminated, as they are completely ineffective on a constant gain. Finally, the two proportional parts, respectively operating on the direct and inverse sequence components of the current error, can be unified, because they turn out to be exactly identical. Therefore, in Fig. 4.9 and the following, gain K_P must be interpreted as the sum of the proportional gains of the direct and inverse sequence controllers.

As can be seen, what we ended up with is a proportional controller whose gain can be designed exactly as that of a single-phase proportional current controller, which operates in parallel with two rotating reference frame integral controllers. The integral gain can be designed recalling that its effect will be to bring down to zero the tracking error with respect

to sinusoidal reference signals having an angular frequency equal to ω. Of course, the higher the integral gain, the faster the achieved speed of response. An interesting problem, however, is how to predetermine and control the settling time of the integral controller action, so as to avoid ringing, for example, in the presence of a step reference variation. This problem can be effectively solved considering a different interpretation of the rotating reference controller, as is presented in the next section.

To conclude the discussion of rotating reference frame PI current controllers, we have to address the problem of its digital implementation. Of course, it is highly recommendable that this solution is implemented digitally, as this makes it very simple to implement the different coordinate transformations involved in the controller operation. Once again, discretization is a very useful tool to accomplish this task. Based on what we have just seen, it is easy to understand that its application to the proportional part of the controller poses no significant problem. The only caution we need to apply may be in the continuous time domain design phase, where the phase margin we require for the open loop gain might be slightly oversized to cope with the calculation delay.

The application of discretization to the integral part of the controller is also relatively simple, because we have seen how Euler, or trapezoidal, numerical integration can effectively replace the analog integrator. The only caution we need to apply is the adjustment of the gain value, which has to be multiplied by the sampling period. In conclusion, a possible schematic of the digital version of the controller presented in Fig. 4.9, is that shown in Fig. 4.10. Note that the different vectors of Fig. 4.10 have now to be interpreted as sampled signals. It is also possible to see that the integral controllers have been discretized using the backward Euler method.

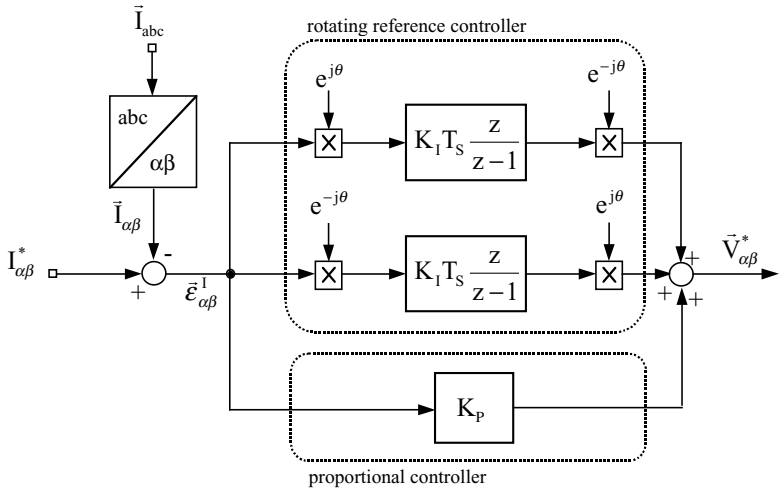

FIGURE 4.10: Discretized version of the rotating reference PI current controller.

4.3.3 A Different Implementation of the Rotating Reference Frame PI Current Controller

We now want to derive an equivalent stationary frame controller to replace the integral part of the rotating reference frame PI of Fig. 4.9. In order to do that, we consider the Laplace operator and, in particular, the following property:

$$\left[L\left(e^{\lambda t}\cdot f(t)\right)\right](s) = \left[L(f)\right](s-\lambda), \tag{4.20}$$

which is going to prove very useful to our purpose. Theorem (4.20) says that the multiplication by $e^{\lambda t}$ in the time domain results into a frequency translation in the s-domain. This means that, in the controller representation of Fig. 4.9, we can operate the substitution shown in Fig. 4.11.

Doing that, we obtain an equivalent stationary frame controller both for the direct sequence and for the inverse sequence components of the voltage reference vector, $\vec{V}^*_{dq+}$ and $\vec{V}^*_{dq-}$ respectively. We then find, summing the two components, that the transfer function between the current error vector and the voltage reference vector, in the stationary reference frame, is as follows:

$$\frac{\vec{V}^*_{\alpha\beta}(s)}{\vec{\varepsilon}_{\alpha\beta}(s)} = \frac{\vec{V}^*_{\alpha\beta+}(s)}{\vec{\varepsilon}_{\alpha\beta}(s)} + \frac{\vec{V}^*_{\alpha\beta-}(s)}{\vec{\varepsilon}_{\alpha\beta}(s)} = \frac{K_I}{s+j\omega} + \frac{K_I}{s-j\omega} = 2K_I\frac{s}{s^2+\omega^2}. \tag{4.21}$$

This very important result [7, 8] shows that the stationary frame equivalent of the rotating frame controller integral part is just a second-order resonant band pass filter, whose resonance frequency is exactly equal to ω. It is worth noting that the resonant filter presents zero damping factor and that the role of the integral gain is to determine the filter selectivity and, consequently, its settling time in response to perturbations. From (4.21) we see that increasing the K_I value determines a reduction of the filter selectivity and, consequently, a faster settling time. In contrast, reducing K_I determines a higher filter selectivity and, consequently, a longer settling time. A detailed explanation of the design criteria for this regulator, which allows us to properly set the K_P and K_I gains, is reported in Aside 7.

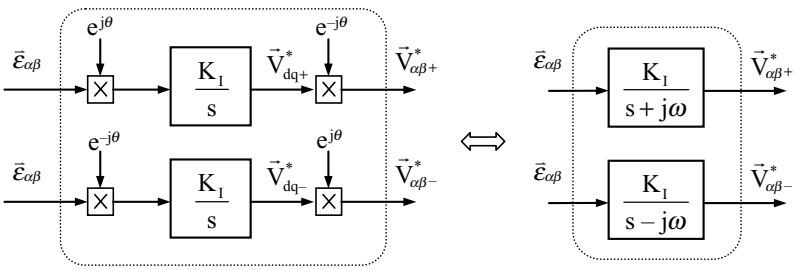

FIGURE 4.11: Laplace transformation of the rotating reference controller.

Aside 7. Design of a Stationary Frame Current Regulator with Zero Steady-State Error

In Asides 2 and 3, we have determined the proportional and integral gains of a PI current controller. In this aside, we would like to illustrate a simple design example of a stationary frame current regulator composed, as shown in Fig. A7.1, of a proportional gain K_P and a single resonant controller $F_o(s)$:

$$F_o(s) = \frac{2K_I s}{s^2 + \omega_o^2}, \qquad (A7.1)$$

tuned at the fundamental frequency ω_o. The considered parameter values are $V_{DC} = 250$ V, $f_o = 60$ Hz, $L_S = 3.5$ mH, $R_S = 1\Omega$, $f_S = 10$ kHz, $G_{TI} = 0.1$. As done in Aside 2, the controller design is first performed in the analog domain and then translated in the z-domain using a discretization process. The proportional gain K_P setting is based on the desired cross-over angular frequency ω_{CL}, as in any conventional PI control. Assuming that the current loop bandwidth is 1/10 of the switching frequency (i.e., $\omega_{CL} = 0.1\omega_S$), $K_P \cong \omega_{CL} L_S/(2V_{DC} G_{TI}) = 0.88$. Instead, the integral gain K_I of the resonant regulator $F_o(s)$ is based on the desired transient response and on the specified phase margin ph_m. Indeed, since $\omega_{CL} \gg \omega_o$, $F_o(s) \approx 2K_I/s$, so that the design of K_I is the same as that of the PI controller of Aside 2, except for a factor 2. In our case, setting the phase margin $ph_m = 45°$, we have $K_I = K_{IN} = 1.25$ krad s^{-1}. Figs. A7.2 and A7.3 report the current reference, I_{OREF}, which is a sinusoidal waveform at ω_o, the output current I_O and the current error ε_I. As can be seen, the stationary frame current regulator is able to ensure zero steady-state errors for any reference or disturbance, whose frequency component is at ω_o.

In order to highlight the properties of the resonant controller, we have reported in Fig. A7.4 the current control loop gain using three different integral gains: (a) $K_I = 2K_{IN}$,

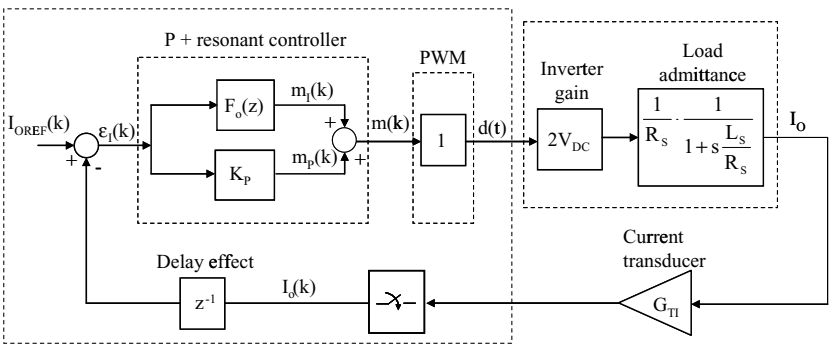

FIGURE A7.1: Block diagram representation of a stationary frame current regulator with zero steady-state error.

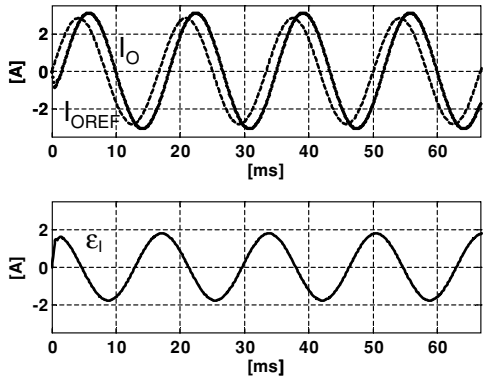

FIGURE A7.2: (top) current reference I_{oref} current I_o and (bottom) current error ε_I with PI.

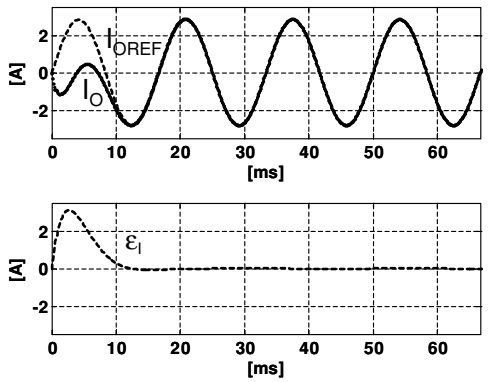

FIGURE A7.3: (top) current reference I_{oref} current I_o and (bottom) current error ε_I with P+Resonant control.

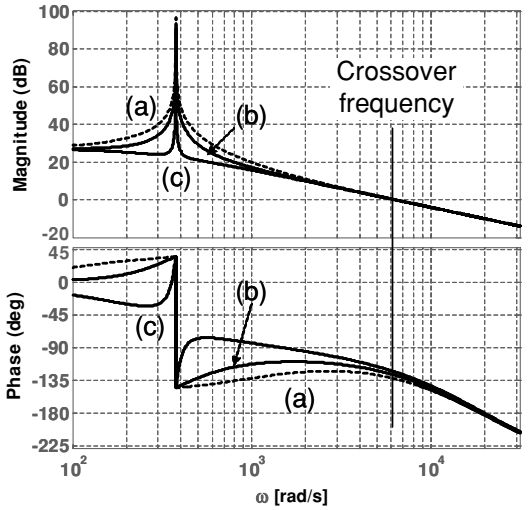

FIGURE A7.4: Current control loop gain for (a) $K_I = 2\ K_{IN}$; (b) $K_I = K_{IN}$; (c) $K_I = K_{IN}$ /10.

(b) $K_I = K_{IN}$, (c) $K_I = 0.1K_{IN}$. As can be seen, the integral gain K_I determines the filter selectivity and, consequently, its settling time in response to perturbations at the angular frequency ω_o; thus the higher the K_I, the lower the filter selectivity and, consequently, the faster the settling time. In contrast, the lower the K_I, the higher the filter selectivity and, consequently, the longer the settling time.

In order to understand the settling time of the resonant controller and to establish an alternative second design criterion for the integral gain K_I, we may interpret the controller organization of Fig. A7.1 as a multiloop scheme, where we first close the current control only with the proportional gain K_P. Then, the resonant filter $F_o(s)$ is designed so as to compensate the residual errors. From this point of view, the transfer function that the resonant controller $F_o(s)$ is going to compensate, once the proportional controller loop is closed, is

$$G_o(s) = \frac{m_I(s)}{e_I(s)} = \frac{1}{K_P} \underbrace{\frac{K_P G(s)}{1 + K_P G(s)}}_{W_p(s)} = \frac{1}{K_P} W_p(s), \qquad (A7.2)$$

where $W_p(s)$ is the transfer function between the current reference and the output current, when only the proportional controller is active. In general, $W_p(s)$ has a gain close to unity up to the desired bandwidth ω_{CL}. In our case $W_p(s)$ is shown in Fig. A7.5. This controller interpretation leads to the following very interesting observations.

(1) In the synchronous reference frame, the integrator controller K_I/s compensates a transfer function which is roughly approximated by $1/K_p$. Thus, the integral gain K_I can be designed given the desired cross-over frequency ω_{ro} (or desired time constant $t_{ro} = 1/\omega_{ro}$), i.e.,

$$K_I = \frac{K_P}{t_{ro}} = 2.2\frac{K_P}{t_r}, \qquad (A7.3)$$

where $t_r = n_o T_o$ is the desired response time (evaluated between 10% and 90% of a step response) for the fundamental frequency f_o. In our case, $t_r = 2$ ms or $n_o = 0.12$. Since we are reasoning in the synchronous reference frame, the time constant is referred to the transient of the *envelope* of the fundamental frequency f_o. The transient response in Fig. A7.3 is longer, since the step reference variation contains other frequencies, besides the fundamental one, f_o.

(2) Taking into account that any resonant controller determines a $+90^\circ$ phase shift before the resonance frequency and -90° phase shift after the resonance frequency, it is intuitive to understand that the resonant controller will be able to compensate only those frequencies f_o for whom $W_p(j2\pi f_o)$ has a phase shift lower than -90°, so that the cross over of the -180° stability limit is avoided. This imposes a limitation of the maximum angular frequency that it

is possible to compensate, which must be kept lower than ω_L, as indicated in Fig. A7.5. This issue may be interesting for high-order harmonic compensation, as described in Aside 8.

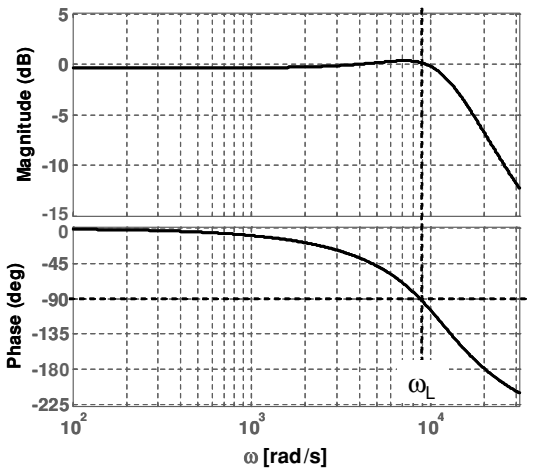

FIGURE A7.5: Bode diagram of $W_p(s)$.

Aside 8. Stationary Frame Resonant Regulator: Extension to High-Order Harmonic Components and Introduction of a Phase Lead Compensation.

The approach presented in Section 4.3.3 can be extended to multiple harmonic compensation [8]. A typical example is the harmonic compensation in active power filters, where the current reference contains several harmonic components. The most straightforward approach for the compensation of the harmonic frequencies is the introduction of a resonant filter for each harmonic component to be compensated. Thus, referring to Fig. A7.1, F_o becomes

$$F_o(s) = \sum_{k \in N_k} \frac{2\,K_{Ik}\,s}{s^2 + (k\omega_o)^2}. \qquad (A8.1)$$

N_k is the set of selected harmonic frequencies. Following the reasoning illustrated in the last part of Aside 7, K_{Ik} design is based on the transient response desired for each harmonic component. Thus, the design of each integrator gain K_{Ik} is given by

$$K_{Ik} = \frac{2.2K_p}{t_r} = \frac{2.2K_p}{n_{ok}\,T_S}, \qquad (A8.2)$$

where $t_r = n_{ok}\, T_S$ is the desired response time (evaluated between 10% and 90% of a step response) for the generic harmonic k and n_{ok} is the number of supply periods T_S. There is, however, a bandwidth limitation that applies to each harmonic component, given by angular frequency ω_L. Indeed, even for angular frequencies below ω_L, the transient response of the harmonic component may be lightly damped. As an example, using the parameters of Aside 7, we have set the harmonic component at 75% of ω_L (i.e., $k = 17$). The result is reported in Fig. A8.1, which clearly shows a lightly damped behavior.

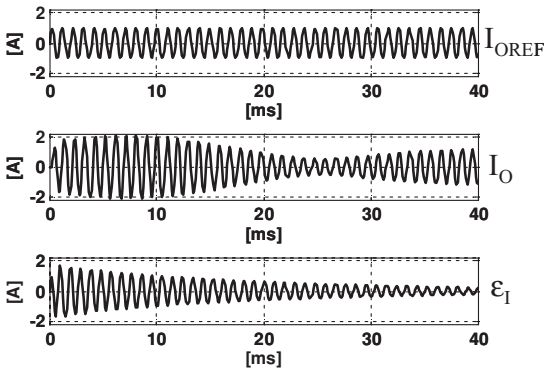

FIGURE A8.1: From top to bottom: current reference I_{OREF}, current I_O, and current error ε_I when the reference current is at $k\omega_0$ and a resonant filter tuned at harmonic k is used.

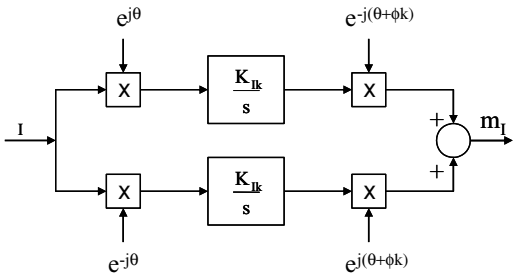

FIGURE A8.2: Rotating reference frame controller with phase lead ϕ_k.

This problem can be easily attenuated compensating the delay of the feedback loop by introducing a phase lead effect in the controller. As shown in Fig. A8.2, the phase lead ϕ_k is added when the outputs of the synchronous frame regulators $R_{kDC}(s)$ are transformed back to the stationary reference frame coordinates. Using theorem (4.20), the relation between synchronous reference frame regulators $R_{kDC}(s)$ and stationary reference frame regulators

$R_{k\mathrm{AC}}(s)$ becomes

$$R_{k\mathrm{AC}}(s) = \cos\phi_k[R_{k\mathrm{DC}}(s - \mathrm{j}k\omega_\mathrm{o}) + R_{k\mathrm{DC}}(s + \mathrm{j}k\omega_\mathrm{o})] + \mathrm{j}\sin\phi_k[R_{k\mathrm{DC}}(s - \mathrm{j}k\omega_\mathrm{o}) - R_{k\mathrm{DC}}(s + \mathrm{j}k\omega_\mathrm{o})] \quad (A8.3)$$

If $R_{k\mathrm{DC}}(s) = K_{\mathrm{Ik}}/s$, (A8.3) becomes

$$R_{k\mathrm{AC}}(s) = \sum_{k\in N_k} \frac{2K_{\mathrm{Ik}}\left(s - \frac{\sin(\phi_k)}{\cos(\phi_k)}\right)\cos(\phi_k)}{s^2 + (k\omega_o)^2}, \quad (A8.4)$$

which, for $\phi_k = 0$, corresponds to (A8.1). The leading angle ϕ_k can be set equal to the delay at frequency k of the transfer function $W_\mathrm{p}(s)$. The results of this provision is described in Fig. A8.3. Comparing this result with Fig. A8.1, we can clearly see the advantages of the introduction of a phase lead angle ϕ_k.

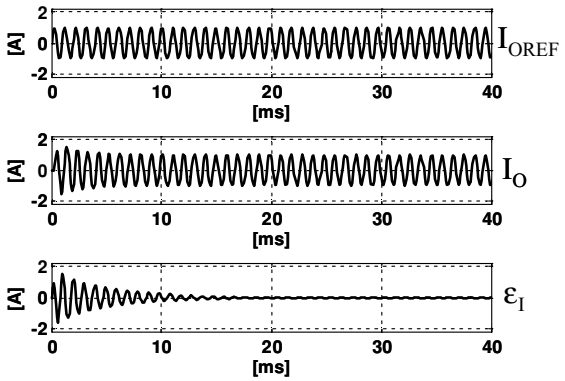

FIGURE A8.3: From top to bottom: current reference I_{OREF}, current I_O, and current error ε_I when the reference current is at $k\omega_0$ and a resonant filter tuned with lead angle ϕ_k at harmonic k is used.

In order to understand the performance of the proposed controller in a typical active filter application, we have simulated a reference signal I_{OREF}, which includes the fundamental component, the fifth and the seventh components, both with an amplitude equal to 50% of the fundamental one. Accordingly, we have implemented a resonant controller that includes the compensation of the fundamental fifth and seventh harmonic components.

The gain of the resonant controller has been set so as to have a response time equal to one fundamental period for all three harmonic components. The results are reported in Fig. A8.4. The figure shows how the residual error is reduced to zero after about one fundamental period, which is consistent with the specified dynamic response. As a comparison, we have

simulated an ideal dead-beat current controller, which ensures reference tracking with a two-sample delay, reporting the results in Fig. A8.5. Note that, at the end of the simulated time interval, the residual error is still quite high, even if the dead-beat can be considered a very fast current controller. Of course, if higher order harmonics were to be compensated (13th, 15th, etc.), the advantages of the resonant controller would be even greater than what the results reported in this example show.

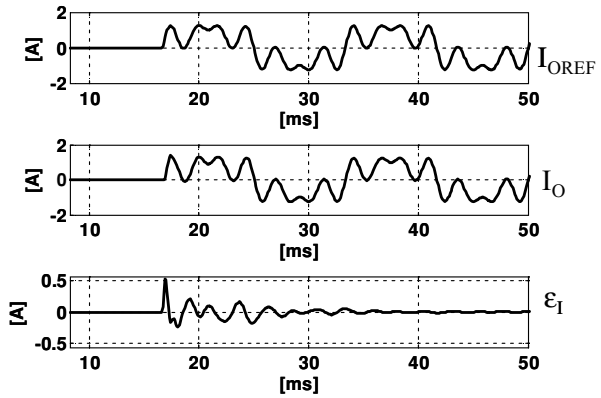

FIGURE A8.4: Transient response of a resonant controller $F_O(s)$: (top) current reference I_{OREF} current I_O; (bottom) current error ε_I.

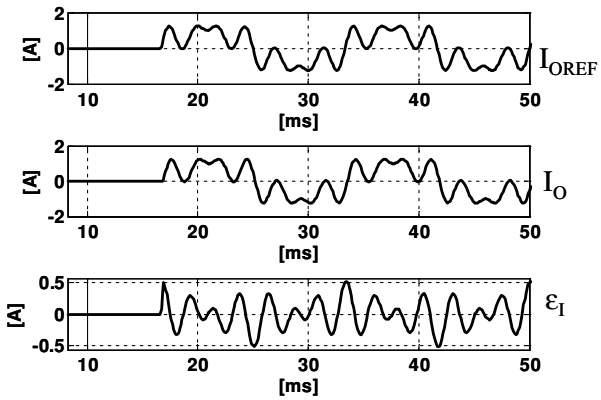

FIGURE A8.5: Transient response of an ideal dead-beat current controller: (top) current reference I_{OREF} current I_O; (bottom) current error ε_I.

The equivalence of the rotating reference frame PI controller with a proportional controller parallel connected to a tuned resonant filter suggests an alternative implementation of the controller

that, not requiring the computation of Park's transformation, may offer a significant reduction of signal processing requirements for the control algorithm. Indeed, it is worth mentioning that the implementation of stationary frame resonant controllers, instead of synchronous reference frame controllers, has received, starting from the year 2000, a significant attention from several research groups around the world, at least for those applications (UPS, PFC, active power filters, etc.) where the frequencies to be compensated are almost constant. Of course, the direct implementation in the discrete time domain of resonant filters with zero or very small damping factors requires some care during the discretization process, in order to avoid warping effects that could shift the resonant frequencies, moving them out of the desired locations.

REFERENCES

[1] N. Mohan, T. Undeland, and W. Robbins, *Power Electronics: Converters, Applications and Design*. New York: Wiley, 2003.

[2] J. Kassakian, G. Verghese, and M. Schlecht, *Principles of Power Electronics*. Reading, MA: Addison Wesley, 1991.

[3] H. W. Van Der Broeck, H. C. Skudenly, and G. V. Stanke, "Analysis and realization of a pulsewidth modulator based on voltage space vectors," *IEEE Trans. Indust. Appl.*, Vol. 24, No. 1, pp. 142–150, 1988.doi.org/10.1109/28.87265

[4] D. G. Holmes and T. A. Lipo, "Pulse width modulation for power converters: principles and practice," *2003 IEEE Press Series Power Eng.*

[5] J. Holtz, W. Lotzkat, and A. Khambadkone, "On continuous control of PWM inverters in the overmodulation range including the six-step mode,"*1992 Int. Conf. Indust. Electron. Control Instrum.* (*IECON*), pp. 307–312.

[6] D. N. Zmood and D. G. Holmes, "Stationary frame current regulation of PWM inverters with zero steady-state error," *IEEE Trans. Power Electron.*, Vol. 18, No. 3, pp. 814–822, 2003.doi.org/10.1109/TPEL.2003.810852

[7] D. N. Zmood, D. G. Holmes, and G. H. Bode, "Frequency-domain analysis of three-phase linear current regulators," *IEEE Trans. Indust. Appl.*, Vol. 37, No. 2, pp. 601–610, 2001.doi.org/10.1109/28.913727

[8] P. Mattavelli, "Synchronous-frame harmonic control for high-performance AC power supplies," *IEEE Trans. Indust. Appl.*, Vol. 37, No. 3, pp. 864–872, 2001.doi.org/10.1109/28.924769

CHAPTER 5

External Control Loops

In the previous chapters we have presented some examples of current control loop implementations, both for single- and for three-phase voltage source inverters. We have discussed how to design a PI current controller in the continuous time domain and how to turn it into a discrete time, or digital, controller. We also introduced the principles of dead-beat, predictive current control. In all these cases, we have seen how the presence of a current control loop actually turns the VSI into a controlled current source with predetermined speed of response and reference tracking accuracy.

However, there are several applications of VSIs where the implementation of a current control loop is just the first step to be taken. For example, in some cases, the control objective is not simply to develop a controlled current source, but rather to turn the VSI into a controlled *voltage* source. In other cases, the controlled current source is automatically regulated by an *external control loop* that is driven by another dynamic variable in the system, like, for instance, the rotational speed of an electrical motor. In these circumstances, the current loop becomes the most internal control loop in a *multiloop* arrangement of the VSI controller.

The purpose of this chapter is to present an overview of multiloop control organizations, discussing some examples of external control loop design. Because a controlled current source can be used for a very large spectrum of different applications, it is practically impossible to deal with all. As we did before, on this occasion we will as well limit our presentation to some typical application cases. In addition, we will limit the discussion to single-phase VSIs, since we have shown, in the previous chapter, how the results can be almost directly applied to three-phase converters as well.

5.1 MODELING THE INTERNAL CURRENT LOOP

The setup of an external control loop around an existing current loop, typical of all multiloop VSI control arrangements, poses questions similar to those we have considered when, discussing the design of a current controller, we first tackled the static and dynamic modeling of PWM. Once again, independent of the nature and purposes of an external controller, its design requires a suitable model of the internal loop, taking into account static gain and dynamic response.

The derivation of this model is, in practice, simplified by the fact that, in deriving the current controller, all the involved transfer functions, associated with the different static and dynamic components of the system under consideration have been identified and calculated, even if, in some cases, under simplifying assumptions. From this standpoint, the designer's task is now easier, since he or she has to deal with a completely linearized dynamic system.

Any dynamic system analysis software allows us to automatically calculate the closed loop transfer function of a given feedback controlled system, once the various involved transfer functions are specified and their interconnection is suitably described by the user's program. This is indeed a very useful way of checking one's results, but we do not recommend this as a design approach. The problem is that the resulting dynamic model is typically of high order, dependent on all the system parameters and affected by all the approximations that were used in the derivation of the single transfer functions. Its practical usefulness for the design of the outer loop is therefore limited.

To effectively design the external loop what the designer actually needs is a first-order simplified model of the internal loop, simple enough to be managed by pencil and paper calculations and, nevertheless, accurate enough to reproduce the system's dynamics in a reasonably faithful way.

In the large majority of cases, this simplified approach is sufficiently accurate to allow the successful design of any external loop. In some particular cases however, for example when the dynamic requirements for the external loop are demanding, the analytical, exact calculation of the internal loop response may be the only option available to the designer.

We can visualize the organization of a multiloop digital controller considering Fig. 5.1. As can be seen, an additional dynamic variable, indicated as the external variable X_O, is introduced, which, after proper conditioning and sampling, is processed by a digital controller. The output of the external variable controller is the reference signal for the current controller, which is therefore driven by the external control loop. The shaded area in Fig. 5.1 represents the part of the system that is controlled by the current regulator and that, consequently, will be seen by the external loop as a single, lumped, dynamic system. Please note that this includes, as well, the holder delay effect embedded in the PWM modulator that, consequently, will not affect the external loop design.

The simplest modeling approach consists in the derivation of the block diagram of Fig. 5.2. As can be seen, the blocks appearing in Fig. 5.1 and pertaining to the external control loop are explicitly indicated; the current controller, the inverter, and the load model are instead lumped in the current control block. Of course, to close the feedback loop, the definition of an additional transfer function that relates the converter output current with the external control variable has to be specified as well.

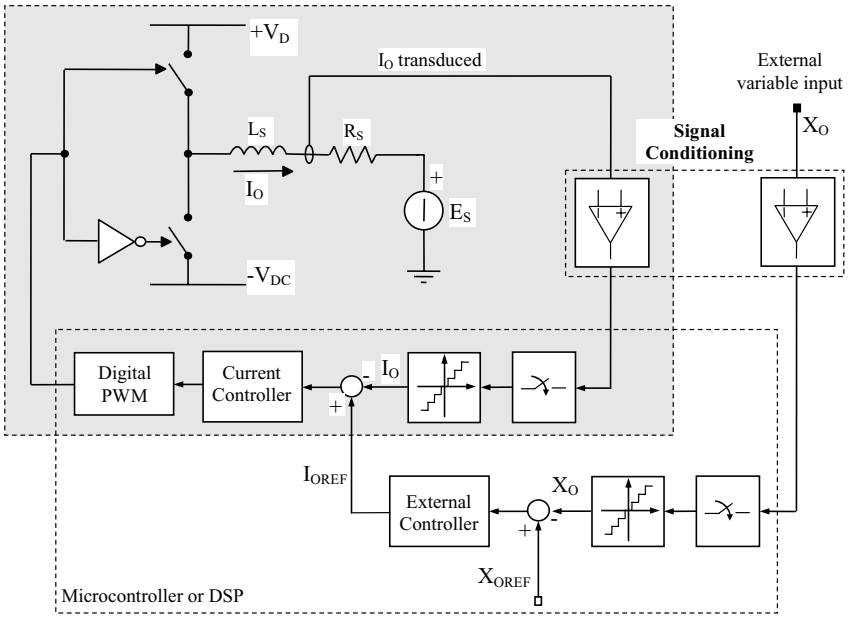

FIGURE 5.1: Typical organization of a multiloop digital controller.

While the latter transfer function depends on the particular application, and we will examine some particular cases in the following sections, the current control block model is independent of anything external to it. We can choose different model structures, based on the type of current controller we have actually implemented. An example that is generally apt to model PI or other conventional regulators is as follows,

$$\frac{I_O}{I_{OREF}}(s) = G_0 \frac{1}{1 + s\,\tau_{CC}}, \tag{5.1}$$

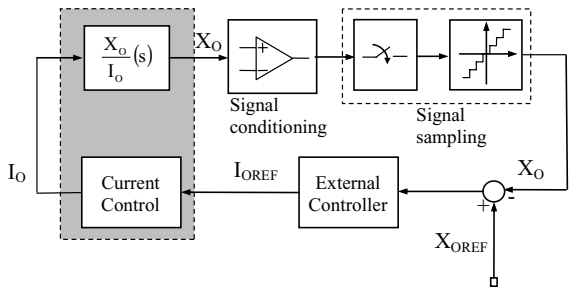

FIGURE 5.2: Block diagram of the external loop digital controller.

which can be used in the case where we proceed with the external controller synthesis in the continuous time domain and, later, apply some form of discretization. Instead, if the external controller synthesis has to be performed directly in the discrete time domain, we can consider a discrete time equivalent of (5.1), i.e.,

$$\frac{I_O}{I_{OREF}}(z) = Z\left[G_0 \frac{1}{1 + s\tau_{CC}}\right], \tag{5.2}$$

that may represent the discretization of (5.1), obtained by following any of the methods we have mentioned in Chapter 3. Please note that, in this latter case, the transfer function $\frac{X_O}{I_O}(s)$ has to be discretized as well.

The determination of gain and pole position for (5.1) and (5.2) is generally simple. The gain depends on the presence of possible scale factors in the current controller implementation. Typically, when the internal variables are represented fractionally, with unity as the full scale range value, gain G_0 equals the inverse of current transducer gain. Without loss of generality, we will assume that this is the case in the following examples. As far as the dynamic part of (5.1) is concerned, the idea is again to simply model the response delay of the current control loop. If current loop design has been properly performed, (5.1) represents a reasonable approximation of the closed loop gain for any time constant τ_{CC} chosen according to the following relation,

$$\tau_{CC} = \frac{1}{2\pi f_{CL}}, \tag{5.3}$$

where f_{CL} represents the crossover frequency considered for the current loop.

A different approach can be used in case the current controller has been implemented as a digital predictive regulator. In that case, the simplest approximation of the current loop is represented by

$$\frac{I_O}{I_{OREF}}(s) = G_0 \frac{1 - s\,T_S}{1 + s\,T_S}, \tag{5.4}$$

where the static gain is identical to that in (5.2), while the dynamic term is the first-order Padé approximation of the two modulation period delay of the current controller. Of course, for the dead-beat current controller the discrete time model equivalent to (5.4) is

$$\frac{I_O}{I_{OREF}}(z) = \frac{G_0}{z^2}, \tag{5.5}$$

which, in this case, contains no approximations. With the exception of (5.5), the modeling approaches we have just presented are fast and practical first-order approximations of the current loop: therefore, it is always recommendable to verify their validity comparing them to a plot of the exact closed loop current control transfer function, calculated by any of the available dynamic system analysis software packages.

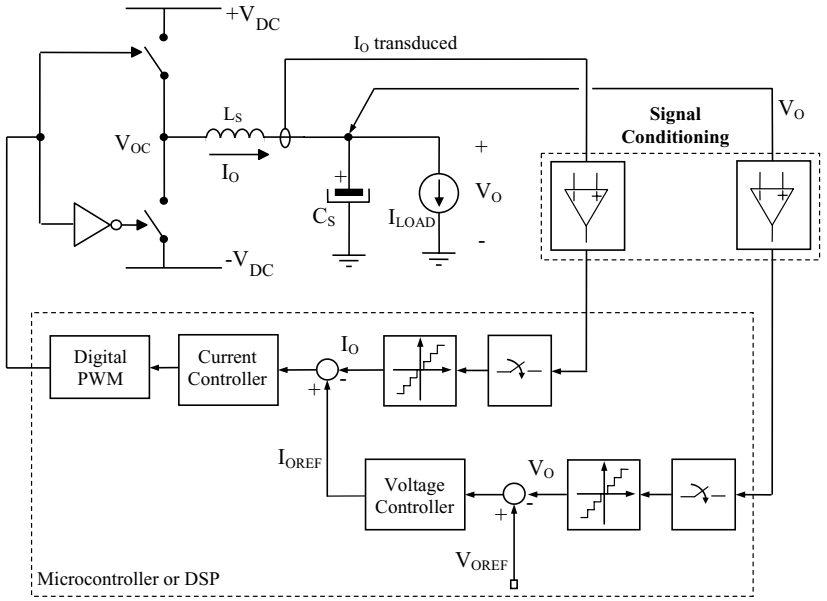

FIGURE 5.3: Typical organization of a single-phase UPS with digital control.

5.2 DESIGN OF VOLTAGE CONTROLLERS

A typical application field of VSIs is that of uninterruptible power supplies (UPSs). In this case, the voltage source inverter is used to implement a high-quality, controlled voltage source. The technology of UPS systems involves a whole lot of other fundamental issues, like, for example, those related to energy storage and to the management of the interaction with the utility grid [1, 2]. For the purpose of this textbook, we will limit the discussion to some possible, and basic, strategies for the implementation of digital controllers of the UPS inverter stage. According to what we have illustrated in the previous section, the typical controller organization is multiloop. The internal current control loop will be driven by an external voltage loop, as in the schematic diagram shown in Fig. 5.3 [3, 4].

There are several aspects related to Fig. 5.3 that deserve further clarification. In the first place, the structure of the inverter output filter has been modified with respect to what we have considered so far. The reason for this modification is that, in order to offer a relatively low impedance to external loads, schematically represented by a current source in Fig. 5.3, the converter output must be capacitive, at least in the frequency range of interest, which in the case of UPS is set around the line frequency. In addition, the output capacitor provides, at least partially, load power factor correction, and gives to the UPS an energy storage capability to sustain the load, in the absence of the primary source of energy, for a predetermined amount of

time, known as hold-up time. For the above reasons, the UPS inverter output filter will always have the configuration of Fig. 5.3.

It is worth noting that, in real cases, the load arrangement can be much more complex, e.g., including a transformer, so that the configuration of Fig. 5.3 represents just a simplified case study that will allow us a relatively easier discussion of the basic control design aspects.

A second important issue related to the considered UPS system configuration is the motivation for the presence of a current control loop. One could observe that, provided the load structure is as shown in Fig. 5.3, there is actually no need for a current loop. The direct control of the output voltage could be implemented as is done, for instance, in dc/dc converters, when direct duty-cycle control is implemented. This approach is of course perfectly possible and sometimes practically adopted too. Its main drawbacks are related to the protection of the inverter from accidental events like, for example, output short circuits. In this event, in order to avoid a potentially fatal overcurrent condition for the inverter switches, it is common practice to implement some form of current limitation mechanism that requires the implementation of current sensing and some form of, at least analog, signal processing. Therefore, there is no significant cost reduction in the removal of the current loop. In addition, the presence of an internal current loop allows us to decouple the second-order output filter dynamics. This fact, differently from what one could expect, does not necessarily offer an advantage in the achievement of a faster dynamic response. However, the modularity, flexibility, higher tolerance to parameter variations, and ease of design that characterizes multiloop solutions make this the preferred strategy in commercial UPS designs.

5.2.1 Possible Strategies: Large and Narrow Bandwidth Controllers

The possible strategies for the implementation of a UPS output voltage controller can be roughly divided into two different categories: (i) large bandwidth controllers and (ii) narrow bandwidth controllers.

The large bandwidth approach is aimed at the *instantaneous compensation* of any deviation of the output voltage from its reference. A typical problem in UPS systems is the limitation of the output voltage waveform harmonic distortion within acceptable, product standard compliant, levels. This is a particularly hard task when nonlinear distorting loads, such as diode bridge rectifiers with capacitive output filters, are connected to the UPS output. Large bandwidth output voltage controllers try to achieve the goal by extending the regulation bandwidth so much as to make it include a significant number of fundamental frequency harmonics (10 or more). We will see in the following how this can be a very demanding control specification. Clearly, if this is achieved, the compensation of unwanted harmonic components of the output voltage will be achieved automatically, at least up to the regulator bandwidth. Typical implementations

of this concept are linear PI regulators and dead-beat controllers. We will discuss both in the next section.

The narrow bandwidth approach is based on the following consideration. Examining the output voltage waveform distortion problem, one can realize that what is really needed is not the instantaneous compensation of all the undesired harmonic components. A harmonic compensation action that settles in a *few fundamental frequency periods* is actually enough to comply with product standards, provided that a relatively fast control of the fundamental harmonic component and a comparatively fast response to load variations is guaranteed. The implementation of this concept can be very diverse, ranging from repetitive-based controllers to the adoption selected harmonic compensation by means of tuned filters. We will see some examples of these strategies in the following section.

5.3 LARGE BANDWIDTH CONTROLLERS

This section is dedicated to the presentation of basic implementations of two output voltage control strategies for UPS systems, namely PI control and dead-beat control. The design approach, for both cases, closely resembles the one we have been following for the current controller implementation, where we have first come across these types of regulators. Therefore, in the following, we will discuss in detail only the aspects that are peculiar to voltage controllers, being the generalities identical to those described in Chapter 3.

5.3.1 PI Controller

The implementation of a digital voltage PI controller is based on the general block diagram of Fig. 5.2, where we are now in a position to determine all the involved transfer functions. Prior to that, we need to summarize the main characteristics of the circuit of Fig. 5.3. We assume that the UPS is built around our original case study VSI. The complete list of converter parameters is given by Table 5.1.

As can be seen, only some of the parameter values are the same as originally reported in Table 2.1. Indeed, the output voltage specifications, relatively to both amplitude and frequency, have been chosen, in the present case, so as to determine the operating conditions that are typical of UPS systems in use in various non-European countries around the world.

The design of the voltage controller requires the knowledge of the current controller dynamic characteristics. We can either assume that the current controller has been designed as in Chapter 2 and successively discretized or that we are dealing with a predictive current controller, of the type described in Chapter 3. One could point out that both these current controllers have been designed assuming a different inverter load configuration, in particular assuming the load voltage to be an exogenous input of the system and, as such, totally independent of the

TABLE 5.1: UPS Inverter Parameters

Rated ouput power, P_O	1000 (V A)
Phase inductance, L_S	1.5 (mH)
Output capacitor, C_S	68 (μF)
Output voltage, V_O	110 (V_{RMS})
Output frequency, f_O	60 (Hz)
DC link voltage, V_{DC}	250 (V)
Switching frequency, f_S	50 (kHz)
Current transducer gain, G_{TI}	0.1 ($V\ A^{-1}$)
Voltage transducer gain, G_{TV}	0.02 ($V\ V^{-1}$)

system's state variables (i.e., from the inverter output current I_O). It is immediate to see that, for the circuit of Fig. 5.3, this is no longer the case. However, it is possible to show that, for a typical UPS design, what we have seen in Chapters 2 and 3 is still valid and can be applied again.

A simple demonstration of this statement can be found in Fig. 5.4. The figure shows the Bode plot of the current control open loop gain, in the case of a PI controller designed exactly as outlined in Chapter 2, Aside 2. The plot is done both for the original load configuration (without capacitor) and for the new load configuration, including the output capacitor. The figure clearly demonstrates that at the crossover frequency, and around it, the magnitude and phase of the two open loop configurations coincide. This is not casual at all: in general, the output capacitor of a UPS inverter is sized to guarantee a certain (typically 50%) compensation of possible inductive loads (typical minimum load cos ϕ is 0.8), thus reducing the apparent load power the inverter has to generate. Because of that, differently from what happens in a dc/dc converter, in the UPS, the output capacitor is usually designed to operate at the *line frequency*. This implies that the inverter's second-order output filter has indeed a very low natural resonance frequency (about 500 Hz, in our example). This is extremely low with respect to the switching frequency, which implies that the filter impedance, close to the switching frequency, i.e., close to the typical desired crossover frequency of the current loop, is almost purely inductive. Therefore, designing a PI current controller on the second-order filter or designing it on the pure inductor, shorting the output capacitor, makes no practical difference.

The case of the predictive controller requires a little more caution, but we will now show that the same conclusion can be reached. In order to do that, we consider a state space linear modeling of the second-order filter (Fig. 5.3), which, recalling Aside 3, can be simply

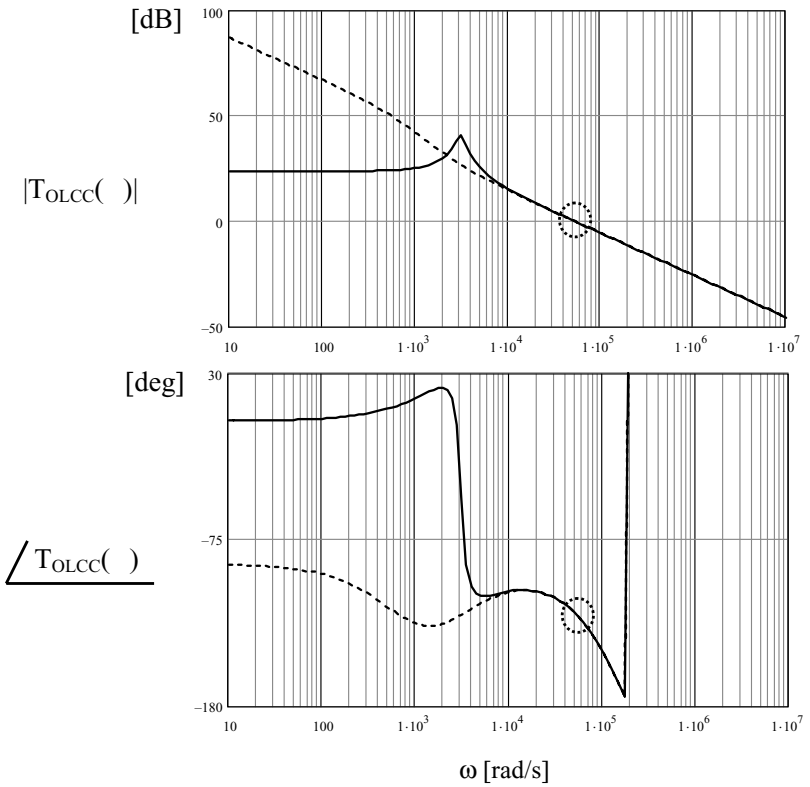

FIGURE 5.4: Bode plot of the current control open loop gain, with (solid line) and without (dashed line) output capacitor. The controller parameters are those calculated in Aside 2.

represented in the following matrix form,

$$\frac{\mathrm{d}}{\mathrm{d}t}x(t) = Ax(t) + B_1\overline{V}_{\mathrm{OC}}(t) + B_2 I_{\mathrm{LOAD}}(t), \tag{5.6}$$

where $x(t) = [\overline{V}_{\mathrm{O}}(t) \quad \overline{I}_{\mathrm{O}}(t)]^T$ is the state vector, average inverter voltage $\overline{V}_{\mathrm{OC}}$ and load current I_{LOAD} are considered system inputs, and

$$A = \begin{bmatrix} 0 & 1/C_{\mathrm{s}} \\ -1/L_{\mathrm{s}} & 0 \end{bmatrix} \quad B_1 = \begin{bmatrix} 0 \\ 1/L_{\mathrm{s}} \end{bmatrix} \quad B_2 = \begin{bmatrix} -1/C_{\mathrm{s}} \\ 0 \end{bmatrix}. \tag{5.7}$$

Assuming, as we have done in Chapter 3, that the inverter voltage $\overline{V}_{\mathrm{OC}}$ and load current I_{LOAD} are constant between sampling instants (zero-order hold equivalence of the system), the discrete time dynamic equations can be written as

$$x(k+1) = \Phi\, x(k) + \Gamma_{\mathrm{V}}\, \overline{V}_{\mathrm{OC}}(k) + \Gamma_{\mathrm{I}}\, I_{\mathrm{LOAD}}(k) \tag{5.8}$$

where

$$\Phi = e^{AT_s} = \begin{bmatrix} \cos(\omega_o T_s) & \frac{1}{\omega_o C_S}\sin(\omega_o T_s) \\ -\frac{1}{\omega_o L_S}\sin(\omega_o T_s) & \cos(\omega_o T_s) \end{bmatrix} \approx \begin{bmatrix} 1 & \frac{T_s}{C_S} \\ -\frac{T_s}{L_S} & 1 \end{bmatrix}, \tag{5.9a}$$

$$\Gamma_V = \left(e^{AT_s} - I_2\right) A^{-1} B_1 = \begin{bmatrix} 1 - \cos(\omega_o T_s) \\ \frac{1}{\omega_o L_S}\sin(\omega_o T_s) \end{bmatrix} \approx \begin{bmatrix} 0 \\ \frac{T_s}{L_S} \end{bmatrix}, \tag{5.9b}$$

$$\Gamma_I = \left(e^{AT_s} - I_2\right) A^{-1} B_2 = \begin{bmatrix} -\frac{1}{\omega_o C_S}\sin(\omega_o T_s) \\ 1 - \cos(\omega_o T_s) \end{bmatrix} \approx \begin{bmatrix} -\frac{T_s}{C_S} \\ 0 \end{bmatrix}. \tag{5.9c}$$

In (5.9), I_2 is the 2×2 identity matrix, T_S is the sampling period, and ω_o is the angular resonance frequency of the second-order L–C filter. Under the assumption that the sampling frequency is much greater than the resonance frequency of the L–C filter (i.e., $\omega_o \cdot T_S \ll 1$), the approximations shown in (5.9*a*)–(5.9*c*) hold. Now, if we consider the second row of each matrix, we can immediately recognize that the current state equation implied by (5.9) is as follows,

$$\overline{I}_O(k+1) = \overline{I}_O(k) + \frac{T_S}{L_S} \cdot \left[\overline{V}_{OC}(k) - V_O(k)\right], \tag{5.10}$$

which, once E_S is substituted by V_O, is exactly coincident with (3.19). Once again, the predictive controller we can design around (5.10) is exactly the same as we have designed around (3.19).

In summary, thanks to the property of the considered topology that guarantees $\omega_o \cdot T_S \ll 1$, all we have mentioned in Chapters 2 and 3 is still valid and can be directly applied to the present case. Therefore, the design of the PI voltage controller can be developed assuming that one of the solutions discussed in Chapters 2 and 3 is used in the current loop.

As an example, we will now discuss the case where the current controller is a dead-beat one. Of course, the same method that we are now going to present can be applied in case a PI or another kind of controller is considered for the current loop.

We know, from Chapter 3, that the dead-beat current controller is dynamically equivalent to a two modulation period delay. The static gain can be, without loss of generality and from the voltage loop controller's standpoint, assumed to be equal to the inverse of current transducer gain. Recalling the discussion of Section 5.1 and in particular (5.4) and Table 5.1, we can consider the transfer function for the closed loop current controller to be equal to

$$\frac{I_O}{I_{OREF}}(s) = \frac{1}{G_{TI}} \frac{1 - s\,T_S}{1 + s\,T_S}, \tag{5.11}$$

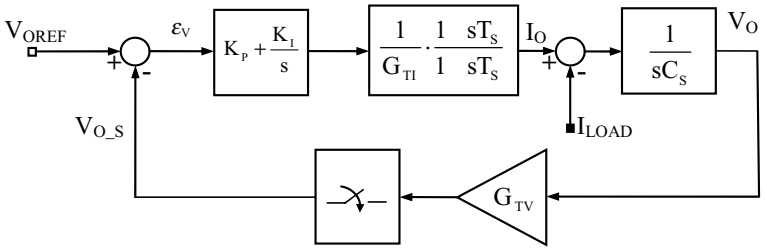

FIGURE 5.5: Block diagram of the voltage loop digital PI controller for the UPS of Fig. 5.3.

while that of the inverter load (Fig. 5.3) can be easily given by

$$\frac{V_O}{I_O}(s) = \frac{1}{s\,C_S}. \tag{5.12}$$

We can now build the block diagram around which the design of the PI voltage controller can be developed. This is shown in Fig. 5.5.

As can be seen, the control problem we are now considering is very similar to that considered in Chapter 2 for the continuous time PI current controller design. An important difference is that the holder delay effect, for the reasons explained above, has not to be considered in this design.

The procedure to solve this problem, determining the PI controller gains K_P and K_I is presented in Aside 9. As can be seen, it closely follows the one we considered in Chapters 2 and 3: first we determine a continuous time voltage PI controller that, later, we turn into a digital one by discretization. The PI voltage controller design is therefore concluded by the calculation of the discrete time equivalent of both gains. As usual, the final step we need to take is the simulation of the complete dynamic system, with current and voltage regulators. An example of the results obtained for our test case is shown in Fig. 5.6.

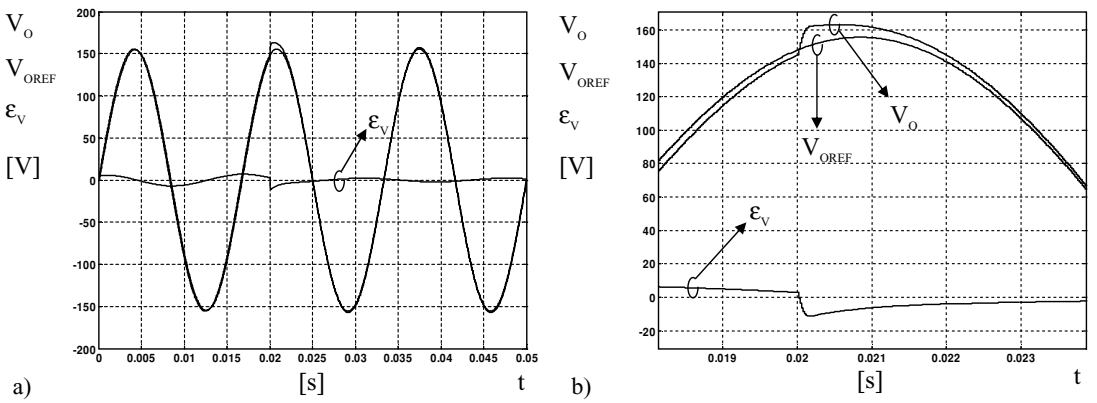

FIGURE 5.6: Dynamic response of the digital PI voltage controller: (a) response to a step load disconnection: the load current instantaneously reduces from about 9.1 A_{RMS} to 0; (b) details of the previous figure.

Aside 9. Example of a PI Voltage Controller Design for a UPS Application

The voltage PI controller gains can be determined once the desired loop bandwidth, f_{CL}, is specified. For a UPS application, in order to achieve a satisfactory control of the voltage waveform in the presence of distorting loads, we can say that, as a rule of thumb, this should be, at least, 15–20 times the line frequency, i.e., from 900 Hz to 1200 Hz in our example.

While this is easy to obtain when the switching frequency is relatively high, as it is in our case, and the current controller is a fast one, like the one we are considering here, in the opposite case, i.e., when a low switching frequency application is considered or when the internal control loop is relatively slow, it may not be too easy to achieve the desired f_{CL} values.

However, once f_{CL} is known, we can consider the open loop gain expression and force its magnitude to be equal to one at the desired crossover frequency. From Fig. 5.5 the open loop gain is found to be

$$G_{OL\_V}(s) = \frac{G_{TV}}{G_{TI}} \frac{1 - s\,T_S}{1 + s\,T_S} \frac{1}{s\,C_S} \left(K_P + \frac{K_I}{s} \right). \qquad (A9.1)$$

It is worth noting that, differently from the current controller case, no delay effect related to the holder has been taken into account. This is possible because the internal current control loop has been designed to compensate for that. Therefore, the only dynamic delay the voltage controller has to compensate is that of the current controller.

Given (A9.1), the first condition we need to satisfy, by suitably choosing K_P and K_I, is as follows,

$$\frac{G_{TV}}{G_{TI}} \frac{\sqrt{K_I^2 + (\omega_{CL} K_P)^2}}{\omega_{CL}^2 C_S} = 1, \qquad (A9.2)$$

where, as usual, $\omega_{CL} = 2\pi f_{CL}$.The second constraint we can impose is requiring a minimum phase margin, ph_m, for the loop gain at the crossover frequency. In order to get a reasonable damping of the dynamic response, this can be set equal to 60°. Consequently, we find the following additional condition:

$$-180^\circ + ph_m = -180^\circ - 2\tan^{-1}(\omega_{CL} T_S) + \tan^{-1}\left(\omega_{CL} \frac{K_P}{K_I} \right). \qquad (A9.3)$$

The solution of the system of equations (A9.2) and (A9.3), considering the parameter values listed in Table 5.1 and imposing $f_{CL} = 1800\,\text{Hz}$, provides us with the following values for the PI gains: $K_P = 3.83$, $K_I = 3.42 \times 10^3$ (rad s^{-1}).

The conversion of the continuous time PI into a discrete time one is simply obtained applying the following relations:

$$\begin{cases} K_{I\_dig} = K_I \cdot T_S \\ K_{P\_dig} = K_P. \end{cases} \qquad (A9.4)$$

Finally, it is worth adding a comment on the calculation delay associated with the voltage controller. Typically, this can be considered equal to zero, because, if the controller hardware has been correctly chosen, the computation of the current reference sample can be done within the same control period where the duty-cycle is updated. In other words, it should always be possible to provide the current controller with the most recent sample of the current reference, without the need to wait for the following modulation period. The minimum requirement is, of course, that the sum of the durations of the voltage controller and current controller algorithms does not exceed one sampling period.

As can be seen, the steady-state reference tracking capabilities of the voltage controller are pretty fair. A steady-state sinusoidal tracking error is recognizable, that, as in the current loop case, is due to the finite amplification the PI controller offers at the reference frequency. This problem can be solved by modifying the controller structure, as will be explained in Section 5.4, or by adding some form of feed-forward compensation, e.g., of the capacitive component of the inverter output current.

To test the voltage PI in dynamic conditions as well, we have considered a typical UPS test case, i.e., step load disconnection. At the instant when the inverter output current is maximum, i.e., the maximum energy is stored in the L_S inductor, the load is disconnected. This causes an immediate output voltage error (negative) that needs to be corrected by the voltage controller. We can therefore evaluate the controller dynamic properties. It is worth noting that neither the current loop nor the voltage loop enters saturation during the test: accordingly, the behavior illustrated by Fig. 5.6 can be considered a consequence of the regulator properties, not influenced by saturation effects or other system nonlinearities. The regulation bandwidth determines the significant voltage error peak at the instant of the load step change. This is recovered in a relatively small fraction of the reference period, with adequate damping, i.e., without ringing or persistent oscillations.

5.3.2 The Predictive Controller

In Chapter 3, we have discussed the dead-beat, predictive current controller. We have seen how this represents the highest performing current controller, determining a dynamic response delay for the current loop that is equal to two modulation periods. It may be quite obvious to ask if, using the same strategy, one could get the same high performance level for the voltage controller as well. The answer is in the affirmative: it is indeed possible to implement a predictive controller for the voltage control loop and get again a very fast dynamic performance. Following this approach, it is possible to set up a multiloop controller based on decoupled current and voltage predictive regulators, whose dynamic response delay turns out to be equal to four modulation periods. This solution, which we identify as the multiloop predictive controller, will be described in the next section.

However, for the sake of completeness, we have to mention that the more direct and well-known application of dead-beat control to the converter structure of Fig. 5.3 does not actually follow the multiloop approach. In this case, the direct pole allocation and dynamic state feedback are applied to the second-order system described by (5.9). A multivariable controller is consequently achieved, whose dynamic response delay is equal to three modulation periods, faster than the previously described one. However, as it almost always happens, the price to pay for the speedup is not negligible. The absence of a current control loop makes the management of some practical operating conditions, such as overload or output short circuit, rather complicated. In the last part of this section, we will discuss the main features of this controller as well.

5.3.2.1 The Multiloop Implementation

The schematic organization of the multiloop predictive controller [5] is shown in Fig. 5.7. As can be seen, the block diagram is complicated by the presence of three functions, i.e., the capacitive current feed-forward (A), the reference current interpolator (B), and the load current estimator (C), which can be considered ancillary. As it will be explained a little further on, the purpose of these blocks is simply to improve the static and dynamic behavior of the regulator, but, for now, we can neglect them and focus on the main controller components.

Doing that, it is immediate to recognize in Fig. 5.7 the same basic organization of a multiloop controller shown in Fig. 5.3. Of course, Fig. 5.7 is based on discrete time representations of both controllers and the converter load. Because of that, no sampling block is explicitly represented in the figure. For the same reason, the load transfer function is represented as $Z_{CS}(z)$, which stands for the discrete time version of (5.12).

Considering now the current controller, we will just say that this is designed exactly following the procedure we described in Chapter 3. After the discussion of Section 5.3.1, we know that this is perfectly applicable to the present case, as the derivation of (5.10) clearly shows.

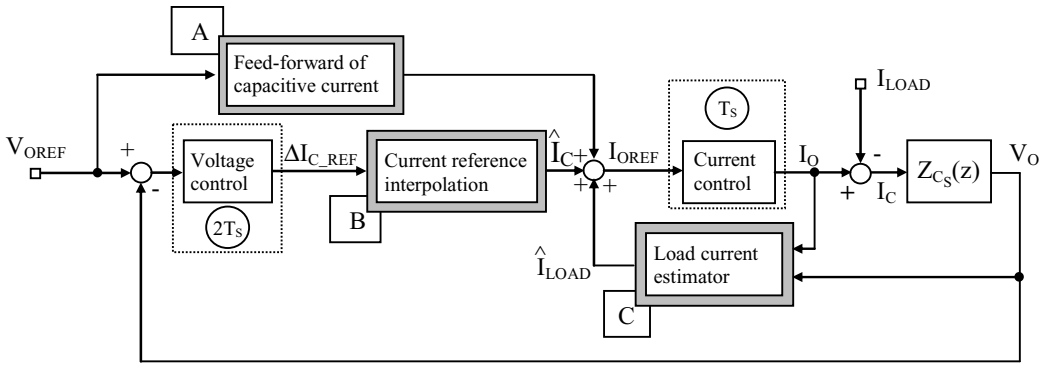

FIGURE 5.7: Schematic organization of the multiloop predictive voltage controller.

The voltage controller can be designed almost identically, considering the first row of the state space description (5.9). This corresponds to the following state equation,

$$\overline{V}_O(h+1) = \overline{V}_O(h) + \frac{T_{S\_V}}{C_S} \cdot \left[\overline{I}_O(h) - I_{LOAD}(h)\right], \tag{5.13}$$

which, as can be seen, presents exactly the same structure of (5.10). Please note that, in order to keep the notation simple and clear, we denoted the considered sampling instant as $h \cdot T_{S\_V}$, to highlight that the sampling process for the voltage loop can be operated, in general, with a different sampling period, T_{S_V}, with respect to that of the current loop, T_S. Following the same reasoning presented in Chapter 3, we can now find the dead-beat control equation for the voltage loop. Once again, this presents exactly the same structure as that derived for the current loop, i.e.,

$$I_{OREF}(h+1) = -I_{OREF}(h) + \frac{C_S}{T_{S\_V}} \cdot \left[V_{OREF}(h) - \overline{V}_O(h)\right] + 2 \cdot I_{LOAD}(h), \tag{5.14}$$

where the load current is assumed to be a relatively slowly varying signal and, consequently, the approximation $I_{LOAD}(h+1) \cong I_{LOAD}(h)$ is considered.

It is essential to underline that the derivation of (5.14) actually hides a very important assumption; that it is possible and correct to *identify* the current reference signal with the actual inverter output current by the end of *every given* control period. This assumption is not always correct: in particular, it is surely *not* correct if the sampling process for the voltage loop and that for the current loop have the same period duration. In this case, the dynamic delay of the current loop, which requires two periods to make the output current equal to its reference, undermines the system stability. In contrast, if the sampling frequency for the voltage loop is set equal to one-half of that used for the current loop, the delay, from the voltage loop standpoint, becomes ineffective and the identification of the current reference with the actual inverter output

current is correct. Therefore, the controller organization of Fig. 5.7 actually requires $T_{S\_V} = 2T_S$. Because of that, the dynamic response delay of the voltage controller, which will be equal to two control cycles, as it was for the current controller, is actually equivalent to four modulation periods.

Several refinements are possible to improve the controller operation with respect to what can be achieved simply by programming (5.14) as the voltage loop control equation. In the first place, it is possible to feed-forward every known component of the inverter output current, like the current in the output capacitor C_S, that is easily precomputed from the voltage reference signal, once the output capacitor value is known. This is exactly what block A of Fig. 5.7 does. The output current has another component, i.e., the load current, that, in general, cannot be precalculated and, therefore, should be measured. Nevertheless, a simple estimation equation can be implemented, exactly as it was done for the current controller, in order to avoid this measurement, which can be sometimes problematic. The basic estimation equation is as follows,

$$\hat{I}_{LOAD}(k-1) = -\frac{C_S}{T_S} \cdot [\overline{V}_O(k) - \overline{V}_O(k-1)] + \overline{I}_O(k-1) \qquad (5.15)$$

which can be actually improved by adding a cascaded low-pass filter, so as to remove possible instabilities or measurement noise. The implementation of (5.15) and of the low-pass filter is essentially the function of block C in Fig. 5.7.

Once, thanks to blocks A and C, the capacitive and load currents are obtained, the function of the voltage controller is only to compensate for the residual feed-forward and estimation errors. Of course, the voltage control equation (5.14) can be rewritten accordingly, obtaining

$$\Delta I_C(h) = \frac{C_S}{2 \cdot T_s} \cdot [V_{OREF}(h) - \overline{V}_O(h)] - \Delta I_C(h-1), \qquad (5.16)$$

which explains why, in Fig. 5.7, the output of the current controller is not I_{O_REF}, but the quantity ΔI_C.

The function of block B is a little more complicated to explain. We have seen before that the voltage control equation is computed at half the frequency of the current control. This means that the current controller reacts to the reference signal generated by the voltage controller as to a stepwise function, updated every two modulation periods. This determines persistent high-frequency oscillations in the inverter output current. In order to eliminate this effect, the interpolator block B of Fig. 5.7 generates an extra reference signal sample to feed the current controller in those control periods when the voltage loop would not update the reference. This makes the reference signal for the current controller practically equivalent to a continuous time signal, correctly sampled with T_S period, and thus eliminates the step response dynamics from the output current.

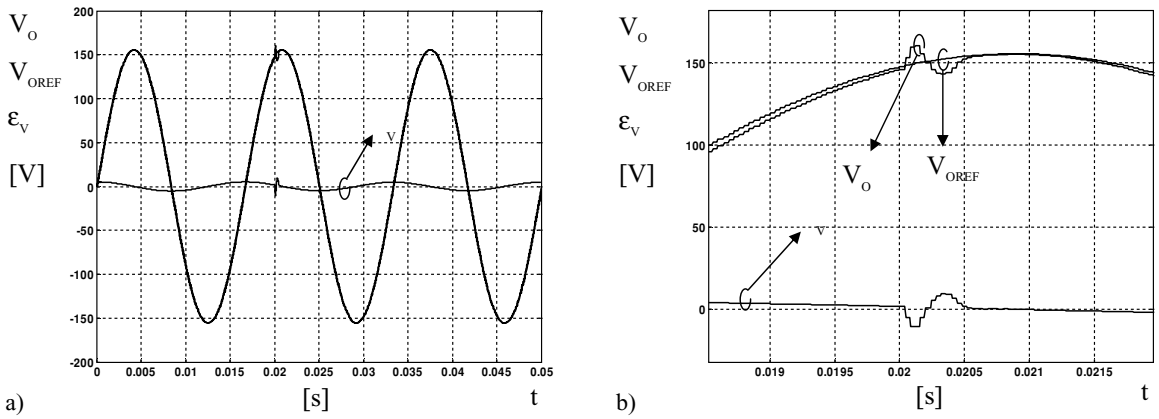

FIGURE 5.8: Dynamic response of the digital dead-beat voltage controller: (a) response to load step disconnection; (b) details of the previous figure.

The provisions we have briefly outlined make the UPS controller of Fig. 5.7 quite effective. We can see the typical performance achievable with this controller in Fig. 5.8. It is interesting to compare Figs. 5.8 and 5.6, since they were obtained for the very same test conditions.

As can be seen, there is a significant difference in the two controllers' performance. Firstly, the dead-beat voltage controller offers a smaller residual steady-state tracking error at the fundamental frequency, which turns out to be about 50% smaller than that achieved by the PI. Additionally, its dynamic performance is much faster, as is clearly visible if one compares the error trajectory after the load transient. This readiness guarantees both a smaller voltage overshoot and a faster recovery of the nominal voltage trajectory.

5.3.2.2 The Multivariable Implementation

The dead-beat controller is more often implemented as in [6–8], i.e., by applying state feedback theory to the second-order dynamic system represented by (5.9). The approach practically replicates the one we have followed in Chapter 3, Aside 5, with the remarkable difference that we are now dealing with a two-component state vector. We can describe the solution considering, at first, the simpler and ideal case where the computation delay is neglected. Accordingly, the basic static state feedback implementation is as follows,

$$x(k+1) = \Phi\, x(k) + \Gamma_V K x(k) = \Phi_F x(k) \qquad (5.17)$$

where the system input $\overline{V}_{OC}$ has been replaced by $K \cdot x$, and $K = \left[K_{\overline{V}_O}\ K_{\overline{I}_O}\right]$ is the feedback gain matrix. Consequently, the closed loop system is now characterized by a new state matrix $\Phi_F = \Phi + \Gamma_V K$, whose eigenvalues can be properly allocated by suitably choosing the gains

$K_{\overline{V}_O}$ and $K_{\overline{I}_O}$. The computation is a little involved, but it is possible to see that the following values,

$$k_{\overline{V}_O} = \frac{1 - 2\cos(\omega_o T_s)}{2 - 2\cos(\omega_o T_s)} \quad k_{\overline{I}_O} = -\omega_o L_S \frac{1 + 2\cos(\omega_o T_s)}{2\sin(\omega_o T_s)} \tag{5.18}$$

achieve the desired results; i.e., both the closed loop system eigenvalues are relocated in the origin of the complex plane. It is interesting to note that Eq. (5.18) is given for the original system matrixes, i.e., without any approximation. It has therefore general validity. More subtly, if we tried to operate the closed loop compensation of the system from the $\overline{V}_{OC}$ input, *after* the system is dynamically *decoupled* and the approximated system matrixes are obtained, we would encounter a serious problem: the approximated dynamic system is no longer state *controllable* from the $\overline{V}_{OC}$ input. This reflects the physical fact that, in the hypothesis of a decoupled system, the output voltage $\overline{V}_O$ no longer depends on the average inverter voltage, but only on the average inverter current.

Therefore, the approach we are discussing here is only meaningful if we do not take into account the dynamic decoupling hypothesis. Please note that this could be the only correct way of synthesizing a dead-beat controller in all those cases where the second-order output filter does not guarantee that the condition $\omega_o \cdot T_S \ll 1$ is satisfied.

In conclusion, the organization of a state feedback loop with gains given by (5.18) guarantees a dead-beat response for the closed loop system. Unfortunately, the practical implementation of this solution is not possible, because of the computation delay, which we have not taken into account. In order to do that, we need to follow again the same approach of Aside 5, i.e., considering a dynamic state feedback implementation. The details of the procedure are given in Aside 10.

Before we conclude our presentation of the multivariable dead-beat controller, we would like to discuss the results of its numerical simulation, shown in Fig. 5.9. In particular, we would like to compare Figs. 5.9 and 5.8.

As can be seen, there is no dramatic performance improvement in the considered implementation. This is because although, in principle, the multivariable dead-beat controller is capable of a three modulation period response delay, i.e., the fastest theoretically possible dynamic response, the need for the reference signal reconstruction, as explained in Aside 10, partially cancels this advantage. Therefore, the achieved dynamic performance is practically comparable to that offered by the multiloop dead-beat implementation. However, we need to highlight, once again, that the multivariable implementation is actually the only possible solution for dead-beat control of second-order output filters with a relatively high resonance frequency.

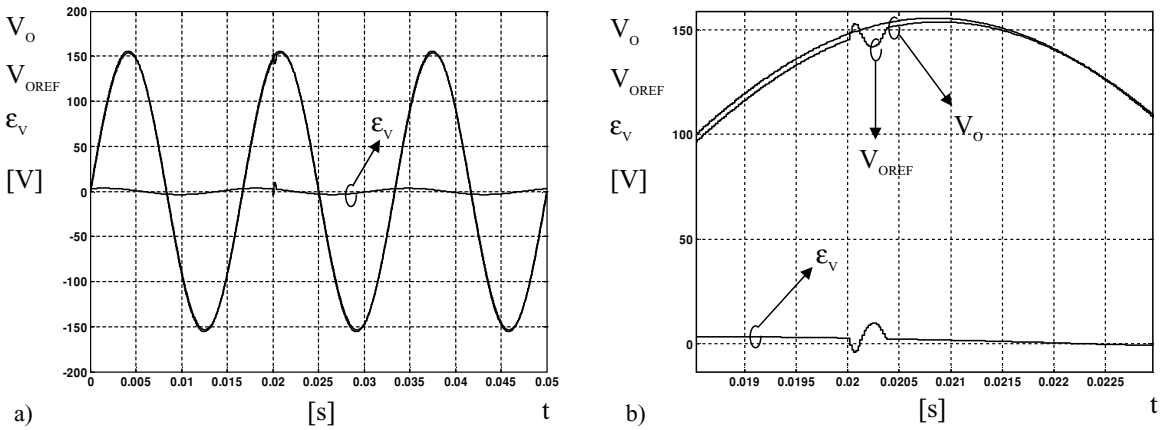

FIGURE 5.9: Dynamic response of the digital dead-beat voltage controller: (a) response to a load step disconnection; (b) details of the previous figure.

Aside 10

We consider the discrete time equivalent model for the UPS system given by (5.9), which we recall here for clearness, i.e.,

$$x(k+1) = \Phi x(k) + \Gamma_V \overline{V}_{OC}(k) + \Gamma_I I_{LOAD}(k), \qquad (A10.1)$$

where $x(k) = \left[\overline{V}_O(k)\ \overline{I}_O(k)\right]^T$ and

$$\Phi = \begin{bmatrix} \cos(\omega_0 T_s) & \frac{1}{\omega_0 C_S}\sin(\omega_0 T_s) \\ -\frac{1}{\omega_0 L_S}\sin(\omega_0 T_s) & \cos(\omega_0 T_s) \end{bmatrix}, \quad \Gamma_V = \begin{bmatrix} 1-\cos(\omega_0 T_s) \\ \frac{1}{\omega_0 L_S}\sin(\omega_0 T_s) \end{bmatrix},$$

$$\Gamma_I = \begin{bmatrix} -\frac{1}{\omega_0 C_S}\sin(\omega_0 T_s) \\ 1-\cos(\omega_0 T_s) \end{bmatrix}. \qquad (A10.2)$$

What we want is to build a dynamic state feedback controller around variable $\overline{V}_{OC}$, which can be represented by the following equation,

$$\overline{V}_{OC}(k+1) = \begin{bmatrix} K_1 & K_2 \end{bmatrix}(x_{REF}(k) - x(k)) + K_3 \overline{V}_{OC}(k), \qquad (A10.3)$$

where gains K_1, K_2, and K_3 have to be determined and $x_{REF}(k) = \left[V_{OREF}(k)\ I_{OREF}(k)\right]^T$

is the state reference trajectory. We can now determine the augmented state matrix that corresponds to the new dynamic system, made up by (A10.1) and (A10.3). It is immediate to find that this is given by

$$\Phi_A = \begin{bmatrix} \Phi_{11} & \Phi_{12} & \Gamma_{V11} \\ \Phi_{21} & \Phi_{22} & \Gamma_{V21} \\ -K_1 & -K_2 & K_3 \end{bmatrix}. \qquad (A10.4)$$

As we did in Aside 4, we now need to calculate the K_1, K_2, and K_3 gain values, so as to force the eigenvalues of matrix Φ_A to move to the origin of the complex plane. Once again, this very simple idea requires some mathematics; however, after that, it is possible to find that the following values

$$\begin{aligned} K_1 &= -\frac{1 + 2\cos(\omega_0 T_S) - 4\cos^2(\omega_0 T_S)}{2[1 - \cos(\omega_0 T_S)]}, \\ K_2 &= -\frac{\omega_0 L_S}{2\sin(\omega_0 T_S)}\left[1 - 2\cos(\omega_0 T_S) - 4\cos^2(\omega_0 T_S)\right], \\ K_3 &= -2\cos(\omega_0 T_S), \end{aligned} \qquad (A10.5)$$

solve the problem. Therefore, substituting (A10.5) gains into (A10.3) control equation, we get the desired multivariable dead-beat controller.

It is important to underline that, differently from the multiloop implementation, in the multivariable dead-beat controller, the computation of the current reference trajectory is not automatic, i.e., inherent in the controller structure. This means that we have to explicitly determine the reference current from the voltage reference trajectory, that is, of course given, and from other system variables. The standard procedure is to precompute the capacitive current component of the output current from the voltage reference and either measure or estimate the load current. Estimation techniques, e.g., based on *disturbance observers* [9, 10], can be implemented that allow one to save the load current measurement. However, in that case, the observer dynamics are responsible for a certain increase in the response delay of the controller.

We conclude this brief discussion of dead-beat voltage controllers observing that, in recent times, a significant research effort has been focused on this control technique. Therefore, several technical papers can be found where this subject is treated in detail and possible refinements or different implementation strategies are presented. The interested reader may take advantage of references [6–10] as far as the multivariable implementation is concerned. Instead, additional details on the multiloop dead-beat controller implementation can be found in [5].

5.4 NARROW BANDWIDTH CONTROLLERS

In this section we present a summary of two very popular *narrow bandwidth* voltage control strategies, frequently employed in UPS systems. These are the repetitive-based voltage controller and, once again, the rotating reference frame voltage controller. The former is based a totally new concept we never encountered before, and the latter, instead, is almost the direct extension of what we have discussed in Chapter 4 for the current loop implementation. Essentially for this reason, we will here discuss a different implementation strategy for the same concept, which is based on DFT (discrete Fourier transform) filters.

5.4.1 The Repetitive-Based Voltage Controller

The concept of repetitive control originates from the internal model control principle. For obvious reasons, we will not present here any of the numerous theoretical issues related to internal model control and, in particular, to the derivation, under general assumptions, of repetitive controllers. The interested reader can find a very good treatment of these topics in specialized textbooks like, for example, [11]. Instead, we would like to open our discussion simply by describing the goal of any repetitive controller, which is to make the controlled system output track a set of predefined reference inputs, without steady-state error. The theory shows that, in general, the achievement of this result requires the stabilization of an augmented system, where the dynamic representation, in terms of Laplace or Z-transform, of the reference signal of interest, has been somehow added to the original system model. This can be, in some cases, a quite complicated control problem.

However, in the particular case of sinusoidal reference signals, which represent exactly what we are interested in, for the UPS output voltage control, the digital implementation of a repetitive controller becomes relatively simple, requiring only the setup of a suitably sized delay line and of a positive feedback loop [12, 13].

An example of the basic structure of a repetitive controller, organized for application to the UPS external voltage loop, is shown in Fig. 5.10(a). According to the required control function, the error on the UPS output voltage, ε_V, represents the controller input, while the controller output is represented by the current reference signal for the internal current loop.

It may not be obvious to see why, once the closed loop system is stabilized, the configuration of Fig. 5.10(a) necessarily implies zero reference tracking error with respect to sinusoidal signals. The formal way to realize why and how this happens consists in computing the transfer function that relates the controller input to the output and plot the frequency response. What can then be found is a very interesting result: the controller transfer function presents infinite gain at all frequencies that are integer multiples of a fundamental one. The fundamental controller frequency is the one associated with the delay line duration. Therefore, if the delay

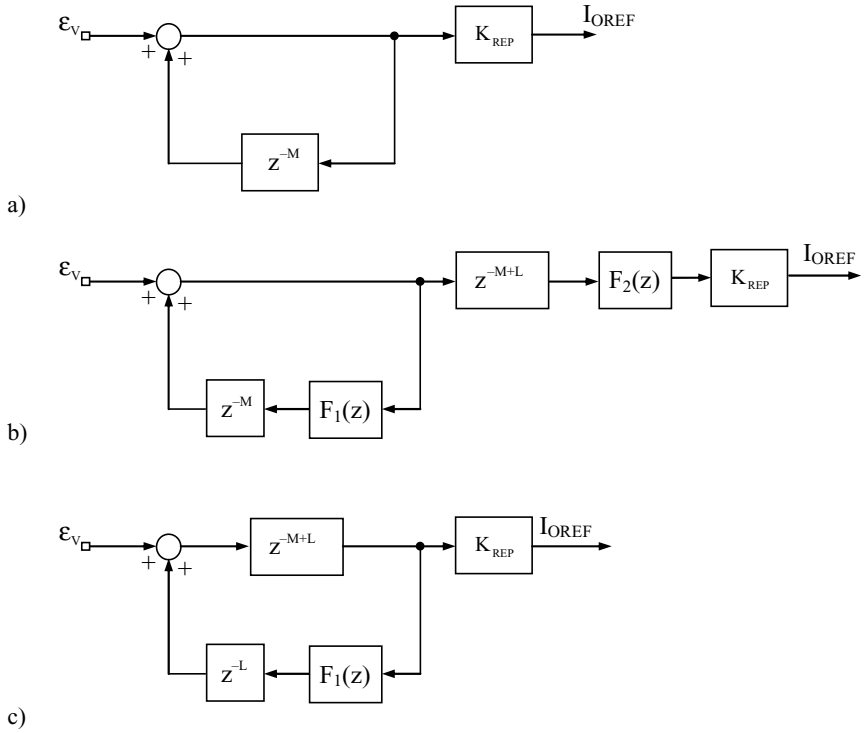

FIGURE 5.10: (a) General implementation of the repetitive controller; (b) provisions to improve the stability margin; (c) the considered implementation.

line duration is made equal to the desired output voltage frequency, the frequency response of the repetitive controller will be approximately equivalent to the parallel connection of a bank of resonant filters, each presenting infinite gain at one integer multiple of the output voltage frequency.

As a matter of fact, this result can also be anticipated simply by referring to Fig. 5.10 (a) and considering the delay line operation. Any signal that repeats itself exactly in the delay line period gets infinite amplification. Therefore, all sinusoidal signals whose period is an integer submultiple of the delay line period, $M \cdot T_S$, get infinite amplification. One way or the other, we see that the controller structure of Fig. 5.10(a) is a practical means to boost to infinity the open loop system gain at every harmonic up to the Nyquist frequency. From this it necessarily derives a zero steady- state tracking error on the output voltage sinusoidal signal and on all of its harmonics.

However, exactly for the same reason, this structure poses serious stability problems for the system. Indeed, the infinite amplification of the highest order harmonic components of the voltage error can reduce the control loop phase margin and undermine the controller stability.

The basic reason is that, as we know, the internal current controller has a limited bandwidth. Therefore, in order not to incur instability, the frequency content of the current reference signal has to be limited accordingly.

Because of this, several additional provisions have been proposed for an effective practical implementation of the repetitive controller. For example, in order to guarantee system stability, some filters can be introduced in the scheme of Fig. 5.10(a), in the feedback path, $F_1(z)$, or in a cascade connection with the repetitive controller, $F_2(z)$, or even both, as shown in Fig. 5.10(b). The goal of these filters is exactly to limit the amplification of the high-order harmonics. In addition, the stability of the repetitive controller has been shown to greatly improve if a delay line of $M - L$ samples is inserted at the output of the regulator. This is actually equivalent to adding a phase lead of L samples for all the harmonic frequencies and has been shown [12] not to change the gain at the harmonic frequencies, but just to increase the system phase margin.

In conclusion, the repetitive controller organization we are going to discuss, that sums up all these considerations, is shown in Fig. 5.10(c), which is, of course, theoretically equivalent to the scheme of Fig. 5.10(b) when $F_2(z) = 1$.

In recent times, a lot of different voltage loop controllers built around the repetitive controller structure of Fig. 5.10(c) have been proposed and applied. The different solutions try to solve the typical problems that are often encountered in the practical application of repetitive controllers. In particular, experience shows that it is normally quite difficult to achieve simultaneously a satisfactory steady-state voltage error compensation and an acceptable large signal behavior from the repetitive controller in a stand-alone configuration. Stability can be obtained, but due to the effects on the control loop phase of the high-frequency resonances in the controller frequency response the phase margin is typically low, with a consequent unsatisfactory performance during transients.

For this reasons, the repetitive controller is more typically employed in parallel connection with a conventional regulator. In the scheme of Fig. 5.11, we can see a simple implementation of this principle: a purely proportional controller is paralleled to the repetitive one. The motivation for the considered controller's organization is to have, in the steady state, the proportional controller action joined by the repetitive controller's one: the latter compensates the periodic error components the former, because of its limited bandwidth, cannot eliminate, thus making the residual tracking error practically equal to zero. In addition, as we will see, the solution allows the designer to better control the loop phase margin. Therefore, it is generally possible to guarantee a conveniently damped response to perturbations.

Seen from this standpoint, the repetitive controller can be considered as an optional function we can employ in parallel to a conventional controller anytime we need to improve its steady-state performance. In the presence of periodic output voltage disturbances, like those induced by nonlinear loads connected to the UPS output, this solution can greatly improve

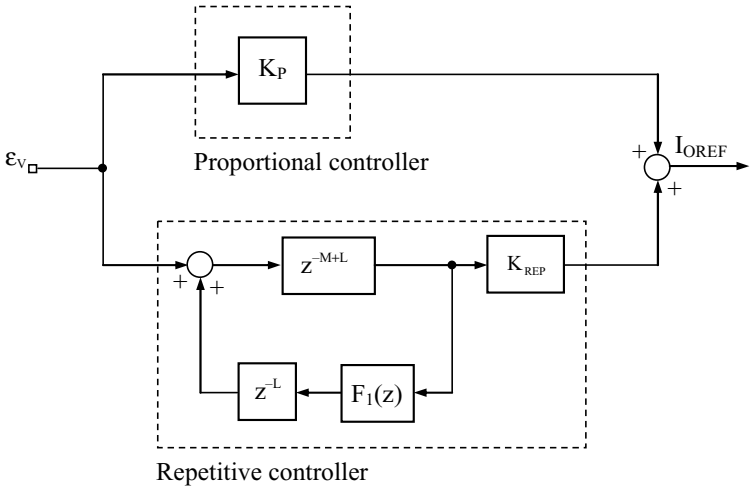

FIGURE 5.11: (a) Suggested repetitive-based voltage controller. The repetitive controller structure of Fig. 5.10(c) is connected parallel to a conventional purely proportional controller.

the quality of the output voltage regulation. Of course, nothing can be gained from this controller organization in the compensation of fast transients, like those determined by step load variations.

The design of the parallel structure of Fig. 5.11 can be performed in two separate steps: (i) design of the proportional regulator and (ii) design of the repetitive controller. The first step is very similar to the standard PI design we have already described in Section 5.3.1 and Aside 9, so we will not comment further on that. As far as the second step is concerned, we basically need to determine (i) the value of parameter M, (ii) the value of parameter L, (iii) the value of gain K_{REP}, and (iv) the structure of $F_1(z)$.

The design of parameter M simply requires the determination of the ratio of the sampling frequency and the fundamental output voltage frequency. Since M must be integer, this may generally require the adjustment of the switching frequency to an integer multiple of the output voltage fundamental. In our test case, the switching and sampling frequencies were adjusted to 48kHz, thus giving $M = 800$.

The design of the other parameters requires a careful consideration of the open loop gain, and in particular of the system phase margin. In order to compute the loop gain, we can refer to the block diagram of Fig. 5.12, where, once again, the basic organization of Fig. 5.5 can be identified, with the important difference that all blocks are now discrete time and, consequently, the ideal sampler block is no longer represented.

As described above, the repetitive-based controller is given by the parallel connection of the purely proportional regulator and the repetitive controller of Fig. 5.10(c), whose transfer

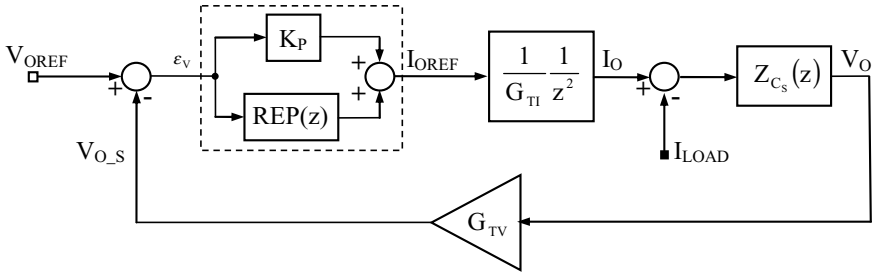

FIGURE 5.12: Repetitive-based voltage control loop. The scheme is used for the computation of the open loop system gain.

function can be easily found to be equal to

$$\mathrm{REP}(z) = K_{\mathrm{REP}} \frac{z^{-M+L}}{1 - z^{-M} F_1(z)}. \tag{5.19}$$

In addition to this, Fig. 5.12 includes the current loop transfer function that, supposed to be of dead-beat type, is given by the usual static gain and an ideal two period delay transfer function. Finally, as we already did, we indicate by $Z_{C_S}(z)$ the discrete time version of (5.12), obtained by any discretization method. Based on this scheme, we can now compute the open loop gain and suitably select the repetitive controller parameters so as to maintain the system phase margin and crossover frequency unaffected, while achieving a significant gain boost at least for the first output voltage harmonic frequencies.

The open loop gain is plotted in Fig. 5.13. As can be seen, with the chosen parameters, the open loop gain of the repetitive-based controller is asymptotically equal to that of the purely proportional one. The repetitive controller contribution on the magnitude is represented by the gain peaks, located at integer multiples of the output voltage fundamental frequency and by the small increase of the equivalent proportional gain that appears as an offset between the two plots. The amplitude of the peaks has been limited in high frequency by using, as $F_1(z)$, a moving average filter with 31 taps. This, together with a suitable choice of parameter K_{REP}, which in our example has been set equal to 2, has allowed us to achieve a phase margin at the crossover frequency that is practically identical to that of the purely proportional controller, thus avoiding any stability problem. In addition, no phase lead action was needed in the example we are considering here, since the sampling frequency is relatively high with respect to the crossover frequency. Finally, the effect on the loop phase determined by the moving average filter has been compensated by reducing the number of taps in the delay line by 15. This provision is required because the 31 tap moving average filter actually gives a contribution to the loop phase that is equal to that of a 15 tap delay line. Therefore, the length of the delay line has to be reduced accordingly, so as to keep the total phase lag of the feedback signal path to the correct value. If

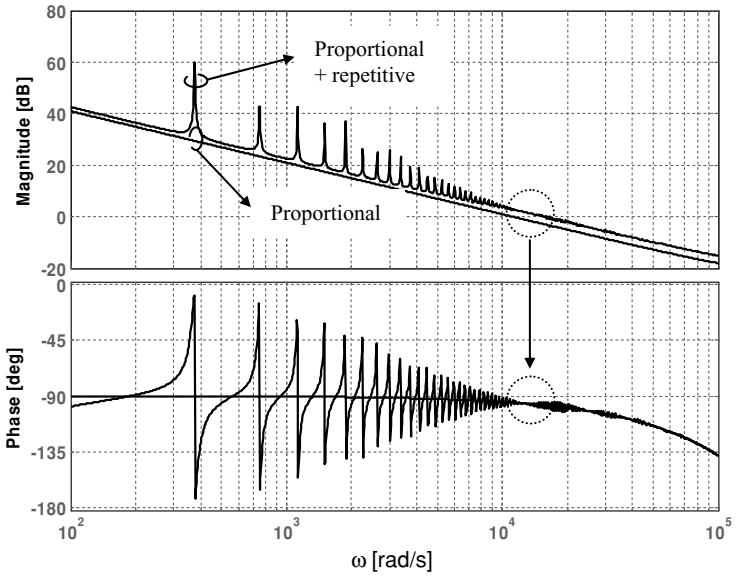

FIGURE 5.13: Open loop system gain for the repetitive-based controller.

this is not done, the frequency allocation of the resonant peaks could be affected and so could be the effectiveness of the regulator.

One could point out that the computational effort required for the implementation of this regulator is relatively high, typically calling for not a negligible amount of hardware resources. We have seen that in our example a 800 tap delay line is *theoretically* required, which implies a significant amount of memory. This limitation can actually be partially overcome by using a M_c sample decimation factor, thus reducing the number of taps the delay line requires. In the example reported hereafter, $M_c = 10$ and consequently the number of delay line taps M has been reduced to 80, i.e., to 79 to take the moving average filter into account. Indeed, the moving average filter $F_1(z)$ has been reduced to only 3 taps. Using this decimation factor the dynamic performance was not affected significantly. One issue related to the adoption of sample decimation is that the output of the repetitive control is updated only every M_c samples and is seen by the proportional controller as a stepwise function. Thus, an interpolator (first-order hold, low-pass filter, etc.) can be useful for the generation of a continuous waveform, especially for higher M_c values. Indeed, the decimation rate can be even higher than what we have considered, since its limit is, theoretically, only represented by the Nyquist frequency for the highest order harmonic one wants to compensate. Of course, practical issues related to system stabilization, i.e., its sensitivity to phase lag effects in the vicinity of the crossover frequency, actually compel us to keep the decimation factor well below this theoretical limit.

The operation of the repetitive-based controller has been simulated with the UPS model already considered for testing the large bandwidth controllers. In order to better highlight the merits of this solution we have considered a typical situation where a distorting load, represented by a high crest factor diode rectifier with capacitive filter, is connected at the UPS output. Because of the nonzero output impedance of the UPS, the load current peaks determine a typical distortion of the output voltage waveform. The repetitive controller is able to slowly compensate for this distortion, reducing it to a minimum in a relatively large number of fundamental frequency periods. This is basically the situation depicted by Fig. 5.14. The figure was obtained by applying, at first, only the proportional controller. The corresponding voltage distortion is shown in Fig. 5.14(b). After a few fundamental frequency periods, at instant $t = 0.1$ s, the repetitive-based controller is activated. Its operation generates a transient that extends through several fundamental frequency periods. This is due to the fact that as the controller reduces the voltage distortion, the crest factor of the load current progressively increases. This typical regenerative effect, which is common to all uncontrolled rectifiers with capacitive filter, is described by the right column of Fig. 5.14, where the inverter output current and its reference are represented. In particular, comparing Fig. 5.14(c) with Fig. 5.14(e) and Fig. 5.14(d) with Fig. 5.14(f), it is possible to realize how the voltage waveform is corrected by the controller, and to appreciate the effect this causes on the load current. In the end, a new steady state is reached, where the voltage distortion is strongly attenuated, even if the load current crest factor has significantly increased.

As Fig. 5.14 clearly demonstrates, the performance of the repetitive-based controller can be quite satisfactory. Nevertheless, some caution is required in the implementation of this type of controller. Indeed, the settling time of the output voltage is in the range of about 10 fundamental frequency periods. It is generally quite difficult to improve this significantly. This implies that, if more frequent load variations can be expected for the considered application, the controller effectiveness is likely to vanish, as it would be operating permanently in transient conditions.

5.4.2 The DFT Filter Based Voltage Controller

A different interpretation of the repetitive control concept, which tends to improve some of its drawbacks while retaining the main positive features, is represented by what we call the *DFT filter based* selective harmonic compensation strategy [14]. We are again referring to a narrow bandwidth controller, whose dynamic response extends itself over several fundamental frequency periods. As the repetitive-based controller, the DFT filter based controller is also conceived to operate in parallel with a conventional voltage regulator and to boost the loop gain only at certain predefined frequencies of interest, which are normally some selected harmonics of the fundamental frequency. This concept is also closely related to that of the rotating reference

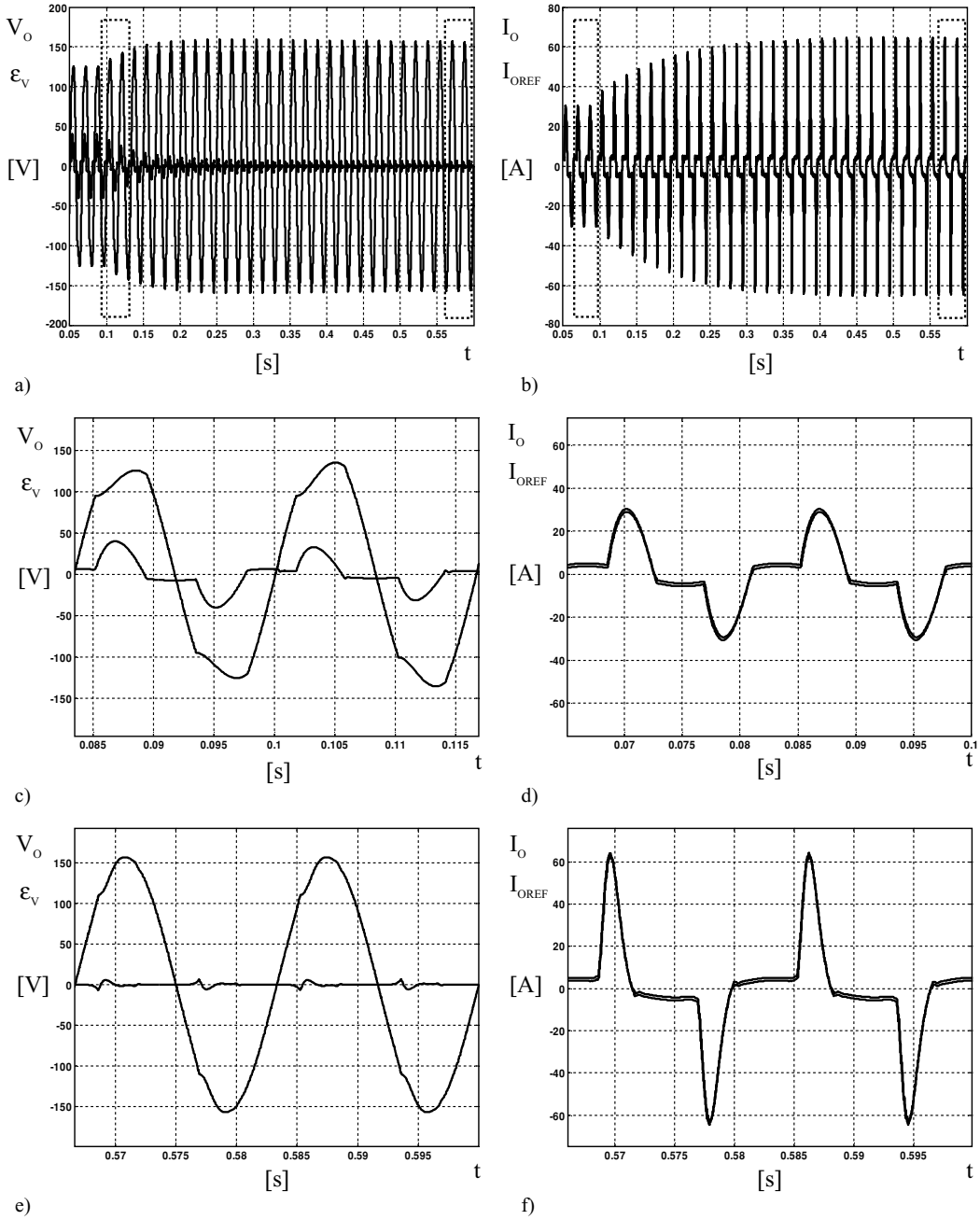

FIGURE 5.14: Repetitive-based controller operation. (a) output voltage transient; (b) output current transient; (c) details of (a) before the repetitive controller is activated; (d) details of (b) before the repetitive controller is activated; (e) details of (a) after the steady state is reached with the repetitive controller; (f) details of (b) after the steady state is reached with the repetitive controller.

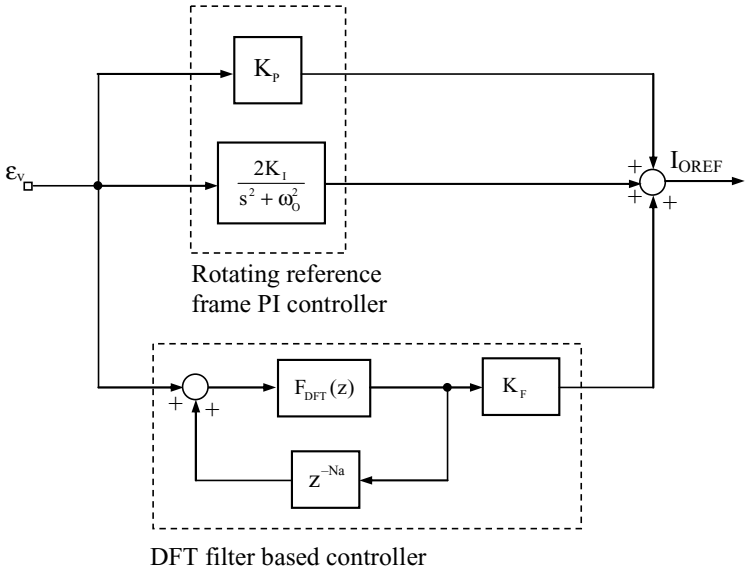

FIGURE 5.15: (a) Suggested DFT filter based voltage controller. A rotating reference frame PI controller is parallel connected to the DFT filter based controller.

controllers considered in Chapter 4. Actually, the DFT filter based controller can be considered an effective way to implement the same control strategy on multiple frequencies.

We have seen how the repetitive-based controller requires that the designer implement some filtering in the delay line to control the system phase in the vicinity of the crossover frequency. The choice of the filter and the control of its interaction with the delay line are the most difficult aspects of the repetitive controller design one has to tackle. The DFT filter based approach tends to mitigate this problem.

The proposed controller organization can be seen in Fig. 5.15, where two parallel components the controller can be identified. The first is a rotating reference frame PI controller, which, as explained in Chapter 4, is fully equivalent to the structure of Fig. 5.15 where a resonant filter centred on the output voltage fundamental frequency is substituted to the integral part of the original PI controller. Please note that this equivalence holds even if the original system is single phase, since the rotating reference frame can be as well used to represent single-phase quantities [14]. From the implementation standpoint however, once the equivalence is exploited and the block diagram of Fig. 5.15 is derived, this interpretation of the rotating reference frame is no longer relevant. The rotating PI controller will guarantee zero steady-state tracking error on the fundamental component of the output voltage.

The second component of the considered voltage controller is designed to take care of high-order harmonics. As in the repetitive-based case, its function is to boost the system open

loop gain at certain predefined frequencies. To achieve this result, once again a positive feedback arrangement is considered. Of course, at any frequency where the gain of the $F_{\mathrm{DFT}}(z)$ filter is unity and its phase is zero, the positive feedback will boost the controller gain to infinity. The nice thing about this controller is that by properly choosing the $F_{\mathrm{DFT}}(z)$ filter, it is possible to have gain amplification *only* where it is actually needed, i.e., at predefined, selected harmonic frequencies, not at *each* harmonic frequency, as it happened for the repetitive-based solution. Please note that this allows us to save the smoothing filter $F_1(z)$, whose design is typically quite complicated, and which was absolutely necessary for the repetitive-based controller.

To achieve the above-mentioned selective compensation and to get an adjustable phase lead, which may be required to ensure a suitable phase margin at the crossover frequency, we propose the use of "moving" or "running" DFT filters, with a window length equal to one fundamental period, such as

$$F_{\mathrm{DFT}}(z) = \frac{2}{M}\sum_{i=0}^{M-1}\left(\sum_{h\in N_h}\cos\left[\frac{2\pi}{M}h(i+N_a)\right]\right)z^{-i}, \tag{5.20}$$

where N_h is the set of selected harmonic frequencies, and N_a is the number of leading steps required to get the phase lead that ensures system stability. Equation (5.20) can be seen as a finite impulse response (FIR) pass-band filter with M taps presenting unity gain at all selected harmonics h. It is also called discrete cosine transform (DCT) filter. One advantage of (5.20) is that the compensation of more harmonics does not increase the computational complexity, and the phase lead can be tuned at the design stage by parameter N_a. Of course, in order to implement the repetitive concept, a delay line with N_a taps is needed in the feedback path to recover zero phase shift of the loop gain ($F_{\mathrm{DFT}}(z)z^{-N_a}$) at the desired frequencies, which is a necessary condition to have gain amplification. Another advantage of (5.20) is that its structure is highly adapted to the typical DSP architecture, where the execution of multiply and accumulate instructions normally requires a single clock cycle. This makes the DFT-based controller extremely effective, particularly if compared to the implementation of a bank resonant filter.

Considering now our example case, we would like to briefly outline the design procedure for the DFT filter based voltage controller. The rotating reference frame PI design is straightforward: a conventional digital PI is designed for the UPS (Section 5.3.1, Aside 9) and then turned into the rotating equivalent of Fig. 5.15. This requires simply the doubling of the integral gain for the resonant filter part of the regulator, while the proportional gain is exactly the same.

The design of the DCT filter is quite easy as well: since we do not need to recover the system phase, thanks to the relatively high ratio of sampling frequency and required crossover frequency, parameter N_a can be simply set to zero. The number of filter taps is then given

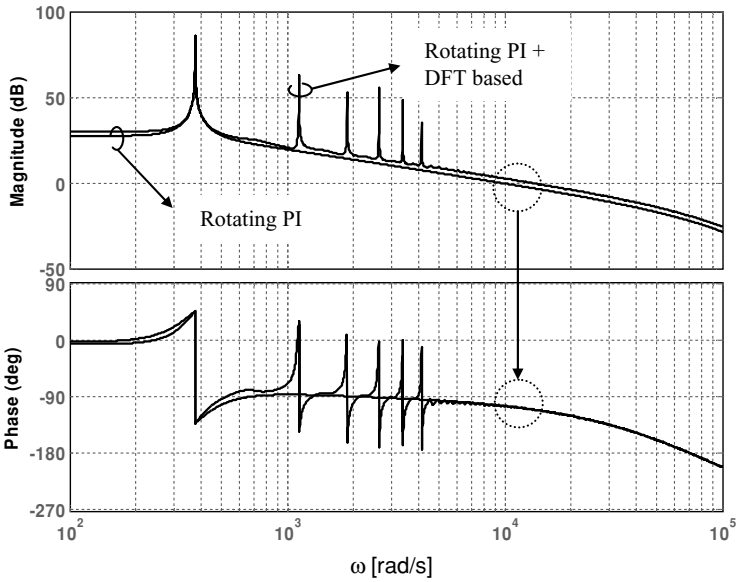

FIGURE 5.16: Open loop system gain for the DFT-based controller.

by the ratio of the sampling frequency and the fundamental output voltage frequency that, in order to avoid leakage effects on the DFT filter, must be an integer number. Because of this constraint, as we did before, we slightly changed the sampling frequency to 48 kHz so as to get $M = 800$. The Bode plot of the obtained open loop gain is shown in Fig. 5.16. It is interesting to compare this figure to Fig. 5.13. As can be seen, gain amplification takes places only at the predefined frequencies, determining little effects on the system phase margin. The stability of the closed loop system is consequently determined by the PI controller's design, as in a conventional implementation. In order to limit the computational effort and the memory occupation, a sample decimation by a factor M_c can be used in the FIR filter implementation, similarly to what we have done for the repetitive control. More precisely, in our example, M has been reduced by a factor of 10 ($M_c = 10$, $M = 80$) or even by a factor of 20 ($M_c = 20$, $M = 40$) without significantly affecting the dynamic performance. Similarly to the repetitive control, the main issue related to the use of decimation is that the output of the DCT filter is updated only every M_c samples and it is seen by the proportional controller as a stepwise function. In order to emulate an interpolator, a moving average filter with M_c taps has been adopted.

As far as the design of the gain K_F is concerned, we can follow the same guidelines that we have illustrated in Chapter 4, Aside 7, when we described the design of a rotating reference current controller. This may seem surprising, at first, but we must recall that the DFT filter is nothing but a bank of parallel resonant filters, each tuned on one of the harmonics to be

compensated. In Chapter 4, we have exactly shown that a rotating reference controller is also equivalent to a tuned resonant filter, therefore the same criteria can be adopted for the design of the controller gain in both cases [14]. In the end, the effect of this gain is to determine the settling time of the DFT-based controller to any disturbance. In the considered example, it was set to a value corresponding to a settling time equal to 10 fundamental periods.

To complete the design, we still need to specify the set of harmonics we want to compensate. In our example case, this was set to {3, 5, 7, 9, 11}.

The controller operation is illustrated by Fig. 5.17, which considers the UPS system behavior in the same conditions of Fig. 5.14. Once again, the controller initially operates only in PI mode. This implies a significant output voltage distortion, which can be observed in Fig. 5.17(c). After 0.1s, the DFT filter based section of the controller is activated, determining the progressive attenuation of the voltage tracking error. As in the previous case, the interaction between the UPS output impedance and the diode rectifier determines an increase in the load current crest factor, as can be seen comparing Figs .5.17(d) and 5.17(f). An important difference with the previous example is represented by the internal current controller: in this case a purely proportional current regulator was employed. This is the reason why the current tracking error, visible in the left column of Fig. 5.17, is somewhat higher than that we can observe in Fig. 5.14. Nevertheless, considering the right column of Fig. 5.17 we can appreciate the very satisfactory performance of the DFT-based controller. This allows us to conclude that as far as a narrow bandwidth voltage controller's effectiveness is concerned, the presence of a high-performance internal current controller is not essential. Indeed, in the steady state the quality of the harmonic compensation can be very high. Of course, in dynamic conditions, i.e., in the presence of load step changes or other fast transients, the system's speed of response and its damping, which are also functions of the current loop bandwidth, could be unacceptable. However, in the case where a limited bandwidth current controller has to be accepted, the phase lead effect of the DFT controller can be exploited to increase the system's phase margin and push the bandwidth very close to the limit.

5.5 OTHER APPLICATIONS OF THE CURRENT CONTROLLED VSI

We would like to conclude the discussion of external control loops for current controlled voltage source inverters by briefly describing a couple of other important applications where the multiloop organization is often employed. These are the controlled rectifier and the active power filter.

They are fundamentally similar, with the hardware organization being exactly the same. In both applications the VSI is connected to a primary source of energy, which can be simply the utility grid or any other, more complex, power system. In both of them, the VSI has to

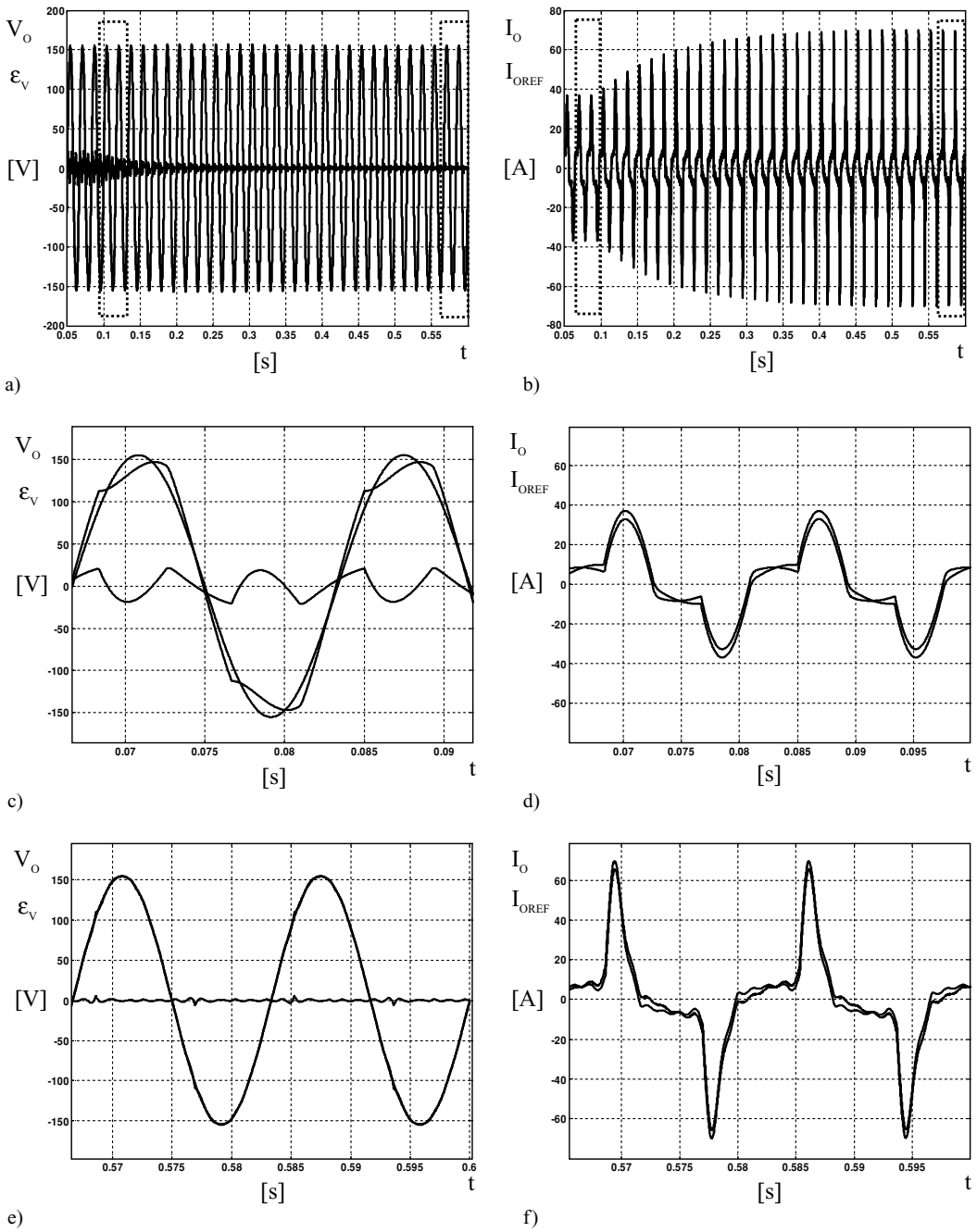

FIGURE 5.17: DFT filter based controller operation. (a) output voltage transient; (b) output current transient; (c) details of (a) before the repetitive controller is activated; (d) details of (b) before the repetitive controller is activated; (e) details of (a) after the steady state is reached with the DFT-based controller; (f) details of (b) after the steady state is reached with the DFT-based controller.

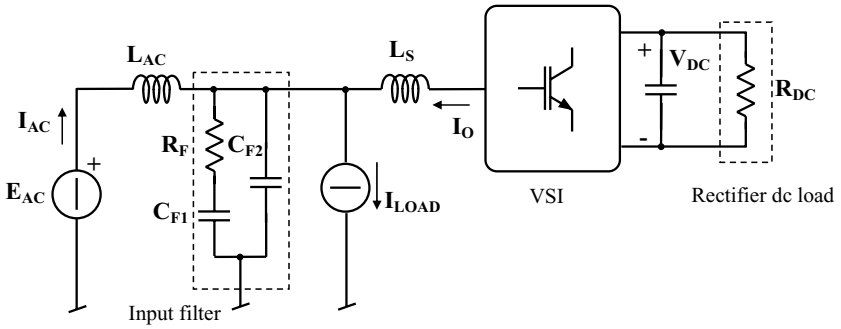

FIGURE 5.18: Typical organization of a controlled rectifier or active power filter.

impose a predefined, controlled current onto the source. The main difference between the two is represented by the fact that the controlled rectifier directly supplies power to a dc load, while the active power filter not necessarily does, being typically employed only to compensate undesired harmonic current components and/or reactive power injected into the source by other distorting and/or reactive loads. Because of this, the design criteria adopted for the power converter can be different in the two cases. In order to visualize the typical organization of both these applications we can refer to Fig. 5.18.

As can be seen, the VSI, which can be single or three phase, is normally connected to the ac power source through an input filter. This is used to attenuate the high-frequency components of the converter output current injected into the source. Apart from that, we can immediately recognize the same basic structure considered in our discussion of current control implementations. We can therefore conclude that, with the exception of minor modifications that may be required to take the input filters into account, current controllers for PWM rectifiers and active filters can be based exactly on the same concepts considered in the previous chapters. Although it is possible, at least from the general organization point of view, to treat the two applications in a unified manner, the different goals of the rectifier and the active filter sometimes call for different control strategies. Therefore, we will now analyze them separately.

5.5.1 The Controlled Rectifier

The PWM controlled rectifier can be represented by Fig. 5.18 once the I_{LOAD} generator is not considered and an equivalent dc load, represented for simplicity by resistor R_{DC}, is connected to the converter output. The typical control objective for this converter is to supply the load with controlled dc power, absorbing high-quality (i.e., harmonic and reactive component free) ac power from the source. This requires two different control loops: (i) a current control loop, which is used to impose an ac current I_{AC} on the source, proportional to the input voltage E_{AC}

and (ii) a dc voltage control loop, which is used to regulate the load voltage, V_{DC}, keeping it to a predefined value, even in the presence of load and/or line voltage variations.

The current control loop does not need to be particularly fast: indeed the typical reference waveform, proportional to the ac source voltage, is represented by a practically sinusoidal signal. Even if the source were affected by a significant harmonic distortion, a current loop bandwidth in the order of 10 to 20 times the source fundamental frequency would allow us to track the reference without appreciable errors. These are the typical grounds for the application of PI current controllers.

In the case of a three-phase power system, the modulator and current controller can take advantage of the techniques discussed in Chapter 4. These become particularly useful in the case where we consider a medium power rectifier, rated for several tens of kVA. In that case, it is likely that the sampling and switching frequency is kept relatively low, making it difficult to guarantee a good reference tracking even at the fundamental frequency. Rotating reference controllers, possibly implemented as banks of resonant filters, are in this case particularly effective.

As far as the outer control loop is concerned, its goal is to adjust the current reference amplitude so as to keep the load voltage on the desired set-point. In single-phase systems, the instantaneous power unbalance determines a dc voltage ripple across the dc link capacitor [1, 2], which has to be filtered by the voltage regulator in order not to determine input current distortion. This implies the need for a limitation of the regulation loop bandwidth to a fraction, typically about one tenth of the fundamental input frequency. Because of this, the design of the output voltage regulator is normally quite easy, due to dynamic specifications not being so stringent. Once again, this is a typical situation where a PI controller is probably the best choice. In three-phase systems, the input power is constant and there is no instantaneous unbalance. Nevertheless, the voltage loop bandwidth is again typically relatively low.

To design the PI regulator, a suitable dc link voltage dynamic model has to be derived. In order to sketch a design procedure, that is referred to in the single-phase case, we must first realize that the voltage controller actually determines the amount of power absorbed by the rectifier from the ac source. In the steady state, this has to be equal to the sum of the load power and the converter losses. Instead, in dynamic conditions, the dc link capacitor can absorb or deliver the instantaneous power unbalance. Therefore, the fundamental equation that describes the power balance of the system is as follows:

$$\frac{\mathrm{d}}{\mathrm{d}t} E_{C_{DC}} = P_{AC} - P_{loss} - P_{LOAD}. \tag{5.21}$$

In (5.21), $E_{C_{DC}} = \frac{1}{2} C_{DC} V_{DC}^2$ is the energy stored in the dc link capacitor, P_{loss} is the power the converter dissipates, $P_{LOAD} = \frac{V_{DC}^2}{R_{DC}}$ is the power delivered to the load, and P_{AC}, the

input active power under the hypothesis of unity power factor rectifier operation, is given by

$$P_{AC} = G_{EQ} \cdot E^2_{AC\_RMS}, \tag{5.22}$$

where G_{EQ} represents the voltage controller output. This, as stated above, represents the desired amplitude of the source current, whose waveform is assumed to be proportional to that of the source input voltage E_{AC}. Rewriting (5.21) in terms of the system parameters we find the following dynamic equation,

$$\frac{1}{2} C_{DC} \frac{d}{dt} V^2_{DC} = G_{EQ} E^2_{AC\_RMS} - P_{loss} - \frac{V^2_{DC}}{R_{DC}}, \tag{5.23}$$

which relates the controlled variable, V_{DC}, to the controller's output G_{EQ}. As can be seen, (5.23) is a nonlinear differential equation; therefore, before a dynamic model can be derived a linearization procedure has to be applied. Of course, since the linearization is based on variable perturbation and small signal approximation, the model will be only valid in the vicinity of a steady-state operating point. However, it is interesting to note that if V^2_{DC} is chosen as the controlled variable, (5.23) becomes linear and can be directly used for the derivation of the system dynamic model, which, in this case, will also be valid for large signals. In other words, controlling V^2_{DC} instead of V_{DC}, which is functionally equivalent, can greatly extend the linearity of the control loop.

In practice, since the dc link voltage V_{DC} is almost constant, affected only by a relatively small low-frequency ripple, the difference in the achievable performance between the two possible approaches is very small.

Linearization of (5.22) is done assuming that E_{AC_RMS} and P_{loss} are constant and considering, as usual, each variable quantity to be equal to the superposition of a steady-state component and a perturbation component, i.e., $V_{DC} = \overline{V}_{DC} + v_{dc}$, $G_{EQ} = \overline{G}_{EQ} + g_{eq}$ with obvious meaning of the symbols. Simple calculations yield the following result,

$$\frac{v_{dc}}{g_{eq}}(s) = \frac{R_{DC} E^2_{AC\_RMS}}{2\overline{V}_{DC}} \frac{1}{1 + s C_{DC} \frac{R_{DC}}{2}}, \tag{5.24}$$

which can be used in the design of the dc link voltage regulator. The design of the regulator can follow the same steps as in Chapters 2 and 3, with continuous time synthesis and successive discretization. The only caution we need to take is to limit the required bandwidth and keep it lower than the source fundamental frequency, so as to avoid source current distortion.

5.5.2 The Active Power Filter

The active power filter application can be represented by Fig. 5.18 as well. In this case, the I_{LOAD} generator is considered and used to represent the distorting or reactive loads the filter

has to compensate, while the dc load, R_{DC}, may not be present. If there is no dc load, the active power filter is not required to process any active power, with the exception of that due to its losses, and can thus be sized to sustain only the reactive and harmonic load currents. A typical control objective for this application is to compensate the harmonic and reactive load currents, so as to make the ac source current proportional to the source voltage. This implies that, from the source standpoint, the load will be seen as an equivalent resistor, absorbing only the active power required by the distorting loads. The achievement of this objective requires again two different control loops: (i) a current control loop, used to impose the desired ac current I_{AC} to the source, and (ii) a dc voltage control loop, used to regulate the load voltage, V_{DC}, keeping it equal to a given reference value.

Apparently, this situation seems identical to that of the rectifier discussed in the previous section. This is actually the case for the voltage loop, which can be designed exactly as that of the rectifier. It is not at all the case for the current loop: the compensation of high-order harmonic currents normally requires some high-performance current control loop. Indeed, the implementation of a simple PI current controller is normally able to offer only a limited harmonic compensation capability, which is very often quite far from being satisfactory.

Therefore, more complex solutions have to be taken into account. As we have illustrated for the UPS voltage loop, in this case it is as well possible to follow two different design philosophies: (i) implementing a large bandwidth current controller or (ii) implementing a narrow bandwidth current controller. The former solution is aimed at the instantaneous compensation of any deviation of the current injected into the line from its reference waveform. The latter is instead aimed at the slow compensation of the same deviation, typically requiring several fundamental frequency periods to be accomplished.

The large bandwidth controllers that, in the digital domain, are exactly of the predictive type we have discussed in Chapter 3 are normally suited to all those situations where the distorting and harmonic load currents are characterized by unpredictable and frequent variations.

The narrow bandwidth controllers can be based on the resonant filters or, equivalently, on the rotating reference frame regulators seen in Chapter 4. In the active filter application, several parallel regulators will be implemented to take care of the different harmonic frequencies to be compensated. Repetitive or DFT filter based controllers, of the type seen in Section 5.4, are also viable solutions. Of course, since the dynamic response of these regulators normally extends to some fundamental frequency periods, their adoption should be limited to those cases where the distorting and reactive load currents are not subject to frequent variations and therefore the controller steady state is not too frequently perturbed. The design of the narrow bandwidth regulators exactly follows the principles we have illustrated for the UPS voltage control case.

The last issue we need to examine to complete this brief description of active power filter control is related to the generation of the inverter reference current signal. From Fig. 5.18 we

can see that in order to achieve the desired compensation and inject a voltage proportional current into the ac source, the inverter simply needs to generate a current equal to the algebraic sum of the desired source current and the load current. Therefore, in the most simple approach the inverter current reference can be built as

$$I_{\mathrm{OREF}} = -I^{*}_{\mathrm{AC}} + I_{\mathrm{LOAD}} = -G_{\mathrm{EQ}}E_{\mathrm{AC}} + I_{\mathrm{LOAD}}, \tag{5.25}$$

where G_{EQ}, as in the rectifier case, represents the output of the dc link voltage regulator. Of course, the implementation of (5.25) is straightforward only if the measurement of the distorting and harmonic loads' current I_{LOAD} is possible. If this is the case, the result of its application will be the cancelation of the reactive current component from the ac source current. In addition, any harmonic current not present in the ac source voltage will also be canceled. The quality of the cancelation is, of course, limited only by the chosen current controller reference tracking capabilities [15].

If current I_{LOAD} cannot be measured, or if the active power filter is designed for more complex tasks, like the partial, controlled compensation of some selected harmonics and/or the compensation of the load reactive power only, different approaches for the computation of the converter current reference can be employed, the illustration of which, however, goes beyond the scope of this book.

REFERENCES

[1] N. Mohan. T. Undeland and W. Robbins, *Power Electronics: Converters, Applications and Design*. New York: Wiley, 2003.

[2] J. Kassakian, G. Verghese and M. Schlecht, *Principles of Power Electronics*. Reading, MA: Addison-Wesley, 1991.

[3] Y. Dote and R. G. Hoft, *Intelligent Control—Power Electronic Systems*. Oxford: Oxford University Press, 1998.

[4] M. J. Tyan, W. E. Brumsickle and R. D. Lorenz, "Control topology options for single-phase UPS inverters," *IEEE Trans. Indust. Appl.*, Vol. 33, No. 2, pp. 493–500, March/April 1997. doi.org/10.1109/28.568015

[5] S. Buso, S. Fasolo and P. Mattavelli, "Uninterruptible power supply multi-loop control employing digital predictive voltage and current regulators," *IEEE Trans. Indust. Appl.*, Vol. 37, No. 6, pp. 1846–1854, Nov. /Dec. 2001.doi.org/10.1109/28.968200

[6] O. Kükrer, "Deadbeat control of a three-phase inverter with an output LC filter," *IEEE Trans. Power Electron.*, Vol. 11, No. 1, pp. 16–23, Jan. 1996.doi.org/10.1109/63.484412

[7] O. Kükrer and H. Komurcugil "Deadbeat control method for single-phase UPS inverters with compensation of computational delay," *IEE Proc., Electr. Power Appl.*, Vol. 146, No. 1, pp. 123–128, Jan. 1999.dx.doi.org/10.1049/ip-epa:19990215

[8] A. Kavamura, T. Haneyoshi and R. G. Hoft, "Deadbeat controlled PWM inverter with parameter estimation using only voltage sensor," *IEEE Trans. Power Electron.*, Vol. 3, No. 2, pp. 118–124, April 1988.doi.org/10.1109/63.4341

[9] T. Yokoyama and A. Kawamura, "Disturbance observer based fully digital controlled PWM inverter for CVCF operation," *IEEE Trans. Power Electron.*, Vol. 9, No. 5, pp. 473–480, Sept. 1994.doi.org/10.1109/63.321031

[10] P. Mattavelli, "An improved dead-beat control for UPS using disturbance observers," *IEEE Trans. Indust. Electron.*, Vol. 52, No. 1, pp. 206–212, Feb. 2005.doi.org/10.1109/TIE.2004.837912

[11] M.Morari and E.Zafiriou, *Robust Process Control*. Englewood Cliffs, NJ: Prentice-Hall, Jan. 1989.

[12] Y. Y. Tzou, R. S. Ou, S. L. Jung and M. Y. Chang, "High-performance programmable AC power source with low harmonic distortion using DSP-based repetitive control technique," *IEEE Trans. Power Electron.*, Vol. 12, No. 4, pp. 715–725, July 1997.doi.org/10.1109/63.602567

[13] K. Zhang, Y. Kang, J. Xiong and J. Chen, "Direct repetitive control of SPWM inverters for UPS purpose," *IEEE Trans. Power Electron.*, Vol. 18, No. 3, pp. 784–792, May 2003.doi.org/10.1109/TPEL.2003.810846

[14] P. Mattavelli, "Synchronous frame harmonic control for high-performance AC power Supplies," *IEEE Trans. Indust. Appl.*, Vol. 37, No. 3, pp. 864–872, May/June 2001.doi.org/10.1109/28.924769

[15] S. Buso, L. Malesani and P. Mattavelli, "Comparison of current control techniques for active filter applications," *IEEE Trans. Indust. Electron.*, Vol. 45, No. 5, pp. 722–729, Oct. 1998.doi.org/10.1109/41.720328

CHAPTER 6

Conclusions

This book has been conceived to give to the reader a basic and introductory knowledge of some typical power converter control problems and of their digital solutions. Although the presented material has been focused on a single converter topology, i.e., the half-bridge voltage source inverter, the control topics we have been dealing with represent, in our opinion, a significant spectrum of the more frequently encountered digital control applications in power electronics.

Moving from the pulse width modulation modeling, we have described the fundamental types of digital current control loop implementation, i.e., the PI controller and the predictive controller. These basic techniques have subsequently allowed us to present the fundamental issues related to three phase current control, with particular consideration for the concepts of rotating reference frame and the controllers that can be based on it.

In the last part of our discussion, we have approached some more advanced control organizations, essentially based on multiloop strategies. We have consequently presented the typical case of the voltage controller for a single-phase uninterruptible power supply. We have seen how both large bandwidth and narrow bandwidth control strategies can be digitally implemented, and analyzed their merits and limitations. In addition, we have seen how the controllers we have analyzed can allow the implementation of other applications of voltage source inverters, like the controlled rectifier of the active power filter.

Of course, we are aware that a lot of other extremely interesting applications could have been dealt with, and also that the more advanced research topics could have been taken into account and presented. We hope the choice we have made, for the sake of conciseness, and the method we have chosen to present the selected material, starting from the very basic issues, will be good enough to give to the readers that we have not been able to completely satisfy the motivation for further autonomous study.

On the other hand, we hope that what has been presented will allow inexperienced readers to successfully experiment with digital control techniques in power electronics.

About the authors

Simone Buso graduated in electronic engineering at the University of Padova in 1992. He received the Ph.D. degree in industrial electronics and informatics from the same university in 1997. Since 1993, he has been cooperating with the power electronics research group of the University of Padova. Currently he is a member of the staff of Department of Information Engineering (DEI) of the University of Padova, where he is holding the position of associate professor. His main research interests are in the industrial and power electronics fields and are specifically related to dc/dc and ac/dc converters, smart power integrated circuits, digital control and robust control of power converters, solid state lighting, electromagnetic compatibility applied to integrated circuits, and switch mode power supplies. Simone Buso is a member of the IEEE.

Paolo Mattavelli graduated (with honors) and received the Ph.D. degree, both in electrical engineering, from the University of Padova (Italy) in 1992 and in 1995, respectively. From 1995 to 2001, he was a researcher at the University of Padova. In 2001 he joined the Department of Electrical, Mechanical and Management Engineering (DIEGM) of the University of Udine, where he was an associate professor of Electronics from 2002 to 2005. From 2001 to 2005 he was leading the Power Electronics Laboratory of the DIEGM at the University of Udine, which he founded in 2001. Since 2005 he has been with the Department of Technology and Management of Industrial System at the University of Padova in Vicenza. He was also Visiting Researcher at the Massachusetts Institute of Technology in 1995 and in 1997.

His major fields of interest include analysis, modeling, and control of power converters, digital control techniques for power electronic circuits, active power filters and power quality issues.

Paolo Mattavelli is a member of IEEE Power Electronics, IEEE Industry Applications, IEEE Industrial Electronics Societies, and the Italian Association of Electrical and Electronic Engineers (AEI). He currently (2006) serves as an associate editor for IEEE Transactions on Power Electronics, IPCC (Industrial Power Converter Committee) Review Chairman for the IEEE Transactions on Industry Applications, and Member-at-Large of PELS Adcom.