UNIVERSITY OF STRATHCLYDE

DEPARTMENT OF ELECTRONICS AND
ELECTRICAL ENGINEERING

MATRIX CONVERTER APPLICATION TO FACTS AND EMBEDDED GENERATION

BY

MOHAMED TAWFIK MOUSTAFA HASSAN

A thesis presented in fulfilment of the requirements for
the degree of Doctor of Philosophy

2007

TEXT BOUND INTO

THE SPINE

To my parents

ABSTRACT

Power electronics is the key technology for connecting flexible ac transmission system devices and renewable power generation to the ac grid. Among the ac to ac converters, the matrix converter represents a promising alternative power electronics technology to back to back converters. It does not incorporate any storage elements which minimises the size and weight of the converter. In addition, with the introduction of the reverse blocking insulated gate bipolar transistor, its efficiency can surpass that of back to back converters.

A new approach in controlling the matrix converter output current is introduced. The approach depends mainly on logic circuitry and facilitates power system applications using the matrix converter. Furthermore, it can provide sinusoidal output current under asymmetry voltage conditions. The output current quality is at the expense of the input current in this case. Both sinusoidal input and output currents cannot be achieved under asymmetry voltage conditions due to the lack of any storage elements.

Matrix converter suitability to power system applications such as flexible ac transmission and embedded generation, is investigated particularly under asymmetrical ac supply conditions. The effect of the absence of an intermediate storage element, on the output power and current quality is considered.

The utilization of the proposed control technique in a unified power flow controller, which is the most versatile flexible ac transmission system device, is investigated. The restriction on unified power flow controller operation due to the absence of any intermediate storage element in the matrix converter is also studied.

Special attention is given to asymmetry ac voltage condition effects and how to mitigate this problem for embedded generation. Extraction of the positive and negative sequence components is essential in obtaining non-oscillatory output power, and a dual current controller is used to this end.

Keywords: FACTs, UPFC, matrix converter, embedded generation, asymmetrical supply

ACKNOWLEDGMENT

All praise is due to Allah, whose praise cannot be described by those who speak, whose blessings cannot be enumerated by those who count those who try to cannot give Him His due rights.

I am deeply and forever indebted to my parents for their love and support throughout my entire life. I am also very grateful to my brother and sisters for their encouragement during my study.

I would like to thank the Egyptian Government for their financial support during my study. Special thanks go to the Egyptian Education Bureau staff for their patience and support. I am thankful to them for allowing me extra time to conclude my experimental work.

My sincere thanks go to my professors at Alexandria University for whom I owe my engineering knowledge. A special thanks goes to Prof. *Samir Deghedi* and Dr. *Mohamed El-Habrouk.*

Thanks to all my colleagues whose assistance and feedback have allowed me to improve the contents of this thesis. I would like to express my gratitude to everybody who shared in the realization of my thesis, as well as expressing my apology that I could not mention them personally one by one.

Finally, if this thesis will be helpful for anyone, all praise is due to Allah; if any of the data supplied with this thesis are wrong, misleading, or misrepresented, it is from me and Allah knows best.

TABLE OF CONTENTS

LIST OF SYMBOLS

A	Frequency estimator gain (rad.s^{-2})
B	Phase angle estimator gain (rad.s^{-1})
C_P	Wind turbine power coefficient (s)
C_Q	Wind turbine torque coefficient
e_a, e_b, e_c	instantaneous grid voltages (V)
e_α, e_β	instantaneous grid voltages in stationary reference frame (V)
e_d, e_q	direct and quadrature axis grid voltage in rotating reference frame (V)
e_{d+}, e_{q+}	direct and quadrature axis grid voltage positive components (V)
e_{d-}, e_{q-}	direct and quadrature axis grid voltage negative components (V)
e_{D+}, e_{Q+}	direct and quadrature axis grid voltage positive components (V)
e_{D-}, e_{Q-}	direct and quadrature axis grid voltage negative components (V)
e_{Di}, e_{Qi}	direct and quadrature axis grid voltage of the ith harmonic(V)
e_{DQ}	,direct and quadrature axis grid voltage in complex form(V)
e_{DQ+}	direct and quadrature axis grid positive voltage in complex form(V)
e_{DQ-}	direct and quadrature axis grid negative voltage in complex form(V)
e_{la}, e_{lb}, e_{lc}	Instantaneous *l* bus bar voltage (V)
e_{ma}, e_{mb}, e_{mc}	Instantaneous *m* bus bar voltage (V)
$I_0 ... I_7$	Inverter matrices
i_a, i_b, i_c	instantaneous output currents(A)
i_A, i_B, i_C	instantaneous input currents (A)
i_α, i_β	instantaneous current in stationary reference frame (V)
i_d	output current direct axis component (A)
i_D	input current direct axis component (A)
i_q	output current quadrature axis component (A)
i_Q	input current quadrature axis component (A)
i_{D+}, i_{Q+}	direct and quadrature axis input current positive components (A)
i_{D-}, i_{Q-}	direct and quadrature axis input current negative components (A)

i_{d+}, i_{q+}	direct and quadrature axis output current positive components (A)
i_{d-}, i_{q-}	direct and quadrature axis output current negative components (A)
i_{Di}, i_{Qi}	direct and quadrature axis input current of the i^{th} harmonic(A)
i_{DQ}	direct and quadrature axis input current in complex form(V)
i_{DQ+}	direct and quadrature axis grid positive current in complex form(A)
i_{DQ-}	direct and quadrature axis grid negative current in complex form(A)
$\overline{I_i}$	Input current vector (A)
$\overline{I_o}$	Output current vector (A)
i_q	shunt injected current (A)
I_{sh}	shunt injected current (A)
K_p	Proportional gain (V/A)
K_i	Integral gain (V/A)
m_α, m_β	Rectifier stage duty cycles
m_γ, m_λ	Inverter stage duty cycles
M_d, M_q	Direct and quadrature axis integral error part (V)
$M(\omega_m t)$	low frequency transfer matrix
q	voltage transfer ratio
R_{oa}, R_{ob}, R_{oc}	Zero state rectifier matrices
$R_1 ... R_6$	Rectifier matrices
$S_1 ... S_6$	Rectifier stage switch states
$S_7 ... S_{12}$	Inverter stage switch states
$S_{i/p}$	Total apparent input power (VA)
S_+	Total apparent power due to positive sequence voltage and current (VA)
S_-	Total apparent power due to negative sequence voltage and current (VA)
T_s	sampling time (s)
$T_{s\,eff}$	effective sampling time (s)
v_a, v_b, v_c	instantaneous output voltages (V)
v_A, v_B, v_C	instantaneous input voltages (V)
v_α, v_β	instantaneous voltages in stationary reference frame (V)

v_d	output voltage direct axis component (V)
v_D	input voltage direct axis component (V)
V_{DC}	virtual link dc voltage (V)
v_q	output voltage quadrature axis component (V)
v_Q	input voltage quadrature axis component (V)
v_{d+}, v_{q+}	direct and quadrature axis controller output positive components (V)
v_{d-}, v_{q-}	direct and quadrature axis controller output negative components (V)
$\bar{V}_i$	Input voltage vector (V)
$\bar{V}_o$	Output voltage vector (V)
v_{pq}	Series injected voltage (V)
V_s	series injected voltage (V)
V_S	sending end voltage (V)
V_R	receiving end voltage (V)
V_M	transmission line mid point voltage (V)
u_1, u_2, u_3	rectifier stage level
u_a, u_b, u_c	inverter stage controller outputs (V)
u_{aa}, u_{bb}, u_{cc}	modified inverter stage controller outputs (V)
u_x	offset voltage to modify the inverter stage controller outputs (V)
$\hat{u}$	maximum of modified inverter stage controller outputs (V)
$\check{u}$	minimum of modified inverter stage controller outputs (V)
$\hat{U}$	carrier peak value
X_s	transmission line reactance (Ω)
ϕ_i	Input displacement angle
$\hat{\theta}$	estimated voltage vector angle (rad)
ω_0	output angular frequency (rad/s)
ω_c	input filter cut-off frequency (rad/s)
ω_i	input angular frequency (rad/s)
ω_m	low frequency transfer matrix angular frequency (rad/s)
$\hat{\omega}$	estimated voltage vector angular frequency (rad/s)

LIST OF ABBREVIATIONS

EG	Embedded Generation
FACTS	Flexible ac Transmission Systems
IGBT	Insulated Gate Bipolar Transistor
MC	Matrix Converter
RB-IGBT	Reverse Blocking Insulated Gate Bipolar Transistor
SSSC	Static Synchronous Series Compensator
SSG	Static Synchronous Generator
SVC	Static VAr Compensator
SVS	Synchronous Voltage Source
TCPST	Thyristor-Controlled Phase-Shifting Transformer
TCSR	Thyristor-Controlled Series Reactor
TSSC	Thyristor-Switched Series Capacitor
UPFC	Unified Power Flow Controller
VSC	Voltage Source Converter
VSI	Voltage Source Inverter

CHAPTER ONE

INTRODUCTION

There is no doubt that electricity plays a vital role in our modern life. It runs the economy and supports the high quality of life of the community. Imagine how unfamiliar your world is when the electrical supply goes out: industrial work is disrupted, water systems shut down, food ruins, transportation is interrupted, communications stop, and people are inconvenienced and even frightened. In other words contemporary life as we know comes to an unpredicted halt.

On the14th of August 2003 the biggest blackout in history occurred on the east coast of the USA. Millions of people were without electricity for an extended period of time [1.1]. This blackout was followed by others in the UK, Sweden, and other countries. These blackouts brought the challenges facing the electric power grid supply to the forefront.

1.1 Challenges facing the electrical grid

There are many challenges facing the existing electrical power system to ensure the continuity and quality of electricity without interruption. In the following sections some of these challenges are presented.

1.1.1 Transmission line capacity

One of the main reasons for the blackout that occurred in the USA was overloading of the transmission lines [1.2]. Transmission lines as normally operate at far lesser than their ratings to ensure reliability. This margin guarantees that the electric system can recover from line tripping or generator outage.

But in recent years, growing demand on electric power has approached transmission systems limits, where outage of certain transmission lines could result in the fast interruption of electricity. Additional transmission capacity where it is needed could be the best solution. Unfortunately, electricity grid upgrades and particularly the construction of new transmission lines cannot match the increase in power plant capacity and energy demand. Finding suitable right-of-ways is particularly difficult in industrialized countries [1.3]-[1.5].

An alternative to constructing new transmission lines is to enhance the use of the existing lines. This could be achieved by implementing Flexible AC Transmission System (FACTS) controllers [1.3]-[1.14]. FACTS devices are used for the dynamic control of voltage, impedance, and phase angle of high voltage AC lines. FACTS devices provide planned benefits for enhanced transmission systems through:

- better utilization of existing transmission assets;
- increased transmission system reliability;
- increased dynamic and transient grid stability; and
- increased supply quality for sensitive industries (e.g. computer chip manufacturing).

What is most attractive about FACTS technology is that it creates opportunities for controlling transmitted power and for improving the practical capacity of existing lines. Controlled current flow through a transmission line increases the capacity of existing lines. These FACTS devices are able to act almost instantaneously to power system changes. Among devices, the unified power flow controller (UPFC) is the most versatile device [1.15]-[1.21].

Another alternative to conventional transmission capacity additions is the reformation of the original grid structure. The existing infrastructure consists of three main functions. First the power has to be generated; involving coal, natural gas, hydroelectric or nuclear power plants. Once generated at the power plant, it has to be transmitted to the area where it is to be consumed; this is the transmission function of power delivery. Finally the power has to be delivered to the different customers in the distribution system. This three-step procedure is summarized in fig. 1.1.

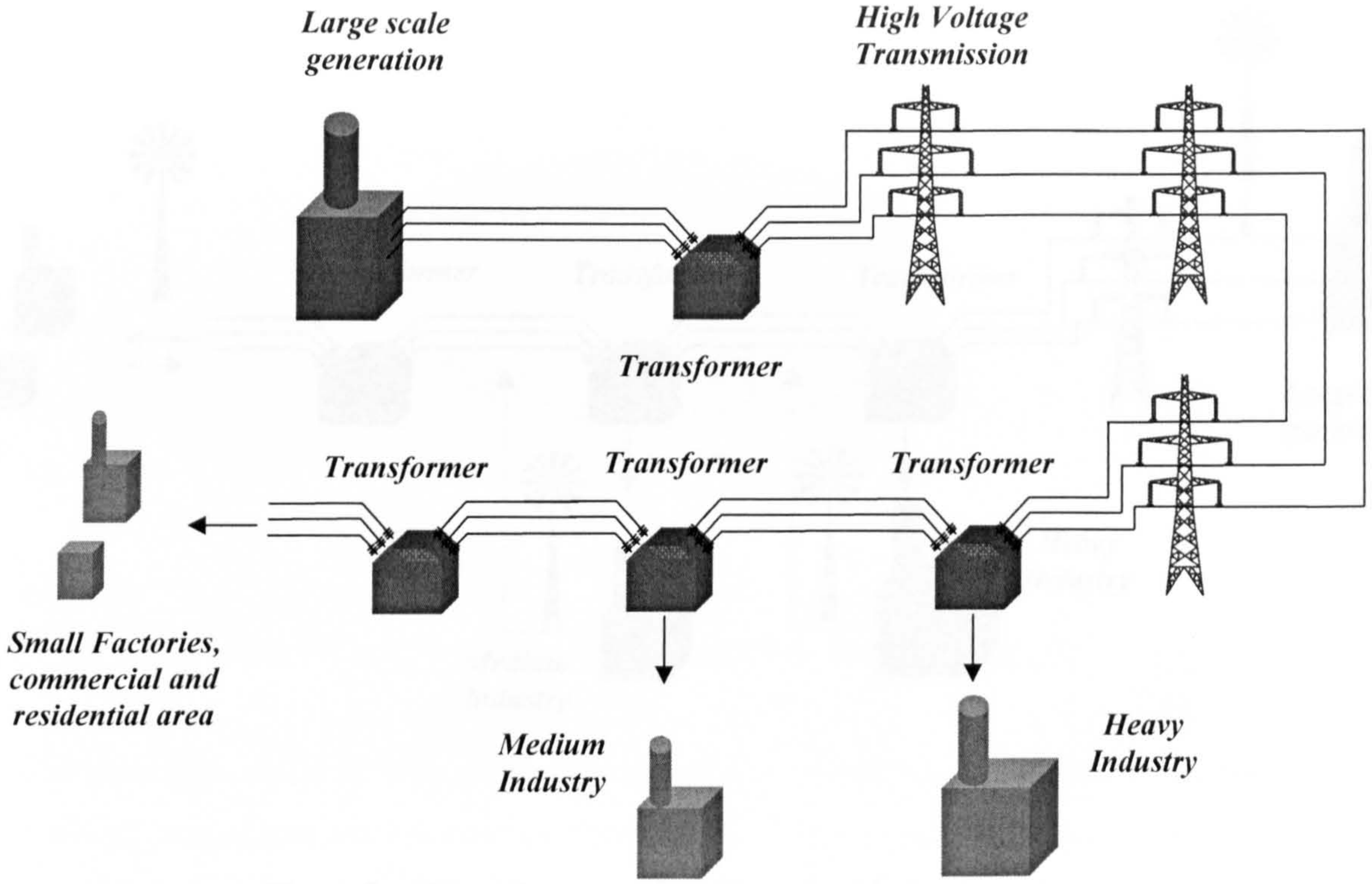


Fig. 1.1 Conventional electric supply network

There is a need to modify the grid structure so that it can perform reliably, even under the higher loads that will unavoidably arise in the future. One proposed addition is to incorporate elements of embedded generation into the system. An embedded generation system [1.22]-[1-30] merges electricity from large and small generation units. Unlike central station power plants, which typically are located far from load centres, embedded generation can produce electricity at or near the place where it is consumed. Embedded generation technologies can run on fossil fuels, renewable energy resources or waste heat. They can meet all or part of a customer's power needs. They can be placed at a customer's site or elsewhere. Fig. 1.2 represents a schematic of embedded generation.

With such a structure, the rate of increase of transmission line loading will be much lower; hence the investment demand on the transmission system will decrease.

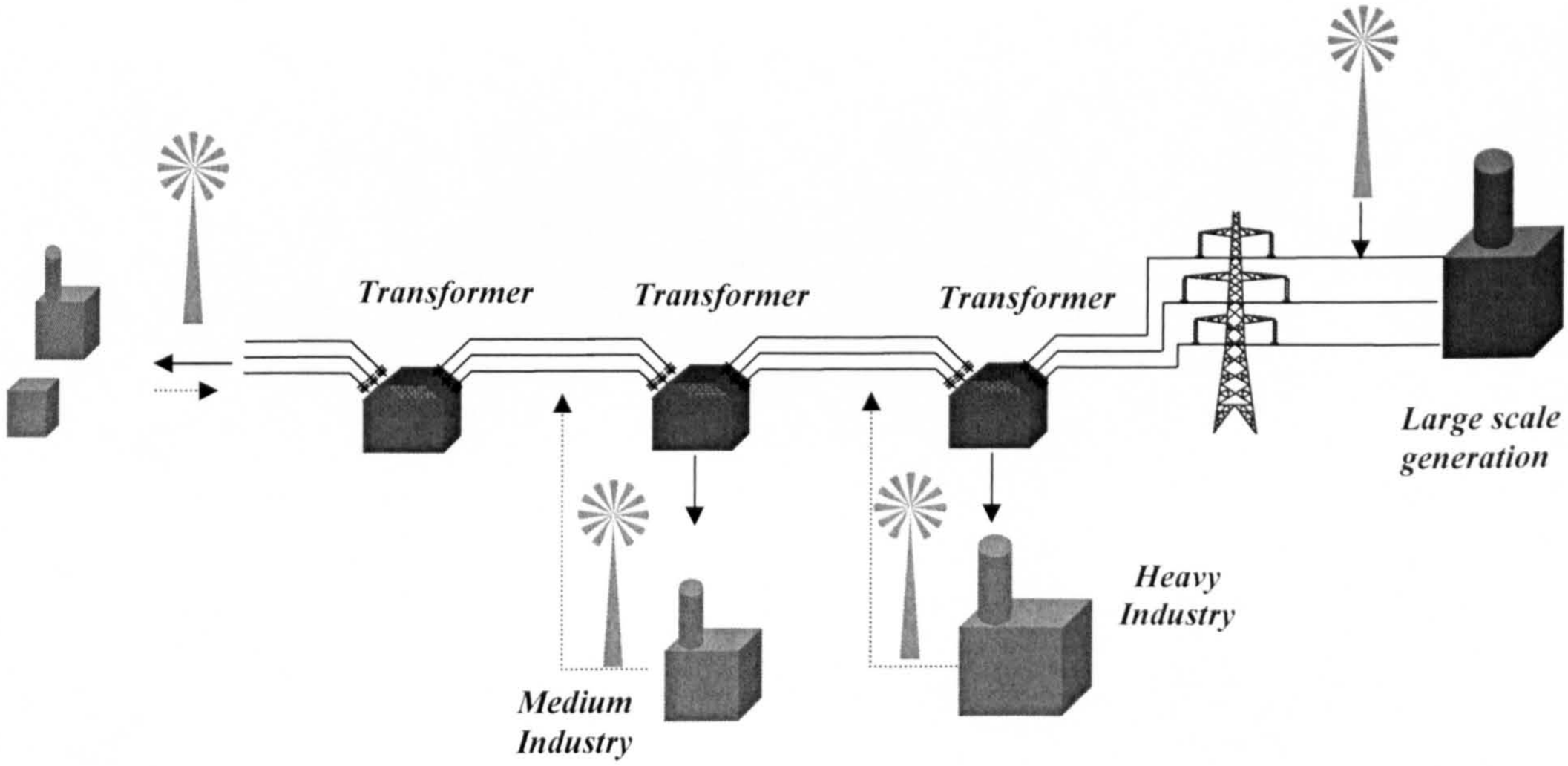


Fig. 1.2 Electric network with embedded generation

1.1.2 Electrical power influence on climate change

Another challenge facing the electrical power system is its influence on climate change. There is no doubt that human activities over the last two decades have influenced the global climate in a dangerous way. During the last two decades, measurements show a significant increase in the emissions of the greenhouse gases such as carbon dioxide (CO_2), methane (CH_4), and nitrous oxide (N_2O), by extensive amounts hence the planet is becoming warmer than it would be otherwise [1.31].

If emissions carry on at the same rate of increase, they will influence rainfall patterns which result in more frequent intense weather conditions world-wide. Higher temperatures will melt the polar ice caps which will cause sea levels to rise. Large areas of the world, including many developing countries, are only a few metres above normal sea level and will suffer from sea level rises as a consequence of climate change.

Carbon dioxide, which is responsible for over 60 per cent of the "enhanced greenhouse effect", are primarily by-products in transport, industry, and the combustion of fossil fuels in electricity generation [1.32] – see Fig. 1.3.

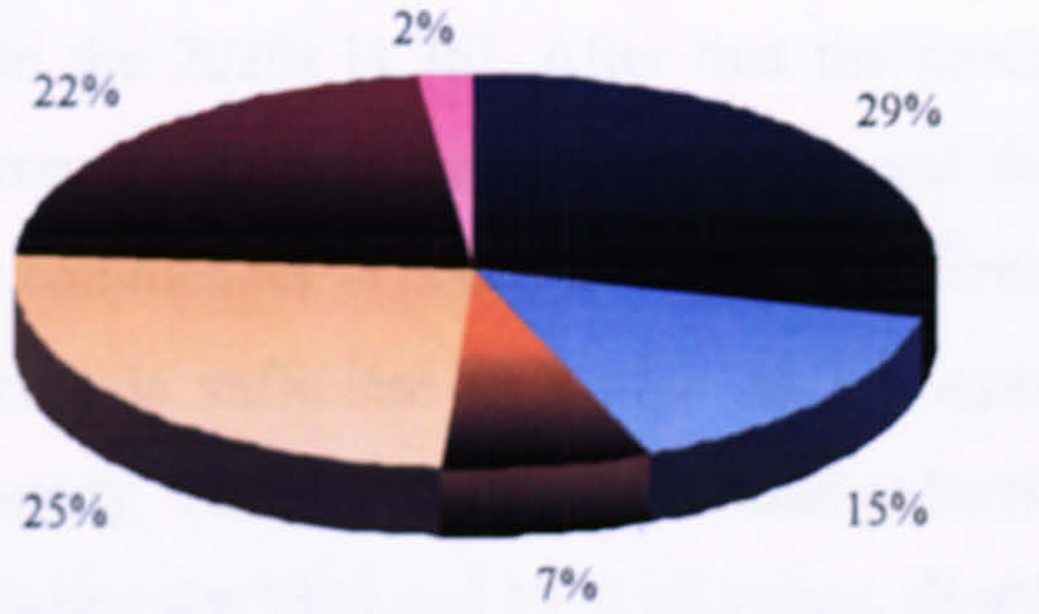


Fig. 1.3 Percentage CO_2 emission resources in UK [1.32]

The principal source of CO_2 is the burning of fossil fuels: coal, natural gas, and oil. Unfortunately, these sources represent most of the resources associated with electricity generation as shown in fig. 1.4.

Most renewable energy sources do not produce any CO_2; hence the UK Government's stated intention to increase their use substantially, particularly in electricity generation [1.33]-[1.35].

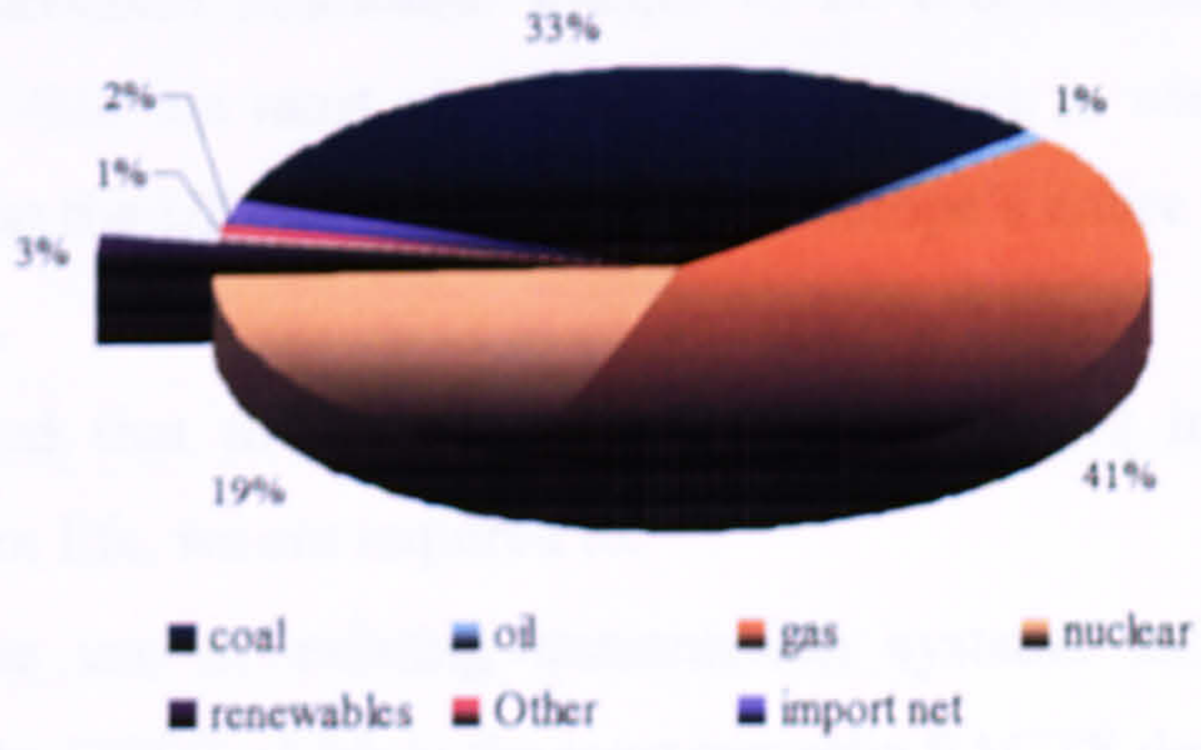


Fig. 1.4 Percentage electricity resources in UK [1.32]

In fact moving towards renewable energy is not only driven by climate change or the Koyoto agreement; if we want to maintain our civilized way of life there is no other option than substitution of oil and natural gas with other sources. Oil and gas were

created during the Earth's long geological history, which means that they are limited resources, subject to depletion. Most research indicates that the peak of oil and gas production could be within the 2020s [1.36]. After that the production will decrease slowly but surely. Our economy is entirely oil-dependant and does not cater for oil reserve depletion before it commences to collapse. A small difference between demand and supply of 15-20 percent is sufficient to shatter an oil-dependent economy and reduce its population to poverty. The effects of even a small reduction in production can be harmful. For instance, during the 1973 and 1978 oil crises, shortfalls in production as low as 5% caused the price of oil to climb to nearly four times its normal price. Fortunately, the crises were only temporary. Future oil shortages will not be so transitory. They represent the beginning of a new, permanent condition [1.37].

Unfortunately, renewable resources are weather dependant which makes it difficult for the electric power grid to rely on them. Nuclear energy does not depend on weather conditions or the time of day in order to produce an output. Also its use does not involve emissions of greenhouse gases. This can make nuclear power a good source for the base load and the remainder can be supplied from renewable resources.

For these reasons, the UK Government has set targets to increase the percentage of renewable electricity generation in the total supply [1.34]. Fig. 1.5 shows the expected timetable for the different renewable sources to be economically viable. From this figure it is noticed that the most vital renewable resource is wind energy. The main reason for this is that the UK has over one third of Europe's entire potential for offshore wind energy [1.34].

It can be concluded that to prevent future electrical power interruption, hence to maintain our modern life, we are required to:

- Enhance the use of existing transmission systems using FACTS devices, especially the UPFC which is the most versatile FACTS device.
- Modify the structure of the existing electrical grid using embedded generation.
- Replace the oil and natural gas used in electrical power generation with other renewable resources; especially wind energy in the UK.

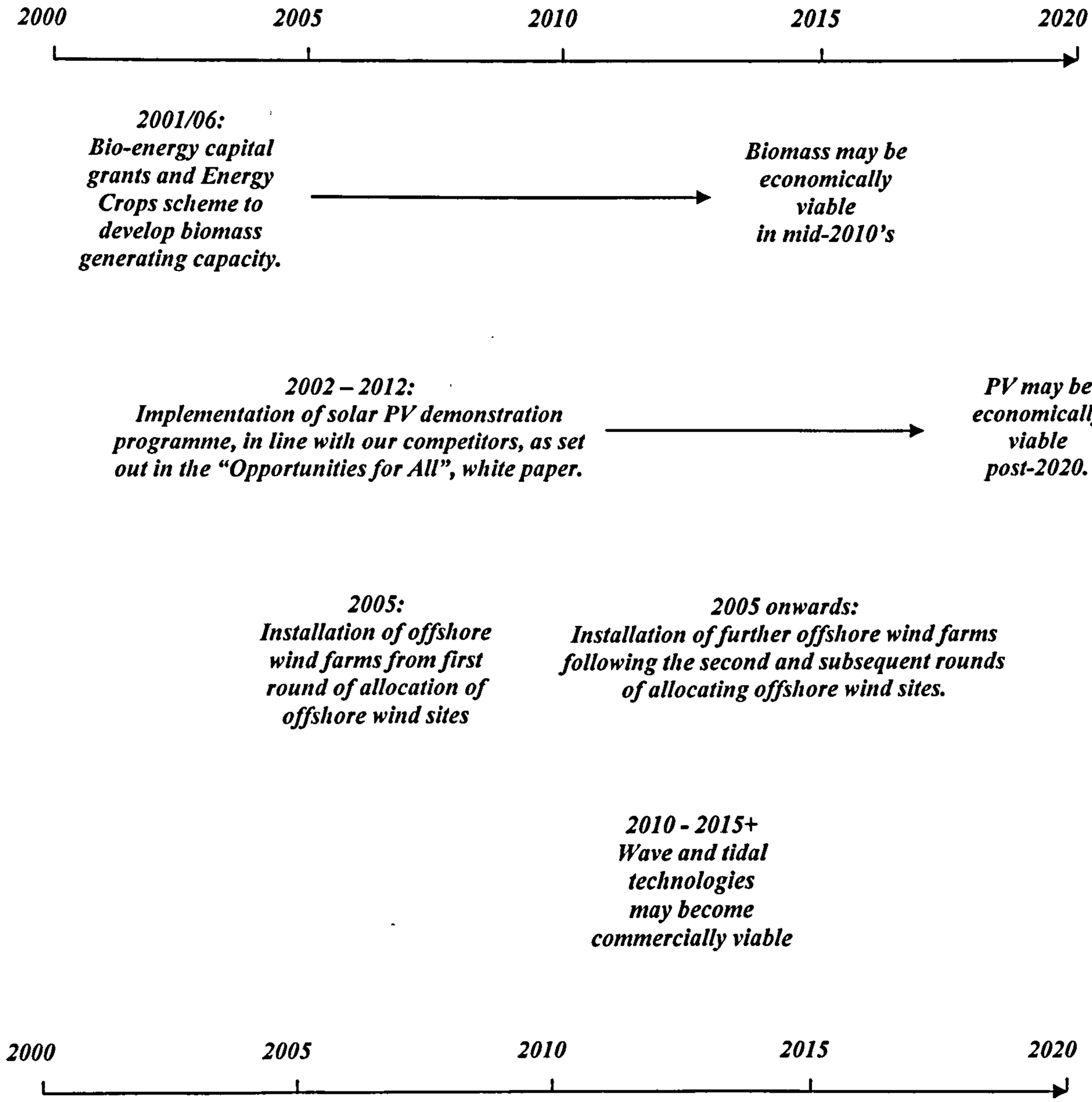


Fig. 1.5 Estimated time table for the installation of renewable energy resources [1.34]

Fig. 1.6 shows a typical UPFC system where series voltage and shunt current are injected into the transmission line. The idea is based on back to back voltage source converters (VSC) where the line voltage is first rectified through VSC1 and then a voltage is injected into the line through VSC2. A variable speed wind turbine system, with full-scale power converters, where the wind turbine is connected through a gear-box to generator G, is shown in fig. 1.7. Usually, the output of the generator is rectified and the resulting power is transferred through a DC link to a voltage source converter.

Typically, filter inductors (L-filters) are used to minimize the current harmonics injected into the grid. Both the UPFC back-to-back converters and the rectifier and the VSC used for power generation from the wind turbine, could be replaced by a matrix converter as shown in figs 1.8 [1.38]-[1.39] and 1.9 respectively.

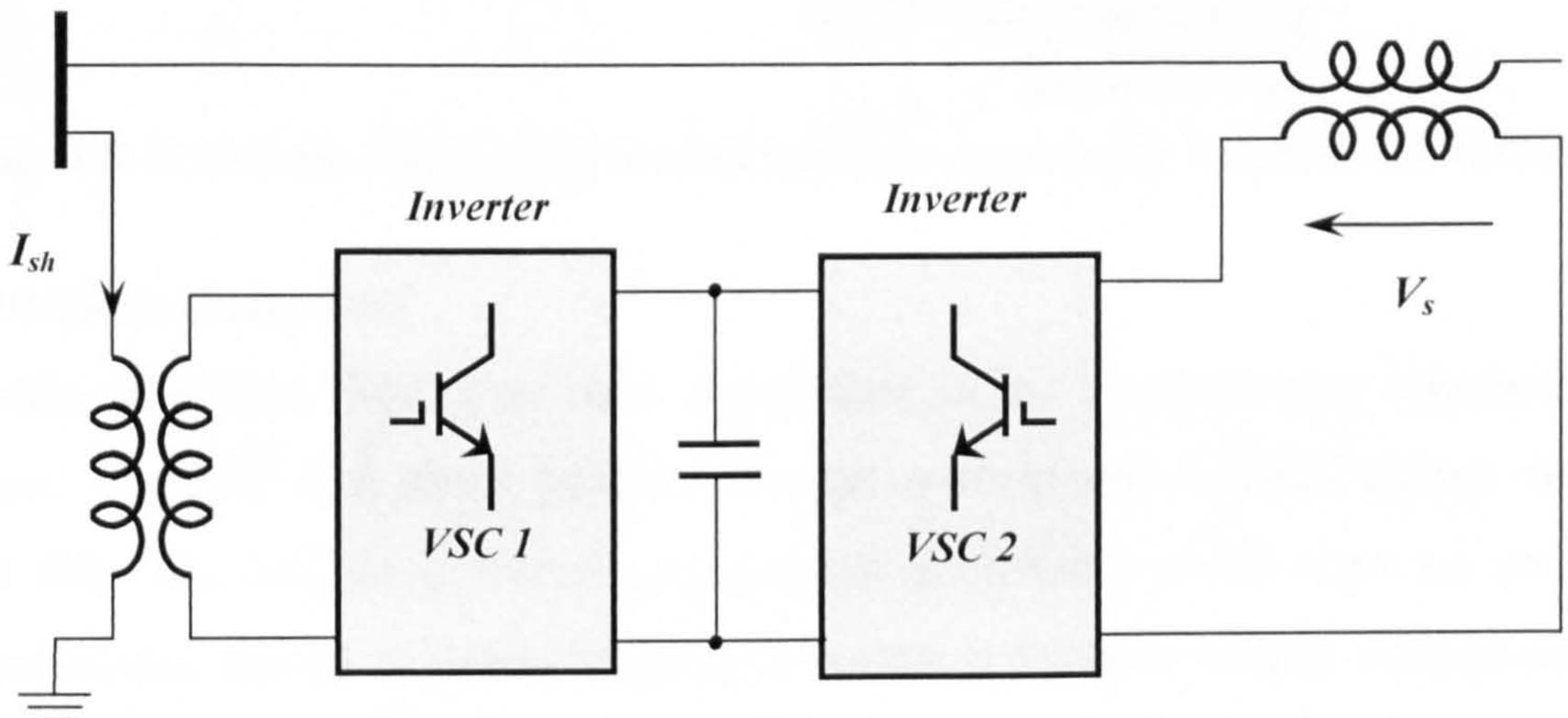


Fig. 1.6 Schematic diagram of the UPFC with back to back converters

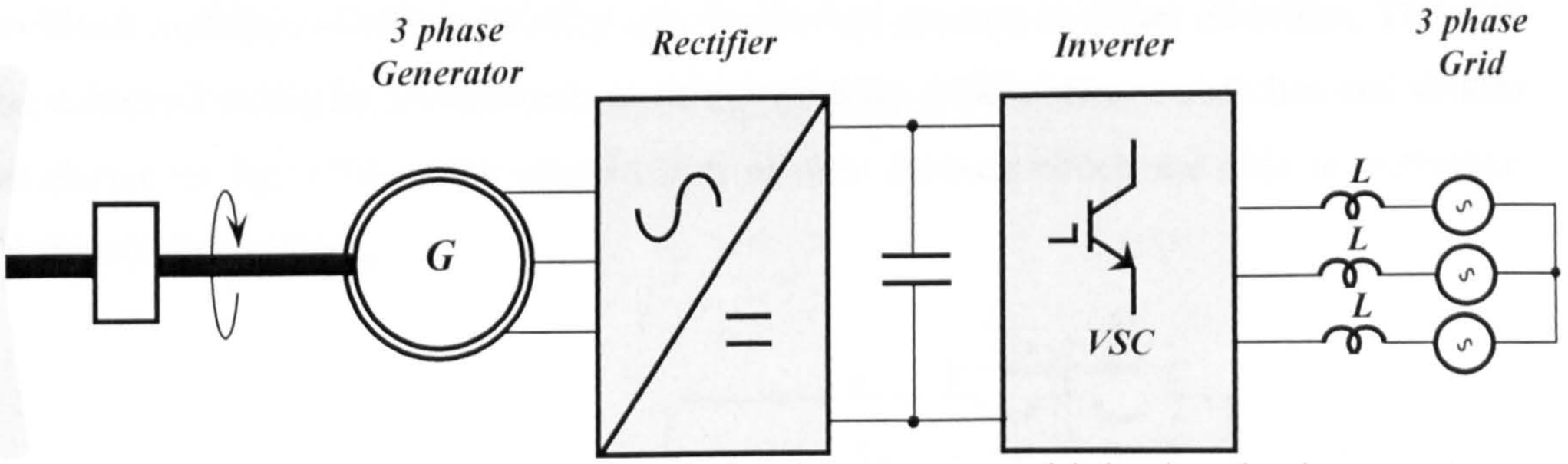


Fig. 1.7 Schematic diagram of a wind turbine system with back to back converters

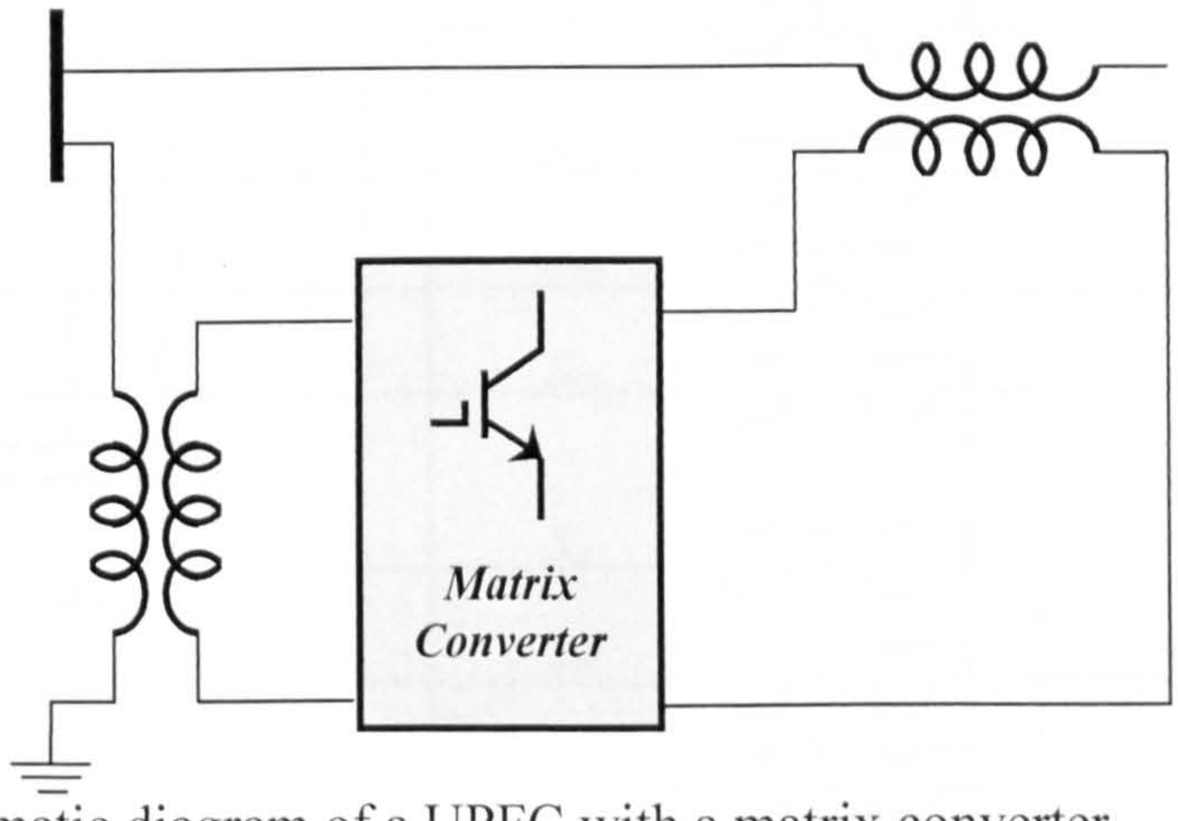


Fig. 1.8 Schematic diagram of a UPFC with a matrix converter

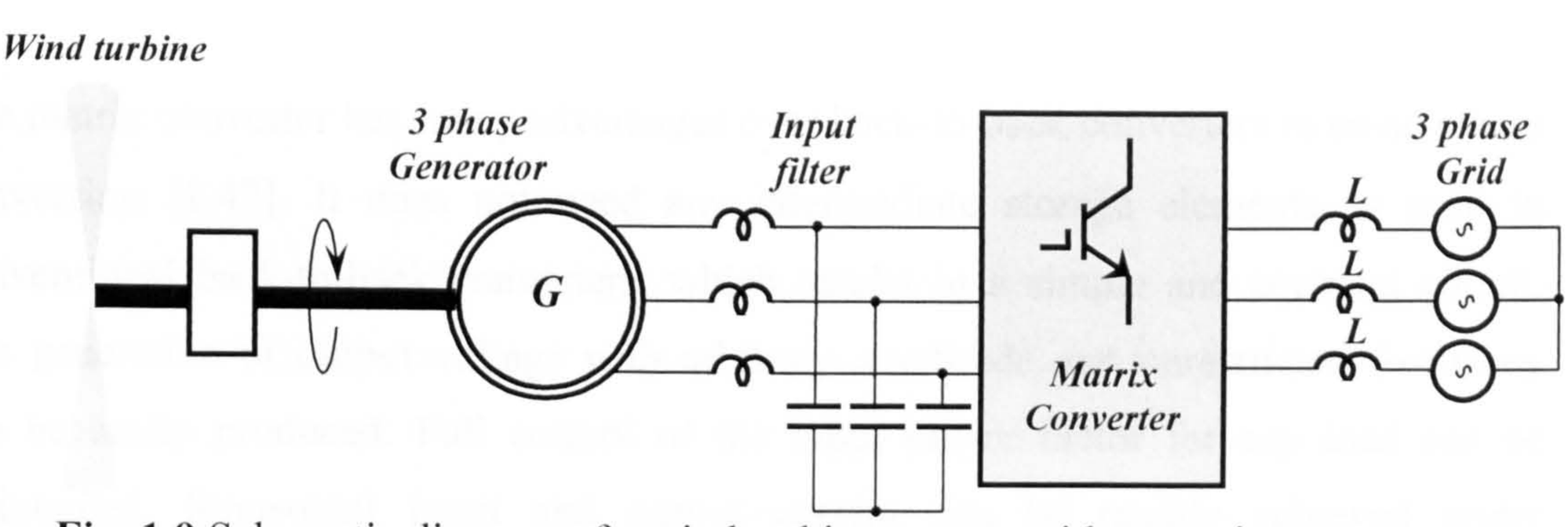


Fig. 1.9 Schematic diagram of a wind turbine system with a matrix converter

1.2 The Matrix Converter

The matrix converter (MC) has been recognized as an alternative to standard VSI converters. The MC is a direct power converter without any internal energy storage stage [1.40]. The MC is a forced commutated converter, which uses an array of controlled bi-directional switches in order to create a variable output voltage system with unrestricted frequency [1.41]. The switches used in a matrix converter must be able to block voltages of either polarity and to conduct current in either direction. This can be achieved either by a combination of the existing unidirectional switches and diodes as shown in fig. 1.10 or the introduction of new devices which are able to withstand bidirectional voltages.

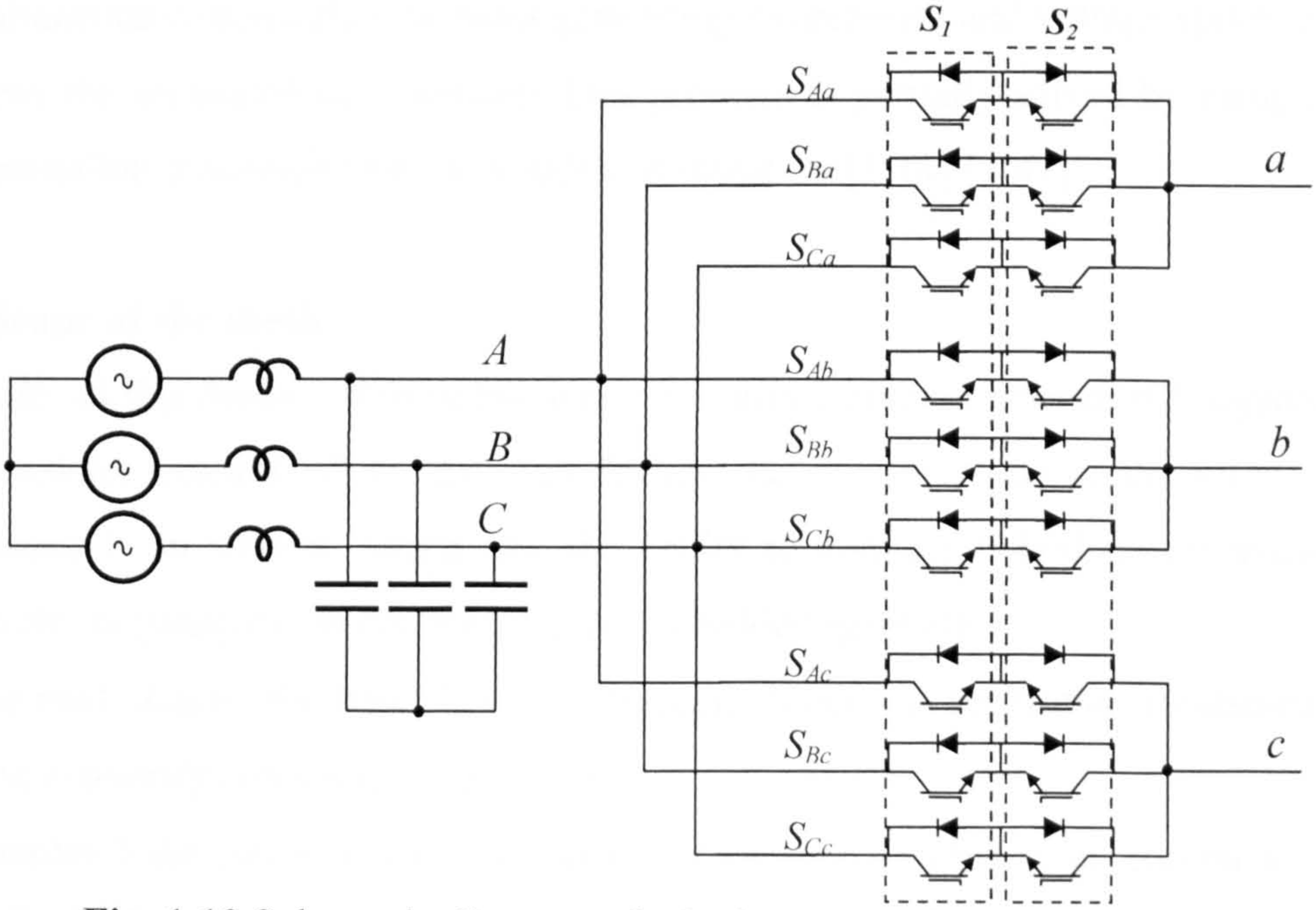


Fig. 1.10 Schematic diagram of a 3-phase to 3-phase matrix converter

The matrix converter has many advantages over back-to-back converters in ac-ac power conversion [1.42]. It does not need any intermediate storage elements as used in conventional back-to-back converters, which results in a simple and compact circuit. The generation of output voltage with arbitrary amplitude and unrestricted frequency can be easily produced. Full control of the input power factor for any load can be maintained. Sinusoidal input and output current can be readily achieved under symmetrical supply conditions.

The reverse blocking IGBT (RB-IGBT) introduced in [1.43] could make a significant impact to the development of the matrix converter since it decreases one semiconductor device per load phase path. If the RB-IGBT device becomes available for mass production, it will minimize the device count and the conduction losses.

On the other hand, the matrix converter has disadvantages [1.40]-[1-42]. Its major drawback is that the maximum voltage transfer ratio between the output and input voltages is limited to 86%. This is a problem if there is need to use a standard machine from a standard transformerless supply. Another drawback is the use of 18 switches in the conventional matrix converter rather than 12 with back to back converters. One more problem is the commutation of the bi-directional switches. It is difficult to achieve simultaneous commutation without generating overcurrent and voltage spikes that could destroy the semiconductor devices. This problem is partially solved by using multistep commutation strategies that allow safe commutation [1.44]-[1.47].

1.3 Scope of the thesis

The aim of this thesis is to describe a novel matrix converter operational approach based on feedback control which not only fulfills the specifications of the UPFC but also surpasses it on several points like the ability to connect linked power systems with different frequencies, which is central to embedded generation.

In the next chapter the basic FACTS concept, embedded generation fundamentals, and matrix converter concepts, are presented.

In chapter 3 the novel matrix converter control approach based on feedback control is described. The performance of this technique is evaluated in chapter 4.

How this approach could be used to fulfill the needs of embedded generation and UPFC specifications, are presented in chapters five and six respectively, which include simulation results.

The experimental test rig and the practical results are presented in chapter seven.

The main conclusions and future work drawn from this thesis are summarized in the final chapter, chapter eight.

References:

[1.1] J. C. Condon, **'Distributed Generation as a Potential Solution to United States Electricity Transmission Bottlenecks'**, University of Arizona, August, 2004.

[1.2] http://www.pserc.wisc.edu/Resources.htm#Resources_NENA_Data.htm.

[1.3] **'Power Compensation in Transmission Systems'**, Siemens Power Engineering Guide, Transmission and Distribution, 4th Edition.

[1.4] R. Grunbaum, A. Peterson and B. Thorvaldsson, **'FACTS-Improving the performance of electrical grids'**, ABB review 3/2002, 11-18.

[1.5] R. Grunbaum, A. Peterson and B. Thorvaldsson, **'FACTS-powerful systems for flexible power transmission'**, ABB review 5/1999, 4-17.

[1.6] P. Moore and P. Ashmole, **'Flexible AC transmission systems'**, Power Engineering Journal [see also Power Engineer], Volume: 9 Issue: 6, Dec. 1995, 282 -286.

[1.7] Ashmole, P.H.; **'Introduction to FACTS'**, Flexible AC Transmission Systems (FACTS) - The Key to Increased Utilisation of Power Systems, IEE Colloquium on (Digest No.1994/005), 12 Jan 1994, 1-2.

[1.8] Xianzhong Duan *et al.*; **'Power flow control with FACTS devices'**, Power Engineering Society Summer Meeting, 2000. IEEE, Volume: 3, 16-20 July 2000, 1585 -1589.

[1.9] Ying Xiao; Song, Y.H. and Sun Y.Z., **'Power flow control approach to power systems with embedded FACTS devices'**, Power Systems, IEEE Transactions on, Volume: 17, Issue: 4, Nov. 2002, 943-950.

[1.10] Chen-Ching Liu, Heydt, G.T. and Edris, A.A., **'Impact of FACTS controllers on transfer capability of power grids'**, Power Engineering Society Winter Meeting, 2002. IEEE, Volume: 1, 27-31 Jan. 2002, 556-561.

[1.11] Mihalic, R., Zunko, P., Povh, D., **'Improvement of transient stability using unified power flow controller'**, Power Delivery, IEEE Transactions on, Volume:11, Issue: 1, Jan. 1996, 485-492.

[1.12] Poshtan, M., Rastgoufard, P., **'Improvement of power system transient stability by using controllable series compensator'**, System Theory, 1998. Proceedings of the Thirtieth Southeastern Symposium, March 1998, 225-228.

[1.13] Hingorani, N.G., **'High Power Electronics and flexible AC Transmission System'**, Power Engineering Review, IEEE, Volume: 8 Issue: 7, July 1988, 3-4.

[1.14] Ralls, K.J., **'The growth of power electronics in electrical power transmission systems'**, Power Engineering Journal, Volume: 9 Issue: 1, Feb. 1995, 15-23.

[1.15] Gyugyi, L., **'A unified flow control concept for flexible AC transmission systems'**, AC and DC Power Transmission, 1991 Conference, Sep 1991, 19-26.

[1.16] Gyugyi, L., **'Unified power-flow control concept for flexible AC transmission systems'**, Generation, Transmission and Distribution IEE Proceedings C, Volume: 139, Issue: 4, July 1992, 323-331.

[1.17] Gyugyi, L., *et al.* **'The unified power flow controller: a new approach to power transmission control'**, Power Delivery, IEEE Transactions, Volume: 10, Issue: 2, April 1995, 1085-1097.

[1.18] Sen, K.K., Stacey, E.J., **'UPFC-unified power flow controller: theory, modeling, and applications'**, Power Delivery, IEEE Transactions, Volume: 13 Issue: 4, Oct. 1998, 1453-1460.

[1.19] Arnez, R.L.V., Zanetta, L.C., **'Unified power flow controller (UPFC): its versatility in handling power flow and interaction with the network'**, Transmission and Distribution Conference and Exhibition 2002: Asia Pacific. IEEE/PES, Volume: 2, Oct. 2002, 1338-1343.

[1.20] Enslin, J.H.R., Zhao, J., Spee, R., **'Operation of the unified power flow controller as harmonic isolator'**, Power Electronics, IEEE Transactions on, Volume: 11, Issue: 6, Nov. 1996, 776-784.

[1.21] Schauder, C.D., *et al.*, **'Operation of the unified power flow controller (UPFC) under practical constraints'**, Power Delivery, IEEE Transactions on, Volume: 13 Issue: 2, April 1998, 630-639.

[1.22] **'Prospects for Distributed Electricity Generation',** The Congress of the

United States, Congressional Budget Office, Sep. 2003.

[1.23] Püttgen, H. B., Macgregor, P., and lambert, F., **'Distributed Generation: Semantic Hype or the Dawn of a New Era?',** IEEE power and energy magazine, Jan./Feb. 2003.

[1.24] **'Distributed Generation: The way forward',** Ofgem factsheet, 26.03.02. http://www.ofgem.gov.uk/Media/FactSheets/Documents1/1102-factsheet060227feb.pdf

[1.25] **'The Tradable Value of Distributed Generation',** ILEX ENERGY CONSULTING, 2005. http://www.aepuk.com/faq_pdf/dgdt100047.pdf

[1.26] L. Schwartz, **'Distributed Generation in Oregon: Overview, Regulatory Barriers and Recommendations',** Feb. 2005. http://www.puc.state.or.us/PUC/electric_gas/dg_report.pdf

[1.27] **'Distributed Generation in Liberalised Electricity Markets',** INTERNATIONAL ENERGY AGENCY, France 2002. http://www.iea.org/textbase/nppdf/free/2000/distributed2002.pdf

[1.28] D. Popovic, **'Managing for Growth in Distributed Generation and Planning for the Future'.** http://www.caenz.com/DistGen/DGdownloads/Popovic1.pdf

[1.29] M. Donkelaar, **'A survey of solutions and options for the integration of distributed generation into electricity supply systems',** Energy & Environment, Volume: 15, No 2 2004. pp 323-332.

[1.30] J. C. Condon, **'Distributed Generation as a Potential Solution to United States Electricity Transmission Bottlenecks',** University of Arizona, Aug. 2004.

[1.31] http://unfccc.int/kyoto_protocol/items/2830.php

[1.32] T. Blundell, **'Wind Power in the UK',** Sustainable Development Commission, May2005.http://www.sd-commission.org.uk/publications/downloads/Wind_Energy-NovRev2005.pdf

[1.33] **'UK Electricity Network',** Parliamentary office of science and technology, Oct. 2001, Number 163. http://www.parliament.uk/post/pn163.pdf

[1.34] **'Our energy future- creating a low carbon economy',** Energy white paper. http://www.dti.gov.uk/files/file10719.pdf

[1.35] J. Bower, **'UK Offshore Wind Generation Capacity: A Return to Picking Winners',** Oxford institute for energy studies, Jul. 2003.

[1.36] H.J. Longwell, director and executive vice president of Exxon Mobil cooperation, **'The future of the oil and gas industry: past approaches, new challenges',** World energy, Volume: 5 Issue: 3, 2002, 100-104.

[1.37] M. Savinar, **'The oil age is over: what to expect as the world runs out of cheap oil 2005-2050'**, http://www.lifeaftertheoilcrash.net/

[1.38] Boon-Teck Ooi; Kazerani, M, **'Unified power flow controller based on matrix converter'**, Power Electronics Specialists Conference, 1996. PESC '96 Record., 27th Annual IEEE, Volume: 1, 23-27 June 1996, 502- 507.

[1.39] Strzelecki R., et al. **'UPFC with matrix converter'**, 9th European Conference on Power Electronics and Applications- EPE 2001. Graz 2001.

[1.40] Wheeler, P., *et al.*, **'Matrix converters'**, Power Engineering Journal, Volume: 16 Issue: 6, Dec. 2002, 273-282.

[1.41] Wheeler, P.W., *et al.*, **'Matrix converters: a technology review',** Industrial Electronics, IEEE Transactions, Volume: 49 Issue: 2, April 2002, 276-288.

[1.42] L. Huber and D. Borojevic, **'Space Vector Modulated Three-Phase to Three-Phase Matrix Converter with Input Power Factor Correction',** IEEE Trans. Industry Appl., Vol. 31, No. 6, pp. 1234-1246, Nov./Dec. 1995.

[1.43] M. Takei, T. Naito, and K. Ueno, **'The reverse blocking IGBT for matrix converter with ultra-thin wafer technology',** Proc. IEEE ISPSD'03, 2003, pp. 156- 159.

[1.44] Adamck, J., Hofmann, W., Ziegler, M., **'Fast commutation process and demand of bidirectional switches in matrix converters',** Power Electronics Specialist, PESC. 2003 IEEE 34th Annual Conference, Volume: 3, June 15-19, 2003, 1281-1286.

[1.45] Ziegler, M., Hofmann, W., **'New one-step commutation strategies in matrix converters',** Power Electronics and Drive Systems, 2001. Proceedings, 2001 4th IEEE International Conference on, Volume: 2, 22-25 Oct. 2001, 560-564.

[1.46] Ziegler, M., Hofmann, W., **'Implementation of a two steps commutated matrix converter',** Power Electronics Specialists Conference, 1999. PESC 99. 30th Annual IEEE, Volume: 1, 27 June-1 July 1999, 175-180.

[1.47] Ziegler, M., Hofmann, W., **'Semi natural two steps commutation strategy for matrix converters',** Power Electronics Specialists Conference, 1998. PESC 98 Record. 29th Annual IEEE, Volume: 1, 17-22 May 1998, 727-731.

CHAPTER TWO

LITERATURE REVIEW

This chapter presents a literature survey of the matrix converter including its fundamentals, the most popular modulation methods, commutation techniques, bidirectional switches, and the effect of input voltages disturbance. Then the basic relationships for power transmission are reviewed. FACTS devices used to control transmitted power are introduced. Special attention is given to the most powerful FACTS device, the unified power flow controller (UPFC). Finally, a brief survey on embedded generation is given, its benefits, problems, and impact on both transmission and distribution networks.

2.1 The Matrix Converter

The matrix converter (MC), shown in fig.2.1, has been recognized as an alternative to the standard voltage source inverter (VSI). The MC is a direct type of power converter without any internal energy storage. It is a forced commutated converter, which uses a matrix of controlled bi-directional switches, in order to create a variable output voltage system with unrestricted frequency. The initial research contributing to matrix converter development used thyristors [2.1]. With this technology, performance was poor. Genuine development of the matrix converter occurred with the "Venturini modulation method" in 1980 [2.2]-[2.4], but the main limitation of the algorithm was its dependence on load power factor. Another control technique based on the "fictitious dc link" scheme was proposed by Rodriguez in 1983 [2.5]. With this method, switching is arranged so that each output line is switched between the most positive and most negative input lines using a pulse width modulation (PWM) technique, as conventionally used in standard VSIs.

This concept is also known as the "indirect transfer function" approach. In 1989, Huber

et al. published the first of a series of papers [2.6]-[2.10] in which the principles of space-vector modulation (SPVM) were applied to the matrix converter modulation problem.

The aim of this section is to present a review of matrix converter fundamentals, including:

- matrix converter fundamentals;
- modulation methods;
- commutation techniques;
- devices for bidirectional switches;
- input voltages disturbance; and
- protection and clamping circuitry.

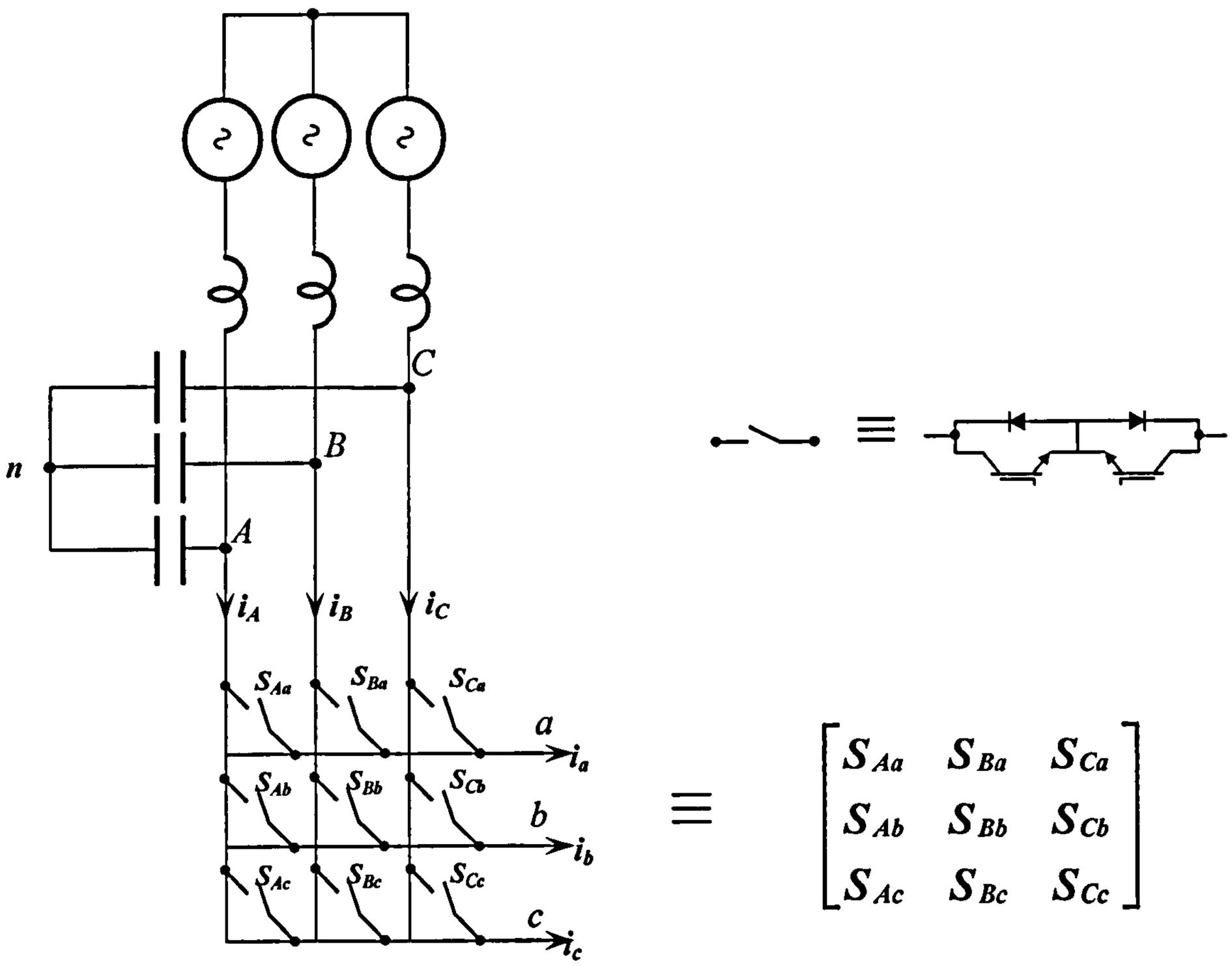


Fig.2.1 Schematic diagram of the conventional matrix converter

2.1.1 Matrix Converter Fundamentals

Normally the matrix converter is fed by a voltage source, and for that reason the input terminals should not be short-circuited. On the other hand, the output side has an inductive nature so an output phase must never be open circuit. These two constraints can be expressed by

$$S_{Aj} + S_{Bj} + S_{Cj} = 1$$

$$S_{ij} = \begin{cases} 1 & switch\, S_{ij}\ closed \\ 0 & switch\, S_{ij}\ open \end{cases} \qquad 2.1$$

where i= {A, B, C}

j= {a, b, c}

With these restrictions, the 3×3 matrix converter has 27 possible switch states. These states are summarised in table 2.1.

The source and load voltages can be expressed as vectors defined by

$$v_i = \begin{bmatrix} v_A(t) \\ v_B(t) \\ v_C(t) \end{bmatrix} \qquad v_o = \begin{bmatrix} v_a(t) \\ v_b(t) \\ v_c(t) \end{bmatrix} \qquad 2.2$$

The relationship between the output and input voltages can be expressed as

$$\begin{bmatrix} v_a(t) \\ v_b(t) \\ v_c(t) \end{bmatrix} = \begin{bmatrix} S_{Aa} & S_{Ba} & S_{Ca} \\ S_{Ab} & S_{Bb} & S_{Cb} \\ S_{Ac} & S_{Bc} & S_{Cc} \end{bmatrix} \begin{bmatrix} v_A(t) \\ v_B(t) \\ v_C(t) \end{bmatrix} \qquad 2.3$$

$$v_o = T \cdot v_i$$

In the same form, the relationship between input and output currents is expressed by

$$i_i = T^T . i_o \qquad 2.4$$

where

$$i_i = \begin{bmatrix} i_A(t) \\ i_B(t) \\ i_C(t) \end{bmatrix} \qquad i_o = \begin{bmatrix} i_a(t) \\ i_b(t) \\ i_c(t) \end{bmatrix}$$

ON switches	Output Voltages			Input currents			Voltage vector		Current vector	
	V_{ab}	V_{bc}	V_{ca}	I_A	I_B	I_C	Magn.	Phase	Magn.	Phase
$S_{Aa}S_{Bb}S_{Cc}$	V_{AB}	V_{BC}	V_{CA}	I_a	I_b	I_c	V_{il}	$\omega_i t$	I_o	$\omega_o t$
$S_{Aa}S_{Cb}S_{Bc}$	$-V_{CA}$	$-V_{BC}$	$-V_{AB}$	I_a	I_c	I_b	V_{il}	$-\omega_i t+4\pi/3$	I_o	$-\omega_o t$
$S_{Ba}S_{Ab}S_{Cc}$	$-V_{AB}$	$-V_{CA}$	$-V_{BC}$	I_b	I_a	I_c	V_{il}	$-\omega_i t$	I_o	$-\omega_o t+2\pi/3$
$S_{Ba}S_{Cb}S_{Ac}$	V_{BC}	V_{CA}	V_{AB}	I_c	I_a	I_b	V_{il}	$\omega_i t+4\pi/3$	I_o	$\omega_o t+2\pi/3$
$S_{Ca}S_{Ab}S_{Bc}$	V_{CA}	V_{AB}	V_{BC}	I_b	I_c	I_a	V_{il}	$\omega_i t+2\pi/3$	I_o	$\omega_o t+4\pi/3$
$S_{Ca}S_{Bb}S_{Ac}$	$-V_{BC}$	$-V_{AB}$	$-V_{CA}$	I_c	I_b	I_a	V_{il}	$-\omega_i t+2\pi/3$	I_o	$-\omega_o t+4\pi/3$
$S_{Aa}S_{Bb}S_{Bc}$	V_{AB}	0	$-V_{AB}$	I_a	$-I_a$	0	kV_{AB}	$\pi/6$	kI_a	$-\pi/6$
$S_{Ba}S_{Ab}S_{Ac}$	$-V_{AB}$	0	V_{AB}	$-I_a$	I_a	0	kV_{AB}	$-5\pi/6$	kI_a	$5\pi/6$
$S_{Ba}S_{Cb}S_{Cc}$	V_{BC}	0	$-V_{BC}$	0	I_a	$-I_a$	kV_{BC}	$\pi/6$	kI_a	$\pi/2$
$S_{Ca}S_{Bb}S_{Bc}$	$-V_{BC}$	0	V_{BC}	0	$-I_a$	I_a	kV_{BC}	$-5\pi/6$	kI_a	$-\pi/2$
$S_{Ca}S_{Ab}S_{Ac}$	V_{CA}	0	$-V_{CA}$	$-I_a$	0	I_a	kV_{CA}	$\pi/6$	kI_a	$-5\pi/6$
$S_{Aa}S_{Cb}S_{Cc}$	$-V_{CA}$	0	V_{CA}	I_a	0	$-I_a$	kV_{CA}	$-5\pi/6$	kI_a	$\pi/6$
$S_{Ba}S_{Ab}S_{Bc}$	$-V_{AB}$	V_{AB}	0	I_b	$-I_b$	0	kV_{AB}	$5\pi/6$	kI_b	$-\pi/6$
$S_{Ba}S_{Bb}S_{Ac}$	V_{AB}	$-V_{AB}$	0	$-I_b$	I_b	0	kV_{AB}	$-\pi/6$	kI_b	$5\pi/6$
$S_{Ca}S_{Bb}S_{Cc}$	$-V_{BC}$	V_{BC}	0	0	I_b	$-I_b$	kV_{BC}	$5\pi/6$	kI_b	$\pi/2$
$S_{Ba}S_{Cb}S_{Bc}$	V_{BC}	$-V_{BC}$	0	0	$-I_b$	I_b	kV_{BC}	$-\pi/6$	kI_b	$-\pi/2$
$S_{Aa}S_{Cb}S_{Ac}$	$-V_{CA}$	V_{CA}	0	$-I_b$	0	I_b	kV_{CA}	$5\pi/6$	kI_b	$-5\pi/6$
$S_{Ca}S_{Ab}S_{Cc}$	V_{CA}	$-V_{CA}$	0	I_b	0	$-I_b$	kV_{CA}	$-\pi/6$	kI_b	$\pi/6$
$S_{Ba}S_{Bb}S_{Ac}$	0	$-V_{AB}$	V_{AB}	I_c	$-I_c$	0	kV_{AB}	$-\pi/2$	kI_c	$-\pi/6$
$S_{Aa}S_{Ab}S_{Bc}$	0	V_{AB}	$-V_{AB}$	$-I_c$	I_c	0	kV_{AB}	$\pi/2$	kI_c	$5\pi/6$
$S_{Ca}S_{Cb}S_{Bc}$	0	$-V_{BC}$	V_{BC}	0	I_c	$-I_c$	kV_{BC}	$-\pi/2$	kI_c	$\pi/2$
$S_{Ba}S_{Bb}S_{Cc}$	0	V_{BC}	$-V_{BC}$	0	$-I_c$	I_c	kV_{BC}	$\pi/2$	kI_c	$-\pi/2$
$S_{Aa}S_{Ab}S_{Cc}$	0	$-V_{CA}$	V_{CA}	$-I_c$	0	I_c	kV_{CA}	$-\pi/2$	kI_c	$-5\pi/6$
$S_{Ca}S_{Cb}S_{Ac}$	0	V_{CA}	$-V_{CA}$	I_c	0	$-I_c$	kV_{CA}	$\pi/2$	kI_c	$\pi/6$
$S_{Aa}S_{Ab}S_{Ac}$	0	0	0	0	0	0	0	0	0	0
$S_{Ba}S_{Bb}S_{Bc}$	0	0	0	0	0	0	0	0	0	0
$S_{Ca}S_{Cb}S_{Cc}$	0	0	0	0	0	0	0	0	0	0

Table 2.1 Possible switch states of the matrix converter

The duty cycle $m_{ij}(t)$ of the switch S_{ij} is defined by

$$m_{ij} = \frac{t_{ij}}{T_s} \qquad 0 \le m_{ij} \le 1 \tag{2.5}$$

The low frequency transfer matrix $M(\omega_m t)$ is defined by

$$M(\omega_m t) = \begin{bmatrix} m_{Aa}(\omega_m t) & m_{Ba}(\omega_m t) & m_{Ca}(\omega_m t) \\ m_{Ab}(\omega_m t) & m_{Bb}(\omega_m t) & m_{Cb}(\omega_m t) \\ m_{Ac}(\omega_m t) & m_{Bc}(\omega_m t) & m_{Cc}(\omega_m t) \end{bmatrix} \tag{2.6}$$

The low frequency (fundamental) component of the output phase voltages is given by

$$v_o(t) = M(\omega_m t).v_i(t) \tag{2.7}$$

and the low frequency component of the input current is

$$i_i(t) = (M(\omega_m t))^T . i_o(t) \tag{2.8}$$

Given a set of sinusoidal input voltages and assuming a set of output currents as follows:

$$v_i(t) = V_i \begin{bmatrix} \cos\omega_i t \\ \cos\left(\omega_i t + {}^{2\pi}/_{3}\right) \\ \cos\left(\omega_i t + {}^{4\pi}/_{3}\right) \end{bmatrix} \tag{2.9}$$

$$i_o(t) = I_o \begin{bmatrix} \cos(\omega_o t + \phi_o) \\ \cos\left(\omega_o t + \phi_o + {}^{2\pi}/_{3}\right) \\ \cos\left(\omega_o t + \phi_o + {}^{4\pi}/_{3}\right) \end{bmatrix} \tag{2.10}$$

Then

$$v_o(t) = qV_i \begin{bmatrix} \cos\omega_o t \\ \cos\left(\omega_o t + {}^{2\pi}/_{3}\right) \\ \cos\left(\omega_o t + {}^{4\pi}/_{3}\right) \end{bmatrix} \tag{2.11}$$

$$i_i(t) = q\frac{\cos\phi_o}{\cos\phi_i} I_o \begin{bmatrix} \cos(\omega_i t + \phi_i) \\ \cos\left(\omega_i t + \phi_i + {}^{2\pi}/_{3}\right) \\ \cos\left(\omega_i t + \phi_i + {}^{4\pi}/_{3}\right) \end{bmatrix} \tag{2.12}$$

where q is the voltage transfer ratio.

The modulation problem is how to drive the modulation matrix $M(\omega_m t)$ to produce a sinusoidal output voltage and a sinusoidal input current.

2.1.2 Modulation Algorithm

There are many control algorithms to solve the modulation problem. Generally they can be divided into two types of control. One is termed the direct modulation method and the other is the indirect modulation method.

i. Direct modulation algorithms

Two main solutions to this problem were derived by Venturini [2.2]-[2.4]:

$$M_1(t)=\begin{bmatrix} m_1 & m_2 & m_3 \\ m_3 & m_1 & m_2 \\ m_2 & m_3 & m_1 \end{bmatrix} \quad with\ \omega_m=(\omega_o-\omega_i) \tag{2.13}$$

$$M_2(t)=\begin{bmatrix} m_1 & m_3 & m_2 \\ m_2 & m_1 & m_3 \\ m_3 & m_2 & m_1 \end{bmatrix} \quad with\ \omega_m=-(\omega_o-\omega_i) \tag{2.14}$$

$m_i=\frac{1}{3}(1+2q\cos(\omega_m t-\theta_i))$, where $\theta_i=0,\ \frac{2\pi}{3},\ \frac{4\pi}{3}$ for i=1, 2, 3 respectively

The solution in 2.13 corresponds to when the output displacement angle ϕ_o is equal to the input displacement angle ϕ_i, and the other solution, equation 2.14, is related to $\phi_o=-\phi_i$. Combining the two solutions yields a general form in which the input displacement angle can be controlled using different weights $\alpha_1$and $\alpha_2$for M_1 and M_2

$$M(t)=\alpha_1 M_1(t)+\alpha_2 M_2(t) \tag{2.15}$$

For unity input displacement factor, the matrix elements expression can be reduced to the form

$$m_{Kj}=\frac{1}{3}\left[1+\frac{2v_K v_j}{v_{im}^2}\right] \tag{2.16}$$

where K= A, B, C, j = a, b, c and v_{im} is the maximum input voltage.

Using these solutions, the maximum voltage transfer ratio that can be achieved is 50%. This is demonstrated in fig. 2.2.

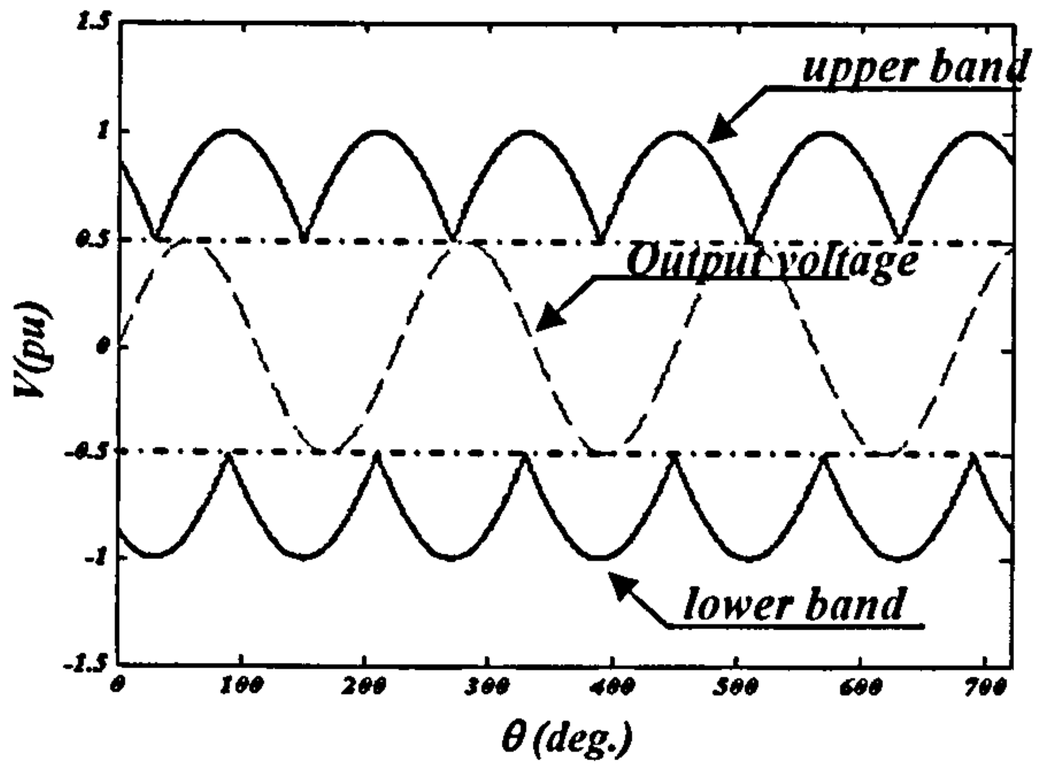


Fig.2.2 Sinusoidal output phase voltage within the inner bounds of the input voltage envelope

This ratio can be improved to 86.6% if third harmonic components are injected into the output voltage. In this case the target output voltages will be

$$v_o(t) = qV_i \begin{bmatrix} \cos(\omega_o t) - \frac{1}{6}\cos(3\omega_o t) + \frac{1}{2\sqrt{3}}\cos(3\omega_i t) \\ \cos(\omega_o t + 2\pi/3) - \frac{1}{6}\cos(3\omega_o t) + \frac{1}{2\sqrt{3}}\cos(3\omega_i t) \\ \cos(\omega_o t + 4\pi/3) - \frac{1}{6}\cos(3\omega_o t) + \frac{1}{2\sqrt{3}}\cos(3\omega_i t) \end{bmatrix} \quad 2.18$$

Injecting third harmonic components has no effect on the output line to line voltage, but allows the output phase voltage to fit within the input voltage envelope as shown in fig.2.3. In addition, this 86% transfer ratio can only be achieved at unity displacement factor. Varying this factor will reduce the voltage transfer ratio.

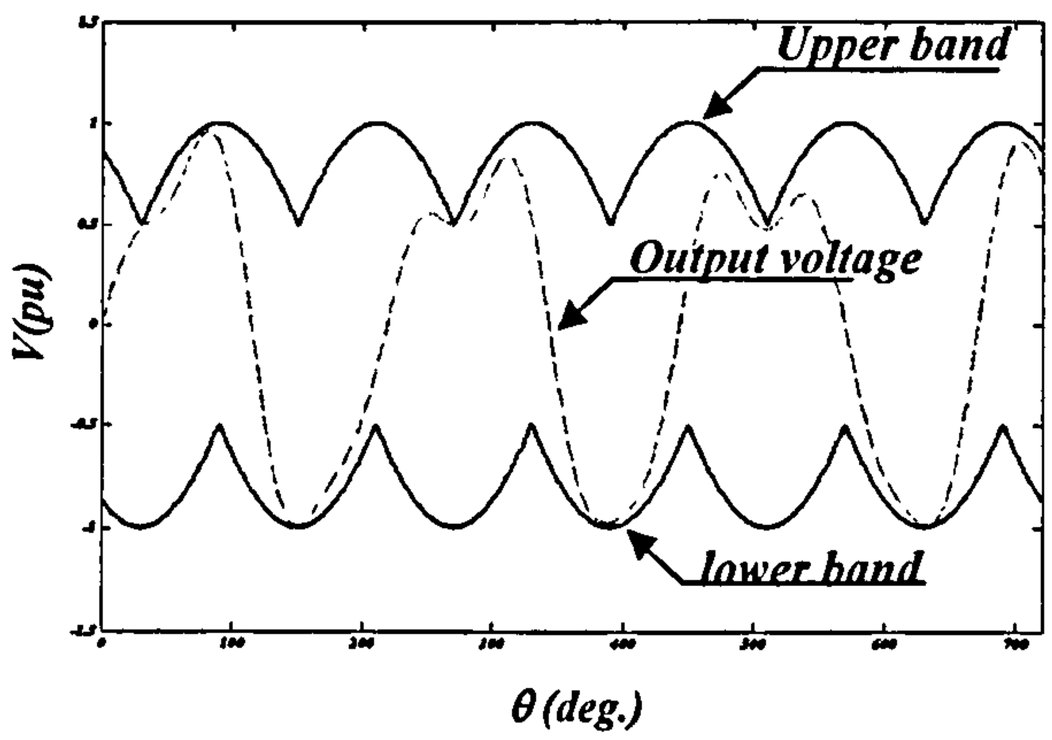


Fig.2.3 Output phase voltage after injecting third harmonic components, within the outer envelope of the input voltage envelope

In this case, the solution of the modulation problem, including displacement angle control, can be expressed in the form of equation 2.19 [2.4]

$$
\begin{aligned}
m_{Kj} = \frac{1}{3}\Bigg\{ & 1 + |1+\theta| q \cos\left((\omega_o - \omega_i)t - \frac{2(X-y)\pi}{3} \right) \\
& + |1-\theta| q \cos\left((\omega_i + \omega_o)t - \frac{2\pi(X+y-2)}{3} \right) \\
& + q\left(\frac{-1}{6\sqrt{3}} \cos\left(4\omega_i t - \frac{2\pi(y-1)}{3} \right) + \frac{7}{6\sqrt{3}} \cos\left(2\omega_i t - \frac{2\pi(1-y)}{3} \right) \right) \\
& - \frac{q}{6}\left(\cos\left((3\omega_o + \omega_i)t - \frac{2\pi(y-1)}{3} \right) + \cos\left((3\omega_o - \omega_i)t - \frac{2\pi(1-y)}{3} \right) \right) \Bigg\}
\end{aligned}
\qquad 2.19
$$

where $\theta = \dfrac{\tan\phi_i}{\tan\phi_o}$

X=1, 2, 3 for K=A, B, C respectively

y=1, 2, 3 for j=a, b, c respectively

This expression is complicated; the modulation matrix elements will be dependant on the load power factor angle, which reduces its applicability. These are the main disadvantages of this method.

If a unity displacement angle is required, equation 2.19 can be reduced to [2.4]:

$$
m_{Kj} = \frac{1}{3}\left[1 + \frac{2v_K v_j}{v_{im}^2} + \frac{4q}{3\sqrt{3}} \sin(\omega_i t + \beta_K) \sin(3\omega_i t) \right] \qquad 2.20
$$

$\beta_K = 0, {}^{2\pi}/_{3}, {}^{4\pi}/_{3}$

where j = a, b, c and K = A, B, C

For the direct method, three signals are compared with a carrier in the order: m_{Aj}, $m_{Aj}+m_{Bj}$, and $m_{Aj}+m_{Bj}+m_{Cj}$ to produce the output matrix sequences as in fig.2.4.

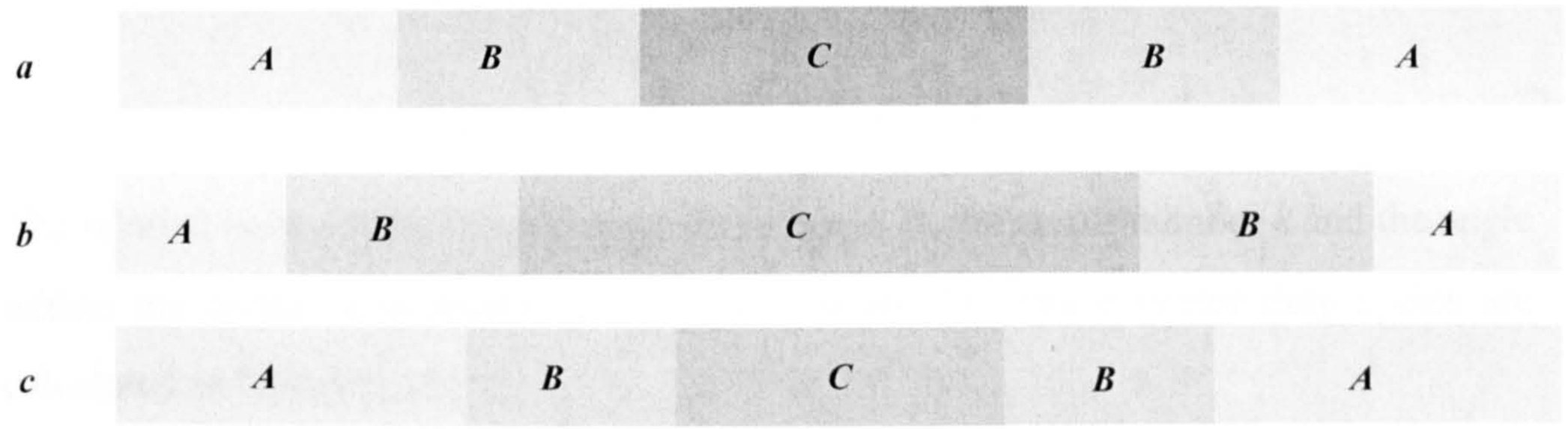


Fig.2.4 Switching sequence for the direct method

ii. Indirect Modulation Algorithm

With indirect space vector modulation [2.6]-[2.10], the matrix converter function can be described as an active rectifier and a cascaded inverter. In this case, all the quantities are referred to a virtual DC link as shown in fig. 2.5. The modulation problem in this case can be separated into two independent stages: a rectifier stage represented by switches S_1 to S_6 and an inverter stage represented by switches S_7 to S_{12}.

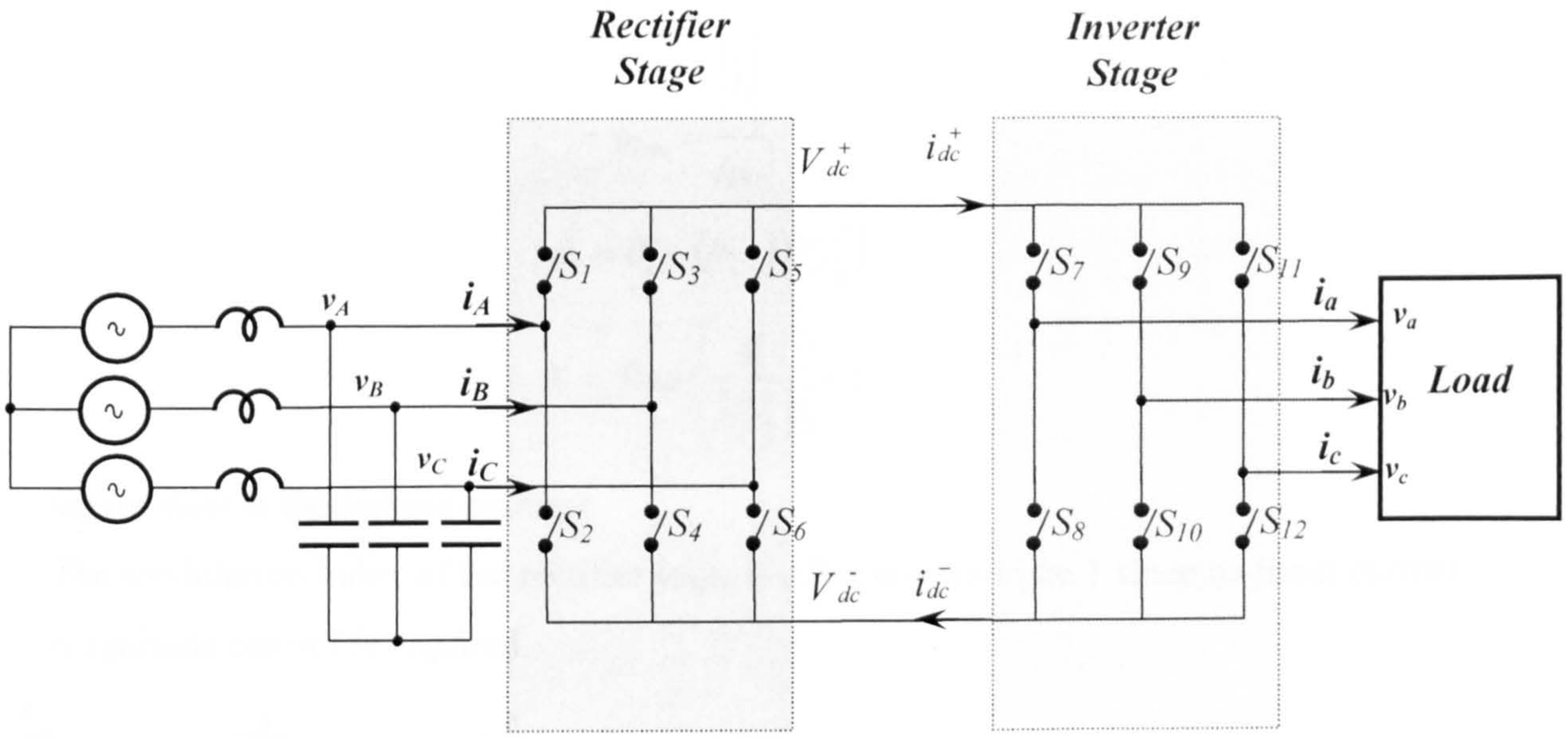


Fig.2.5 Arrangement of the matrix converter as virtual back to back converters

a- Space vector for the rectifier stage

The input voltages and currents can be expressed in space vector forms using the transforms in equations 2.21 and 2.22 [2.9]

$$\bar{V}_i = \frac{2}{3}\left(v_A + v_B \cdot e^{-j\frac{2\pi}{3}} + v_C \cdot e^{-j\frac{4\pi}{3}} \right) \qquad 2.21$$

$$\bar{I}_i = \frac{2}{3}\left(i_A + i_B \cdot e^{-j\frac{2\pi}{3}} + i_C \cdot e^{-j\frac{4\pi}{3}} \right) \qquad 2.22$$

The relation between the input current phase angle θ_i, the sector number k and the angle within the sector δ_i is shown in fig. 2.6. The rectifier space vector duty cycles are calculated as follows [2.9]:

$$m_\alpha = m_{rec}.sin\left(\frac{\pi}{3} - \delta_i\right)$$

$$m_\beta = m_{rec}.sin\,\delta_i$$

$$m_0 = 1 - m_\alpha - m_\beta \qquad 2.23$$

$$m_{rec} = \frac{\left|\overline{I_i}\right|}{i_{DC}}$$

$$\delta_i = \theta_i - (k-1)(\pi/3)$$

$$k = floor\left(\frac{\theta_i}{(\pi/3)}\right) + 1$$

where *floor* is the integer quotient.

The modulation index of the rectifier stage is often chosen to be 1 since no input current magnitude control is required.

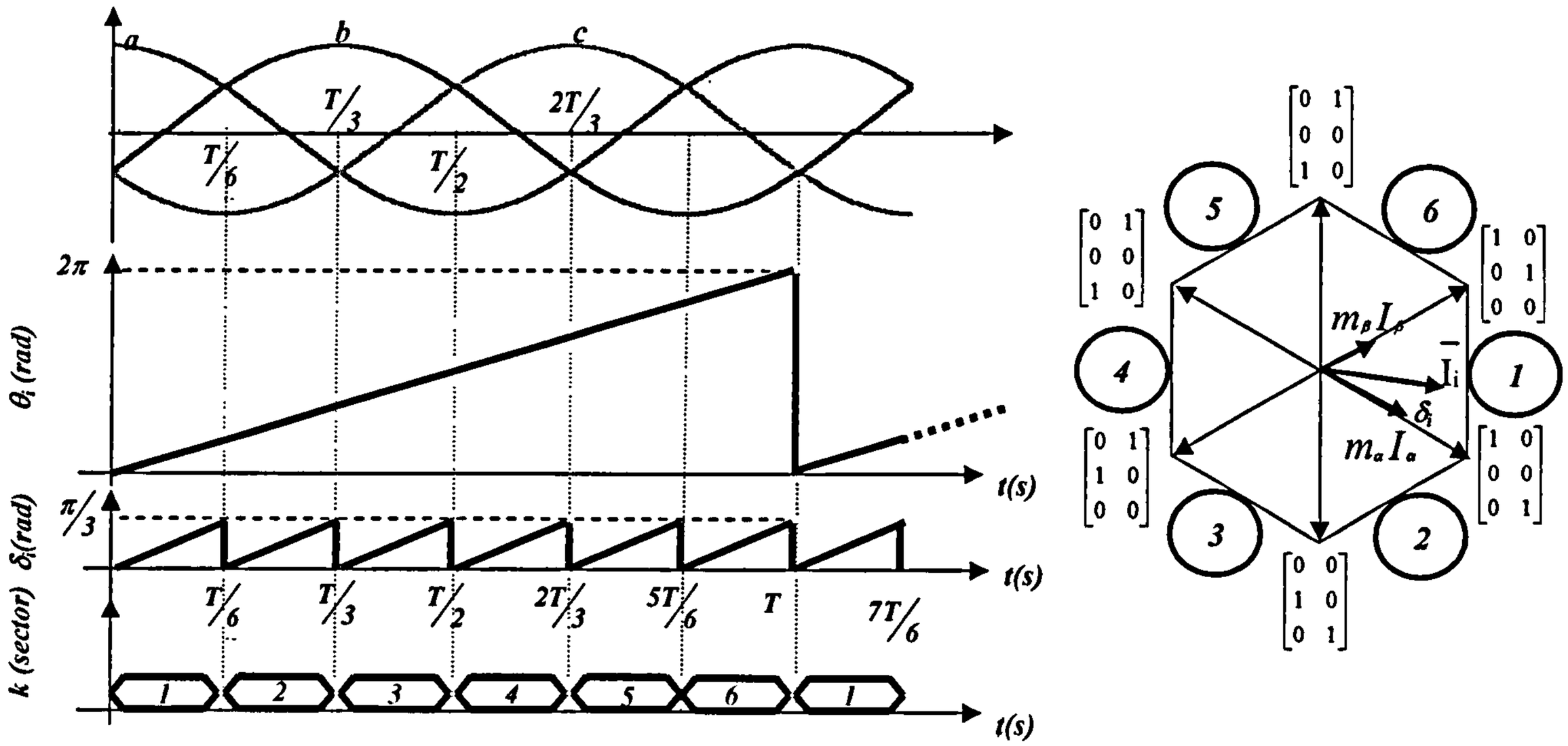


Fig.2.6 Rectifier stage space vector.

b- Space vector for the inverter stage

The output voltages and currents can be expressed in space vector forms using the transforms in equations 2.24 and 2.25 (which are similar to equations 2.21 and 2.22)

$$\bar{V}_o = \frac{2}{3}\left(v_a + v_b.e^{-j\frac{2\pi}{3}} + v_c.e^{-j\frac{4\pi}{3}}\right) \qquad 2.24$$

$$\bar{I}_o = \frac{2}{3}\left(i_a + i_b.e^{-j\frac{2\pi}{3}} + i_c.e^{-j\frac{4\pi}{3}}\right) \qquad 2.25$$

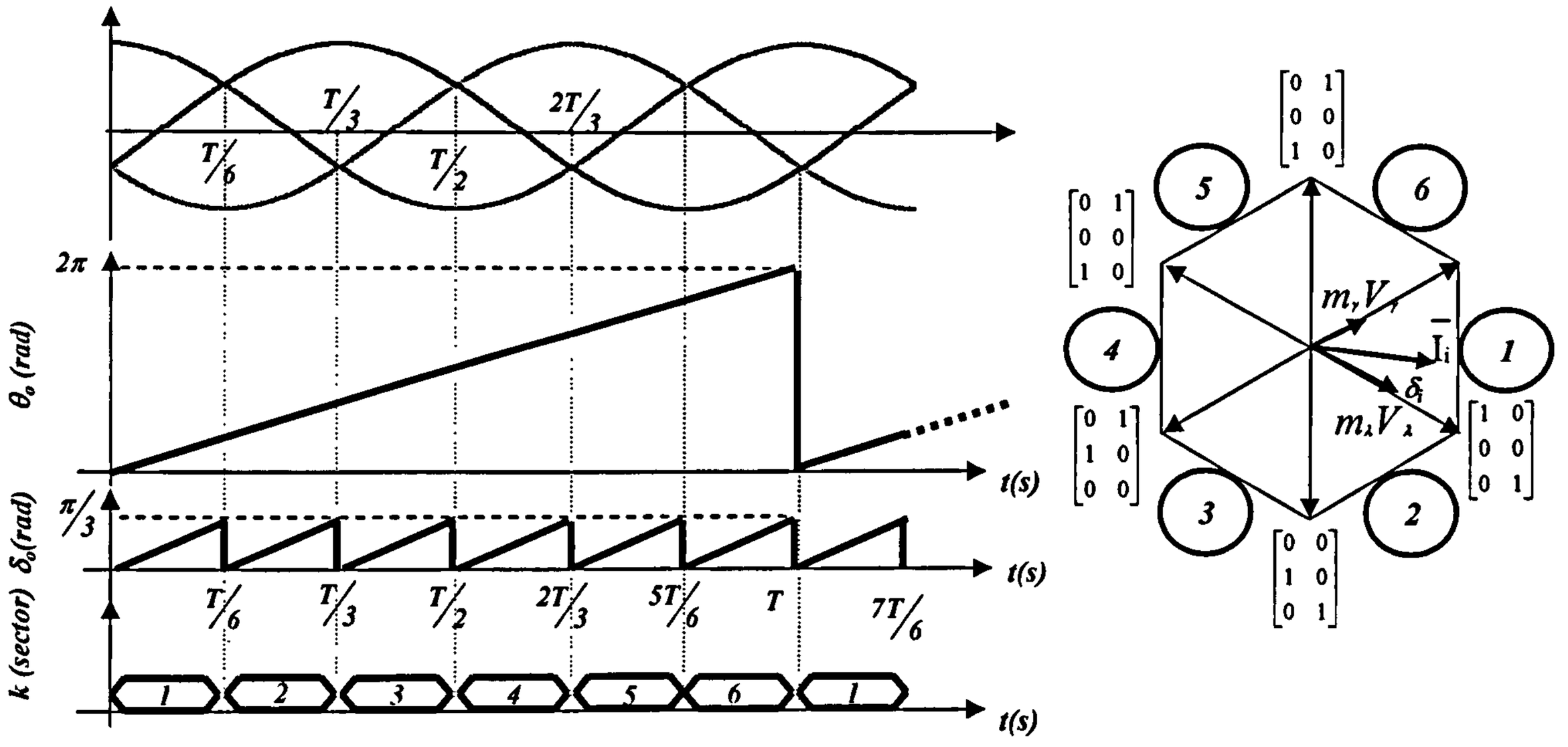


Fig.2.7 Inverter stage space vector

$$\begin{aligned} m_\gamma &= m_{inv}.sin\left(\frac{\pi}{3} - \delta_o\right) \\ m_\lambda &= m_{inv}.sin\,\delta_o \\ m_0 &= 1 - m_\gamma - m_\lambda \end{aligned} \qquad 2.26$$

$$m_{inv} = \frac{\left|\bar{V}_o\right|}{v_{DC}}$$

The modulation index of the inverter stage should be less than 0.86, this being the maximum voltage transfer ratio.

The overall duty cycles will be:

$$m_\gamma.m_\alpha = m_{inv}.sin\left(\frac{\pi}{3} - \delta_o\right).sin\left(\frac{\pi}{3} - \delta_i\right)$$

$$m_\lambda \cdot m_\alpha = m_{inv} . \sin \delta_o . \sin\left(\frac{\pi}{3} - \delta_i\right)$$

$$m_\gamma \cdot m_\beta = m_{inv} . \sin\left(\frac{\pi}{3} - \delta_o\right) . \sin \delta_i$$

$$m_\lambda \cdot m_\beta = m_{inv} . \sin \delta_o . \sin \delta_i$$

$$m_0 = 1 - \left(m_\alpha + m_\beta\right)\left(m_\gamma + m_\lambda\right)$$

$$m_{inv} = \frac{\left|\overline{V_o}\right|}{v_{DC}}$$

For the indirect method, the order of the four active vectors within the switching period and which of the zero vectors are used is the next procedural step. From the possible combinations, the state which limits switching to one transition during each state change is usually used in order to minimize total switching losses. Also, the zero vector is selected using a criteria where the number of 'Branch Switch Overs' (BSO) in the matrix converter is minimized [2.11].

- When the sum of the current and voltage hexagon sectors is odd, the output vector sequence is $m_\alpha m_\gamma \rightarrow m_\alpha m_\lambda \rightarrow m_\beta m_\lambda \rightarrow m_\beta m_\gamma \rightarrow m_0$
- When the sum of the current and voltage hexagon sectors is even, the output vector sequence is $m_\alpha m_\lambda \rightarrow m_\alpha m_\gamma \rightarrow m_\beta m_\gamma \rightarrow m_\beta m_\lambda \rightarrow m_0$
- When the input hexagon sector is odd, the output zero vector is *000*. Otherwise *111*.

The switching sequence for optimized indirect SVM is listed in Table 2.2 and fig. 2.8. The output voltage and input current vectors both lie in sector 1.

$\alpha\lambda$	$\alpha\gamma$	$\beta\gamma$	$\beta\lambda$	*0*	$\beta\lambda$	$\beta\gamma$	$\alpha\gamma$	$\alpha\lambda$
ABB	*ABA*	*ACA*	*ACC*	*CCC*	*ACC*	*ACA*	*ABA*	*ABB*
$\frac{1}{2}t_{\alpha\lambda}$	$\frac{1}{2}t_{\alpha\gamma}$	$\frac{1}{2}t_{\beta\gamma}$	$\frac{1}{2}t_{\beta\lambda}$	T_0	$\frac{1}{2}t_{\beta\lambda}$	$\frac{1}{2}t_{\beta\gamma}$	$\frac{1}{2}t_{\alpha\gamma}$	$\frac{1}{2}t_{\alpha\lambda}$

Table 2.2 Switching sequence in current sector = *1*, voltage sector = *1* $\left(t_{ij} = m_i m_j T_s\right)$

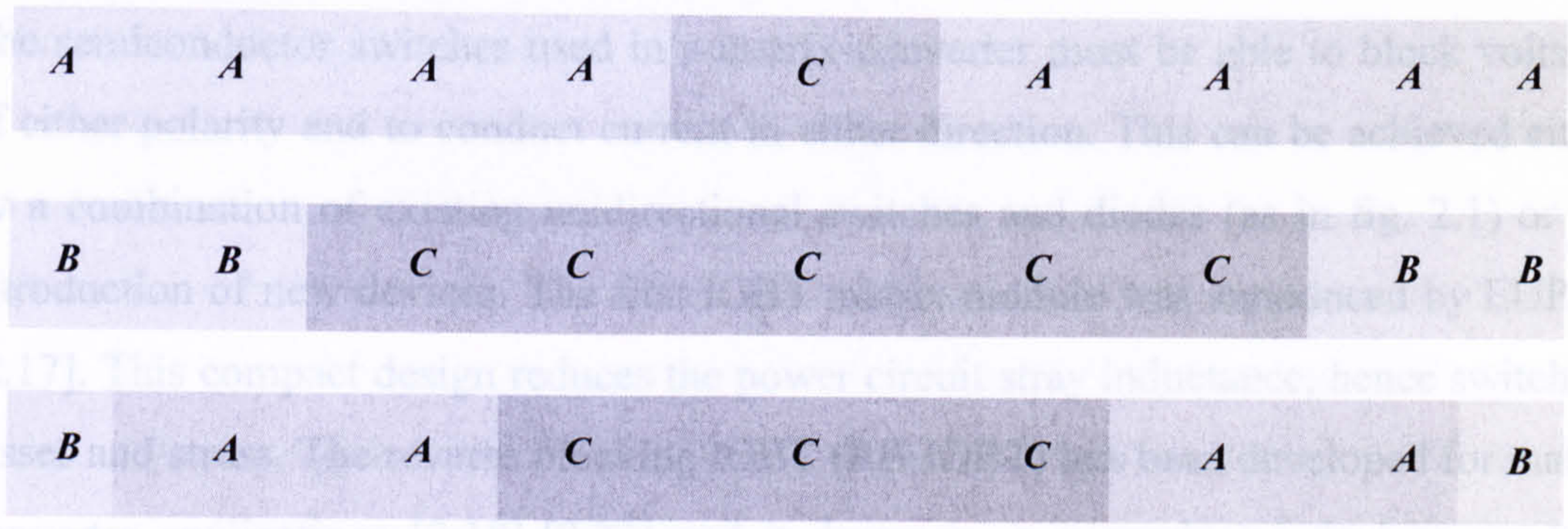


Fig. 2.8 Switching sequence in current sector = *1*, voltage sector = *1*

2.1.3 Switch commutation

The matrix converter arrangement suffers from a lack of free-wheeling paths for the inductive load current. This makes the commutation process different to that of the normal voltage source inverter. To achieve safe commutation, a precise sequence must be followed in order to prevent either short circuiting the input terminals or open circuiting the load current path. A technique that complies with these two rules was proposed in [2.12] and named semi-soft commutation or four-step commutation, which means that half of the switching during any sampling period is performed naturally. A different approach based on the relative magnitudes of the converter input voltages was also used but it does not reveal any semi-soft switching properties [2.13]. Two-step commutation based on the input voltage magnitude was first introduced in [2.14] in order to minimize the number of commutation steps and to reduce the complexity of the commutation procedure. However, these commutation methods depend completely on the measurement accuracy of either the output current or the input voltage, and may lead to commutation failure, particularly at low V-I magnitudes. Other ideas have been proposed to eliminate the polarity detection circuit, thereby reducing converter costs. Gate drive intelligence has been used to indicate the current direction [2.15]. Another approach was to employ a dual bridge matrix converter together with a new modulation technique [2.16].

2.1.4 Switching devices

The semiconductor switches used in a matrix converter must be able to block voltages of either polarity and to conduct current in either direction. This can be achieved either by a combination of existing unidirectional switches and diodes (as in fig. 2.1) or the introduction of new devices. The first IGBT matrix module was announced by EUPEC [2.17]. This compact design reduces the power circuit stray inductance, hence switching losses and stress. The reverse blocking IGBT (RB-IGBT) has been developed for matrix converter applications [2.18]-[2.23] and it decreases semiconductor devices per load phase path from four to two by eliminating the diodes. This may increase the efficiency of the matrix converters to above that of the diode-bridge VSI, because conduction losses are produced only by a single RB- IGBT per phase.

2.1.5 Input voltages disturbance

Due to the absence of a DC link capacitor, matrix converter performance is affected by unbalance and distortion of the input voltage system. The matrix converter generates low-order harmonics in the output voltage when unbalanced supply voltages are present due to input grid condition [2.24]. Therefore research has been directed to compensate the influence of any input voltage disturbance. Balanced sinusoidal output voltages can be generated with unbalanced input voltages by modifying the modulation index in the inversion stage according to the virtual dc-link voltage, which can be estimated from the instantaneous line voltages. The same approach was extended to the Veturini modulation method under three-phase simultaneous voltage sag or swells [2.25]. The input current harmonic content and the voltage transfer ratio limits were determined analytically for different operational conditions [2.26]-[2.28]. Also the input current harmonic content is dependent on the input current modulation strategy employed to control the matrix converter.

In [2.29], the input imbalanced problem was overcome by taking into account the input/output power balance equation. It was implemented in a max-mid-min modulation algorithm which improved the input current quality without output performance degradation. In [2.30], a direct feed-forward unbalanced control method was proposed

for a dual-bridge matrix converter topology. The input voltages are detected and then used to adjust the switching function of the line side converter. However, the input currents are slightly distorted and the method is effective only if the locus of the output voltage vector fits within the input voltage outer locus.

2.1.6 Protection and diode clamping circuitry

The matrix converter needs protection against overvoltage and overcurrent. A clamp circuit first used for protection purposes consisted of a fast recovery diode rectifier bridge and a capacitor, which provided safe converter shutdown during overcurrent on the output side or voltage disturbances on the input side [2.31], [2.32]. Another approach was to dissipate the energy of inductive currents in varistors and in the semiconductors by employing active gate drives [2.33].

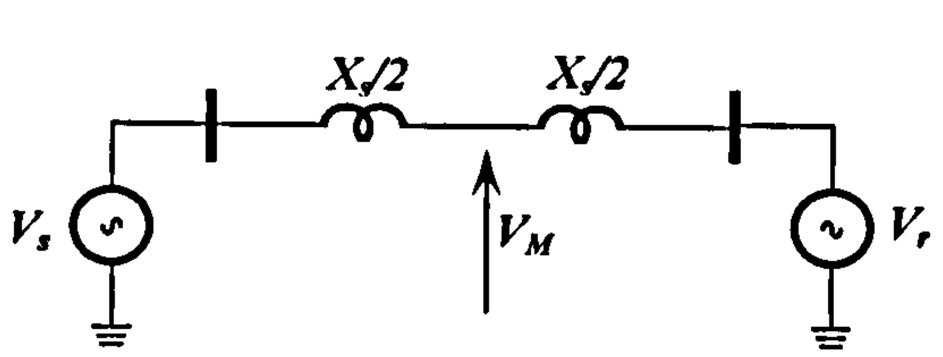
2.2 Flexible AC Transmission Systems (FACTS)

Developed by Hingorani [2.34], FACTS devices are based on power electronic controllers that improve the capacity of transmission lines. These devices control the transferred power through a transmission line by appropriately compensating network parameters, such as line series impedance, line shunt impedance, current, voltage, and real and reactive power. FACTs allow transmission lines to operate much closer to their thermal limits.

2.2.1 Basic relationships for power transmission

This review of the basic relationships for power transmission, for brevity, is limited to the two-machine model shown in Figure 2.9. This model includes the sending-end generator with voltage phasor V_s, the receiving-end generator with voltage phasor V_R, and the transmission line reactance X_s (neglecting the resistance part of the impedance for simplicity). The magnitudes of the sending and the receiving end voltages are assumed equal to V and the phase shift between the two phasors is δ. The power transmitted between the sending-end and receiving-end generators is then obtained from the relationship:

$$P = \frac{V^2}{X_s} \sin\delta \qquad 2.28$$



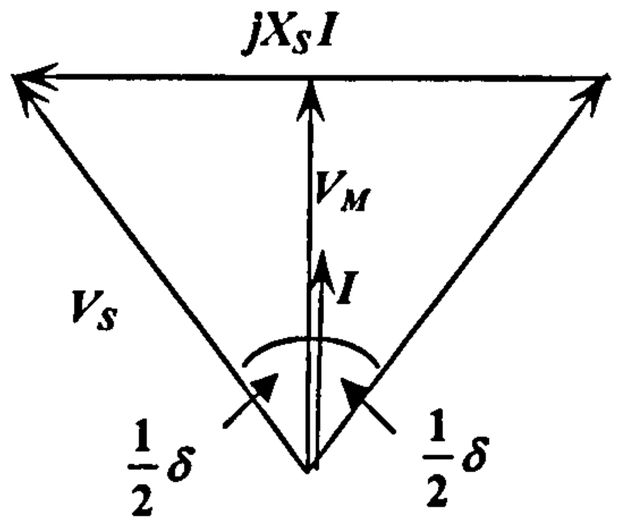


Fig. 2.9 Simple power system model

As equation 2.28 indicates, the transmitted power is a function of:

- the transmission line impedance;
- the magnitude of the sending and receiving end voltages; and
- the phase angle between these voltages.

The objective of dynamic transmission system compensation is to control these parameters. The generalized power flow controller is an ideal device capable of varying the three transmission parameters (voltage, impedance, and phase angle) by appropriate series reactive compensation, shunt reactive compensation, and phase shifting.

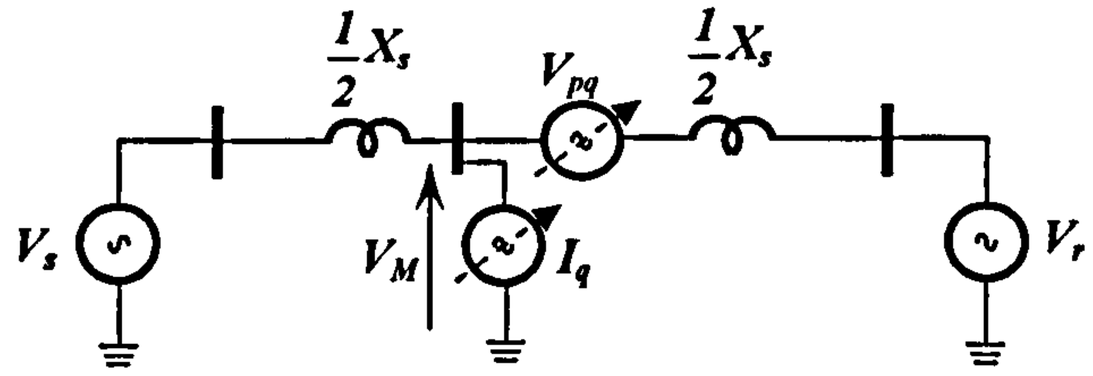


Fig.2.10 Simple power system model including the generalized power flow controller

2.2.2 Types of FACTS controllers

FACTS controllers can be categorized into four groups:

- series controllers;
- shunt controllers;
- phase angle shifters controllers; and
- combined series-shunt controllers.

i. Principles of Series Controllers

A series controller may be regarded as variable capacitive reactance inserted at the transmission line mid point in order to reduce the overall line reactance[2.33]-[2.35]. This is achieved by injecting an appropriate voltage phasor in series with the mid-line point of the transmission line. The degree of series compensation is defined by the coefficient *k (0<k<1)*.

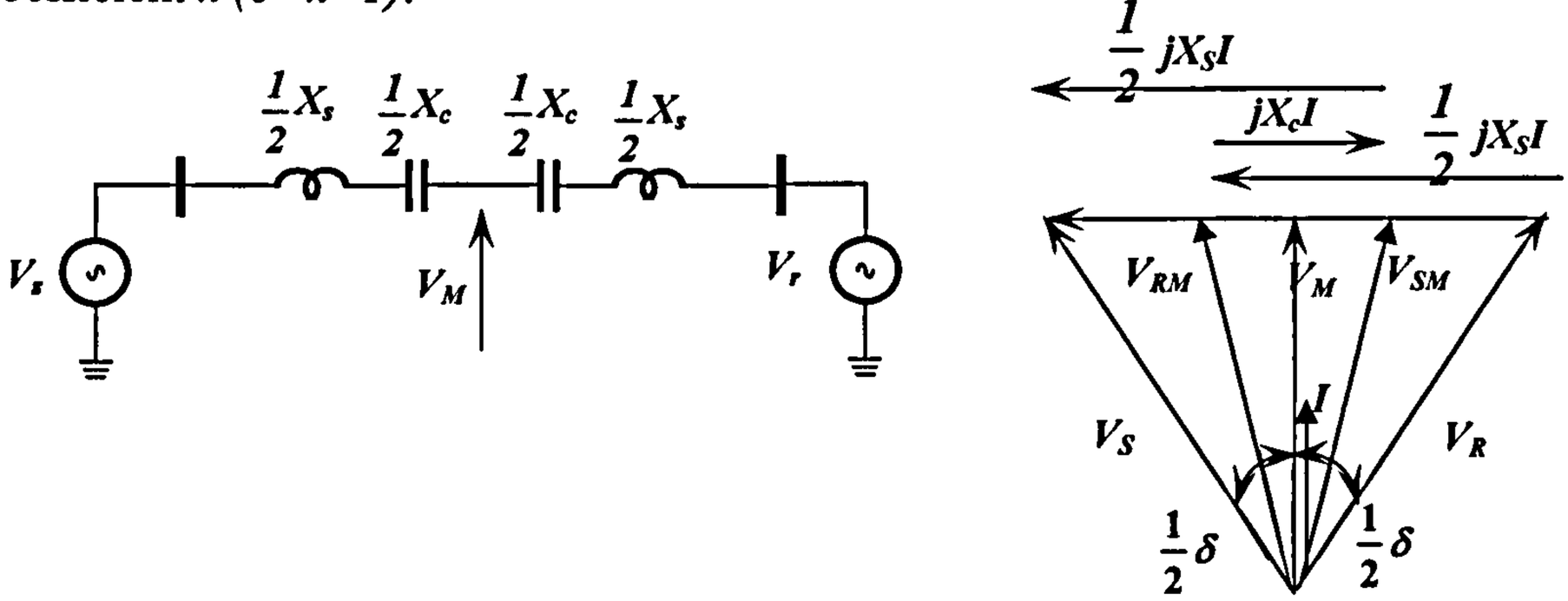


Fig. 2.11 Simple power system model with series capacitive compensation

From the phasor diagram shown in fig. 2.11, the series capacitive compensation leads to an increment in the current flow through the line, hence an increment in the power flow. The P versus δ relationship becomes:

$$P_2 = \frac{V^2}{X_s(1-k)} \sin\delta \qquad 2.29$$

Examples of such controllers are the Static Synchronous Series Compensator (SSSC), the Thyristor-Switched Series Capacitor (TSSC), and the Thyristor-Controlled Series Reactor (TCSR), to cite a few. They can be effectively used to control current and power flow in the system and to damp system oscillations.

ii. Principles of Shunt Controllers

In order to obtain a flat voltage profile, or in other words, a constant voltage magnitude along the transmission line, it is necessary to introduce shunt capacitive compensation distributed along the line to compensate the voltage drop at every point. This is impractical. However, the usual method is to provide compensation at the mid point. Shunt controllers inject current into the system at the connection point [2.33]-[2.35]. If the injected current is in phase quadrature with the line voltage, the controller adjusts

reactive power. This could be implemented by assuming that $V_{pq} = 0$ and $I_q = -j (4V/X_s)(1-\cos \frac{1}{2}\delta)$.

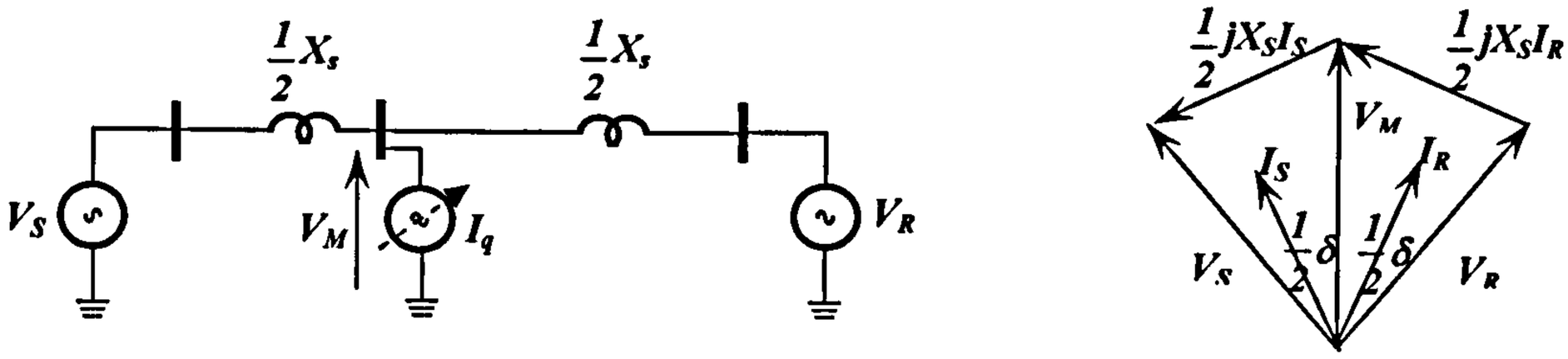


Fig. 2.12 Simple power system model for shunt compensation

For this case of ideal mid-point compensation, the P versus δ relationship can be written as:

$$P_3 = 2\frac{V^2}{X_s}\sin\frac{1}{2}\delta \qquad 2.30$$

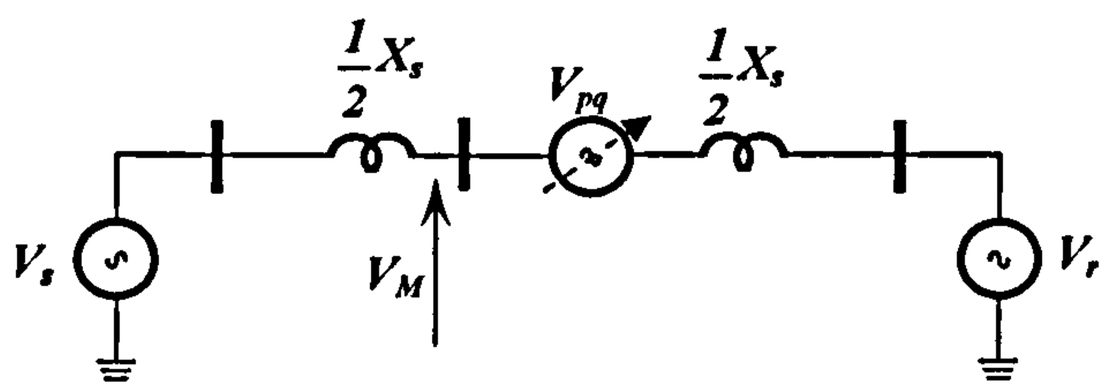
Examples of such systems are the Static Synchronous Generator (SSG) and the Static VAr Compensator (SVC). They can be used as an effective way to control the voltage at and around the point of connection by injecting active or reactive current into the system.

iii. Controllers with phase angle control

In this case, I_q is set to *zero* and V_{pq} equals $\pm jV_M tan\alpha$; or in other words, a voltage V_{pq} with amplitude $\pm V_M tan\alpha$ is added in quadrature to the mid-point voltage V_M [2.33]-[2.35]. The phase-shifter maintains the transmitted power at its peak for angle δ exceeding $\pi/2$ (the peak power angle) by controlling the amplitude of the quadrature voltage V_{pq} so that the effective phase angle $\delta - \alpha$ between the sending and receiving end voltages is held at $\pi/2$. Thus it can increase the stability limit. In this way, the actual transmitted power may be increased significantly, even though the phase-shifter does not increase the steady-state power transmission limit.

Considering $\delta - \alpha$ as the effective phase angle between the sending and receiving end voltages, then the transmitted power P can be expressed as follows:

$$P_4 = \frac{V^2}{X_s} sin(\delta - \alpha) \qquad 2.31$$



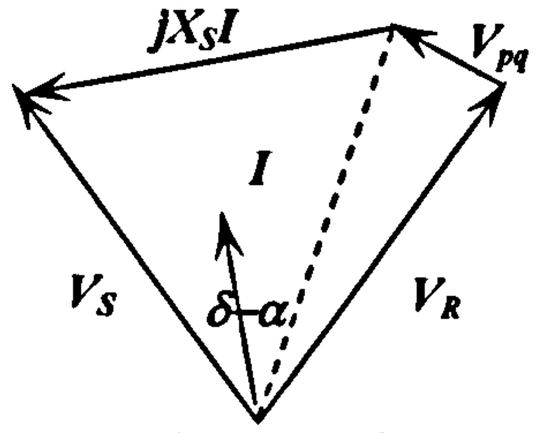


Fig. 2.13 Simple power system model for phase angle control

iv. Combined Series-Shunt Controller Principles

In a combined series-shunt controller, the shunt component injects a current into the system while the series component interposes a series voltage. When these two elements are unified, real power can be exchanged between the two via the power link. Examples of such controllers are the unified power flow controllers (UPFC) and the Thyristor-Controlled Phase-Shifting Transformer (TCPST). Both make use of the advantages of series and shunt controllers, hence facilitate effective and independent power/current flow and line voltage control.

2.2.3 Unified Power Flow Controller (UPFC)

The UPFC is the most versatile FACTS controller, with capabilities of voltage regulation, series compensation, and phase shifting. The UPFC is a member of the family of compensators and power flow controllers. The latter utilizes the synchronous voltage source (SVS) concept to provide unique comprehensive transmission system control [2.36]-[2.38]. The UPFC is able to control simultaneously or selectively all the parameters affecting power flow patterns in a transmission network, including voltage magnitudes and phases, and real and reactive powers. These capabilities make the UPFC a powerful device in present day transmission and control systems.

i. Basic Operating Principles

As illustrated in Fig 2.14, the UPFC is a generalized SVS, represented at the fundamental frequency by controllable voltage phasor of magnitude V_{pq} and angle ρ

injected in series with the transmission line. The angle ρ can be controlled over the range *0* to 2π. For the system shown in Fig 2.15, the converters exchange both real and reactive power with the transmission system. In the UPFC, the real power supplied to or absorbed from the system is provided by one of the end buses to which it is connected. This meets the objective of the UPFC to control power flow rather than increasing the generation capacity of the system.

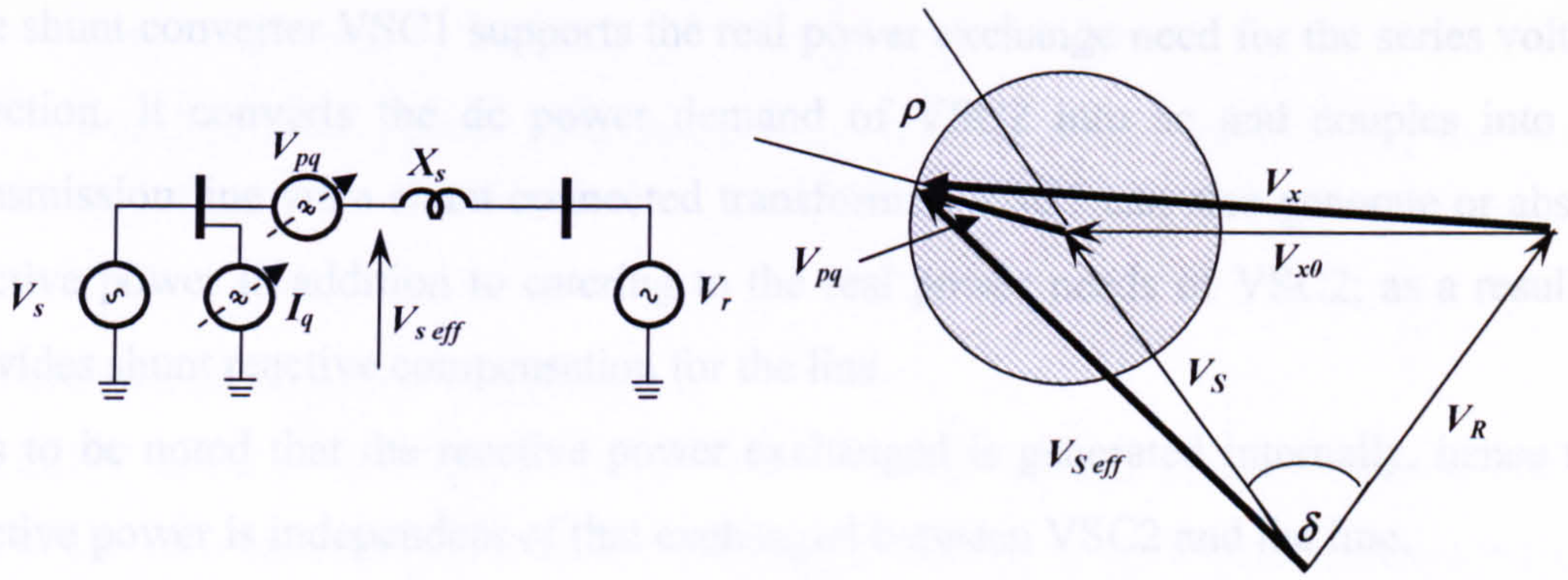


Fig. 2.14 Simple power system model including the unified power flow controller

As shown in Fig 2.15, the UPFC consists of two back-to-back voltage-source converters, one connected in series and the other connected in shunt. This configuration facilitates the free flow of real power in either direction between the ac terminals of the two converters while enabling each converter to independently generate or absorb reactive power at its ac terminals.

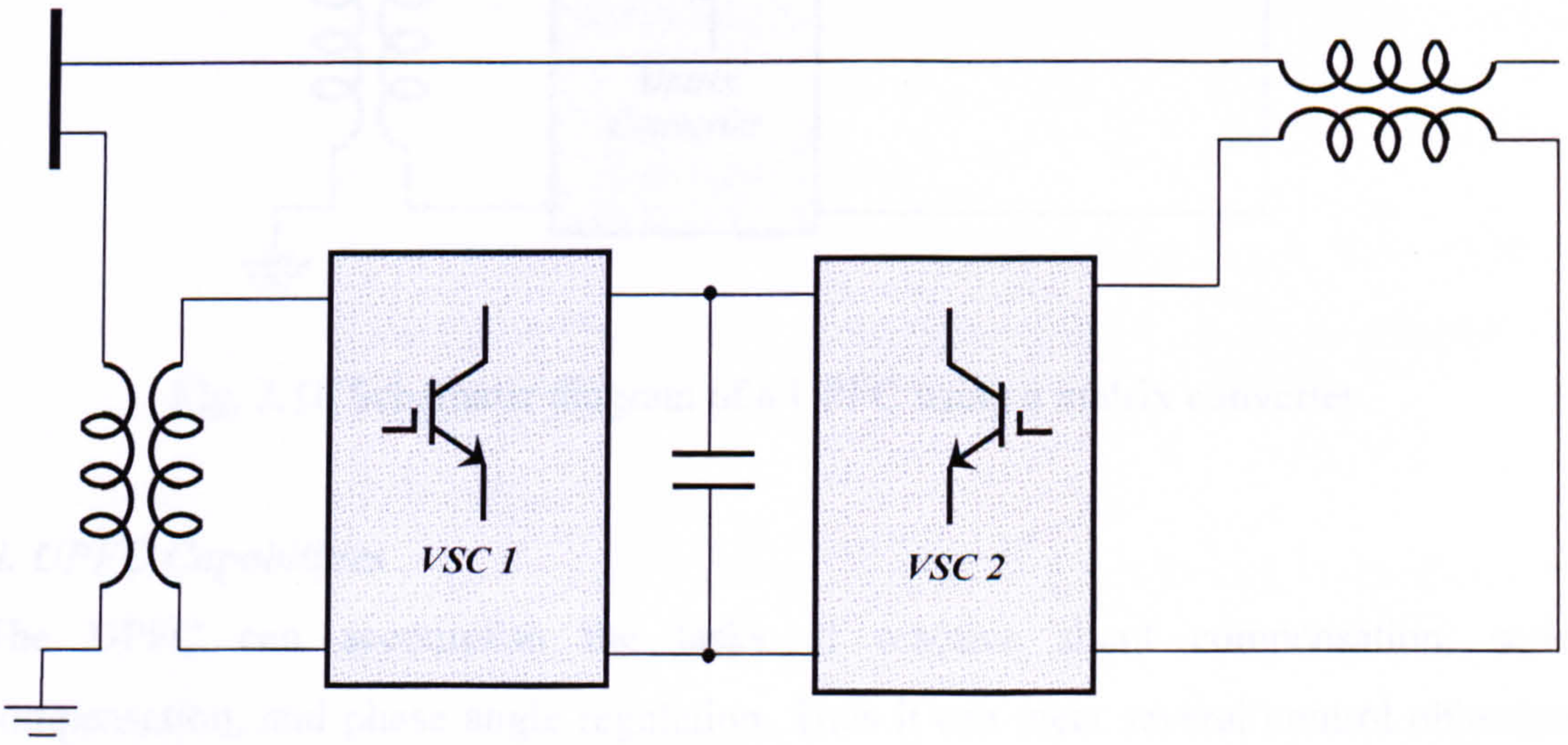


Fig. 2.15 Schematic diagram of a UPFC with back to back converters

The main function of the UPFC is performed by VSC2. It injects a series voltage with controllable magnitude V_{pq} and phase ρ via a series insertion transformer. This injected voltage phasor supplies real and reactive power exchange between the line and converter. The reactive power is generated locally while the real power is converted into dc power and appears on the inverters' interposing dc link as a positive or negative real power demand.

The shunt converter VSC1 supports the real power exchange need for the series voltage injection. It converts the dc power demand of VSC2 into ac and couples into the transmission line via a shunt connected transformer. VSC1 can also generate or absorb reactive power in addition to catering to the real power needs of VSC2; as a result, it provides shunt reactive compensation for the line.

It is to be noted that the reactive power exchanged is generated internally, hence this reactive power is independent of that exchanged between VSC2 and the line.

The matrix converter could be used instead of the two back to back converters, as shown in fig. 2.16 [2.39], [2.40].

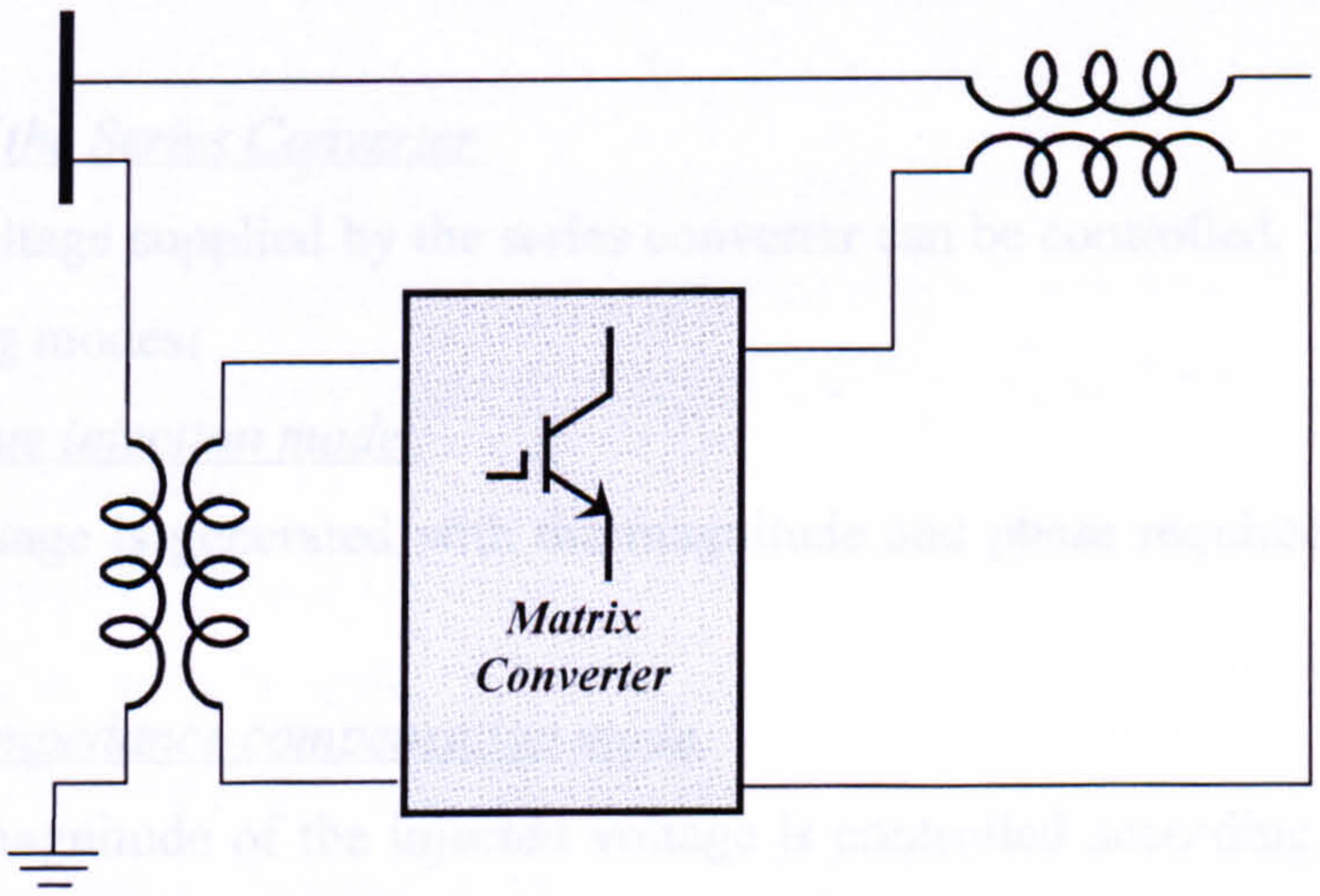


Fig. 2.16 Schematic diagram of a UPFC using a matrix converter

ii. UPFC Capabilities

The UPFC can accomplish the tasks of reactive shunt compensation, series compensation, and phase angle regulation. Thus it can meet several control objectives by introducing a series voltage with controllable amplitude and phase angle. The basic UPFC functions are

- Series reactive compensation by the series injection of voltage, hence the ability to control the transmitted power;
- Voltage regulation by controlling the reactive power of the shunt converter;
- Phase-shifting control that is achieved by injecting a voltage phasor with any particular angular relationship to the terminal voltage. In other words, the desired phase shift can be obtained without any change in the voltage magnitude; and
- Multifunction power flow control by simultaneously performing the above functions.

iii. Control and Dynamic Performance

The UPFC has the capability to introduce a controlled voltage phasor in series with the line to perform the required function. It can not only set up an operating point within a wide range of possible P and Q flows on the line but can also relocate that operating point.

a. Control of the Series Converter

The series voltage supplied by the series converter can be controlled. This converter has four operating modes:

i. voltage injection mode:

A voltage is generated with the magnitude and phase required by the reference input.

ii. line impedance compensation mode:

The magnitude of the injected voltage is controlled according to the magnitude of the line current such that the series injection emulates the reactive impedance, when viewed from the line.

iii. phase angle shift mode:

The injected voltage is controlled with respect to the reference input so that the output bus voltage phasor is shifted by a specific angle.

iv. automatic power flow control mode:

The magnitude and angle of the injected voltage phasor is controlled so as to adjust the line current to achieve the required real or reactive power flow.

b. Control of the Shunt Converter

The shunt converter draws a controlled current phasor from the line, the real part of which is determined by the real power requirement of the series converter while the reactive part can be set to any desired level within the converter's capability.

The shunt part of the UPFC has two modes of operation:

i- Reactive power control:

The reference input is an inductive or capacitive VAr demand. This demand is translated into a corresponding shunt current demand by the shunt converter, which adjusts the gating of the converters to establish the desired current.

ii- Automatic voltage control:

The shunt converter reactive current is automatically regulated to maintain the transmission line voltage at the point of connection.

2.3 Embedded Generation

Today's electrical power systems typically connect a few quite large generators to a large number of small and widely distributed loads. For several reasons, future generation in the UK and elsewhere will probably include many more units of rather smaller size and much closer to the customers end. This is called embedded generation (EG). Drivers to this new network configuration include the need of electricity market liberalization, the efficiency improvements of small-scale generation units, and the aim to reduce CO_2 emission.

Embedded generation can offer many benefits to the electric grid network [2.42]. To make best use of these benefits, EG units have to be connected at appropriate locations and have suitable capabilities [2.43]. On the other hand, some EG units, such as solar and wind, are weather dependant which means that there is no guarantee that they can meet all the necessary operational conditions unless huge storage units are used. Some questions related to power quality, proper system operation, and network protection should be answered before these units are connected to power systems.

This section gives a brief summary of EG units and discusses their possible impact on transmission and distribution networks.

2.3.1 Definition of embedded generation

EG is defined as the integrated or stand-alone utilization of small, modular electric generation near the end-user terminals [2.45]. Another generic definition assigns the EG definition as any generation utilized near consumers in spite of the size or the type of the unit [2.44], [2.46]. According to the latter definition, EG may comprise any generation integrated into the distribution or transmission system, commercial and residential back-up generation, stand-alone onsite generators, and generators installed by the utility for voltage support or other reliability purposes.

2.3.2 Benefits of utilizing EG units

In many applications, EG technology can provide valuable benefits for both the consumers and the electric system [2.42]-[2.50]. The small size of EG units encourages their utilization in a wide range of applications. The location of EG units in distribution systems may reduce energy losses but delays upgrades to both transmission and distribution networks.

The main advantages of utilizing EG can be summarized as follows:

- ***Reduced grid costs:*** Embedded generation can cut utility costs by postponing the need for upgrading distribution and transmission facilities, if generators are located where the grid is constrained.
- ***Improved reliability:*** A large number of small units is more reliable than a small number of large units. This is because a failure of any one unit has far less impact.
- ***Reduced environmental impact:*** Some types of EG produce power with less environmental impact than conventional generation.
- ***Reduced investment risk:*** Smaller, more modular units require less project capital and less lead-time than large power plants. Power plant additions normally exceed demand, leaving extensive amounts of capacity unused until demand catches up. Smaller units can better match gradual increases in demand.

Other benefits of embedded generation are:

- Distributed energy makes energy systems more secure from terrorist attack.
- Generating power where it is needed decreases the losses over transmission lines.

- Small generating equipment can more readily be resold or moved to a better location.
- Sale of power is a potential profit centre for businesses.
- Increasing the number of suppliers selling energy and capacity increases competition and stems market monopoly.

2.3.3 Impact of EG on the transmission network

The installation of EG should lower transmission losses, hence increase system efficiency. Additionally, upgrades and construction of new transmission lines can be avoided if the bulk of the EG units are introduced at the end of existing transmission lines. Consequently, the pressure on transmission companies to build new lines decreases.

2.3.4 Impact of EG on the distribution network

The extensive use of EG units in power systems may influence both grid steady state and transient performance. Some concerns of importance are:

a- Network voltage changes

The voltage regulation concept depends on radial power flow from substations to loads. The use of EG units could cause electrical power to flow in a different manner, distorting the usual concept.

b- Power quality

High power quality requires appropriate voltage and frequency levels at the customer point. EG units have to operate effective controllers to maintain both voltage and frequency within standard levels. In addition to the level itself, the voltage contents of flicker and harmonics have to be minimized.

c- Protection system for the distribution network

Distribution networks are conventionally designed for unidirectional power flow, which means power is normally from the higher voltage level to customers located along radial feeders [2.51]. EG implementation affects protection, necessitating vital alterations to protection coordination [2.49].

The contribution of a small EG unit minimally influences the system short circuit current level. Yet a number of small EG units or a single large unit could cause a

considerable variation in short circuit currents sensed by the protective devices [2.52]-[2.53].

After an unexpected loss of grid connection, some network parts containing EG units may continue to operate as an island [2.49]. This mode of operation is unwanted since the reconnection and synchronization of the islanded part to the network will be complicated [2.54].

References:

[2.1] L. Gyugi and B. Pelly, **'Static Power Frequency Changers: Theory, Performance and Applications'**. New York: Wiley, 1976.

[2.2] A. Alesina and M.B. Venturini, **'Solid-state Power Conversion: A Fourier Analysis Approach to Generalized Transformer Synthesis'**. IEEE Trans. Circuits Syst., Volume: CAS-28, pp. 319-330, Apr. 1981.

[2.3] A. Alesina and M.B. Venturini, **'Intrinsic Amplitude Limits and Optimum Design of 9-Switches Direct PWM AC-AC Converters'**, in 19th Power Electronics Specialists Conf. Rec., 1988, Volume: 2, pp. 1284-1291.

[2.4] A. Alesina and M.B. Venturini, **'Analysis and Design of Optimum-Amplitude Nine Switch Direct AC-AC Converters'**, IEEE Trans. Power Elec., Volume: 4. Issue: 1, pp. 101-112, Jan. 1989.

[2.5] J. Rodriguez, **'A new control technique for AC-AC converters,'** Proc. IFAC Control in Power Electronics and Electrical Drives Conf., Lausanne, Switzerland, 1983, pp. 203-208.

[2.6] L. Huber and D. Borojevic, **'Space Vector Modulator for Forced Commutated Cycloconverters'**, Industry Applications Society Annual Meeting, 1989, Conf. Rec. 1989 IEEE, Volume: 1, pp. 871-876.

[2.7] L. Huber, D. Borojevic and N. Burány, **'Voltage Space Vector Based PWM Control of Forced Commutated Cycloconverters'**, Industrial Electronics Society 1989. IECON'89, 15th Annual Conf. IEEE 1989, Volume: 1, pp. 106-111.

[2.8] L. Huber, D. Borojevic and N. Burány, **'Digital Implementation of the Space Vector Modulator for Forced Commutated Cycloconverters'**, Power Electronics and Variable-Speed Drives 1991, Fourth International

Conference 1990, pp. 63-68.

[2.9] L. Huber and D. Borojevic, **'Space Vector Modulation With Unity Input Power Factor For Forced Commutated Cycloconverters'**, Industry Applications Society Annual Meeting 1991, Conf. Rec. 1991 IEEE, Volume: 1, pp. 1032-1041.

[2.10] L. Huber and D. Borojevic, **'Space Vector Modulated Three-Phase to Three-Phase Matrix Converter with Input Power Factor Correction'**, IEEE Trans. Industry Appl., Volume: 31, Issue: 6, pp. 1234-1246, Nov. /Dec. 1995.

[2.11] P. Nielsen, F. Blaabjerg, and J. K. Pedersen, **'Space vector modulated matrix converter with minimized number of switchings and a feedforward compensation of input voltage unbalance'**, Proc. IEEE International Conference on Power Electronics, *Drives and Energy Systems for Industrial Growth,* Volume: 2, 1996, pp. 833-839.

[2.12] N. Burany, **'Safe control of four-quadrant switches,'** in Conf. Rec. IEEE-IAS Annu. Meeting, Volume: 1, 1989, pp. 1190-1194.

[2.13] L. Zhang, C. Wathanasarn, and W. Shepherd, **'Switch sequencer circuit for safe commutation of a matrix converter,'** Electronics Letters, Volume: 31, pp. 1530-1532, Aug. 1995.

[2.14] J. Youm, and B. Kwon, **'Switching technique for current-controlled AC-to-AC converters,'** IEEE Transactions on Industrial Electronics, Volume: 46, pp. 309-318, April 1999.

[2.15] P. W. Wheeler *et al.*, **'Gate drive level intelligence and current sensing for matrix converter current commutation,'** IEEE Trans. on Industrial Electronics, Volume: 49, pp. 382-389, April 2002.

[2.16] L. Wei, and T. A. Lipo, **'A novel matrix converter topology with simple commutation,'** Conf. Rec. IEEE-IAS Annu. Meeting, Volume: 3, 2001, pp.1749-1754.

[2.17] M. Hornkamp *et al.* **'Economac, the first all-in-one IGBT module for matrix converters,'** Proc. Drivers and Controls and Power Electronics Conf., London, U.K, 2001, pp. 35-39.

[2.18] S. Bernet, T. Matsuo, and T. A. Lipo, **'A matrix converter using reverse**

blocking NPN-IGBTs and optimized pulse patterns,' Proc. IEEE PESC'96, Volume: 1, 1996, pp. 107-113.

[2.19] Motto, E.R. *et al.*, **'Application characteristics of an experimental RB-IGBT (reverse blocking IGBT) module'**, Industry Applications Conference, 2004. 39th IAS Annual Meeting. Conference Record of the 2004 IEEE, Volume 3, 3-7 Oct. 2004 pp.1540-1544.

[2.20] Naito, T., *et al.*, **'1200V reverse blocking IGBT with low loss for matrix converter'**, Power Semiconductor Devices and ICs, 2004. Proceedings. ISPSD '04. The 16th International Symposium, 24-27 May 2004 pp.125-128.

[2.21] K. Sun *et al.*, **'Performance Improvements for Matrix Converter Based on Reverse Blocking IGBT'**, Power Electronics Specialists Conference, 2006. PESC '06. 37th IEEE 18-22 June 2006 pp.1-6.

[2.22] M. Takei, T. Naito, and K. Ueno, **'The reverse blocking IGBT for matrix converter with ultra-thin wafer technology,'** Proc. IEEE ISPSD'03, 2003, pp. 156-159.

[2.23] Klumpner, C.; Blaabjerg, F.; **'Using reverse-blocking IGBTs in power converters for adjustable-speed drives'**, Industry Applications, IEEE Transactions on Volume 42, Issue 3, May-June 2006 pp. 807-816.

[2.24] P. N. Enjeti, and X. Wang, **'A critical evaluation of harmonic generated by forced commutated cycloconverters under unbalance,'** Proc. IEEE IECON'90, Volume: 2, 1990, pp. 1162-1166.

[2.25] S. Kwak, R. M. O. Dabbousi, and H. A. Toliyat, **'An instantaneous compensation technique for supply disturbances in direct frequency converters,'** Proc. IEEE IECON'02, Volume: 2, 2002, pp. 956-960.

[2.26] D. Casadei, G. Serra, and A. Tani, **'Reduction of the input current harmonic content in matrix converters under input/output unbalance,'** IEEE Trans. On Industrial Electronics, Volume: 45, pp. 401- 411, June 1998.

[2.27] D. Casadei, G. Serra, and A. Tanl, **'A general approach for the analysis of the input power quality in matrix converters,'** Proc. IEEE PESC'96, Volume: 2, 1996, pp. 1128-1134.

[2.28] F. Blaabjerg, *et al.*, **'Comparison of two current modulation strategies for matrix converters under unbalanced input voltage conditions,'** IEEE

Trans. on Industrial Electronics, Volume: 49, April 2002 pp. 289- 296.

[2.29] J. Kang, *et al.*, **'The matrix converter drive performance under abnormal input voltage conditions,'** IEEE Trans. on Power Electronics, Volume: 17, Sept. 2002, pp. 721-730.

[2.30] W. Lixiang, Y. Matsushita, and T. A. Lipo, **'Investigation of dual-bridge matrix converter operating under unbalanced source voltages,'** Proc. IEEE PESC'03, Volume: 3, 2003, pp. 1293-1298.

[2.31] C. L. Neft, and C. D. Schauder, **'Theory and design of a 30-hp matrix converter'**, Conf. Rec. IEEE-IAS Annu. Meeting, Volume: 1, 1988, pp. 934-939.

[2.32] Nielsen, P., Blaabjerg, F., and Pedersen, J.K., **'Novel solutions for protection of matrix converter to three phase induction machine'**, Industry Applications Conference, 1997. 32nd IAS Annual Meeting, IAS '97, Conference Record of the 1997 IEEE, Volume: 2, 5-9 Oct. 1997 , pp. 1447-1454.

[2.33] J. Mahlein, and M. Braun, **'A matrix converter without diode clamped overvoltage protection,'** Proc. IEEE PIEMC'00, Volume: 2, 2000, pp. 817-822.

[2.34] Hingorani, N.G., **'High Power Electronics and flexible AC Transmission System'**, Power Engineering Review, IEEE, Volume: 8 Issue: 7 , July 1988, 3-4.

[2.35] Gyugyi, L., *et al.* **'The unified power flow controller: a new approach to power transmission control'**, Power Delivery, IEEE Transactions, Volume:10, Issue:2 , April 1995 pp. 1085-1097.

[2.36] Gyugyi, L., *et al,* **'Understanding FACTS: Concepts and Technology of Flexible AC Transmission Systems',** John Wiley & Sons Inc (10 Dec 1999), **ISBN-10:** 0780334558

[2.37] K. Schoder, **'Analysis and Robust Decentralized Control of Power Systems Using FACTS Devices',** PhD thesis, Morgantown, West Virginia 2002.

[2.38] L. M. Tolbert, **'Power electronics for distributed energy systems and transmission and distribution applications',** The University of Tennessee-

Knoxville and U.S. Department of Energy, December 2005.

[2.39] Boon-Teck Ooi; Kazerani, M, **'Unified power flow controller based on matrix converter'**, Power Electronics Specialists Conference, 1996. PESC '96 Record., 27th Annual IEEE, Volume: 1, 23-27 June 1996, 502-507.

[2.40] Strzelecki R., *et al.* **'UPFC with matrix converter'**, 9th European Conference on Power Electronics and Applications- EPE 2001. Graz 2001.

[2.41] Sangshin K., **'Design and analysis of modern three phase AC/AC power converters for ac drives interface'**, PhD thesis, Texas A&M University, May 2005.

[2.42] C. Timpe, Öko-Institut, M.J.J. Scheepers, ECN, **'A look into the future: Scenario for distributed generation in Europe'**, December 2003.

[2.43] A. Collinson *et al.*, **'Solutions for the connections and operation of distributed generation'**, Prepared under the DTI New & Renewable Energy Programme July 2003.

[2.44] Peter A. Daly and Jay Morrison, **'Understanding the potential benefits of distributed generation on power delivery systems'** Rural Electric Power Conference, 29 April-1 May 2001, pp. A2-1-A2-13, Little Rock, Arkansas, USA.

[2.45] M. Begovic *et al.*, **'Impact of renewable distributed generation on power systems'**, System Sciences, 2001, Proceedings of the 34th Annual Hawaii International Conference on, Jan. 3-6, 2001, pp. 654- 663, Maui, Hawaii.

[2.46] T. Ackermann and V. Knyazkin, **'Interaction between distributed generation and the distribution network: operation aspect'** IEEE/PES Transmission and Distribution Conf. and Exhibition 2002: Asia Pacific, Volume: 2, 6-10 Oct. 2002, pp. 1357-1362, Yokohama, Japan.

[2.47] Bruno Delfino, **'Modelling of the integration of distributed generation into the electrical system'** Power Engineering Society Summer Meeting, 2002 IEEE, Chicago USA, Volume: 1, 21-25 July 2002, pp. 170-175.

[2.48] P. P. Barker and R. W de Mello, **'Determining the impact of distributed generation on power systems: Part I. Radial distribution systems'** Power Engineering Society Summer Meeting, Seattle, Washington, USA, 2000 IEEE, Volume: 3, 16-20 July 2000, pp. 1645-1656.

[2.49] Stan Blazwicz and David Kleinschmidt, **'Distributed Generation: System Interfaces'** White paper prepared by Arthur D. Little, Cambridge, MA. 1999, http://www.encorp.com/dwnld/pdf/whitepaper/wp_ADL_2.pdf.

[2.50] P. P. Barker and B. K. Johnson, **'Power system modeling requirements for rotating machine interfaced distributed resources'** Power Engineering Society Summer Meeting, 2002 IEEE, Chicago, USA, Volume: 1, 21-25 July 2002, pp. 161-166.

[2.51] Kauhaniemi, K., Kumpulainen, L.: **'Impact of distributed generation on the protection of distribution networks'.** Developments in Power System Protection, IEE International Conference, 5-8 April 2004, Amsterdam, pp. 315-318.

[2.52] N. Nimpitiwan, G. Heydt, **'Consequences of Fault Currents Contributed by Distributed Generation',** Power Systems Engineering Research Centre, Arizona State University, June 2006.

[2.53] A. Narang, **'Impact of large scale distributed generation penetration on power system stability',** Chad Abbey Natural Resources Canada, March 2003.

[2.54] V. V. Menon, **'A new islanding detection technique for distributed generation',** Master of science, Montana state university, January 2006.

CHAPTER THREE

A NOVEL CONTROL TECHNIQUE FOR THE MATRIX CONVERTER

The matrix converter is a direct frequency conversion device with high input power quality and regeneration capability. In this chapter, a novel technique for controlling the matrix converter output current, hence the output power is introduced. Not only can sinusoidal output currents be obtained but any output current waveform is possible even if the input voltage is slightly distorted or unbalanced. Like the space vector modulation technique, the method is based on separating the operation of the conventional matrix converter into two virtual stages: the rectifier stage and the inverter stage as shown in Fig. 3.1.

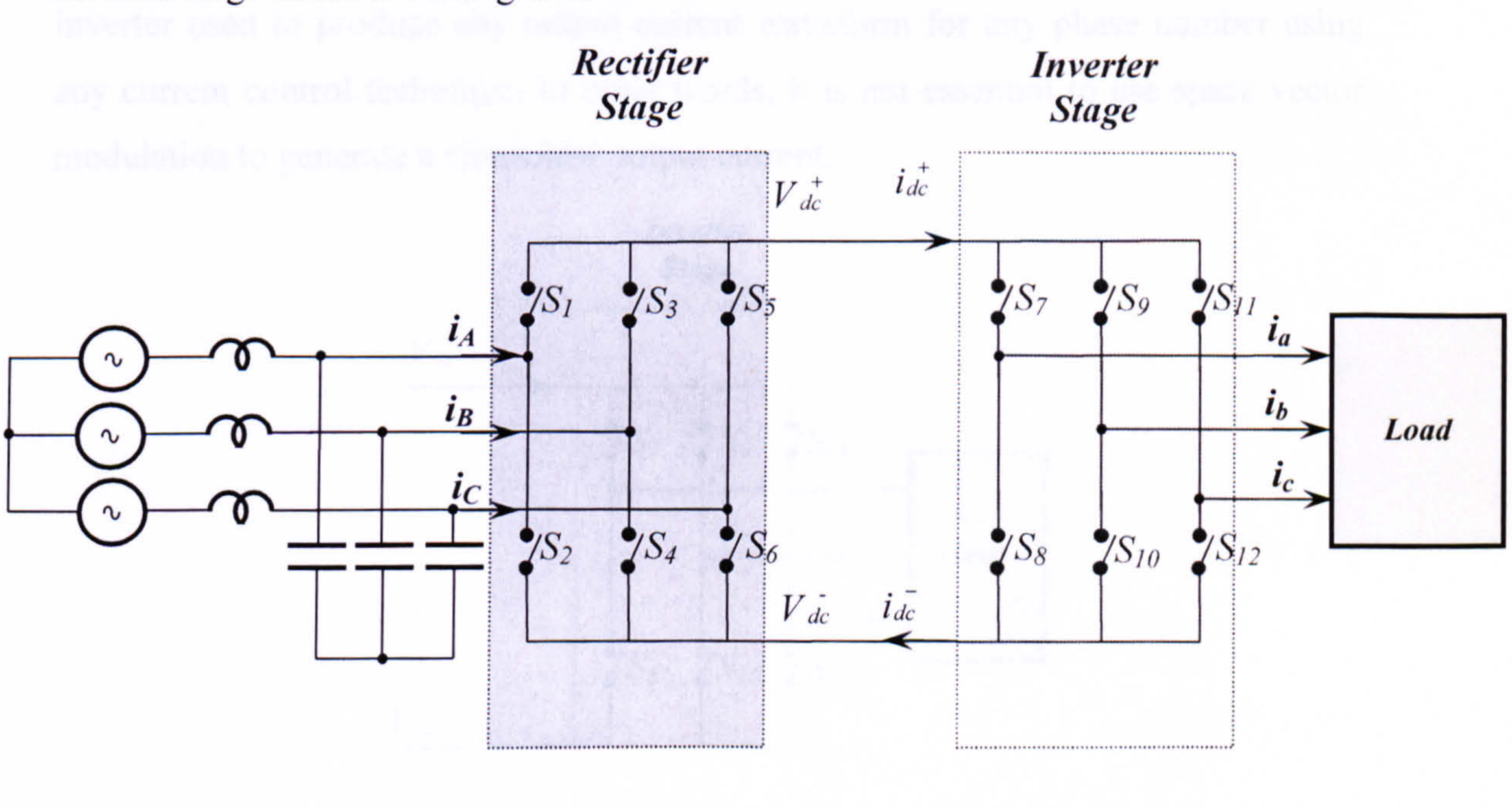


Fig 3.1 Back to back rectifier and inverter virtual stages.

3.1 Control technique [3.1]-[3.8]

A simple control technique for the operation of the matrix converter can be based on current feedback control combined with control of the input displacement angle ϕ_i. The operation of the matrix converter is separated into two stages: an output inverter stage and an active input rectifier stage. The inverter stage is responsible for controlling the output currents (i_a, i_b and i_c) while input displacement angle control is achieved by the active rectifier stage. Each stage generates a matrix. The inverter matrix indicates the switch states that control the output currents for a conventional VSI by any current controller. The rectifier matrix represents the switch states that control the input displacement angle. Subsequently, the unified matrix converter switch configuration can be realised from the resulting matrices using logic circuitry.

3.1.1 The inverter stage

The inverter stage is considered first since the rectifier stage is partially dependant on the inverter stage as will be explained in the rectifier stage section. The main objective of the inverter control stage is to determine the switching states of the inverter switches S_7 to S_{12}, which are shown in Fig. 3.2. This is similar to a normal inverter used to produce any output current waveform for any phase number using any current control technique. In other words, it is not essential to use space vector modulation to generate a sinusoidal output current.

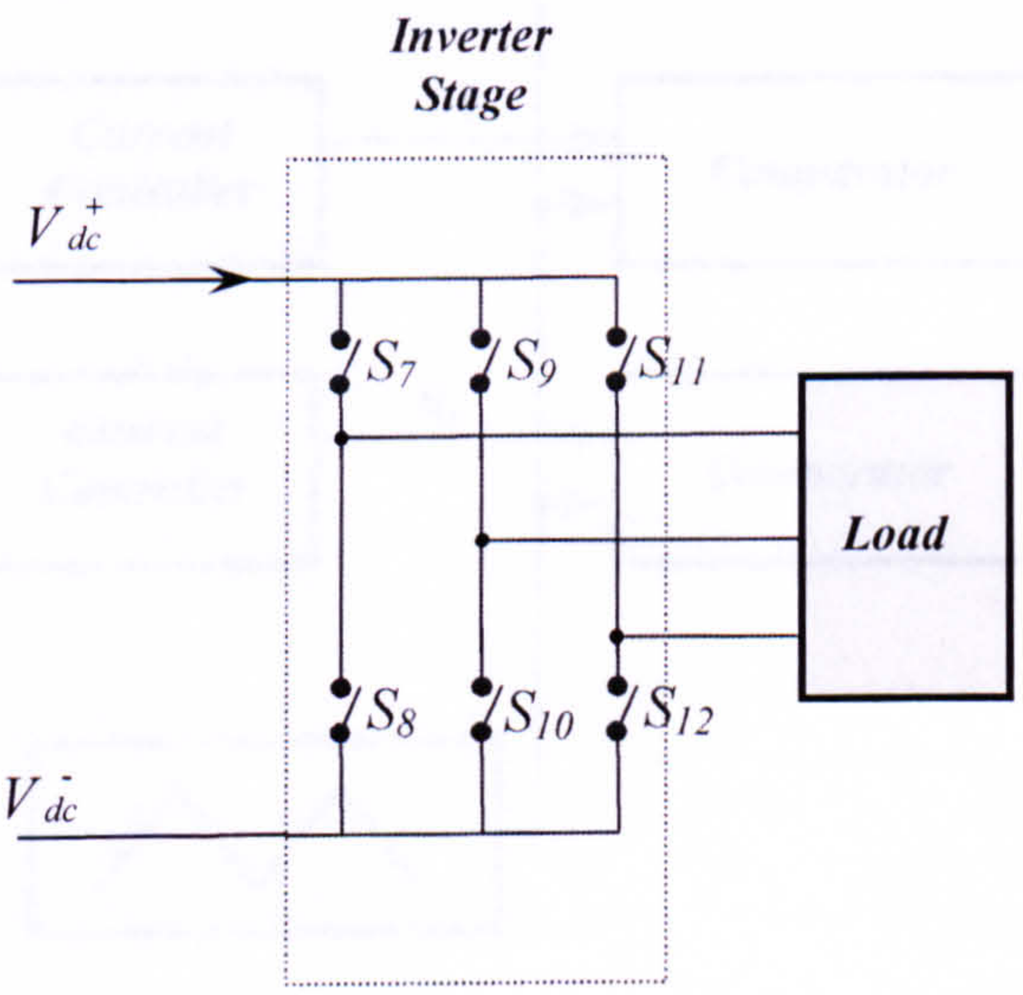


Fig 3.2 Output inverter stage diagram.

For simplicity, a three-phase system and a ramp comparator current controller scheme (triangular carrier) are used for the explanation of the inverter stage controller. The inputs to the controller are the difference error between the output current commands i_a^*, i_b^* and i_c^* and the feedback output currents i_a, i_b and i_c as shown in Fig. 3.3. In the simplest form, each controller output (u_a, u_b or u_c) will be a function of the input references and the actual currents.

$$\begin{aligned} u_a &= f\left(i_a^*, i_a\right) \\ u_b &= f\left(i_b^*, i_b\right) \\ u_c &= f\left(i_c^*, i_c\right) \end{aligned} \qquad 3.1$$

The output reference voltages of the inverter controller are responsible for generating the switching signals, $S_7(t)$, $S_8(t)$..... $S_{12}(t)$, by comparing these voltages with the carrier. Up to four different inverter matrices from a possible eight are generated during every sampling cycle depending on the magnitude of the controller outputs.

Fig 3.4 represents the distribution of these matrices during one sampling period for the case $\hat{U} > u_a > u_b > u_c > -\hat{U}$ ($\hat{U}$ is the peak of the carrier). Each inverter matrix I_1 is one of the eight matrices shown in table 3.1.

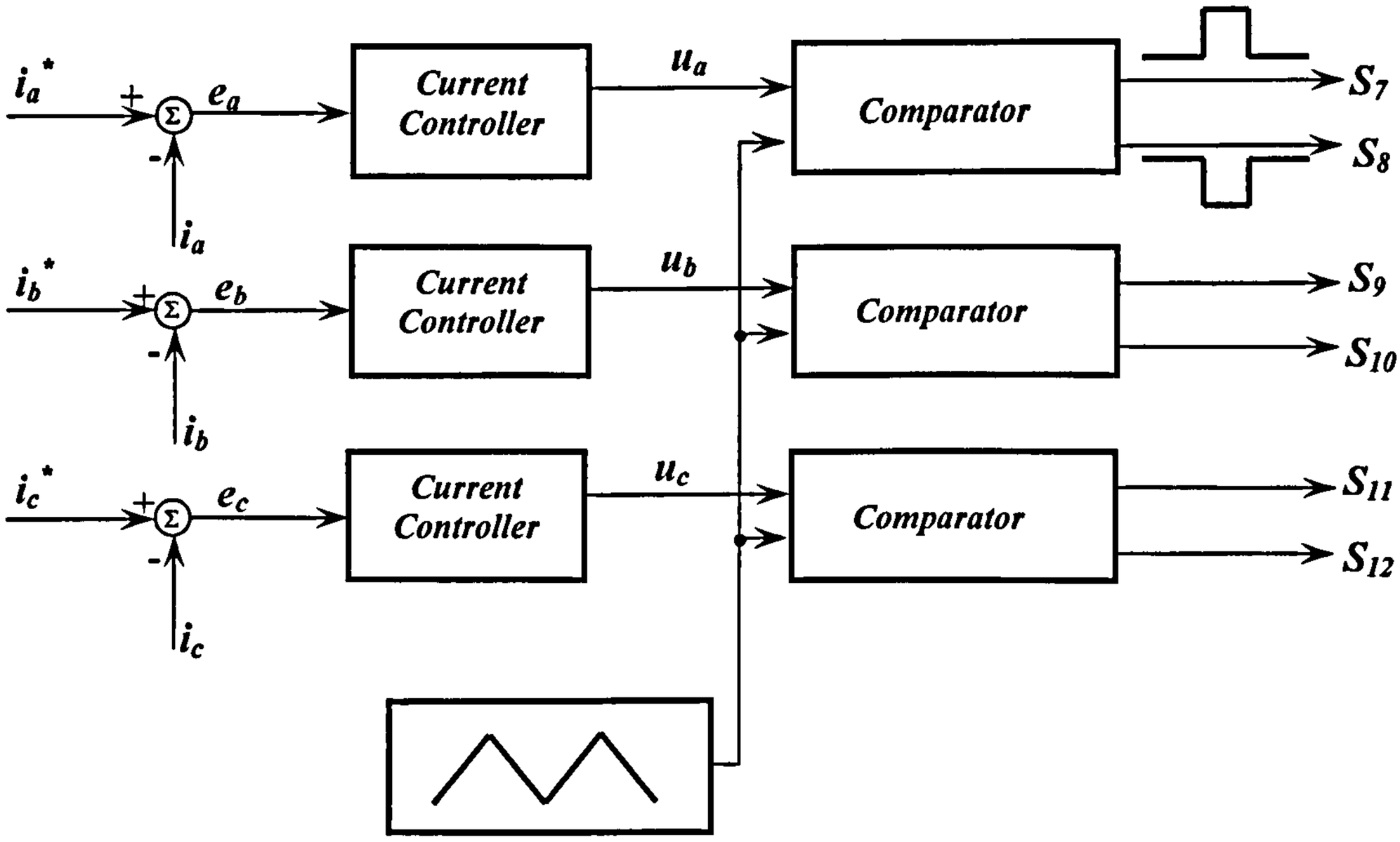


Fig 3.3 The inverter stage block diagram.

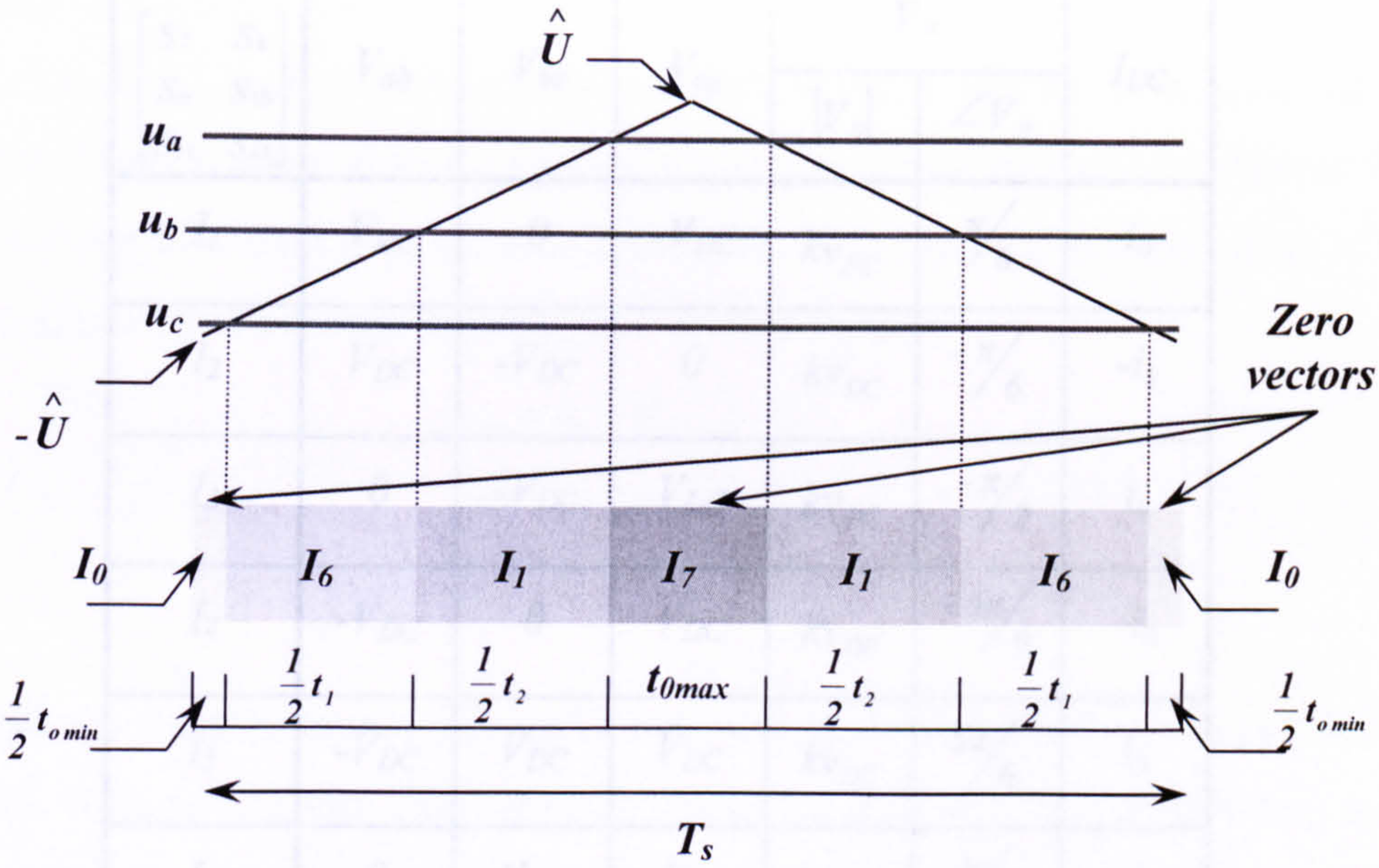


Fig 3.4 The inverter output matrices for the case $\hat{U} > u_a > u_b > u_c > -\hat{U}$.

	I_0	I_1	I_2	I_3	I_4	I_5	I_6	I_7
Inverter Switch states $\begin{bmatrix} S_7 & S_8 \\ S_9 & S_{10} \\ S_{11} & S_{12} \end{bmatrix}$	$\begin{bmatrix} 1 & 0 \\ 1 & 0 \\ 1 & 0 \end{bmatrix}$	$\begin{bmatrix} 1 & 0 \\ 0 & 1 \\ 0 & 1 \end{bmatrix}$	$\begin{bmatrix} 1 & 0 \\ 0 & 1 \\ 1 & 0 \end{bmatrix}$	$\begin{bmatrix} 0 & 1 \\ 0 & 1 \\ 1 & 0 \end{bmatrix}$	$\begin{bmatrix} 0 & 1 \\ 1 & 0 \\ 1 & 0 \end{bmatrix}$	$\begin{bmatrix} 0 & 1 \\ 1 & 0 \\ 0 & 1 \end{bmatrix}$	$\begin{bmatrix} 1 & 0 \\ 1 & 0 \\ 0 & 1 \end{bmatrix}$	$\begin{bmatrix} 0 & 1 \\ 0 & 1 \\ 0 & 1 \end{bmatrix}$

Table 3.1 Inverter matrices corresponding to the different inverter switching states.

The corresponding output voltage expressions and the DC link current can be calculated using equations 3.2 and 3.3

$$\begin{bmatrix} v_a \\ v_b \\ v_c \end{bmatrix} = \begin{bmatrix} S_7 & S_8 \\ S_9 & S_{10} \\ S_{11} & S_{12} \end{bmatrix} \cdot \begin{bmatrix} v_{DC+} \\ v_{DC-} \end{bmatrix} \qquad 3.2$$

$$\begin{bmatrix} i_{DC+} \\ i_{DC-} \end{bmatrix} = \begin{bmatrix} S_7 & S_9 & S_{11} \\ S_8 & S_{10} & S_{12} \end{bmatrix} \begin{bmatrix} i_a \\ i_b \\ i_c \end{bmatrix} \qquad 3.3$$

The relation between the output values (voltages and currents) and the virtual dc link values for the different switching states are summarized in table 3.2

$\begin{bmatrix} S_7 & S_8 \\ S_9 & S_{10} \\ S_{11} & S_{12} \end{bmatrix}$	V_{ab}	V_{bc}	V_{ca}	V_o		I_{DC}
				$\lvert V_o \rvert$	$\angle V_o$	
I_1	V_{DC}	0	$-V_{DC}$	kv_{DC}	$\pi/6$	i_a
I_2	V_{DC}	$-V_{DC}$	0	kv_{DC}	$-\pi/6$	$-i_b$
I_3	0	$-V_{DC}$	V_{DC}	kv_{DC}	$--\pi/2$	i_c
I_4	$-V_{DC}$	0	V_{DC}	kv_{DC}	$-5\pi/6$	$-i_a$
I_5	$-V_{DC}$	V_{DC}	V_{DC}	kv_{DC}	$5\pi/6$	i_b
I_6	0	V_{DC}	$-V_{DC}$	kv_{DC}	$\pi/2$	$-i_c$
I_0 or I_7	0	0	0	0	0	0

Table 3.2 Possible switch state combinations of the inverter

The tips of the reference voltage signals form a hexagon as shown in Fig. 3.5. The converter can only produce voltages that are within this hexagon. Knowing that the voltage vector length is kv_{DC}, the maximum length of the reference voltage is $kv_{DC}\cos 30^0$.

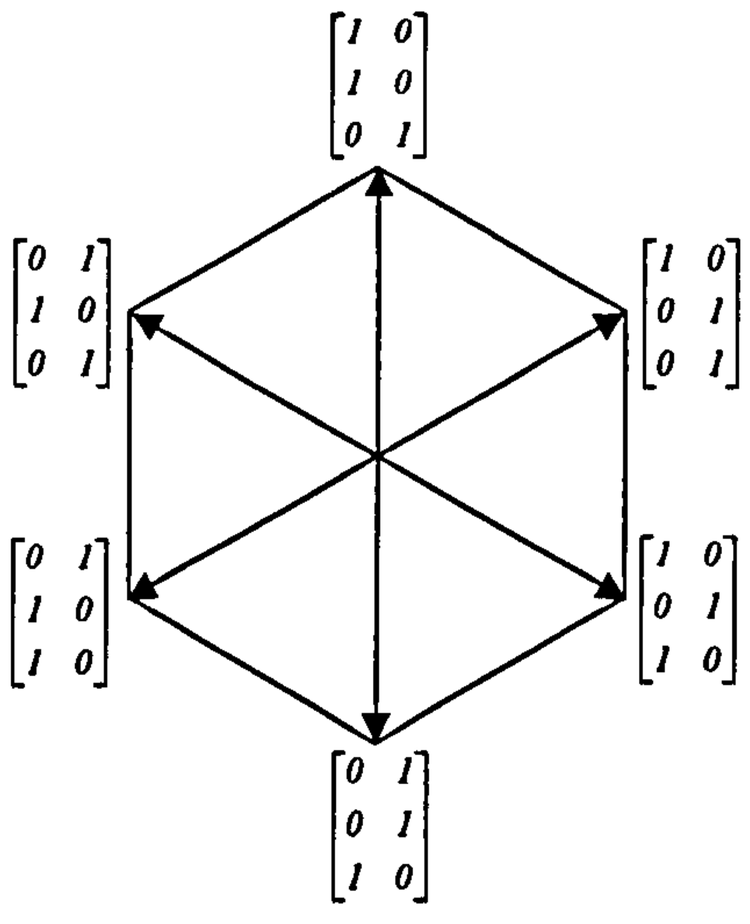


Fig. 3.5 Voltage vectors for a 2 level VSI.

This limit is reached when any of the reference voltage peaks are equal to the peak of the carrier signal, and is termed unity modulation index ($m=1$). If the peak of the reference signal is larger than the peak of the carrier signal, over-modulation occurs, which creates low frequency current harmonics. This effect is reduced if an offset voltage u_x is subtracted from the reference voltage signals:

$$\begin{aligned} u_{aa} &= u_a - u_x \\ u_{bb} &= u_b - u_x \\ u_{cc} &= u_c - u_x \end{aligned} \qquad 3.4$$

Although the phase reference voltages are no longer sinusoidal as shown in Fig. 3.6, the load voltages remain sinusoidal. In order to optimize the modulation, u_x is selected such that the reference signals are symmetrical as in the following equation

$$u_x = \frac{max(u_a, u_b, u_c) + min(u_a, u_b, u_c)}{2} \qquad 3.5$$

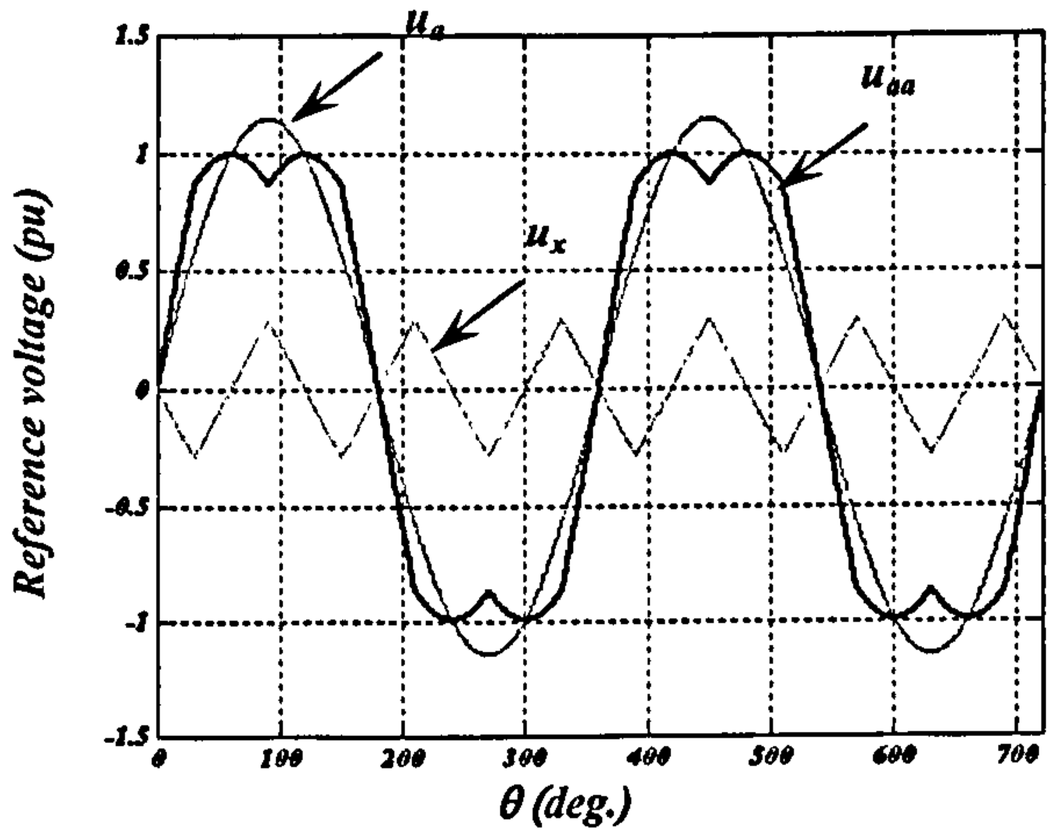


Fig. 3.6 Modified command signal for phase a, by adding $u_x = \frac{max(u_a^*, u_b^*, u_c^*) + min(u_a^*, u_b^*, u_c^*)}{2}$

The reference signal adjustment can be obtained by logic circuitry as shown in Fig. 3.7. The voltage command signals are compared to determine their relative magnitudes. According to these levels the signals are multiplexed to determine the maximum and the minimum command signals (*max, min*). Then the offset signal is obtained by adding the (*max* and *min*) signals then divided by 2 (right shifted). The command signals are adjusted by adding this resulting offset value as shown in Fig. 3.8.

Referring to Fig. 3.6, it can be concluded that the modulation index can be reduced by a factor of 0.866 without altering the output voltage. Another benefit obtained is that the minimum and maximum values are equal in magnitude as shown in Fig. 3.9, which means that symmetrical PWM is obtained.

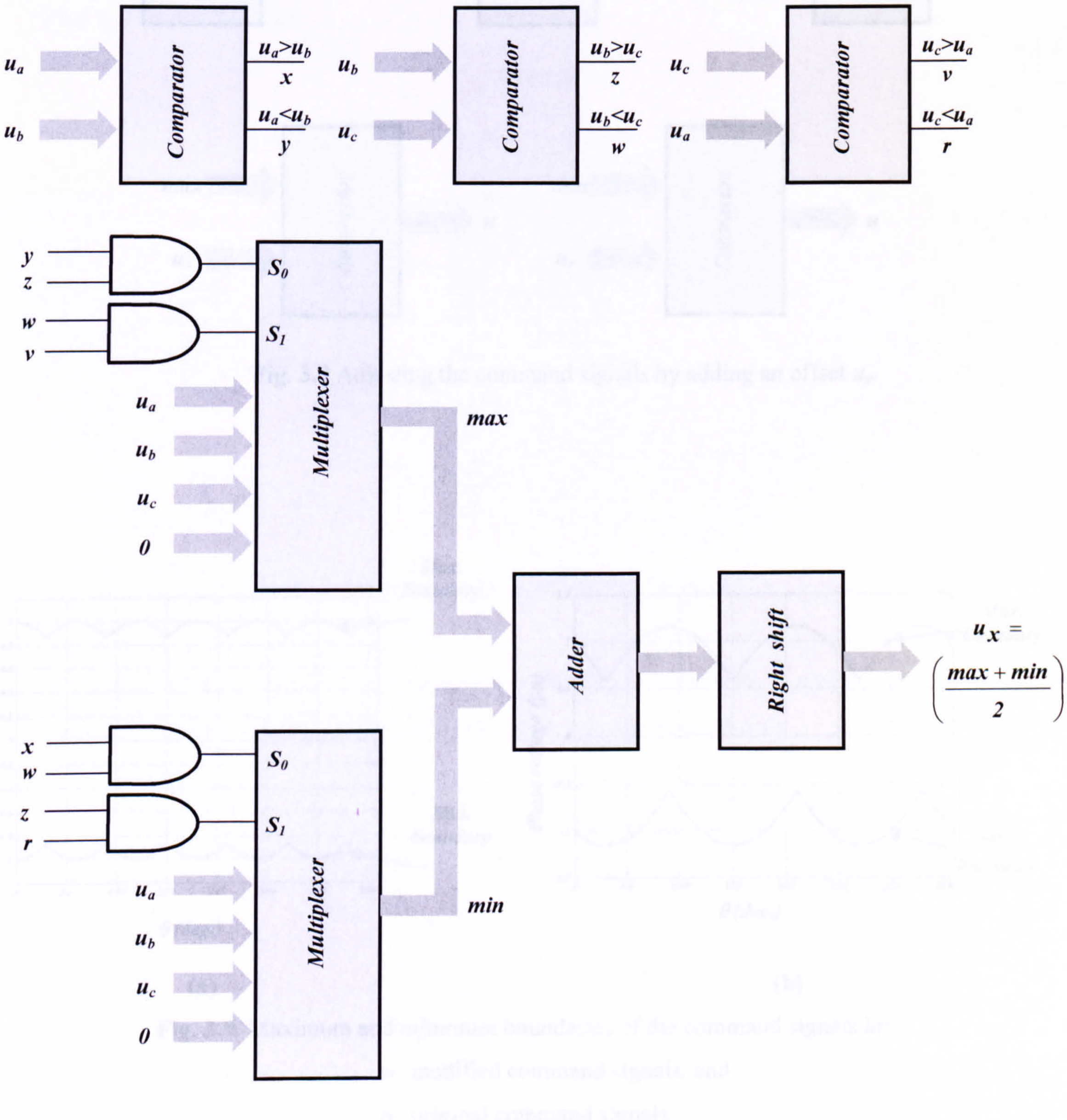
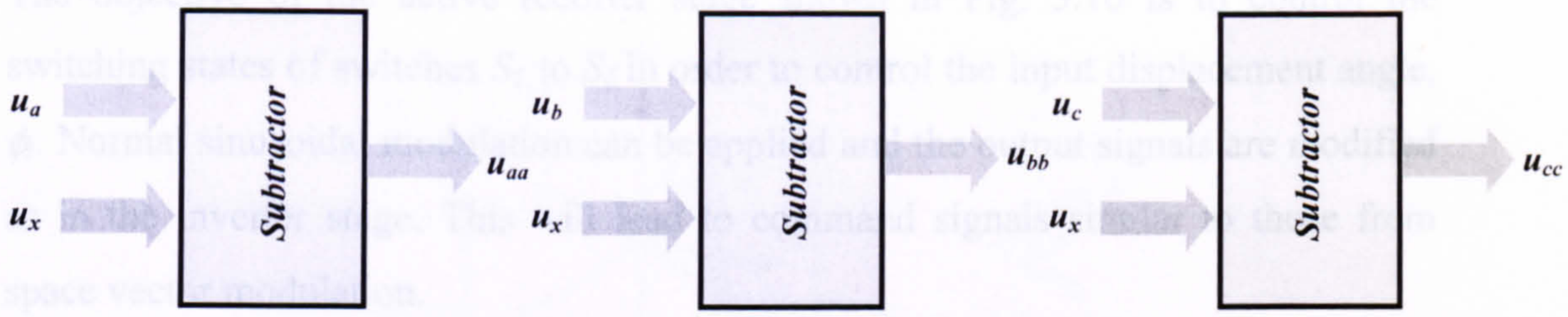


Fig. 3.7 Logic circuitry used to obtain the offset signal u_x.



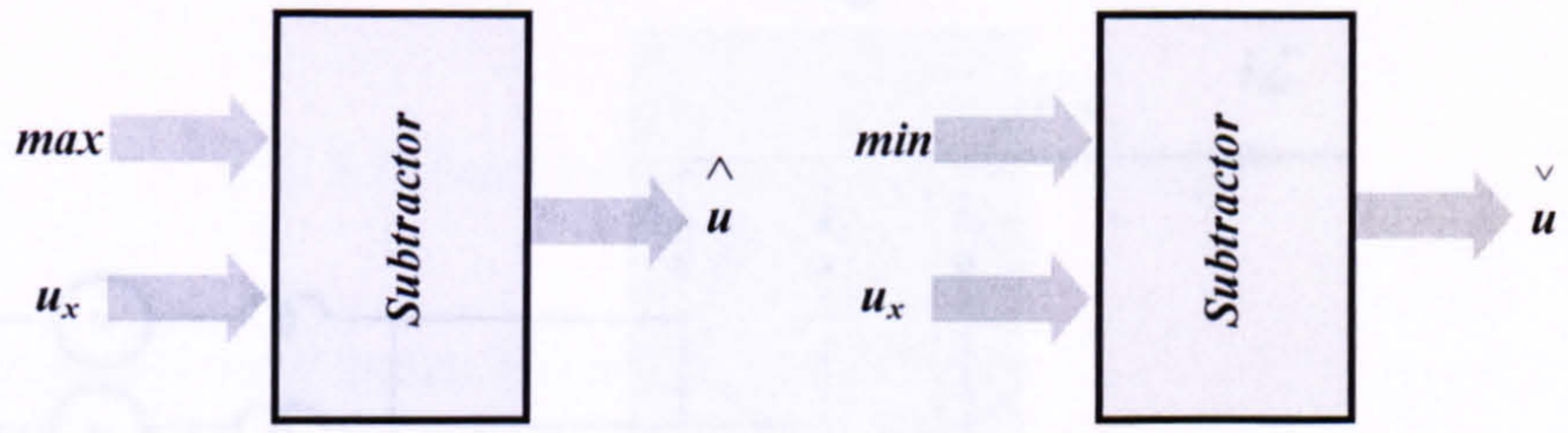


Fig. 3.8 Adjusting the command signals by adding an offset u_x.

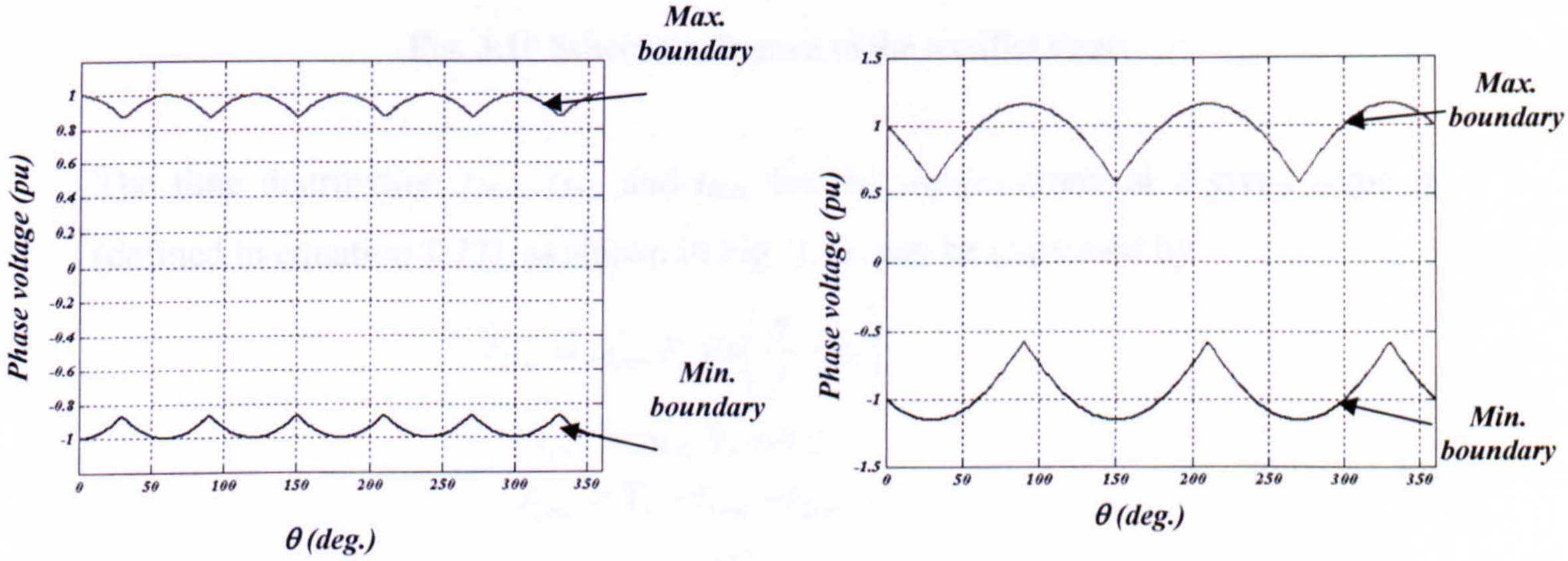


Fig. 3.9 Maximum and minimum boundaries of the command signals in:

a- modified command signals, and

b- original command signals.

3.1.2 The rectifier stage

The objective of the active rectifier stage shown in Fig. 3.10 is to control the switching states of switches S_1 to S_6 in order to control the input displacement angle, ϕ_i. Normal sinusoidal modulation can be applied and the output signals are modified as in the inverter stage. This will lead to command signals similar to those from space vector modulation.

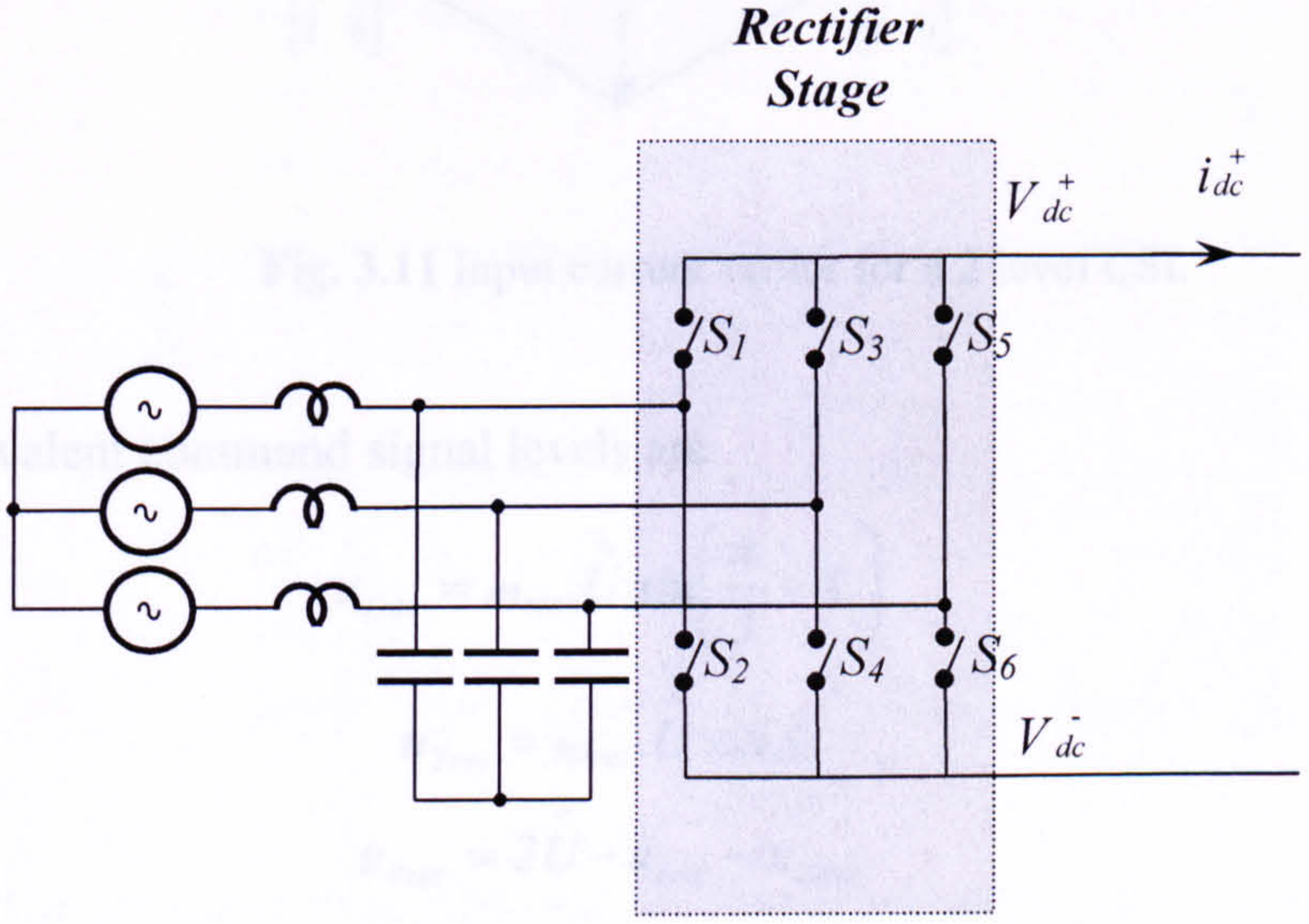


Fig. 3.10 Schematic diagram of the rectifier stage.

The time distribution t_{1rec}, t_{2rec} and t_{0rec} for the input current at a given angle δ_i (defined in equation 2.23), as shown in Fig. 3.11, can be expressed by

$$\begin{aligned} t_{1rec} &= m_{rec}.T_s \sin\left(\frac{\pi}{3} - \delta_i\right) \\ t_{2rec} &= m_{rec}.T_s \sin\delta_i \\ t_{0rec} &= T_s - t_{1rec} - t_{2rec} \\ m_{rec} &= \frac{|I_i|}{i_{DC}} \\ \delta_i &= (\theta + \phi_i) - (k-1)\left(\pi/3\right) \\ k &= floor\left(\frac{\theta_i}{\left(\pi/3\right)}\right) + 1 \end{aligned} \qquad 3.6$$

Where θ is the input voltage phase angle, ϕ_i is the controlled input displacement angle

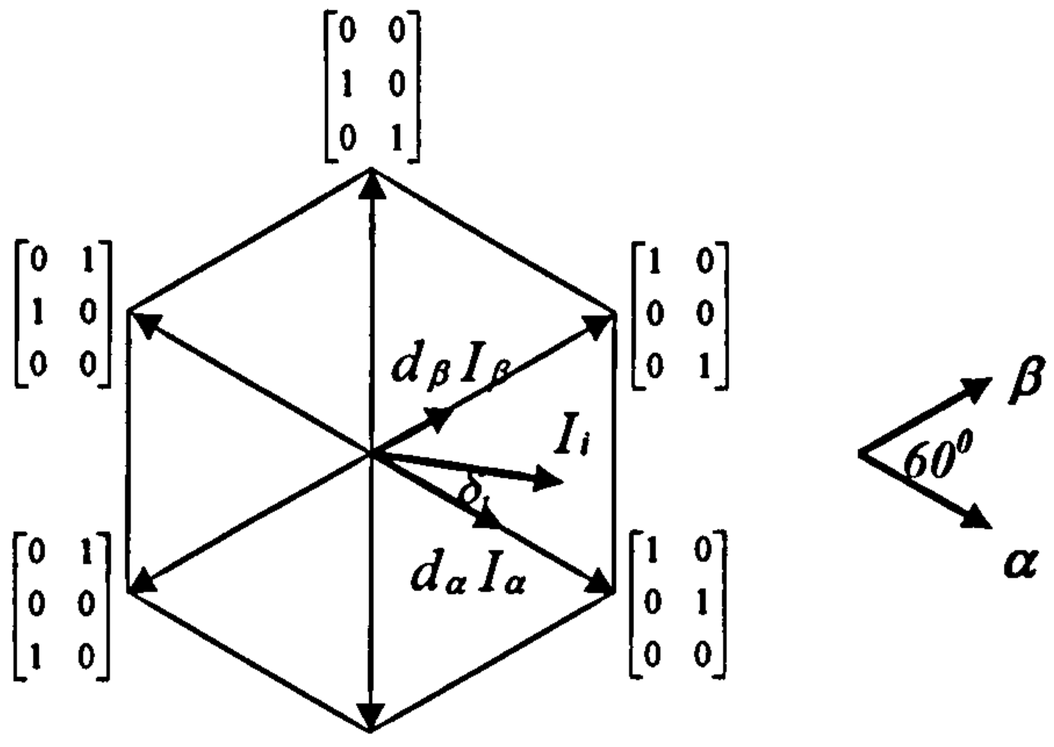


Fig. 3.11 Input current vector for a 2 level CSI.

The equivalent command signal levels are

$$u_{1rec} = m_{rec}.\hat{U}\sin\left(\frac{\pi}{3}-\delta_i\right)$$

$$u_{2rec} = m_{rec}.\hat{U}\sin\delta_i \qquad 3.7$$

$$u_{0rec} = 2\hat{U}-u_{1rec}-u_{2rec}$$

It should be noted that not all the sampling period is effective for the rectifier stage. The zero vectors of the inverter stage reduce the effective sampling period. In other words, no input displacement angle control can be achieved during the inverter zero vectors since no current is drawn during these intervals. Referring to Fig. 3.4, the equivalent time for the zero vectors t_x can be expressed as

$$t_x = T_s - \frac{T_s}{2\hat{U}}\left(\hat{u}-\check{u}\right) \qquad 3.8$$

where

$$\check{u} = \begin{cases} min(u_{aa},u_{bb},u_{cc}) & u_{aa},u_{bb}\text{and}\,u_{cc} > -\hat{U} \\ -\hat{U} & \text{otherwise} \end{cases}$$

$$\hat{u} = \begin{cases} max(u_{aa},u_{bb},u_{cc}) & u_{aa},u_{bb}\text{and}\,u_{cc} < \hat{U} \\ \hat{U} & \text{otherwise} \end{cases}$$

For the reference signal shown in Fig. 3.6, the equivalent time of the zero vectors with respect to the different angle positions of the output voltage command signals is presented in Fig. 3.12. It can be concluded that at unity modulation index, the maximum of this equivalent time is 0.133 pu.

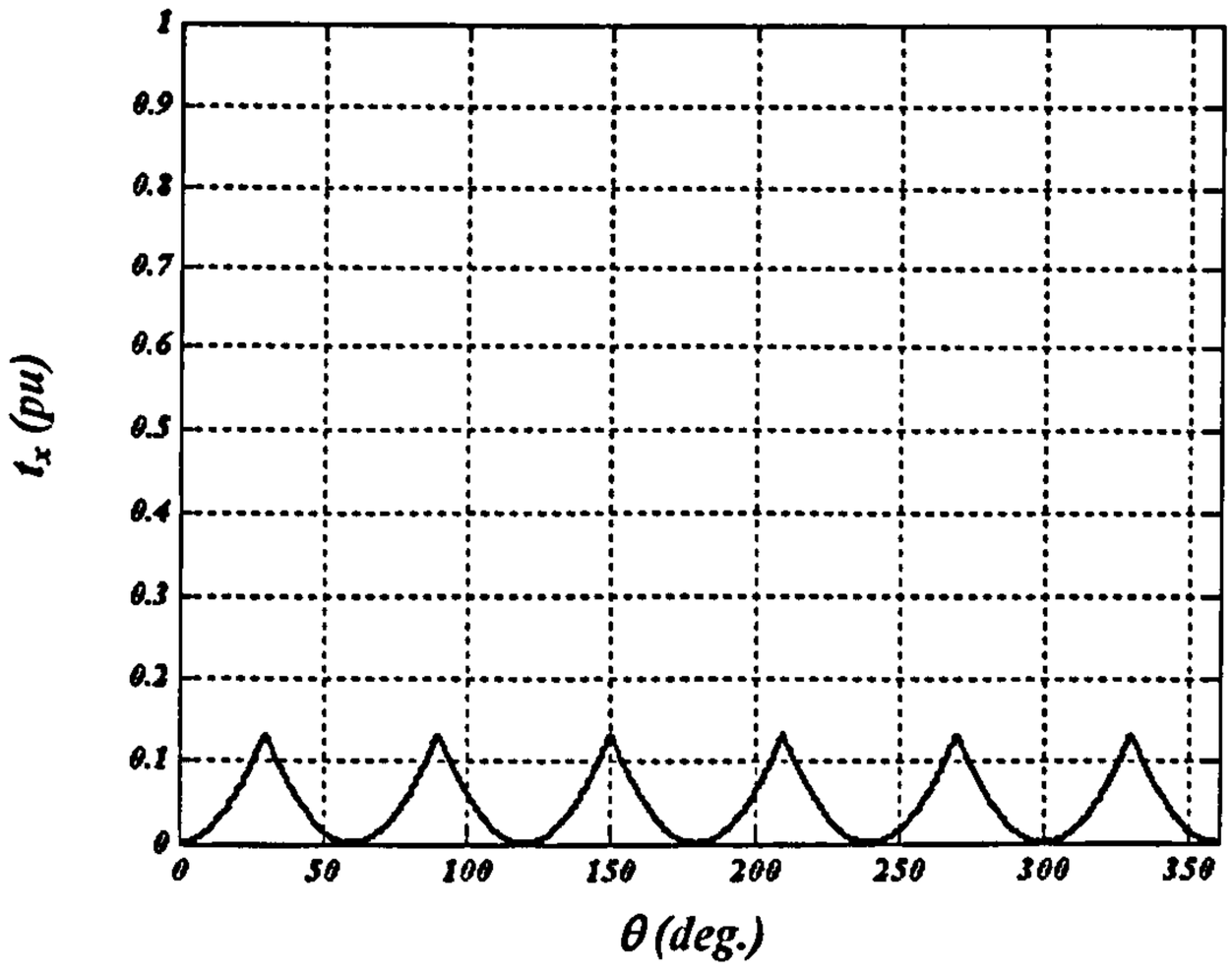


Fig. 3.12 The equivalent time of the zero vectors with respect to the different angle positions of the output voltage commands at unity modulation index.

The effective sampling time for the rectifier stage is the difference between the inverter stage sampling time T_s and the equivalent time for the zero vectors t_x. This time $T_{s\,eff}$ can be expressed as

$$T_{s\,eff} = \frac{T_s}{2\hat{U}}\left(\hat{u} - \check{u}\right) \qquad 3.9$$

The hexagon representing the input current vector will shrink by the ratio of $T_{s\,eff}/T_s$ as shown in Fig. 3.13. The main objective of the rectifier stage is to control the input displacement angle, not the current magnitude. The latter is automatically determined and adjusted by the power balance of the system.

The time distribution t_{1rec}, t_{2rec} and t_{0rec} for the input current at a given angle δ_i can be expressed by

$$t_{1rec} = m_{rec}.T_{s\,eff}\ sin\left(\frac{\pi}{3} - \delta_i\right)$$

$$t_{2rec} = m_{rec}.T_{s\,eff}\ sin\,\delta_i$$

$$t_{0rec} = T_{s\,eff} - t_{1rec} - t_{2rec} \qquad 3.10$$

$$m_{rec} = \frac{|I_i|}{i_{DC}}$$

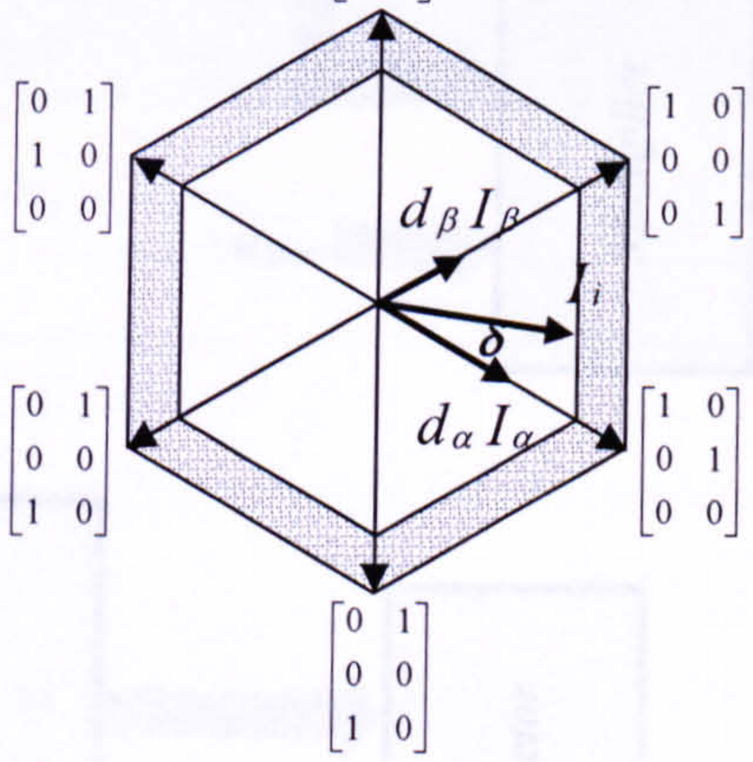


Fig. 3.13 Adjusted hexagon for the shown input current vector.

The equivalent command signal levels that should be used are

$$u^*_{1rec} = m_{rec}.\left(\hat{u} - \check{u}\right) sin\left(\frac{\pi}{3} - \delta_i\right)$$

$$u^*_{2rec} = m_{rec}.\left(\hat{u} - \check{u}\right) sin\,\delta_i \qquad 3.11$$

$$u^*_{0rec} = 2\hat{U} - u^*_{1rec} - u^*_{2rec}$$

These times can be converted into level values which are compared with the triangular carrier

$$u_1 = \frac{1}{2} u^*_{0rec} \qquad 3.12$$

$$u_2 = u^*_{1rec} + u_1$$

$$u_3 = u^*_{2rec} + u_2$$

Again the adjustment in the rectifier stage signals from the normal values is achieved by using logic circuitry as shown in Fig.3.14. Based on the available cited references, this is the first time the effect of the effective sampling time is taken into account.

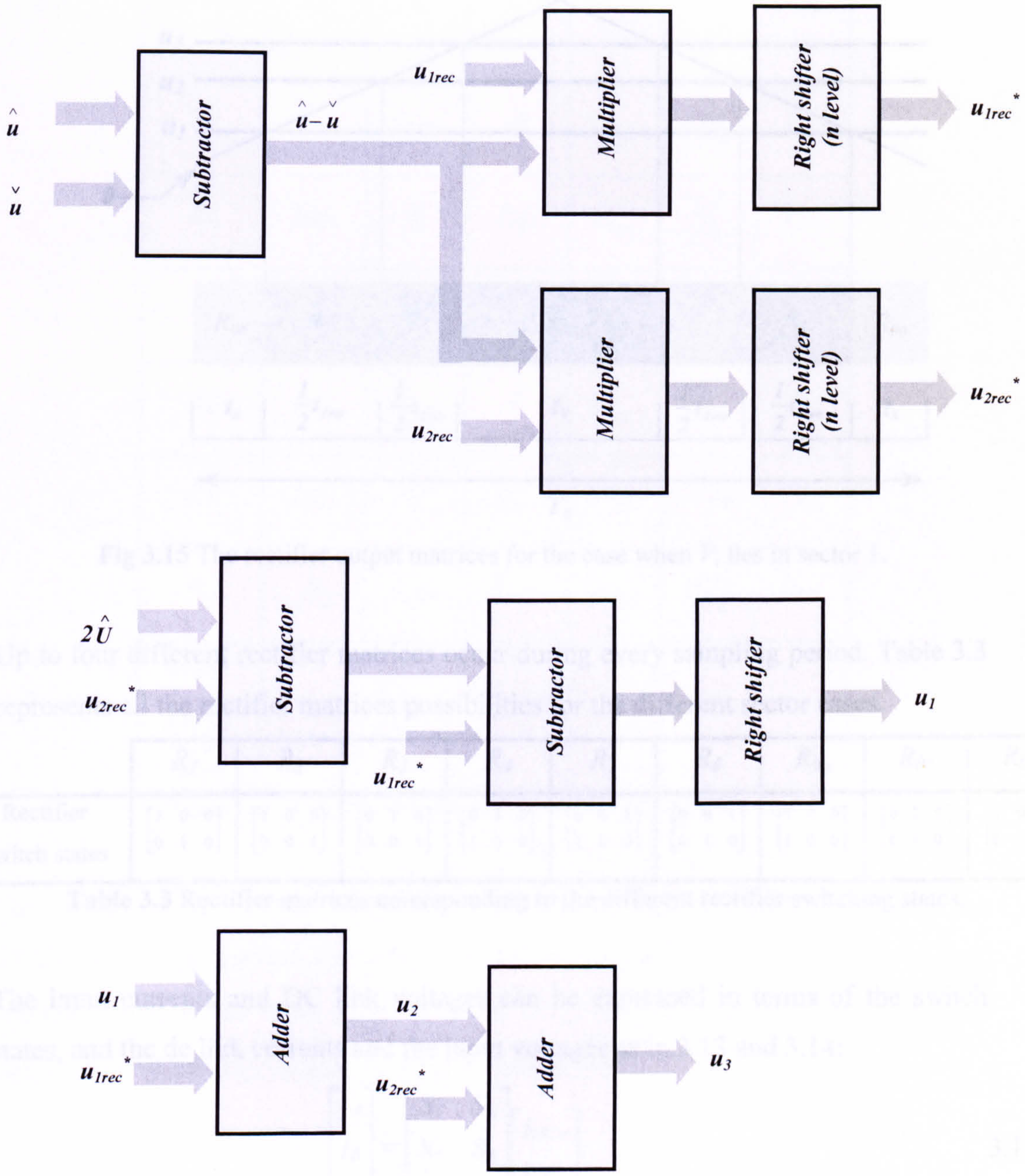


Fig. 3.14 Logic circuitry used to adjust the values for the active rectifier stage.

Fig 3.15 represents the actual time distribution for the rectifier matrices for the case in which the input voltage lies in sector 1.

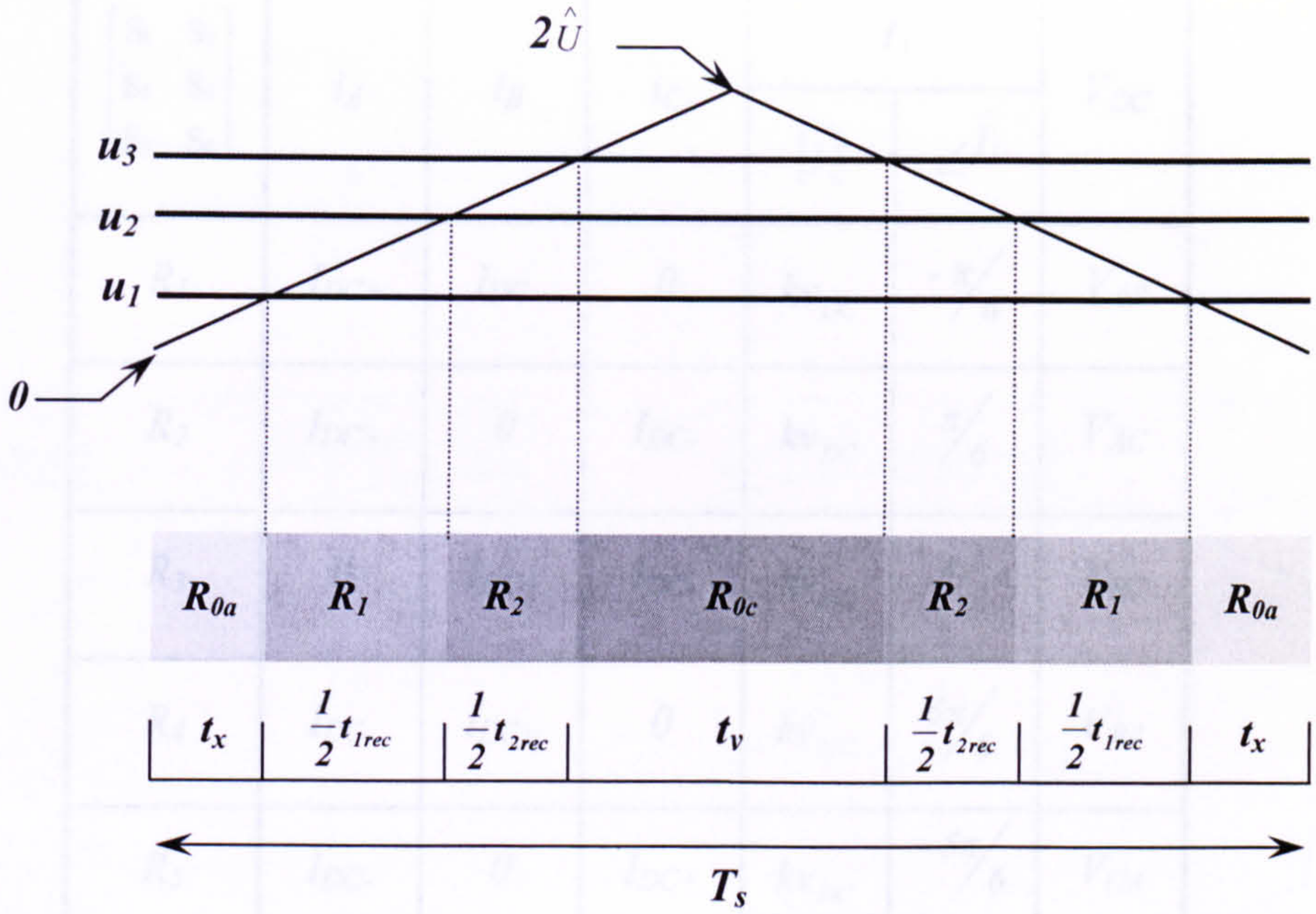


Fig 3.15 The rectifier output matrices for the case when V_i lies in sector 1.

Up to four different rectifier matrices occur during every sampling period. Table 3.3 represents all the rectifier matrices possibilities for the different sector cases.

	R_1	R_2	R_3	R_4	R_5	R_6	R_{0a}	R_{0b}	R_{0c}
Rectifier Switch states	$\begin{bmatrix}1&0&0\\0&1&0\end{bmatrix}$	$\begin{bmatrix}1&0&0\\0&0&1\end{bmatrix}$	$\begin{bmatrix}0&1&0\\0&0&1\end{bmatrix}$	$\begin{bmatrix}0&1&0\\1&0&0\end{bmatrix}$	$\begin{bmatrix}0&0&1\\1&0&0\end{bmatrix}$	$\begin{bmatrix}0&0&1\\0&1&0\end{bmatrix}$	$\begin{bmatrix}1&0&0\\1&0&0\end{bmatrix}$	$\begin{bmatrix}0&1&0\\0&1&0\end{bmatrix}$	$\begin{bmatrix}0&0&1\\0&0&1\end{bmatrix}$

Table 3.3 Rectifier matrices corresponding to the different rectifier switching states.

The input currents and DC link voltages can be expressed in terms of the switch states, and the dc link currents and the input voltages as in 3.13 and 3.14:

$$\begin{bmatrix} i_A \\ i_B \\ i_C \end{bmatrix} = \begin{bmatrix} S_1 & S_2 \\ S_3 & S_4 \\ S_5 & S_6 \end{bmatrix} \begin{bmatrix} i_{DC+} \\ i_{DC-} \end{bmatrix} \qquad 3.13$$

$$\begin{bmatrix} v_{DC+} \\ v_{DC-} \end{bmatrix} = \begin{bmatrix} S_1 & S_3 & S_5 \\ S_2 & S_4 & S_6 \end{bmatrix} \begin{bmatrix} v_A \\ v_B \\ v_C \end{bmatrix} \qquad 3.14$$

The relationship between the input values (voltages and currents) and the dc link values, are summarized in table 3.4

$\begin{bmatrix} S_1 & S_2 \\ S_3 & S_4 \\ S_5 & S_6 \end{bmatrix}$	i_A	i_B	i_C	$\bar{I}_i$		V_{DC}
				$\lvert\bar{I}_i\rvert$	$\angle\bar{I}_i$	
R_1	I_{DC+}	I_{DC-}	0	kv_{DC}	$-\pi/6$	V_{AB}
R_2	I_{DC+}	0	I_{DC-}	kv_{DC}	$\pi/6$	V_{AC}
R_3	0	I_{DC+}	I_{DC-}	kv_{DC}	$\pi/2$	V_{BC}
R_4	I_{DC-}	I_{DC+}	0	kv_{DC}	$5\pi/6$	V_{BA}
R_5	I_{DC-}	0	I_{DC+}	kv_{DC}	$-5\pi/6$	V_{CA}
R_6	0	I_{DC-}	I_{DC+}	kv_{DC}	$-\pi/2$	V_{CB}
R_{0a} or R_{0b} or R_{0c}	0	0	0	0	0	0

Table 3.4 Possible rectifier stage switch state combinations.

The quality of the output current from the inverter stage depends mainly on the robustness of the current controller and the virtual dc link voltage. The dc link voltage instantaneous value varies between $\pm v_{AB}, \pm v_{BC}, \pm v_{CA}$ and 0. The average dc link voltage within any sector can be calculated from:

$$V_{DC} = v_1.m_1 + v_2.m_2 \qquad 3.15$$

$$m_1 = \sin\left(\frac{\pi}{3} - \delta_i\right).\frac{T_{s\,eff}}{T_s} \qquad m_2 = \sin\delta_i.\frac{T_{s\,eff}}{T_s}$$

where

$$v_1 = \hat{V}_{iL}\cos(\delta_i + \phi) \qquad v_2 = \hat{V}_{iL}\cos\left(\delta_i + \phi - \frac{\pi}{3}\right)$$

which gives:

$$V_{DC} = \left(\hat{V}_{iL} \cos(\delta_i + \phi) \sin\left(\frac{\pi}{3} - \delta_i\right) + \hat{V}_{iL} \cos\left(\delta_i + \phi - \frac{\pi}{3}\right) \sin\delta_i \right) . \frac{T_{s\,eff}}{T_s}$$

$$= \hat{V}_{iL} . \frac{\sqrt{3}}{2} \cos\phi . \frac{T_{s\,eff}}{T_s} \qquad 3.16$$

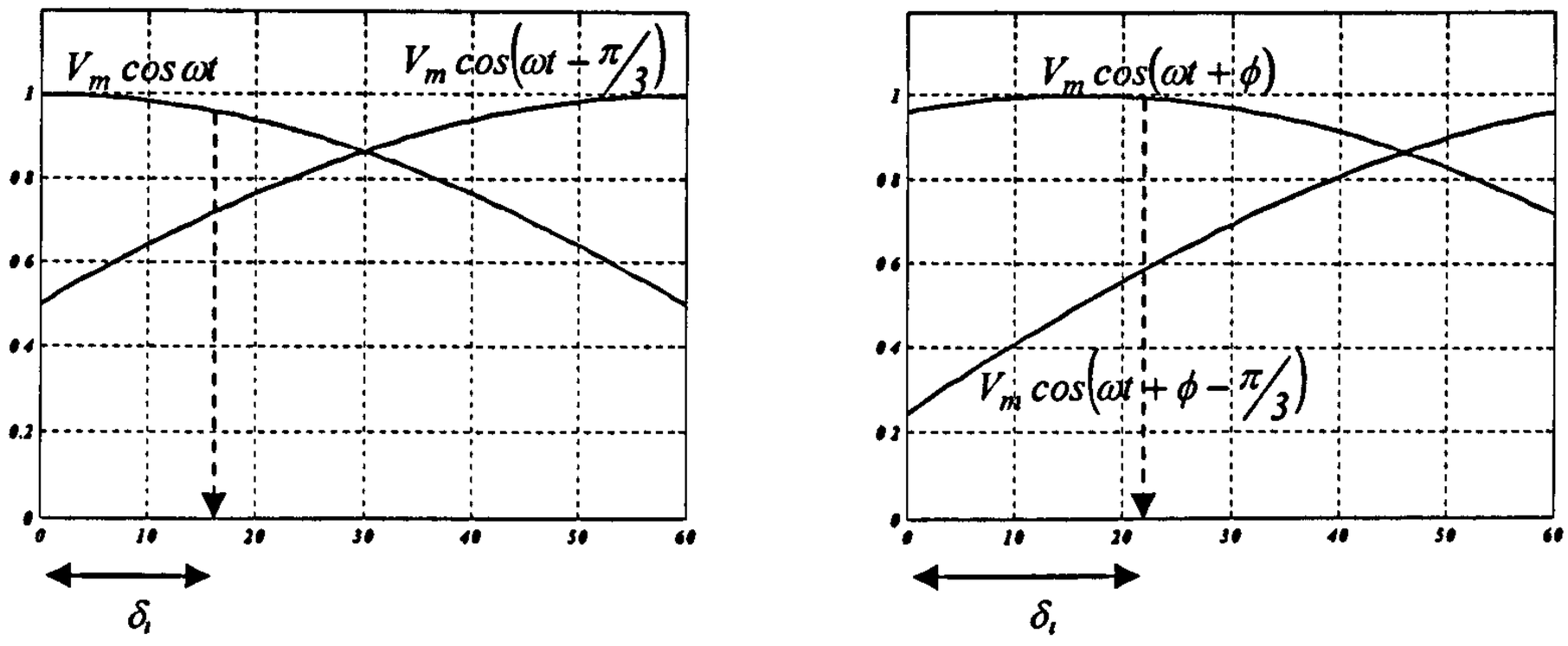


Fig. 3.16 Input voltages in each sector for:

a- Zero displacement angle and

b- ϕ displacement angle.

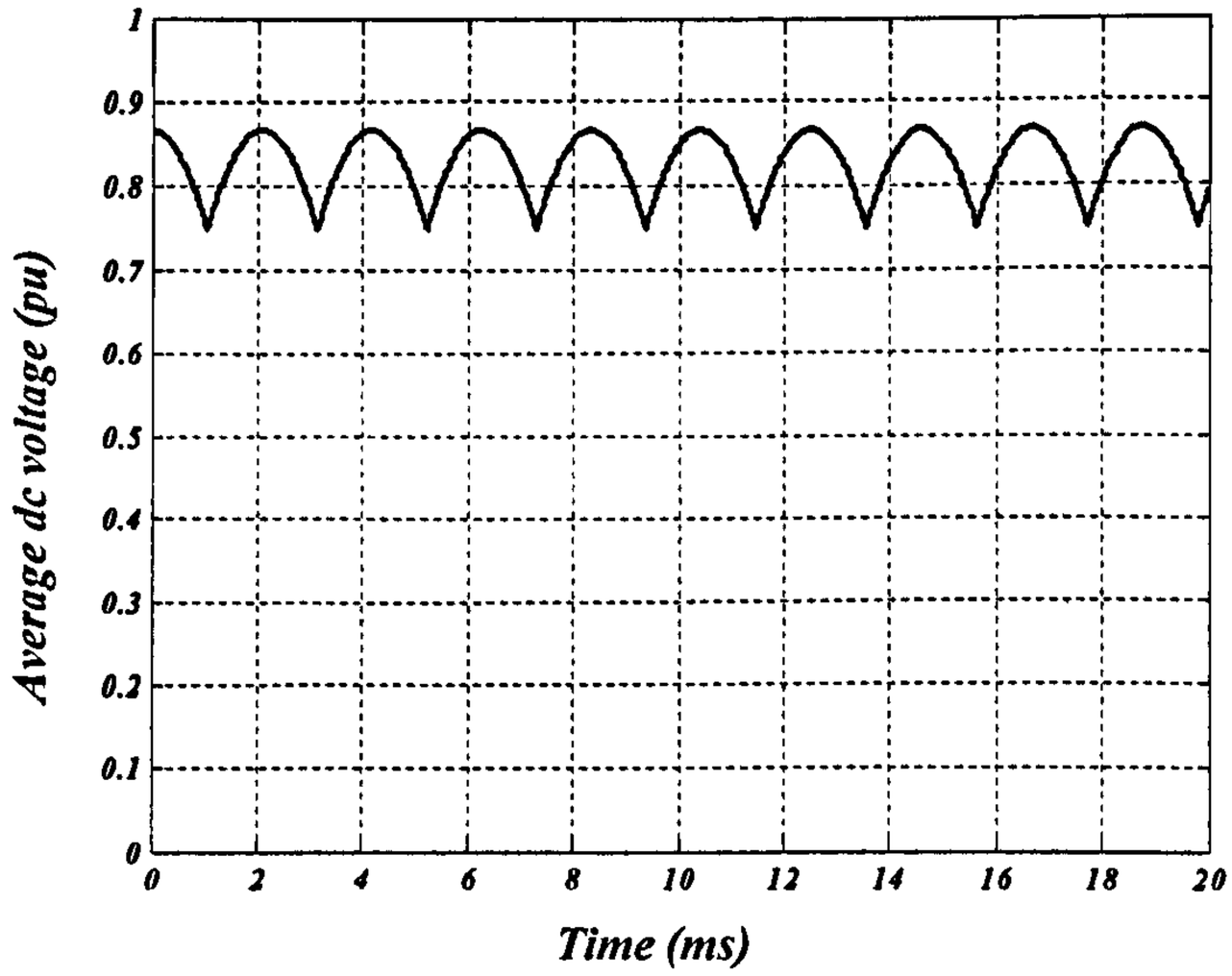


Fig. 3.17 The virtual dc link average voltage at unity modulation index and zero displacement angle for f_o=80 Hz.

By referring to figures 3.9 and 3.17 it can be proved that the virtual dc link voltage has the same shape as the maximum and minimum boundaries of the inverter stage. Also from equation 3.16, the average dc voltage is proportional to the effective sampling period. This means that at high output modulation indices (high $T_{s\,eff}$) the

dc link voltage has a high value and the opposite at low modulation indices. This guarantees valid operation of the inverter stage if a robust controller is used.

Another factor affecting the dc voltage is the input phase angle measurement. It must be correctly estimated in order to properly control the input displacement angle, hence the virtual dc voltage. Frequency and phase angle estimation are performed by transforming the input voltage vectors to the *d-q* frame. It is assumed that θ is the voltage vector angle, $\hat{\theta}$ is the estimated voltage vector angle, and that the grid voltage E has only a quadrature component as shown in Fig. 3.18. For small angle estimation error, the error signal Δ between to the actual and the estimated voltage E vectors, can be written as

$$\Delta = -\hat{E}\sin(\theta-\hat{\theta}) \approx -\hat{E}(\theta-\hat{\theta}) \approx -\hat{E}\frac{e_D}{e_Q} \tag{3.17}$$

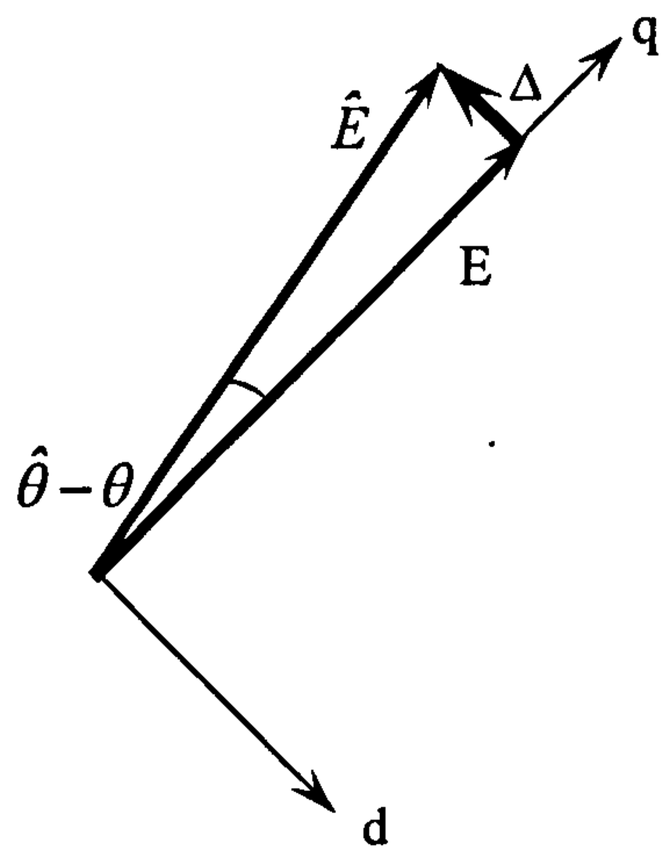


Fig. 3.18 Actual and estimated voltage vectors.

The error signal can be used to estimate both the estimated angular frequency $\hat{\omega}$ and the phase angle using gains A and B as follows [3.9]:

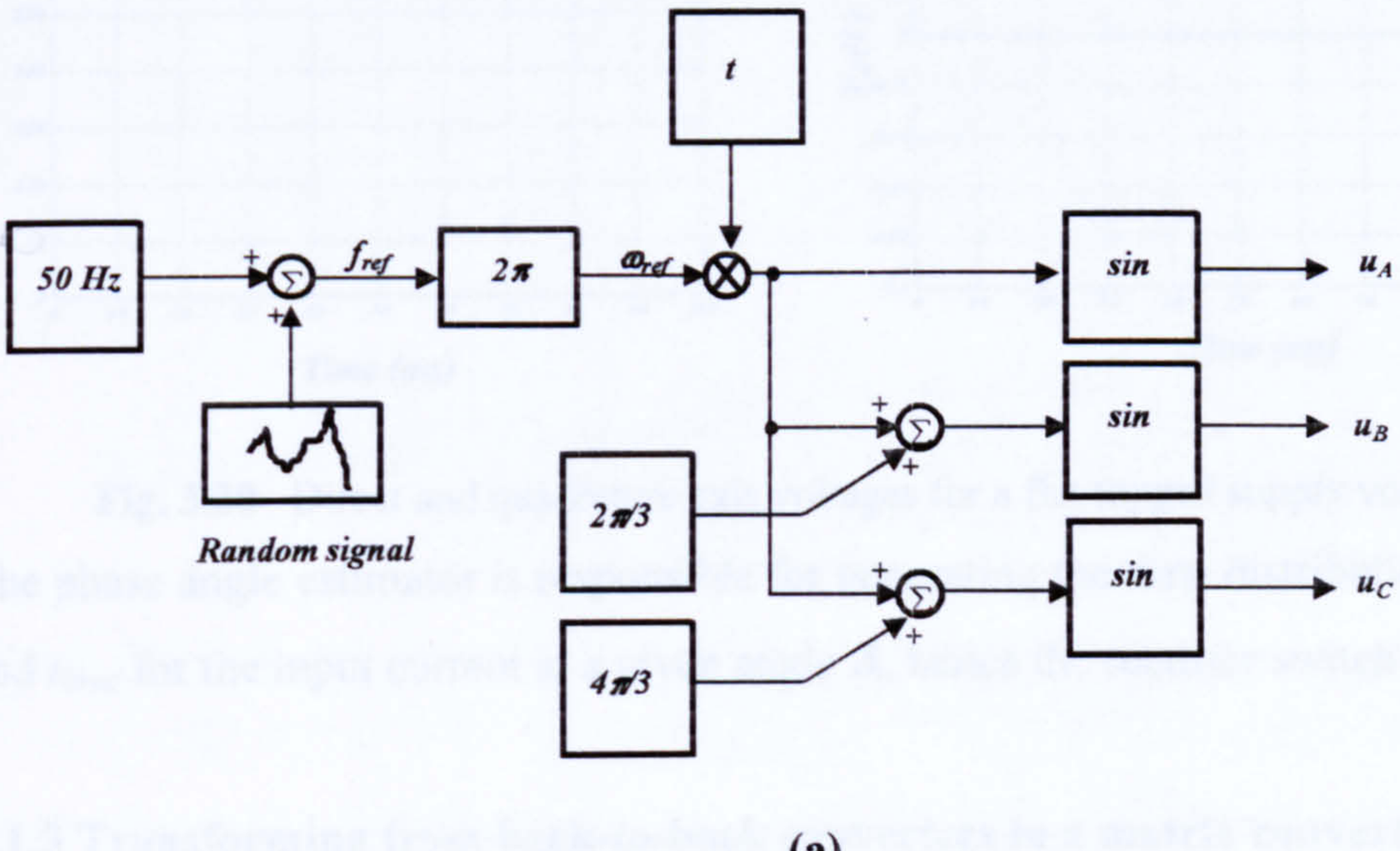
$$\frac{d\hat{\omega}}{dt} = A(\theta-\hat{\theta}) \approx -A\frac{e_D}{e_Q}$$
$$\frac{d\hat{\theta}}{dt} = \hat{\omega} + B(\theta-\hat{\theta}) \approx \hat{\omega} - B\frac{e_D}{e_Q} \tag{3.18}$$

$$\frac{d\hat{\omega}}{dt} \approx \frac{\omega_{k+1}-\omega_k}{T_s}$$
$$\frac{d\hat{\theta}}{dt} \approx \frac{\theta_{k+1}-\theta_k}{T_s} \tag{3.19}$$

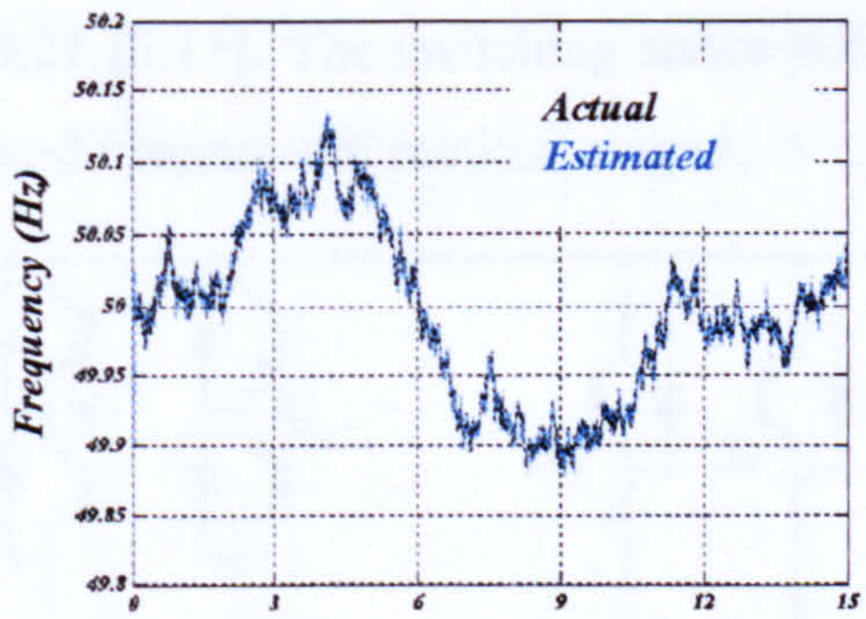
which lead to

$$\omega_{k+1} \approx \omega_k - A\frac{e_{D,k}}{e_{Q,k}}T_s$$
$$\theta_{k+1} \approx \theta_k + \omega_k T_s - B\frac{e_{D,k}}{e_{Q,k}}T_s \qquad 3.20$$

Values of A, B were arrived at by trial and error (*A=1136, B=34*), using the block diagram shown in Fig. 3.19 a. For a sinusoidal input voltage, the frequency estimator performs as shown in Fig. 3.19 b. The initial values for the angular frequency ω_0 and phase angle θ_0 are set at 314 rad/sec and zero rad respectively.



(a)



(b)

Fig. 3.19 Frequency estimator: (a) block diagram (b) steady state response.

But in the case of a flat-topped voltage source (a voltage source that contains fifth and seventh harmonics) the quadrature axis voltage v_q will contain sixth order

harmonic components as shown in Fig. 3.20. The direct axis voltage will remain null in the case of constant frequency since the voltage vector is aligned with the quadrature axis. This is also the case with an unbalanced power supply, where the quadrature axis voltage will contain a second order harmonic. For these reasons, the quadrature voltage should be filtered in order to correctly estimate the fundamental supply frequency.

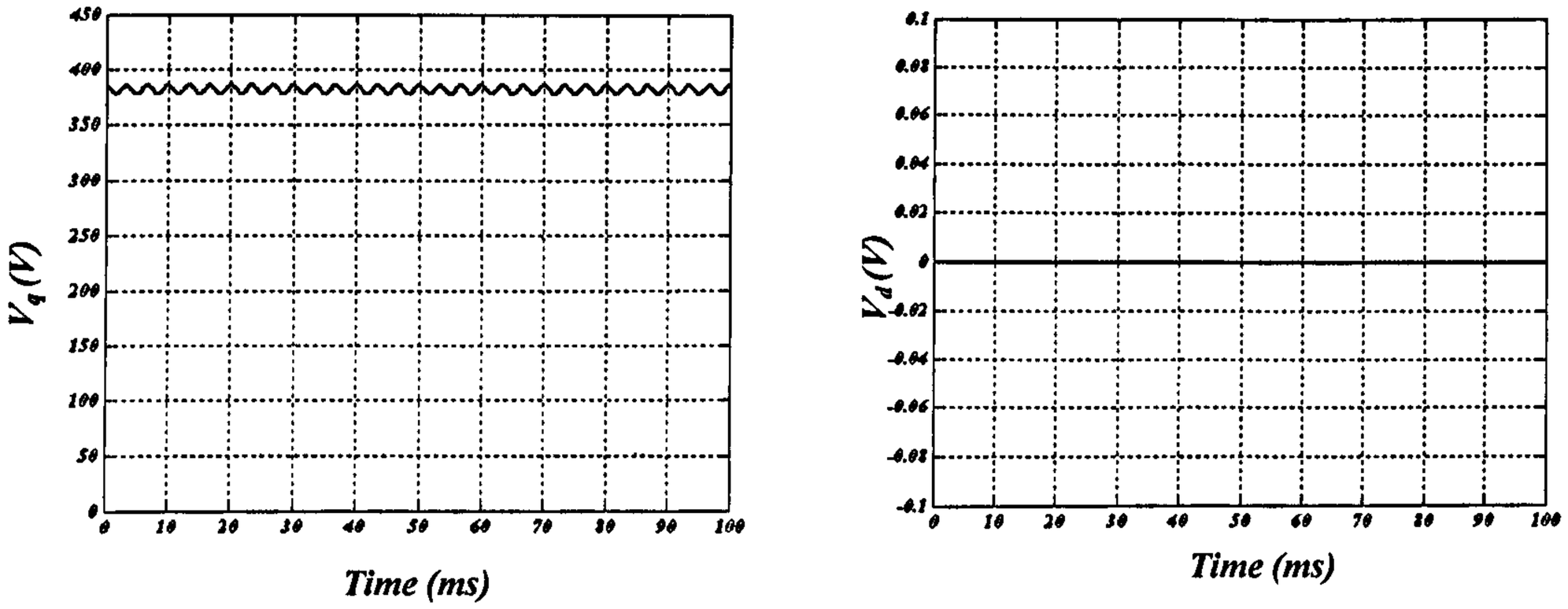


Fig. 3.20 Direct and quadrature axis voltages for a flat topped supply voltage.

The phase angle estimator is responsible for generating the time distribution t_{1rec}, t_{2rec} and t_{0rec} for the input current at a given angle δ_i, hence the rectifier switch states.

3.1.3 Transforming from back-to-back converters to a matrix converter

The previous switching strategy can be applied directly to the sparse matrix converter shown in Fig. 3.21 [3.11]. The switching states will be the same as those obtained from the considered inverter and rectifier stages.

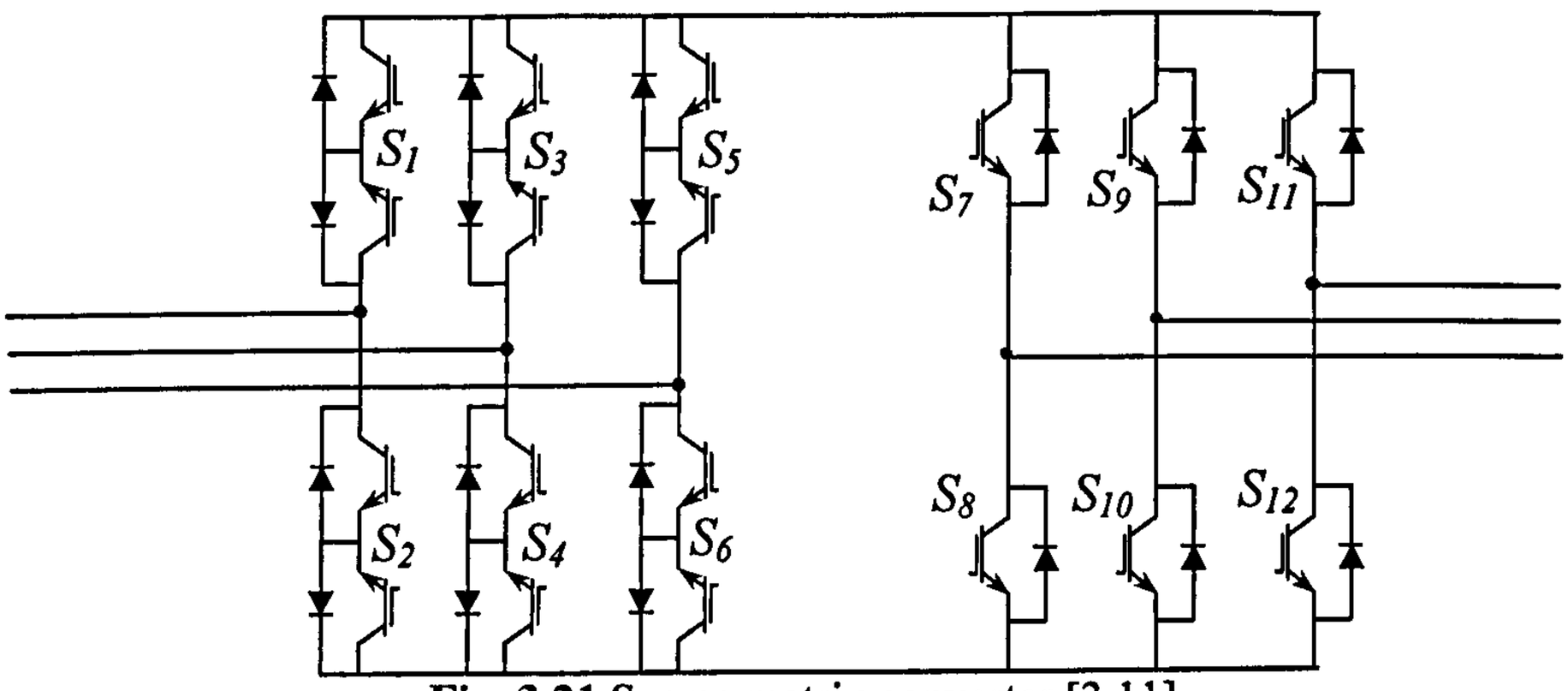


Fig. 3.21 Sparse matrix converter [3.11].

Where as considered matrix converter switching is needed for the conventional matrix converter shown in Fig. 3.22 since it is not constructed using a rectifier and an inverter,

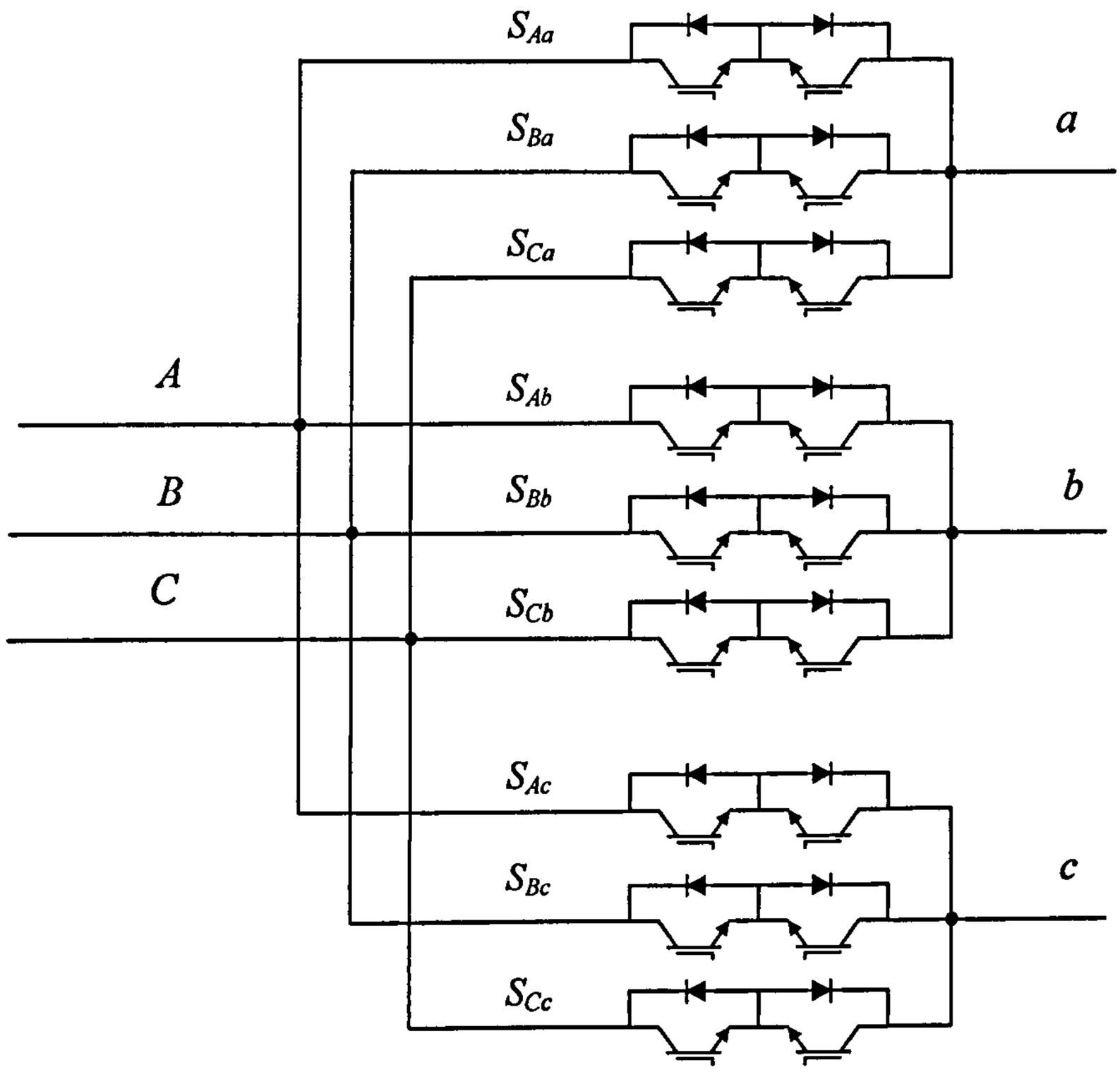


Fig. 3.22 Conventional matrix converter.

From equations 3.2, 3.3, 3.13, and 3.14, the relationship between the matrix converter and the corresponding rectifier and inverter matrices is derived as in equation 3.21

$$T = \begin{bmatrix} S_{Aa} & S_{Ba} & S_{Ca} \\ S_{Ab} & S_{Bb} & S_{Bc} \\ S_{Ac} & S_{Bc} & S_{Cc} \end{bmatrix} = \begin{bmatrix} S_7 & S_8 \\ S_9 & S_{10} \\ S_{11} & S_{12} \end{bmatrix} \begin{bmatrix} S_1 & S_3 & S_5 \\ S_2 & S_4 & S_6 \end{bmatrix} \qquad 3.21$$

$$T = \begin{bmatrix} S_7 S_1 + S_8 S_2 & S_7 S_3 + S_8 S_4 & S_7 S_5 + S_8 S_6 \\ S_9 S_1 + S_{10} S_2 & S_9 S_3 + S_{10} S_4 & S_9 S_5 + S_{10} S_6 \\ S_{11} S_1 + S_{12} S_2 & S_{11} S_3 + S_{12} S_4 & S_{11} S_5 + S_{12} S_6 \end{bmatrix}$$

Instead of using multiplication, logic circuits can be used since all the terms of the rectifier and inverter matrices are zeros or ones. Considering switch S_{Aa}, each element in this matrix can be realised by using logic circuitry consisting of 2 AND gates and 1 OR gate as shown in Fig. 3.23. This means that the corresponding matrix converter control signals can be generated without any need for further calculation. Based on the available cited references, this is the first time that multiplication of the virtual inverter and rectifier matrices has been realized using logic circuits.

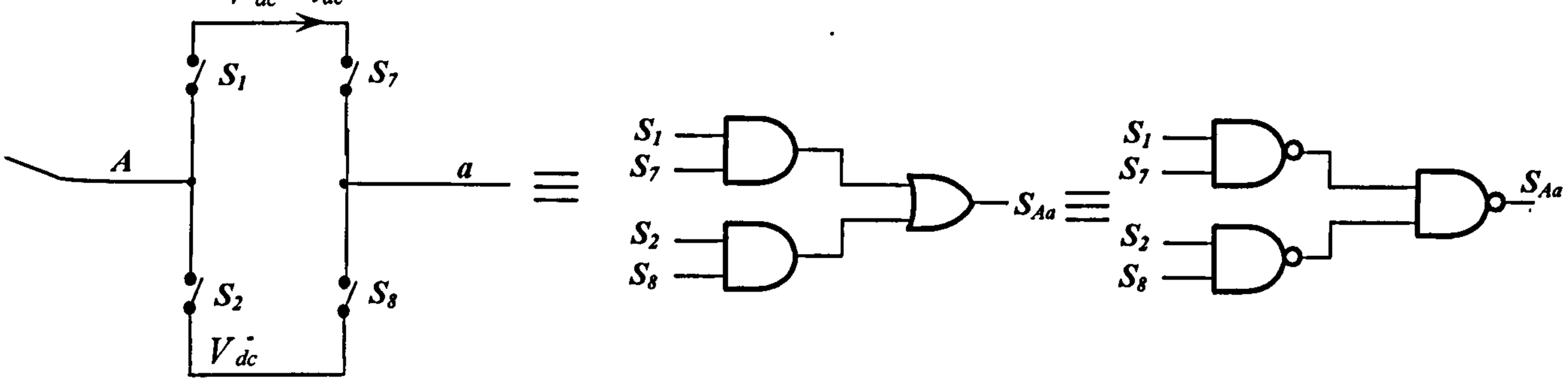


Fig 3.23 Logic circuitry to transform from back to back converters to the matrix converter.

The switching pattern during a sampling period when the input voltage lies in sector 1 and $U > u_a > u_b > u_c > -U$ is represented in Fig. 3.24. Nine different matrices can be obtained from the multiplication of the inverter and rectifier matrices (logic circuitry is used to perform this multiplication). From this diagram, it can be concluded that the number of commutations per cycle increased to 11 rather than 9 in the space vector modulation technique described in chapter 2.

The procedure is summarized in the flow chart in Fig. 3.23. The most important features of the technique are

- minimal calculations are required to produce the inverter current controller outputs and the rectifier timing distributions. Most functional realization can be achieved using logic circuits.
- The output matrices are generated automatically.

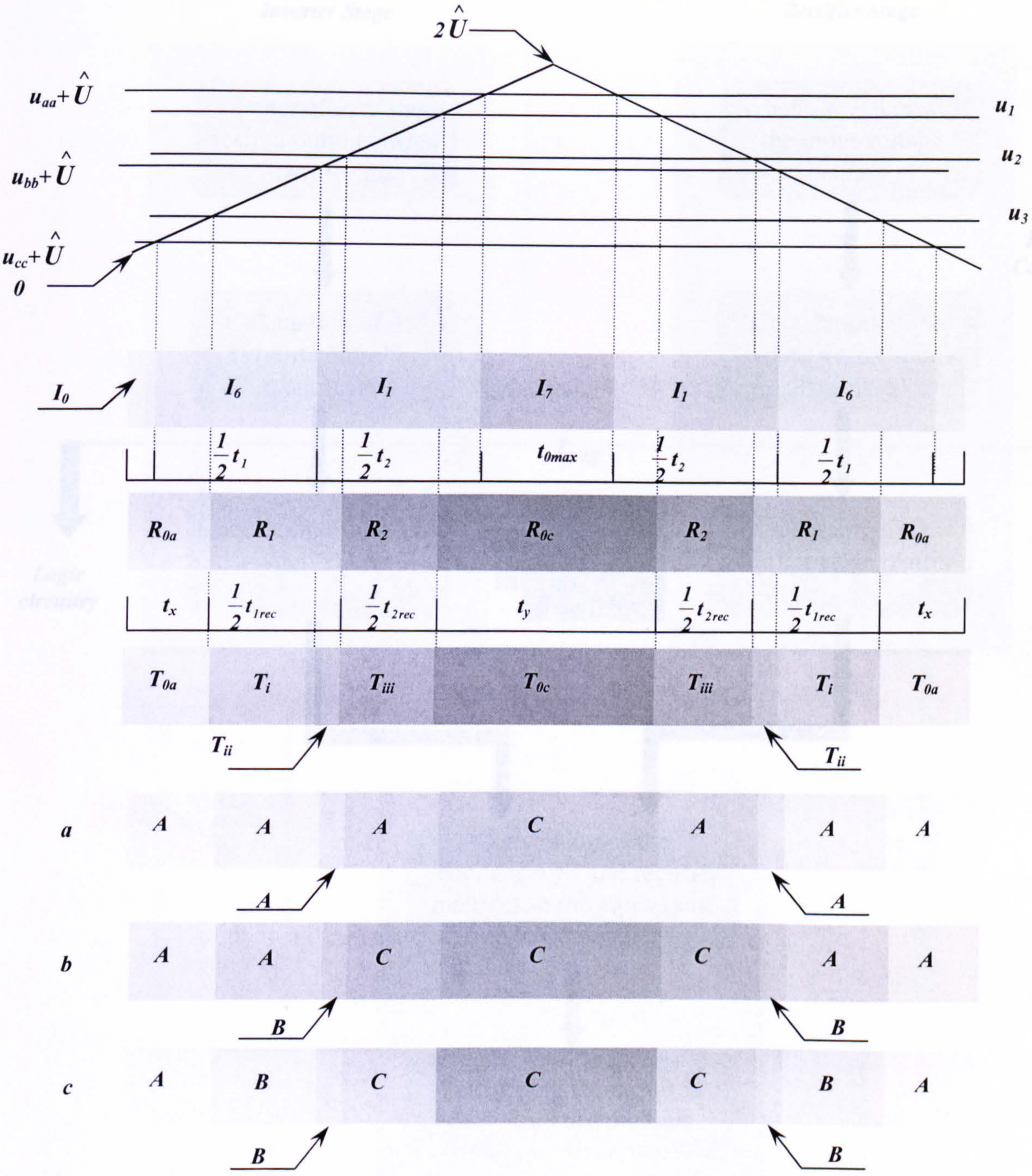


Fig 3.24 Switching pattern during a sampling period when the input voltage lies in sector 1 and $\hat{U} > u_a > u_b > u_c > -\hat{U}$.

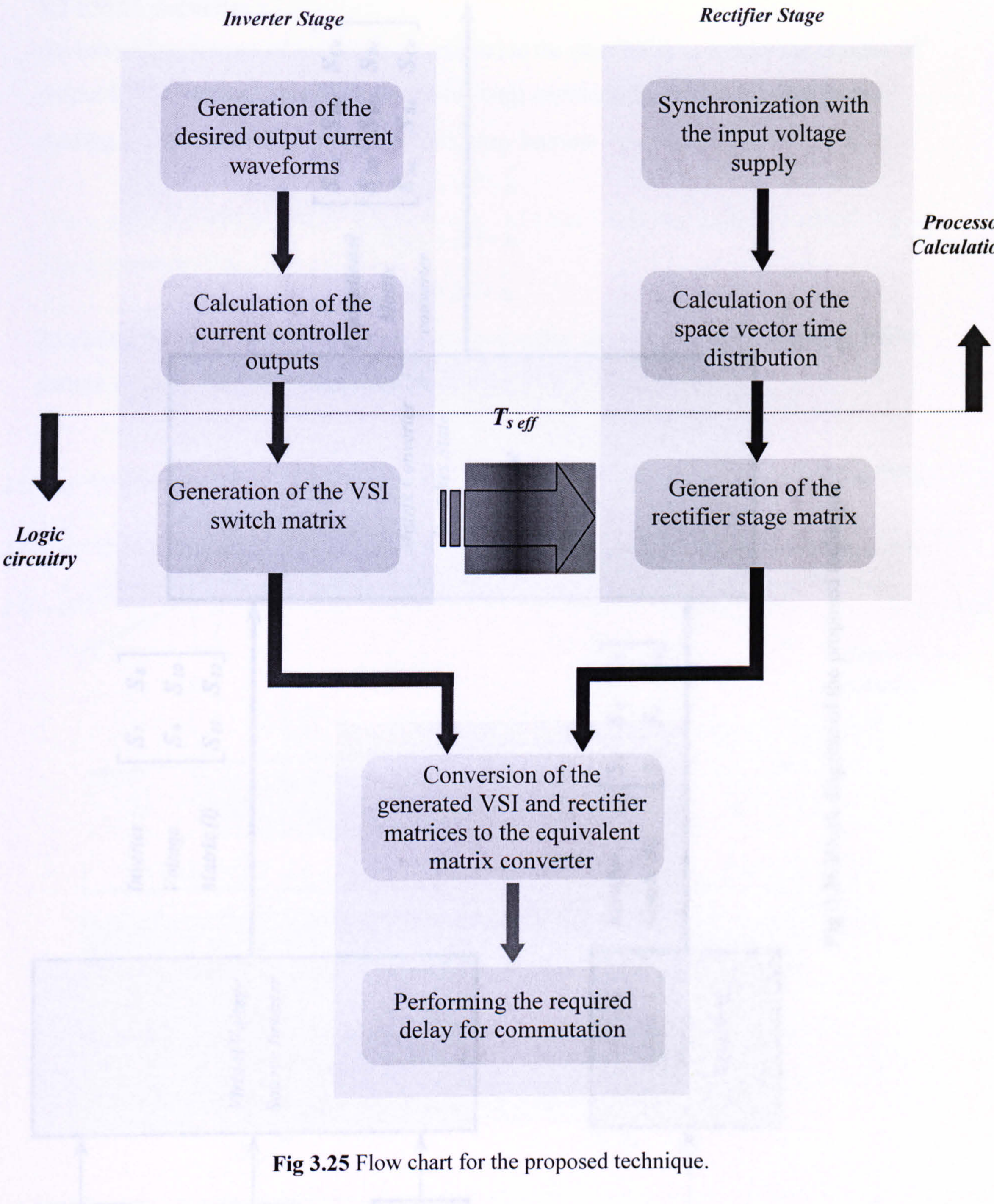


Fig 3.25 Flow chart for the proposed technique.

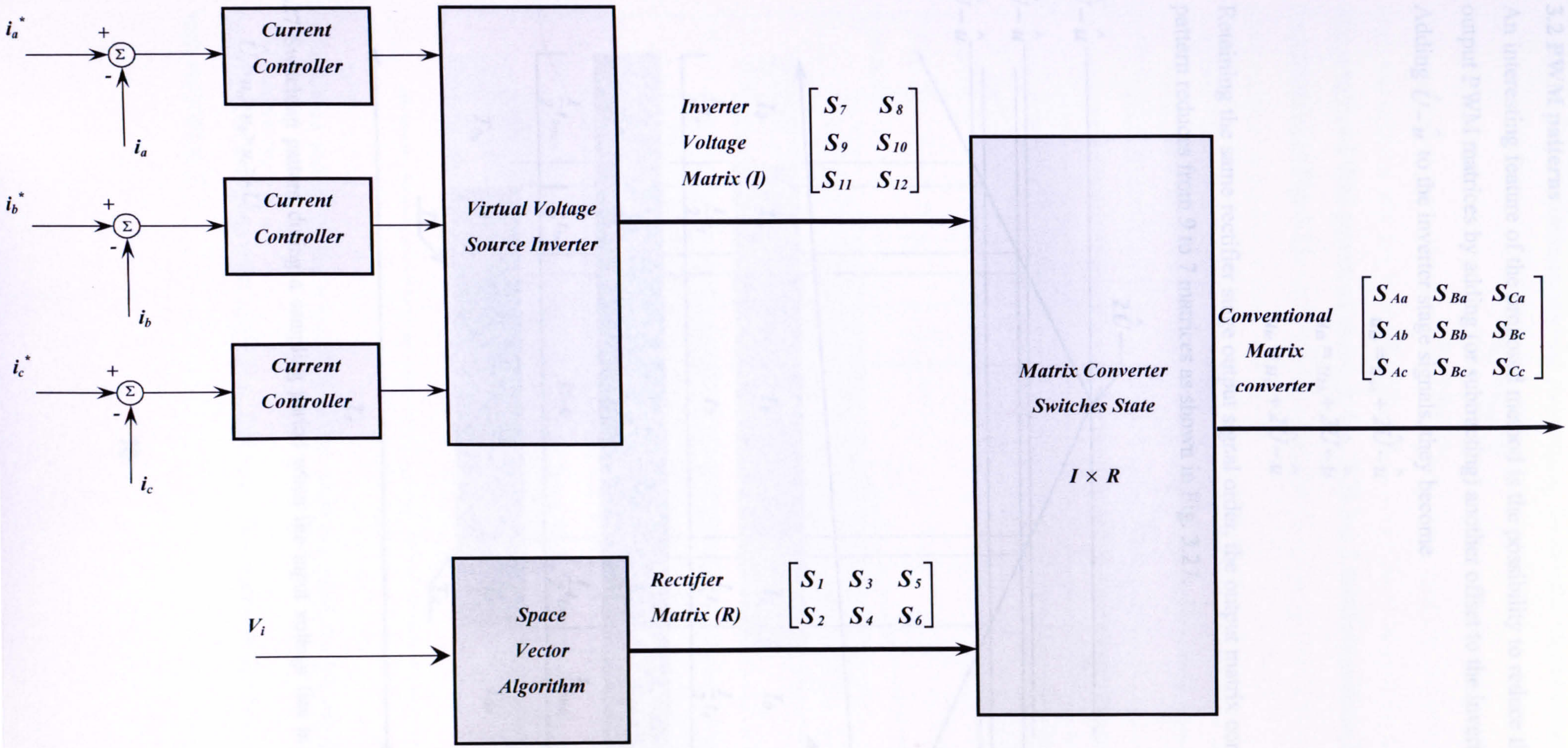


Fig 3.26 Block diagram of the proposed technique.

3.2 PWM patterns

An interesting feature of the proposed method is the possibility to reduce the number of output PWM matrices by adding (or subtracting) another offset to the inverter stage.

Adding $\hat{U}-\hat{u}$ to the inverter stage signals, they become

$$\begin{aligned} u_{ia} &= u_{aa} + 2\hat{U} - \hat{u} \\ u_{ib} &= u_{bb} + 2\hat{U} - \hat{u} \\ u_{ic} &= u_{cc} + 2\hat{U} - \hat{u} \end{aligned} \qquad 3.22$$

Retaining the same rectifier stage output signal order, the output matrix converter PWM pattern reduces from 9 to 7 matrices as shown in Fig. 3.27.

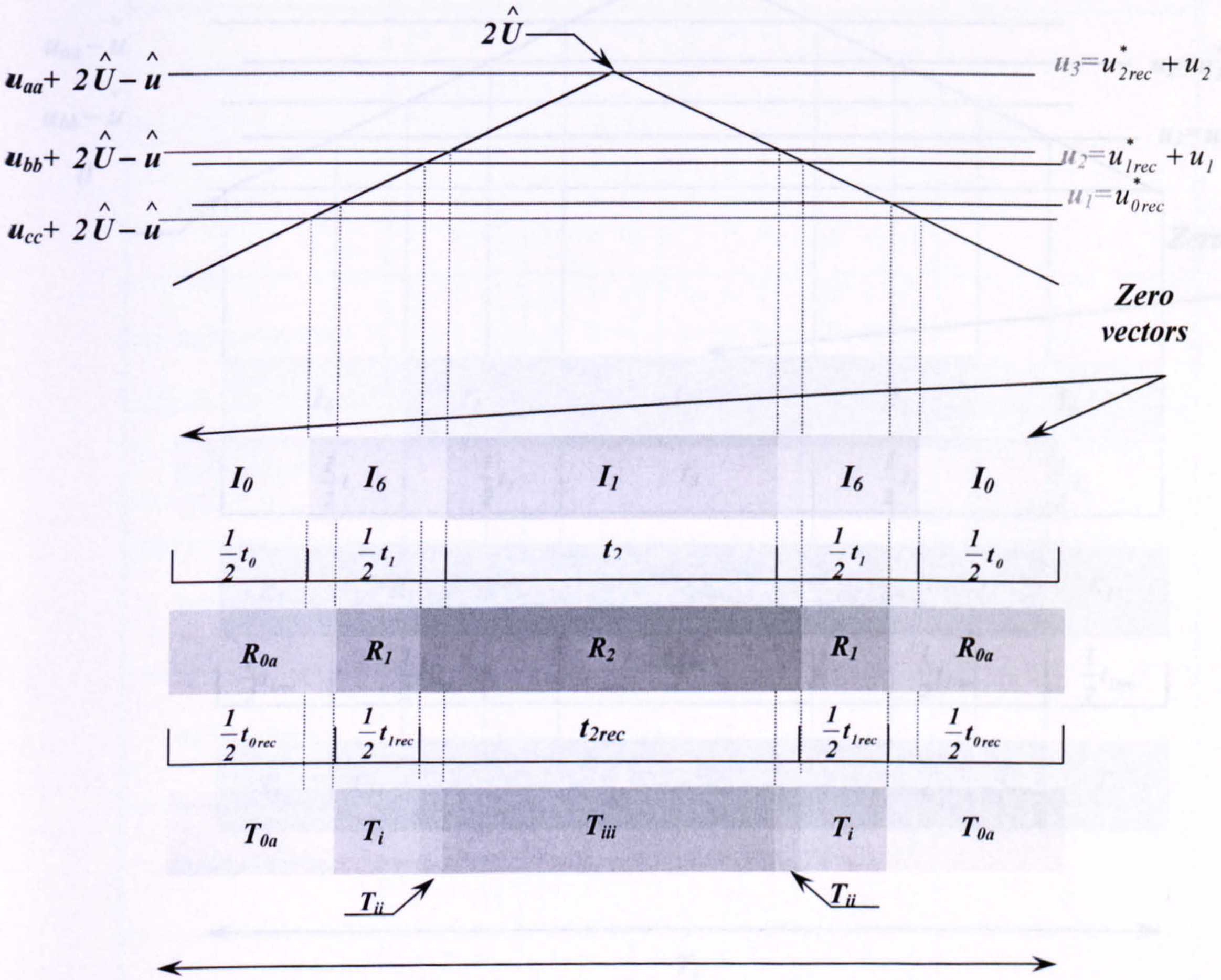


Fig 3.27 Switching pattern during a sampling period when the input voltage lies in sector 1 and $\hat{U} > u_a > u_b > u_c > -\hat{U}$.

By subtracting the minimum inverter stage signal, the inverter stage outputs become

$$\begin{aligned} u_{ia} &= u_{aa} - \overset{\vee}{u} \\ u_{ib} &= u_{bb} - \overset{\vee}{u} \\ u_{ic} &= u_{cc} - \overset{\vee}{u} \end{aligned} \qquad 3.23$$

The resultant PWM pattern as shown in Fig. 3.28 has 7 matrices patterns instead of the 9 matrices as in Fig. 3.23.

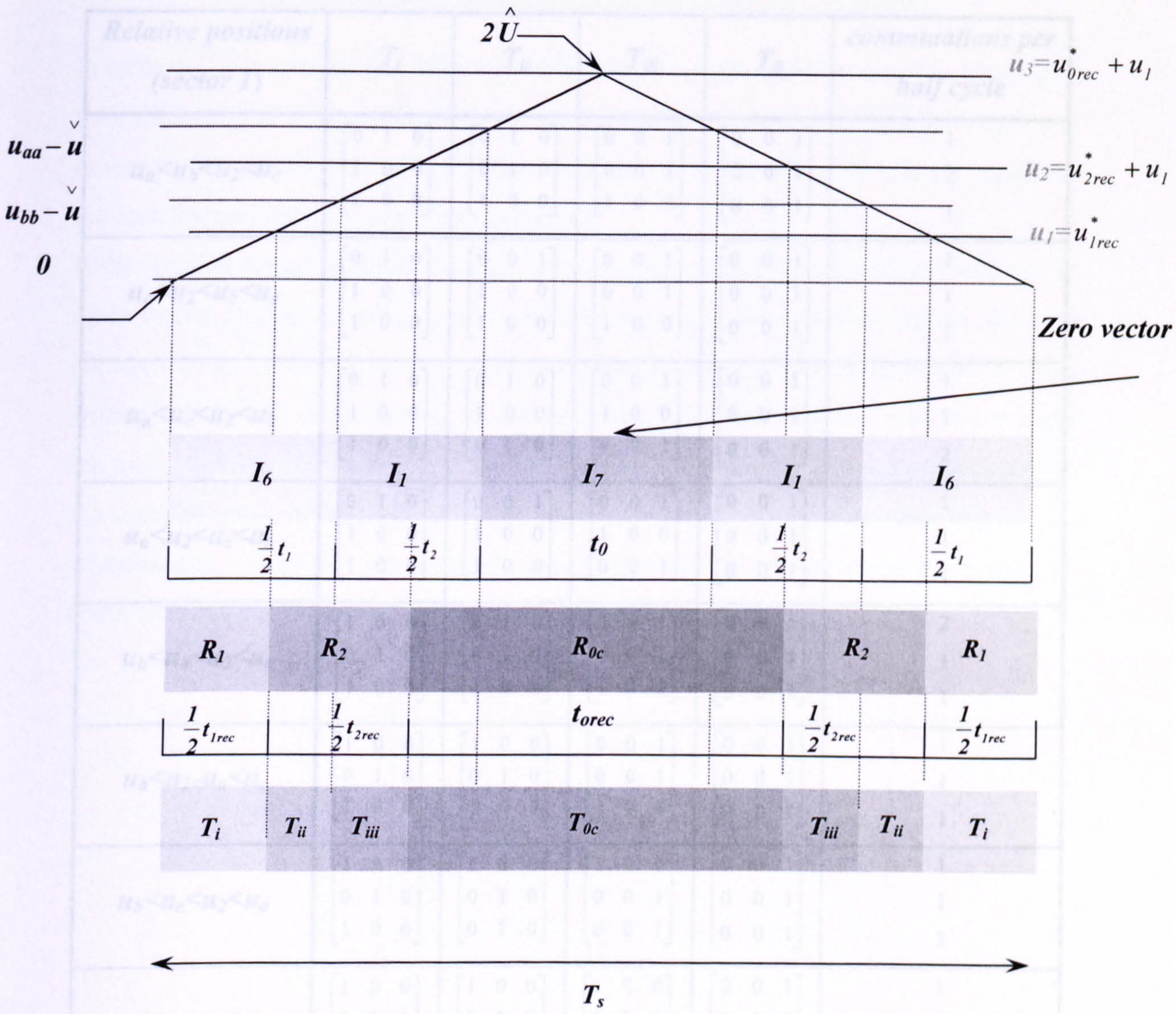


Fig 3.28 Switching pattern during a sampling period when the input voltage lies in sector 1 and $\hat{U} > u_a > u_b > u_c > -\hat{U}$.

Based on the available cited references, this is the first time the number of matrices has been reduced to 7.

3.3 Number of commutations per cycle

One of the main problems associated with the control method explained is the number of commutations per cycle for each phase. If the final matrices are generated directly from the rectifier and the inverter matrices, there will be no control on the number of commutations associated with each switch per cycle. Table 3.5 shows switch commutations for the twelve relative magnitudes of u_a, u_b, u_c and u_2 assuming that the input voltage vector lies in the 1st sector.

Relative positions (sector 1)	T_i	T_{ii}	T_{iii}	T_0	*commutations per half cycle*
$u_a<u_b<u_2<u_c$	$\begin{bmatrix}0&1&0\\1&0&0\\1&0&0\end{bmatrix}$	$\begin{bmatrix}0&1&0\\0&1&0\\1&0&0\end{bmatrix}$	$\begin{bmatrix}0&0&1\\0&0&1\\1&0&0\end{bmatrix}$	$\begin{bmatrix}0&0&1\\0&0&1\\0&0&1\end{bmatrix}$	1 2 1
$u_a<u_2<u_b<u_c$	$\begin{bmatrix}0&1&0\\1&0&0\\1&0&0\end{bmatrix}$	$\begin{bmatrix}0&0&1\\1&0&0\\1&0&0\end{bmatrix}$	$\begin{bmatrix}0&0&1\\0&0&1\\1&0&0\end{bmatrix}$	$\begin{bmatrix}0&0&1\\0&0&1\\0&0&1\end{bmatrix}$	1 1 1
$u_a<u_c<u_2<u_b$	$\begin{bmatrix}0&1&0\\1&0&0\\1&0&0\end{bmatrix}$	$\begin{bmatrix}0&1&0\\1&0&0\\0&1&0\end{bmatrix}$	$\begin{bmatrix}0&0&1\\1&0&0\\0&0&1\end{bmatrix}$	$\begin{bmatrix}0&0&1\\0&0&1\\0&0&1\end{bmatrix}$	1 1 2
$u_a<u_2<u_c<u_b$	$\begin{bmatrix}0&1&0\\1&0&0\\1&0&0\end{bmatrix}$	$\begin{bmatrix}0&0&1\\1&0&0\\1&0&0\end{bmatrix}$	$\begin{bmatrix}0&0&1\\1&0&0\\0&0&1\end{bmatrix}$	$\begin{bmatrix}0&0&1\\0&0&1\\0&0&1\end{bmatrix}$	1 1 1
$u_b<u_a<u_2<u_c$	$\begin{bmatrix}1&0&0\\0&1&0\\1&0&0\end{bmatrix}$	$\begin{bmatrix}0&1&0\\0&1&0\\1&0&0\end{bmatrix}$	$\begin{bmatrix}0&0&1\\0&0&1\\1&0&0\end{bmatrix}$	$\begin{bmatrix}0&0&1\\0&0&1\\0&0&1\end{bmatrix}$	2 1 1
$u_b<u_2<u_a<u_c$	$\begin{bmatrix}1&0&0\\0&1&0\\1&0&0\end{bmatrix}$	$\begin{bmatrix}1&0&0\\0&1&0\\1&0&0\end{bmatrix}$	$\begin{bmatrix}0&0&1\\0&0&1\\1&0&0\end{bmatrix}$	$\begin{bmatrix}0&0&1\\0&0&1\\0&0&1\end{bmatrix}$	1 1 1
$u_b<u_c<u_2<u_a$	$\begin{bmatrix}1&0&0\\0&1&0\\1&0&0\end{bmatrix}$	$\begin{bmatrix}1&0&0\\0&1&0\\0&1&0\end{bmatrix}$	$\begin{bmatrix}1&0&0\\0&0&1\\0&0&1\end{bmatrix}$	$\begin{bmatrix}0&0&1\\0&0&1\\0&0&1\end{bmatrix}$	1 1 2
$u_b<u_2<u_c<u_a$	$\begin{bmatrix}1&0&0\\0&1&0\\1&0&0\end{bmatrix}$	$\begin{bmatrix}1&0&0\\0&1&0\\1&0&0\end{bmatrix}$	$\begin{bmatrix}1&0&0\\0&0&1\\0&0&1\end{bmatrix}$	$\begin{bmatrix}0&0&1\\0&0&1\\0&0&1\end{bmatrix}$	1 1 1
$u_c<u_b<u_2<u_a$	$\begin{bmatrix}1&0&0\\1&0&0\\0&1&0\end{bmatrix}$	$\begin{bmatrix}1&0&0\\0&1&0\\0&1&0\end{bmatrix}$	$\begin{bmatrix}1&0&0\\0&0&1\\0&0&1\end{bmatrix}$	$\begin{bmatrix}0&0&1\\0&0&1\\0&0&1\end{bmatrix}$	1 2 1

$u_c<u_2<u_b<u_a$	$\begin{bmatrix}1&0&0\\1&0&0\\0&1&0\end{bmatrix}$	$\begin{bmatrix}1&0&0\\1&0&0\\0&0&1\end{bmatrix}$	$\begin{bmatrix}1&0&0\\0&0&1\\0&0&1\end{bmatrix}$	$\begin{bmatrix}0&0&1\\0&0&1\\0&0&1\end{bmatrix}$	1 1 1
$u_c<u_a<u_2<u_b$	$\begin{bmatrix}1&0&0\\1&0&0\\0&1&0\end{bmatrix}$	$\begin{bmatrix}0&1&0\\1&0&0\\0&1&0\end{bmatrix}$	$\begin{bmatrix}0&0&1\\1&0&0\\0&0&1\end{bmatrix}$	$\begin{bmatrix}0&0&1\\0&0&1\\0&0&1\end{bmatrix}$	2 1 1
$u_c<u_2<u_a<u_b$	$\begin{bmatrix}1&0&0\\1&0&0\\0&1&0\end{bmatrix}$	$\begin{bmatrix}1&0&0\\1&0&0\\0&0&1\end{bmatrix}$	$\begin{bmatrix}0&0&1\\1&0&0\\0&0&1\end{bmatrix}$	$\begin{bmatrix}0&0&1\\0&0&1\\0&0&1\end{bmatrix}$	1 1 1

Table 3.5 Number of commutations per switch per cycle in sector 1.

In table 3.5 it is noticed that if u_2 (the middle signal of the rectifier stage) is lower than the middle signal of the inverter stage, the maximum commutations per half cycle is 1 otherwise the maximum number of commutations is 2. Therefore the middle signals of the rectifier and inverter stages are compared.

If the rectifier signal is higher than the inverter stage signal, the signals are swapped. In other words, instead of using t_0, t_0+t_1 and $t_0+t_1+t_2$ as the three rectifier signals, they are changed to t_0, t_0+t_2 and $t_0+t_1+t_2$. This reduces the number of commutations to 1 for approximately half the switching, as shown in fig. 3.29. This function is achieved using the logic circuitry shown in Fig. 3.30. The inverter stage middle signal is compared with the rectifier stage middle signal. If the latter is smaller, the rectifier stage signals are interchanged. . Based on the available cited references, this is the minimum number of commutations per switching cycle that can be achieved.

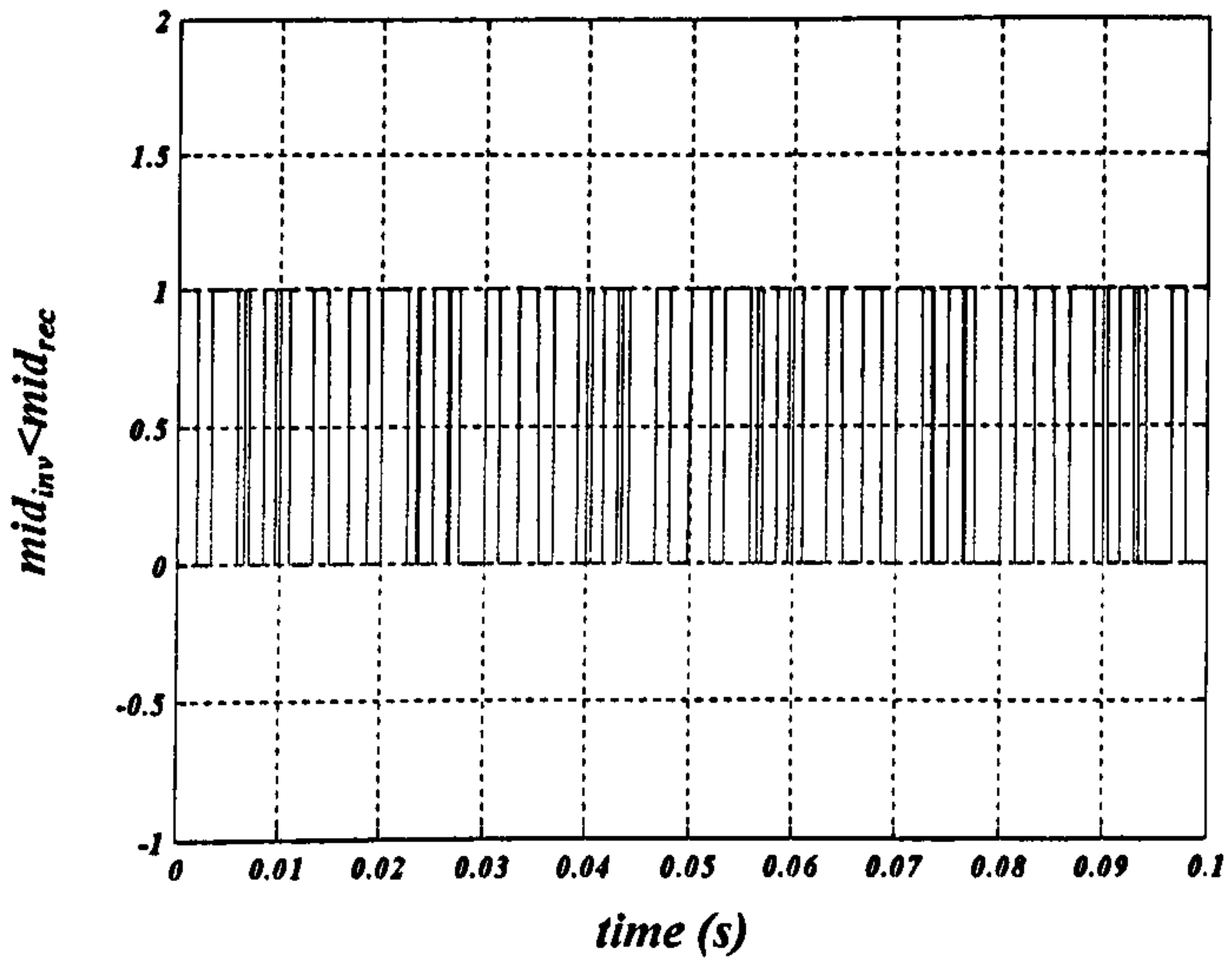


Fig. 3.29 Relative position of the inverter and rectifier mid signals

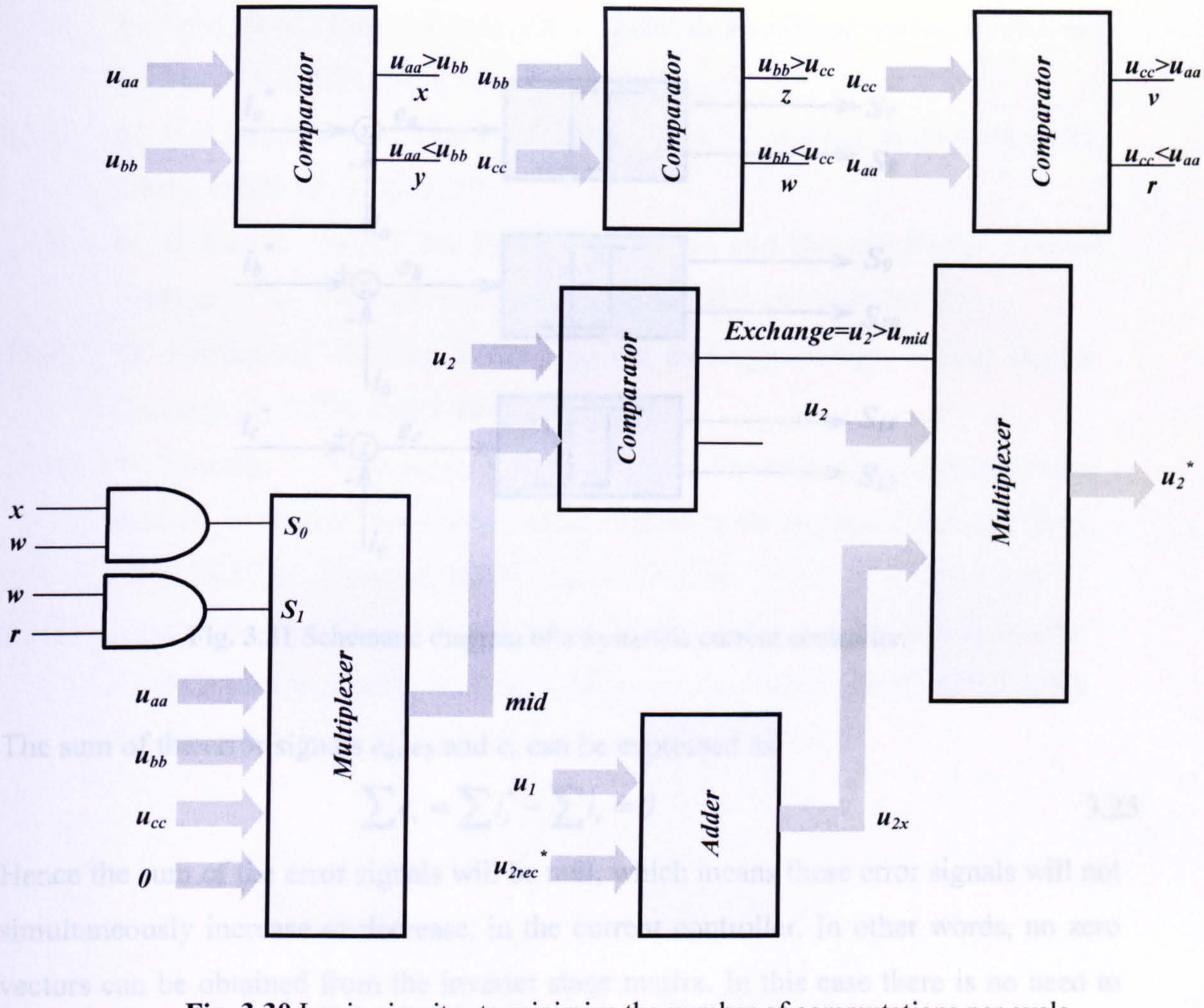


Fig. 3.30 Logic circuitry to minimize the number of commutations per cycle.

3.4 Hysteresis controller

The control technique can be implemented with the hysteresis current controller shown in Fig. 3.31. Assuming a three-phase three-wire system, the inputs to the controller are the difference between the output current commands i_a^*, i_b^*, and i_c^* and the feedback output currents i_a, i_b, and i_c. In its simplest form, the hysteresis controller for phase a will be

$$S_7 = \begin{cases} 1 & i_a^* - i_a > i_{threshold} \\ 0 & i_a^* - i_a < -i_{threshold} \\ NC & -i_{threshold} \leq i_a^* - i_a \leq i_{threshold} \end{cases} \qquad 3.24$$

$$S_8 = \overline{S_7}$$

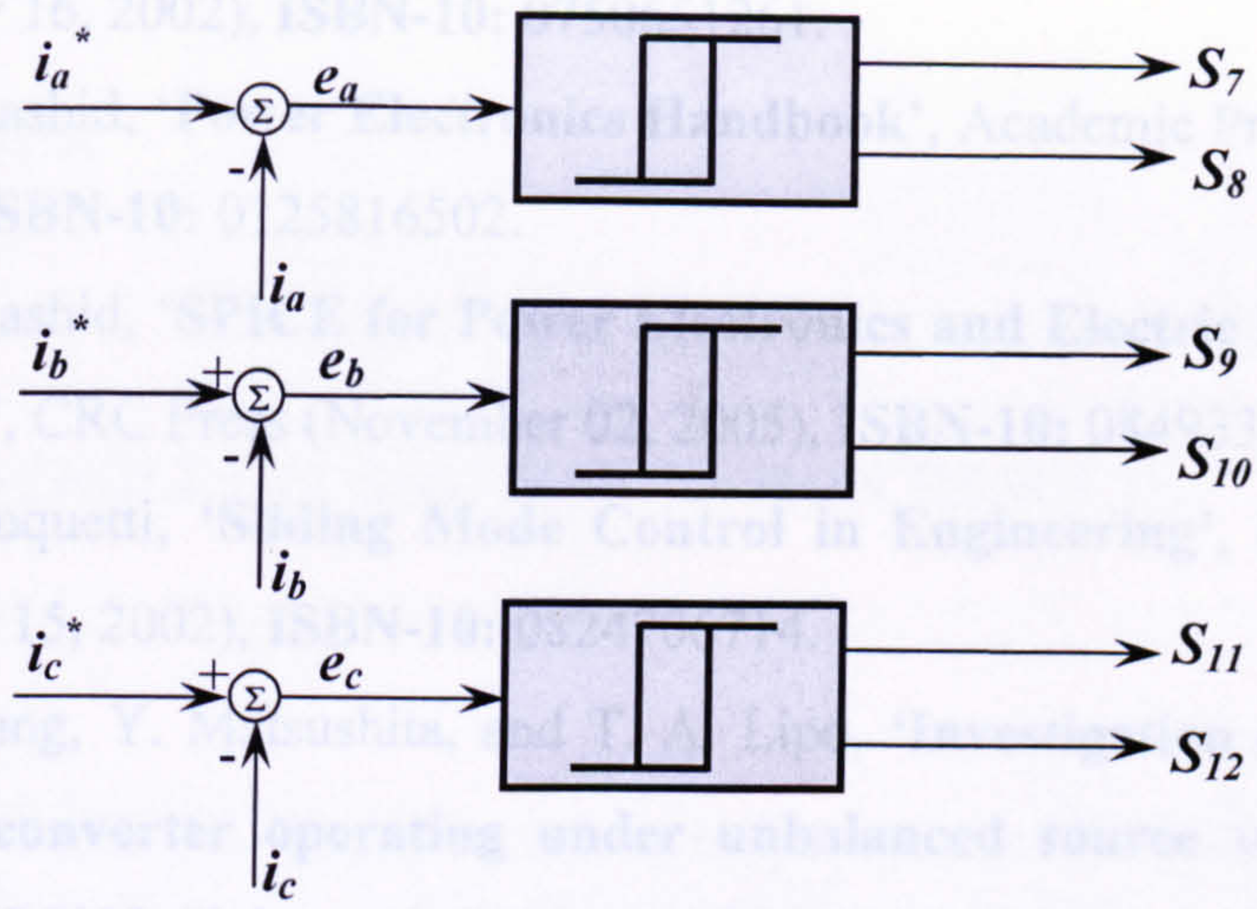


Fig. 3.31 Schematic diagram of a hysterisis current controller.

The sum of the error signals e_a, e_b and e_c can be expressed as

$$\sum e_i = \sum i_i^* - \sum i_i = 0 \qquad 3.25$$

Hence the sum of the error signals will be null, which means these error signals will not simultaneously increase or decrease, in the current controller. In other words, no zero vectors can be obtained from the inverter stage matrix. In this case there is no need to apply $T_{s\,eff}$ to modify the rectifier stage levels.

References:

[3.1] R. W. Erickson, D. Maksimovic, **'Fundamentals of Power Electronics'**, Springer Press (January 2001), Second Edition, **ISBN-10:** 0792372700

[3.2] Bimal K. Bose, '**Modern Power Electronics and AC Drives**', Prentice Hall PTR (October 12, 2001), **ISBN-10:** 0130167436

[3.3] B. Pal, B. Chaudhuri, **'Robust Control in Power Systems'**, Springer (June 21, 2005), **ISBN-10:** 038725949X

[3.4] H. J. Sira-Ramirez, R.-Ortigoza**, 'Control Design Techniques in Power Electronics Devices',** Springer Press (July 5, 2006), **ISBN-10:** 1846284589.

[3.5] D. A. Paice**, 'Power Electronics Converter Harmonics: Multipulse Methods for Clean Power',** Wiley-IEEE Press (September 15, 1999), **ISBN-10:** 0780353943.

[3.6] E. Acha, *et al.* **'Power Electronic Control in Electrical Systems'**, Newnes (January 16, 2002), **ISBN-10:** 0750651261.

[3.7] M. H. Rashid, **'Power Electronics Handbook'**, Academic Press (August 24, 2001), **ISBN-10:** 0125816502.

[3.8] M. H. Rashid, '**SPICE for Power Electronics and Electric Power, Second Edition**', CRC Press (November 02, 2005), **ISBN-10:** 0849334187.

[3.9] W. Perruquetti, **'Sliding Mode Control in Engineering'**, Marcel Dekker (January 15, 2002), **ISBN-10:** 0824706714.

[3.10] W. Lixiang, Y. Matsushita, and T. A. Lipo, **'Investigation of dual-bridge matrix converter operating under unbalanced source voltages,'** Proc. IEEE PESC'03, Volume: 3, 2003, pp. 1293-1298.

[3.11] Kolar, J.W e*t al.* **'Novel three-phase AC-DC-AC sparse matrix converter'** Applied Power Electronics Conference and Exposition, 2002. APEC 2002. Seventeenth Annual IEEE, Volume: 2, pp 777 - 791 vol.2

CHAPTER FOUR

SYSTEM CONTROLLERS AND SIMULATION RESULTS

In this chapter a simple R-L load is connected to the output of the matrix converter as shown in Fig. 4.1 while the input is connected to a flat topped three-phase supply. A PI current controller is derived to control the output current, while a space vector algorithm is used to control the input displacement angle. The performance of the converter is evaluated for different output frequencies.

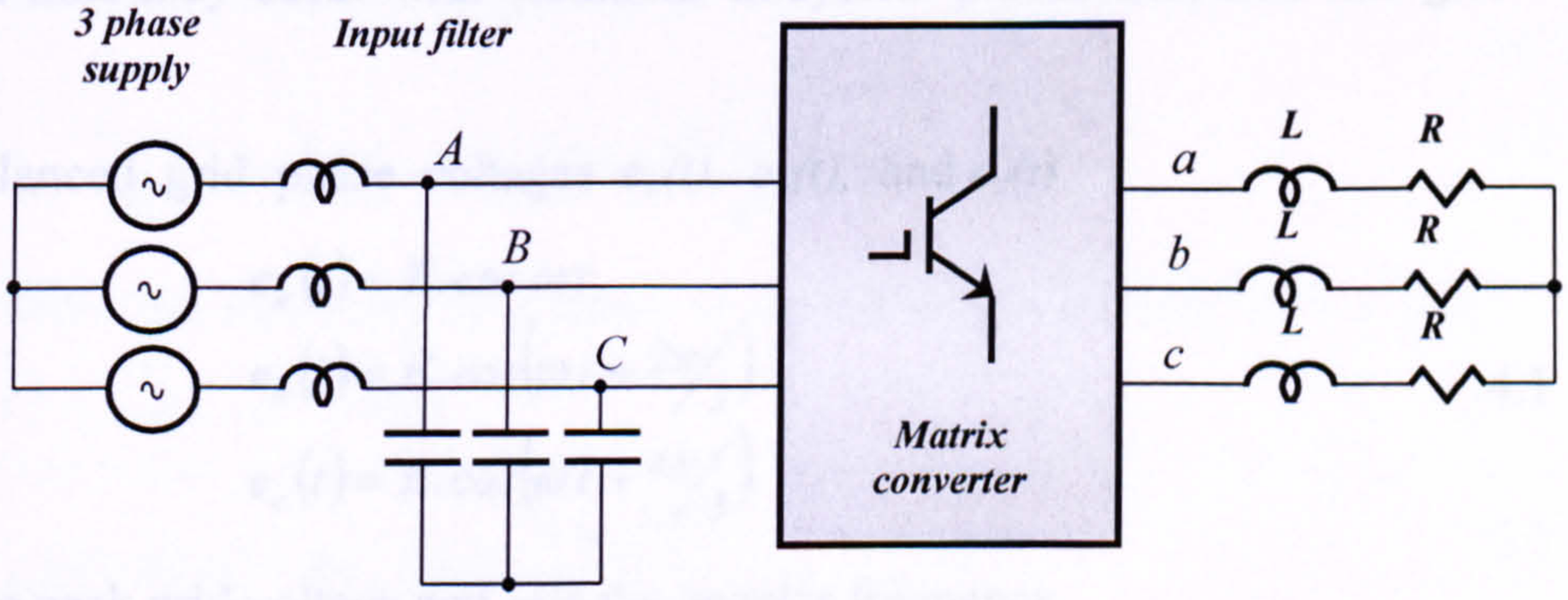


Fig. 4.1 Matrix converter system connected to a three-phase load.

4.1 The inverter stage

Although the matrix converter is tested using a passive R-L load in this section, the derivation of the current controller will consider the presence of an electric grid as shown in Fig. 4.2. This general case avoids repeating the derivation in the next chapter. A simple PI current controller is derived in the *d-q* rotating reference frame for the inverter stage output currents and its transient performance is evaluated [4.1]-[4.8].

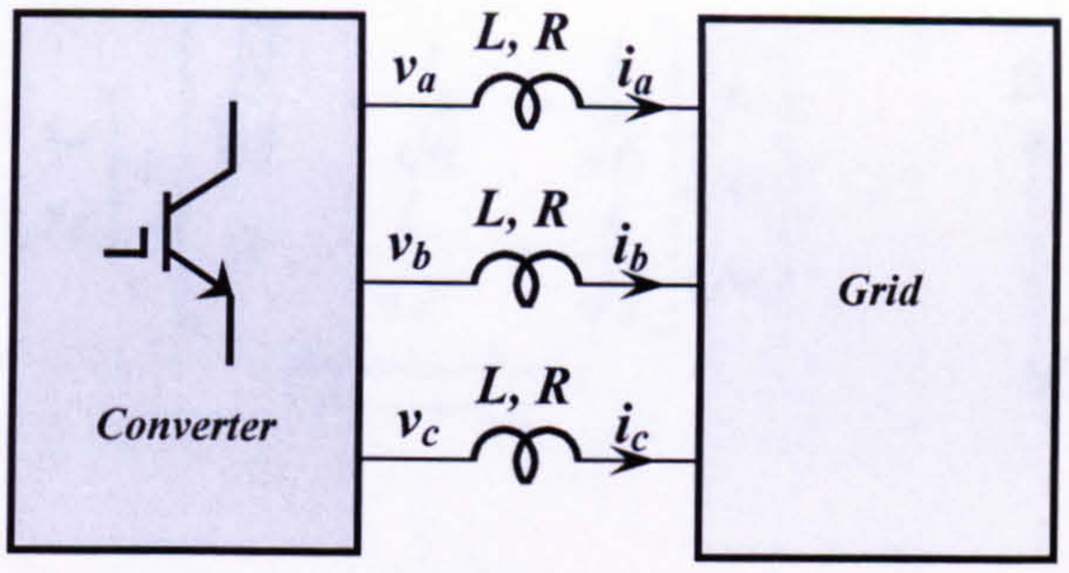


Fig. 4.2 Matrix converter inverter stage schematic diagram.

4.1.1 The current controller

In order to produce sinusoidal output currents, a PI current controller is used in the rotating *d-q* frame. If the *d-q* frame is synchronized with the grid ac voltage, both voltage and current signals become constant vectors or dc quantities. A fast response time can be achieved and zero steady state error can be obtained. However, in the case of grid voltage transients or harmonics, response is limited [4.5]. Moreover, a long transient time may occur with variations in system parameters, load and grid voltage.

Assuming balanced grid phase voltages $e_a(t)$, $e_b(t)$, and $e_c(t)$

$$\begin{aligned} e_a(t) &= E.\cos\omega t \\ e_b(t) &= E.\cos\left(\omega t + {}^{2\pi}/_{3}\right) \\ e_c(t) &= E.\cos\left(\omega t + {}^{4\pi}/_{3}\right) \end{aligned} \qquad 4.1$$

where: E is the peak grid voltage and ω is the angular frequency.

Applying KVL to the output circuit gives

$$v_a(t) - e_a(t) - Ri_a(t) - L\frac{di_a(t)}{dt} = 0$$

$$v_b(t) - e_b(t) - Ri_b(t) - L\frac{di_b(t)}{dt} = 0 \qquad 4.2$$

$$v_c(t) - e_c(t) - Ri_c(t) - L\frac{di_c(t)}{dt} = 0$$

A three-phase system constituted of the three quantities $x_a(t)$, $x_b(t)$ and x_c (t) can be transformed into a vector in a complex reference frame, usually called the α–β frame, as shown in Fig. 4.3.

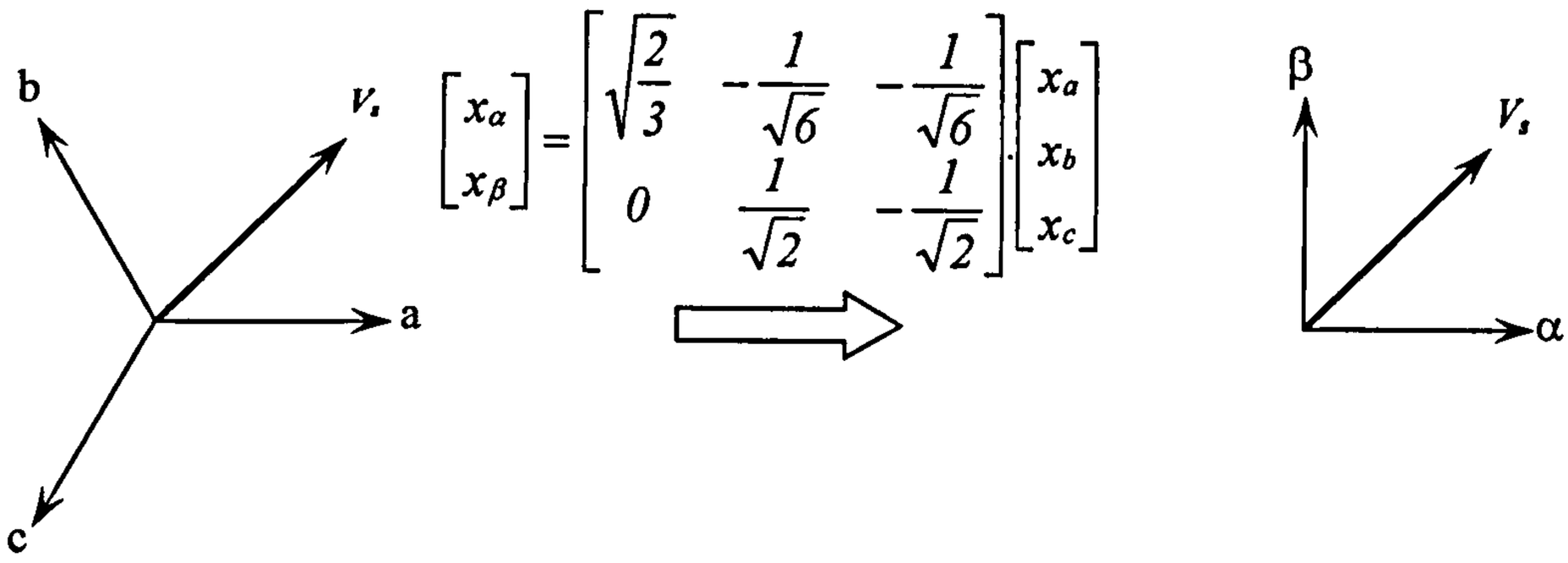


Fig. 4.3 Transformation from the *abc* frame to the α–β frame

In the α–β frame these equations become

$$v_\alpha(t) - e_\alpha(t) - Ri_\alpha(t) - L\frac{di_\alpha(t)}{dt} = 0$$
$$v_\beta(t) - e_\beta(t) - Ri_\beta(t) - L\frac{di_\beta(t)}{dt} = 0 \qquad 4.3$$

which can be written in vector form in the α–β frame

$$\bar{v}_{\alpha\beta}(t) - \bar{e}_{\alpha\beta}(t) - R\bar{i}_{\alpha\beta}(t) - L\frac{d\bar{i}_{\alpha\beta}(t)}{dt} = 0 \qquad 4.4$$

Transforming from the stationary α–β frame to the rotating *d*-*q* frame shown in fig 4.4, gives $\vec{x}_{dq}(t) = \vec{x}_{\alpha\beta}(t).e^{j\omega t}$, thus

$$\bar{v}_{dq}(t) - \bar{e}_{dq}(t) - R\bar{i}_{dq}(t) - L\frac{d\bar{i}_{dq}(t)}{dt} - j\omega L\bar{i}_{dq}(t) = 0 \qquad 4.5$$

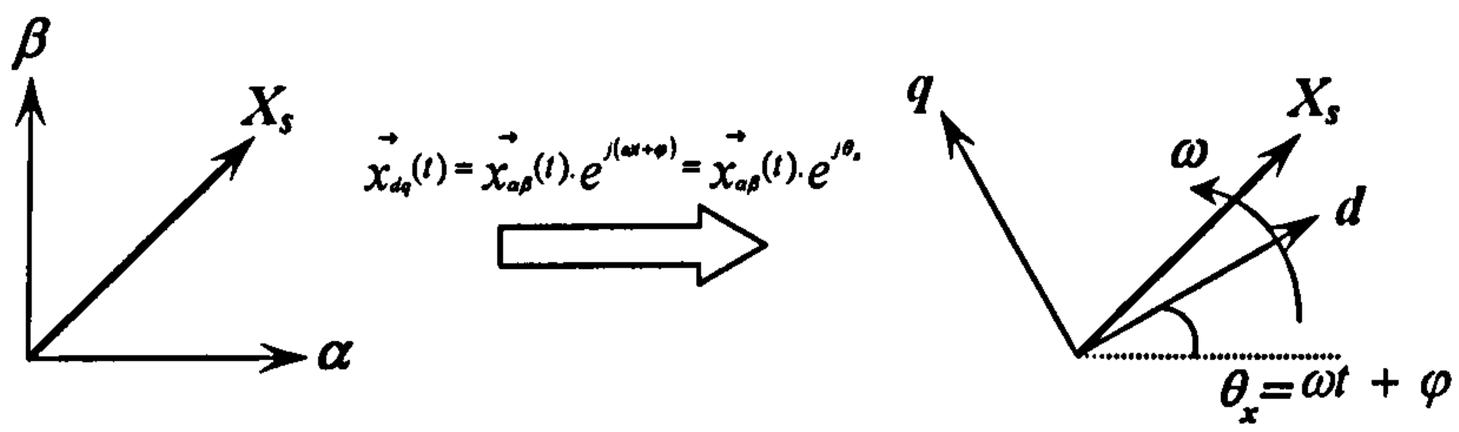


Fig. 4.4 Transformation from the α–β frame to the *d*-*q* frame.

This equation can be resolved into two equations

$$v_d(t) - e_d(t) - Ri_d(t) - L\frac{di_d(t)}{dt} + \omega L i_q(t) = 0$$
$$v_q(t) - e_q(t) - Ri_q(t) - L\frac{di_q(t)}{dt} - \omega L i_d(t) = 0 \qquad 4.6$$

In discrete form, the currents are assumed to vary linearly during one sample period $[k, k+1]$ which leads to the following approximations:

$$\frac{di_d(t)}{dt} \approx \frac{i_{d,k+1} - i_{d,k}}{T_s}$$
$$\frac{di_q(t)}{dt} \approx \frac{i_{q,k+1} - i_{q,k}}{T_s} \qquad 4.7$$

$$i_d(t) \approx \frac{i_{d,k+1} + i_{d,k}}{2}$$
$$i_q(t) \approx \frac{i_{q,k+1} + i_{q,k}}{2} \qquad 4.8$$

These approximations lead to:

$$v_{d,k+1} - e_{d,k+1} - R\left(\frac{i_{d,k+1} + i_{d,k}}{2}\right) - L\left(\frac{i_{d,k+1} - i_{d,k}}{T_s}\right) + \omega L\left(\frac{i_{q,k+1} + i_{q,k}}{2}\right) = 0$$
$$v_{q,k+1} - e_{q,k+1} - R\left(\frac{i_{q,k+1} + i_{q,k}}{2}\right) - L\left(\frac{i_{q,k+1} - i_{q,k}}{T_s}\right) - \omega L\left(\frac{i_{d,k+1} + i_{d,k}}{2}\right) = 0 \qquad 4.9$$

In order to achieve fast transient performance, a dead-beat controller is used which means that the current error can be eliminated in one sampling period. In other words the output current signal at any sampling instant $k+1$ should be equal to the reference current signal at the previous sampling instant k.

$$i_{d,k+1} \approx i^*_{d,k}$$
$$i_{q,k+1} \approx i^*_{q,k} \qquad 4.10$$

With this assumption equations 4.9 become

$$v_{d,k+1} = e_{d,k+1} + R\,i_{d,k} + \left(\frac{L}{T_s} + \frac{R}{2}\right)\left(i^*_{d,k} - i_{d,k}\right) - \omega L\left(\frac{i^*_{q,k} + i_{q,k}}{2}\right)$$
$$v_{q,k+1} = e_{q,k+1} + R\,i_{q,k} + \left(\frac{L}{T_s} + \frac{R}{2}\right)\left(i^*_{q,k} - i_{q,k}\right) + \omega L\left(\frac{i^*_{d,k} + i_{d,k}}{2}\right) \qquad 4.11$$

The resistive voltage drop can be replaced by an integral on the assumption that in steady state the current is equal to the sum of the current errors.

$$v_{d,k+1} = e_{d,k+1} + R\sum_{n=0}^{k}\left(i^*_{d,n} - i_{d,n}\right) + \left(\frac{L}{T_s} + \frac{R}{2}\right).\left(i^*_{d,k} - i_{d,k}\right) - \omega L\left(\frac{i^*_{q,k} + i_{q,k}}{2}\right)$$
$$v_{q,k+1} = e_{q,k+1} + R\sum_{n=0}^{k}\left(i^*_{q,n} - i_{q,n}\right) + \left(\frac{L}{T_s} + \frac{R}{2}\right).\left(i^*_{q,k} - i_{q,k}\right) + \omega L\left(\frac{i^*_{d,k} + i_{d,k}}{2}\right) \qquad 4.12$$

The grid voltage change between two successive samples is minimal, thus

$$e_{d,k+1} \approx e_{d,k}$$
$$e_{q,k+1} \approx e_{q,k} \qquad 4.13$$

Now equation 4.12 can be rewritten for an R-L load in the form

$$v_{d,k+1} = k_i\sum_{n=0}^{k}\left(i^*_{d,n} - i_{d,n}\right) + k_p.\left(i^*_{d,k} - i_{d,k}\right) - \omega L\left(\frac{i^*_{q,k} + i_{q,k}}{2}\right)$$
$$v_{q,k+1} = k_i\sum_{n=0}^{k}\left(i^*_{q,n} - i_{q,n}\right) + k_p.\left(i^*_{q,k} - i_{q,k}\right) + \omega L\left(\frac{i^*_{d,k} + i_{d,k}}{2}\right) \qquad 4.14$$

For an R-L load the grid components $e_{d,k}$ and $e_{q,k}$ are zero. Fig. 4.5 presents the block diagram of the current controller.

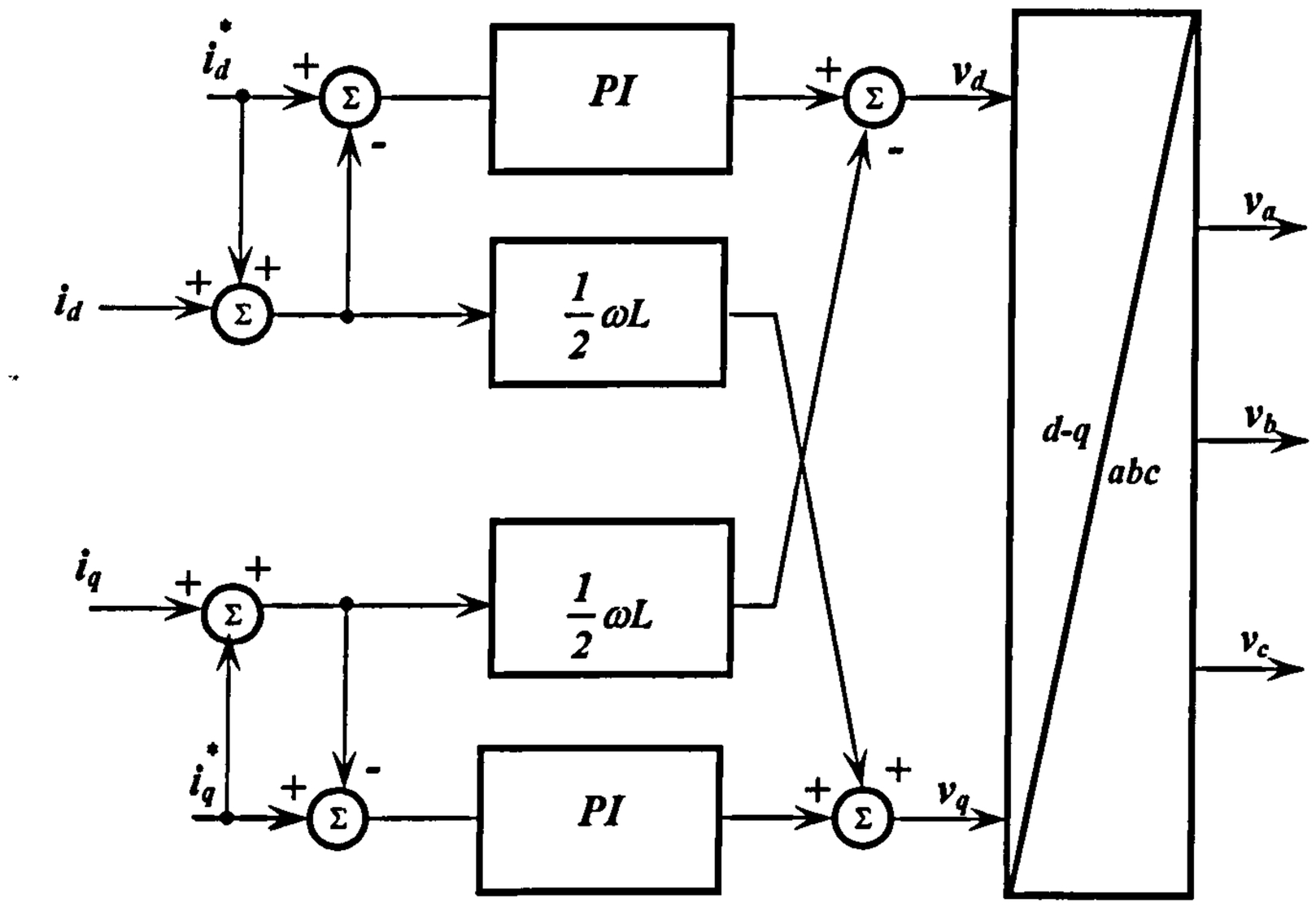


Fig. 4.5 Current controller schematic diagram.

The command signals are then transformed from the *d-q* frame to the α–β frame and then to the *abc* frame as shown in Fig. 4.6.

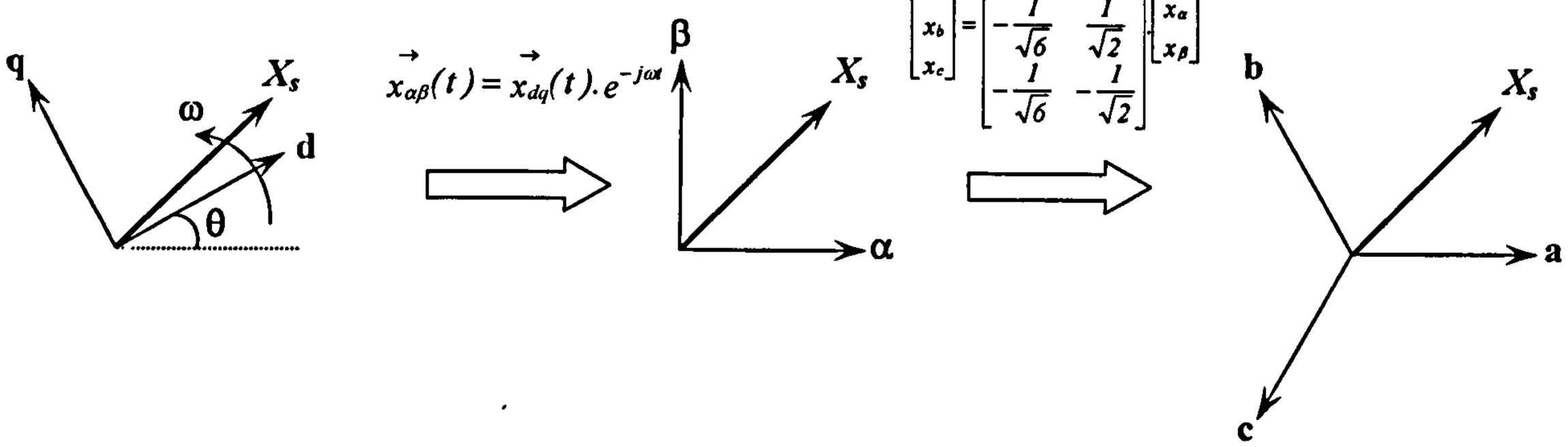


Fig. 4.6 Transforming from the *d-q* frame to the *abc* frame.

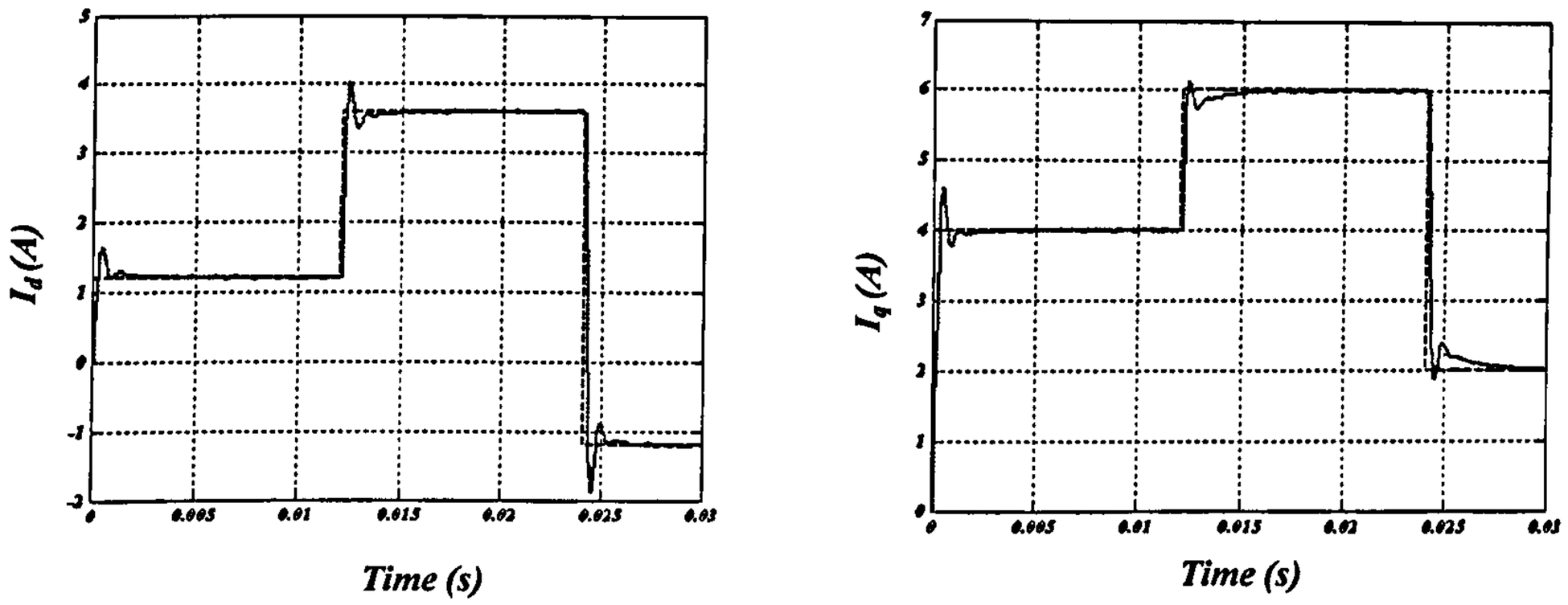


Fig. 4.7 Simulated direct and quadrature axis current responses when neglecting measurement time delay.

Fig. 4.7 shows the output response for a variation in direct and quadrature currents respectively (using Simulink). Practically, one sample delay is added to account for the DSP computational time. If this time delay is considered, oscillatory controller behaviour is obtained as shown in Fig. 4.8. If the proportional gain of the system is reduced to 50% the dead-beat gain, the oscillations decrease [4.7]. However, since the gain is reduced, the current settling time is longer and overshoot is introduced, as shown in Fig. 4.9.

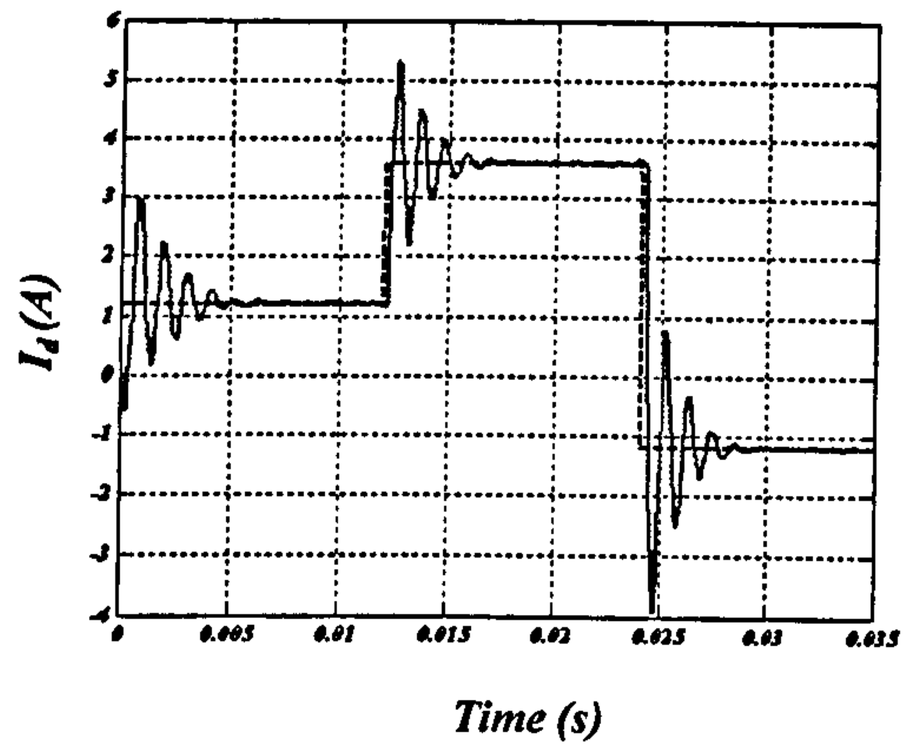


Fig. 4.8 Simulated direct and quadrature axis current responses taking into account one sample time delay due to the measurements.

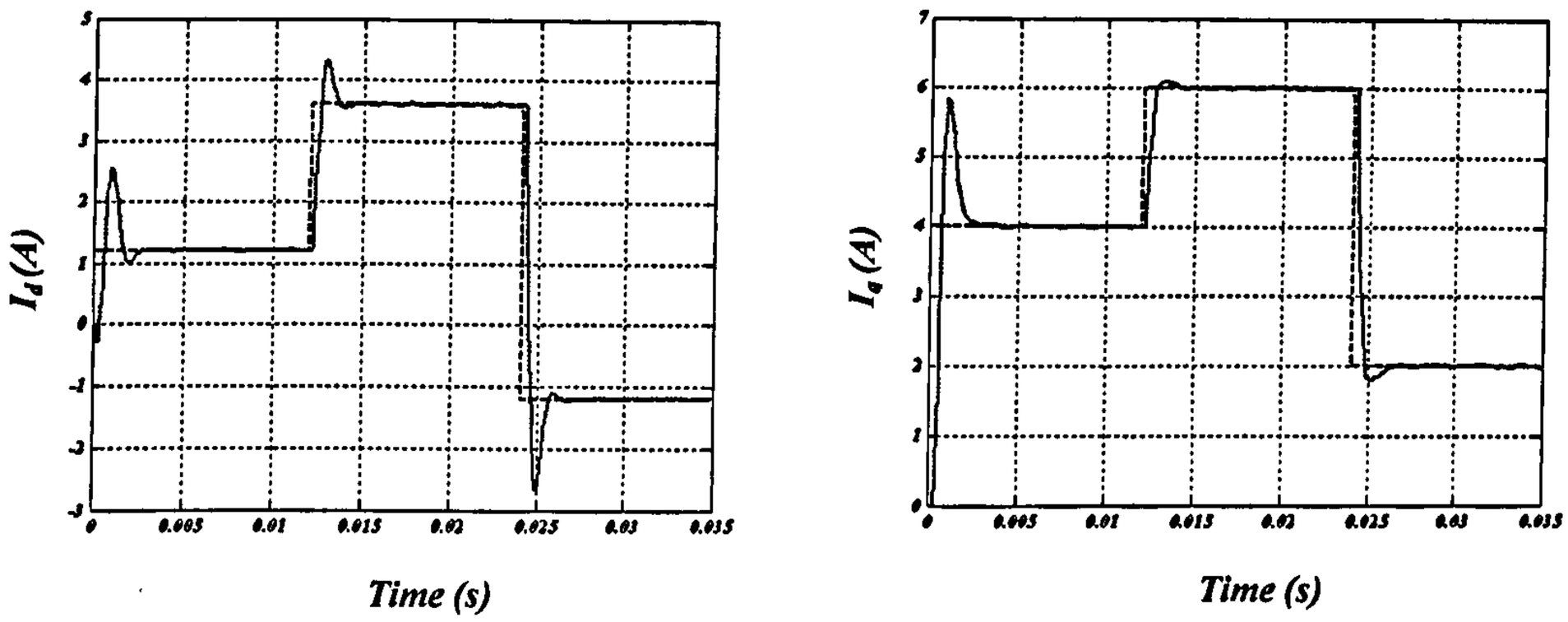


Fig. 4.9 Simulated direct and quadrature axis current responses taking into account one sample time delay due to the measurements and reducing the proportional gain to 50%.

4.1.2 Overmodulation

The output reference voltages of the inverter stage $u_a(t)$, $u_b(t)$, and $u_c(t)$ are responsible for generating the switching signals, $S_7(t)$, $S_8(t)$..... $S_{12}(t)$, by comparing these voltage peaks with the carrier signal. The limit is reached when any of the reference voltages are equal to the peak of the carrier signal, which is known as unity modulation index ($m=1$). If the reference signal peak is larger than the carrier signal peak then over-modulation occurs, which results in low frequency current harmonics [4.4]. The overmodulation effect can be limited if an offset voltage v_x is subtracted from the reference voltage signals as shown in equation 4.15.

$$\begin{aligned} v_{aa} &= v_a - v_x \\ v_{bb} &= v_b - v_x \\ v_{cc} &= v_c - v_x \end{aligned} \qquad 4.15$$

Although the phase reference voltages are no longer sinusoidal as shown in Fig. 4.10, the load voltages are sinusoidal. In order to optimize the modulation, u_x is selected such that the reference signals are symmetrical as in the following equation

$$v_x = \frac{max\left(v_a^*, v_b^*, v_c^*\right) + min\left(v_a^*, v_b^*, v_c^*\right)}{2} \qquad 4.16$$

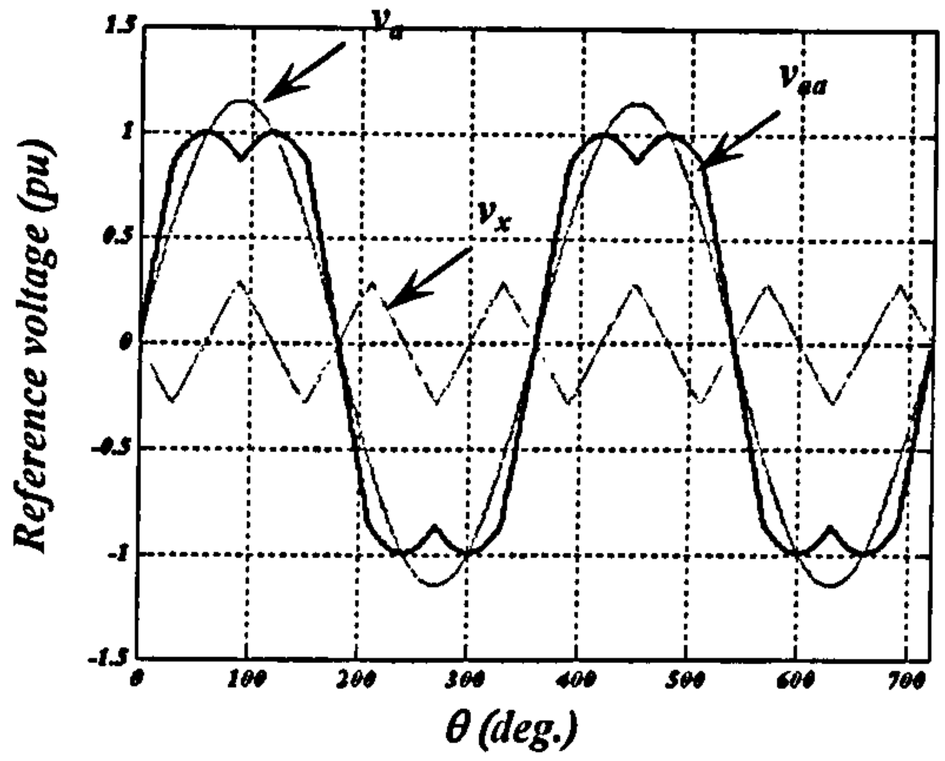


Fig. 4.10 Modified command signal for phase a by adding $v_x = \frac{max\left(v_a^*, v_b^*, v_c^*\right) + min\left(v_a^*, v_b^*, v_c^*\right)}{2}$

After subtracting the offset, if the reference signal is still larger than the peak of the carrier, saturation occurs which results in integrator windup. In this case the integral of the current control error becomes large resulting in uncontrolled phase currents. In order to avoid windup the author suggests pausing the integration, but this results in reduced control system performance. Anti-windup is achieved by the algorithm shown in Fig. 4.11.

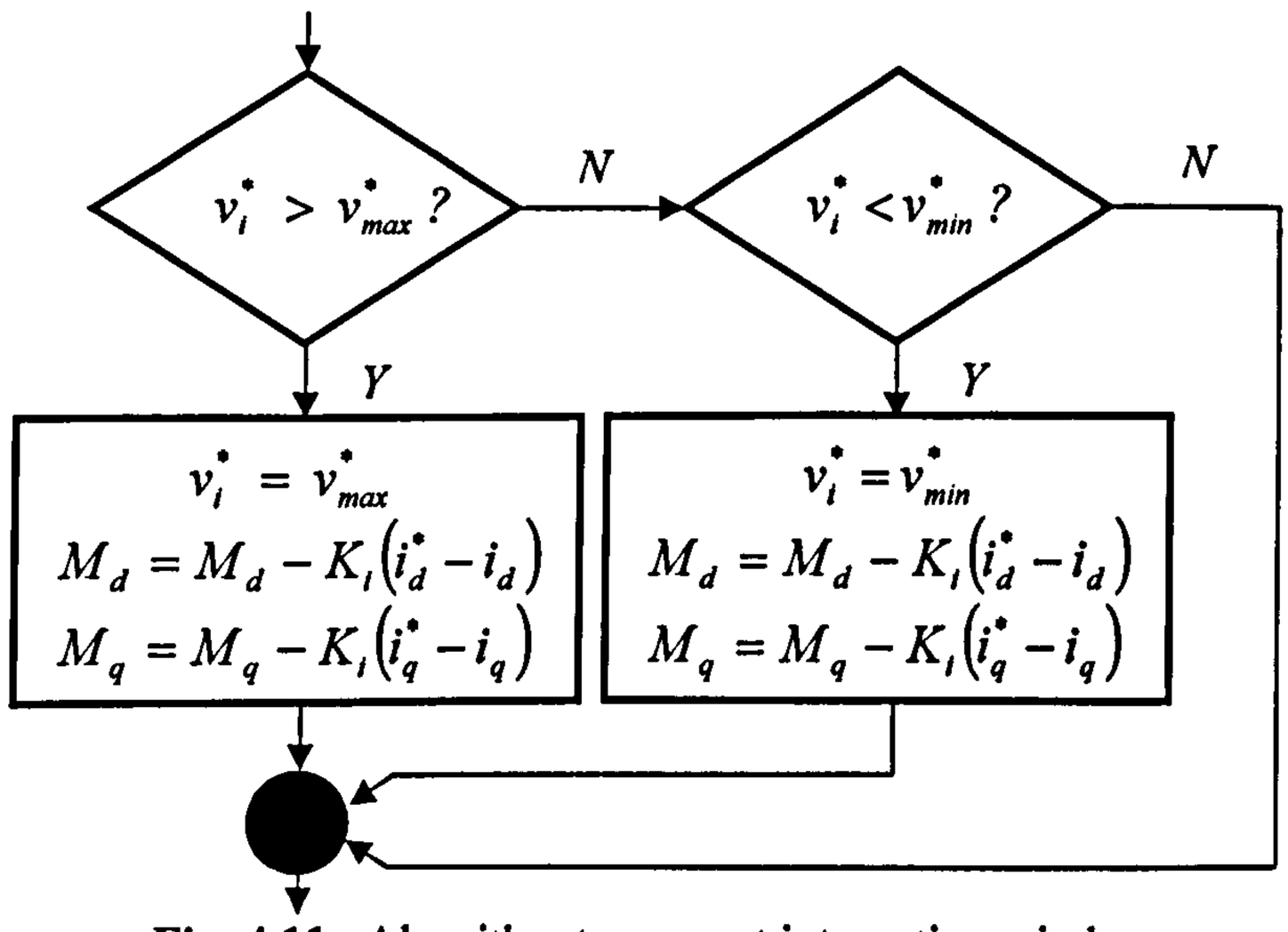


Fig. 4.11 Algorithm to prevent integration windup.

4.1.3 Generation of the reference current

The only restriction on the output current reference signal is the over modulation boundary ($m \geq 0.866$) shown in Fig. 4.12, which assumes that the rectifier stage operates at unity power factor. This boundary can be computed from the relation

$$I = \frac{0.866 V_{in}}{\sqrt{R^2 + (\omega L)^2}} \qquad 4.17$$

Any arbitrary value below this boundary can be used as a reference current. The inverter stage to control the output current is shown in Fig. 4.13

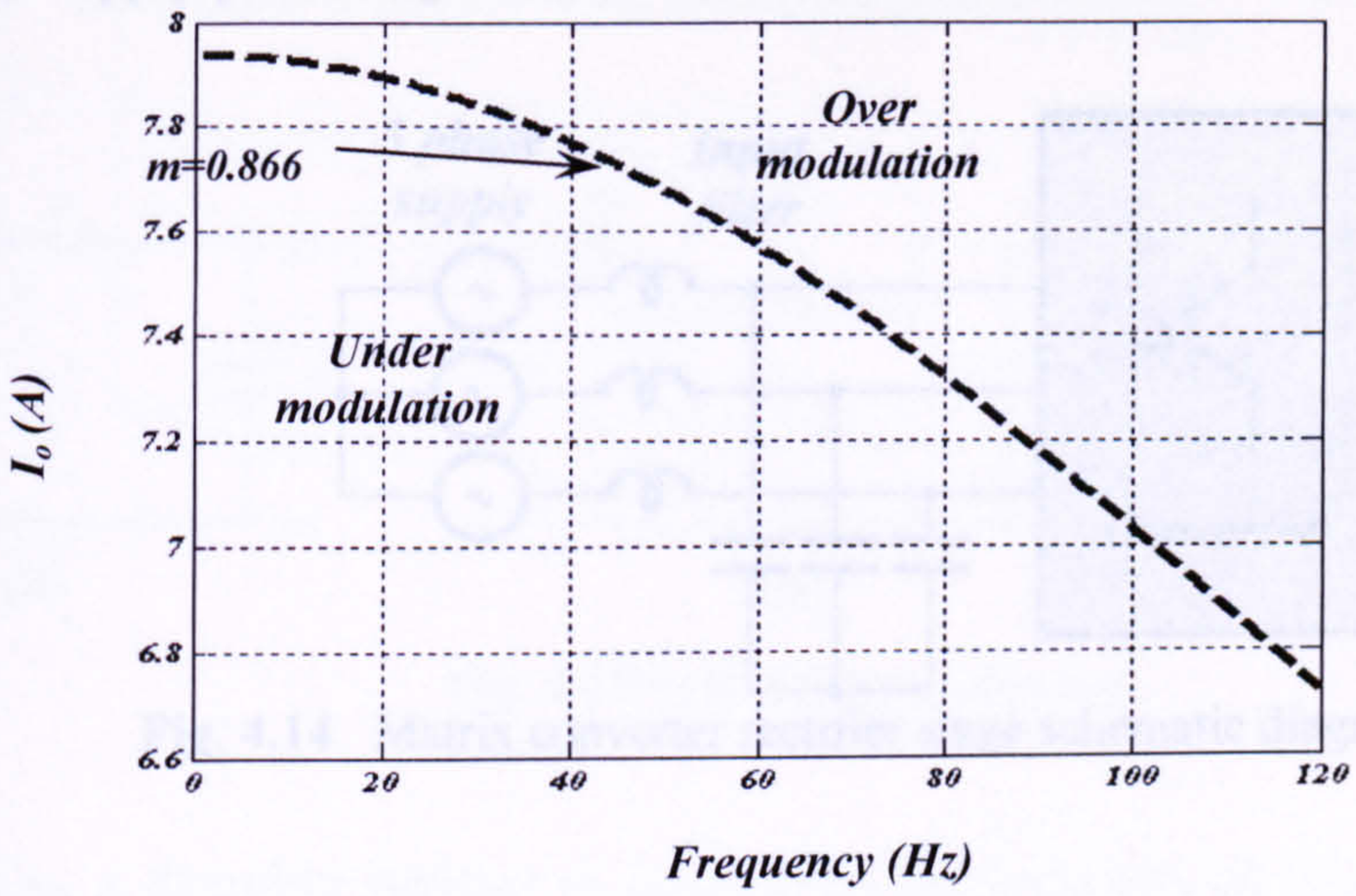


Fig. 4.12 Over modulation boundary.

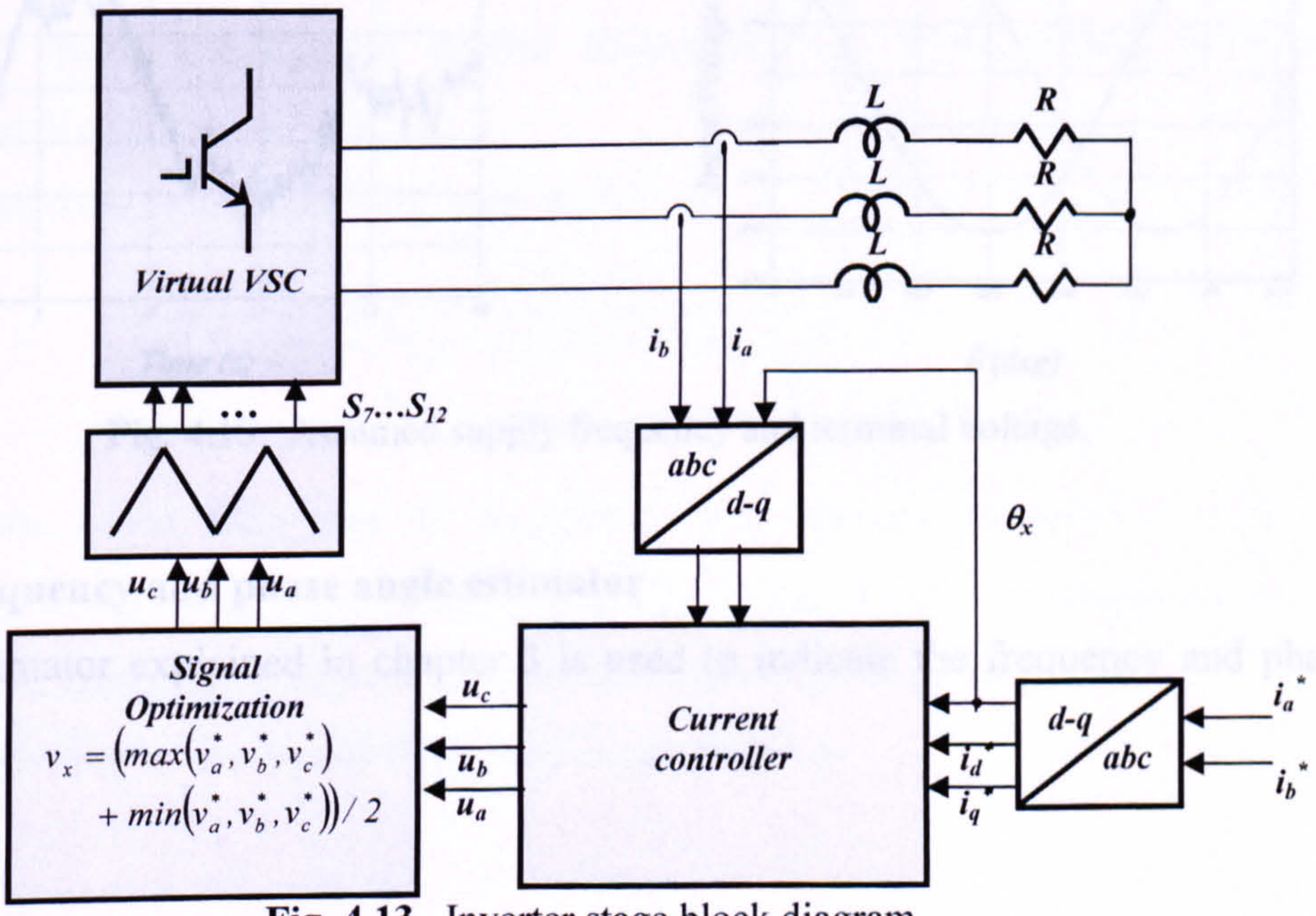


Fig. 4.13 Inverter stage block diagram.

4.2 The active rectifier stage

The aim of the matrix converter rectifier stage is to control the displacement angle of the input current as shown in Fig. 4.14, using the space vector control mentioned in chapter 3. The input to the matrix converter is the three-phase grid voltage which means that the supply frequency is assumed to fluctuate slightly around 50 Hz as shown in Fig. 4.14. Also the input voltage is assumed flat topped as shown in Fig. 4.15, which contains 0.6% and 0.45% fifth and seventh order harmonics respectively. In order to achieve proper displacement angle control, the supply frequency and hence the supply phase angle must be measured or estimated accurately.

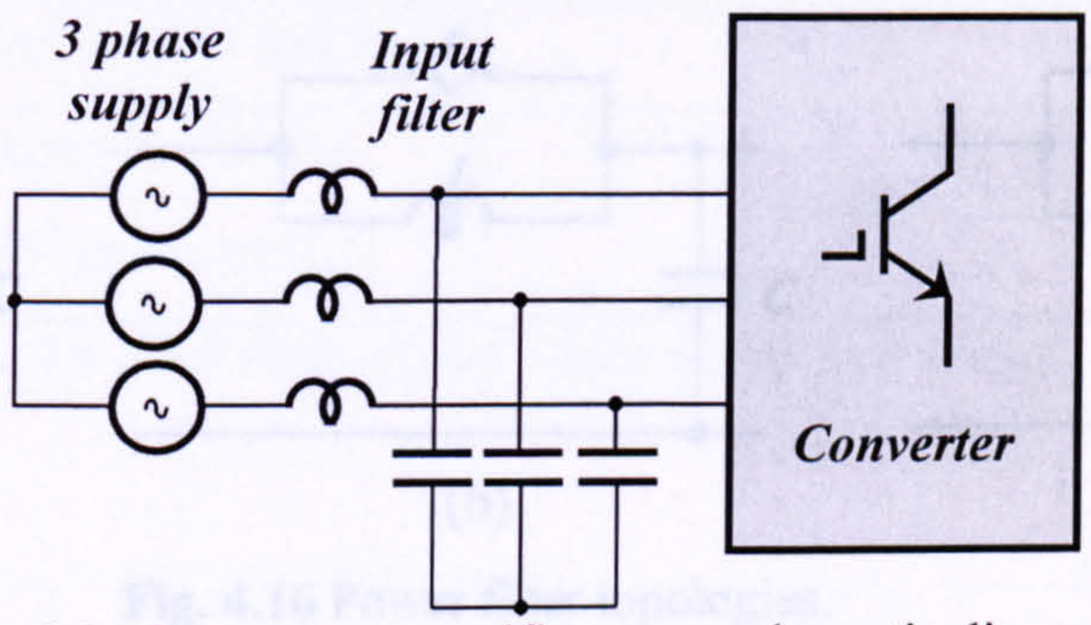


Fig. 4.14 Matrix converter rectifier stage schematic diagram.

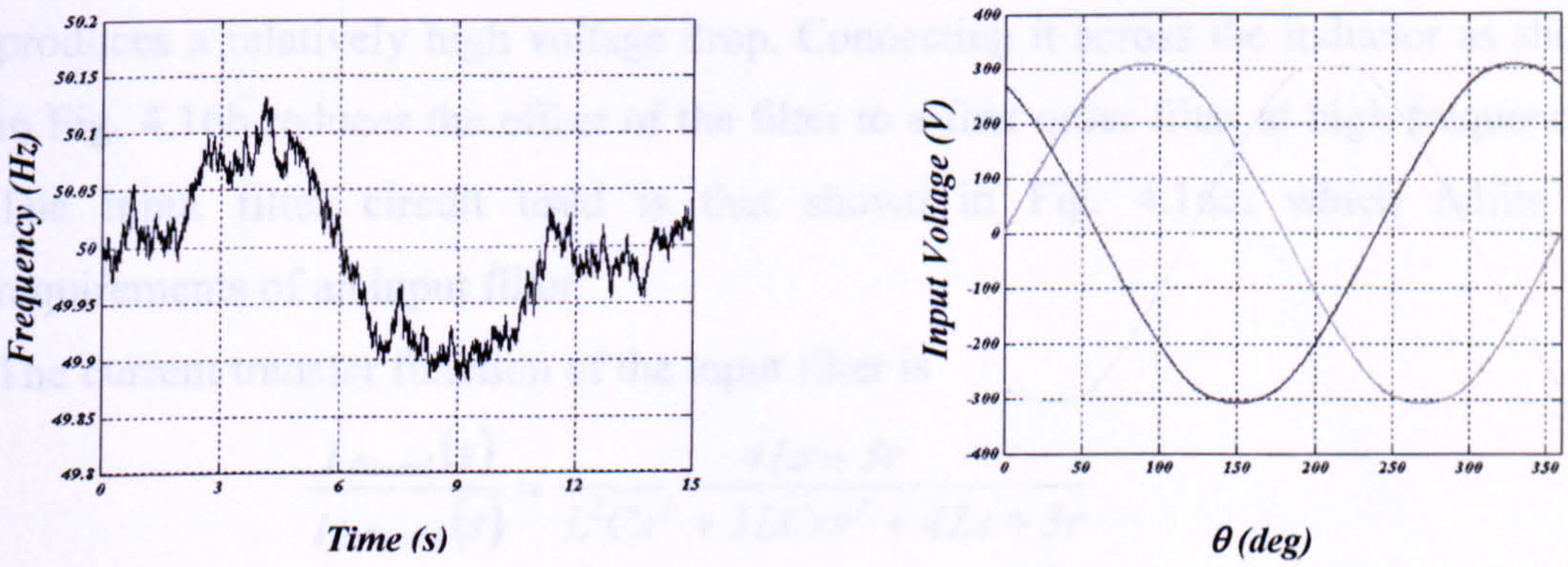


Fig. 4.15 Assumed supply frequency and terminal voltage.

4.2.1 Frequency and phase angle estimator

Same estimator explained in chapter 3 is used to indicate the frequency and phase angle.

4.2.2 The input filter

The main function of the input filter [4.10] is to reduce the input current ripple with minimum reactive element energy. Common topologies are shown in Fig. 4.16.

The input filter has to meet the following requirements:

- Produce a high attenuation at the switching frequency.
- Minimal voltage drop across the filter to achieve a high overall voltage transfer ratio.
- Minimal filter weight and volume.
- Maximize the achievable operating power factor.

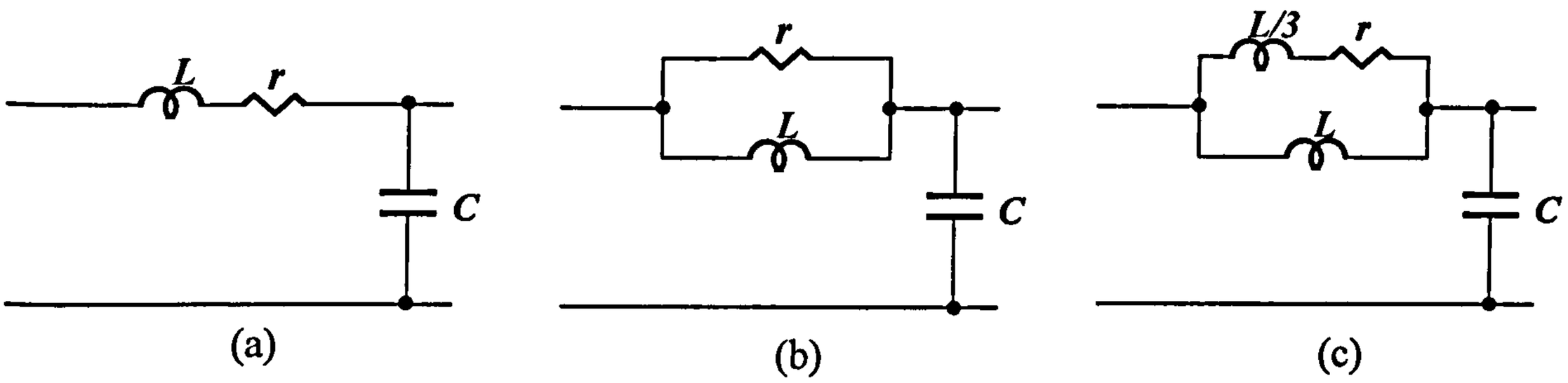


Fig. 4.16 Power filter topologies.

Connecting a damping resistor in series with the inductor, as shown in Fig. 4.16a, produces a relatively high voltage drop. Connecting it across the inductor as shown in Fig. 4.16b reduces the effect of the filter to a first order filter at high frequencies. The input filter circuit used is that shown in Fig. 4.16c, which fulfils the requirements of an input filter.

The current transfer function of the input filter is

$$\frac{I_{filtered}(s)}{I_{unfiltered}(s)} = \frac{4Ls + 3r}{L^2Cs^3 + 3LCrs^2 + 4Ls + 3r} \qquad 4.18$$

Usually the input filter cut-off frequency ω_c is chosen to provide sufficient attenuation at the switching frequency (6.1 kHz). Also the cut-off frequency should be above the harmonic components of the unfiltered input current.

The voltage drop at the supply frequency is calculated as:

$$\frac{\Delta E}{E} \approx 1 - \sqrt{1 - (\omega_1 . L)^2 . \left(\frac{I_{FL}}{E}\right)^2} \qquad 4.19$$

The input filter characteristics are shown in table 4.1 and the filter frequency response is shown in Fig. 4.17.

L/phase	6.3 mH
C/phase	10 μF
r/phase	20 Ω
ω_c	4500 rad/s
% voltage drop at full load	0.15%
Attenuation at the switching frequency	-26dB

Table 4.1 Input filter typical values and characteristics.

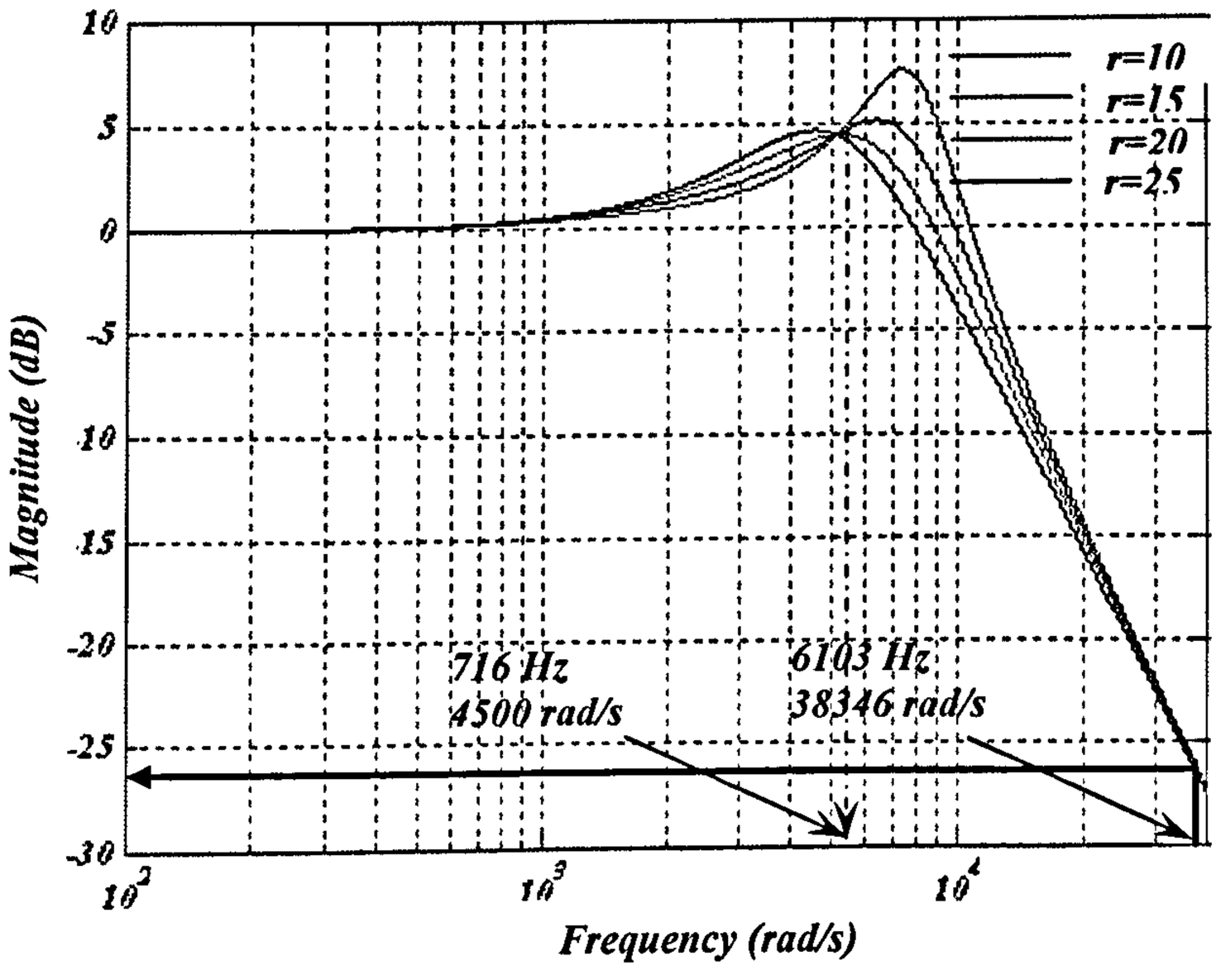


Fig. 4.17 Input filter frequency response.

The expression for the filtered input current is

$$I_{filtered}(s) = \left(\frac{4Ls + 3r}{L^2Cs^3 + 3LCrs^2 + 4Ls + 3r}\right)\left(I_{unfiltered}(s) + Cs\,V_{in}(s)\right) \qquad 4.20$$

The direct input voltage to the matrix converter is

$$v_c(t) = \frac{1}{C}\int\left(i_{filtered}(t) - i_{unfiltered}(t)\right)\,dt$$
$$V_c(s) = \frac{1}{Cs}\left(I_{filtered}(s) - I_{unfiltered}(s)\right) \qquad 4.21$$

Fig. 4.18 shows the input filter block diagram.

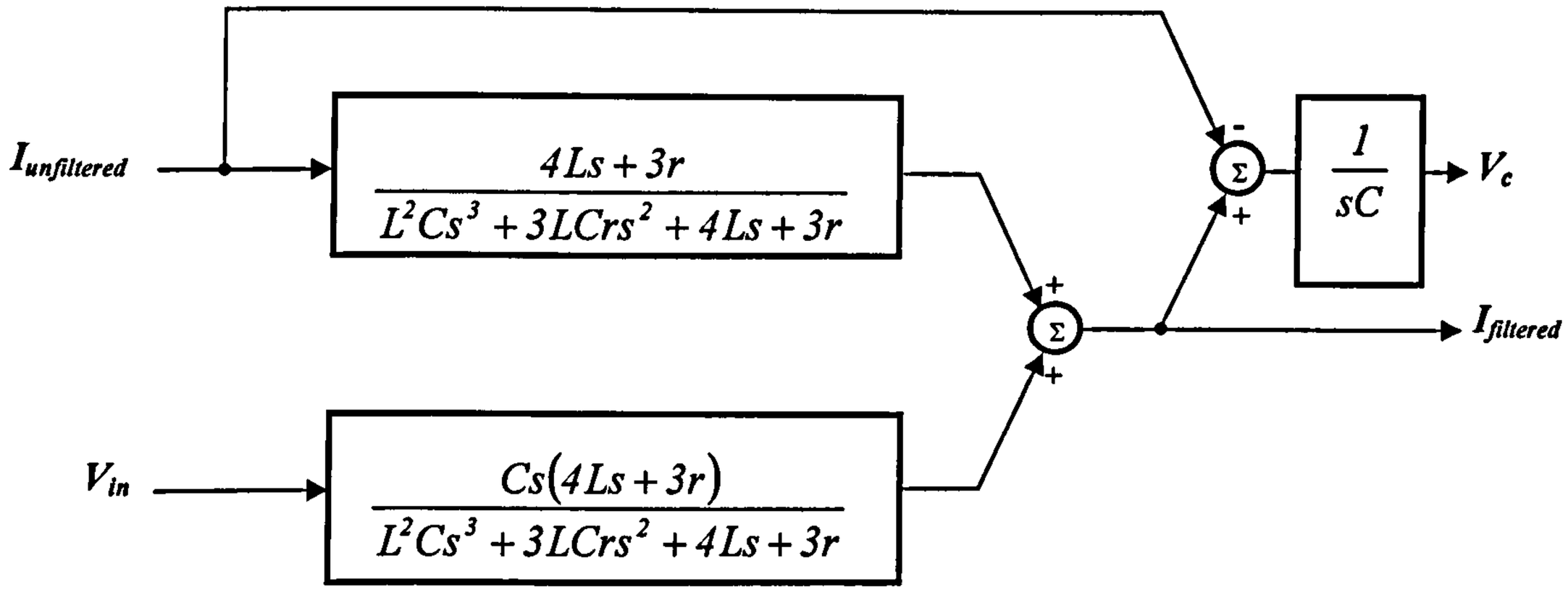


Fig. 4.18 Input filter block diagram per phase.

Another interesting aspect related to the frequency estimator is the point of measurement of the input voltage to the matrix converter. Normally, the voltage between the input filter and the converter is used for measurement in order to avoid input filter angle compensation in the rectifier stage calculation. Here the measured voltage is slightly distorted as shown in Fig. 4.19a. In this case the frequency estimator response is as shown in Fig. 4.19b. By reducing the gain of the frequency and phase angle estimators (*A=284, B=17*), the oscillations are eliminated as shown in Fig. 4.19c. Another solution is to measure the actual grid voltage as opposed to the voltage at the point between the filter and the converter, and compensate for the effect of the input filter angle.

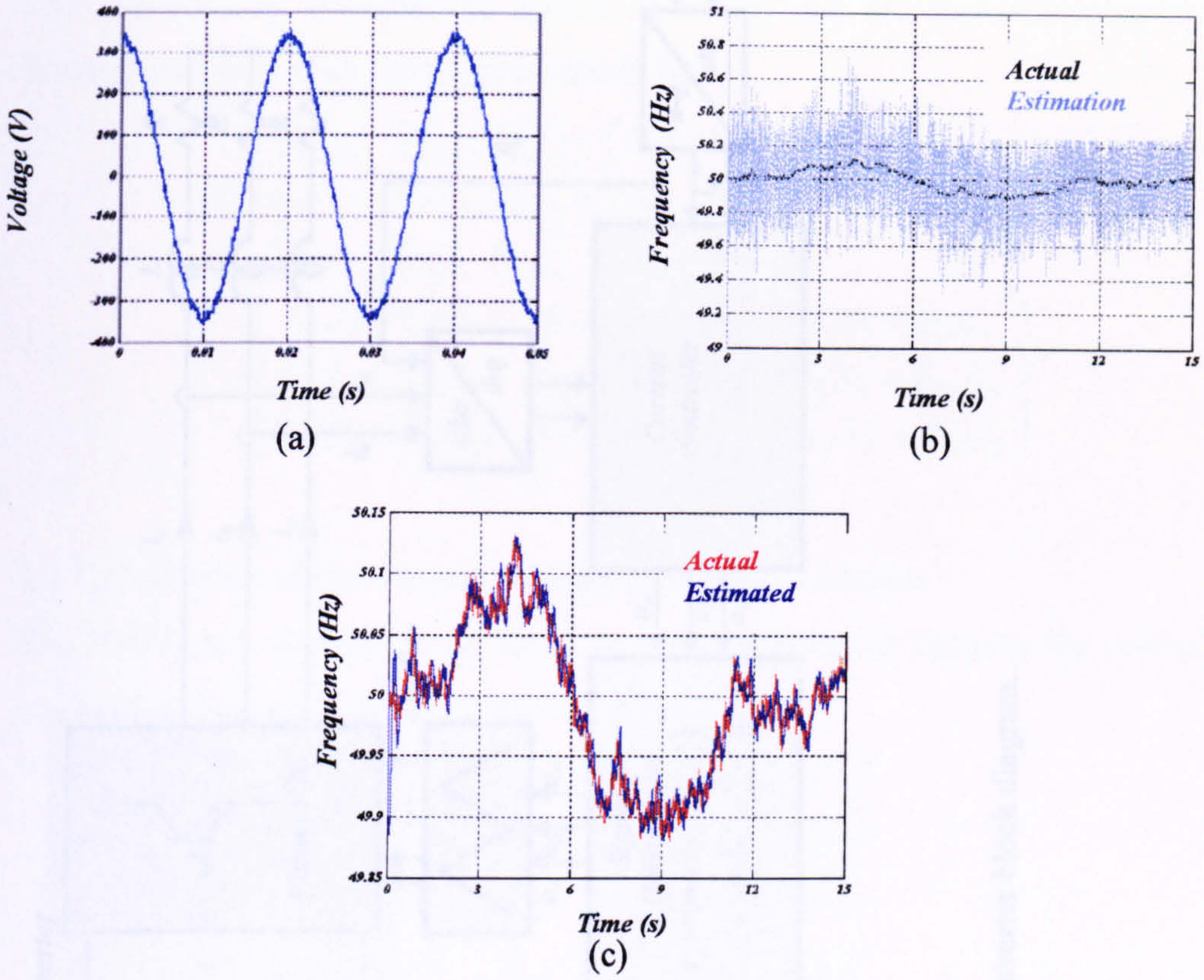


Fig. 4.19 Frequency estimator response under distorted input voltage:

(a) matrix converter input voltage.

(b) frequency estimator response without reducing the gain factor.

(c) frequency estimator response when reducing the gain factor to 25%.

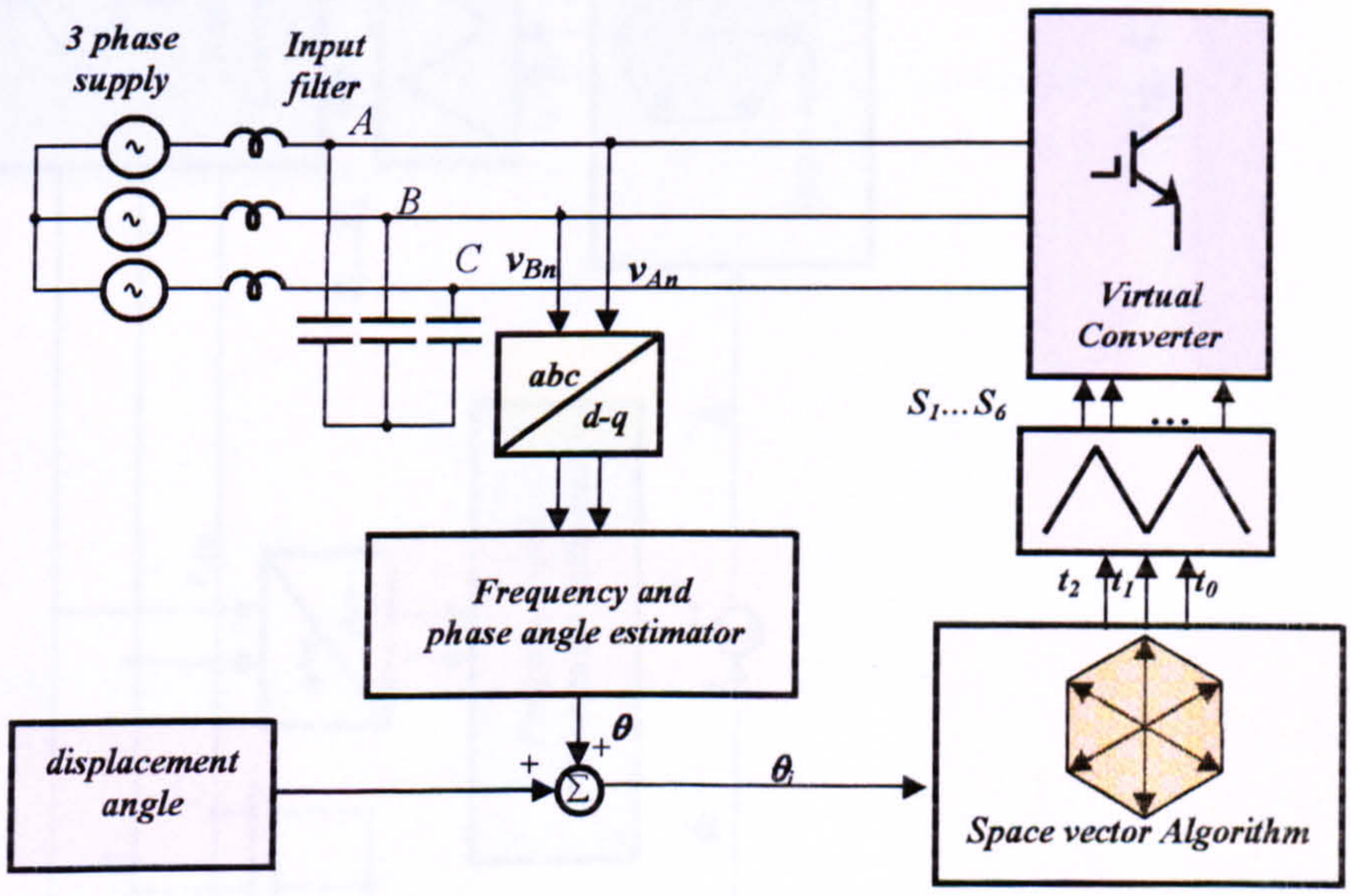


Fig. 4.20 Rectifier stage block diagram

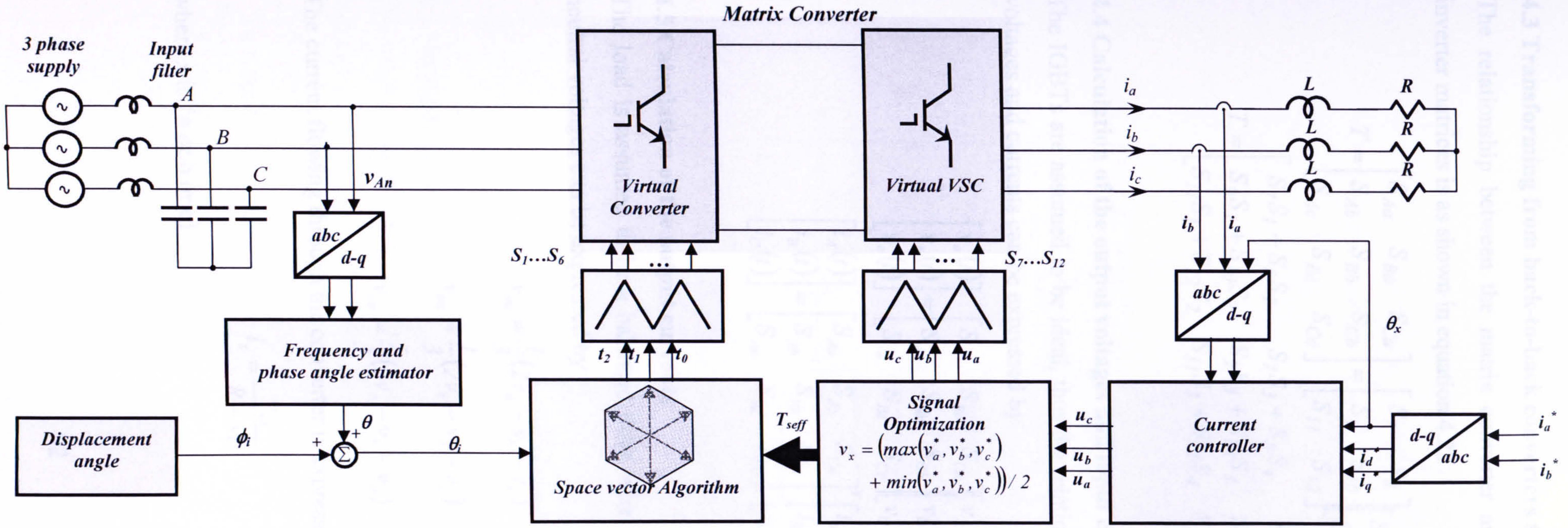


Fig. 4.21 The matrix converter block diagram.

4.3 Transforming from back-to-back converters to the matrix converter

The relationship between the matrix converter and the corresponding rectifier and inverter matrices is as shown in equation 4.22

$$T=\begin{bmatrix} S_{Aa} & S_{Ba} & S_{Ca} \\ S_{Ab} & S_{Bb} & S_{Cb} \\ S_{Ac} & S_{Bc} & S_{Cc} \end{bmatrix}=\begin{bmatrix} S_7 & S_8 \\ S_9 & S_{10} \\ S_{11} & S_{12} \end{bmatrix}\begin{bmatrix} S_1 & S_3 & S_5 \\ S_2 & S_4 & S_6 \end{bmatrix} \tag{4.22}$$

$$T=\begin{bmatrix} S_7S_1+S_8S_2 & S_7S_3+S_8S_4 & S_7S_5+S_8S_6 \\ S_9S_1+S_{10}S_2 & S_9S_3+S_{10}S_4 & S_9S_5+S_{10}S_6 \\ S_{11}S_1+S_{12}S_2 & S_{11}S_3+S_{12}S_4 & S_{11}S_5+S_{12}S_6 \end{bmatrix}$$

4.4 Calculation of the output voltages and input currents

The IGBTs are assumed to be ideal, thus the relationship between the output and input voltages and currents can be expressed by

$$\begin{bmatrix} v_a(t) \\ v_b(t) \\ v_c(t) \end{bmatrix}=\begin{bmatrix} S_{Aa} & S_{Ba} & S_{Ca} \\ S_{Ab} & S_{Bb} & S_{Cb} \\ S_{Ac} & S_{Bc} & S_{Cc} \end{bmatrix}\begin{bmatrix} v_A(t) \\ v_B(t) \\ v_C(t) \end{bmatrix} \tag{4.23}$$

$$\begin{bmatrix} i_A(t) \\ i_B(t) \\ i_C(t) \end{bmatrix}=\begin{bmatrix} S_{Aa} & S_{Ba} & S_{Ca} \\ S_{Ab} & S_{Bb} & S_{Cb} \\ S_{Ac} & S_{Bc} & S_{Cc} \end{bmatrix}^T\begin{bmatrix} i_a(t) \\ i_b(t) \\ i_c(t) \end{bmatrix}$$

4.5 Calculation of the output currents

The load is assumed to be a balanced three wire star connection. The output line to neutral voltages can be expressed by

$$v_{ao}=\frac{1}{3}(2v_a-v_b-v_c)$$

$$v_{bo}=\frac{1}{3}(2v_b-v_a-v_c) \tag{4.24}$$

$$v_{co}=\frac{1}{3}(2v_c-v_b-v_a)$$

The current flowing through the converter is expressed by

$$i_x=\frac{v_{xo}}{R+sL} \tag{4.25}$$

where x is {a or b or c}

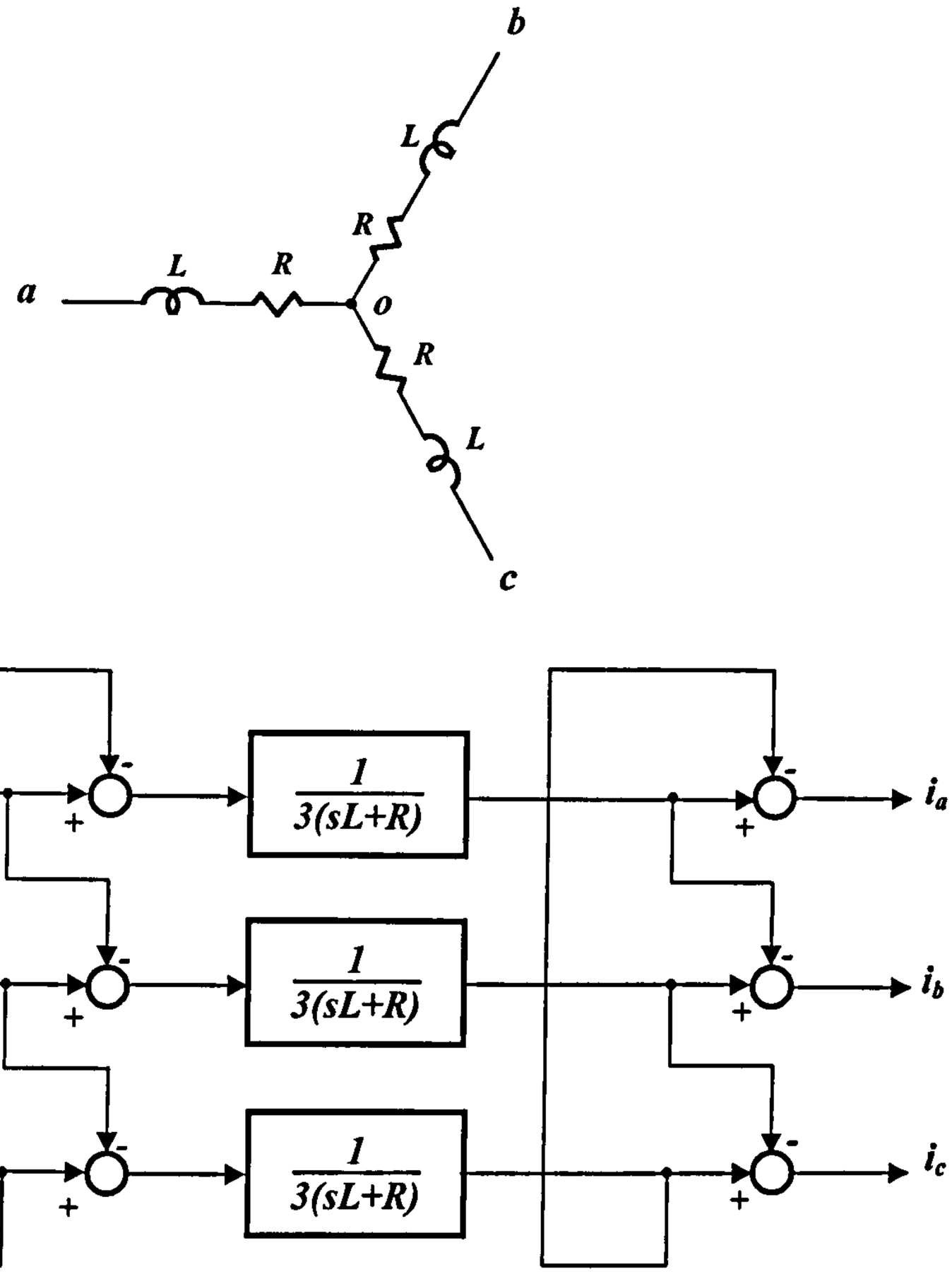


Fig. 4.22 Output connected load in the *s* domain

R/phase	21.5Ω
L/phase	20mH

Table 4.3 Load typical Values

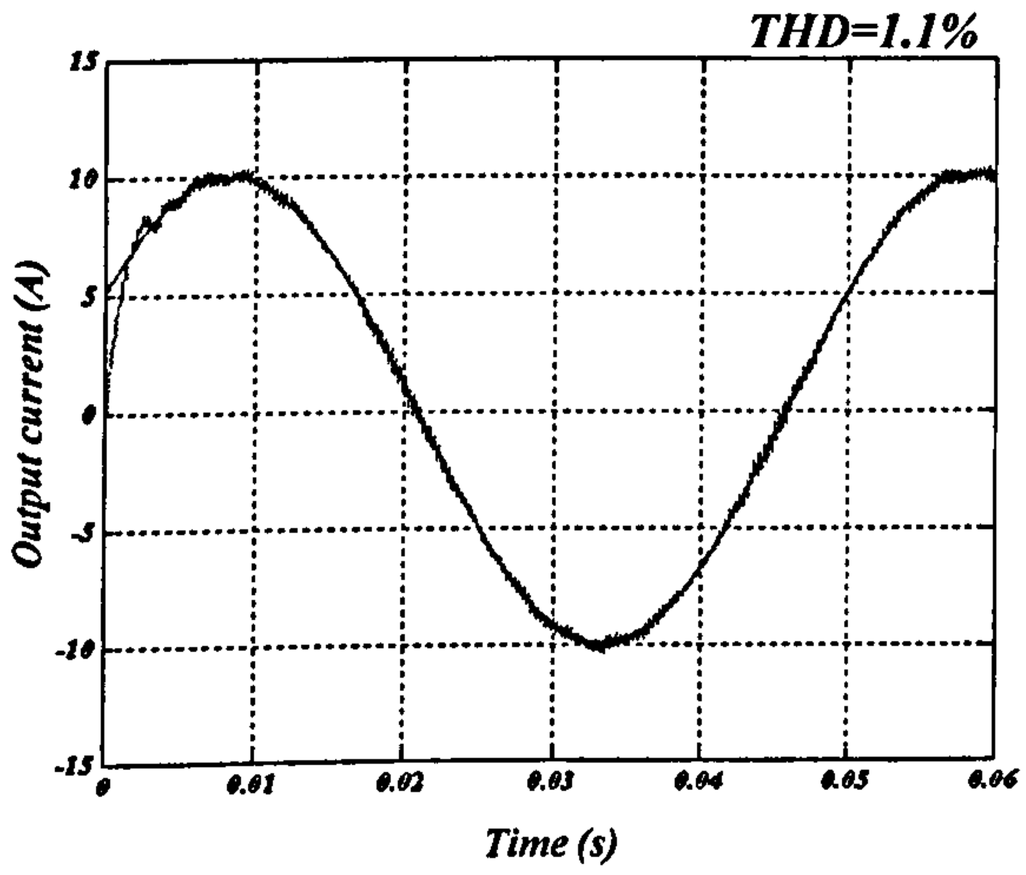
4.6 Simulation results

The proposed controllers are simulation tested using Simulink at two different output frequencies: 80 Hz and 20 Hz, with three different input supply conditions [4.4], [4.11]:

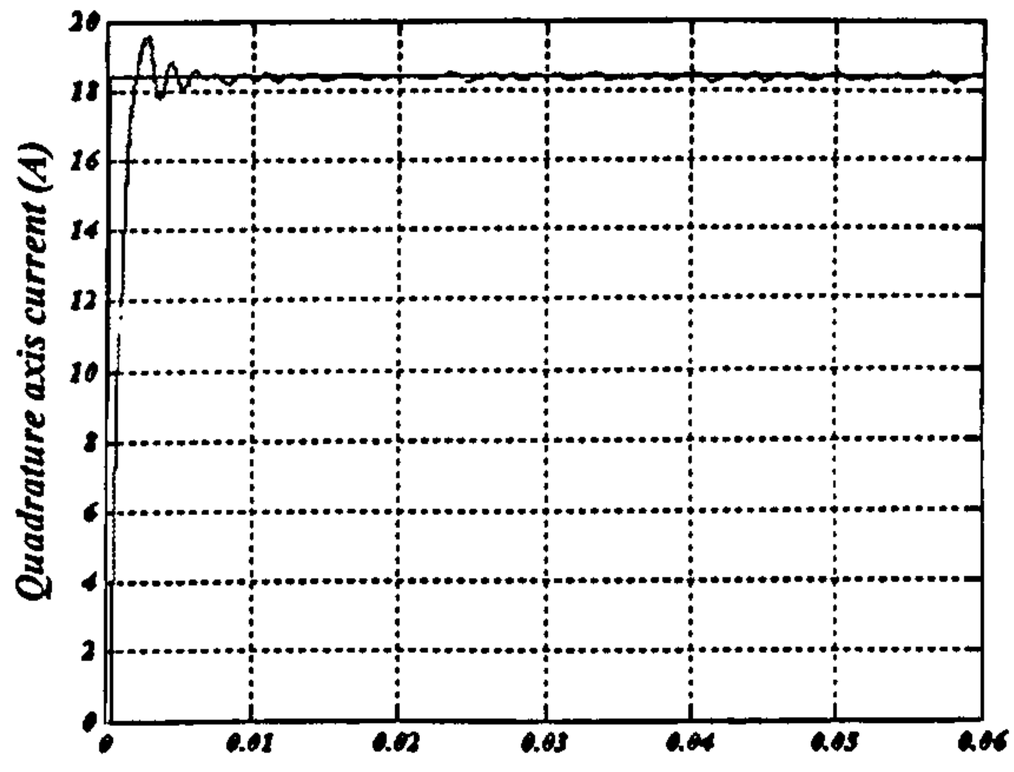
- pure sinusoid;
- flat topped; and
- unbalanced.

4.6.1 Pure sinusoidal supply

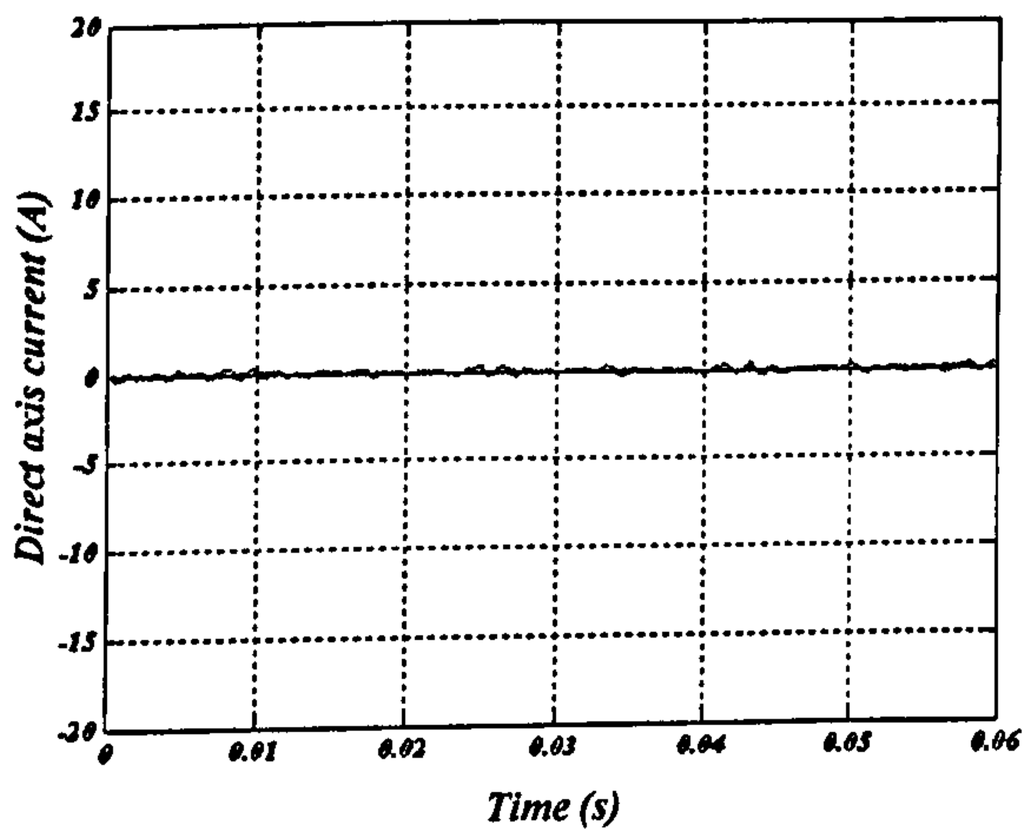
Fig. 4.26 parts a to f show the output performance of the matrix converter when controlled by the proposed technique. The output frequency is set at 20 Hz, the output reference current is 10 A (peak) which is equivalent to a modulation index of 0.77, the input displacement angle is zero (unity power factor), and the system switching frequency is 6.1 kHz. The controller gain are $k_p = 34, k_i = 5.125$. For the output stage, Fig. 4.23 parts a to c shows the fast transient response that can be achieved from the controller. Also the steady state error in the output current is zero. For the input stage, Fig. 4.23 d shows that the reference displacement angle has been modified to compensate for the effect of the input filter. A zero input displacement angle between the supply voltage (and not the input to the matrix converter) and the filtered input current is maintained as shown in Fig. 4.23 f.



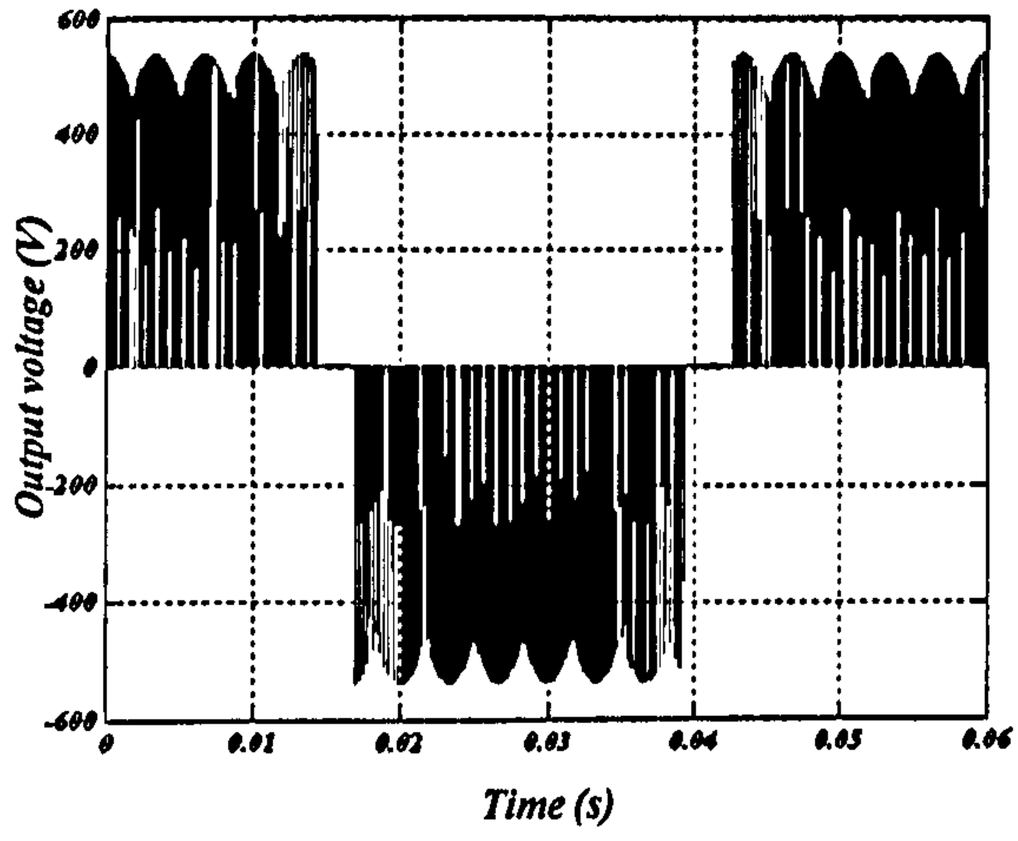
(a)



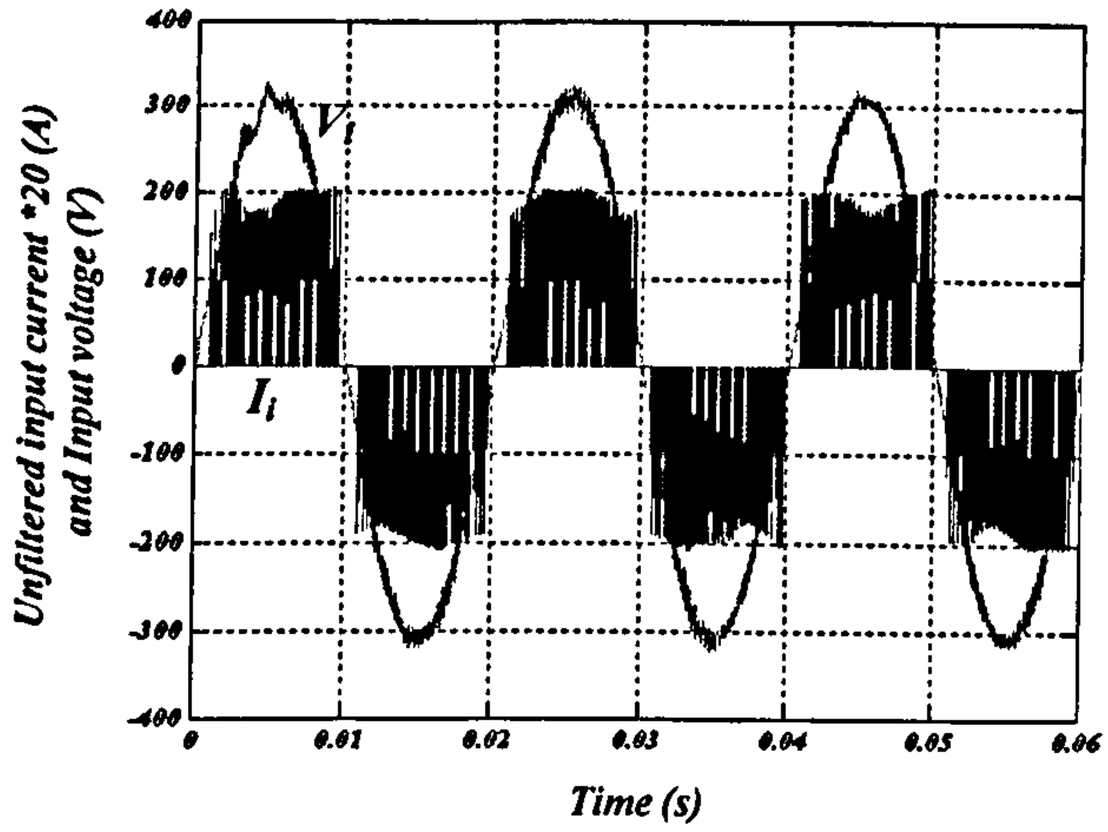
(b)



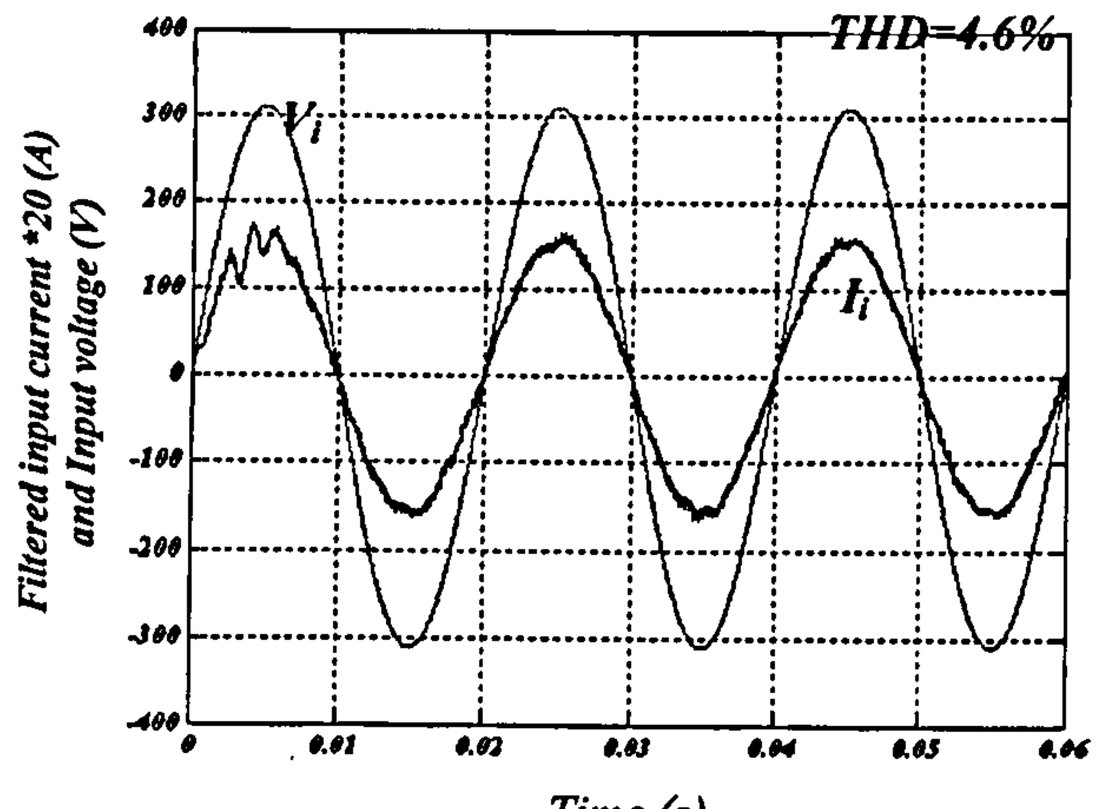
(c)



(d)



(e)

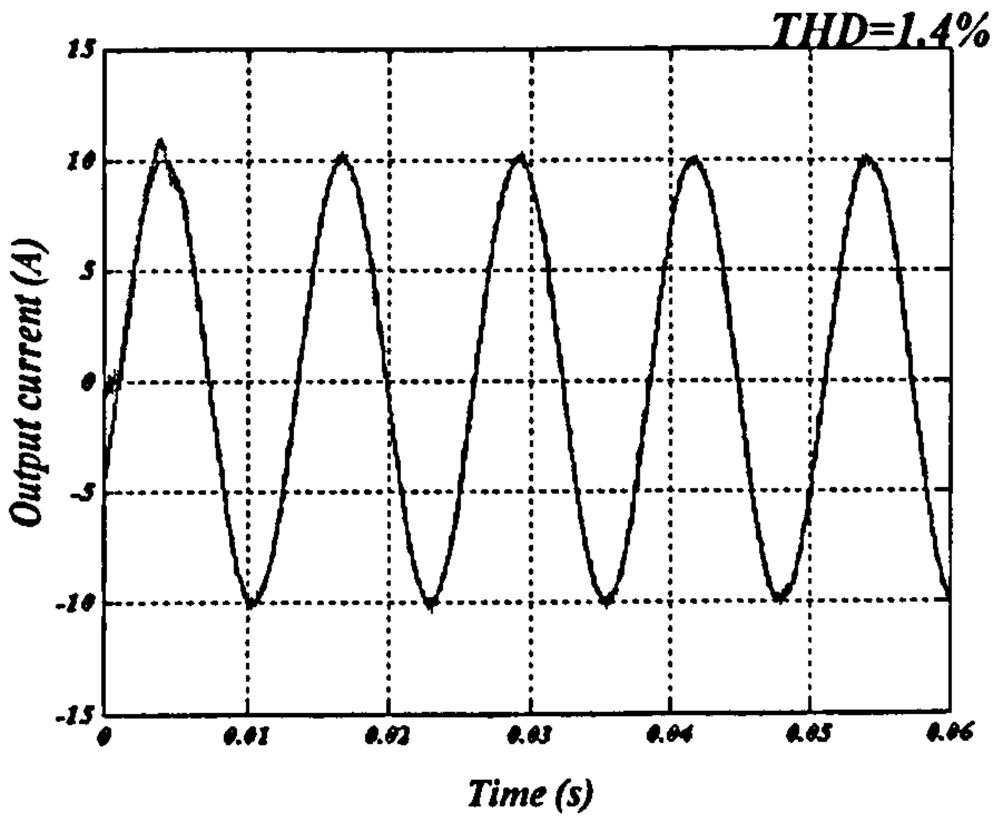


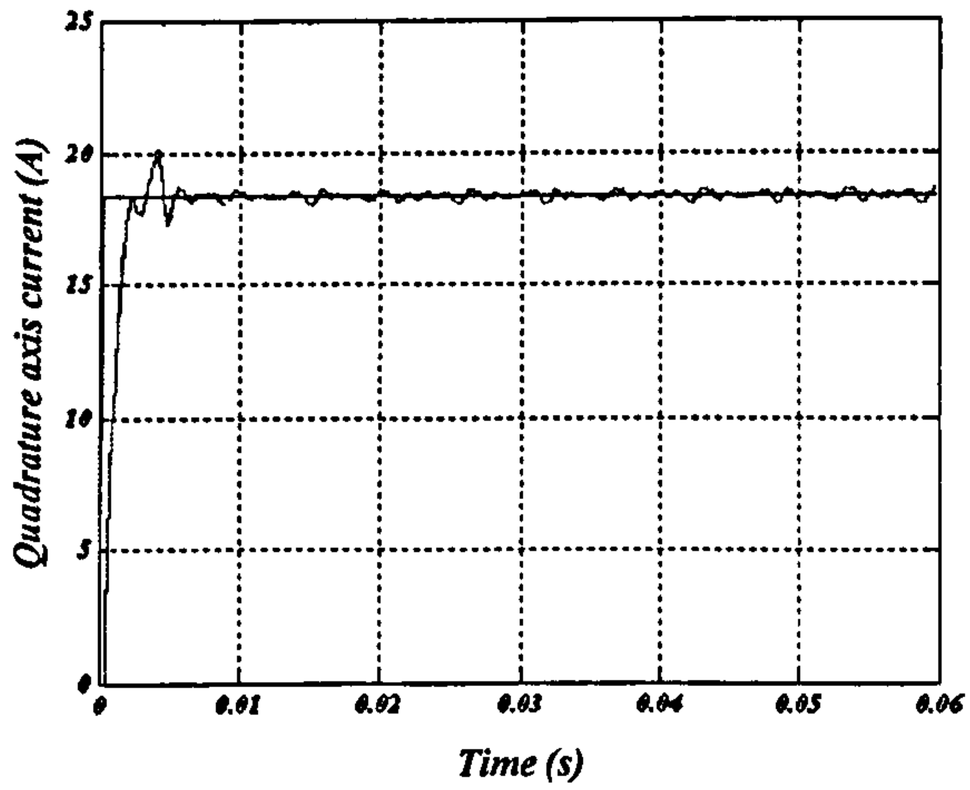
(f)

Fig. 4.23 Output response for f_o=20 Hz, unity input power factor at 6.1 kHz switching frequency and sinusoidal input voltage: (a) output current, (b) quadrature axis current, (c) direct axis current, (d) output line voltage, (e) unfiltered input current *20 and input voltage, and (f) filtered input current*20 and input voltage.

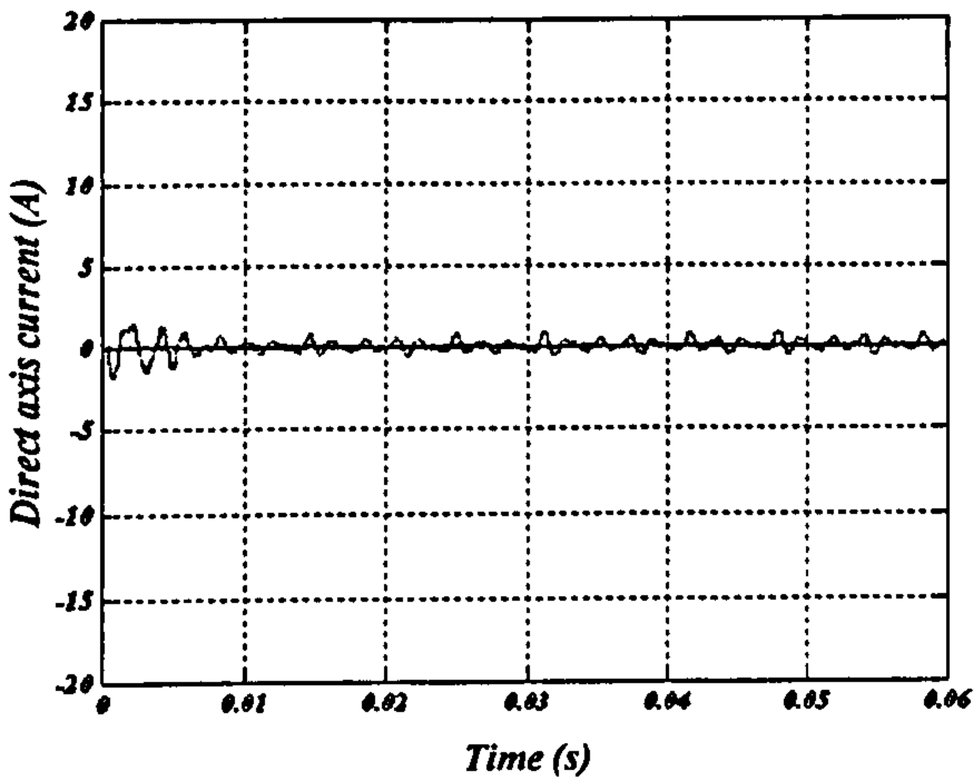
The 80 Hz output performance shown in Fig. 4.24 parts a to f, is almost the same as that at 20 Hz if the transient response and the steady state error are considered but the output waveform quality itself is not as good as that at 20 Hz. This is because the switching frequency is lower relative to 80 Hz. For the input stage, the input current is also slightly distorted when compared to that at 20 Hz. This is explained by referring to input filter magnitude performance shown in Fig. 4.17. Any harmonic component from the input current near the filter corner frequency (4500 rad/s; 716 Hz) will be amplified resulting in distortion in the input current waveform. If the filter corner frequency is increased the distortion can be reduced but the attenuation at the switching frequency will be decreased.

If the input power factor is set to 0.866 lagging and then leading, the corresponding results are shown in Figures 4.25 and 4.26 respectively. For the output stage, the reference current is lowered since the maximum output modulation index is affected by the input power factor variation. The results show the ability of the matrix converter to produce an output current at any input displacement angle.

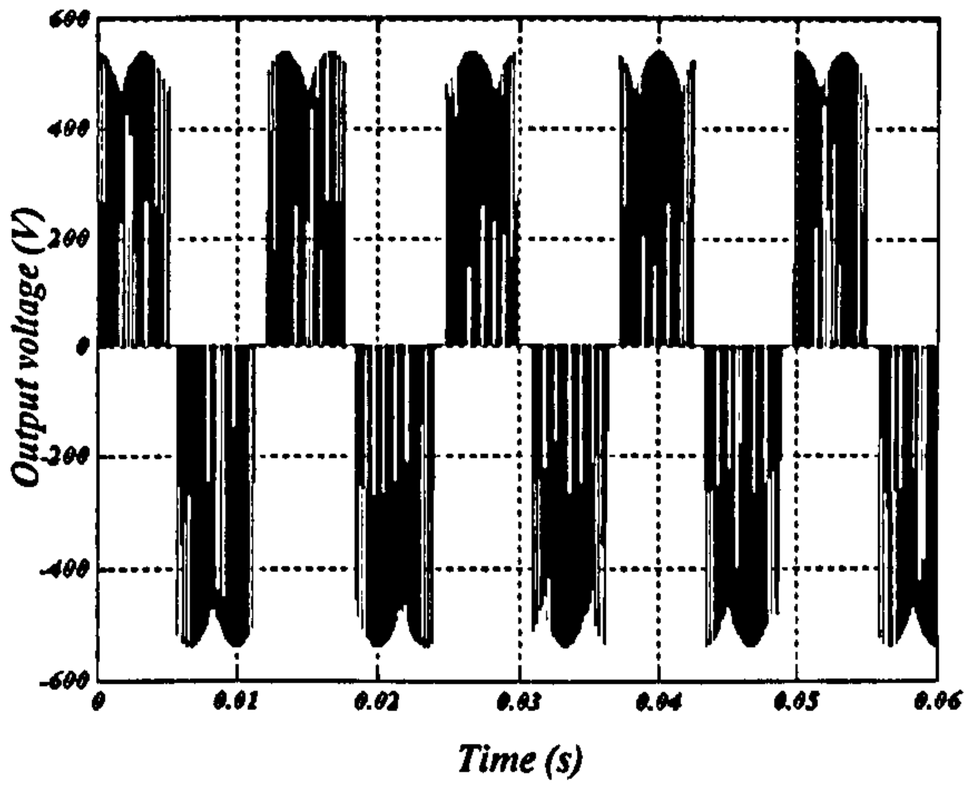




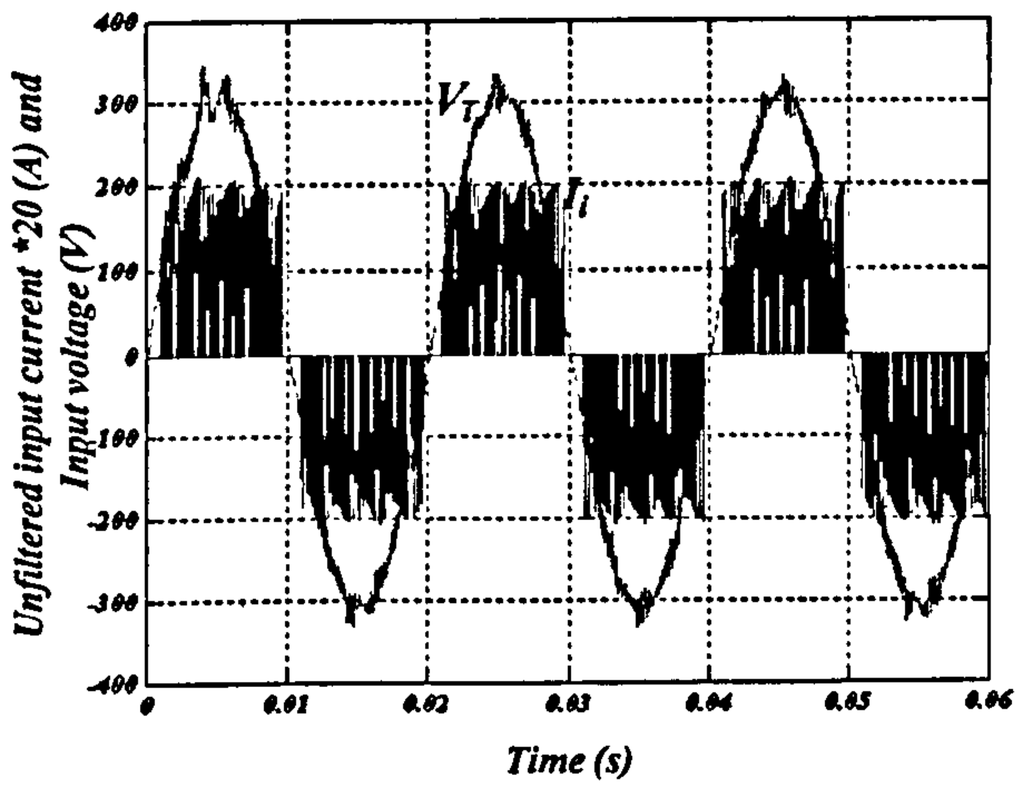
(a)



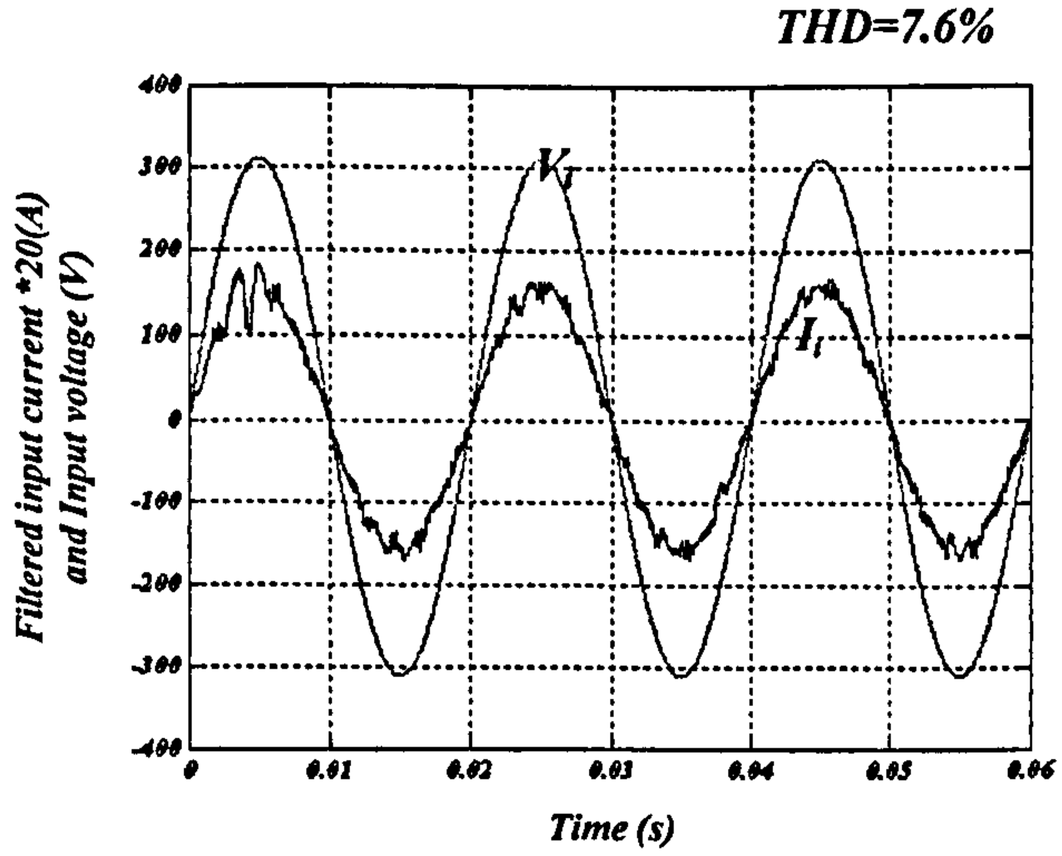
(b)



(c)

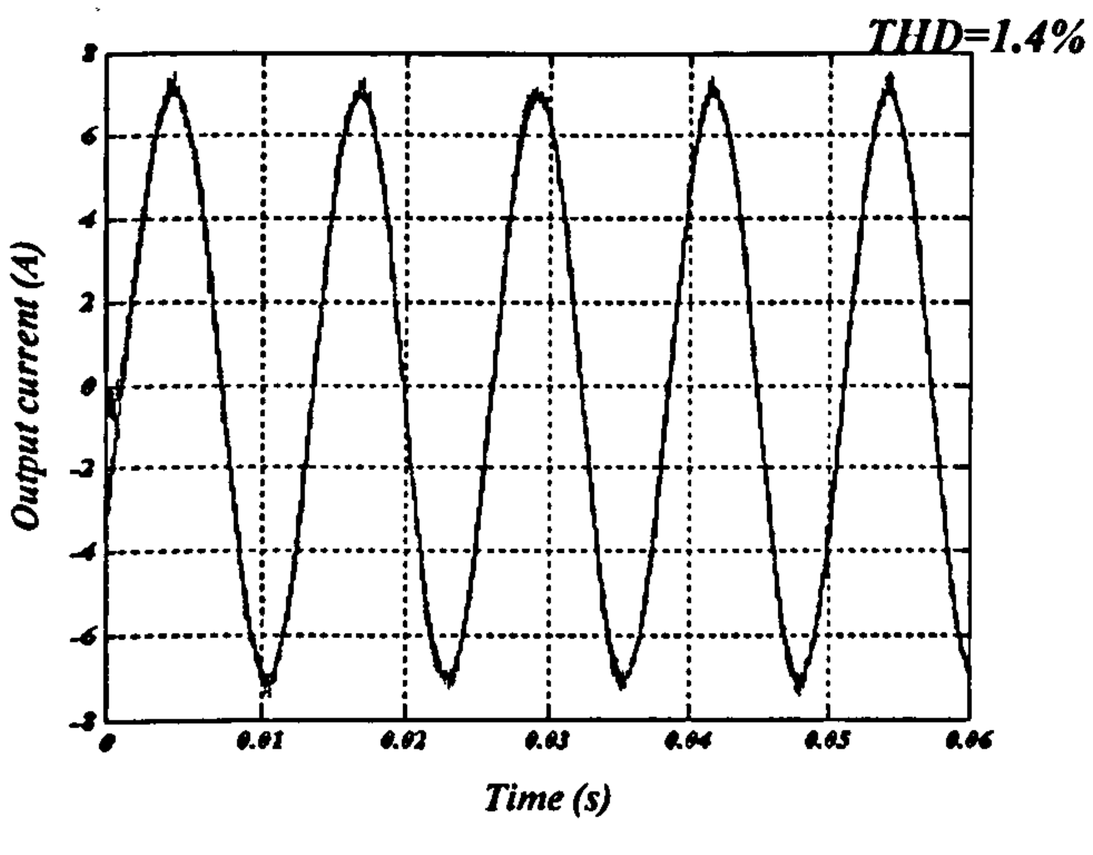


(d)

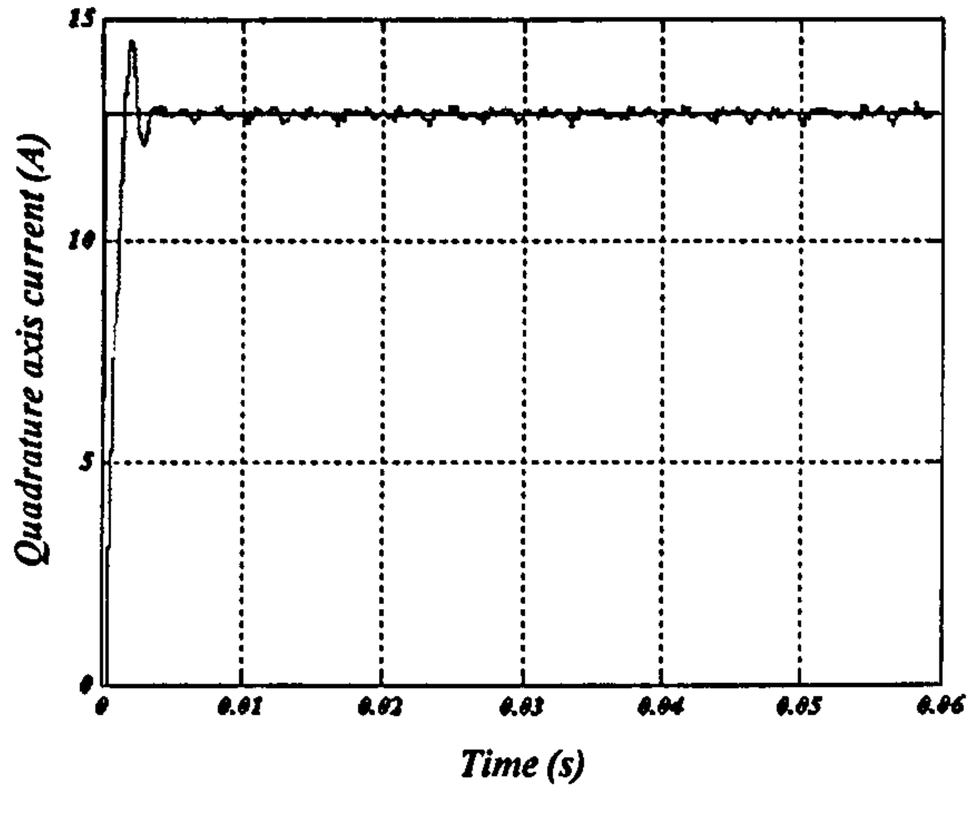


(e) (f)

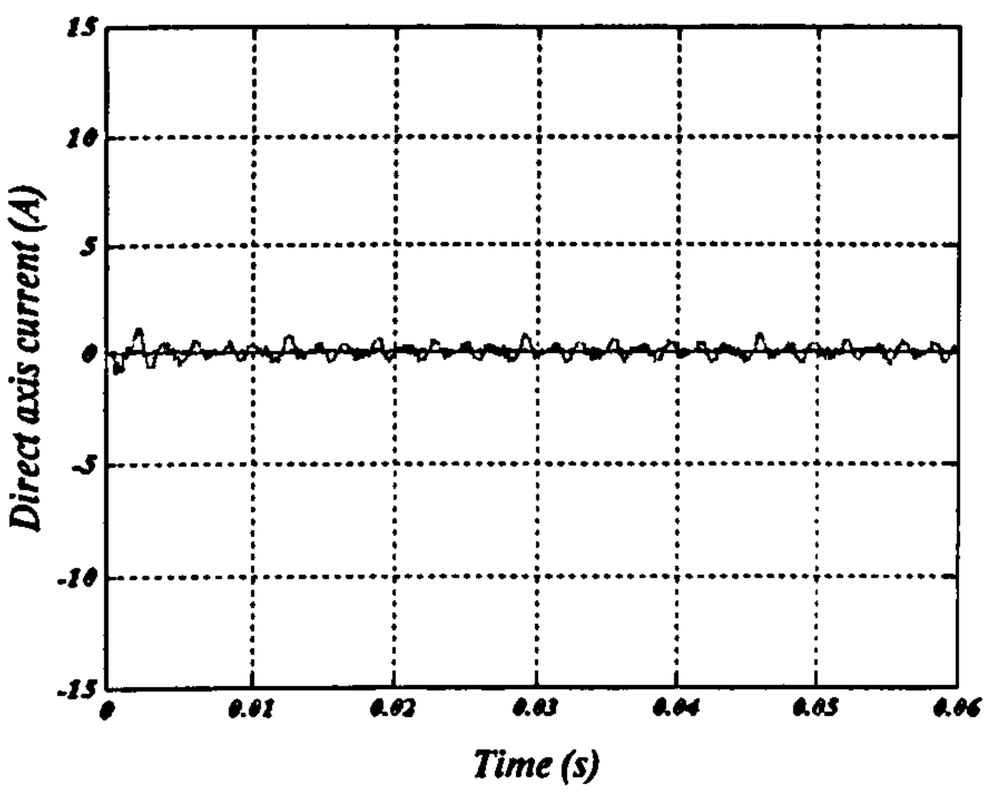
Fig. 4.24 Output response for f_o=80 Hz, unity input power factor at 6.1 kHz switching frequency and sinusoidal input voltage: (a) output current, (b) quadrature axis current, (c) direct axis current, (d) output line voltage, (e) unfiltered input current *20 and input voltage, and (f) filtered input current*20 and input voltage.



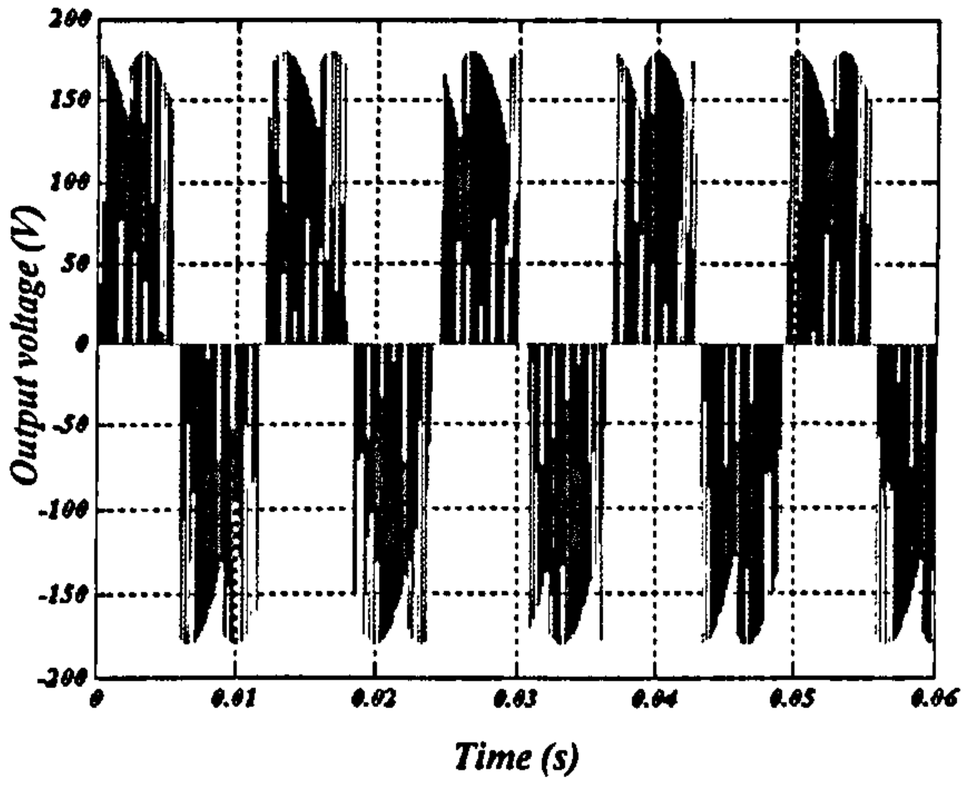
(a)



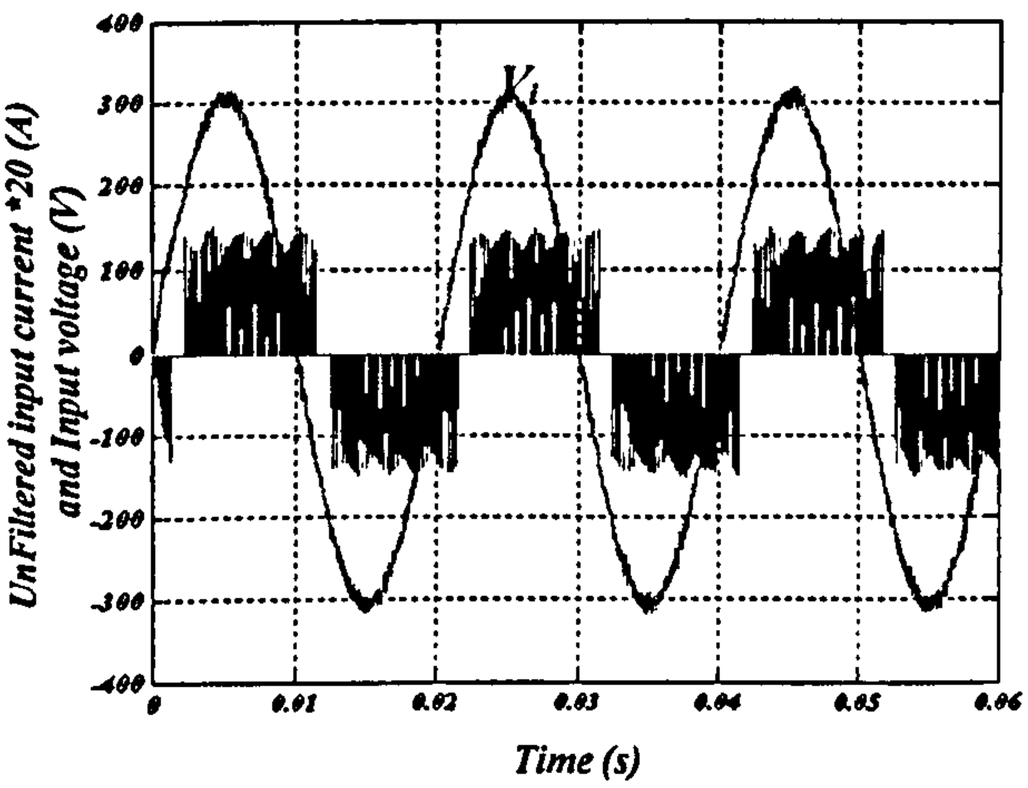
(b)



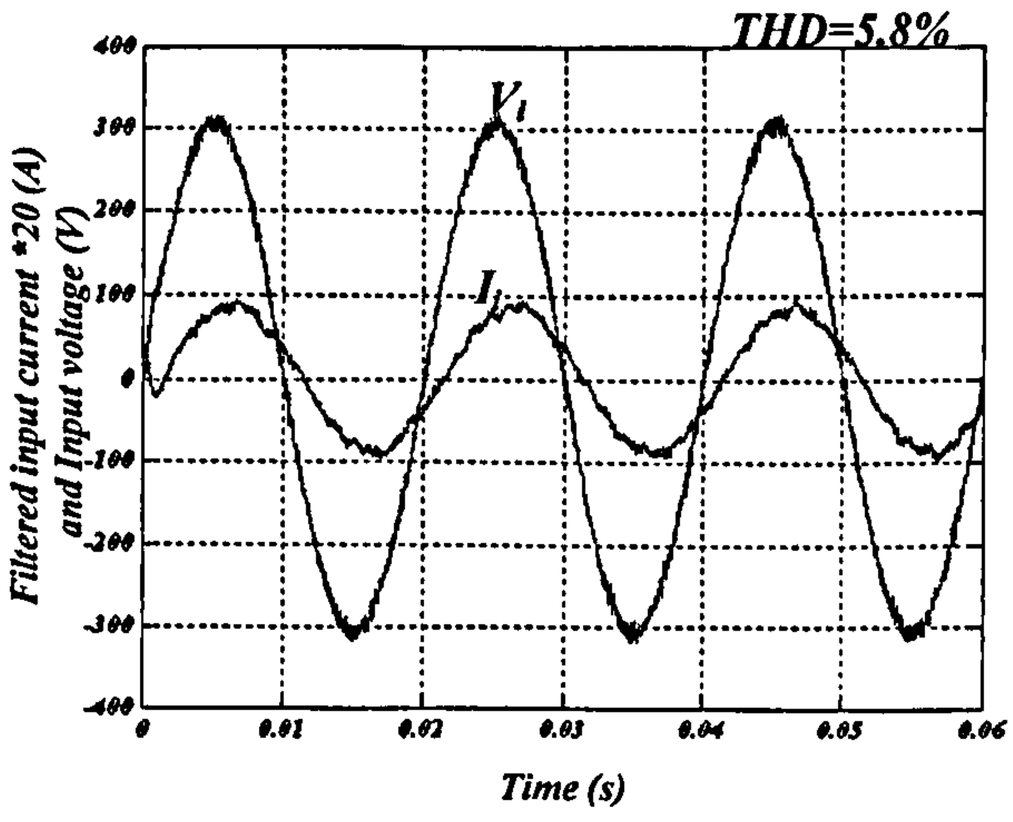
(c)



(d)

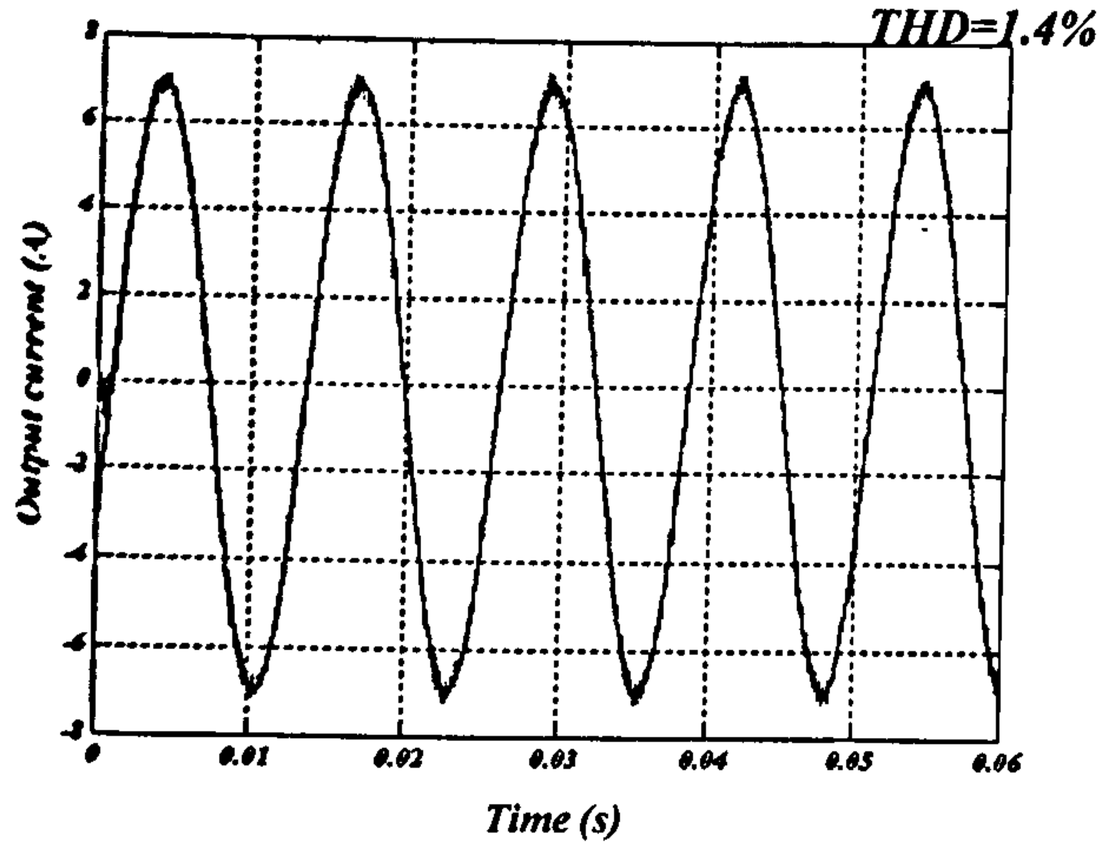


(e)

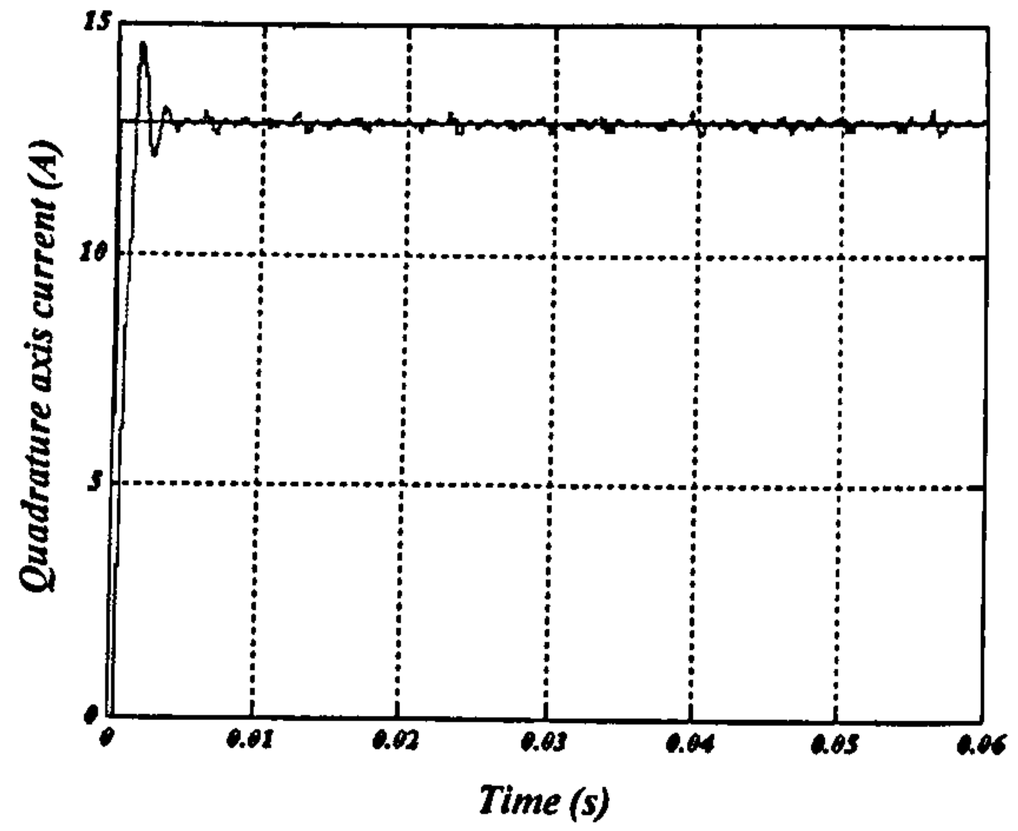


(f)

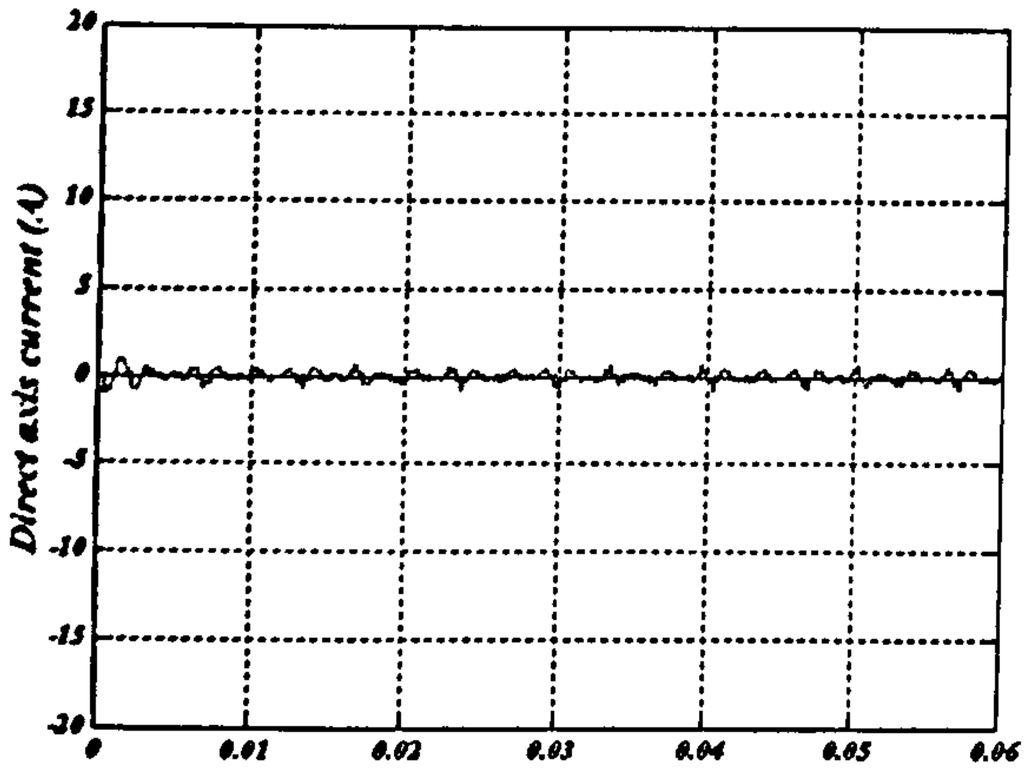
Fig. 4.25 Output response for f_o=80 Hz, 0.86 input power factor lagging at 6.1 kHz and sinusoidal input voltage: (a)output current, (b) quadrature axis current, (c) direct axis current, (d) output line voltage, (e) unfiltered input current *20 and input voltage, and (f) filtered input current*20 and input voltage.



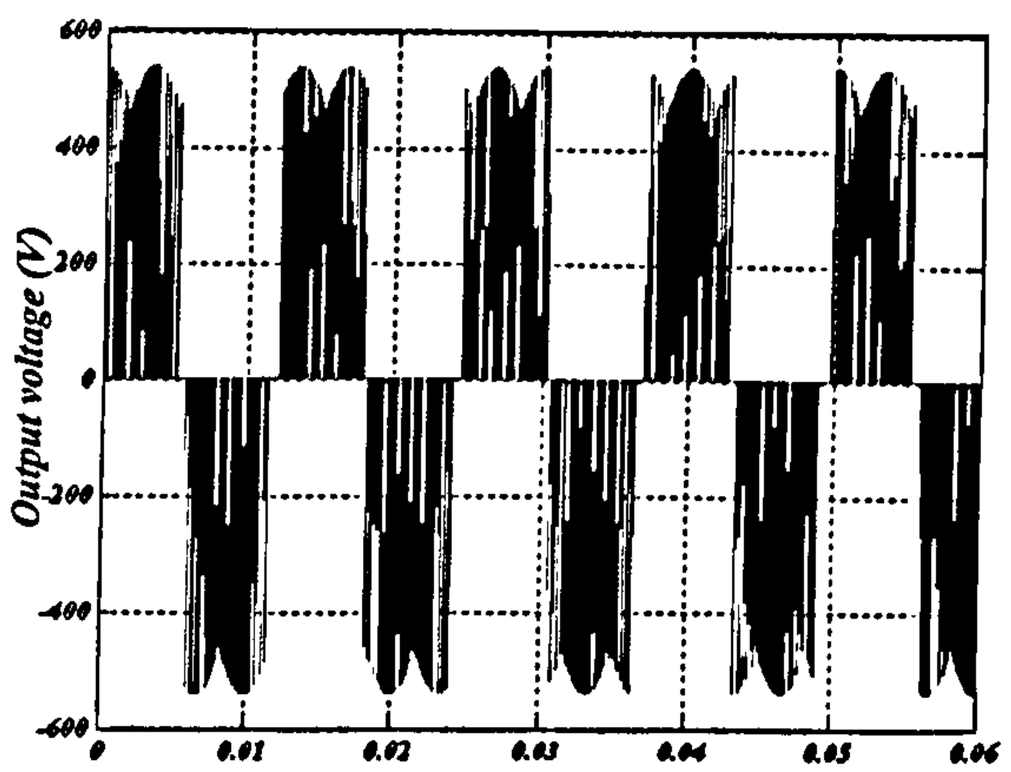
(a)



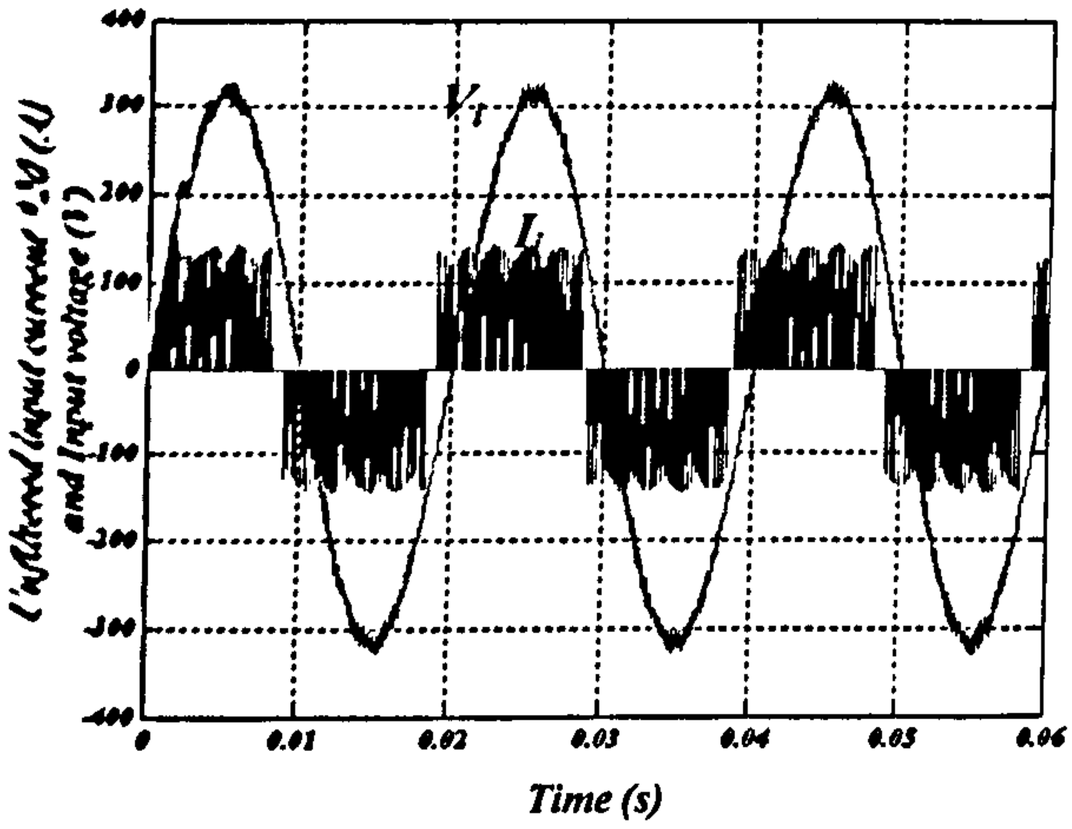
(b)



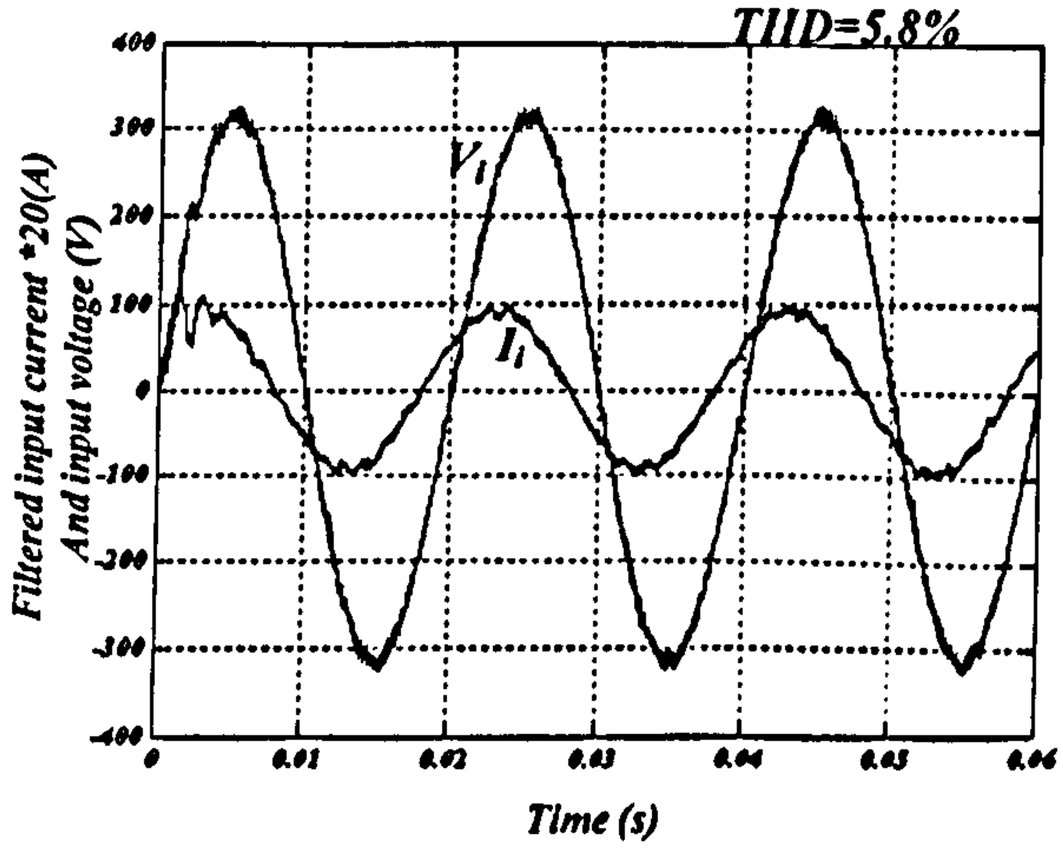
(c)



(d)



(e)



(f)

Fig. 4.26 Output response for f_o=80 Hz, 0.86 input power factor leading at 6.1 kHz and sinusoidal input voltage: (a) output current, (b) quadrature axis current, (c) direct axis current, (d) output line voltage, (e) unfiltered input current *20 and input voltage, and (f) filtered input current*20 and input voltage.

If a 12 kHz switching frequency is used, the input filter corner frequency attenuation margin is increased. The same filter topology is used as in Fig. 4.19c but with a fourth the capacitance (while keeping the inductances and resistance as per the previous designed values). The filter frequency response is shown in Fig.4.27. The current controllers are set at $k_p = 68, k_i = 5.125$.

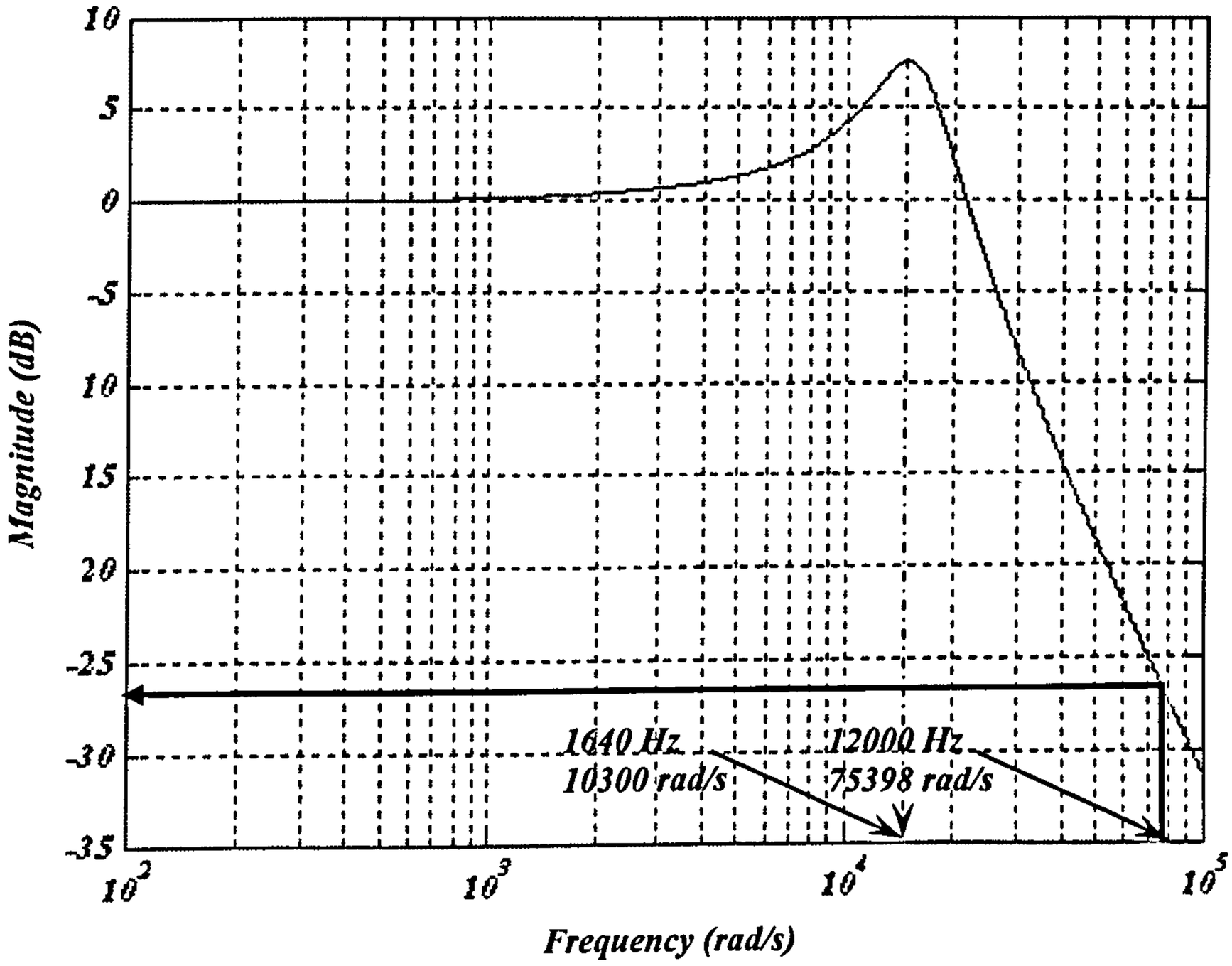


Fig. 4.27 Input filter frequency response

Considering the matrix converter response at 20 and 80 Hz with unity displacement angle, the output signals are as shown in figures 4.28 and 4.29 for 20 and 80 Hz, respectively. It is noticed that the output current quality has been enhanced. This is expected when increasing the switching frequency. The main advantage is that the distortion in the input current waveforms has been reduced.

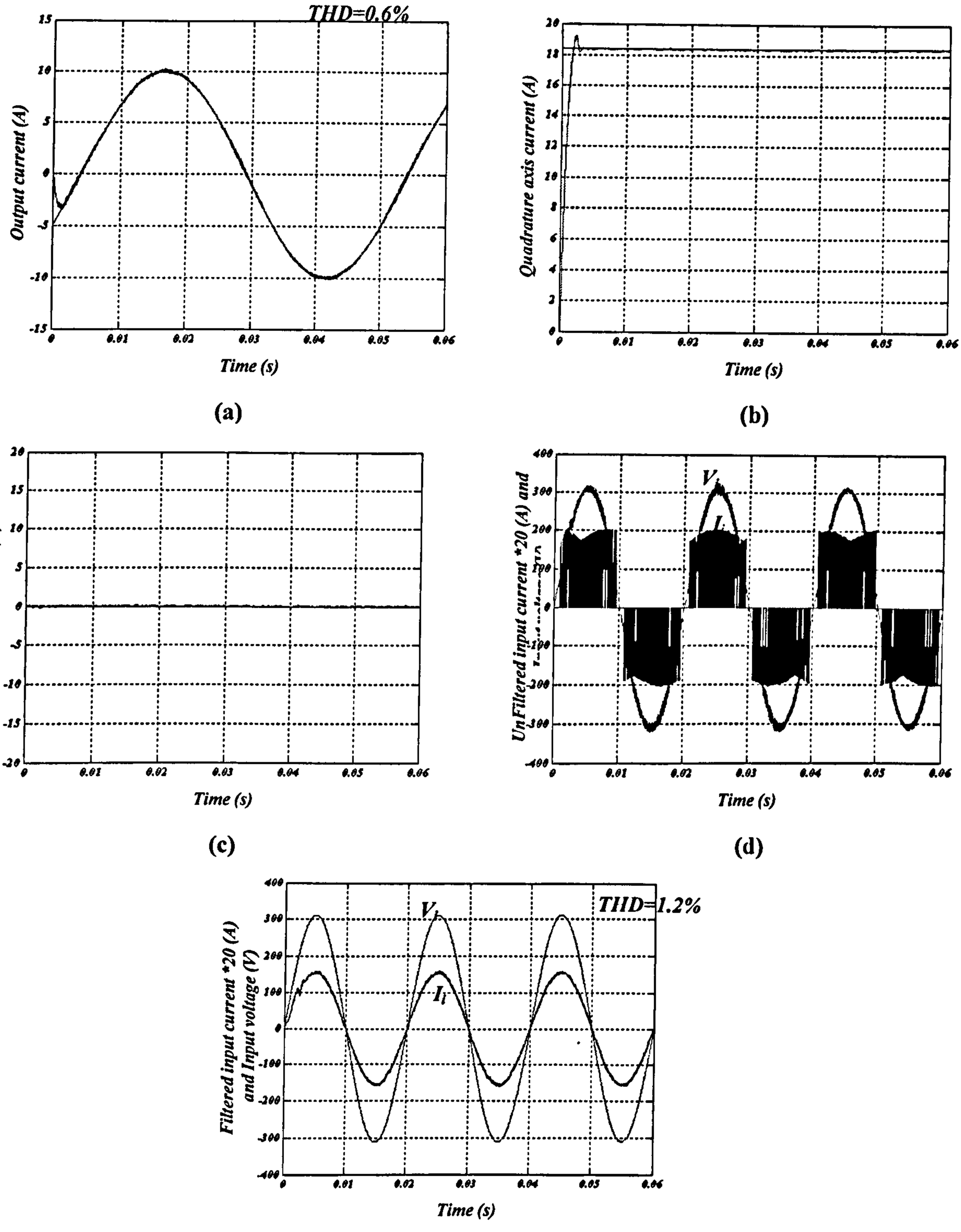
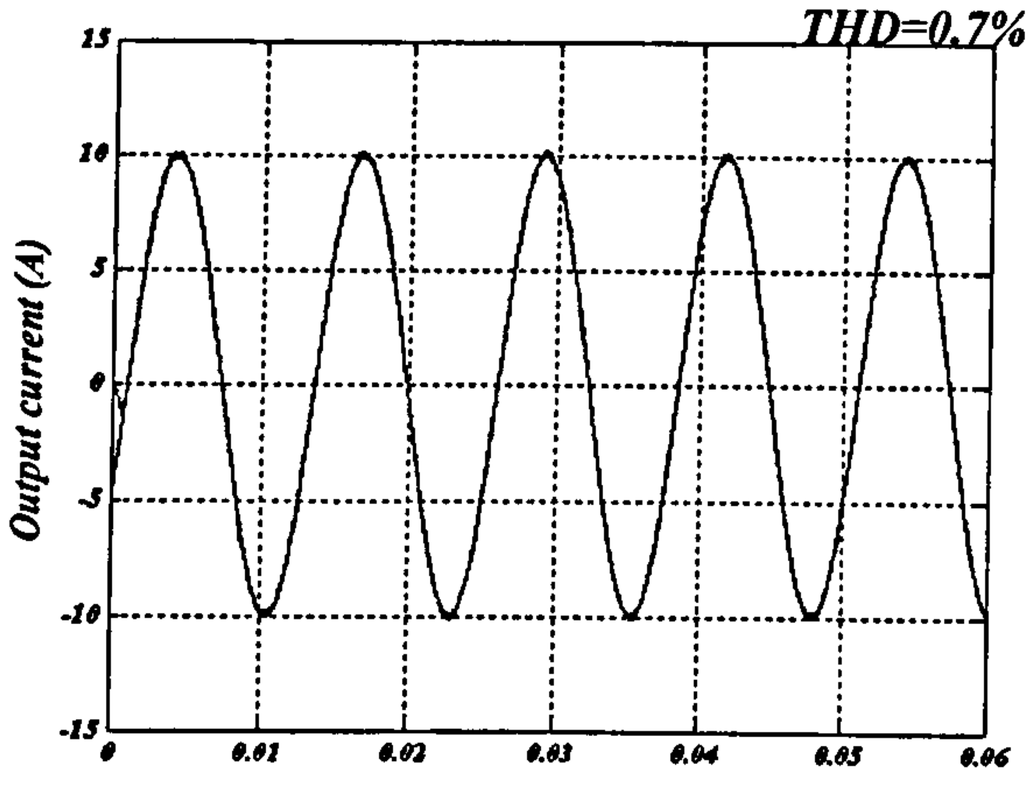
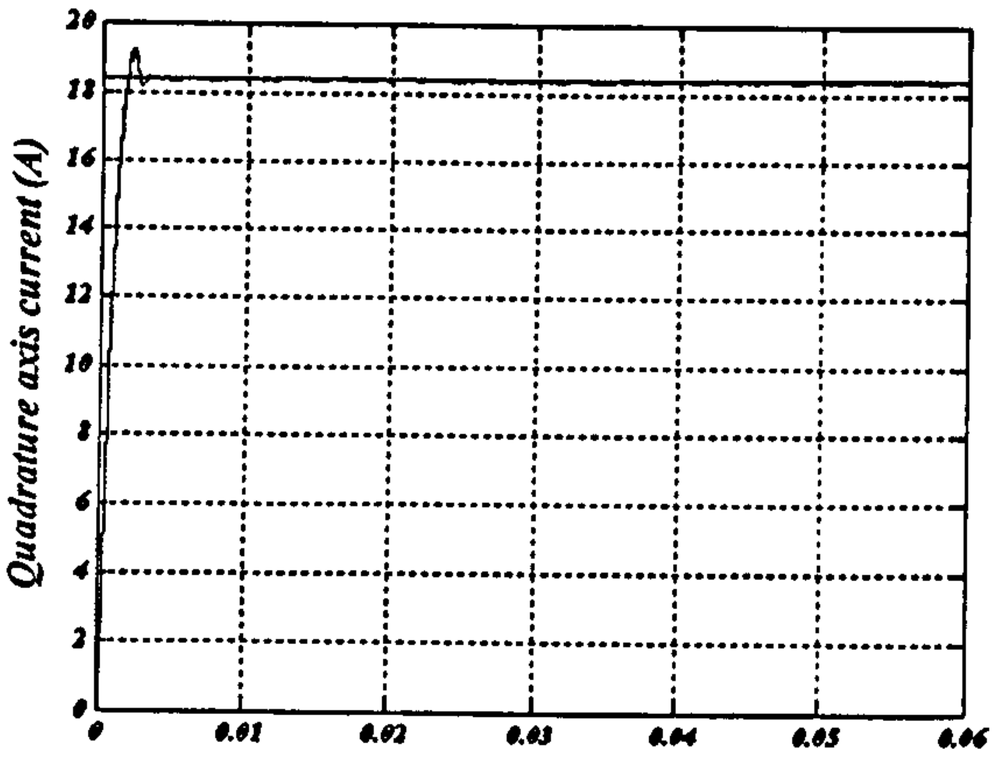


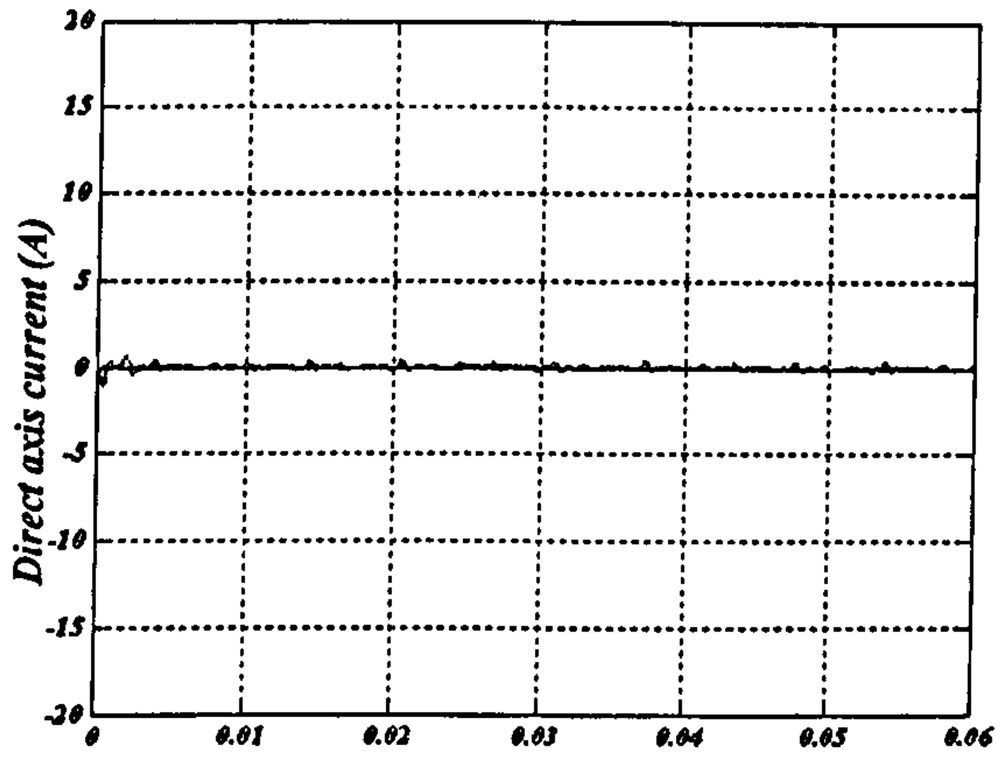
Fig. 4.28 Output response for f_o=20 Hz, unity input power factor at 12kHz and sinusoidal input voltage: (a) output current, (b) quadrature axis current, (c) direct axis current, (d) unfiltered input current *20 and input voltage, and (e) filtered input current*20 and input voltage.



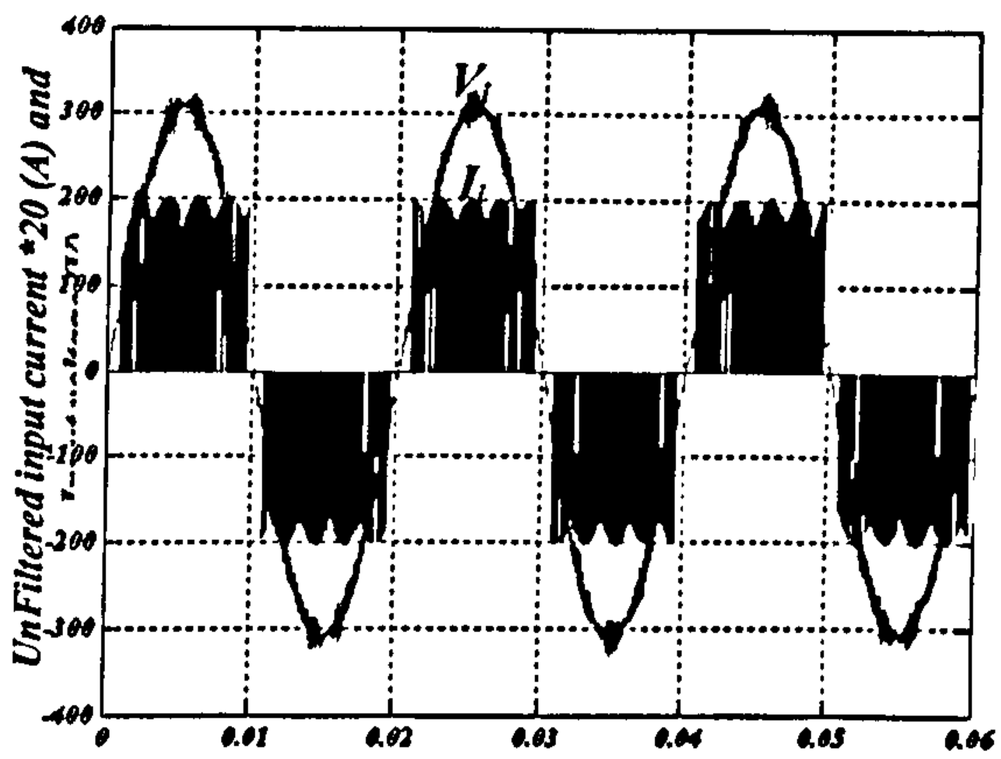
(a)



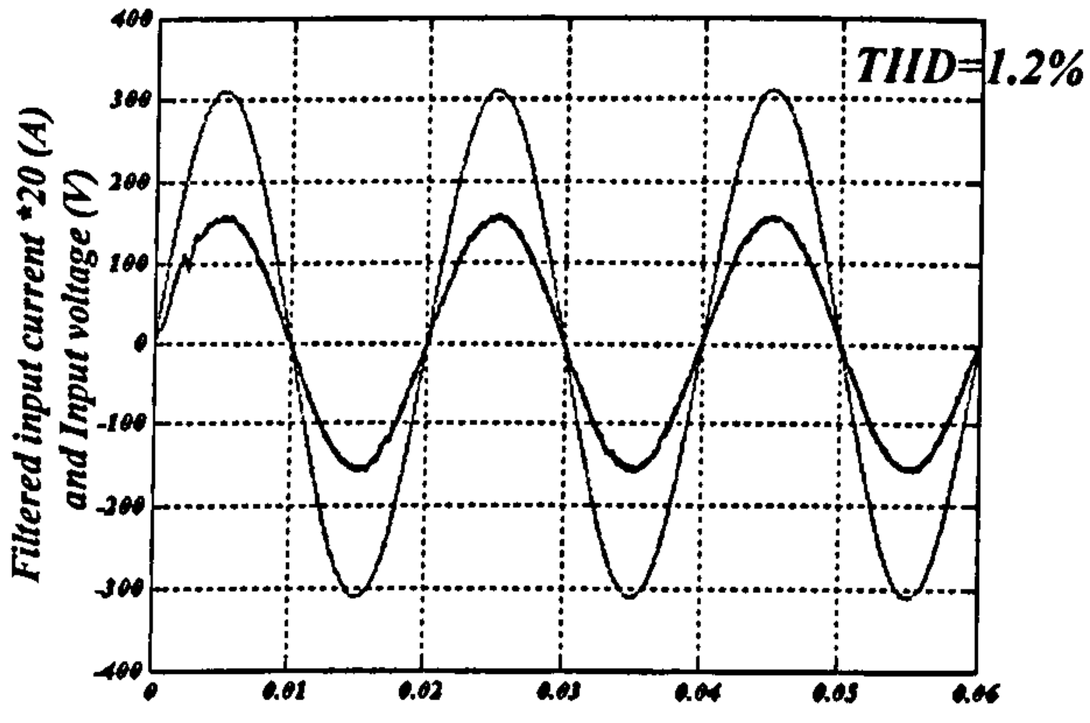
(b)



(c)



(d)



(e)

Fig. 4.29 Output response for f_o=80 Hz, unity input power factor at 12 kHz and sinusoidal input voltage: (a) output current, (b) quadrature axis current, (c) direct axis current, (d) unfiltered input current *20 and input voltage, and (e) filtered input current*20 and input voltage.

Another benefit from increasing the filter corner frequency by decreasing the filter capacitance, is the reduction of the no load filter current. This increases the ability to control the input displacement angle of the system at low modulation indices.

4.6.2 Unbalanced supply voltage

A power supply with unbalanced voltage magnitudes (10% unbalance) as shown in fig 4.30 is connected to the matrix converter input terminals.

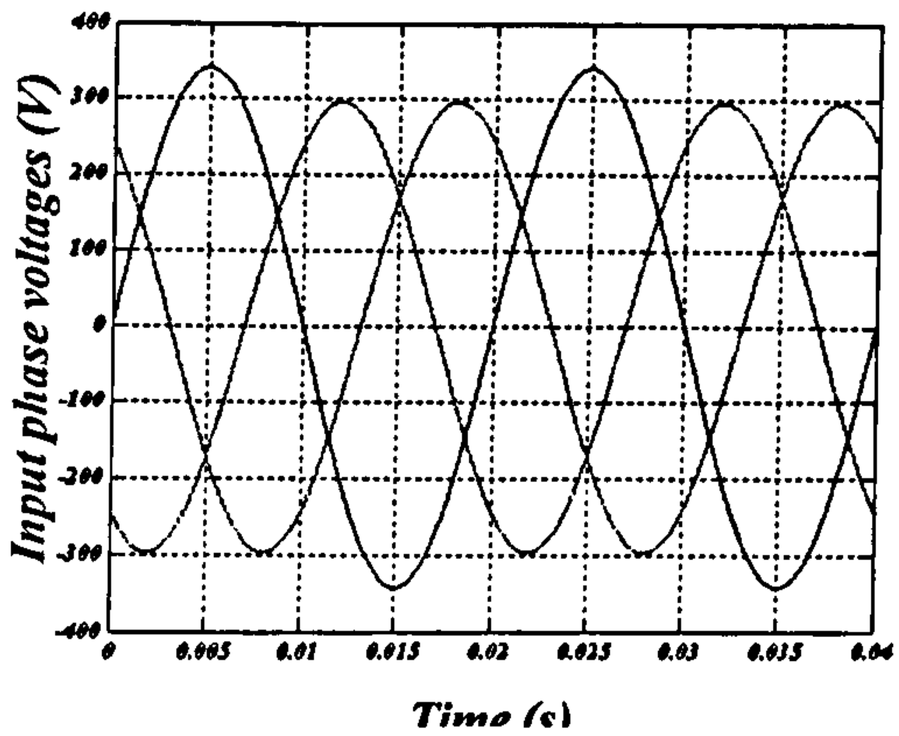
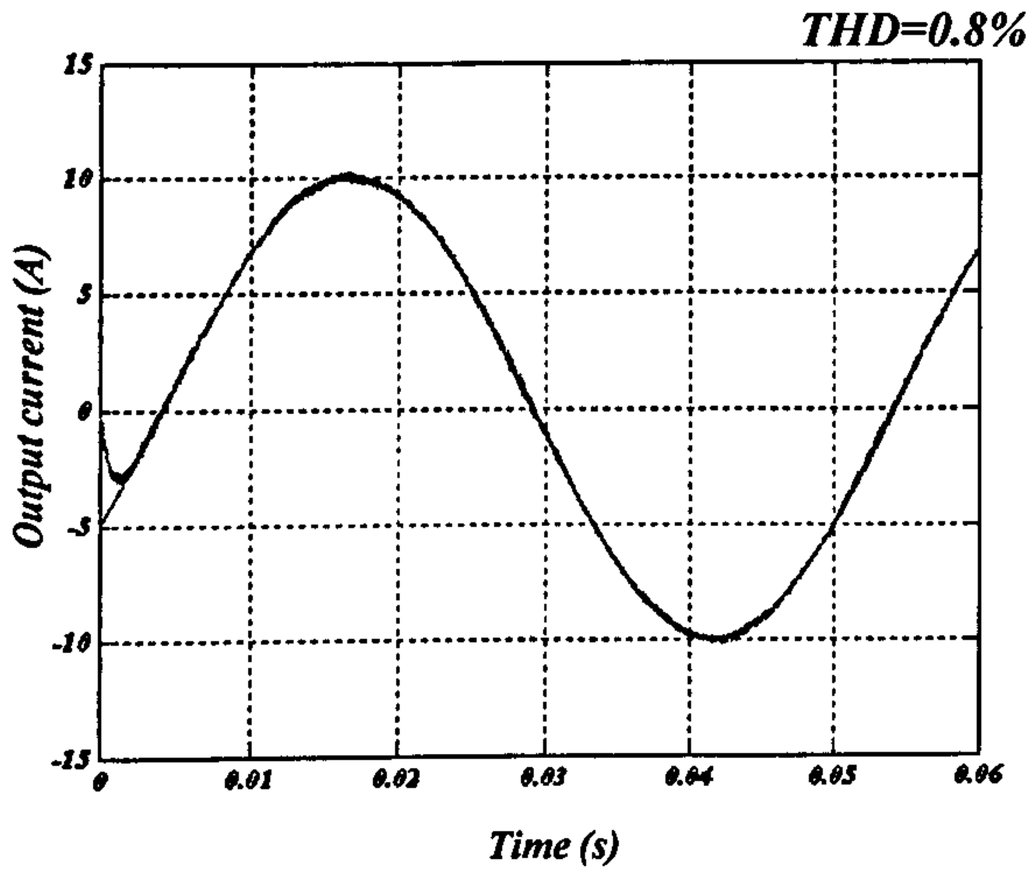


Fig. 4.30 Unbalanced voltage supply.

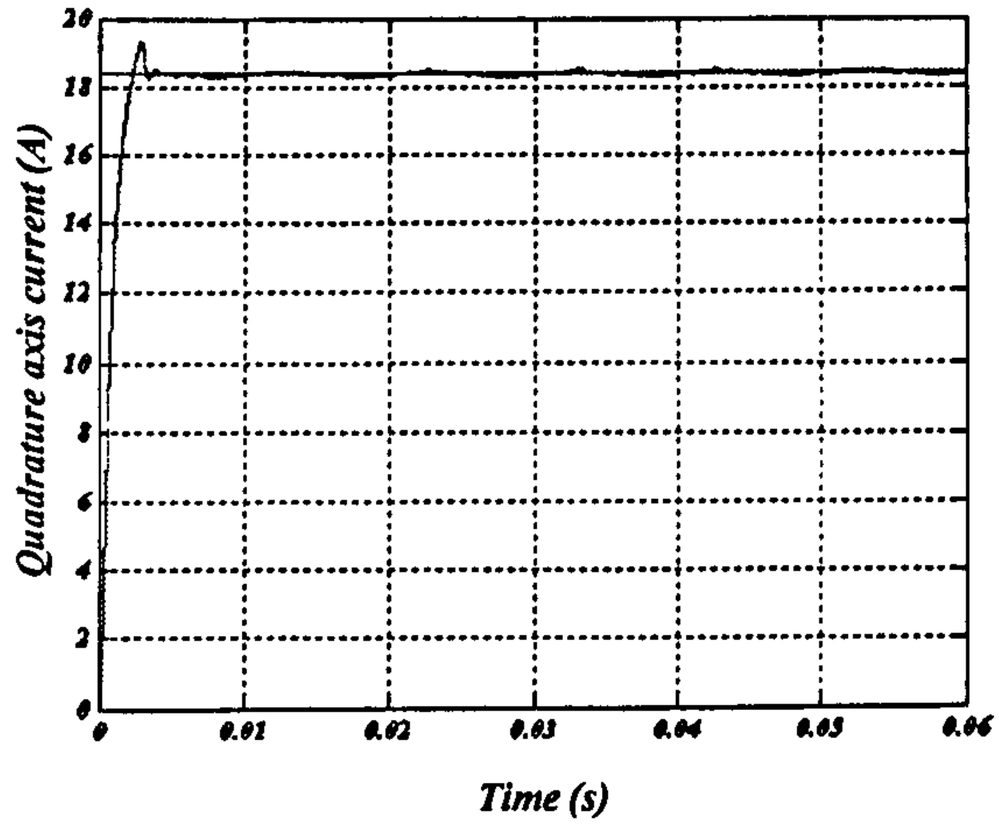
The system is simulated with an output reference current of 10 A peak, zero input displacement angle, and a 12 kHz switching frequency as shown in Fig. 4.31 for 20 Hz and Fig. 4.32 for an 80 Hz output. A sinusoidal output current can be obtained at 20 Hz with a zero displacement input angle. But the input current is no longer a pure sinusoid. It has second order harmonic components. This can be explained after considering the power balance in the system. Since the output currents are balanced and pure sinusoids, the instantaneous output power is constant. This should be the same for that power drawn from the input side. If a balanced and pure sinusoidal input current is drawn from the input side, then the input power is not constant in the case of an unbalanced supply voltage.

The input power has two components: one a positive sequence input voltage which will be constant and the other a negative sequence voltage which will be oscillatory. Hence it is not possible to achieve balanced sinusoidal input current unless the quality of the output side is compromised. At 80 Hz the output current is slightly distorted; this is mainly due to the higher output voltage required which means a higher modulation

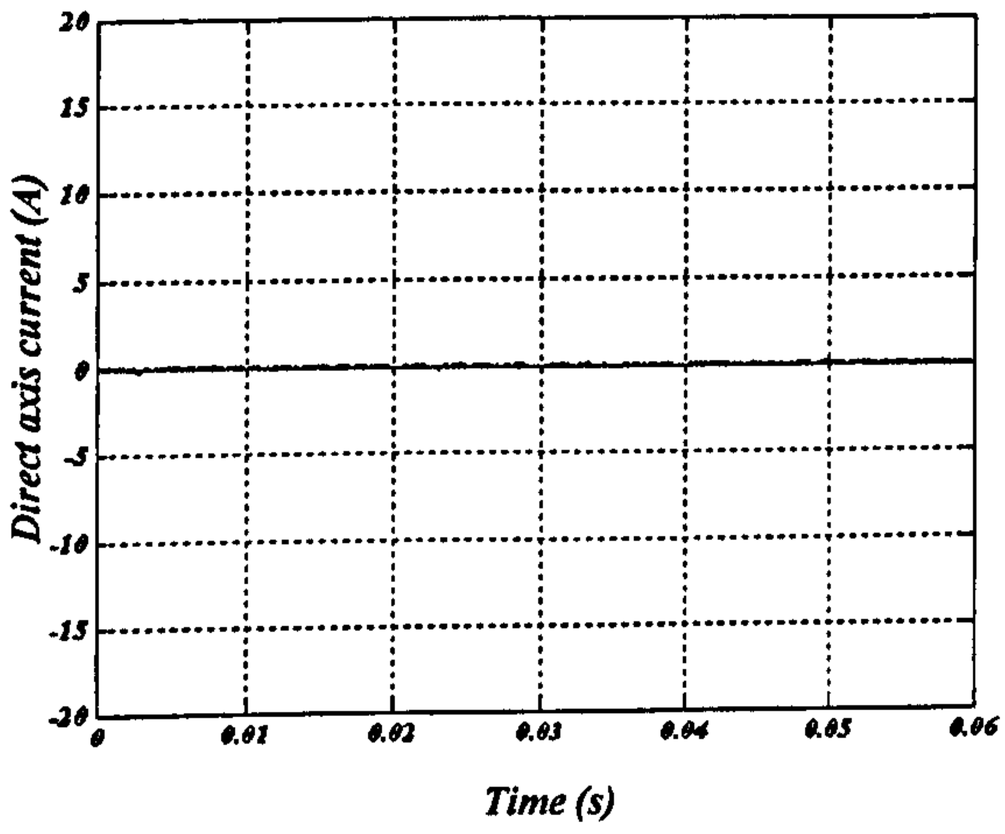
index, 0.84 rather than 0.78 in the 20 Hz case which cannot be achieved at certain periods with the unbalance in the supply voltage.



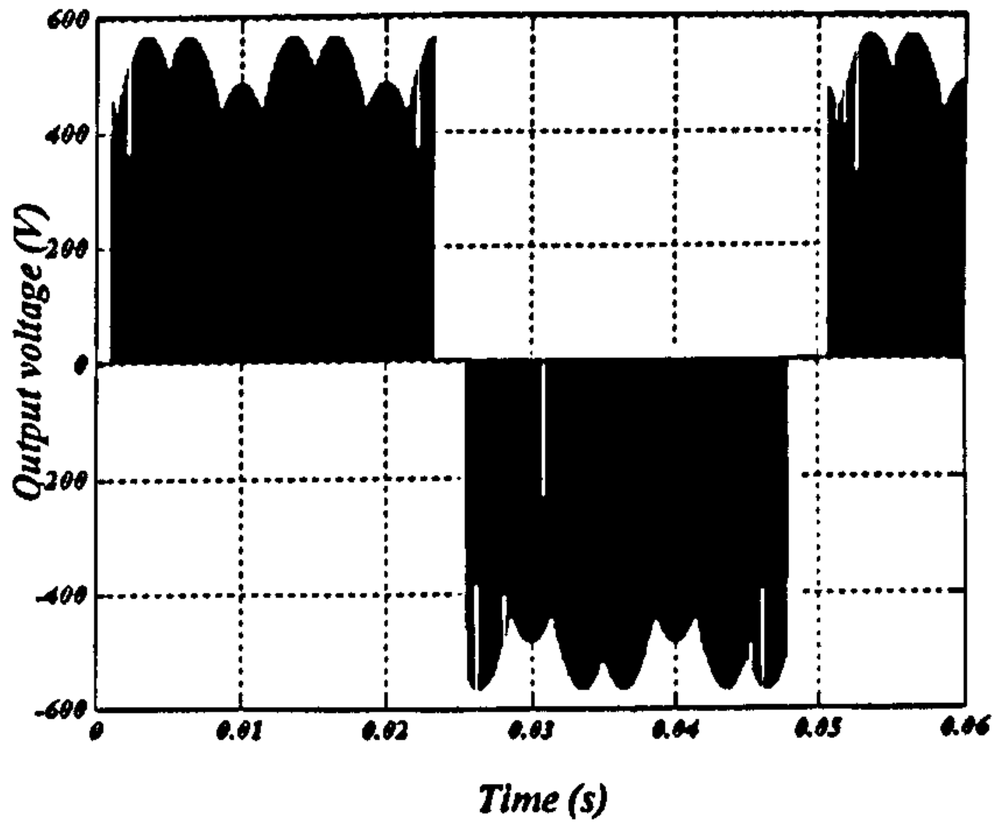
(a)



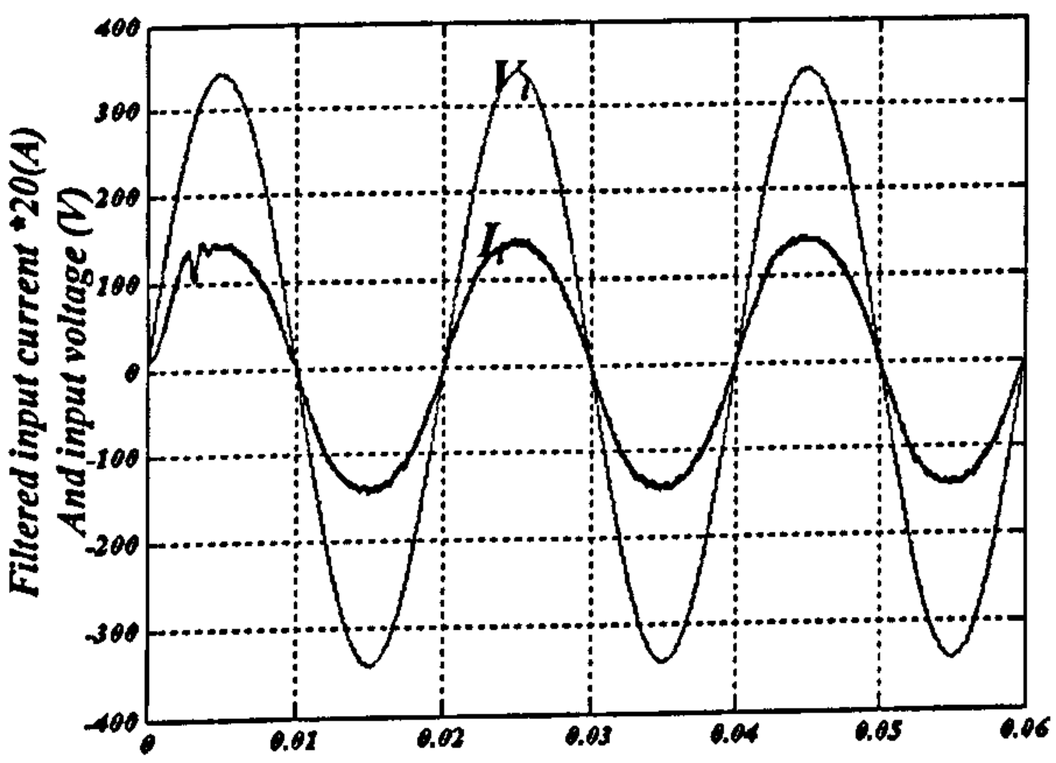
(b)



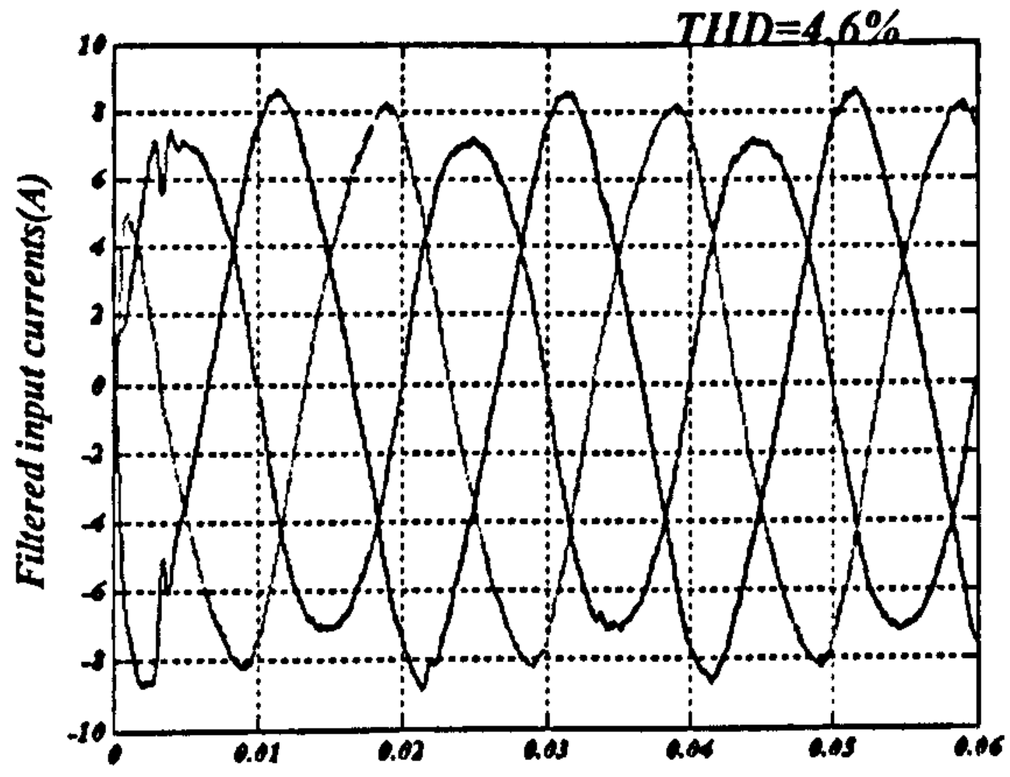
(c)



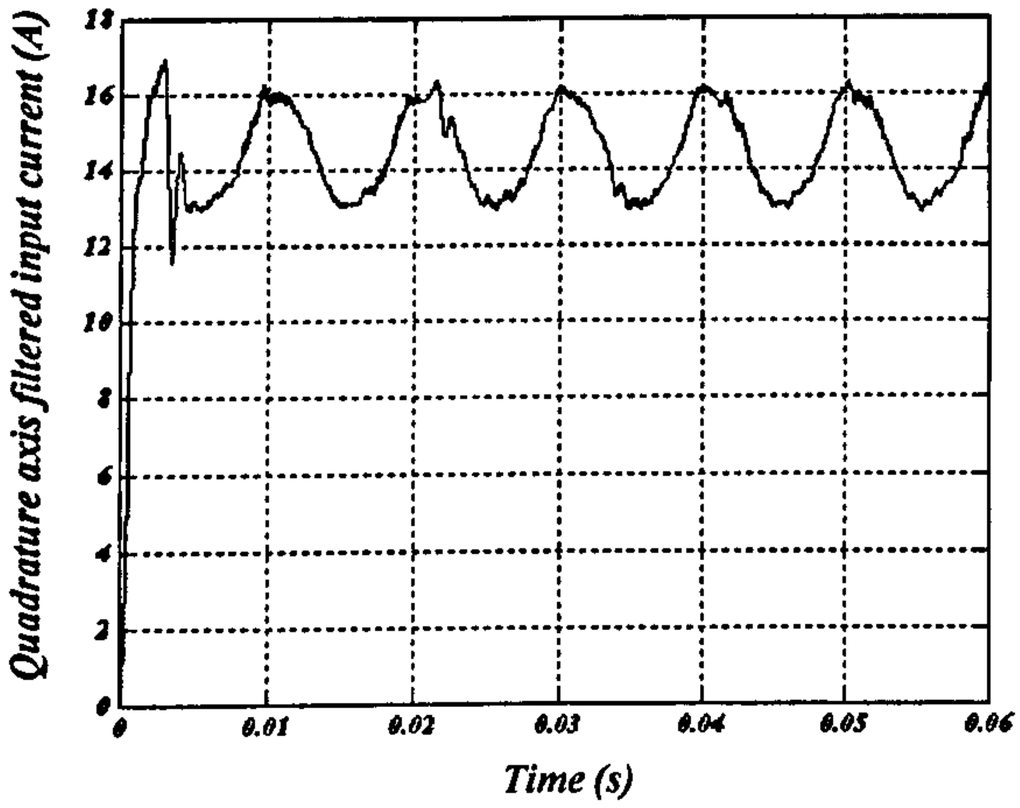
(d)



(e)



(f)



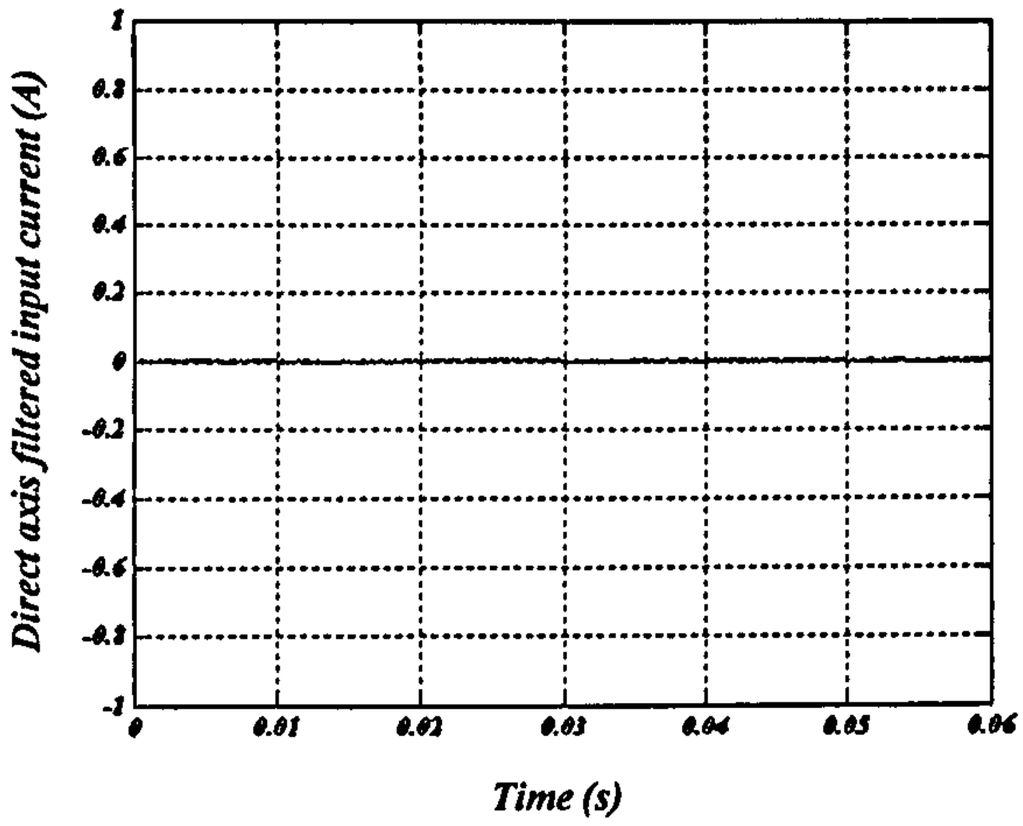
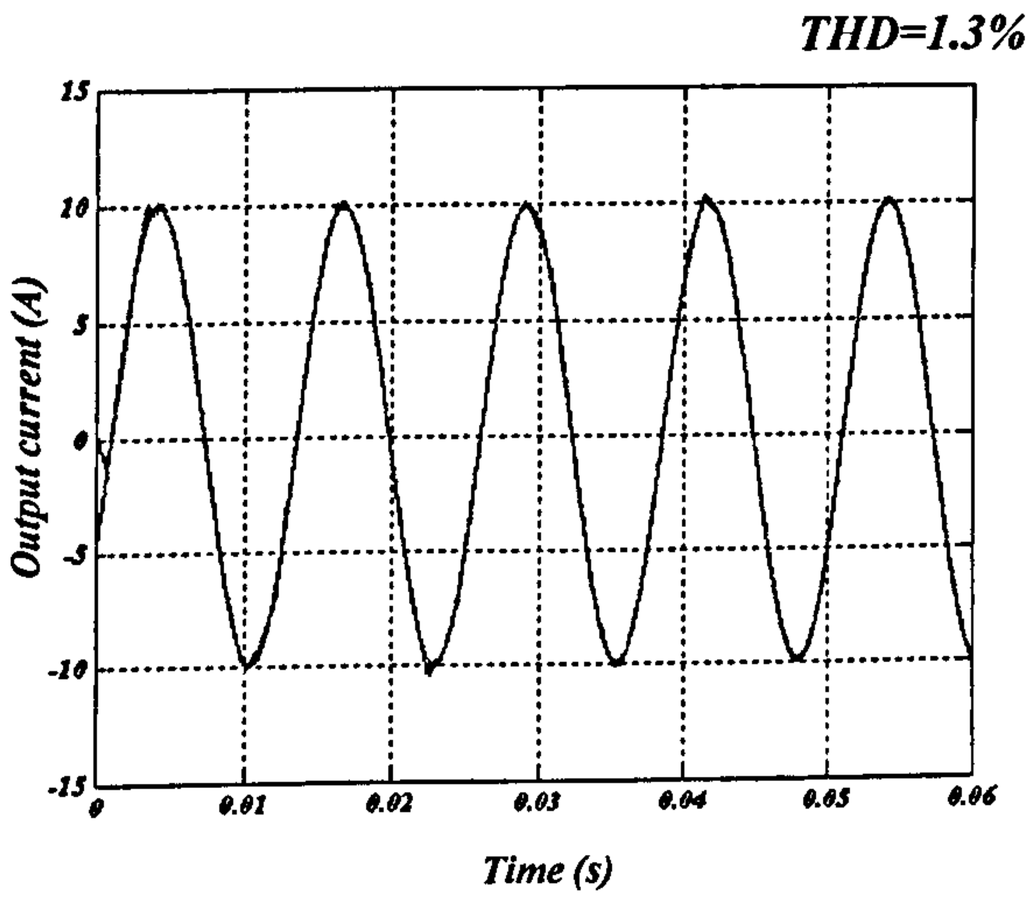
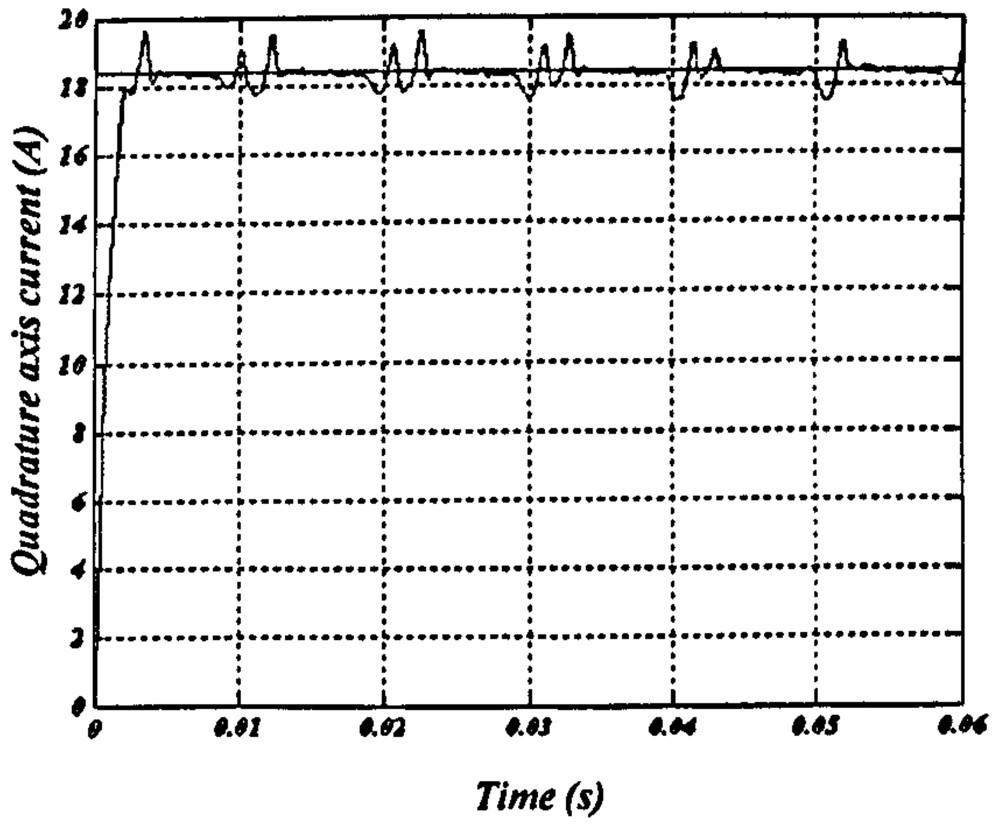
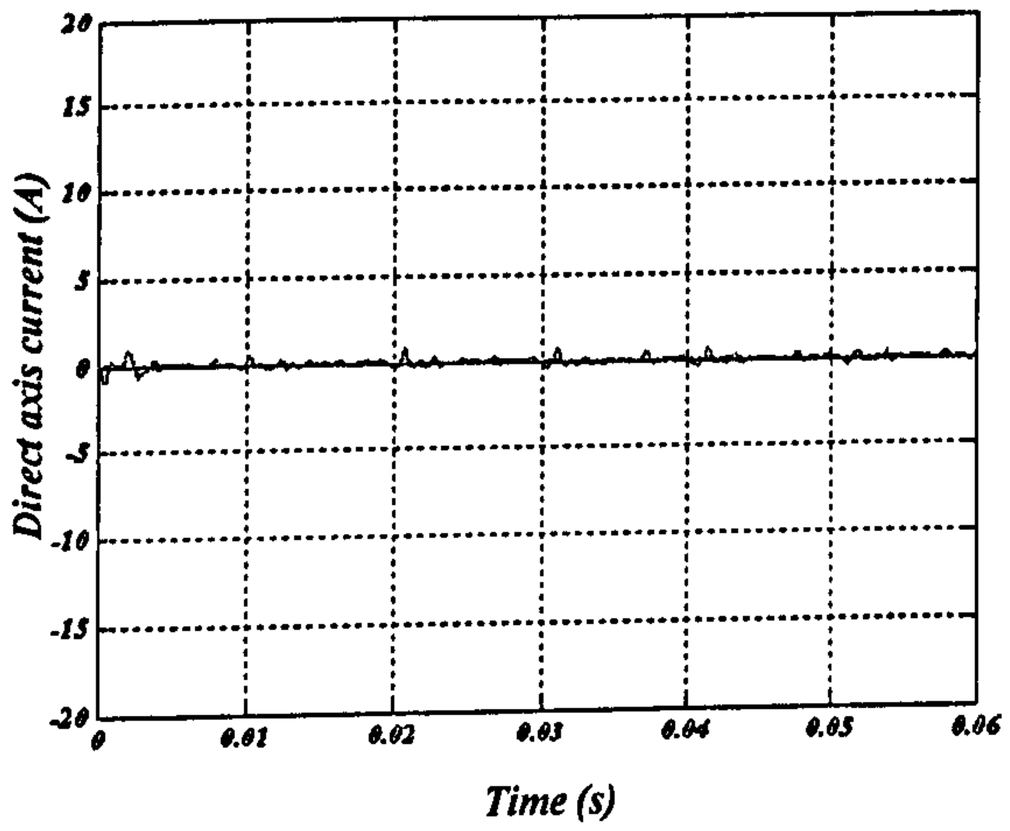


Fig. 4.31 Output response for f_o=20 Hz, unity input power factor at 12 kHz and unbalanced input voltage: (a) output current, (b) quadrature axis current, (c) direct axis current, (d) unfiltered input current *20 and input voltage, and (e) filtered input current*20 and input voltage.

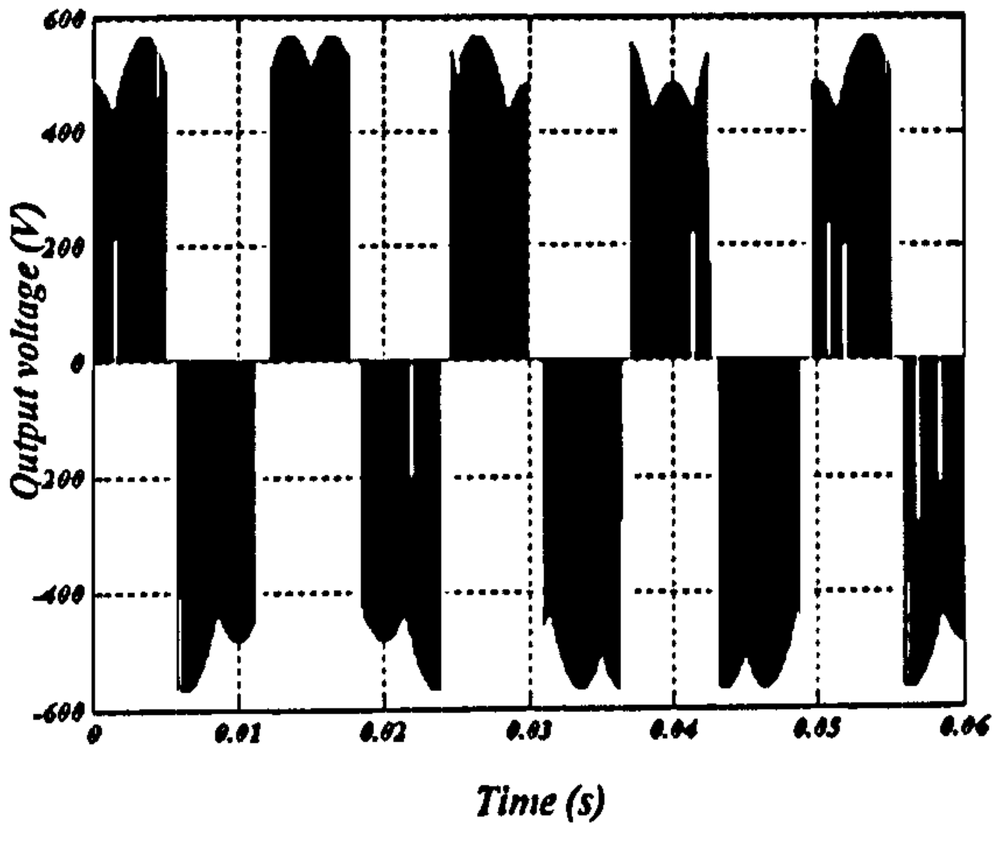




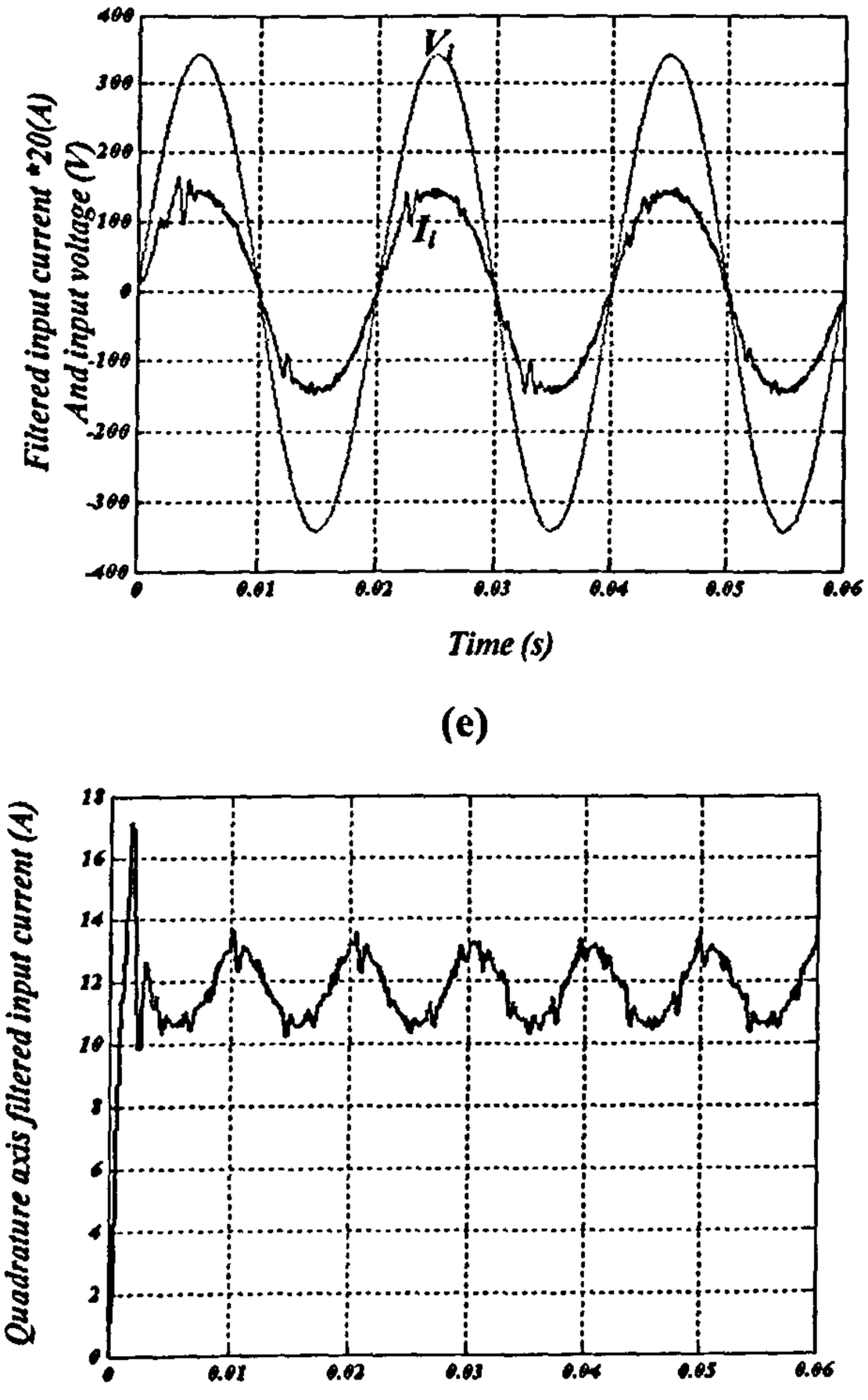
(b)



(c)



(a)



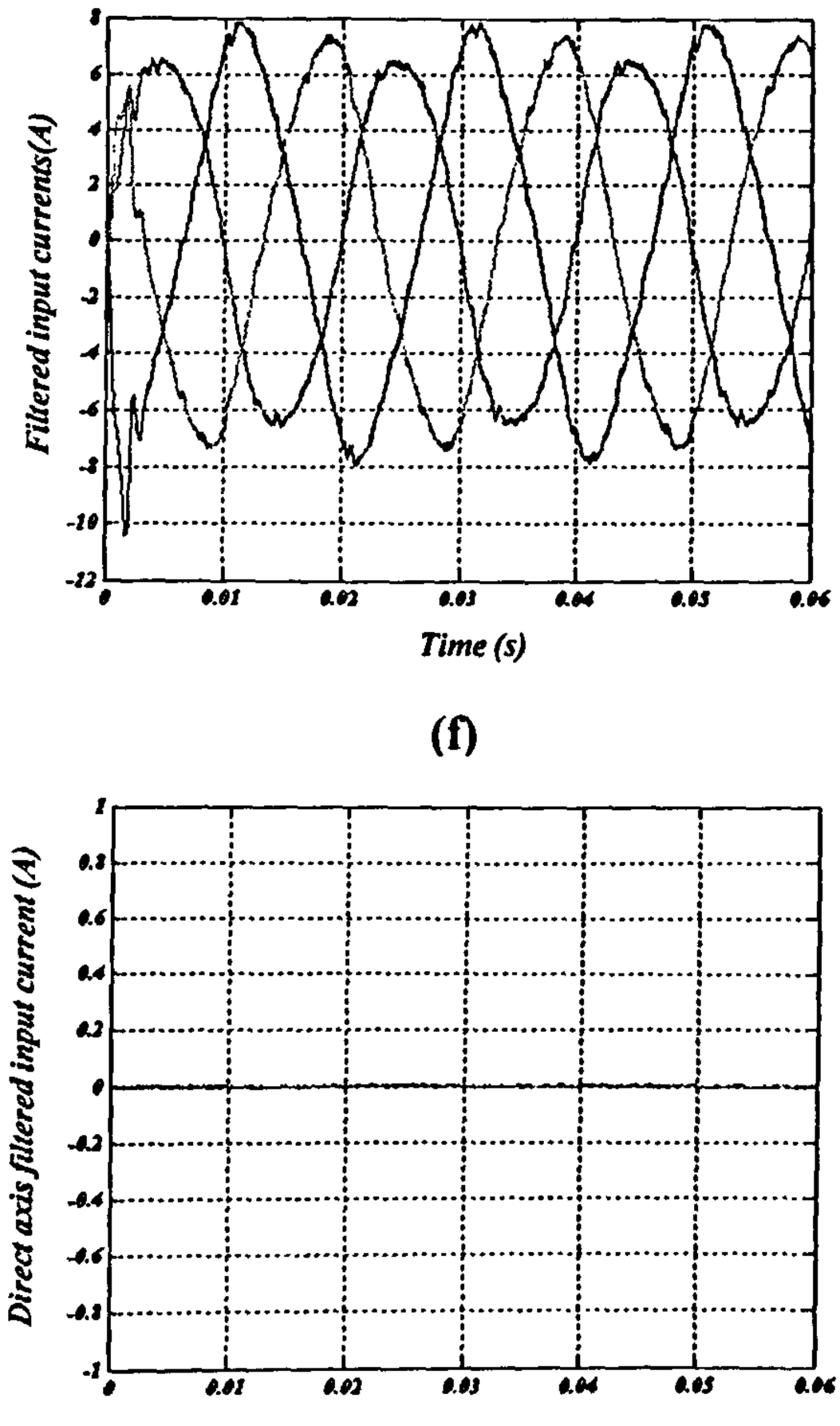


Fig. 4.32 Output response for f_o=80 Hz, unity input power factor at 12 kHz and unbalanced input voltage: (a) output current, (b) quadrature axis current, (c) direct axis current, (d) unfiltered input current *20 and input voltage, and (e) filtered input current*20 and input voltage.

If the reference output current is lowered with a lower modulation index, better output performance is obtained, as shown in Fig. 4.33.

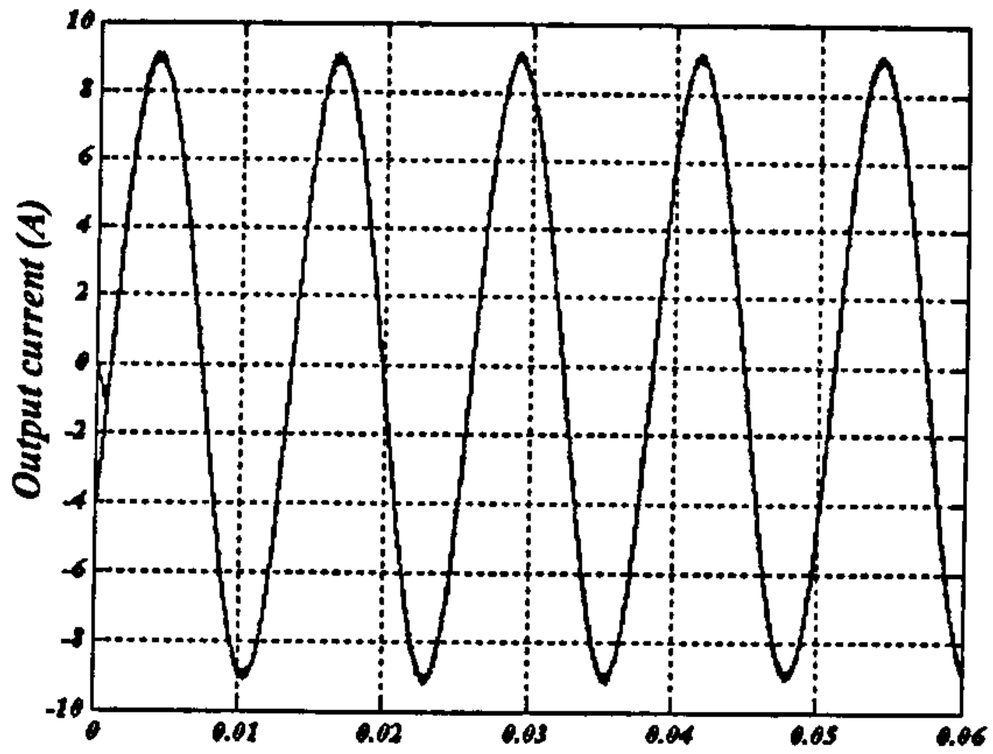


Fig. 4.33 Output current versus time response.

In case of an unbalanced voltage supply, either the input voltage or the input current vectors are the sum of two vectors: positive and negative sequence vectors as seen in Fig. 4.34.

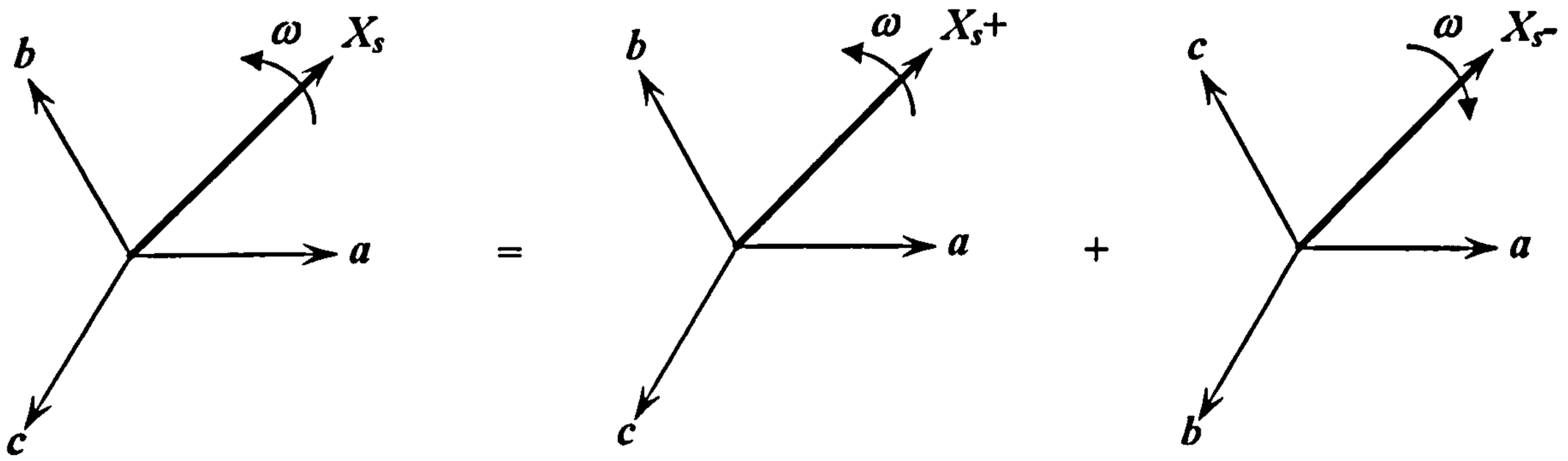


Fig. 4.34 Decomposition of signals into positive and negative sequence components.

The apparent power S in the case of unbalance can be further defined in terms of the grid voltages and currents as a positive and a negative sequence in the *d-q* frame:

$$S_{i/p} = \vec{e_{DQ}} . \vec{i^*_{DQ}} = \left(e_{DQ+}e^{j\omega t} + e_{DQ-}e^{-j\omega t}\right)\left(i_{DQ+}e^{j\omega t} + i_{DQ-}e^{-j\omega t}\right)^* \quad 4.26$$

which becomes:

$$\begin{aligned} S_{i/p} &= \left(e_{DQ+}e^{j\omega t} + e_{DQ-}e^{-j\omega t}\right)\left(i^*_{DQ+}e^{-j\omega t} + i^*_{DQ-}e^{j\omega t}\right) \\ &= e_{DQ+}i^*_{DQ+} + e_{DQ-}i^*_{DQ-} + e_{DQ+}i^*_{DQ-}e^{j2\omega t} + e_{DQ-}i^*_{DQ+}e^{-j2\omega t} \end{aligned} \quad 4.27$$

Rearranging the above equation gives:

$$S_{i/p} = S_+ + S_- + (a+jb)e^{j2\omega t} + (c+jd)e^{-j2\omega t}$$
$$= S_+ + S_- + ((a+c)\cos 2\omega t + (d-b)\sin 2\omega t) \qquad 4.28$$
$$+ j((a+c)\sin 2\omega t - (d-b)\cos 2\omega t)$$

where

$$S_+ = (e_{D+}i_{D+} + e_{Q+}i_{Q+}) + j(e_{Q+}i_{D+} - e_{D+}i_{Q+})$$
$$S_- = (e_{D-}i_{D-} + e_{Q-}i_{Q-}) + j(e_{Q-}i_{D-} - e_{D-}i_{Q-}) \qquad 4.29$$
$$a = e_{D+}i_{D-} + e_{Q+}i_{Q-} \qquad b = e_{D-}i_{Q+} - e_{Q-}i_{D+}$$
$$c = e_{D-}i_{D+} + e_{Q-}i_{Q+} \qquad d = e_{D+}i_{Q-} - e_{Q+}i_{D-}$$

Equation 4.29 can be written in matrix form as:

$$\begin{bmatrix} P \\ Q \\ d-b \\ a+c \end{bmatrix} = \begin{bmatrix} e_{D+} & e_{Q+} & e_{D-} & e_{Q-} \\ e_{Q+} & -e_{D+} & e_{Q-} & -e_{D-} \\ e_{Q-} & -e_{D-} & -e_{Q+} & e_{D+} \\ e_{D-} & e_{Q-} & e_{D+} & e_{Q+} \end{bmatrix} \begin{bmatrix} i_{D+} \\ i_{Q+} \\ i_{D-} \\ i_{Q-} \end{bmatrix} \qquad 4.30$$

Hence the input current can be calculated from:

$$\begin{bmatrix} i_{D+} \\ i_{Q+} \\ i_{D-} \\ i_{Q-} \end{bmatrix} = \begin{bmatrix} e_{D+} & e_{Q+} & e_{D-} & e_{Q-} \\ e_{Q+} & -e_{D+} & e_{Q-} & -e_{D-} \\ e_{Q-} & -e_{D-} & -e_{Q+} & e_{D+} \\ e_{D-} & e_{Q-} & e_{D+} & e_{Q+} \end{bmatrix}^{-1} \cdot \begin{bmatrix} P \\ Q \\ d-b \\ a+c \end{bmatrix} \qquad 4.31$$

Assuming a balanced output load, in steady-state the instantaneous output active power is constant. By neglecting the semiconductors and input filter losses, the instantaneous input active power should equal this dc value, hence the oscillating components (a+c) and (d-b) should both be zero. In equation 4.31, if the oscillating components are equal to zero, the negative sequence input current cannot be eliminated. Based on the available cited references, this is the first time mathematical analysis has proved it is not possible to achieve balanced input and output currents with unbalanced input voltages. Also this contrasts with the results obtained in [2.29]

If a balanced sinusoidal input current is required, the control algorithm should be modified. The average output power should be first estimated. In the case of an R-L load, the steady state active power will be:

$$P_{o/p} = 3I_{ph,rms}^2 R \qquad 4.32$$

For a pure sinusoidal input current, the negative sequence current components should be eliminated, hence equation 4.32 will be:

$$\begin{bmatrix} P \\ Q \\ d-b \\ a+c \end{bmatrix} = \begin{bmatrix} e_{D+} & e_{Q+} & e_{D-} & e_{Q-} \\ e_{Q+} & -e_{D+} & e_{Q-} & -e_{D-} \\ e_{Q-} & -e_{D-} & -e_{Q+} & e_{D+} \\ e_{D-} & e_{Q-} & e_{D+} & e_{Q+} \end{bmatrix} \begin{bmatrix} i_{D+} \\ i_{Q+} \\ 0 \\ 0 \end{bmatrix} \quad 4.33$$

Hence:

$$P_{i/p} = e_{D+}i_{D+} + e_{Q+}i_{Q+} = P_{o/p}$$
$$Q_{i/p} = P_{i/p}.tan\left(cos^{-1}\ pf\right) = e_{Q+}i_{D+} - e_{D+}i_{Q+} \quad 4.34$$

The modified instantaneous active output power (P_{total}) becomes:

$$P_{total}(\omega t) = e_{D+}i_{D+} + e_{Q+}i_{Q+} + \left(e_{D-}i_{D+} + e_{Q-}i_{Q+}\right)cos\,2\omega t + \left(e_{Q-}i_{D+} - e_{D-}i_{Q+}\right)sin\,2\omega t \quad 4.35$$

The output reference current magnitude is modified according to the instantaneous input power. A pure balanced sinusoidal input current is at the expense of distorting the output current since there is no energy storage element within the matrix converter. This will not be the case if normal back to back converters are used with an interconnecting capacitor dc link. The latter configuration can readily control both the output and input power simultaneously.

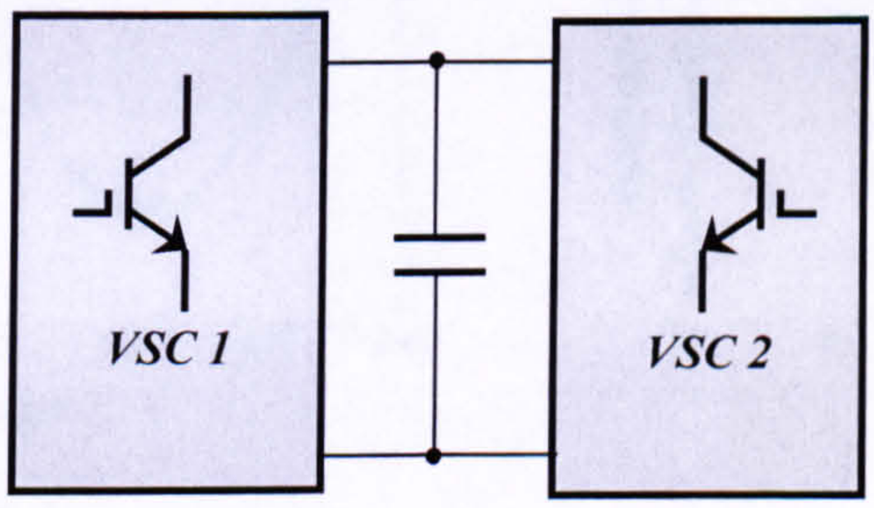


Fig. 4.35 Conventional back to back converters

4.6.3 Flat topped voltage supply

A supply which contains 1.5% and 1% fifth and seventh order harmonics respectively as shown in Fig. 4.36 is connected to the input terminals.

There is no problem with the quality of the output current, but the input current is distorted as shown in Figures 4.37 and 4.38. This is the same power balance problem as with an unbalanced power supply.

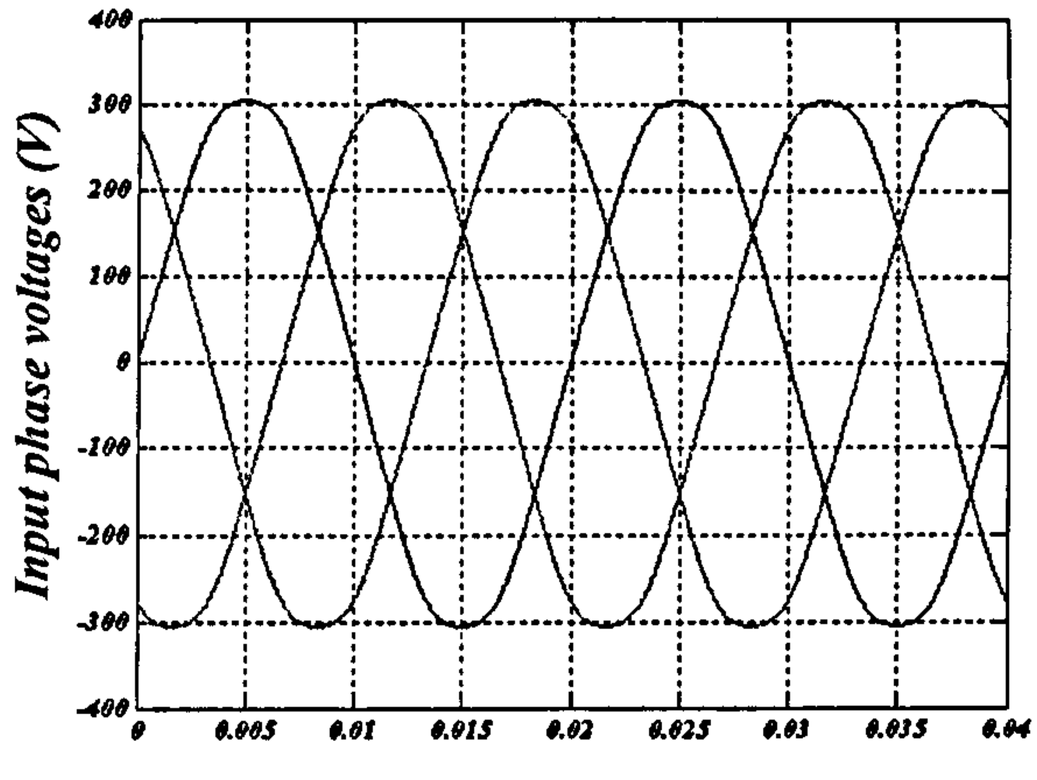
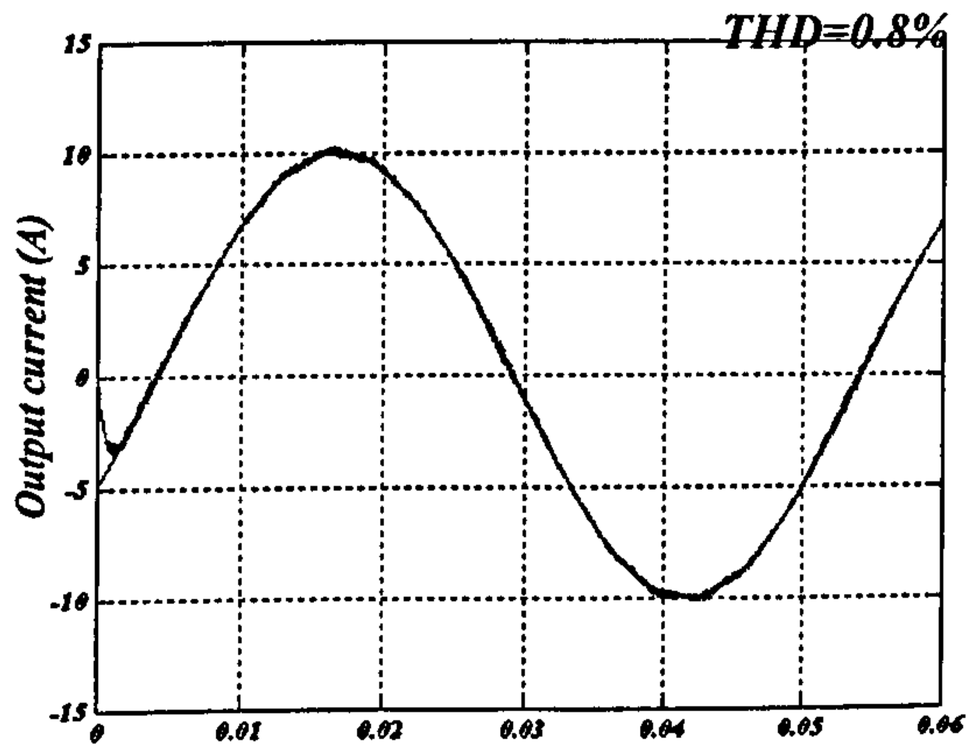
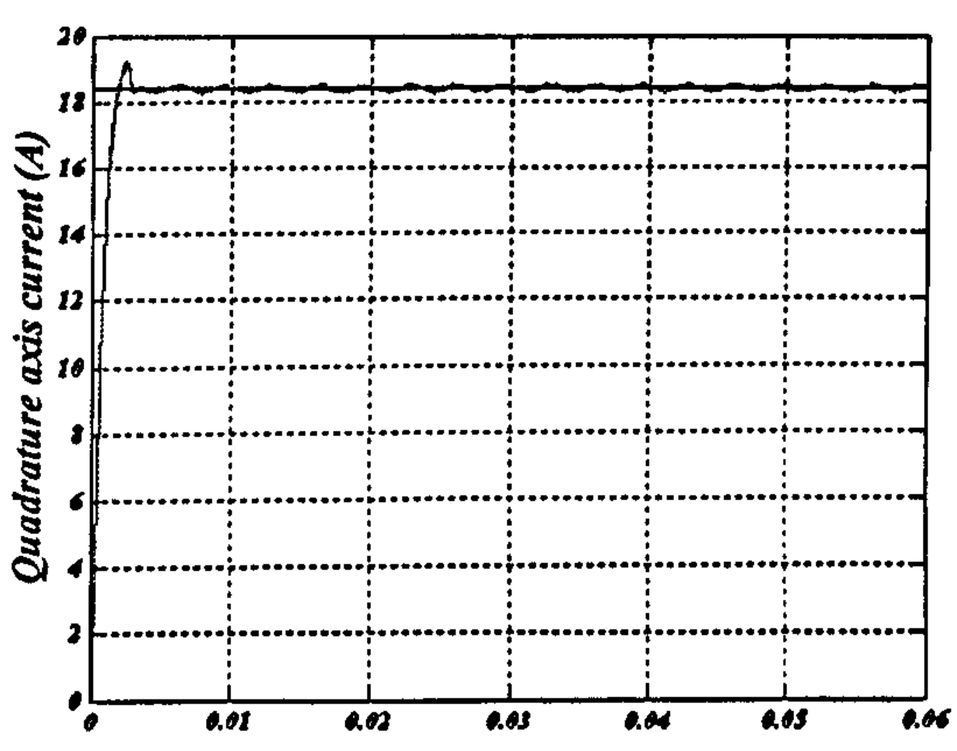


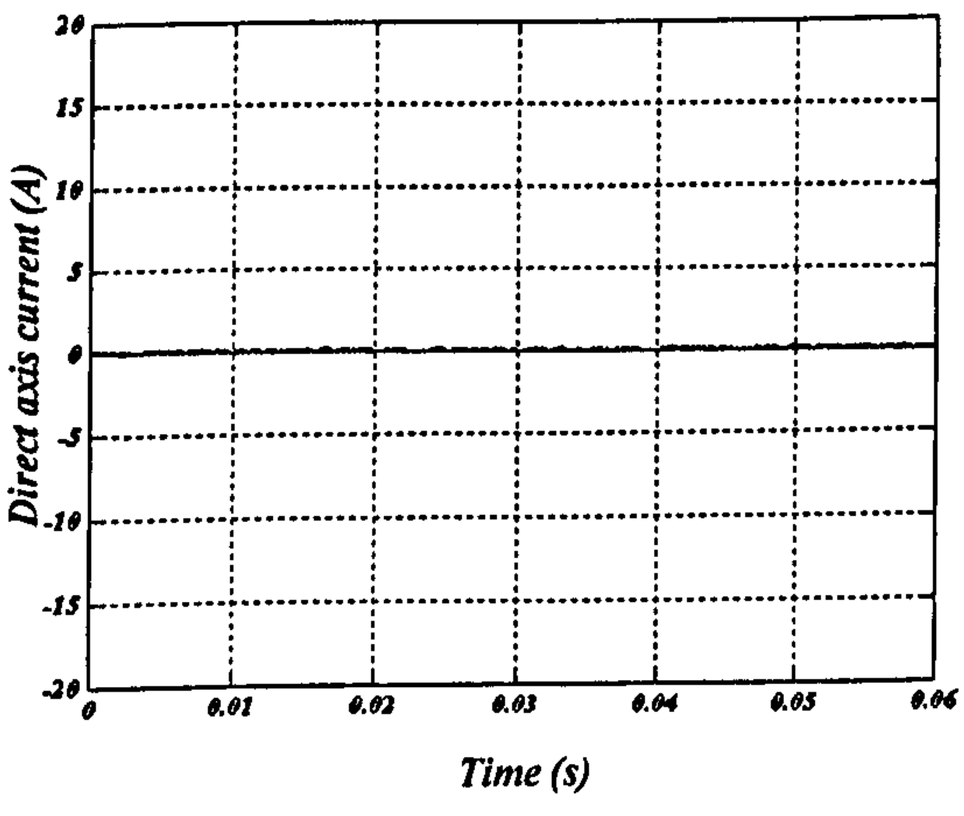
Fig. 4.36 Flat topped input voltage



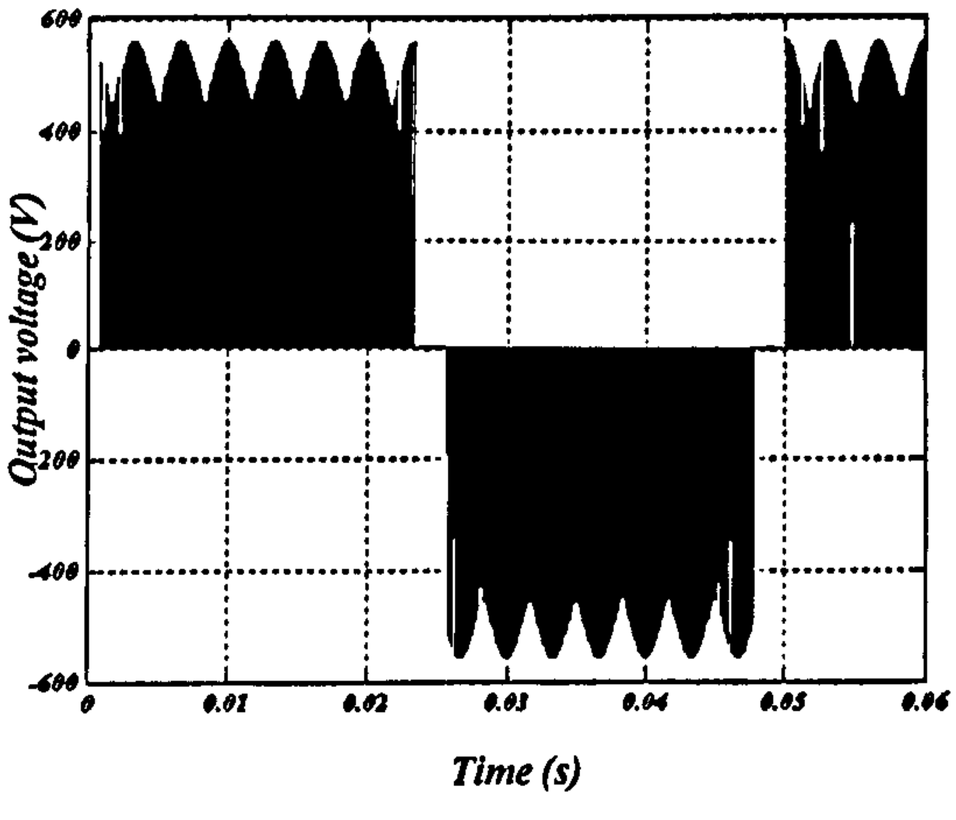
(a)



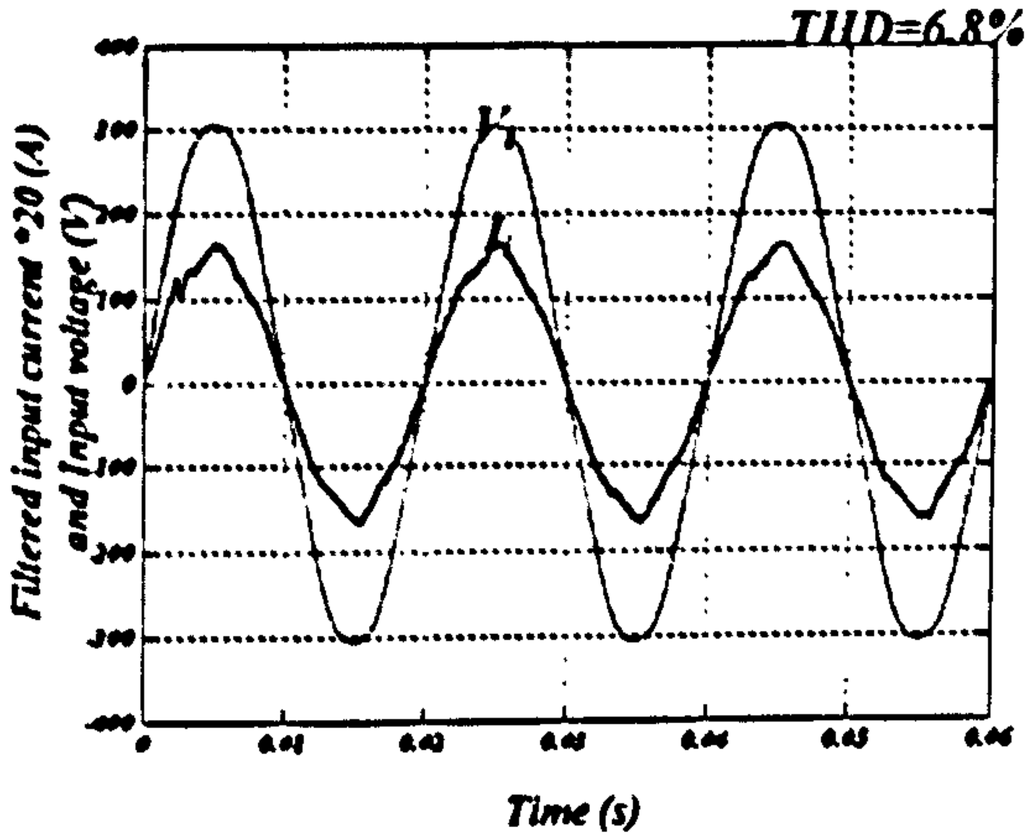
(b)



(c)

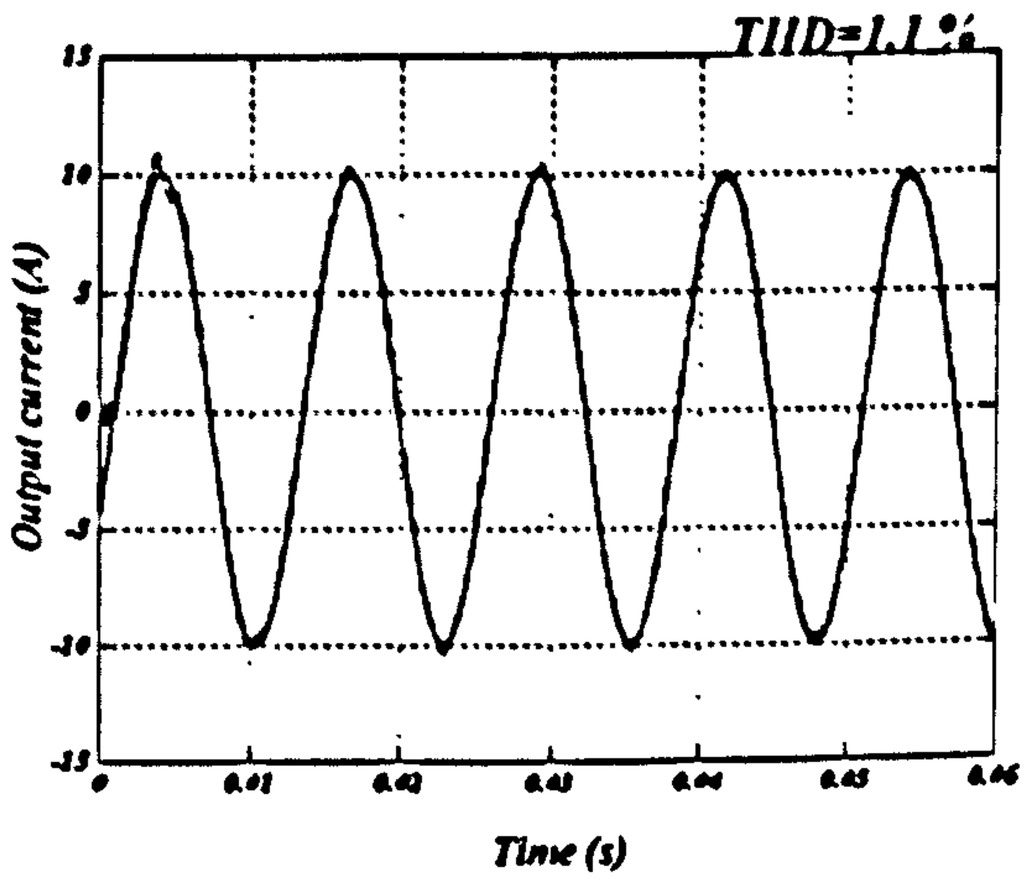


(d)

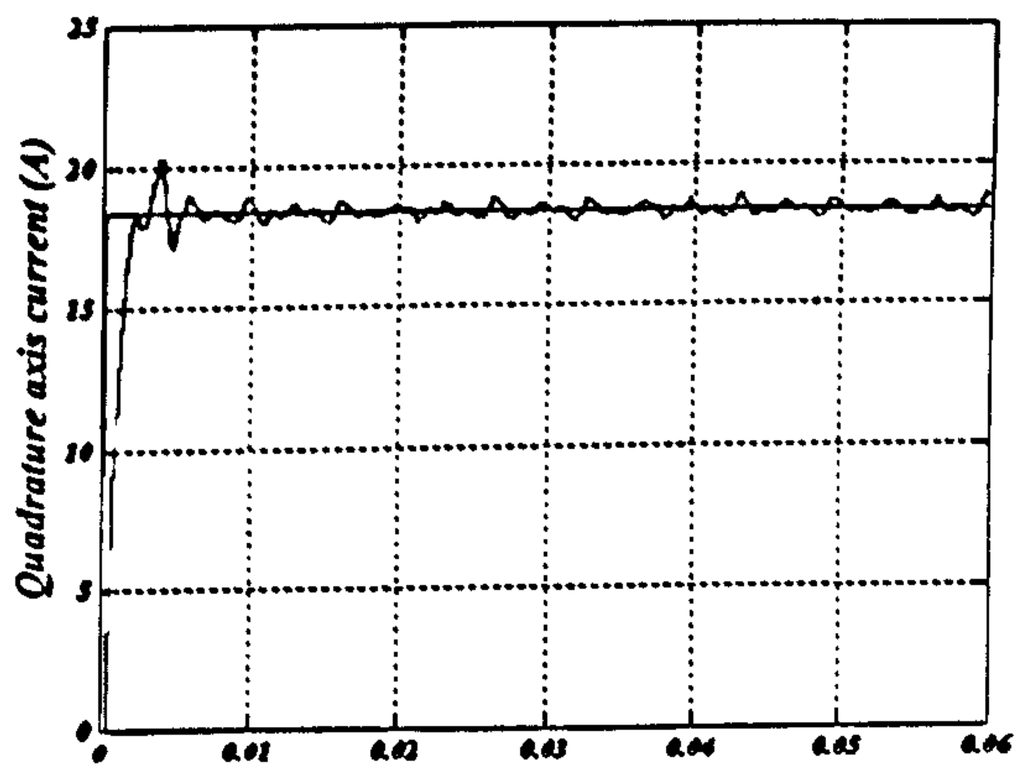


(e)

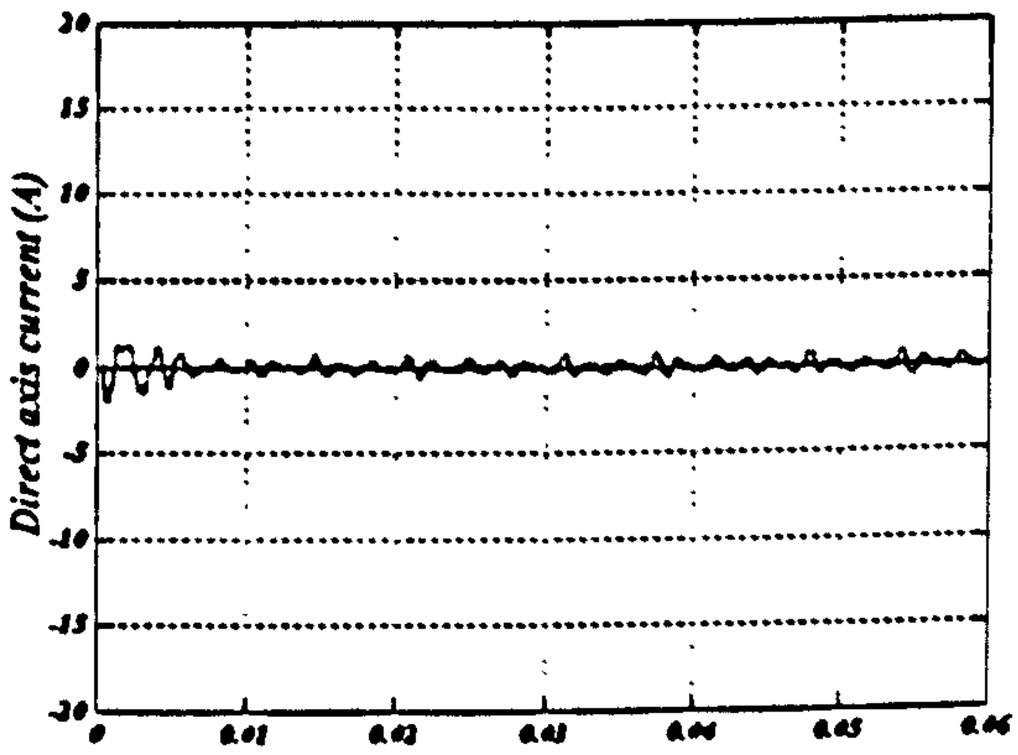
Fig. 4.37 Output response for f_o=20 Hz, unity input power factor at 12 kHz and flat topped input voltage (a) output current, (b) quadrature axis current, (c) direct axis current, (d) output voltage, and (e) filtered input current*20 and input voltage.



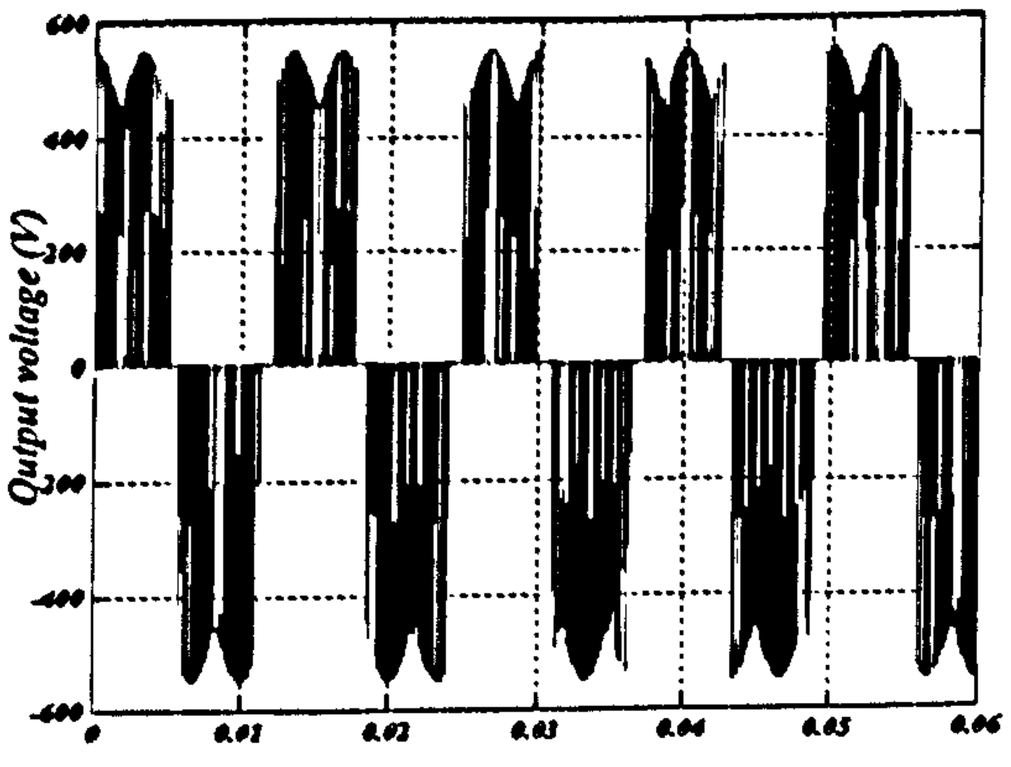
(a)



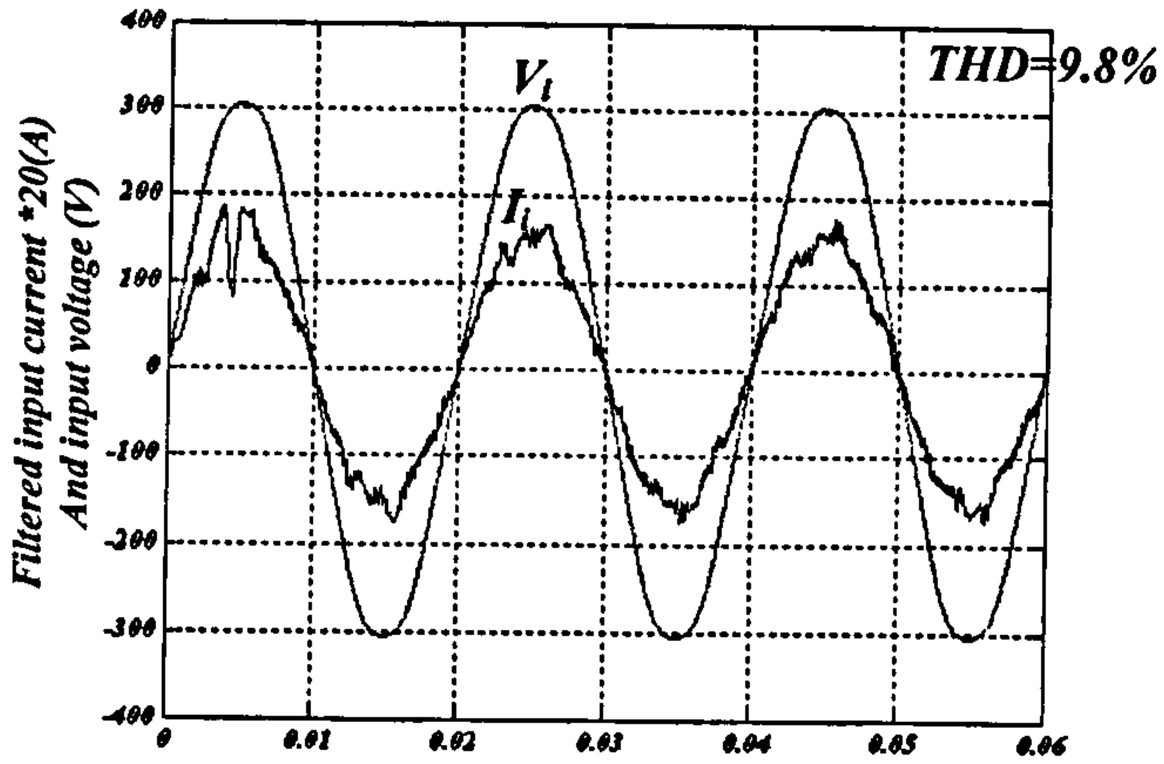
(b)



(c)



(d)



(e)

Fig. 4.38 Output response for f_o=80 Hz, unity input power factor at 12 kHz and flat topped input voltage: (a) output current, (b) quadrature axis current, (c) direct axis current, (d) output voltage, and (e) filtered input current*20 and input voltage.

In case of a flat topped voltage supply, either the input voltage or the input current vectors can be viewed as the sum of three vectors as in Fig. 4.42.

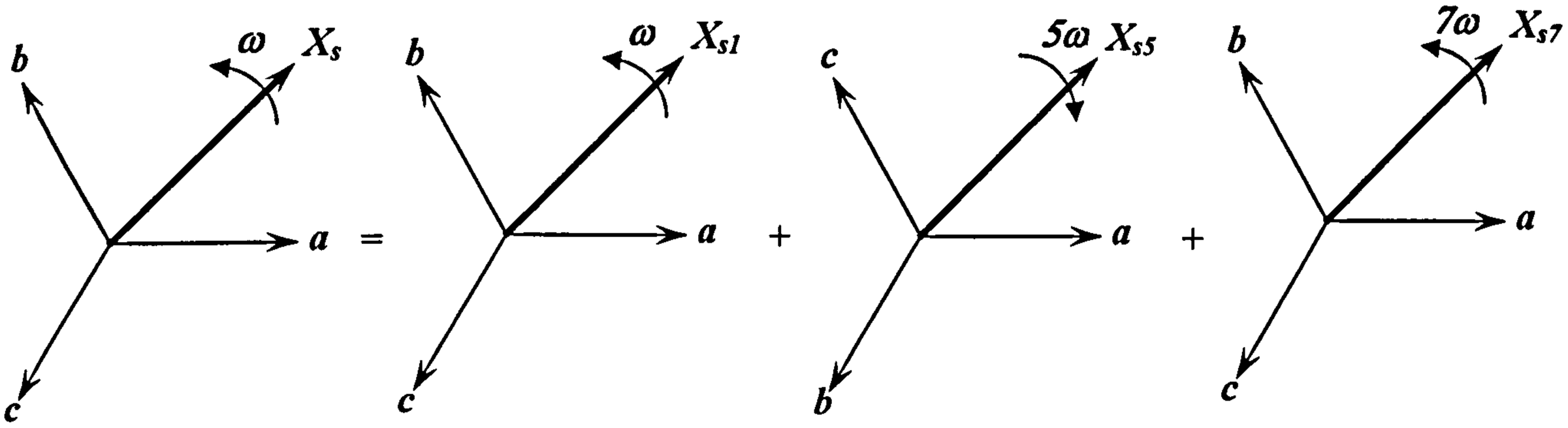


Fig. 4.39 Decomposition of the flat topped source into fundamental, fifth, and seventh harmonic components.

The apparent power S can be defined similarly to the unbalanced case. The grid voltages and currents in terms of positive and negative sequences in the d-q frame give:

$$\begin{aligned} S_{o/p} &= \vec{e}_{DQ} \cdot \vec{i}^{*}_{DQ} \\ &= \left(e_{DQ1}e^{j\omega t} + e_{DQ5}e^{-j5\omega t} + e_{DQ7}e^{j7\omega t}\right)\left(i_{DQ1}e^{j\omega t} + i_{DQ5}e^{-j5\omega t} + i_{DQ7}e^{j7\omega t}\right)^{*} \end{aligned} \quad 4.36$$

This can be written as:

$$\begin{aligned} S_{o/p} &= \vec{e}_{DQ} \cdot \vec{i}^{*}_{DQ} \\ S_{o/p} &= \left(e_{DQ1}e^{j\omega t} + e_{DQ5}e^{-j5\omega t} + e_{DQ7}e^{j7\omega t}\right)\left(i^{*}_{DQ1}e^{-j\omega t} + i^{*}_{DQ5}e^{j5\omega t} + i^{*}_{DQ7}e^{-j7\omega t}\right) \\ &= \left(e_{DQ1}i^{*}_{DQ1} + e_{DQ5}i^{*}_{DQ5} + e_{DQ7}i^{*}_{DQ7}\right) + \left(e_{DQ1}i^{*}_{DQ5} + e_{DQ7}i^{*}_{DQ1}\right)e^{j6\omega t} \\ &\quad + \left(e_{DQ1}i^{*}_{DQ7} + e_{DQ5}i^{*}_{DQ1}\right)e^{-j6\omega t} + \left(e_{DQ5}i^{*}_{DQ7}\right)e^{-j12\omega t} + \left(e_{DQ7}i^{*}_{DQ5}\right)e^{j12\omega t} \end{aligned} \quad 4.37$$

Rearranging the above equation gives:

$$
\begin{aligned}
S_{o/p} &= S_1 + S_5 + S_7 + (a+jb)e^{j6\omega t} + (c+jd)e^{-j6\omega t} + (e+jf)e^{j12\omega t} + (g+jh)e^{-j1} \\
&= S_1 + S_5 + S_7 + ((a+c)\cos 6\omega t + (d-b)\sin 6\omega t) \\
&\qquad + j((a+c)\sin 6\omega t - (d-b)\cos 6\omega t) \\
&\qquad + ((e+g)\cos 6\omega t + (h-f)\sin 12\omega t) \\
&\qquad + j((e+g)\sin 6\omega t - (h-f)\cos 12\omega t)
\end{aligned} \tag{4.38}
$$

where

$$
\begin{aligned}
S_1 &= (e_{D1}i_{D1} + e_{q1}i_{q1}) + j(e_{Q1}i_{D1} - e_{D1}i_{Q1}) \\
S_5 &= (e_{D5}i_{D5} + e_{Q5}i_{Q5}) + j(e_{Q5}i_{D5} - e_{D5}i_{Q5}) \\
S_7 &= (e_{D7}i_{D7} + e_{Q7}i_{Q7}) + j(e_{Q7}i_{D7} - e_{D7}i_{Q7})
\end{aligned}
$$

$$
\begin{aligned}
a &= e_{D1}i_{D5} + e_{Q1}i_{Q5} + e_{D7}i_{D1} + e_{Q7}i_{Q1} & b &= e_{D1}i_{Q5} - e_{Q1}i_{D5} + e_{D7}i_{Q1} - e_{Q7}i_{D1} \\
c &= e_{D1}i_{D7} + e_{Q1}i_{Q7} + e_{D5}i_{D1} + e_{Q5}i_{Q1} & d &= e_{D1}i_{Q7} - e_{Q1}i_{D7} + e_{D5}i_{Q1} - e_{Q5}i_{D1} \\
e &= e_{D5}i_{D7} + e_{Q5}i_{Q7} & f &= e_{D5}i_{Q7} - e_{Q5}i_{D7} \\
g &= e_{D7}i_{D5} + e_{Q7}i_{Q5} & h &= e_{D7}i_{Q5} - e_{Q7}i_{D5}
\end{aligned} \tag{4.39}
$$

Equation 4.39 can be written in matrix form as:

$$
\begin{bmatrix} P \\ Q \\ a+c \\ d-b \\ e+g \\ f-h \end{bmatrix} = \begin{bmatrix} e_{D1} & e_{Q1} & e_{D5} & e_{Q5} & e_{D7} & e_{Q7} \\ e_{Q1} & -e_{D1} & e_{Q5} & -e_{D5} & e_{Q7} & -e_{D7} \\ e_{D5}+e_{D7} & e_{Q5}+e_{Q7} & e_{D1} & e_{Q1} & e_{D1} & e_{Q1} \\ e_{Q7}-e_{Q5} & e_{D5}-e_{D7} & e_{Q1} & -e_{D1} & -e_{Q1} & e_{D1} \\ 0 & 0 & e_{D7} & e_{Q7} & e_{D5} & e_{Q5} \\ 0 & 0 & e_{Q7} & -e_{D7} & -e_{Q5} & e_{D5} \end{bmatrix} \begin{bmatrix} i_{D1} \\ i_{Q1} \\ i_{D5} \\ i_{Q5} \\ i_{D7} \\ i_{Q7} \end{bmatrix} \tag{4.40}
$$

Hence the input current can be calculated from the formula:

$$
\begin{bmatrix} i_{D1} \\ i_{Q1} \\ i_{D5} \\ i_{Q5} \\ i_{D7} \\ i_{Q7} \end{bmatrix} = \begin{bmatrix} e_{D1} & e_{Q1} & e_{D5} & e_{Q5} & e_{D7} & e_{Q7} \\ e_{Q1} & -e_{D1} & e_{Q5} & -e_{D5} & e_{Q7} & -e_{D7} \\ e_{D5}+e_{D7} & e_{Q5}+e_{Q7} & e_{D1} & e_{Q1} & e_{D1} & e_{Q1} \\ e_{Q7}-e_{Q5} & e_{D5}-e_{D7} & e_{Q1} & -e_{D1} & -e_{Q1} & e_{D1} \\ 0 & 0 & e_{D7} & e_{Q7} & e_{D5} & e_{Q5} \\ 0 & 0 & e_{Q7} & -e_{D7} & -e_{Q5} & e_{D5} \end{bmatrix}^{-1} \begin{bmatrix} P \\ Q \\ a+c \\ d-b \\ e+g \\ f-h \end{bmatrix} \tag{4.41}
$$

Assuming a balanced output load, at steady-state the instantaneous output active power is a constant dc value. The instantaneous input active power should equal this dc value, hence the oscillating components $(a+c)$, $(b-d)$, $(e+g)$ and $(f-h)$ should all be zero. By referring to equation 4.41, the fifth and seventh harmonic components in the input current cannot be eliminated. Again, the advantage of back to back converters being able to exploit the dc link energy is applicable.

References:

[4.1] B. Pal, B. Chaudhuri, **'Robust Control in Power Systems'**, Springer, 2005, **ISBN-10:** 038725949X.

[4.2] H. J. Sira-Ramirez, R.-Ortigoza, **'Control Design Techniques in Power Electronics Devices'**, Springer, 2006, **ISBN-10:** 1846284589.

[4.3] D. A. Paice, **'Power Electronics Converter Harmonics: Multipulse Methods for Clean Power'**, Wiley-IEEE Press, 1999, **ISBN-10:** 0780353943.

[4.4] E. Acha, *et al.* **'Power Electronic Control in Electrical Systems'**, Newnes, 2002, **ISBN-10:** 0750651261.

[4.5] M. H. Rashid, **'Power Electronics Handbook'**, Academic Press, 2001, **ISBN-10:** 0125816502.

[4.6] M. H. Rashid, **'SPICE for Power Electronics and Electric Power, Second Edition'**, CRC Press 2005, **ISBN-10:** 0849334187.

[4.7] J.A. Rossiter, **'Model-Based Predictive Control: A Practical Approach'**, 2003, **ISBN-10:** 0849312914.

[4.8] C. A. Smith; A. B. Corripio, **'Principles and Practice of Automatic Process Control'**, John Wiley & Sons, 1997, 2nd Edition, **ISBN-10:** 0471575887.

[4.9] W. Perruquetti, **'Sliding Mode Control in Engineering'**, Marcel Dekker, 2002, **ISBN-10:** 0824706714.

[4.10] J. Arrillaga, N. R. Watson, **'Power System Harmonics'**, John Wiley & Sons, 2003, **ISBN-10:** 0470851295.

[4.11] J. O. Attia, **'Electronics and Circuit Analysis using MATLAB'**, CRC Press, 1999, **ISBN-10:** 0849311764.

CHAPTER FIVE

GRID-CONNECTED MATRIX CONVERTER WITH AN L-FILTER INTERFACE

In this chapter the variable-speed wind turbine system shown in Fig. 5.1 is considered. A PI current controller is developed for output current control and its transient performance is evaluated. Some adjustments are proposed to attain acceptable controller performance. For the rectifier stage, a space vector algorithm is used to control the generator end displacement angle.

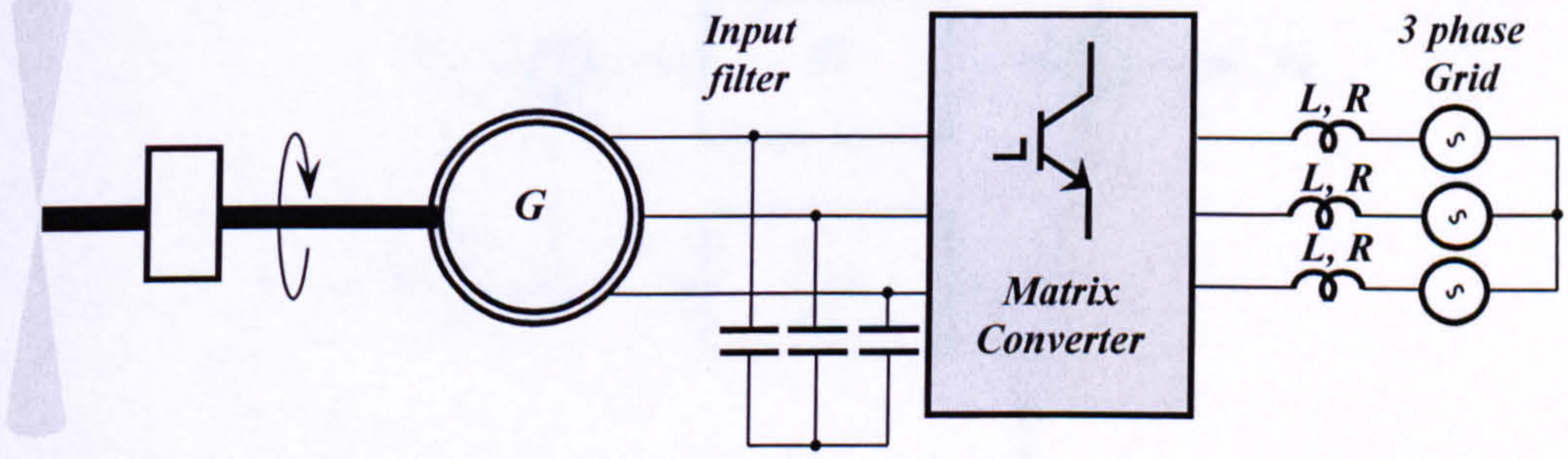


Fig 5.1 Matrix converter system connected to a three-phase grid

5.1 The inverter stage

A virtual VSC is assumed to be connected to the grid through an L-filter interface as shown in figure 5.2. A PI controller is used to control the inverter output voltage in order to provide the desired output currents [5.1]-[5.4]. The reference currents are varied in-line with the input power to the wind turbine.

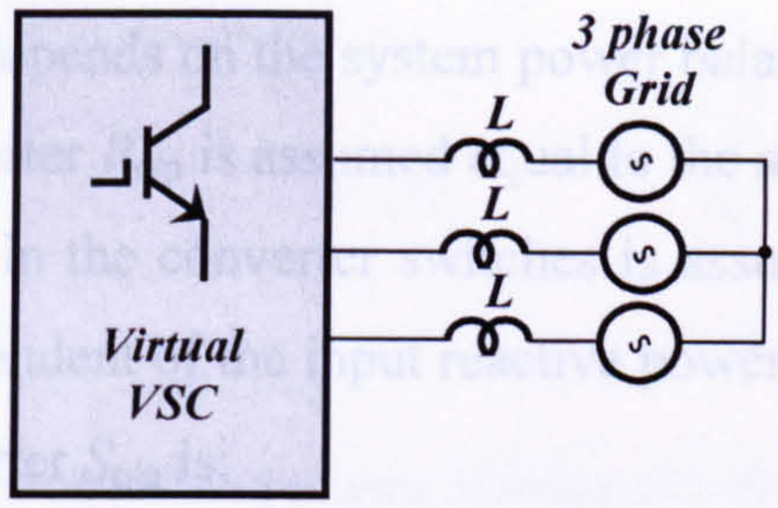


Fig 5.2 Grid-connected with an L-Filter interface inverter stage.

5.1.1 The current controller

The current controller explained in chapter 4 equation 4.14 is utilized to produce the inverter output signals $v_{d,k+1}$ and $v_{q,k+1}$.

$$v_{d,k+1} = e_{d,k} + k_i \sum_{n=0}^{k}\left(i^*_{d,n} - i_{d,n}\right) + k_p.\left(i^*_{d,k} - i_{d,k}\right) - \omega L \frac{i^*_{q,k} + i_{q,k}}{2}$$

$$v_{q,k+1} = e_{q,k} + k_i.\sum_{n=0}^{k}\left(i^*_{q,n} - i_{q,n}\right) + k_p.\left(i^*_{q,k} - i_{q,k}\right) + \omega L \frac{i^*_{d,k} + i_{d,k}}{2} \qquad 5.1$$

The current controller block diagram is shown in fig. 5.3.

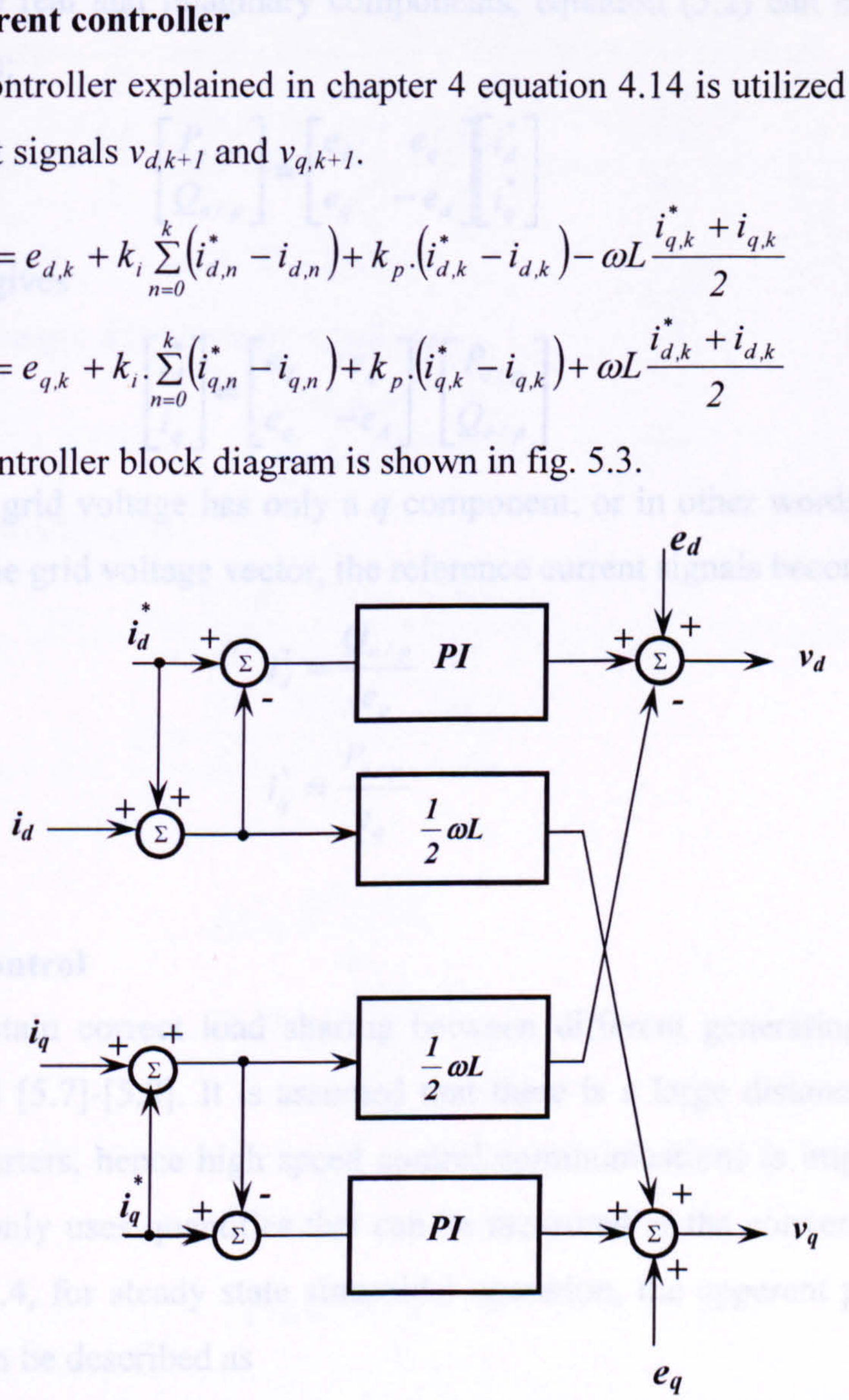


Fig 5.3 Current controller block diagram

5.1.2 Generation of the reference currents

The control algorithm depends on the system power balance. The active power at the output side of the converter $P_{o/p}$ is assumed equal to the active power at the input side $P_{i/p}$, that is zero losses in the converter switches is assumed. The reactive power at the output $Q_{o/p}$ is independent of the input reactive power $Q_{i/p}$. The apparent power at the ac side of the converter $S_{o/p}$ is:

$$S_{o/p} = \vec{e}_{dq} . \vec{i}^{*}_{dq} = \left(e_d i_d + e_q i_q\right) + j\left(e_q i_d - e_d i_q\right) \tag{5.2}$$

Separating the real and imaginary components, equation (5.2) can be expressed in matrix form as:

$$\begin{bmatrix} P_{o/p} \\ Q_{o/p} \end{bmatrix} = \begin{bmatrix} e_d & e_q \\ e_q & -e_d \end{bmatrix} \begin{bmatrix} i^{*}_d \\ i^{*}_q \end{bmatrix} \tag{5.3}$$

Re-arranging gives

$$\begin{bmatrix} i^{*}_d \\ i^{*}_q \end{bmatrix} = \begin{bmatrix} e_d & e_q \\ e_q & -e_d \end{bmatrix}^{-1} \begin{bmatrix} P_{o/p} \\ Q_{o/p} \end{bmatrix} \tag{5.4}$$

Assuming the grid voltage has only a q component, or in other words, the q axis is aligned with the grid voltage vector, the reference current signals become:

$$\begin{aligned} i^{*}_d &= \frac{Q_{o/p}}{e_q} \\ i^{*}_q &= \frac{P_{o/p}}{e_q} \end{aligned} \tag{5.5}$$

5.1.3 Droop control

In order to obtain correct load sharing between different generating units, droop control is used [5.7]-[5.8]. It is assumed that there is a large distance between the different converters, hence high speed control communications is impractical. Thus the converter only uses quantities that can be measured at the converter connection point. In fig. 5.4, for steady state sinusoidal operation, the apparent power flowing into the line can be described as

$$S = \overline{U}.\overline{I^{*}} = U\left(\frac{U - V e^{-j\delta}}{Z e^{j\theta}}\right)^{*} = U\left(\frac{U - V e^{+j\delta}}{Z e^{-j\theta}}\right) = \frac{U^2}{Z} e^{j\theta} - \frac{UV}{Z} e^{j(\theta+\delta)} \tag{5.6}$$

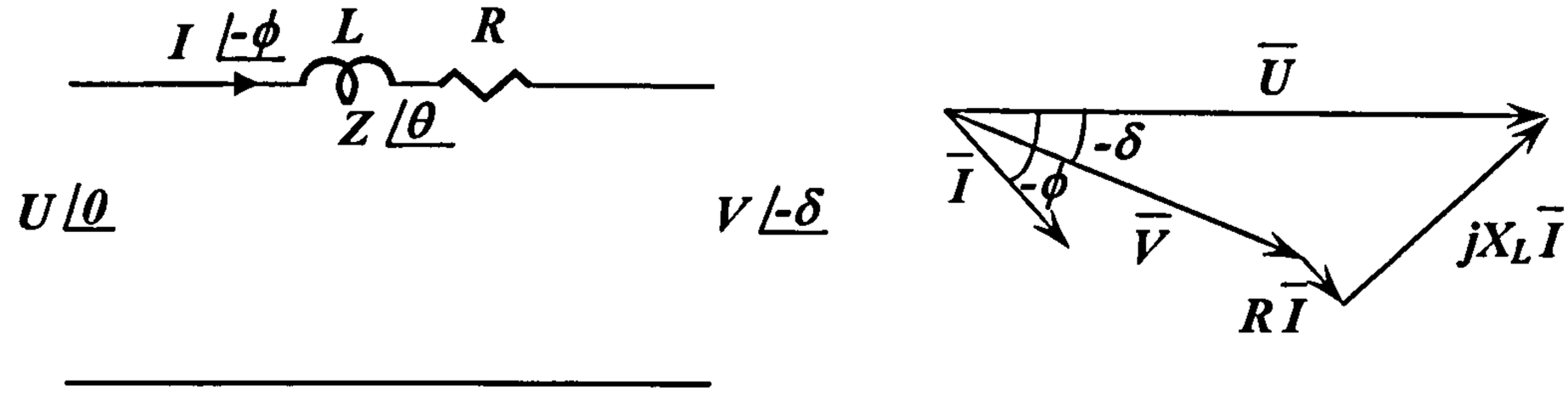


Fig 5.4 Equivalent circuit and phasor diagram of a grid connected generator.

Thus the active and reactive powers become

$$P = \frac{U^2}{Z}\cos\theta - \frac{UV}{Z}\cos(\theta+\delta) = \frac{U^2R}{Z^2} - \frac{UVR}{Z^2}\cos\delta + \frac{UVX}{Z^2}\sin\delta$$
$$Q = \frac{U^2}{Z}\sin\theta - \frac{UV}{Z}\sin(\theta+\delta) = \frac{U^2}{Z^2} - \frac{UVR}{Z^2}\sin\delta - \frac{UVX}{Z^2}\cos\delta \qquad 5.7$$

where $R = Z\cos\theta$, and $X = Z\sin\theta$

Since the load angle δ is normally small

$$\sin\delta \approx \delta$$
$$\cos\delta \approx 1 \qquad 5.8$$

Equation 5.7 then becomes

$$P = \frac{U^2R}{Z^2} - \frac{UVR}{Z^2} + \frac{UVX}{Z^2}\delta$$
$$Q = \frac{U^2}{Z^2} - \frac{UVR}{Z^2}\delta - \frac{UVX}{Z^2} \qquad 5.9$$

which leads to

$$V\delta \approx \frac{XP - RQ}{U}$$
$$U - V \approx \frac{XQ + RP}{U} \qquad 5.10$$

Since $X>>R$

$$\delta \approx \frac{XP}{UV}$$
$$U - V \approx \frac{XQ}{U} \qquad 5.11$$

Since the power angle depends on the frequency, the active power can be controlled by controlling the frequency. On the other hand, the reactive power depends mainly on the voltage difference, *U-V*, as shown in fig. 5.5.

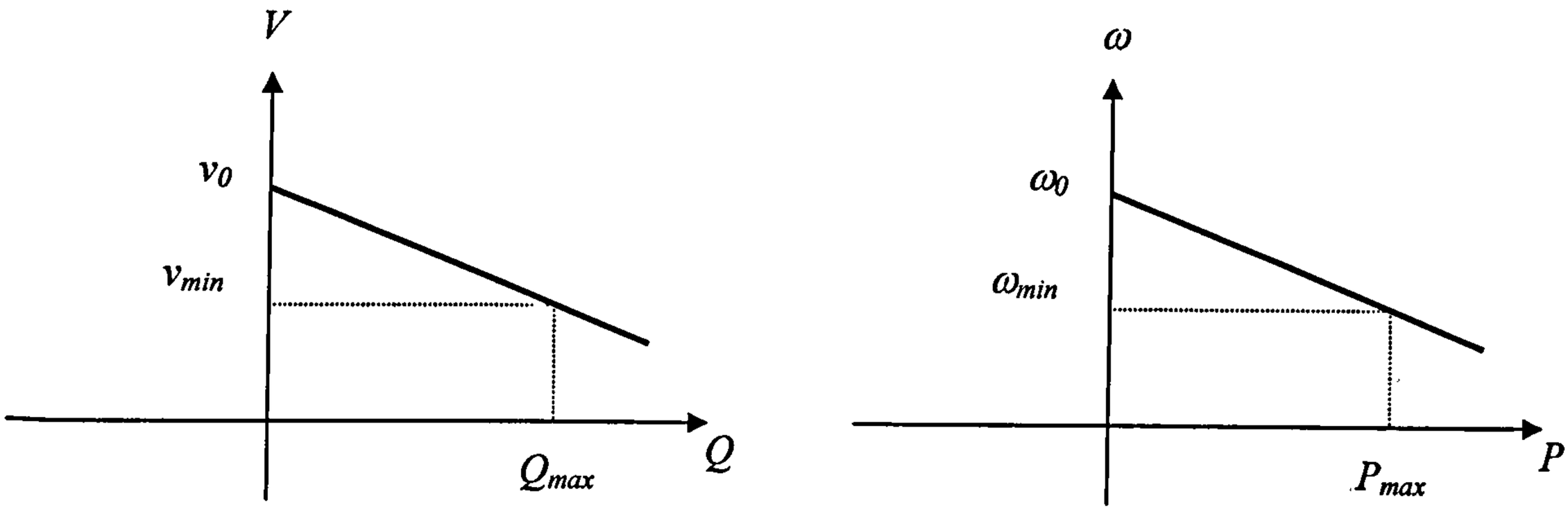


Fig 5.5 Active and reactive power droop characteristics.

$$P_{o/p} = C_P(\omega_o - \omega)$$
$$Q_{o/p} = C_Q(v_o - v) \qquad 5.12$$

For load sharing purposes, these droop functions are used in order to make the system appear as a synchronous machine. The active power is dependant on the wind power input which can be controlled mechanically within certain limits by adjusting the angle of the blades. The reactive power is independent of the input values.

5.1.4 Frequency and phase angle estimator

The estimators used in chapter 4 are used, namely [5.9]

$$\omega_{k+1} \approx \omega_k - A\frac{e_{d,k}}{e_{q,k}}T_s$$
$$\theta_{k+1} \approx \theta_k + \omega_k T_s - B\frac{e_{d,k}}{e_{q,k}}T_s \qquad 5.13$$

To reduce the effect of noise on the estimated values, the direct axis voltage is replaced by

$$e_{d,k} \approx v_{d,k} - Ri_{d,k} + \omega Li_{q,k} \qquad 5.14$$

The inverter stage block diagram is shown in fig. 5.6.

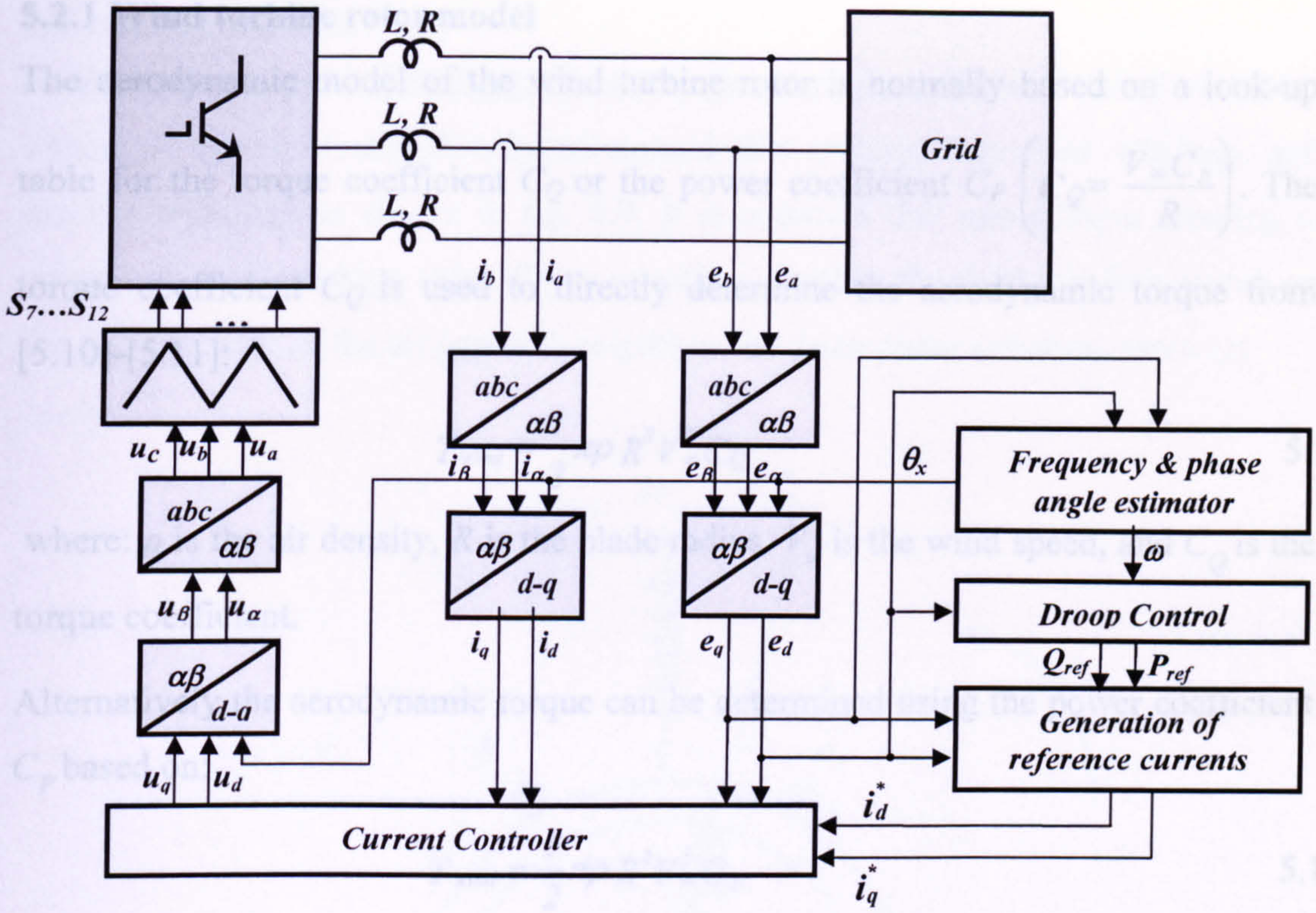


Fig 5.6 Block diagram of a grid-connected inverter with an L-filter interface.

5.2 The active rectifier stage

The rectifier stage shown in fig. 5.7 is assumed to have a three-phase generator connected to a wind turbine. The input displacement angle can be controlled by measuring the generator position angle θ. Modelling of the wind turbine, the synchronous generator, and the input filter are briefly introduced.

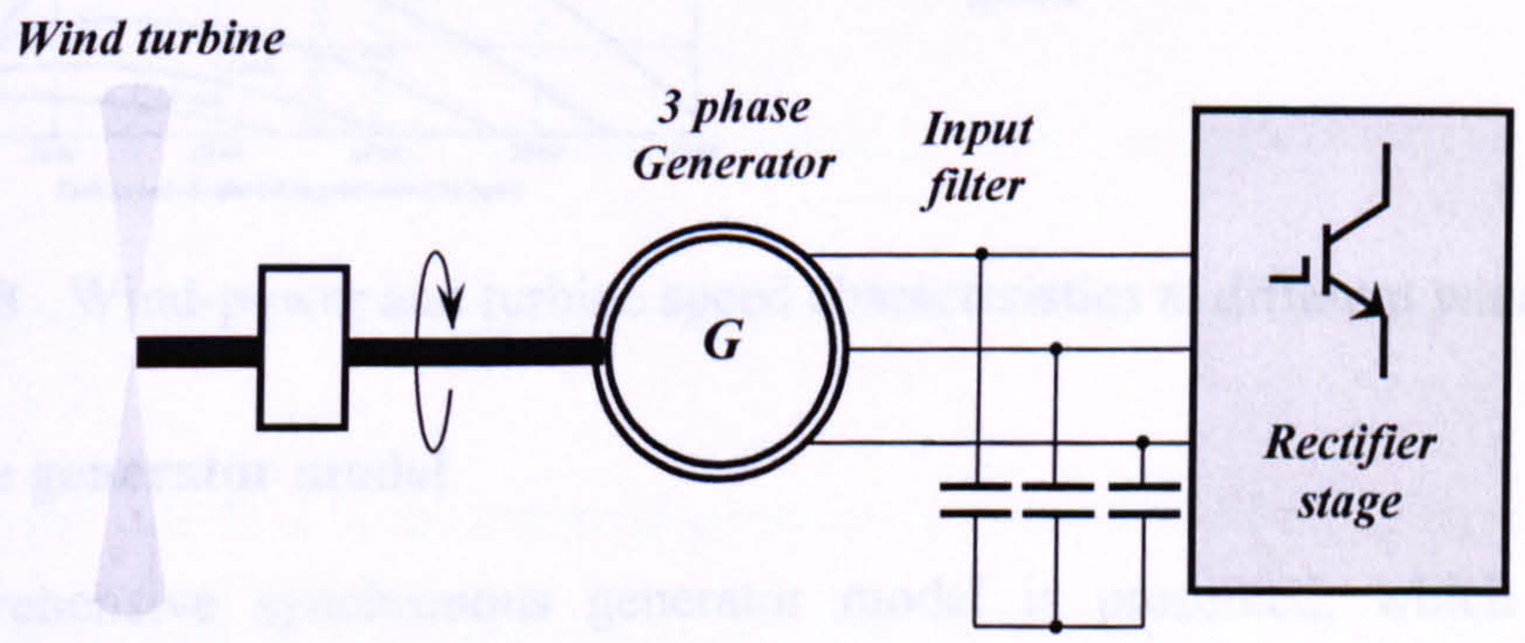


Fig 5.7 Virtual rectifier stage of L filter Grid-connected matrix converter

5.2.1 Wind turbine rotor model

The aerodynamic model of the wind turbine rotor is normally based on a look-up table for the torque coefficient C_Q or the power coefficient C_P $\left(C_Q = \frac{V_\infty C_P}{R}\right)$. The torque coefficient C_Q is used to directly determine the aerodynamic torque from [5.10]-[5.11]:

$$T_{wind} = \frac{1}{2}\pi\rho R^3 V_\infty^2 C_Q \qquad 5.15$$

where: ρ is the air density, R is the blade radius, V_∞ is the wind speed, and C_Q is the torque coefficient.

Alternatively the aerodynamic torque can be determined using the power coefficient C_P based on:

$$T_{wind} = \frac{1}{2}\pi\rho R^2 V_\infty^3 C_p \qquad 5.16$$

Alternatively, fig. 5.8 is used when modelling the wind turbine as a look-up table.

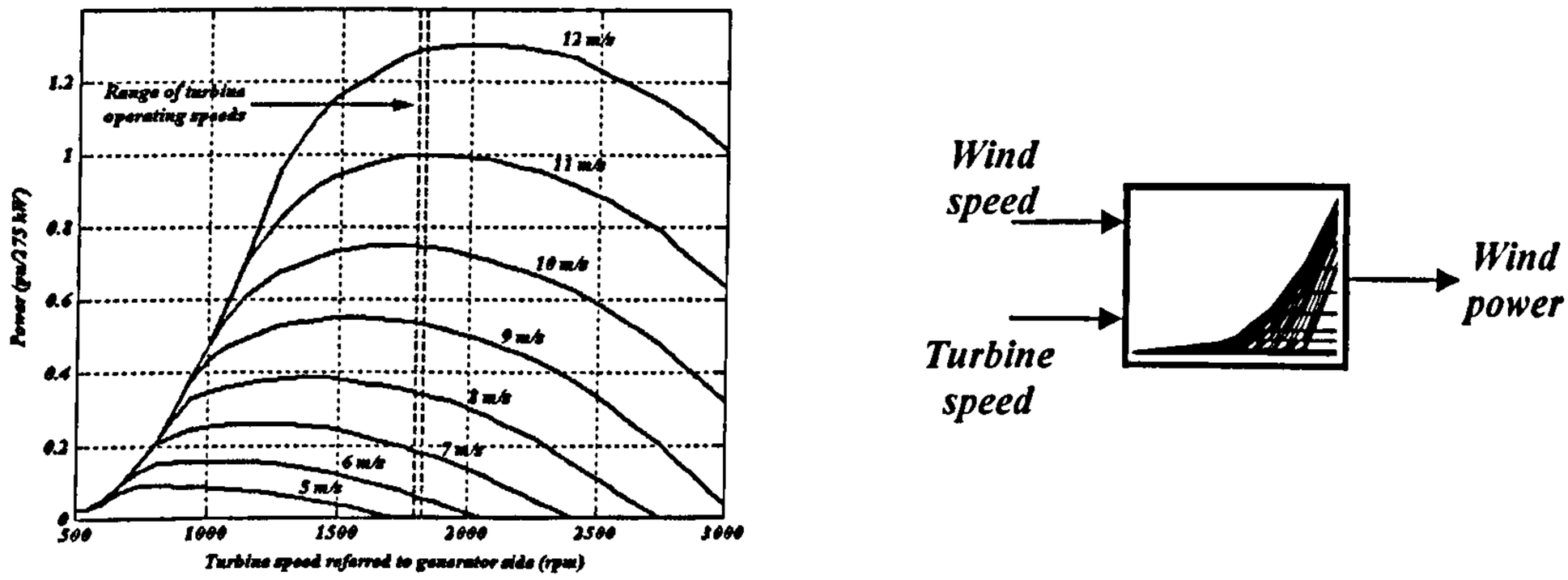


Fig 5.8 Wind-power and turbine speed characteristics at different wind speeds.

5.2.2 The generator model

A comprehensive synchronous generator model is presented, which takes into account dynamic phenomena occurring in the machine. Since this model has been widely known since the 1930s, only the main assumptions and results are presented. The dynamic equations of a synchronous machine are derived for a non salient pole machine with damping bars.

i-Model assumptions

A three-phase, wound-field synchronous generator has three identical armature windings symmetrically distributed around the air-gap, one field winding, and damper windings as shown in fig. 5.9. It is assumed that one damper winding is present in each machine axis. 'Rotor windings' implies the field winding placed on the opposite side of the air gap with respect to the three-phase armature windings.

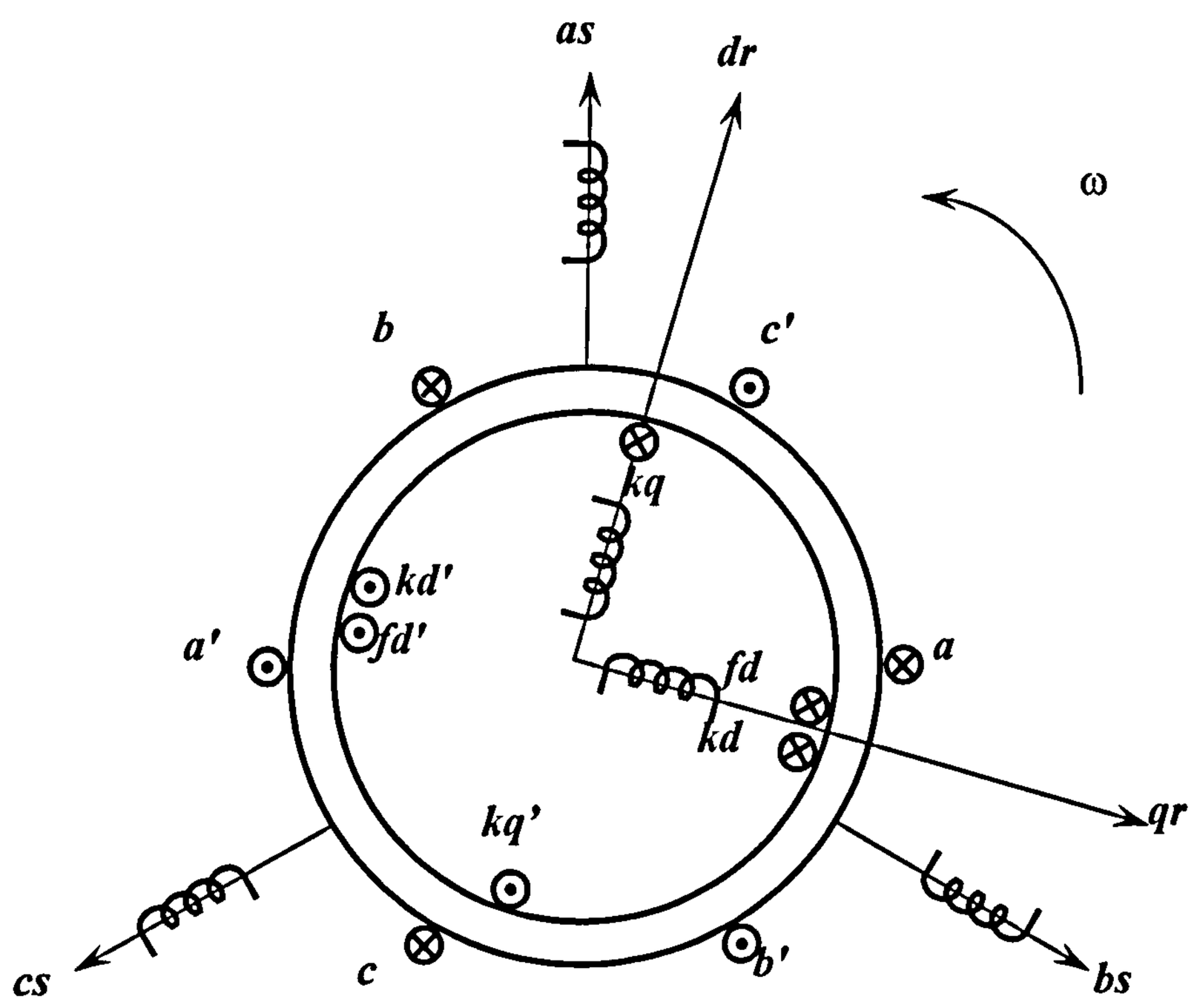


Fig. 5.9 Synchronous machine model

Some approximations are used in order to simplify the real synchronous machine model:

- Every winding in the machine generates a sinusoidal MMF along the air gap.
- The effects of magnetic saturation and flux fringing are neglected, in other words infinite machine iron permeability.

Inaccuracies introduced by these approximations are normally small enough to be negligible, particularly from the point of view of the machine's dynamic performance.

ii- Development of the model equations and equivalent circuit

Electrical equations are obtained by applying Kirchoff's voltage law to all windings [5.12]-[5.14].

$$\begin{aligned} v_{abcs} &= -r_s i_{abcs} + p\lambda_{abcs} \\ v_{dqr} &= r_r i_{dqr} + p\lambda_{dqr} \end{aligned} \qquad 5.17$$

where

$$(f_{abcs})^T = [f_{as} \quad f_{bs} \quad f_{cs}]$$
$$(f_{dqr})^T = [f_{fd} \quad f_{kd} \quad f_{kq}] \qquad 5.18$$

$$r_s = \begin{bmatrix} r_s & 0 & 0 \\ 0 & r_s & 0 \\ 0 & 0 & r_s \end{bmatrix} \qquad r_r = \begin{bmatrix} r_{fd} & 0 & 0 \\ 0 & r_{kd} & 0 \\ 0 & 0 & r_{kq} \end{bmatrix}$$

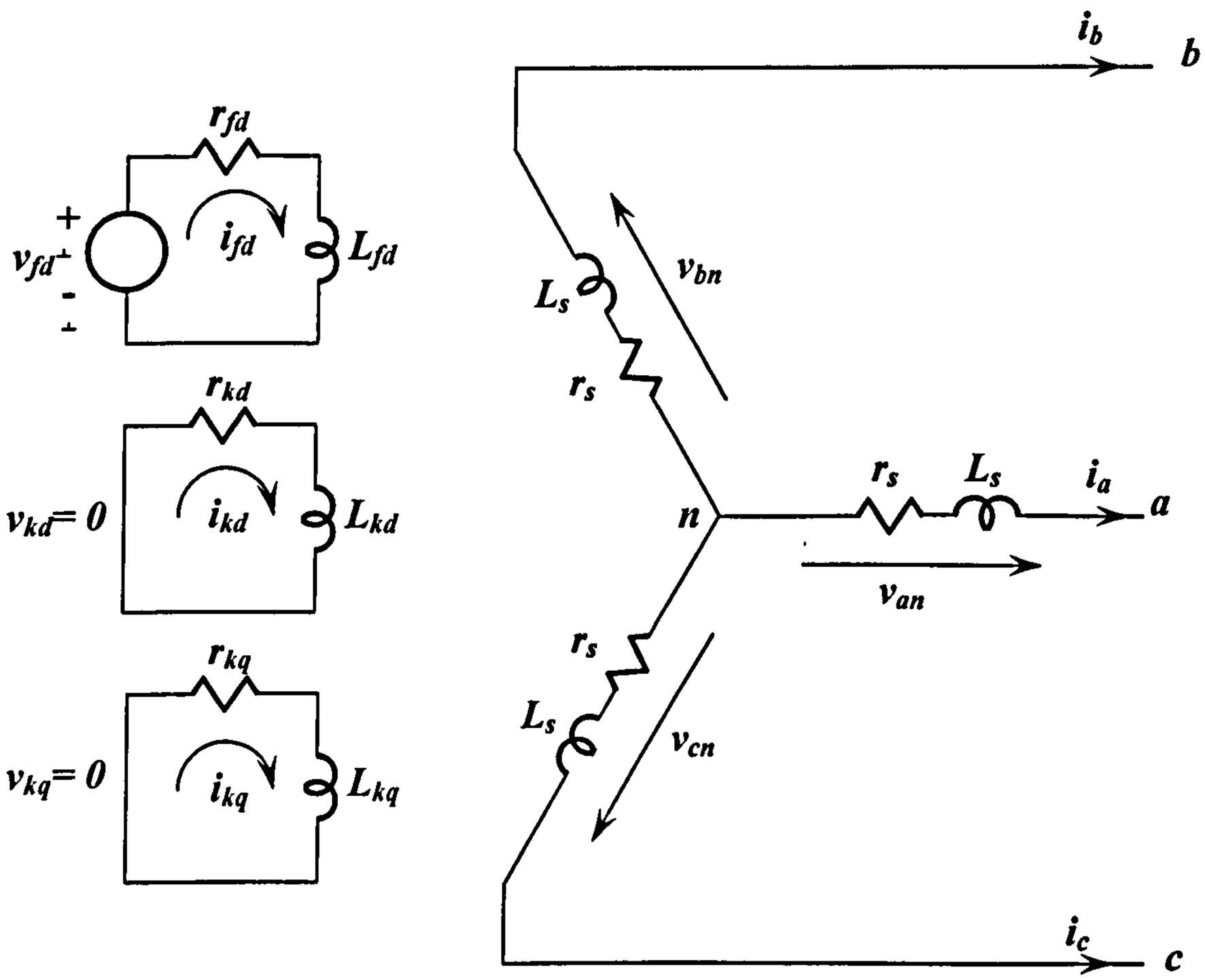


Fig. 5.10 Synchronous machine equivalent circuit in the *abc* frame.

The relationship between the currents and the flux linkages is

$$\begin{bmatrix} \lambda_{abcs} \\ \lambda_{dqr} \end{bmatrix} = \begin{bmatrix} L_s & L_{sr} \\ (L_{sr})^T & L_r \end{bmatrix} \begin{bmatrix} -i_{abcs} \\ i_{dqr} \end{bmatrix} \tag{5.19}$$

It is convenient to transform these equations into the rotor *d-q* reference frame. The voltage and flux linkage equations become time independent. Transformation from the *abc* to the *d-q* reference frame is given by

$$K_s = \frac{2}{3} \begin{bmatrix} \cos\theta & \cos\left(\theta - \frac{2\pi}{3}\right) & \cos\left(\theta + \frac{2\pi}{3}\right) \\ \sin\theta & \sin\left(\theta - \frac{2\pi}{3}\right) & \sin\left(\theta + \frac{2\pi}{3}\right) \\ \frac{1}{2} & \frac{1}{2} & \frac{1}{2} \end{bmatrix} \tag{5.20}$$

where

$$\theta = \int_0^t \omega_r(\zeta)d\zeta + \theta(0) \tag{5.21}$$

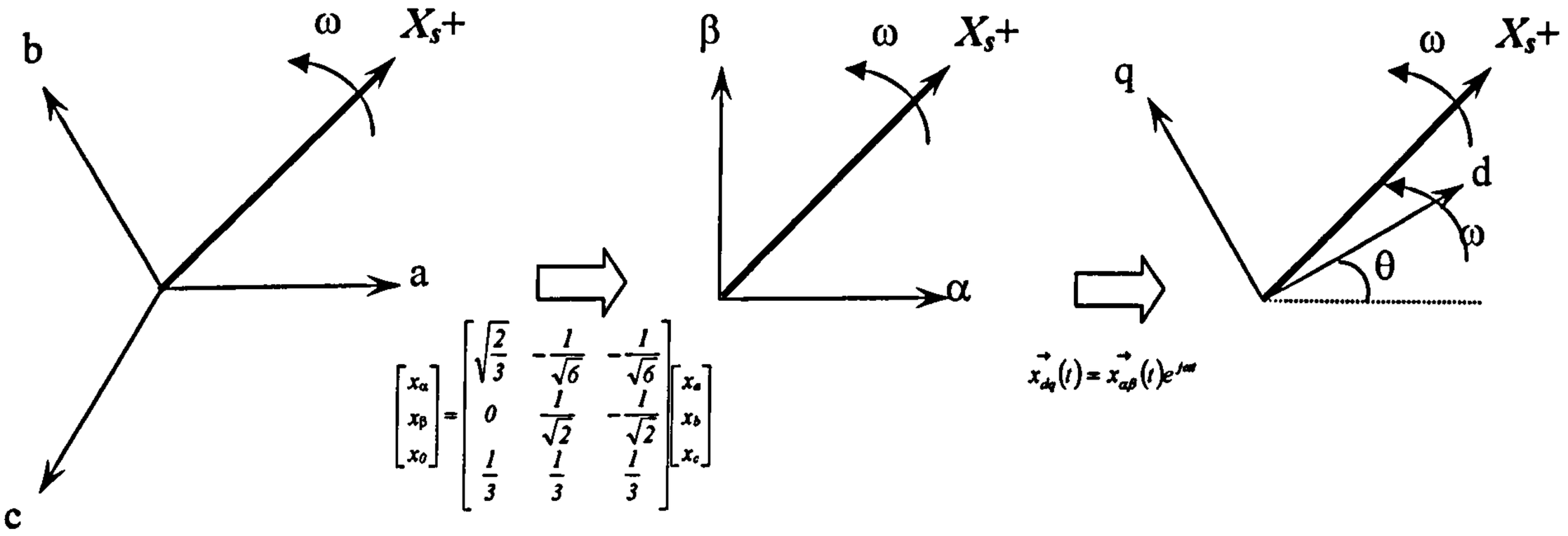


Fig. 5.11 Transformation from the *abc* to the *d-q* frame.

Note that transformation of the variables preserves total system power: in every time instant, power in the *abc* reference frame is equal to the power in the *d-q* reference frame.

$$\begin{aligned} v^r_{dq0s} &= -r_s i^r_{dq0s} + \omega_r \lambda^r_{dq0s} + p\lambda^r_{dq0s} \\ v^r_{dqr} &= r_r i^r_{dqr} + p\lambda^r_{dqr} \end{aligned} \tag{5.22}$$

where

$$\begin{bmatrix} \lambda^r_{dq0s} \\ \lambda^r_{dqr} \end{bmatrix} = \begin{bmatrix} K_s L_s (K_s)^T & K_s L_{sr} \\ L_{sr} (K_s)^T & L_r \end{bmatrix} \begin{bmatrix} -i^r_{dq0s} \\ i^r_{dqr} \end{bmatrix} \tag{5.23}$$

where

$$K_s L_s (K_s)^T = \begin{bmatrix} L_{ls} + L_m & 0 & 0 \\ 0 & L_{ls} + L_m & 0 \\ 0 & 0 & L_{ls} \end{bmatrix} \tag{5.24}$$

$$K_s L_{sr} = \begin{bmatrix} L_m & L_m & 0 \\ 0 & 0 & L_m \\ 0 & 0 & 0 \end{bmatrix} \tag{5.25}$$

$$L_{sr} (K_s)^T = \begin{bmatrix} L_m & 0 & 0 \\ L_m & 0 & 0 \\ 0 & L_m & 0 \end{bmatrix} \tag{5.26}$$

$$\begin{bmatrix} v_d \\ v_q \\ v_0 \\ -v_{fd} \\ -v_{kd} \\ -v_{kq} \end{bmatrix} = -\begin{bmatrix} r_s & \omega(L_{ls} + L_m) & 0 & 0 & 0 & \omega L_m \\ -\omega(L_{ls} + L_m) & r_s & 0 & -\omega L_m & -\omega L_m & 0 \\ 0 & 0 & r_s & 0 & 0 & 0 \\ 0 & 0 & 0 & r_{fd} & 0 & 0 \\ 0 & 0 & 0 & 0 & r_{kd} & 0 \\ 0 & 0 & 0 & 0 & 0 & r_s \end{bmatrix} \begin{bmatrix} i_d \\ i_q \\ i_0 \\ i_{fd} \\ i_{kd} \\ i_{kq} \end{bmatrix} \tag{5.27}$$

$$-\begin{bmatrix} L_{ls} + L_m & 0 & 0 & L_m & L_m & 0 \\ 0 & L_{ls} + L_m & 0 & 0 & 0 & L_m \\ 0 & 0 & L_{ls} & 0 & 0 & 0 \\ L_m & 0 & 0 & L_{ls} + L_m & 0 & 0 \\ L_m & 0 & 0 & 0 & L_{ls} + L_m & 0 \\ 0 & L_m & 0 & 0 & 0 & L_{ls} + L_m \end{bmatrix} \begin{bmatrix} pi_d \\ pi_q \\ pi_0 \\ pi_{fd} \\ pi_{kd} \\ pi_{kq} \end{bmatrix}$$

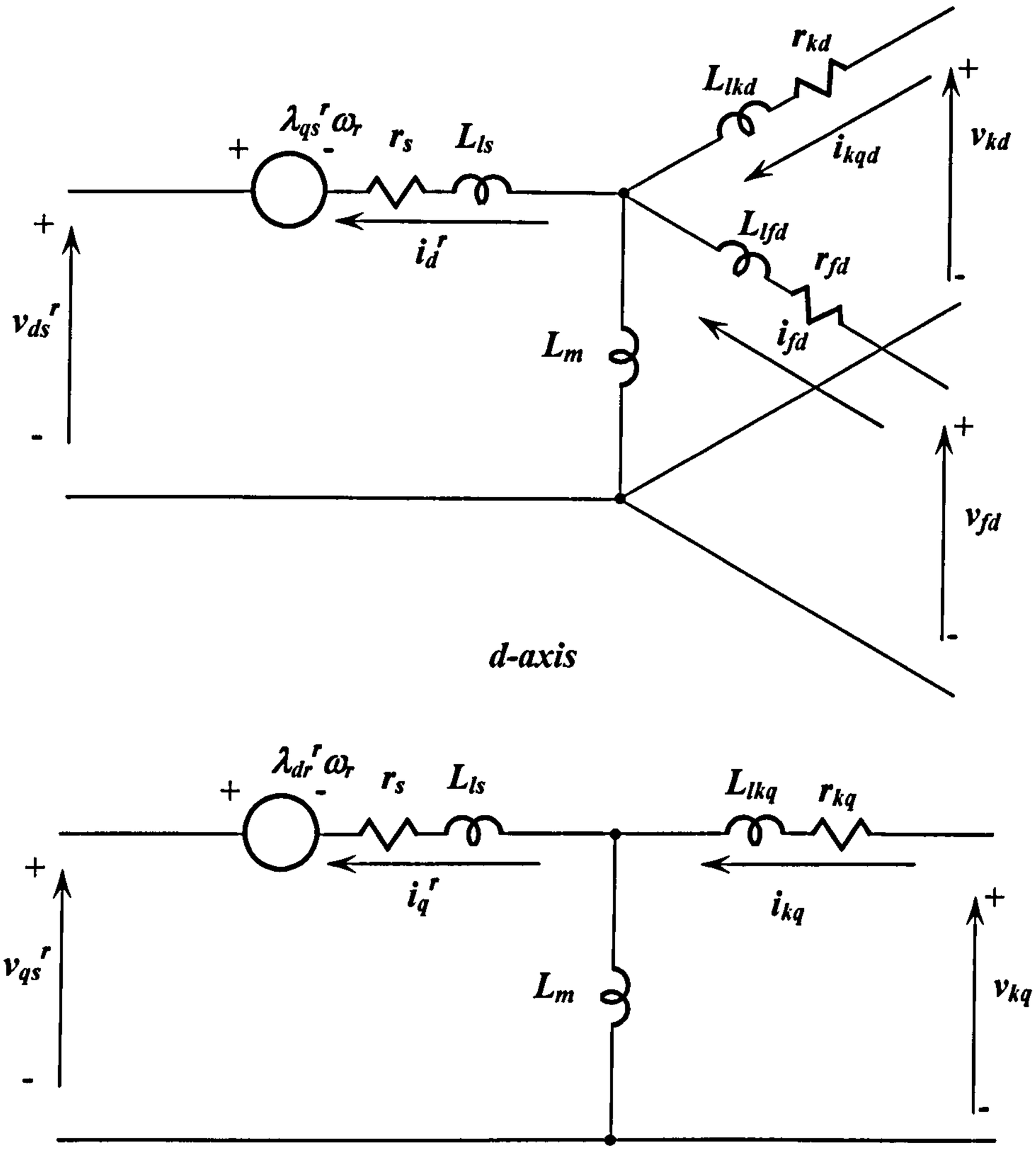


Fig. 5.12 Synchronous machine equivalent circuit in the *d-q* frame.

5.2.3 The input filter

Since the synchronous generator normally has a relatively high magnetizing inductance, there is no need to use a high order input filter. The filter is reduced to only capacitors as shown in fig. 5.13.

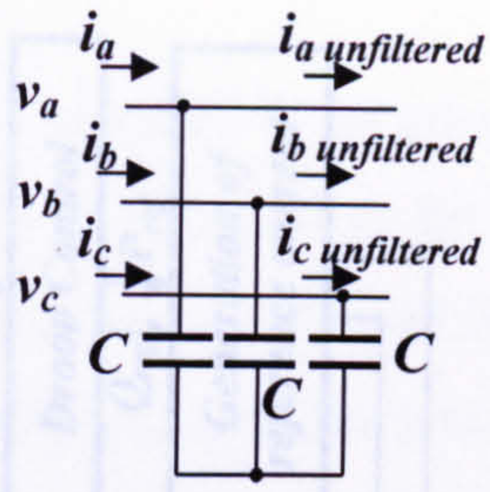


Fig. 5.13 Input filter reduced to only capacitors.

The filter equation in vector form is:

$$C\frac{d\bar{v}_s}{dt} = \bar{i}_s - \bar{i}_{unfiltered} \tag{5.29}$$

and in the *d-q* form

$$C\frac{d\bar{v}_{dq}}{dt} - j\omega C\bar{v}_{dq} = \bar{i}_{dq} - \bar{i}_{dq\ unfiltered} \tag{5.30}$$

$$C\frac{dv_d}{dt} + \omega Cv_q = i_d - i_{d\ unfiltered} \tag{5.31}$$

$$C\frac{dv_q}{dt} - \omega Cv_d = i_q - i_{q\ unfiltered}$$

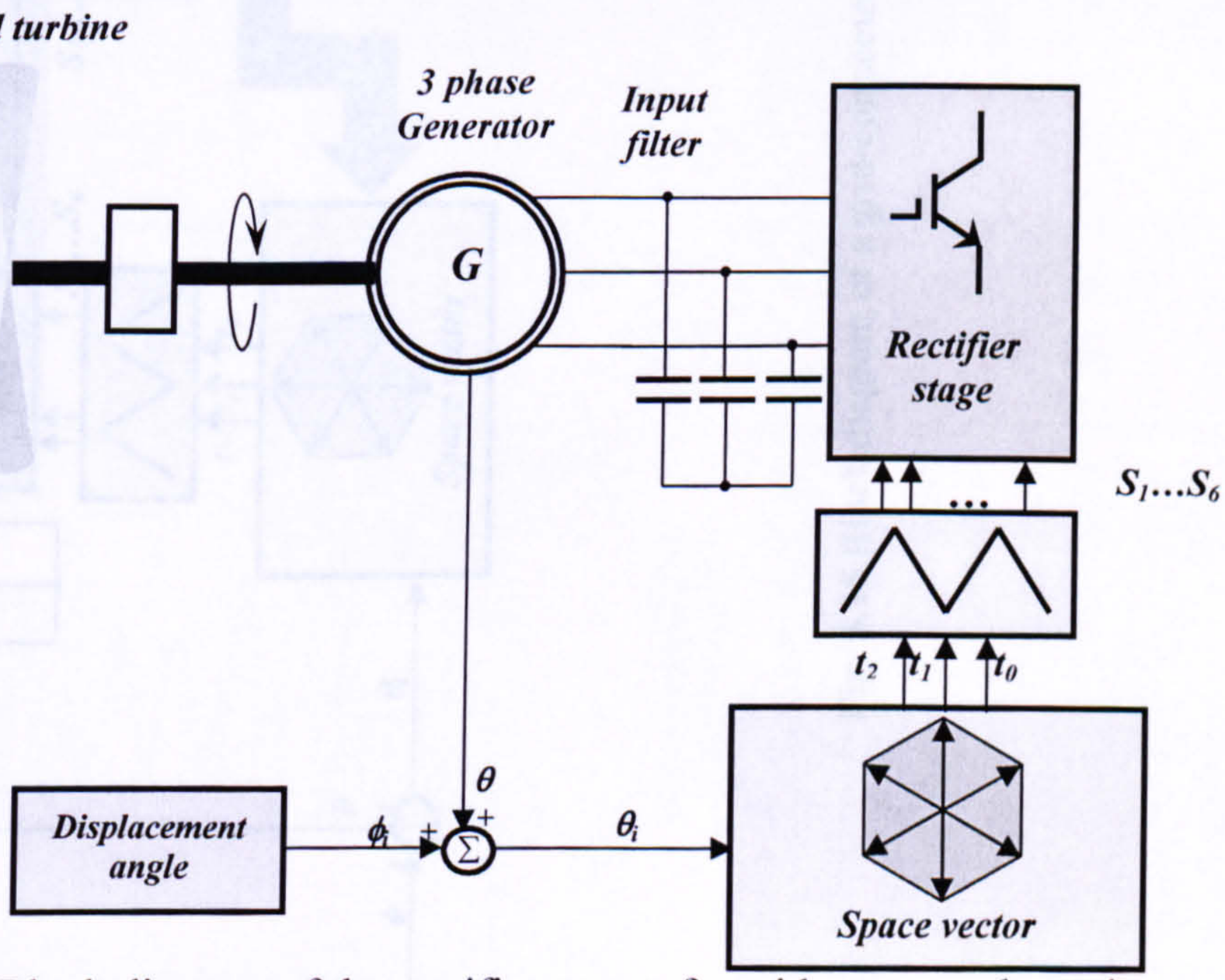


Fig. 5.14 Block diagram of the rectifier stage of a grid-connected matrix converter with an L-filter interfaced.

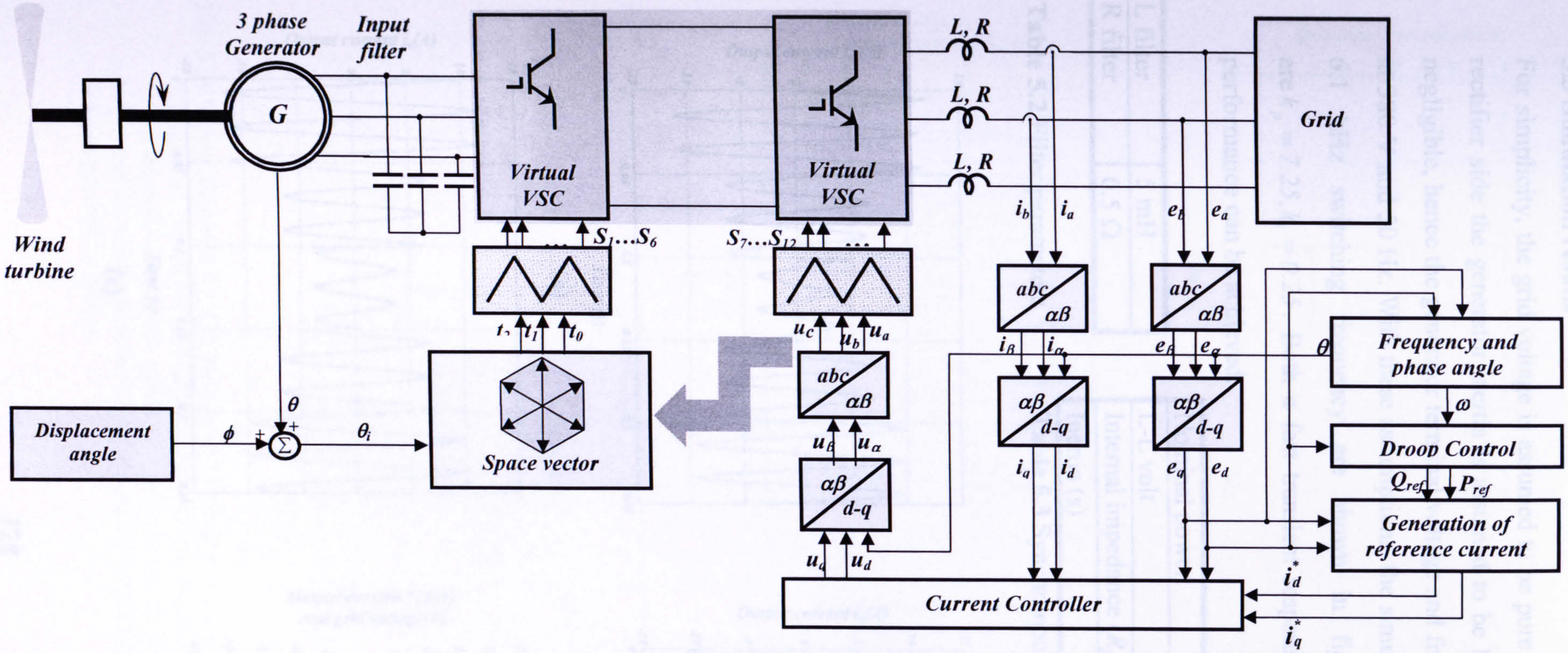


Fig. 5.15 Block diagram of a grid-connected matrix converter with an L-filter interface.

5.3 Simulation results

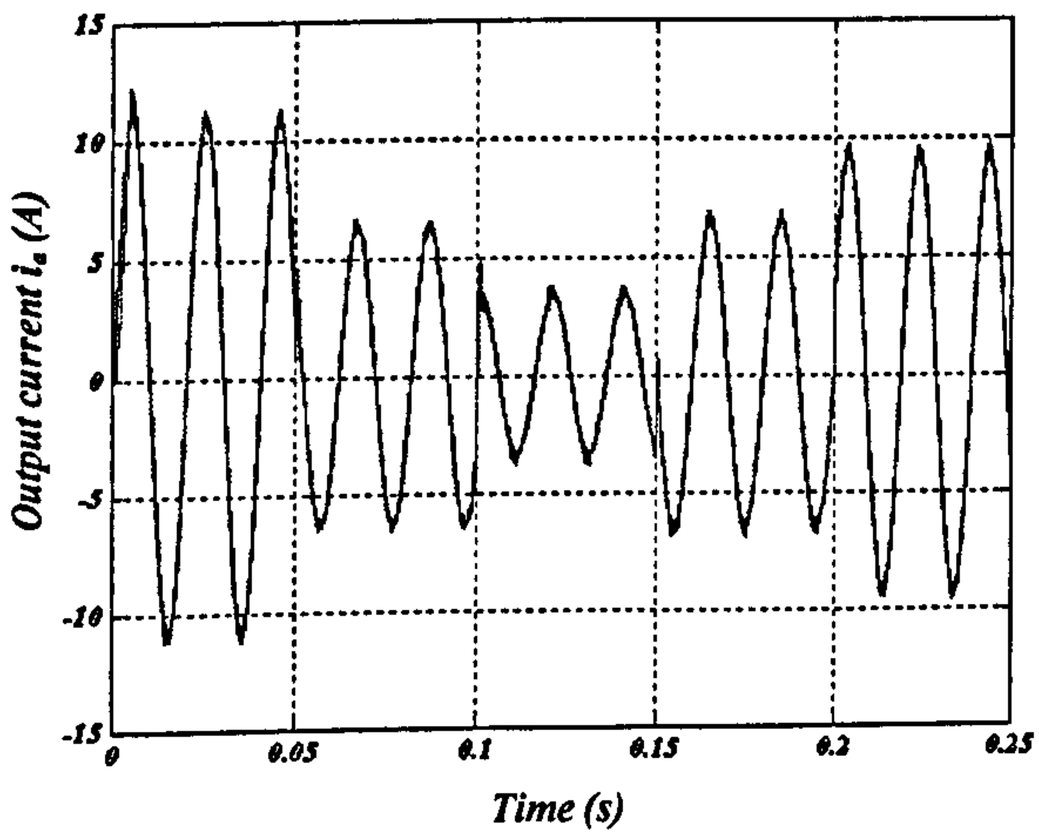
For simplicity, the grid voltage is assumed to be pure balanced sinusoids. Also, for the rectifier side the generator inertia is assumed to be high, making the speed variation negligible, hence the generator terminal voltage and frequency can be assumed constant at 380 V and 50 Hz. With these assumptions, the simulation using Simulink results at a 6.1 kHz switching frequency are shown in fig. 5.16. The controller gains are $k_p = 7.25, k_i = 0.25$. Both a fast transient response and a zero steady state error performance can be achieved.

L filter	5 mH
R filter	0.5 Ω

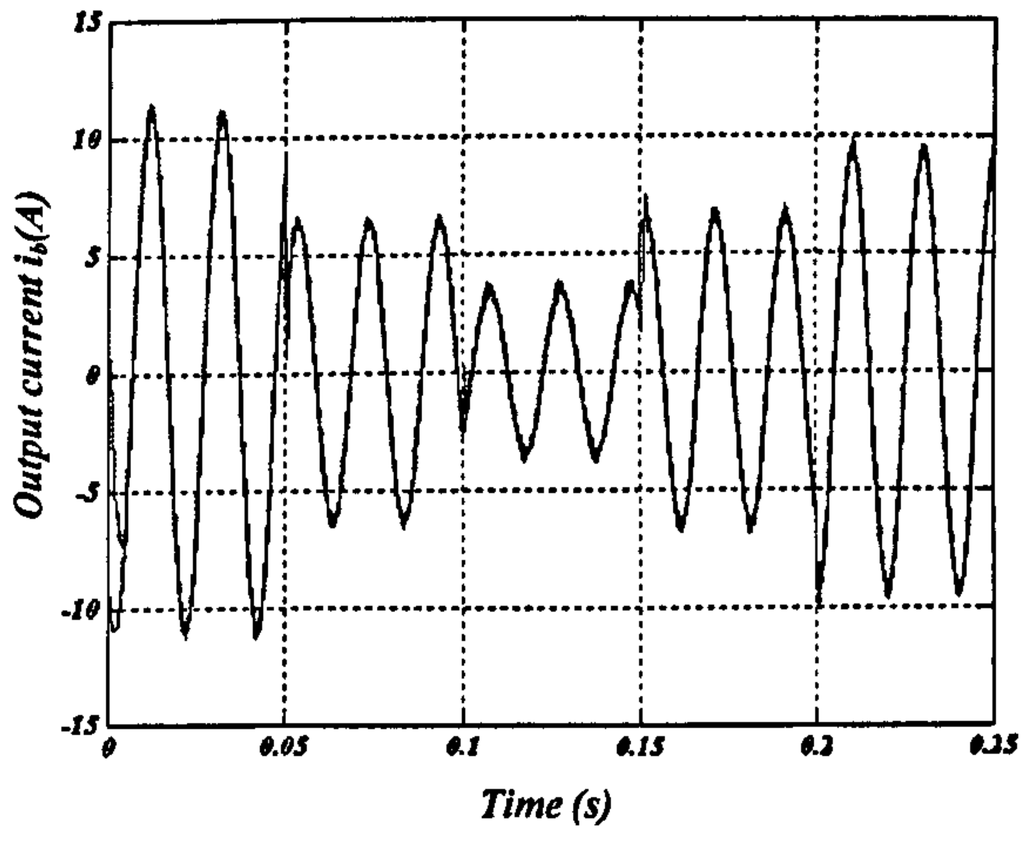
Table 5.2 Filter parameters

Nominal power	4 kW
L-L volt	380V
Internal impedance (R_{pu}, X_{pu})	0.2 & 3 pu
Inertia (s)	3.7

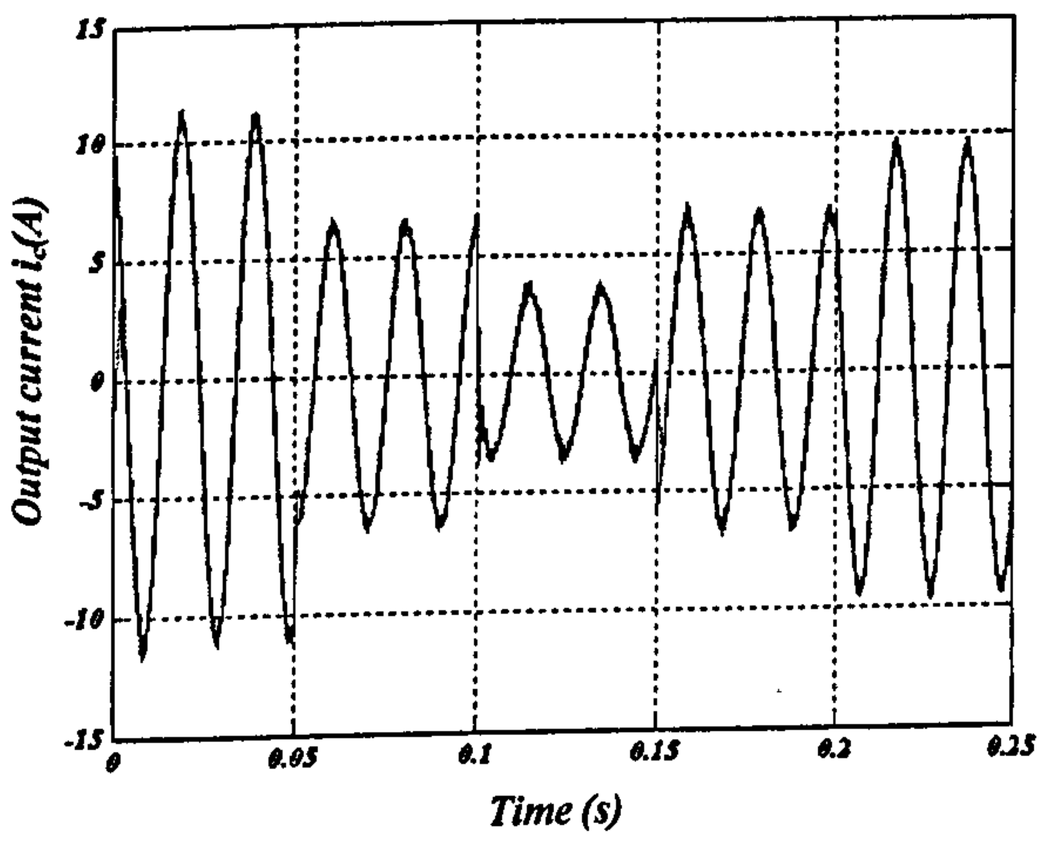
Table 5.3 Synchronous machine parameters



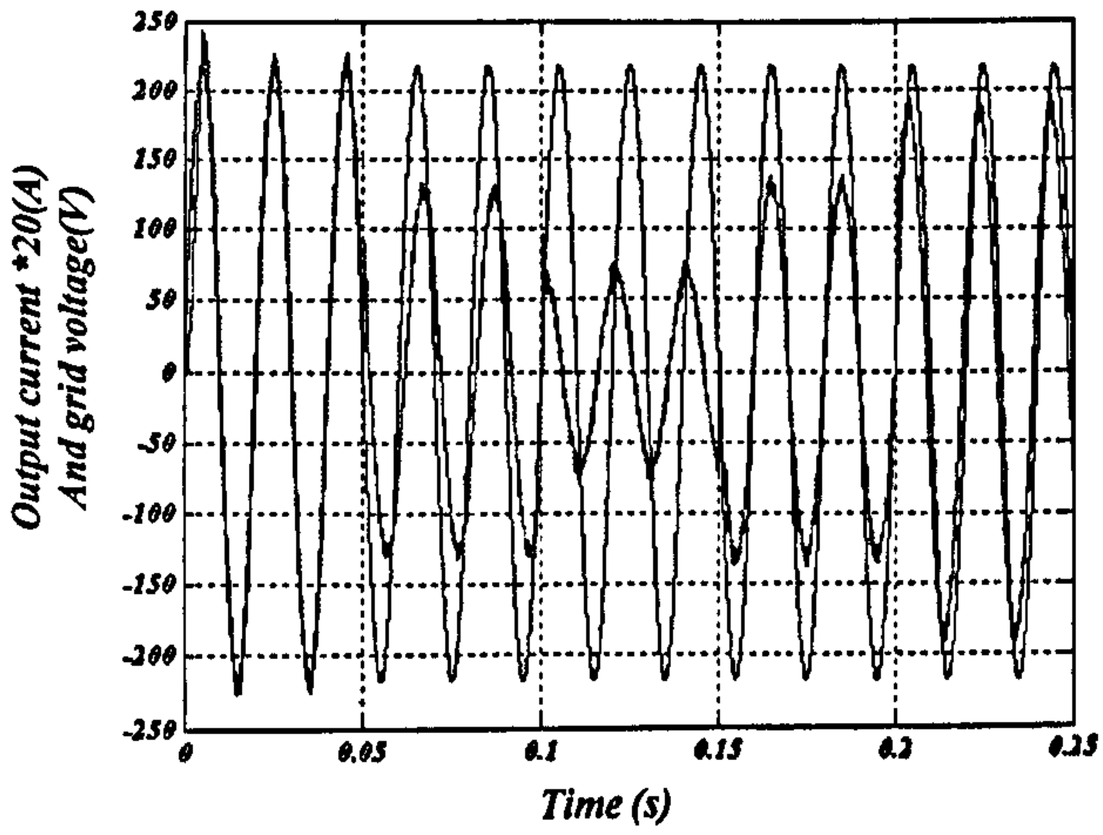
(a)



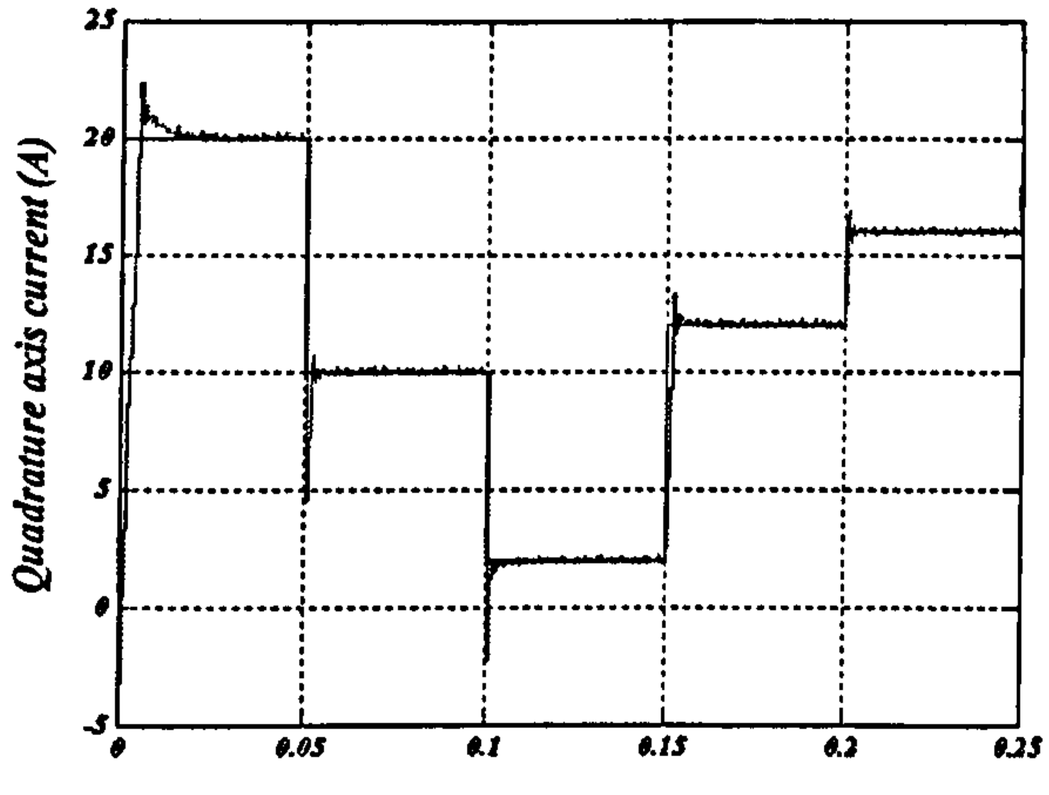
(b)



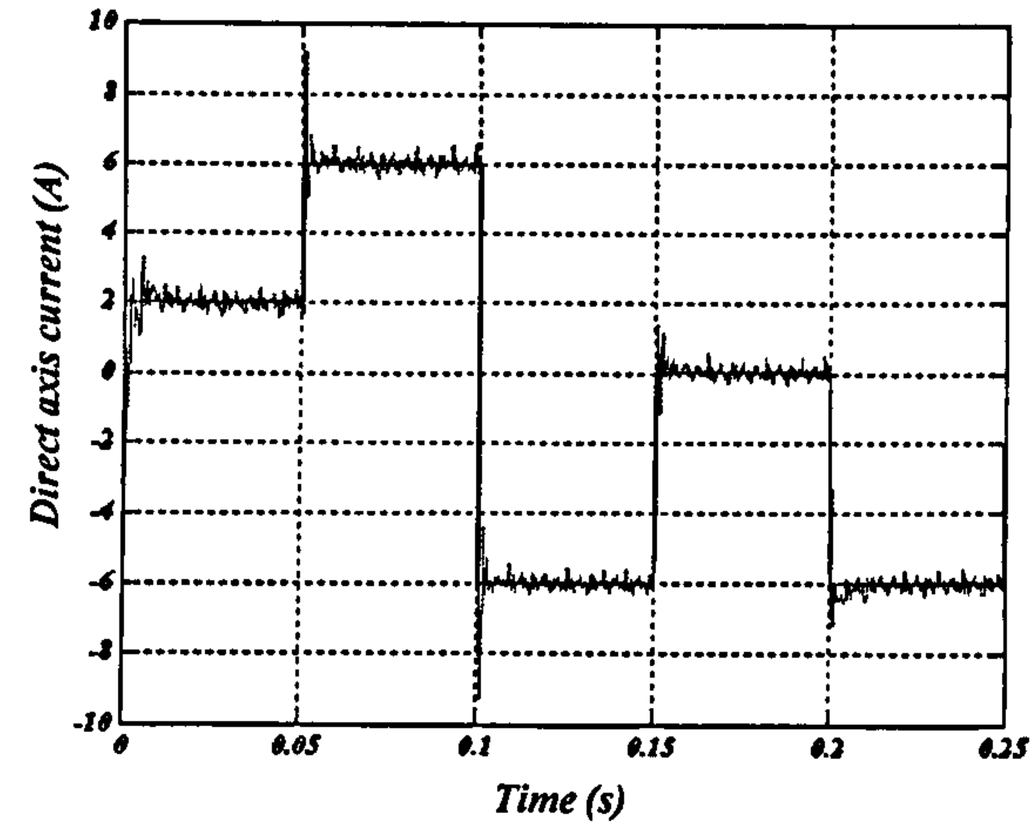
(c)



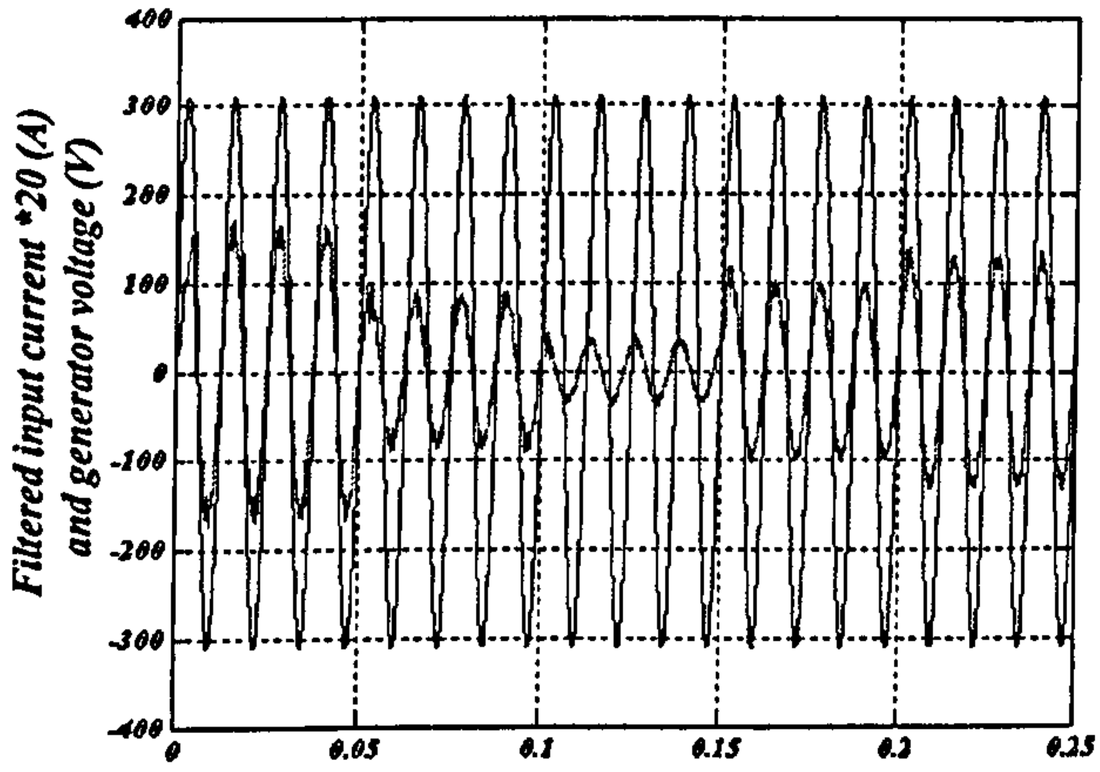
(d)



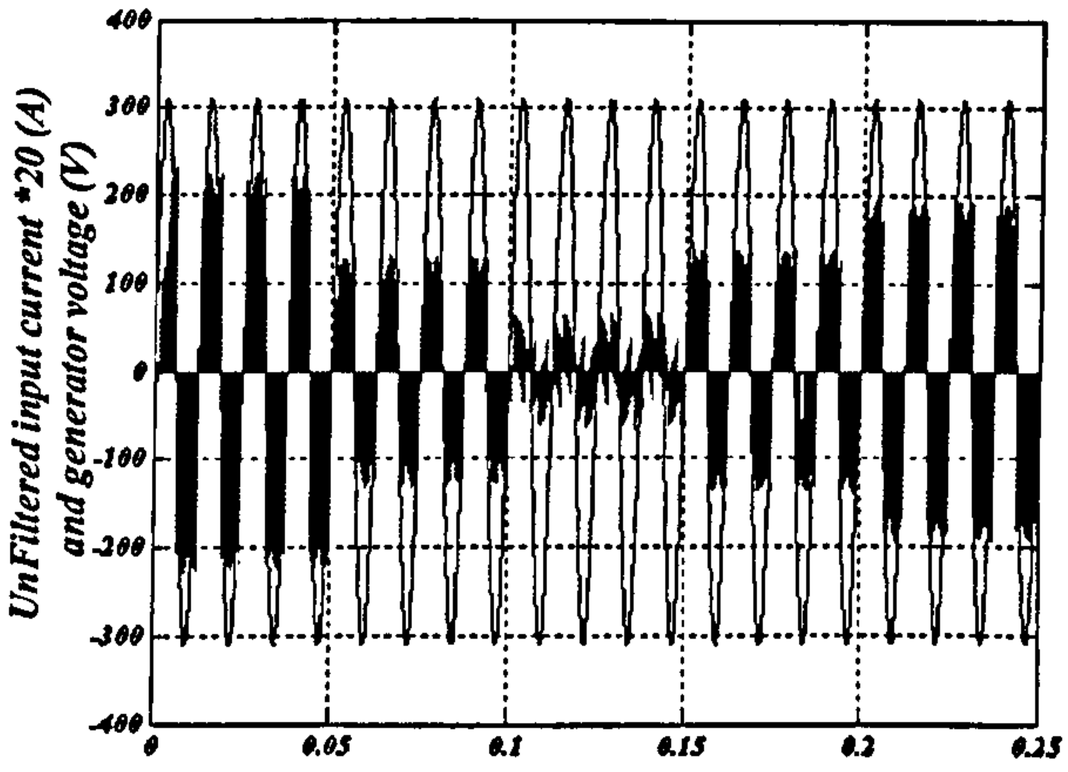
(e)



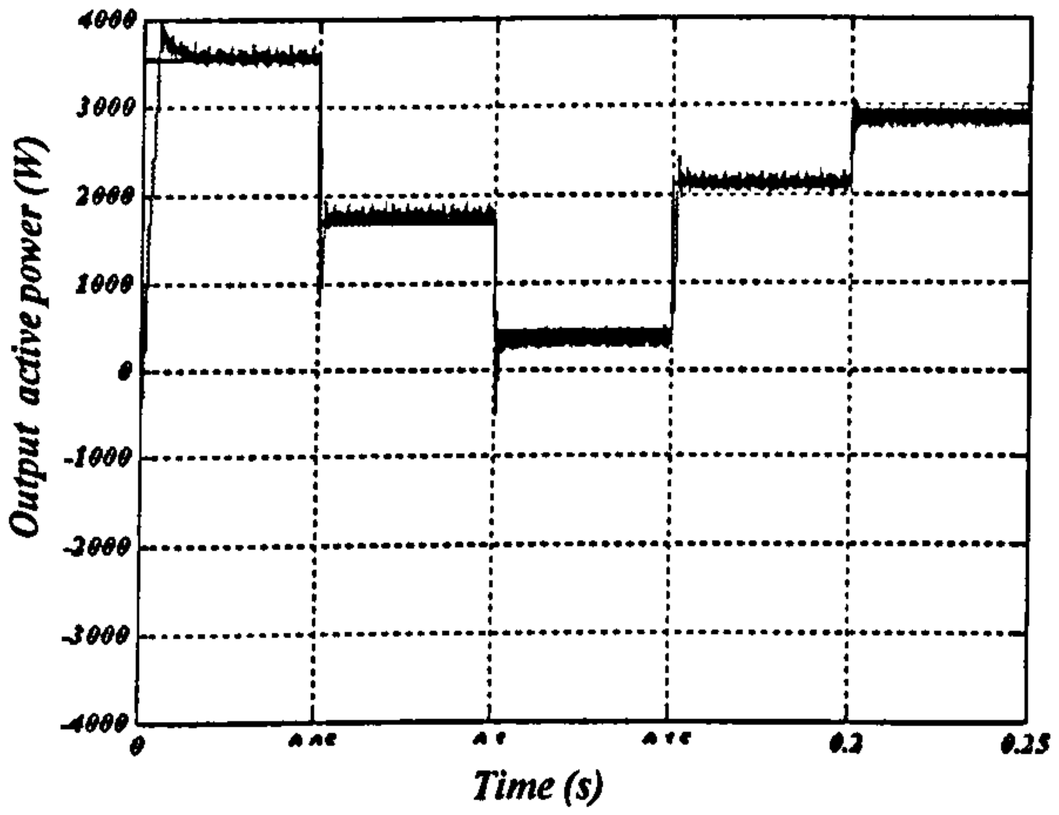
(f)



(g)



(h)



(i)

Fig. 5.16 Output response for a sinusoidal grid voltage at a 6.1 kHz switching frequency: (a) output current phase 'a', (b) output current phase 'b', (c) output current phase 'c', (d) output current phase (a)*20 and the grid voltage, (e) quadrature axis current, (f) direct axis current, (g) filtered input current*20 and the generator voltage, (h) unfiltered input current*20 and the generator voltage, and (i) output active power

5.4 Unbalanced grid condition

In case of an unbalanced grid voltage as shown in fig. 5.17, the actual quadrature and direct axis voltages with and without the frequency and phase angle estimators are shown in fig. 5.18.

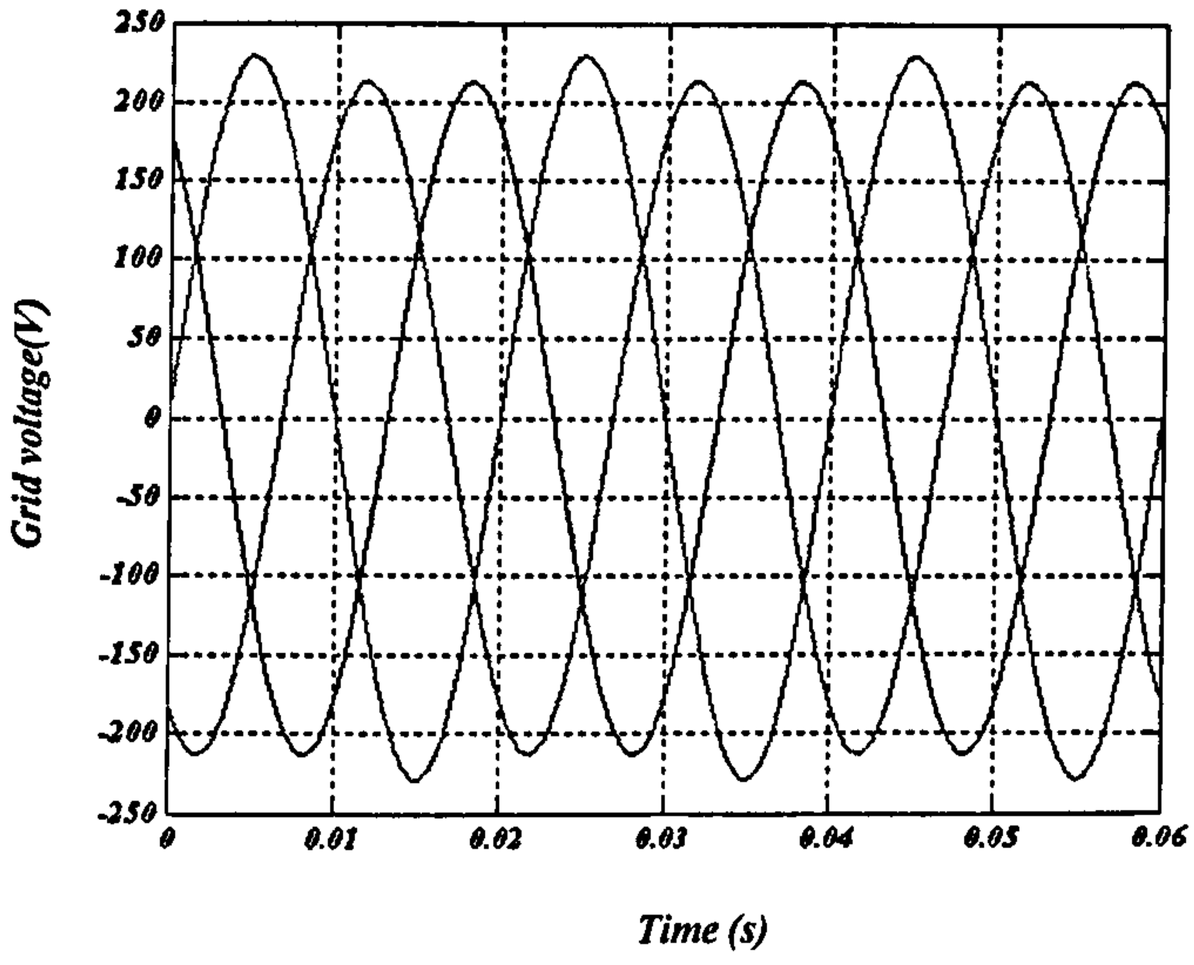


Fig. 5.17 Unbalanced grid voltage

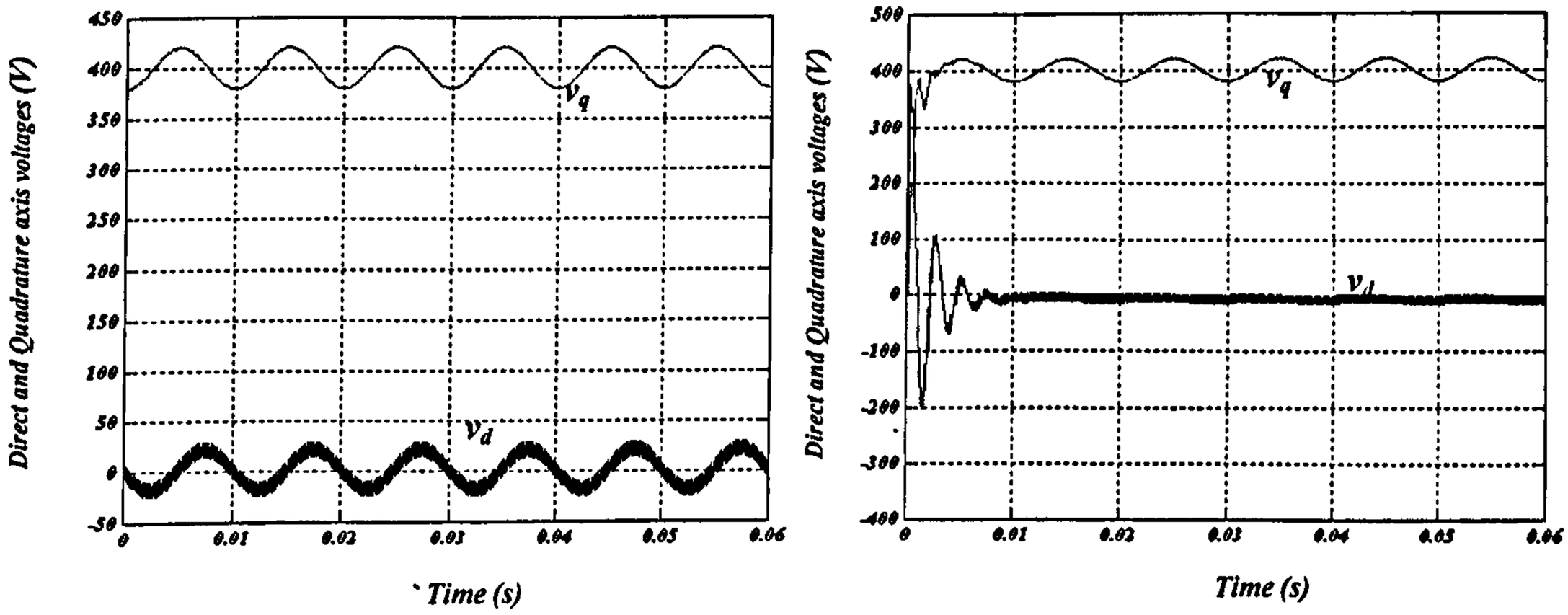
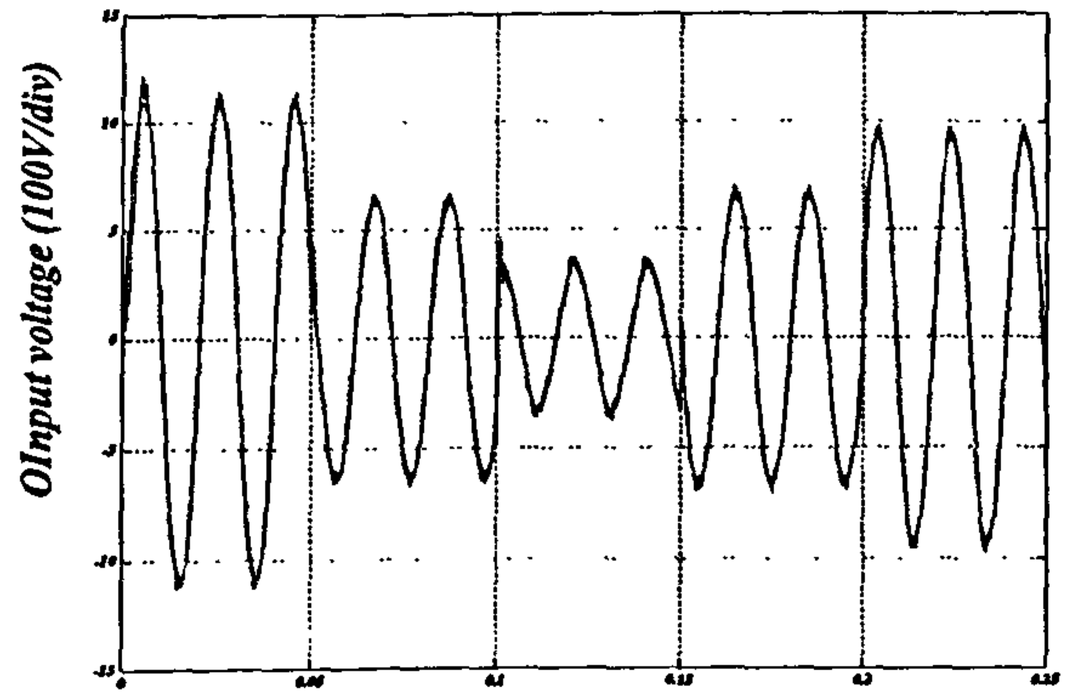
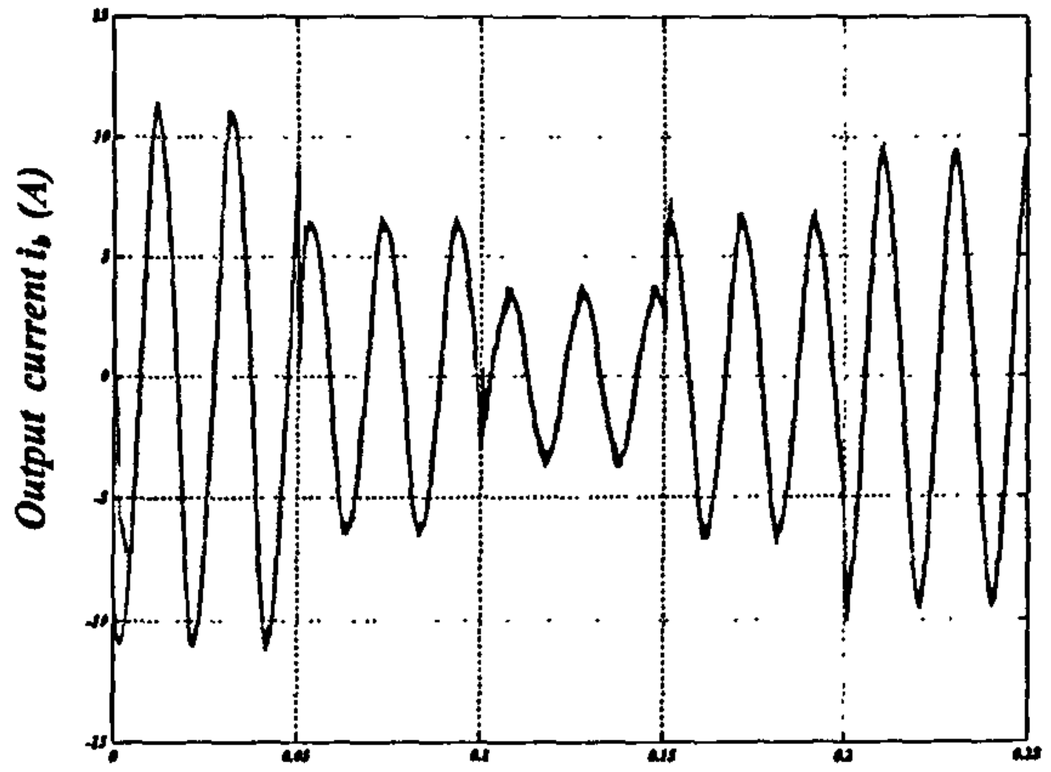


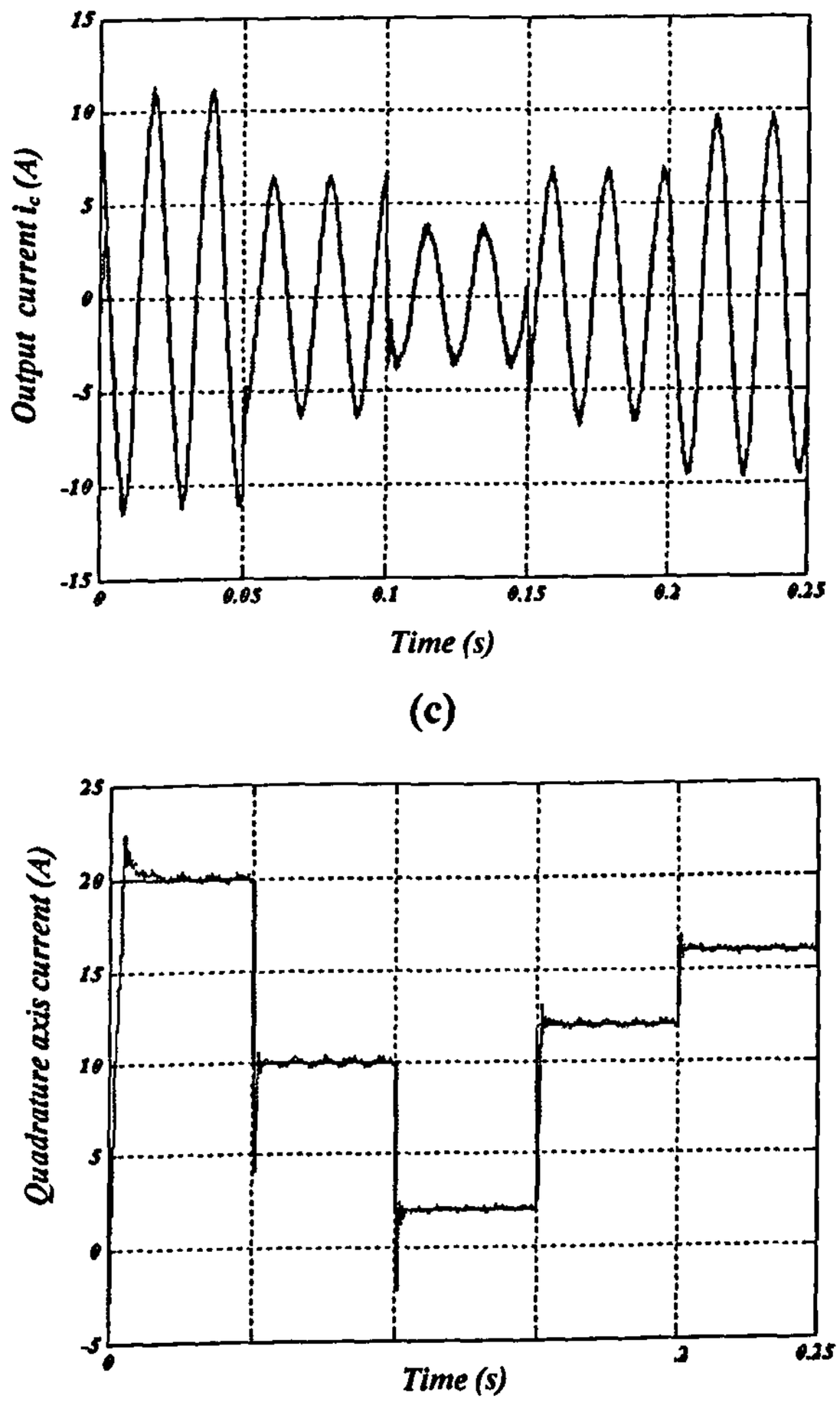
Fig. 5.18 Direct and quadrature axis voltages:
(a) actual direct and quadrature axis voltages versus time response and
(b) direct and quadrature axis voltages versus time response using the frequency estimator.


Olnput voltage (100V/div)
Time (s)

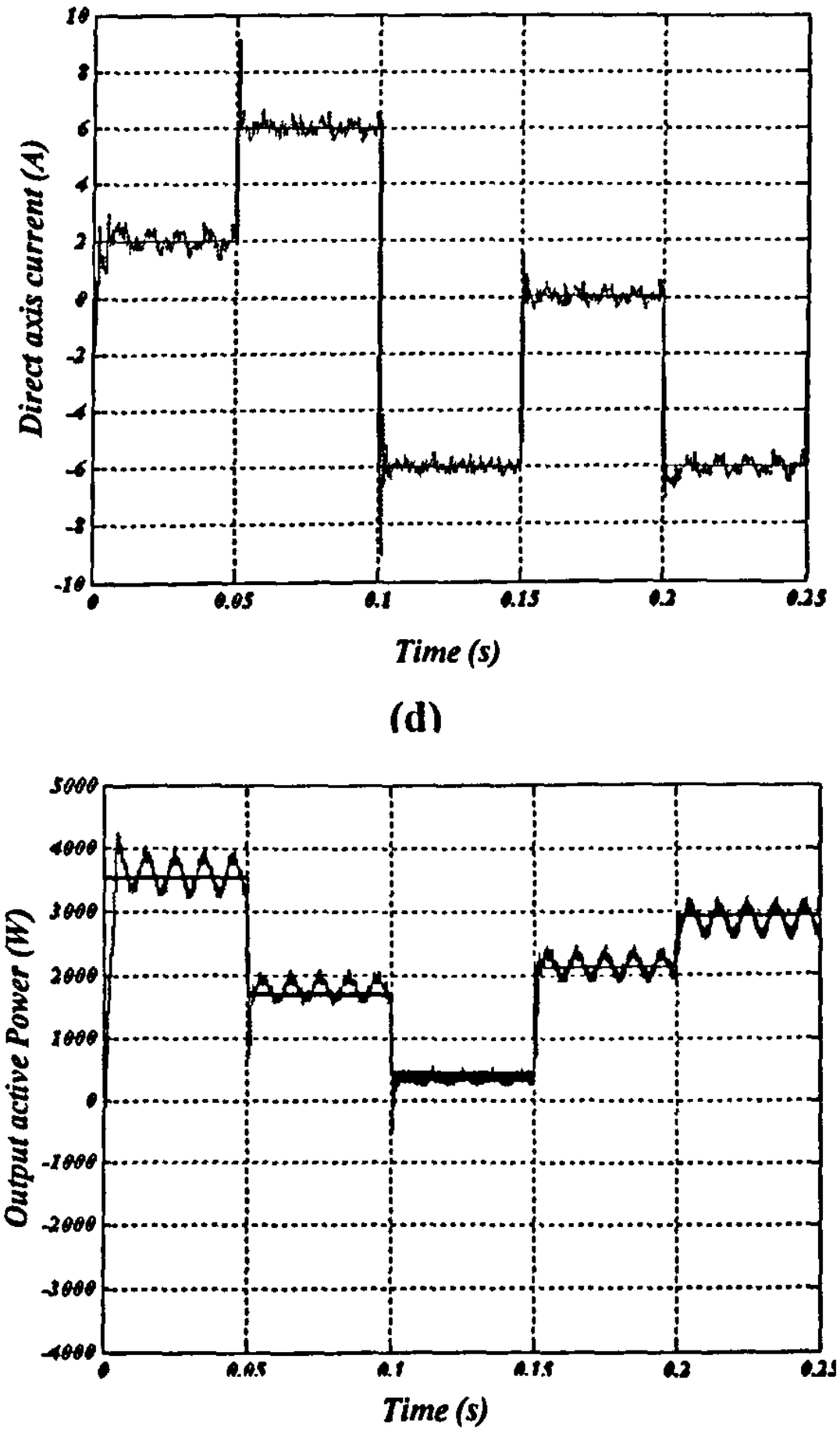
(a)


Output current i_b (A)
Time (s)

(b)


Output current i_c (A)
Time (s)

(c)


Direct axis current (A)
Time (s)

(d)

Quadrature axis current (A)
Time (s)

(e)

Output active Power (W)
Time (s)

(f)

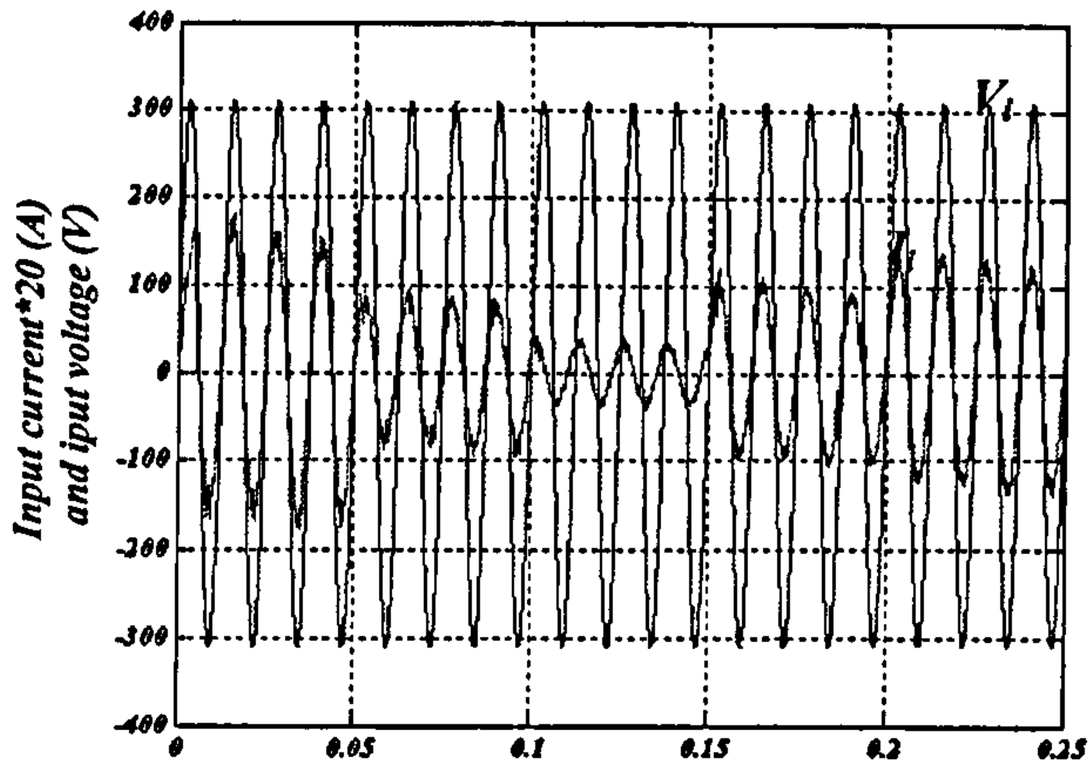


Fig. 5.19 Output response for unbalanced grid voltage at a 6.1 kHz switching frequency: (a) output current phase 'a', (b) output current phase 'b', (c) output current phase 'c', (d) quadrature axis current, (e) direct axis current, (f) output active power, and (g) filtered input current*20 and the generator voltage.

Even though balanced pure sinusoidal output currents can be achieved, the output active power oscillates at double the grid frequency around the reference value due to the negative sequence voltage. This is also reflected in the input current waveform. In order to obtain non-oscillatory output power, the controller should be modified to take into account the effect of the negative sequence voltage by using dual current control.

5.4.1 Dual vector current controller

In the case of unbalanced grid voltages, the normal vector current controller does not give satisfactory performance.

A dual vector current controller which consists of two separate PI controllers, one for controlling the positive-sequence and the other for controlling the negative-sequence, should be considered. This controller gives good performance when used for grid current control. A simplified block diagram for this controller is shown in Fig. 5.20. The positive sequence PI-controller is described in the positive *d-q* frame as:

$$v_{d+,k+1} = e_{d+,k} + R.\sum_{n=0}^{k}\left(i^{*}_{d+,n} - i_{d+,n}\right) + \left(\frac{L}{T_s} + \frac{R}{2}\right).\left(i^{*}_{d+,k} - i_{d+,k}\right) - \omega L\frac{i^{*}_{q+,k} + i_{q+,k}}{2}$$

$$v_{q+,k+1} = e_{q+,k} + R.\sum_{n=0}^{k}\left(i^{*}_{q+,n} - i_{q+,n}\right) + \left(\frac{L}{T_s} + \frac{R}{2}\right).\left(i^{*}_{q+,k} - i_{q+,k}\right) + \omega L\frac{i^{*}_{d+,k} + i_{d+,k}}{2} \qquad 5.32$$

The negative sequence PI-controller is described in the positive *d-q* frame as:

$$v_{d-,k+1} = e_{d-,k} + R.\sum_{n=0}^{k}\left(i^{*}_{d-,n} - i_{d-,n}\right) + \left(\frac{L}{T_s} + \frac{R}{2}\right).\left(i^{*}_{d-,k} - i_{d-,k}\right) - \omega L\frac{i^{*}_{q-,k} + i_{q-,k}}{2}$$

$$v_{q-,k+1} = e_{q-,k} + R.\sum_{n=0}^{k}\left(i^{*}_{q-,n} - i_{q-,n}\right) + \left(\frac{L}{T_s} + \frac{R}{2}\right).\left(i^{*}_{q-,k} - i_{q-,k}\right) + \omega L\frac{i^{*}_{d-,k} + i_{d-,k}}{2} \qquad 5.33$$

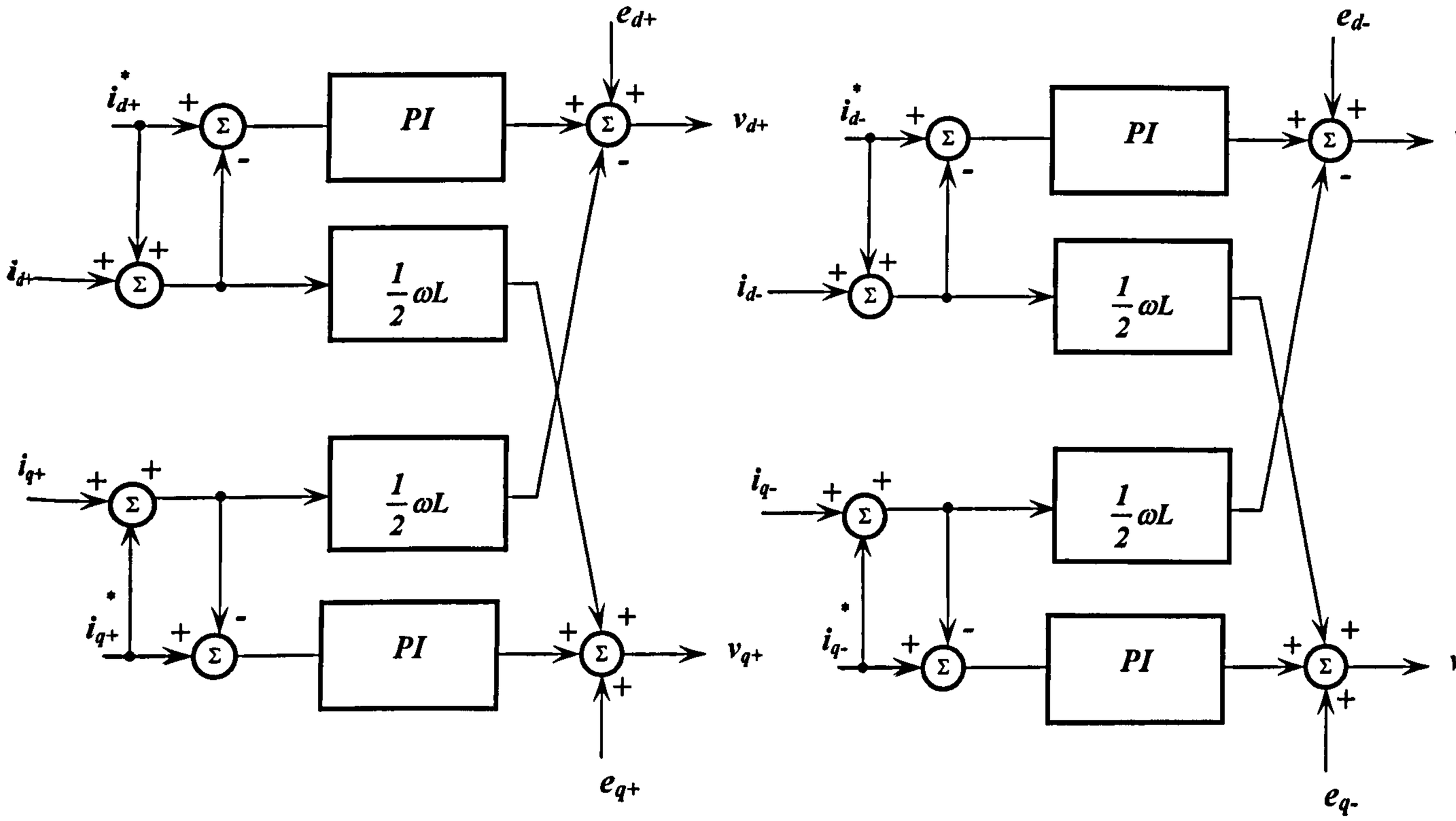


Fig. 5.20 Dual PI current controllers for an unbalanced power supply

5.4.2 Extraction of the positive and negative sequence components

Decomposition of either the grid voltage or the output current into positive and negative sequences is performed in a *d-q* frame that rotates in the positive direction. From fig. 5.21, it can be established that the positive sequence vector rotates in the same direction and with the same speed which means that the *d-q* components will be dc values. On the other hand the negative sequence rotates in the negative direction, which means that its *d-q* component vectors rotate with double the line frequency in the opposite direction. If the measured quantity in the *d-q* frame is delayed by 90^0, the negative sequence component is shifted by 180^0. Adding this to the original quantity nulls the negative sequence component.

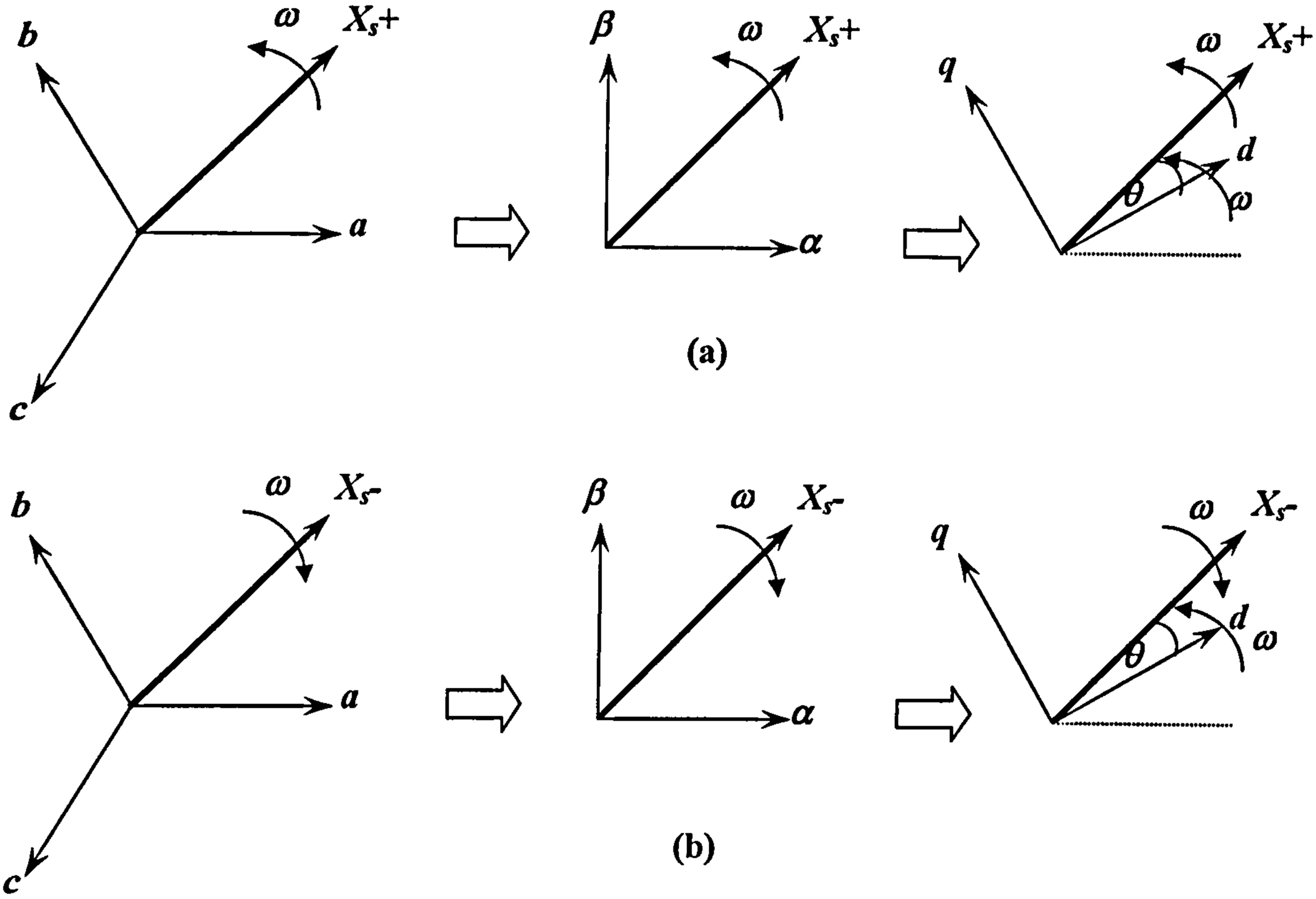


Fig. 5.21 Transformation from the *abc* to the *d-q* frame for:
(a) the positive sequence and
(b) the negative sequence.

The positive sequence components can thus be extracted from the measured values as:

$$x_{d+}(t)=\frac{1}{2}\left(x_d(t)+x_d\left(t-\frac{T}{4}\right)\right)$$
$$x_{q+}(t)=\frac{1}{2}\left(x_q(t)+x_q\left(t-\frac{T}{4}\right)\right) \qquad 5.34$$

while the negative sequence components are:

$$x_{d-}(t)=\frac{1}{2}\left(x_d(t)-x_d\left(t-\frac{T}{4}\right)\right)$$
$$x_{q-}(t)=\frac{1}{2}\left(x_q(t)-x_q\left(t-\frac{T}{4}\right)\right) \qquad 5.35$$

Noted that the negative sequence values $x_{d-}(t)$ and $x_{q-}(t)$ calculated in 5.35 are sinusoidal signals with double the supply frequency. In order to convert these signals to dc values for the current controller equation 5.33, the following transformation is used

$$x^{n}{}_{dq-}(t) = e^{-2j\theta}\, x_{dq-}(t) \tag{5.36}$$

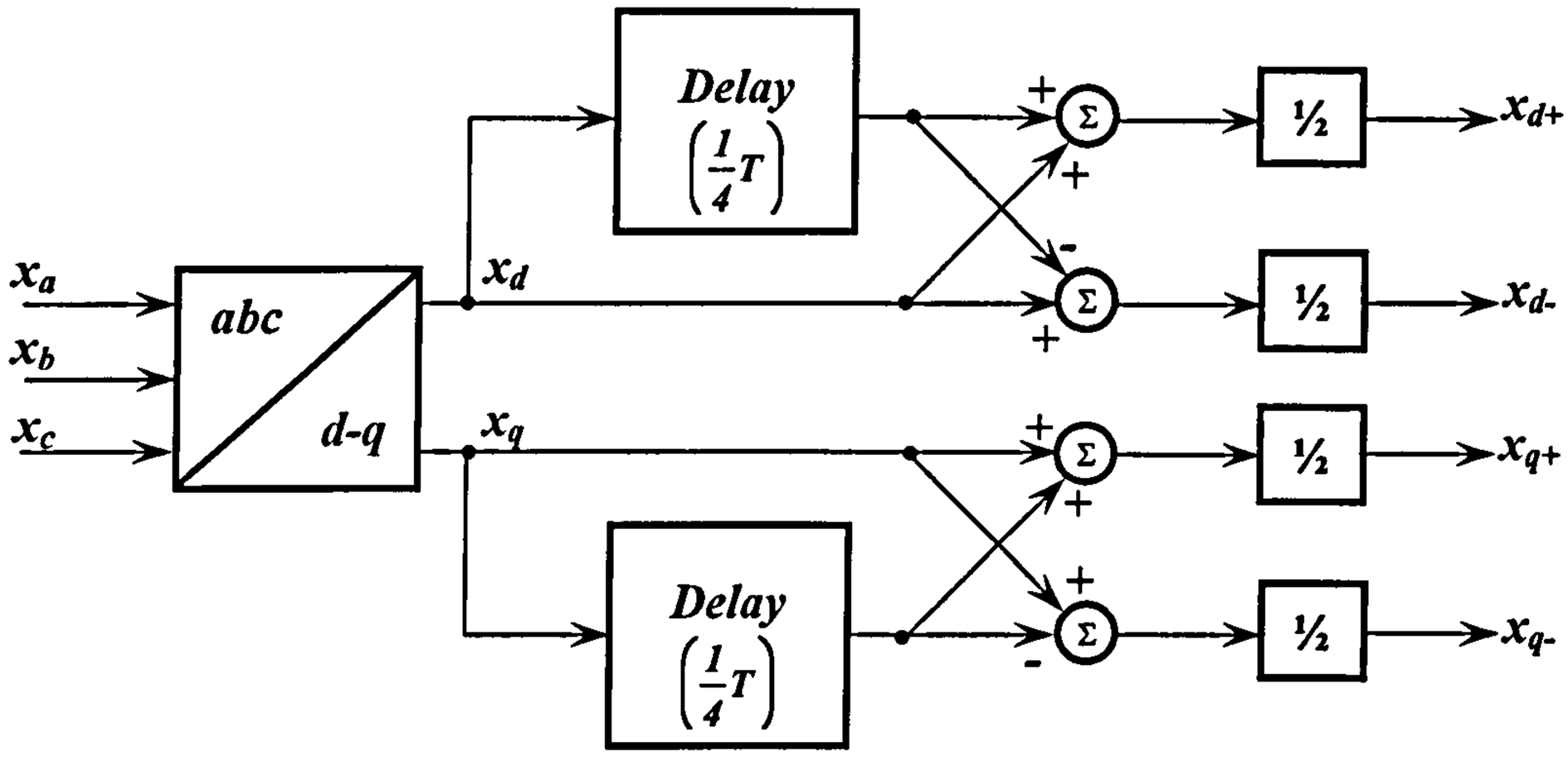


Fig. 5.22 Extraction of the positive and negative sequence components

5.4.3 Generation of the reference currents

For an unbalance supply, the apparent power S can be defined in terms of grid voltages and currents resolved into positive and negative sequences in the d-q frame as:

$$S_{o/p} = \vec{e}_{dq}.\vec{i}^{*}_{dq} = \left(e_{dq+}e^{j\omega t} + e_{dq-}e^{-j\omega t}\right)\left(i_{dq+}e^{j\omega t} + i_{dq-}e^{-j\omega t}\right)^{*} \tag{5.37}$$

which can be written as :

$$\begin{aligned} S_{o/p} &= \left(e_{dq+}e^{j\omega t} + e_{dq-}e^{-j\omega t}\right)\left(i^{*}_{dq+}e^{-j\omega t} + i^{*}_{dq-}e^{j\omega t}\right) \\ &= e_{dq+}i^{*}_{dq+} + e_{dq-}i^{*}_{dq-} + e_{dq+}i^{*}_{dq-}e^{j2\omega t} + e_{dq-}i^{*}_{dq+}e^{-j2\omega t} \end{aligned} \tag{5.38}$$

Rearranging this equation gives

$$\begin{aligned} S_{o/p} &= S_{+} + S_{-} + (a+jb)e^{j2\omega t} + (c+jd)e^{-j2\omega t} \\ &= S_{+} + S_{-} + \left((a+c)\cos 2\omega t + (d-b)\sin 2\omega t\right) \\ &\quad + j\left((a+c)\sin 2\omega t - (d-b)\cos 2\omega t\right) \end{aligned} \tag{5.39}$$

where

$$\begin{aligned} S_{+} &= \left(e_{d+}i_{d+} + e_{q+}i_{q+}\right) + j\left(e_{q+}i_{d+} - e_{d+}i_{q+}\right) \\ S_{-} &= \left(e_{d-}i_{d-} + e_{q-}i_{q-}\right) + j\left(e_{q-}i_{d-} - e_{d-}i_{q-}\right) \\ a &= e_{d+}i_{d-} + e_{q+}i_{q-} \qquad b = e_{d-}i_{q+} - e_{q-}i_{d+} \\ c &= e_{d-}i_{d+} + e_{q-}i_{q+} \qquad d = e_{d+}i_{q-} - e_{q+}i_{d-} \end{aligned} \tag{5.40}$$

In order to achieve non oscillating power, the terms $(a + c)$ and $(d - b)$ are set to zero. Equation 5.40 can be written in matrix form as:

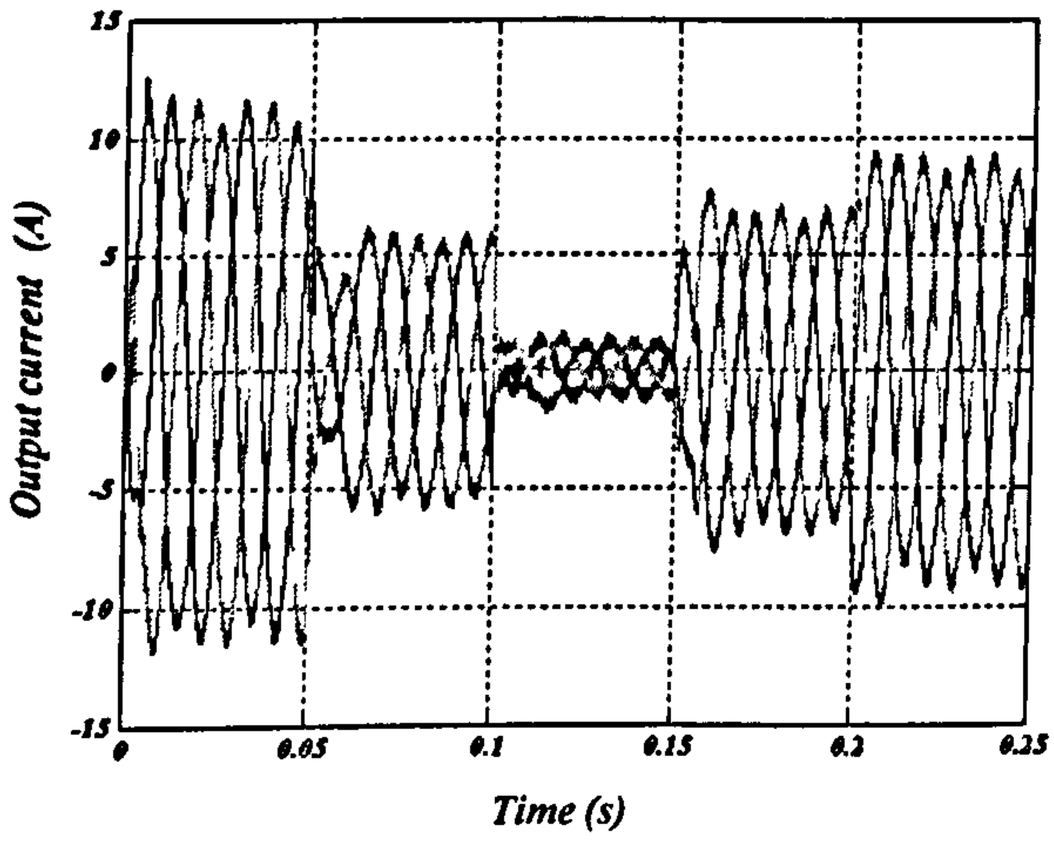
$$\begin{bmatrix} P \\ Q \\ a+c \\ d-b \end{bmatrix} = \begin{bmatrix} e_{d+} & e_{q+} & e_{d-} & e_{q-} \\ e_{q+} & -e_{d+} & e_{q-} & -e_{d-} \\ e_{q-} & -e_{d-} & -e_{q+} & e_{d+} \\ e_{d-} & e_{q-} & e_{d+} & e_{q+} \end{bmatrix} \begin{bmatrix} i_{d+} \\ i_{q+} \\ i_{d-} \\ i_{q-} \end{bmatrix} \qquad 5.41$$

Hence the reference currents can be calculated from:

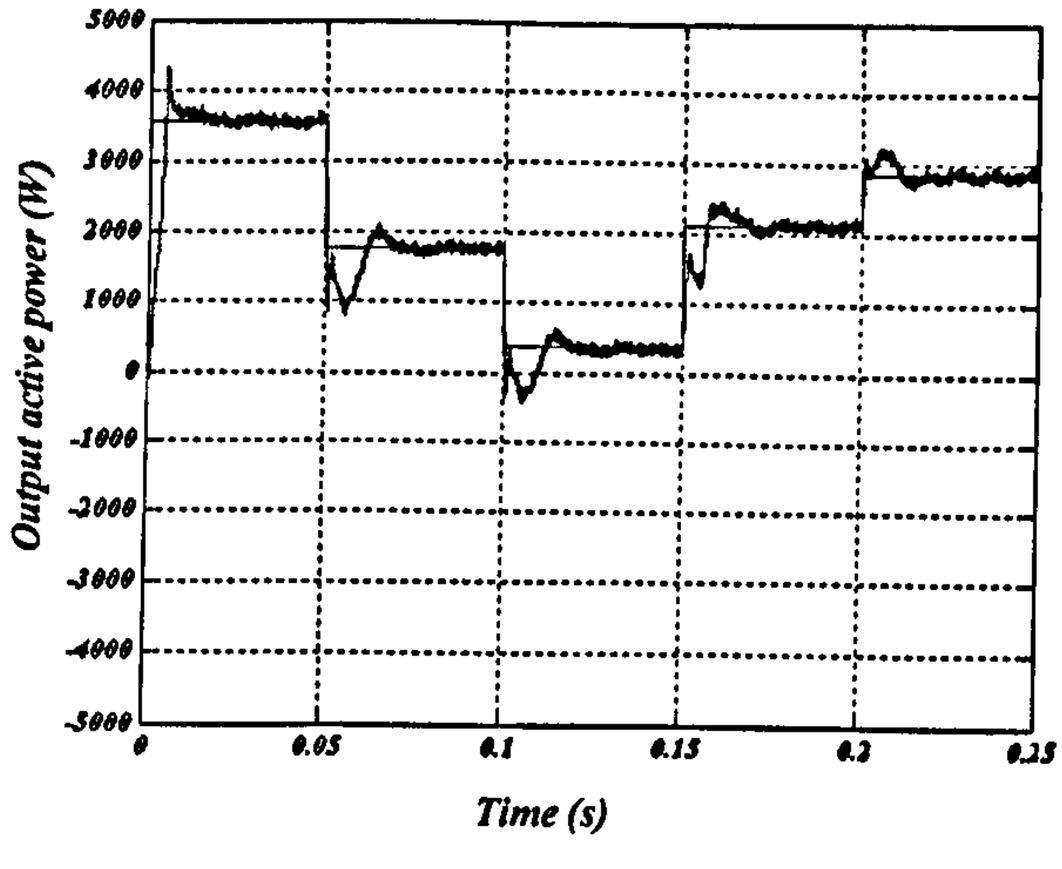
$$\begin{bmatrix} i_{d+} \\ i_{q+} \\ i_{d-} \\ i_{q-} \end{bmatrix} = \begin{bmatrix} e_{d+} & e_{q+} & e_{d-} & e_{q-} \\ e_{q+} & -e_{d+} & e_{q-} & -e_{d-} \\ e_{q-} & -e_{d-} & -e_{q+} & e_{d+} \\ e_{d-} & e_{q-} & e_{d+} & e_{q+} \end{bmatrix}^{-1} \cdot \begin{bmatrix} P \\ Q \\ 0 \\ 0 \end{bmatrix} \qquad 5.42$$

5.4.4 Simulation results

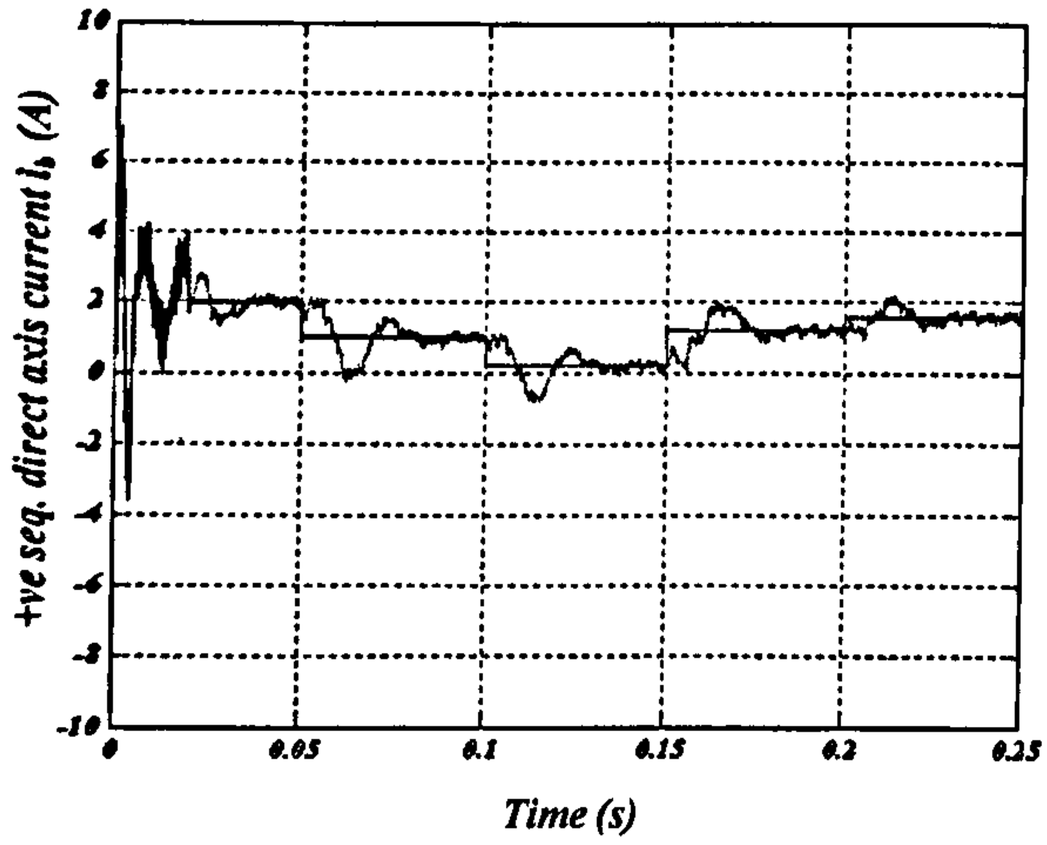
The simulation results with the modified current controller (same controller gains) are shown in fig. 5.23. The active power tracking is improved and the double frequency oscillations in the output power are suppressed. This improvement is at the expense of system transient response. The settling time increases since a time delay of a quarter of the cycle is introduced in order to extract the true values of the positive and negative sequence components.

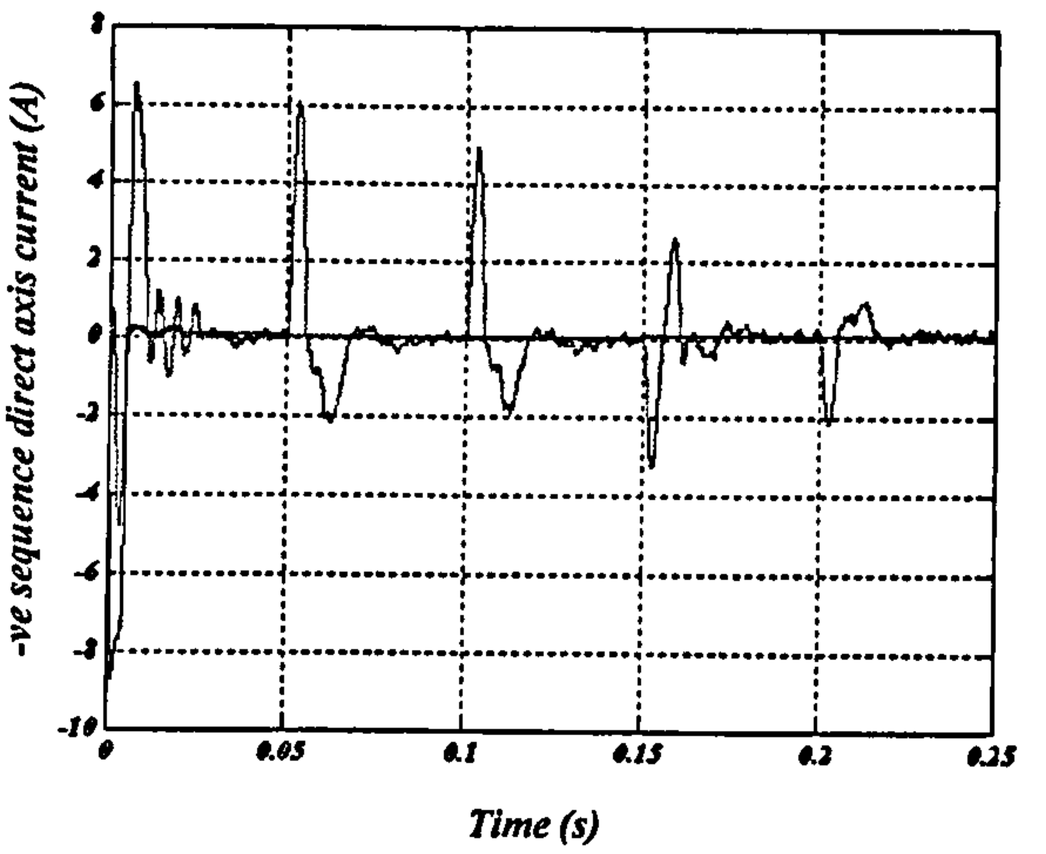


(a)

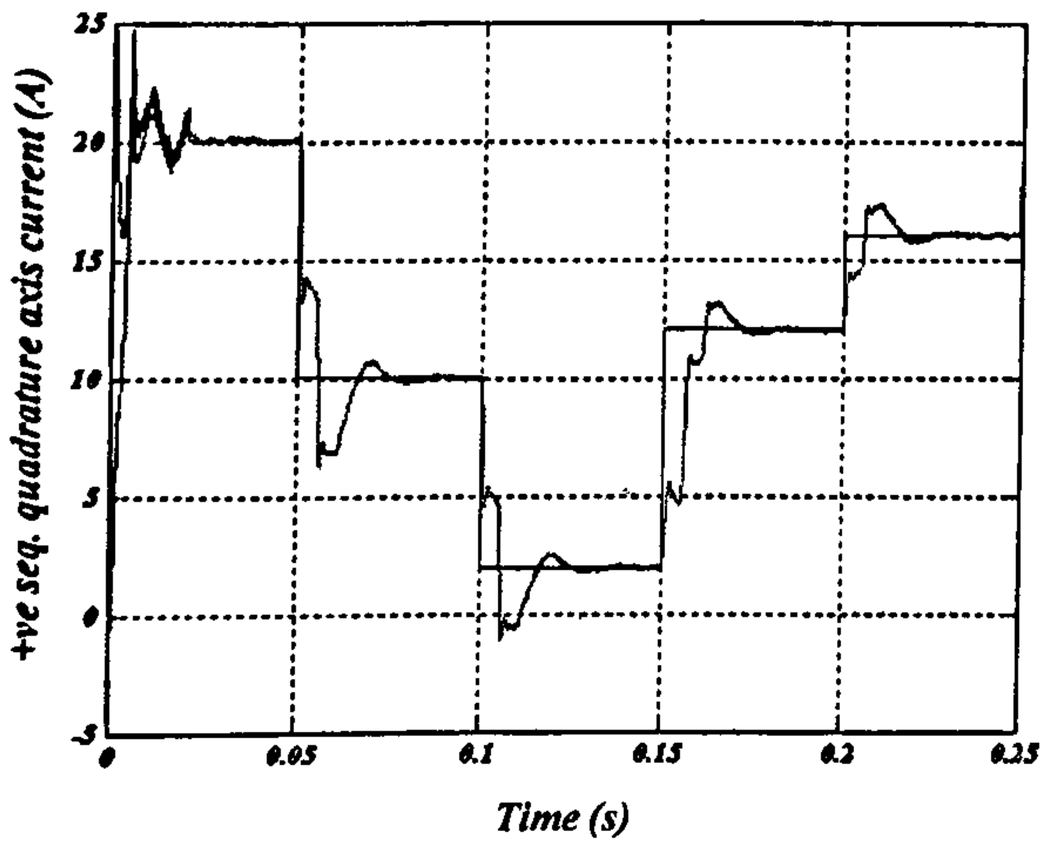


(b)

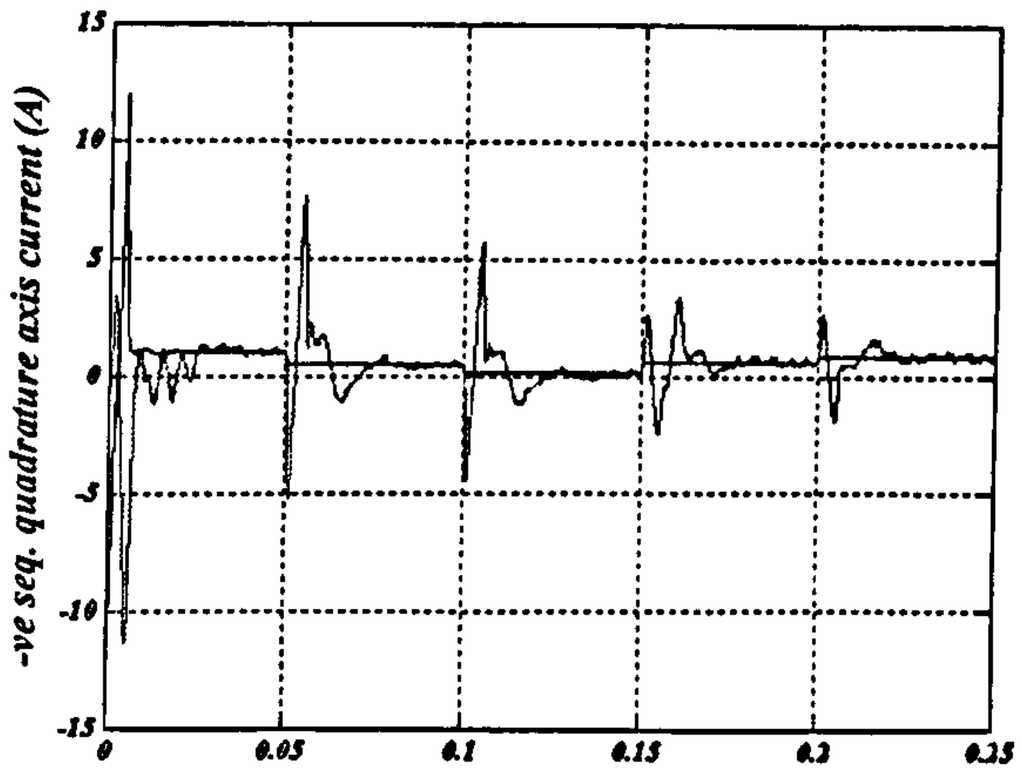




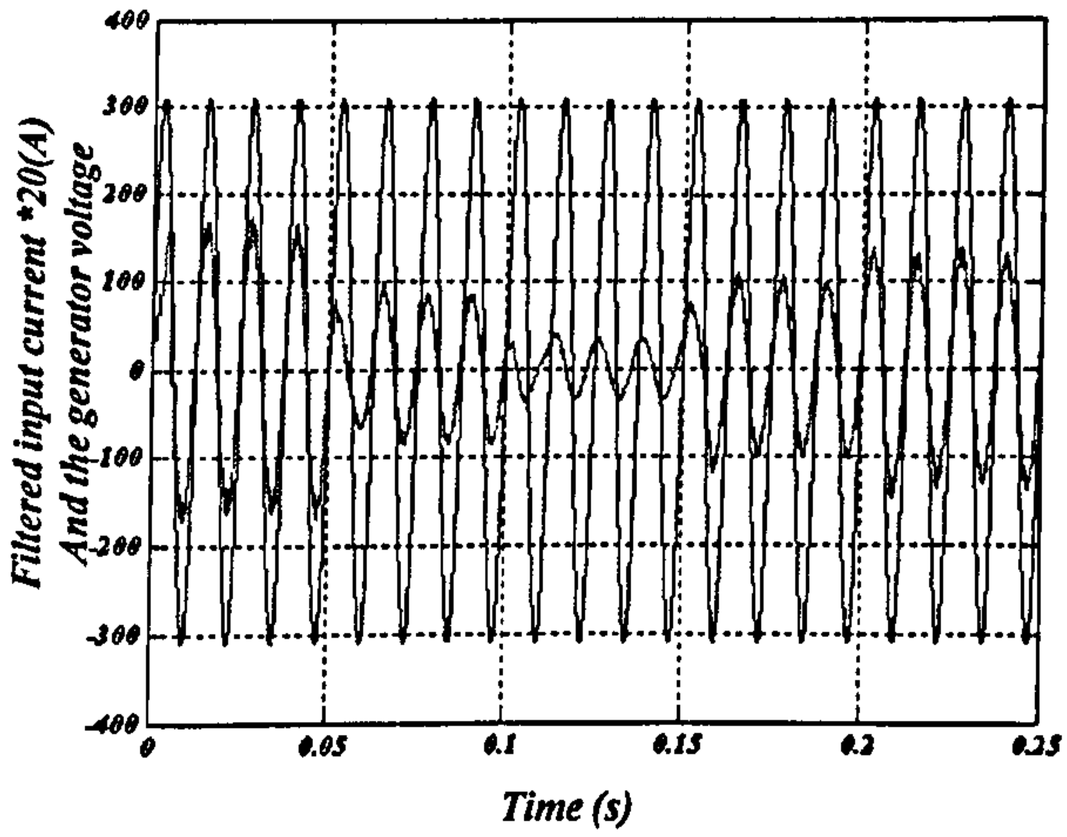
(c)



(d)



(e)



(f)

(g)

Fig. 5.23 Output response at 6.1 kHz and unbalanced input voltage: (a) output current, (b) output active power, (c) +ve seq. direct axis current, (d) -ve seq. direct axis current, (e) +ve seq. quadrature axis current, (f) -ve seq. quadrature axis current, and (g) filtered input current

*20 and generator voltage.

References:

[5.1] B. Pal, B. Chaudhuri, **'Robust Control in Power Systems'**, Springer (June 21, 2005), **ISBN-10:** 038725949X.

[5.2] H. J. Sira-Ramirez, R.-Ortigoza, **'Control Design Techniques in Power Electronics Devices'**, Springer Press (July 5, 2006), **ISBN-10:** 1846284589.

[5.3] D. A. Paice, **'Power Electronics Converter Harmonics: Multipulse Methods for Clean Power'**, Wiley-IEEE Press (September 15, 1999).

[5.4] E. Acha, *et al.* **'Power Electronic Control in Electrical Systems'**, Newnes (January 16, 2002), **ISBN-10:** 0780353943.

[5.5] M. H. Rashid, **'Power Electronics Handbook'**, Academic Press (August 24, 2001), **ISBN-10:** 0125816502.

[5.6] M. H. Rashid, **'SPICE for Power Electronics and Electric Power, Second Edition'**, CRC Press (November 02, 2005), **ISBN-10:** 0849334187.

[5.7] H. Lee Willis, **'Power Distribution Planning Reference Book'**, CRC (January 02, 2004), Second Edition, **ISBN-10:** 0824748751.

[5.8] L. Philipson, H. Lee Willis, **'Understanding Electric Utilities and De-Regulation'**, CRC (September 29, 2005), Second Edition, **ISBN-10:** 0824727738.

[5.9] W. Perruquetti, **'Sliding Mode Control in Engineering'**, Marcel Dekker (January 15, 2002), **ISBN-10:** 0824706714.

[5.10] Fernando D. Bianchi, Hernan De Battista, Ricardo J. Mantz, **'Wind Turbine Control Systems: Principles, Modelling and Gain Scheduling Design'**, Springer Press (July 10, 2006), **ISBN-10:** 1846284929.

[5.11] Tony Burton, David Sharpe, Nick Jenkins, Ervin Bossanyi, **'Wind Energy Handbook'**, John Wiley & Sons (November 15, 2001), **ISBN-10:** 0471489972.

[5.12] John Chiasson, **'Modeling and High Performance Control of Electric Machines'**, Wiley-IEEE Press (April 07, 2005), **ISBN-10:** 047168449X.

[5.13] Ion Boldea, **'Synchronous Generators (Electric Power Engineering Series)'**, CRC Press (November 09, 2005), **ISBN-10:** 084935725X.

[5.14] Chee-Mun Ong, **'Dynamic Simulations of Electric Machinery: Using MATLAB/SIMULINK'**, Prentice Hall PTR (September 19,1997), **ISBN-10:** 0137237855.

CHAPTER SIX

UNIFIED POWER FLOW CONTROLLER USING THE MATRIX CONVERTER

One of the main reasons for the USA 14th of August 2003 blackout was the increased loading of the transmission lines, which brought some transmission lines to operate close to their full capacities. FACTS devices can provide realistic solutions to reinforce transmission systems by better utilising the existing transmission network without the need for constructing new transmission lines. The unified power flow controller (UPFC) is a FACTS device that was developed for real-time control and dynamic compensation of ac transmission systems [6.1]-[6.4], providing multifunctional flexibility as explained in chapter 2.

The UPFC has control over all the parameters (voltage, impedance, and phase angle) affecting power flow on a transmission line. It has the ability to control the real and reactive power on the transmission line and the bus voltage simultaneously and independently. The control of the simplified UPFC using the matrix converter shown in fig. 6.1 is considered in this chapter.

6.1 Simplified UPFC Model

In fig. 6.1, the series converter (VSC2) is responsible for the UPFC main function of controlling the power flow on the transmission line by series injecting from a synchronous voltage source with controllable magnitude and angle. The shunt converter VSC1 provides both controllable shunt reactive power to control the bus voltage magnitude *l* and the real power requirements of VSC2.

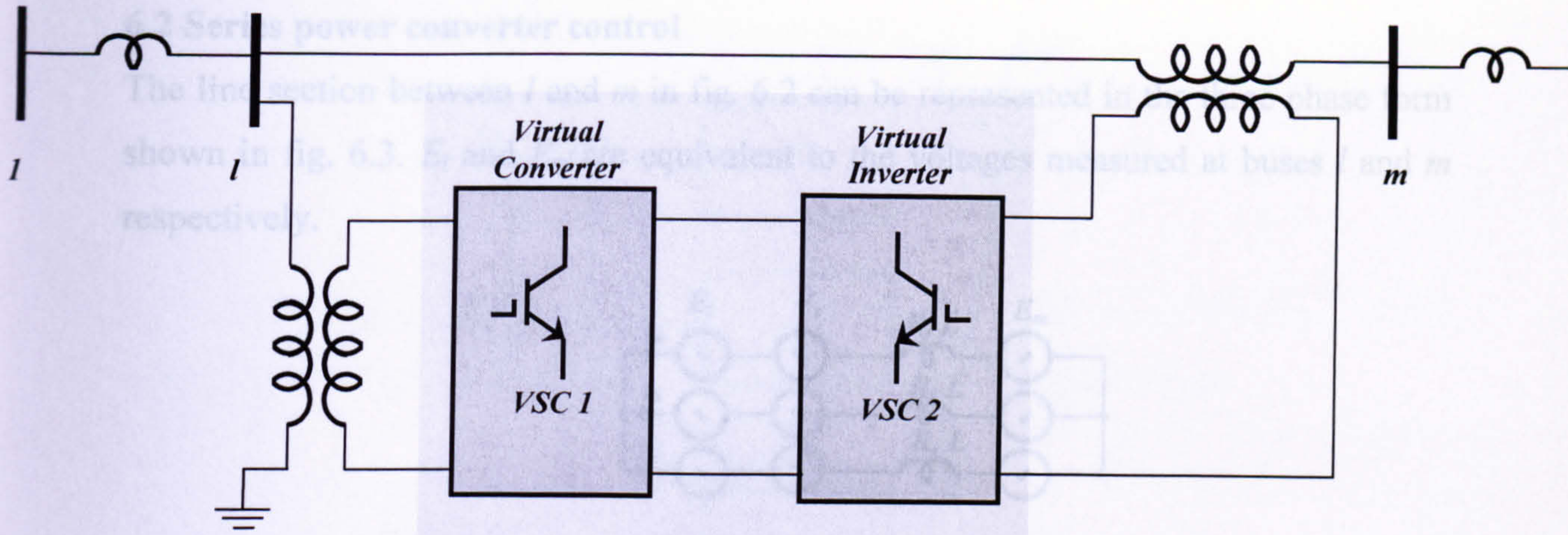


Fig 6.1 UPFC schematic employing the matrix converter.

The schematic model of a UPFC system is shown in Fig. 6.2. In this figure, I_{sh} and V_s represent the shunt current and series voltage of the UPFC respectively, and Z_s and Z_{sh} represent the impedances of their coupling transformers. Buses *l* and *m* are the sending and receiving buses.

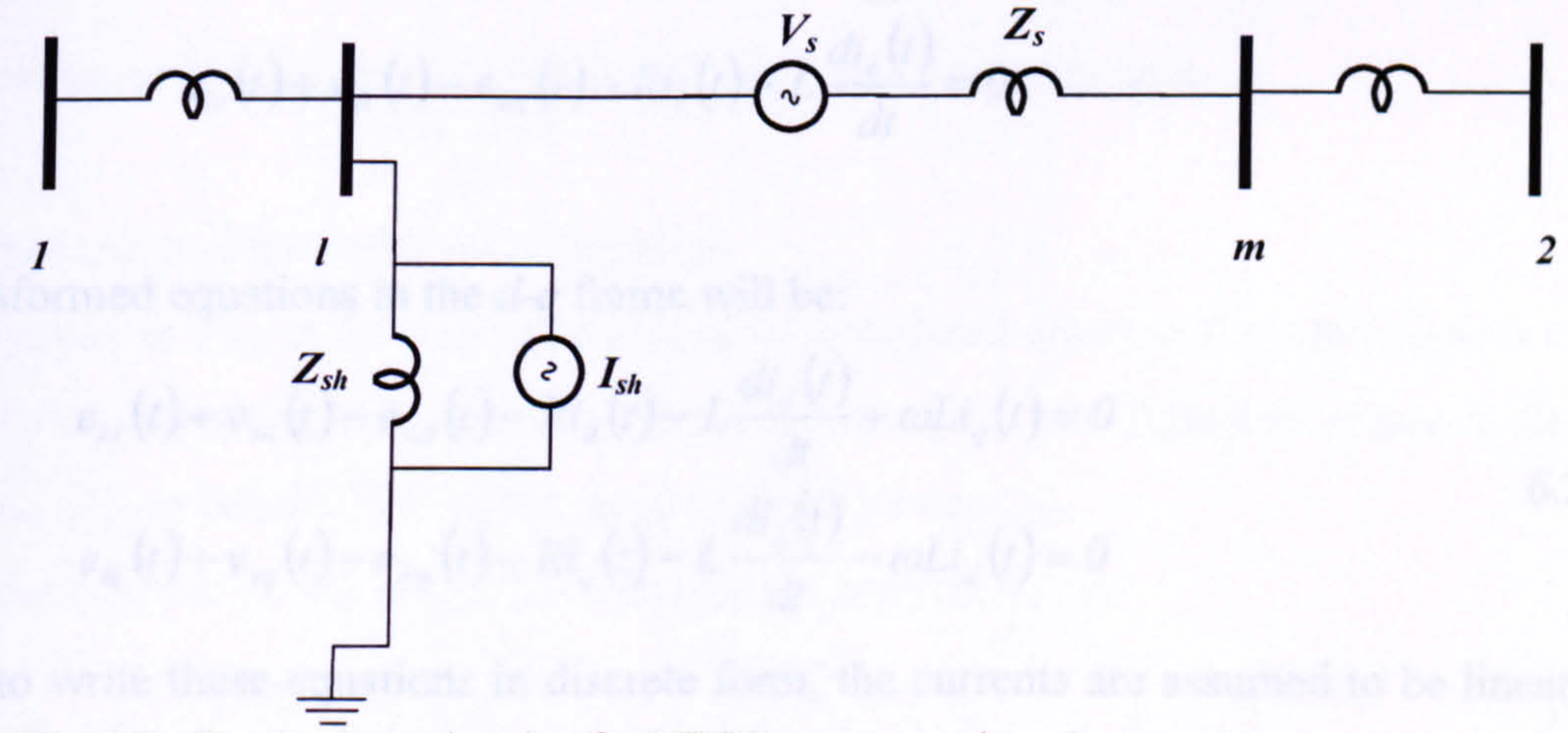


Fig 6.2 Equivalent circuit of a UPFC system using the matrix converter.

There are numerous functional control modes for the UPFC such as direct voltage injection, bus voltage regulation and control, line impedance compensation, phase angle regulation, and automatic power flow control. This chapter deals with the automatic power flow control mode of the series converter since this operating mode fully utilizes the capabilities of the UPFC, and the shunt converter is used in the voltage magnitude control mode.

6.2 Series power converter control

The line section between *l* and *m* in fig. 6.2 can be represented in the three phase form shown in fig. 6.3. E_l and E_m are equivalent to the voltages measured at buses *l* and *m* respectively.

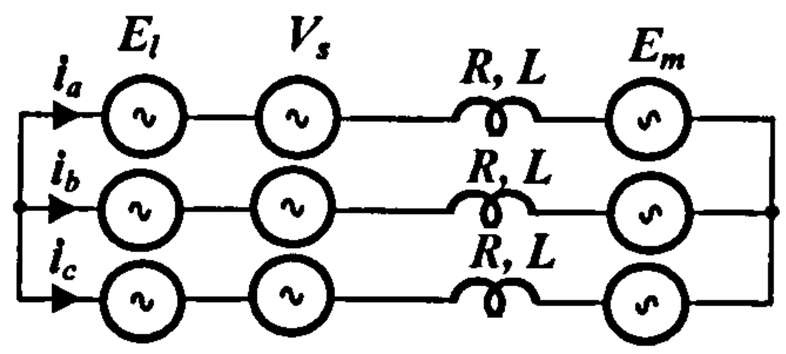


Fig 6.3 Inverter stage (series power converter).

Applying KVL to the output circuit gives

$$\begin{aligned} e_{la}(t)+v_{sa}(t)-e_{ma}(t)-Ri_a(t)-L\frac{di_a(t)}{dt}=0 \\ e_{lb}(t)+v_{sb}(t)-e_{mb}(t)-Ri_b(t)-L\frac{di_b(t)}{dt}=0 \\ e_{lc}(t)+v_{sc}(t)-e_{mc}(t)-Ri_c(t)-L\frac{di_c(t)}{dt}=0 \end{aligned} \qquad 6.1$$

The transformed equations in the *d-q* frame will be:

$$\begin{aligned} e_{ld}(t)+v_{sd}(t)-e_{md}(t)-Ri_d(t)-L\frac{di_d(t)}{dt}+\omega Li_q(t)=0 \\ e_{lq}(t)+v_{sq}(t)-e_{mq}(t)-Ri_q(t)-L\frac{di_q(t)}{dt}-\omega Li_d(t)=0 \end{aligned} \qquad 6.2$$

In order to write these equations in discrete form, the currents are assumed to be linear during one sample period [*k, k+1*]. Then using the same approximations as used in chapter 4, the *dq* voltages become:

$$\begin{aligned} v_{sd,k+1} = e_{md,k} - e_{ld,k} + R\sum_{n=0}^{k}\left(i^*_{d,n}-i_{d,n}\right)+\left(\frac{L}{T}+\frac{R}{2}\right).\left(i^*_{d,k}-i_{d,k}\right)-\omega L\left(\frac{i^*_{q,k}+i_{q,k}}{2}\right) \\ v_{sq,k+1} = e_{mq,k} - e_{lq,k} + R\sum_{n=0}^{k}\left(i^*_{q,n}-i_{q,n}\right)+\left(\frac{L}{T_s}+\frac{R}{2}\right).\left(i^*_{q,k}-i_{q,k}\right)+\omega L\left(\frac{i^*_{d,k}+i_{d,k}}{2}\right) \end{aligned} \qquad 6.3$$

The controller block diagram is shown in fig. 6.4

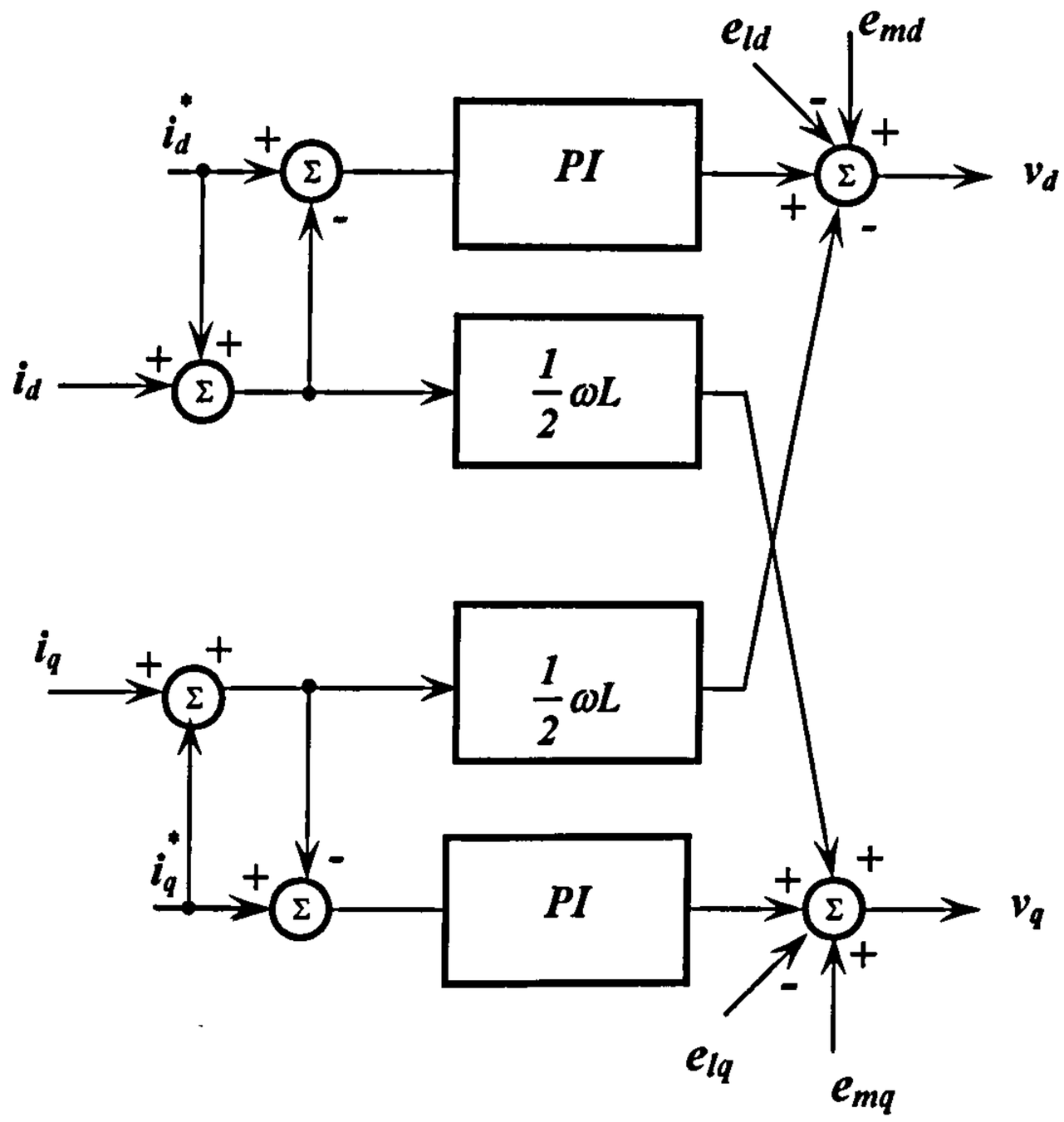


Fig 6.4 Current controller block diagram

The *d-q* voltages are then transformed to the equivalent *abc* voltages.

6.2.1 Generation of the reference currents

The active power $P_{o/p}$ at the converter output is considered equal to the active power at the input $P_{i/p}$, assuming zero losses in the converter switches. Reactive power at the output side $Q_{o/p}$ is independent of the input reactive power $Q_{i/p}$.

The apparent output power at the AC-side of the converter $S_{o/p}$ is:

$$S_{mo/p} = \vec{e}_{mdq} \cdot \vec{i}^{*}_{mdq} = \left(e_{md} i_{md} + e_{mq} i_{mq}\right) + j\left(e_{mq} i_{md} - e_{md} i_{mq}\right) \qquad 6.4$$

Separating the real and imaginary parts, (6.4) can be expressed in matrix form as:

$$\begin{bmatrix} P_{m\,o/p} \\ Q_{m\,o/p} \end{bmatrix} = \begin{bmatrix} e_{md} & e_{mq} \\ e_{mq} & -e_{md} \end{bmatrix} \begin{bmatrix} i^*_{md} \\ i^*_{mq} \end{bmatrix} \qquad 6.5$$

The current references are then calculated from:

$$\begin{bmatrix} i_{md}^{*} \\ i_{mq}^{*} \end{bmatrix} = \begin{bmatrix} e_{md} & e_{mq} \\ e_{mq} & -e_{md} \end{bmatrix}^{-1} \begin{bmatrix} P_{m\,o/p} \\ Q_{m\,o/p} \end{bmatrix} \tag{6.6}$$

Assuming the grid voltage has only a q component or in other words that the q axis is aligned with the grid voltage vector, the reference current signals become:

$$i_{md}^{*} = \frac{Q_{m\,o/p}}{e_{mq}} \tag{6.7}$$

$$i_{mq}^{*} = \frac{P_{m\,o/p}}{e_{mq}}$$

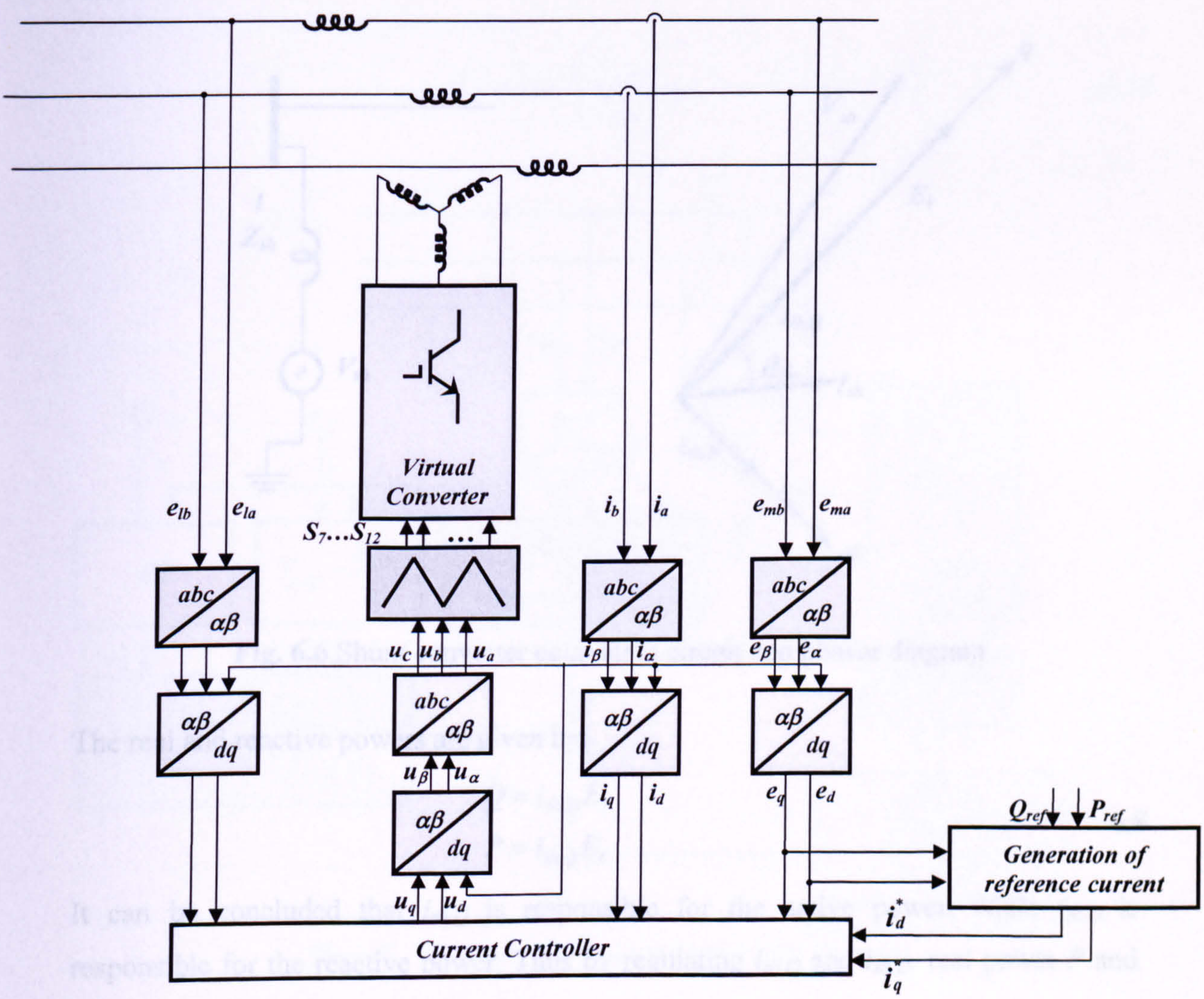


Fig 6.5 Series power converter block diagram

6.4 Shunt power converter control

The main function of the shunt converter is to control the magnitude of the voltage at bus *l*. This is achieved by adjusting the rectifier stage displacement angle. The shunt converter current can be divided into two parts:

- I_p which is in phase with the voltage at bus *l*. This means that it is responsible for producing the active power needed by the series converter.
- I_q is in quadrature to the phase voltage, so that it is responsible for the reactive power taken by the matrix converter. It should be noted that this value is independent of the controlled reactive power supplied by the series converter.

The equivalent circuit of the shunt converter and its phasor diagram are shown in fig. 6.6.

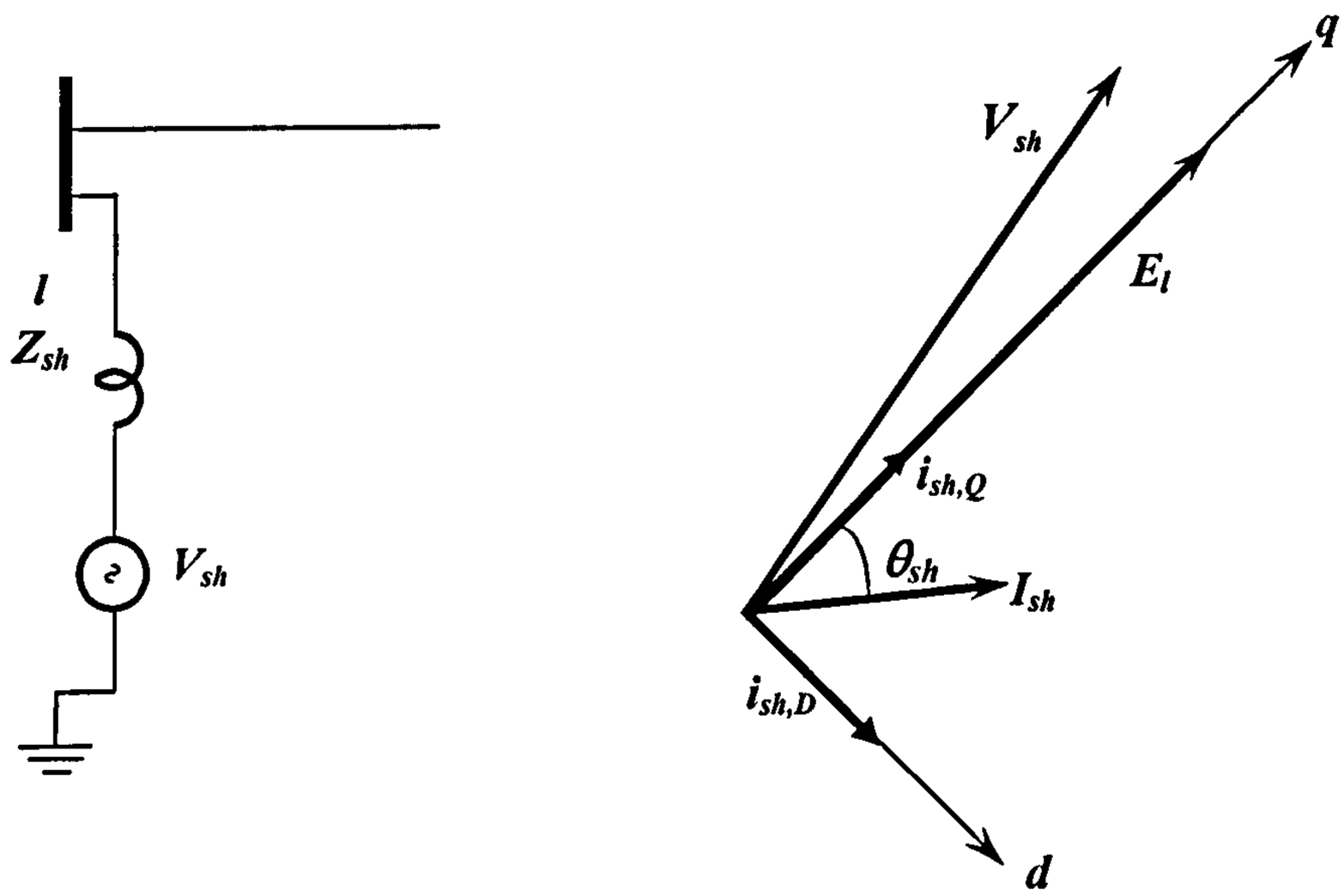


Fig. 6.6 Shunt converter equivalent circuit and phasor diagram

The real and reactive powers are given by:

$$Q = i_{sh,D} E_l$$

$$P = i_{sh,Q} E_l \qquad 6.8$$

It can be concluded that $i_{sh,Q}$ is responsible for the active power, while $i_{sh,D}$ is responsible for the reactive power. Thus by regulating $i_{sh,Q}$ and $i_{sh,D}$, real power P and reactive power Q to/from the shunt inverter can be controlled independently. The

active power is the same as that of the series converter, so the only controllable variable is the reactive power component.

The quadrature axis reference current can be estimated by using the output active power and the voltage magnitude at bus bar *m*, while the direct axis reference current is calculated using a PI controller in order to control the voltage magnitude at bus *l*, as given by:

$$i^*_{sh,D} = k_p\left(E^*_l - E_l\right) + k_i \sum\left(E^*_l - E_l\right) \qquad 6.9$$

The input displacement angle is estimated as:

$$\theta^*_{sh} = tan^{-1}\left(\frac{i^*_{sh,D}}{i^*_{sh,Q}}\right) \qquad 6.10$$

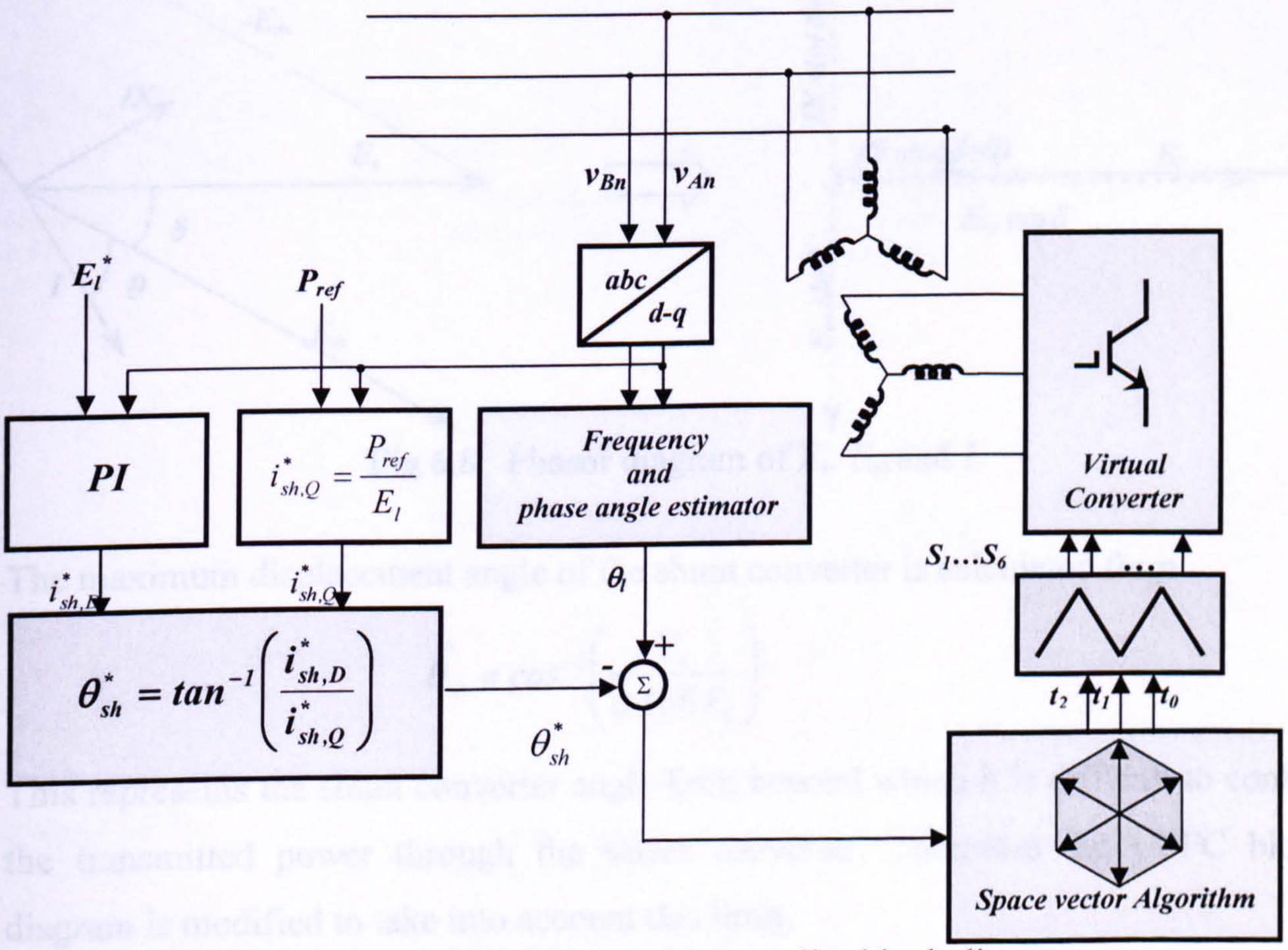


Fig 6.7 Shunt converter controller block diagram

The virtual dc link average voltage is given by

$$v_{DC} = \hat{V}_{iL} . \frac{\sqrt{3}}{2} cos\,\theta_{sh} . \frac{T_{s\,eff}}{T_s} \qquad 6.11$$

This means that the virtual dc link voltage is affected by the input displacement angle θ_{sh}, hence the output voltage maximum transfer ratio is also affected. This reduces the capability of the series converter to control the active and reactive power flowing through the line.

From fig. 6.8, the series voltage required can be estimated from

$$\bar{V}_s = \bar{E}_l - \bar{E}_m - \bar{I}\,\bar{Z} \approx \bar{E}_l - \bar{E}_m - j\,\bar{I}\,X \qquad 6.12$$

and its magnitude is calculated from

$$V_s \approx \sqrt{[E_l - E_m\,cos\,\delta - IX\,sin(\delta+\theta)]^2 + [E_m\,sin\,\delta - IX\,cos(\delta+\theta)]^2} \qquad 6.13$$

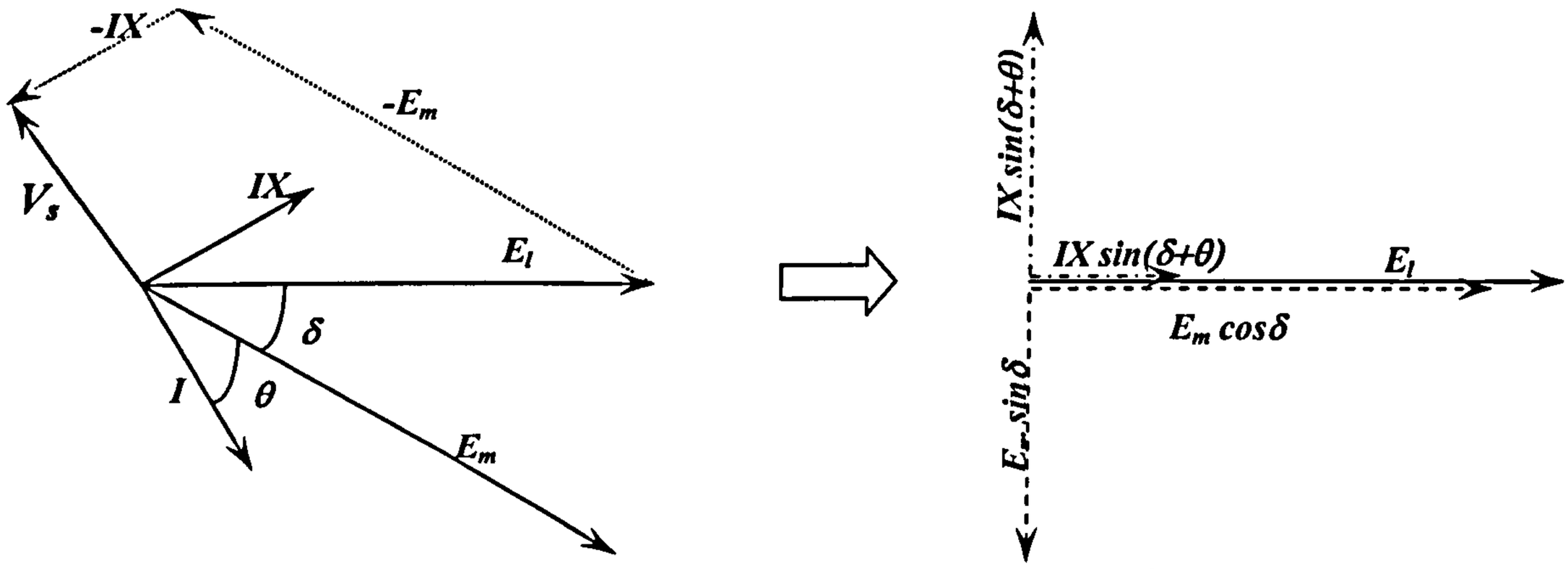
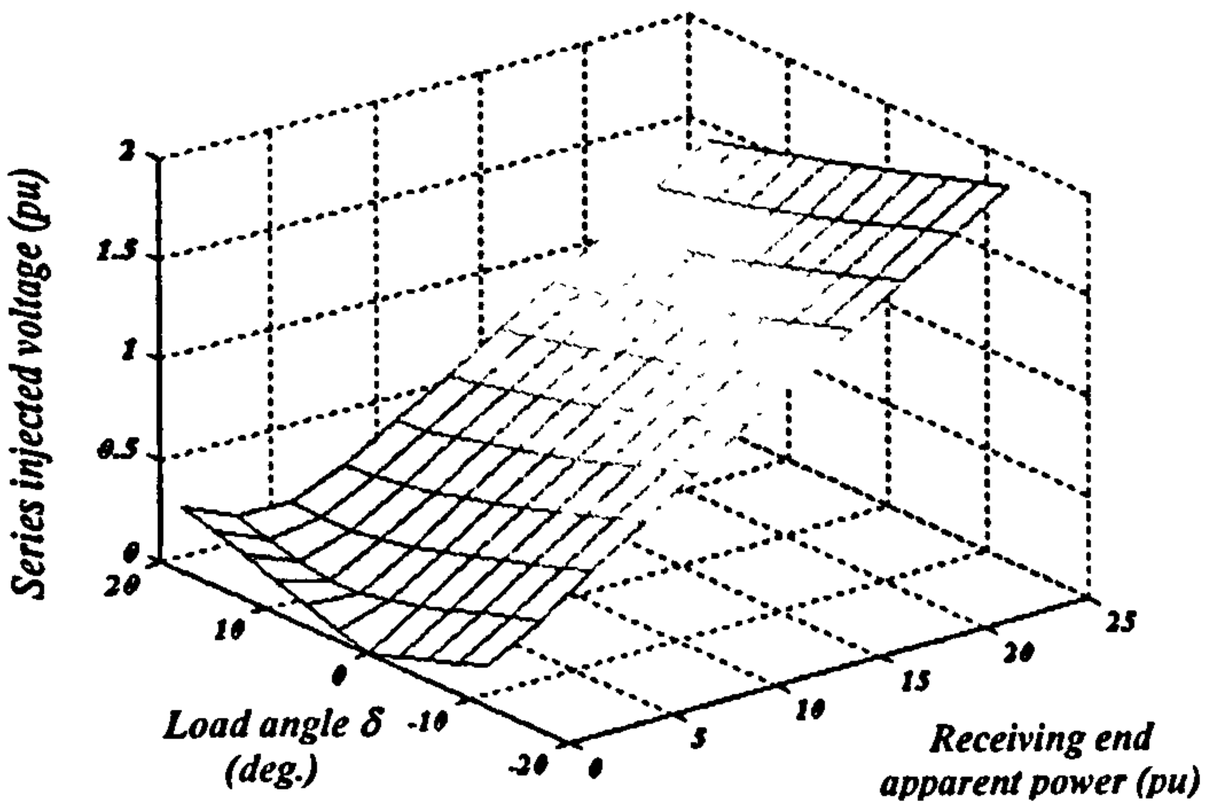


Fig 6.8 Phasor diagram of E_l, E_m and I.

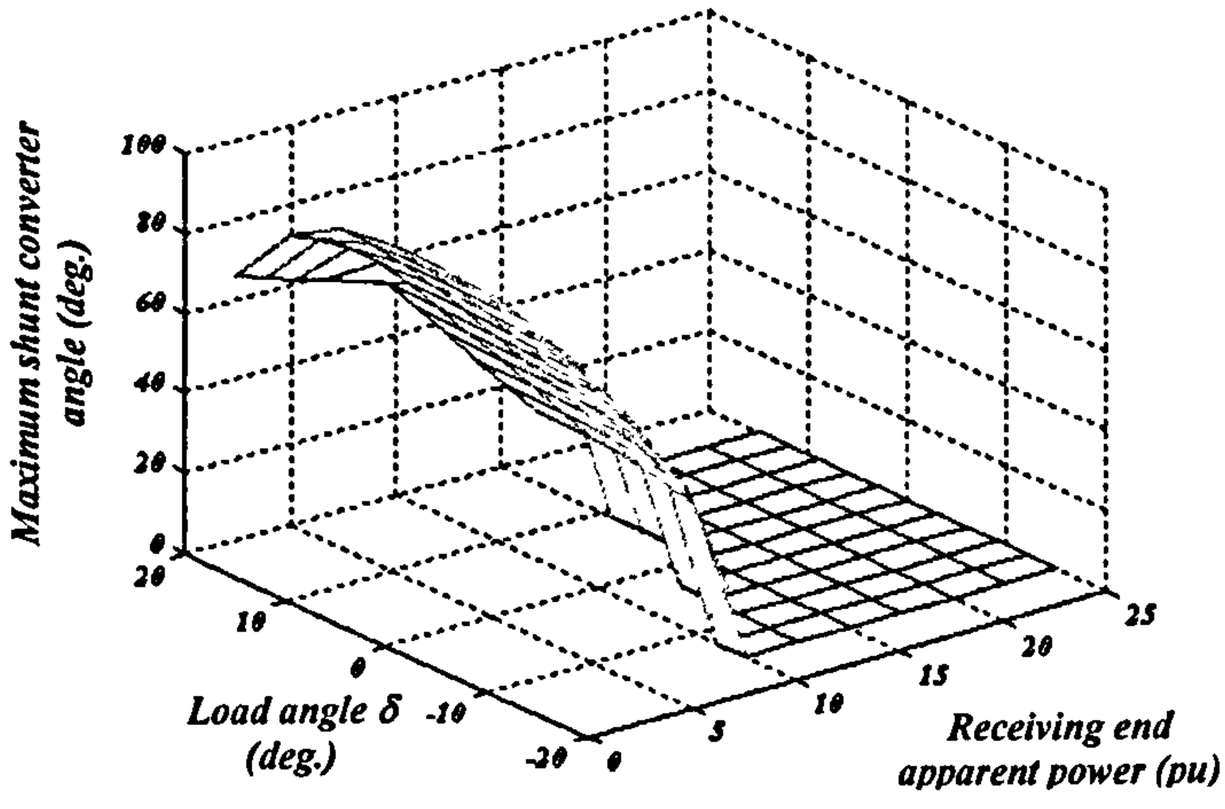
The maximum displacement angle of the shunt converter is calculated from

$$\hat{\theta}_{sh} = cos^{-1}\left(\frac{V_s}{0.866\,E_l}\right) \qquad 6.14$$

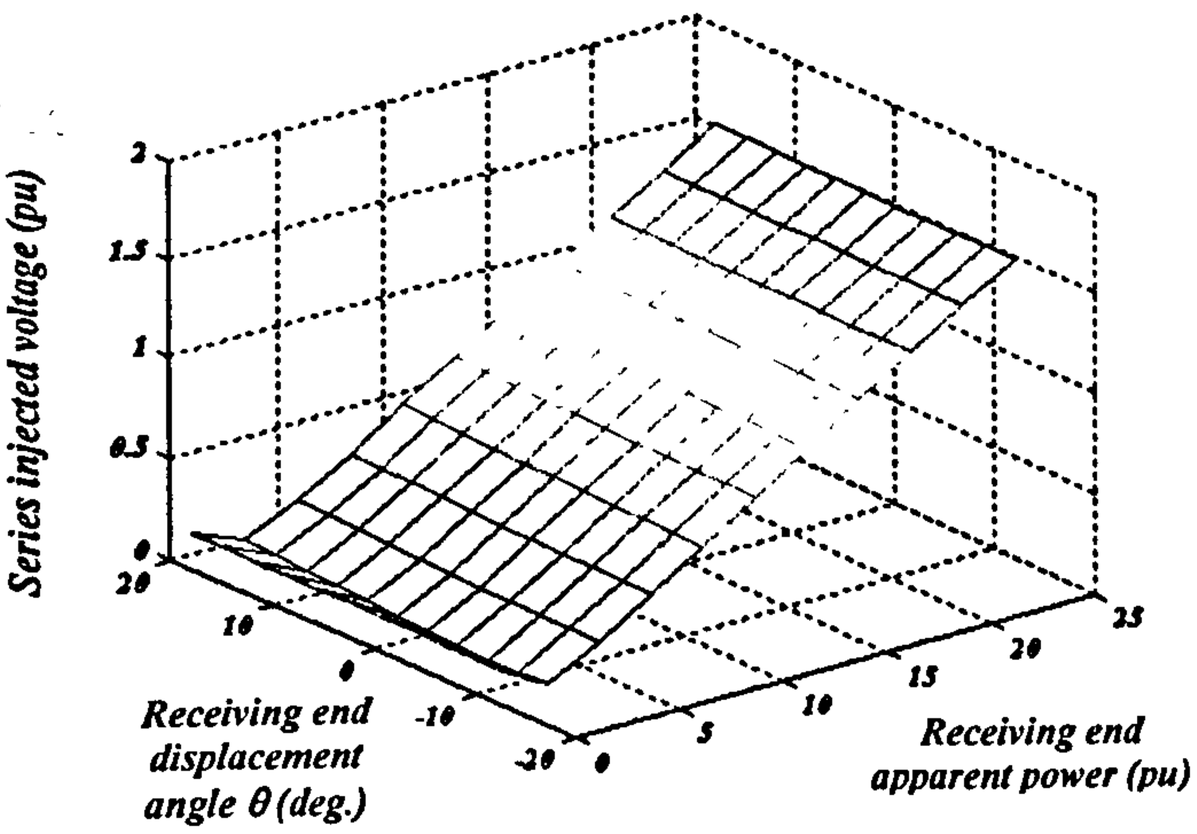
This represents the shunt converter angle limit beyond which it is difficult to control the transmitted power through the series converter. Therefore the UPFC block diagram is modified to take into account this limit.


Series injected voltage (pu)
2
1.5
1
0.5
0
20
10
0
-10
-20
Load angle δ (deg.)
0
5
10
15
20
25
Receiving end apparent power (pu)

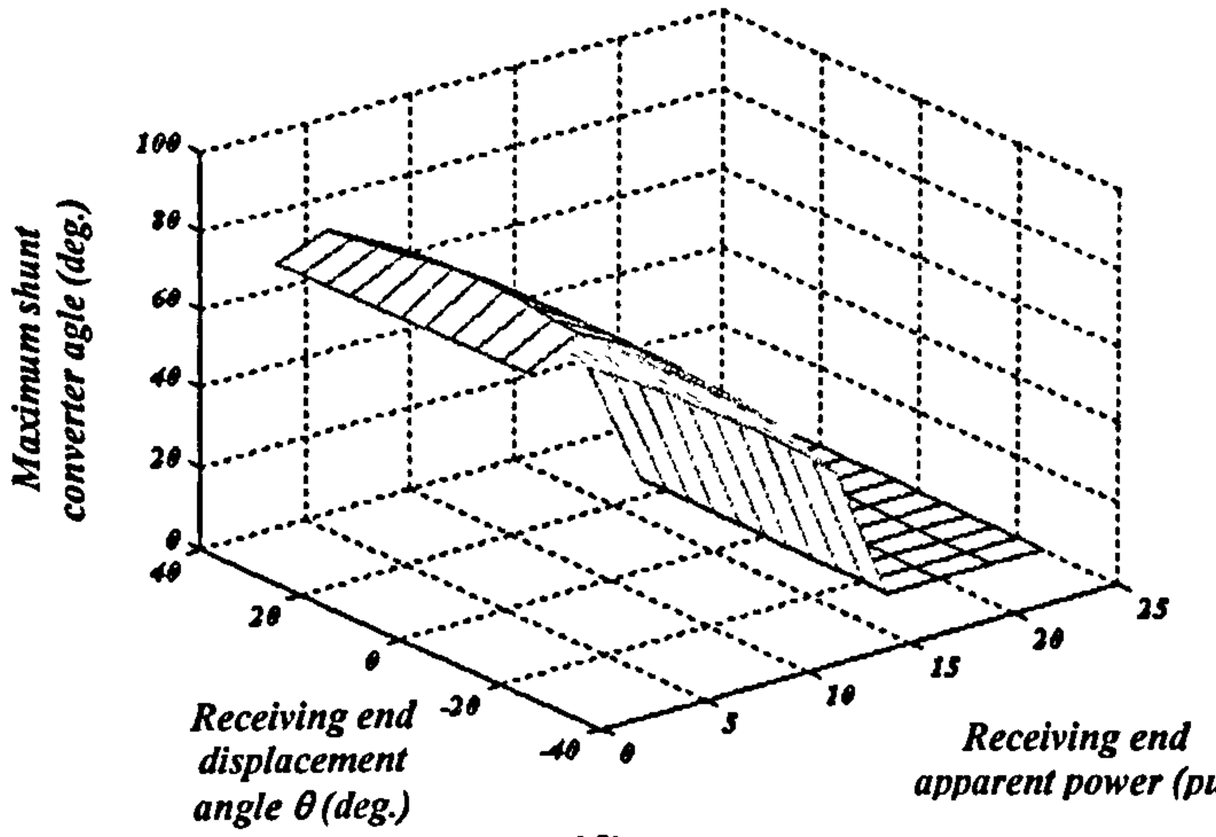
(a)


Maximum shunt converter angle (deg.)
100
80
60
40
20
0
20
10
0
-10
-20
Load angle δ (deg.)
0
5
10
15
20
25
Receiving end apparent power (pu)

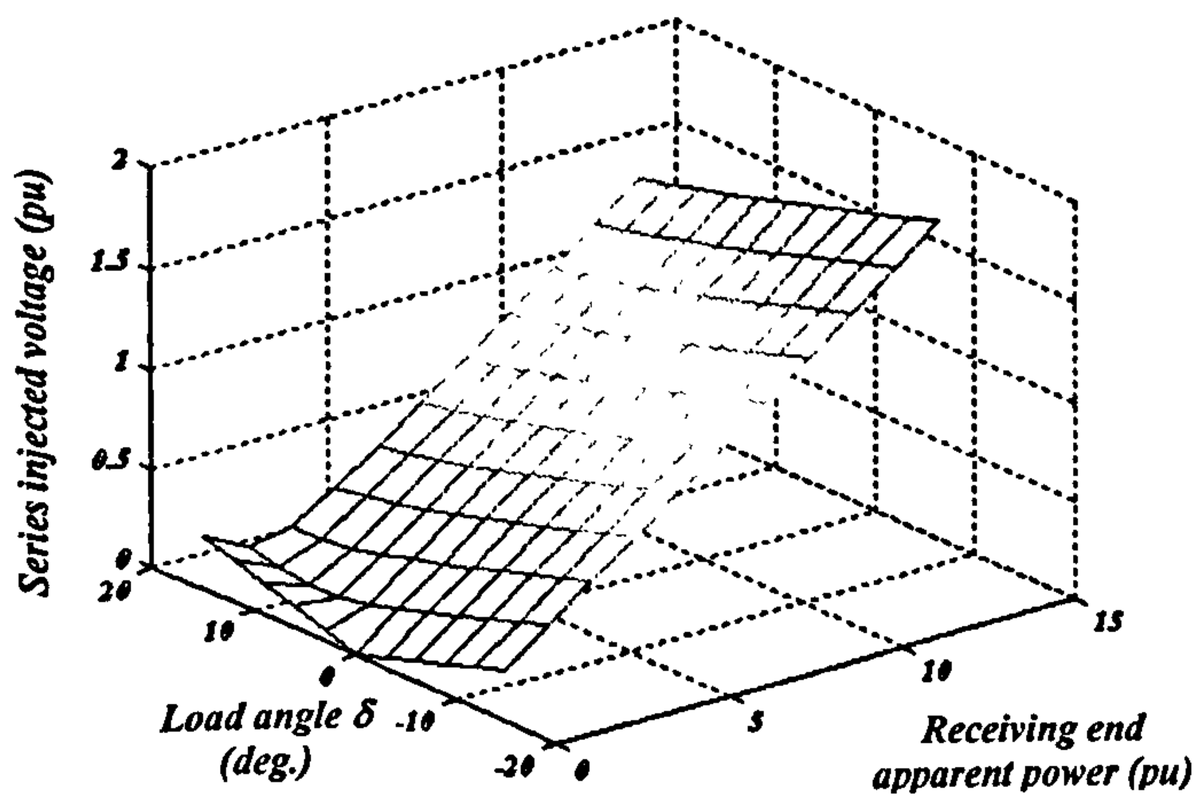
(b)


Series injected voltage (pu)
2
1.5
1
0.5
0
20
10
0
-10
-20
Receiving end displacement angle θ (deg.)
0
5
10
15
20
25
Receiving end apparent power (pu)

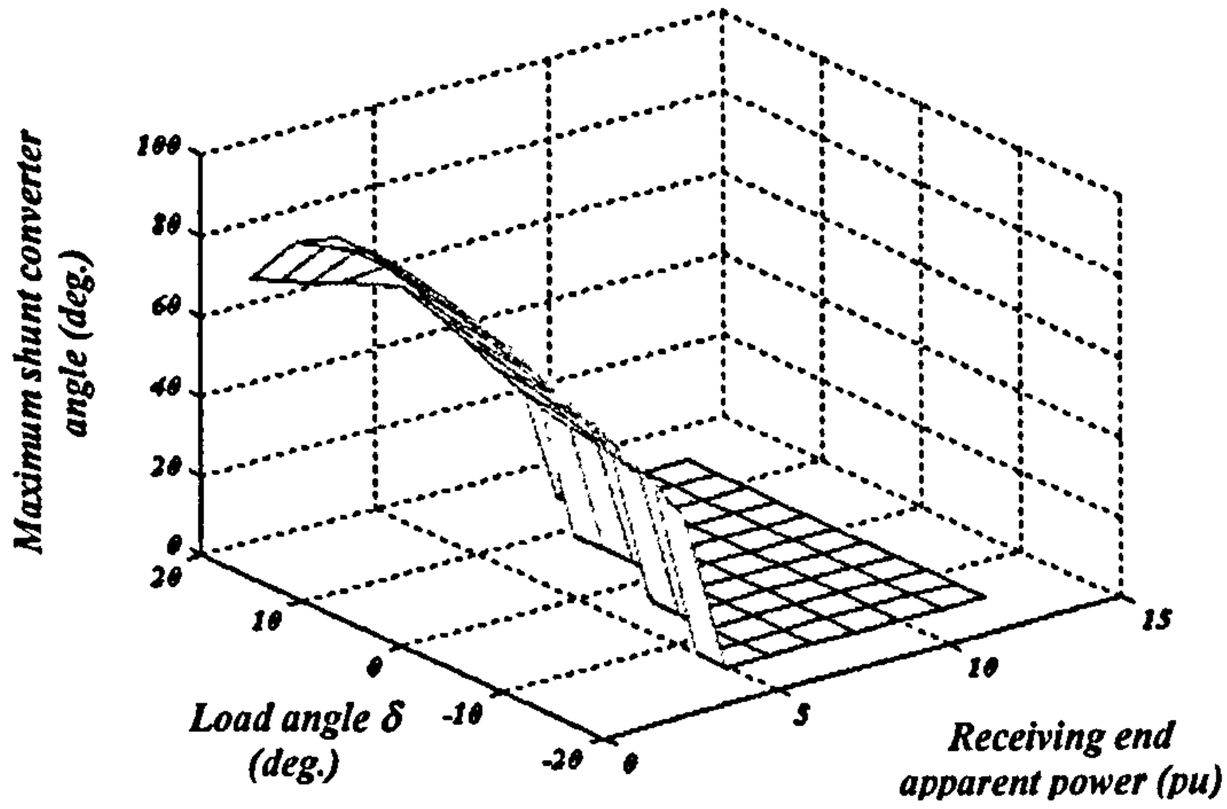
(c)


Maximum shunt converter agle (deg.)
100
80
60
40
20
0
40
20
0
-20
-40
Receiving end displacement angle θ (deg.)
0
5
10
15
20
25
Receiving end apparent power (pu)

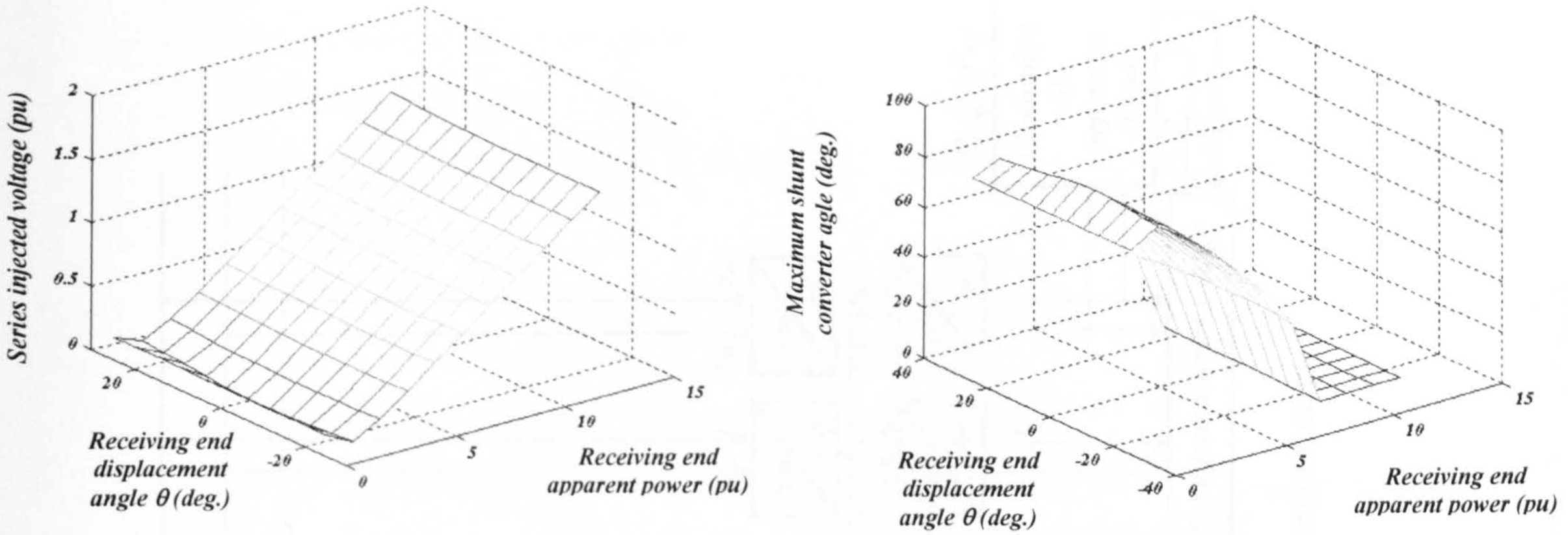
(d)


Series injected voltage (pu)
2
1.5
1
0.5
0
20
10
0
-10
-20
Load angle δ (deg.)
0
5
10
15
Receiving end apparent power (pu)

(e)


Maximum shunt converter angle (deg.)
100
80
60
40
20
0
20
10
0
-10
-20
Load angle δ (deg.)
0
5
10
15
Receiving end apparent power (pu)

(f)



(g) **(h)**

Fig 6.9 Different cases of series injected voltage and the corresponding maximum shunt converter angle: (a-d) x=0.2 pu, (e-h) x=0.4 pu

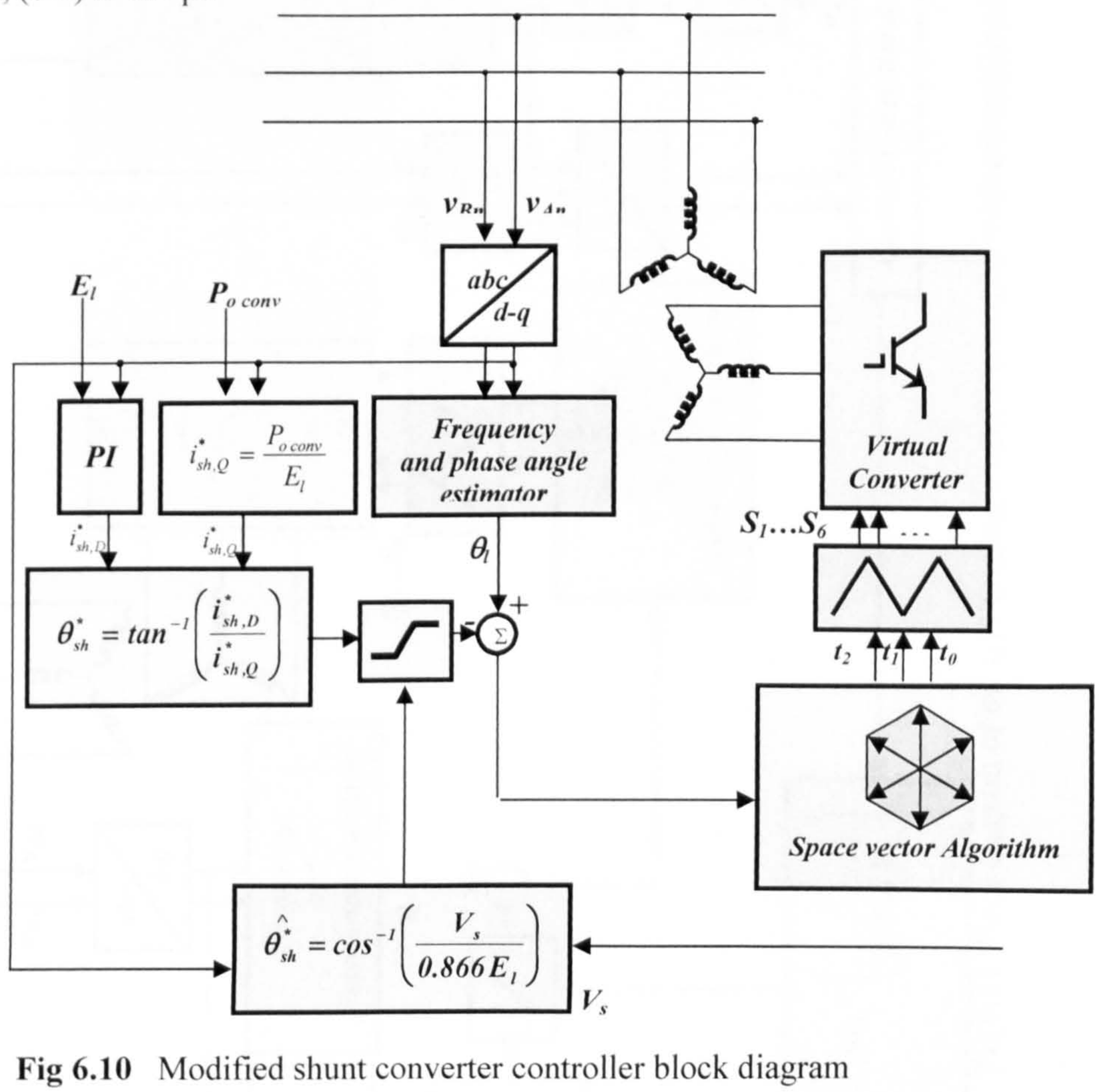


Fig 6.10 Modified shunt converter controller block diagram

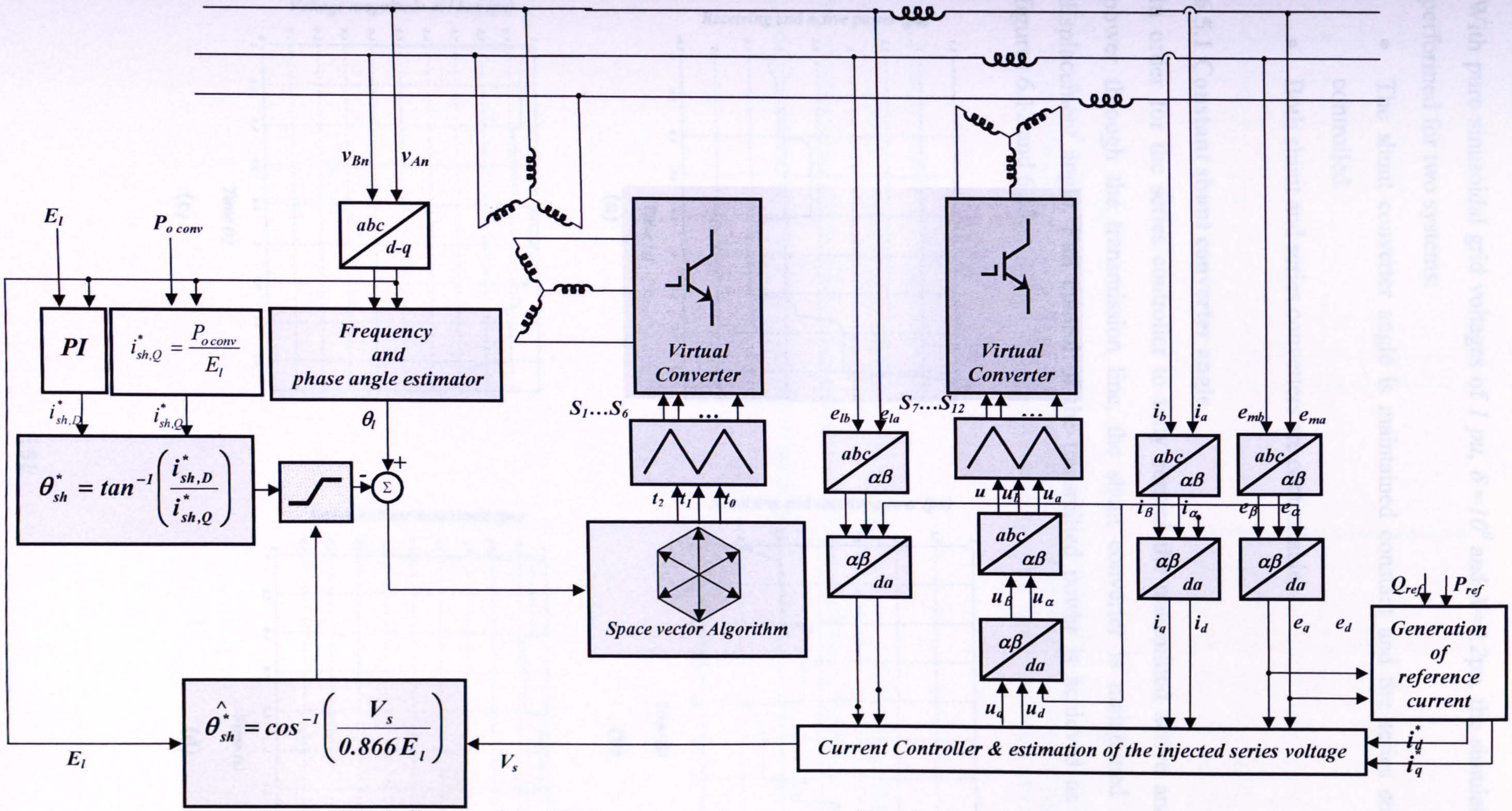


Fig 6.11 Block diagram of the UPFC using a matrix converter with estimation of the injected series voltage.

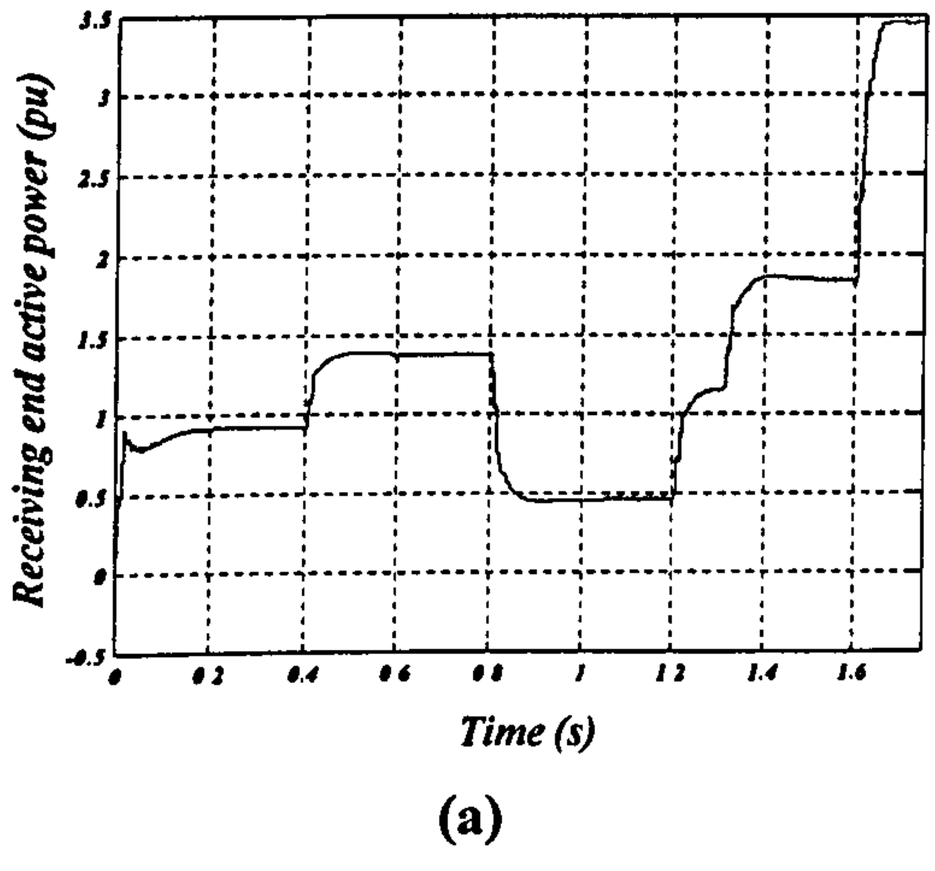
6.5 Simulation results

With pure sinusoidal grid voltages of *1 pu*, $\delta = 10^0$ and $X = 0.2$pu, the simulations were performed for two systems:

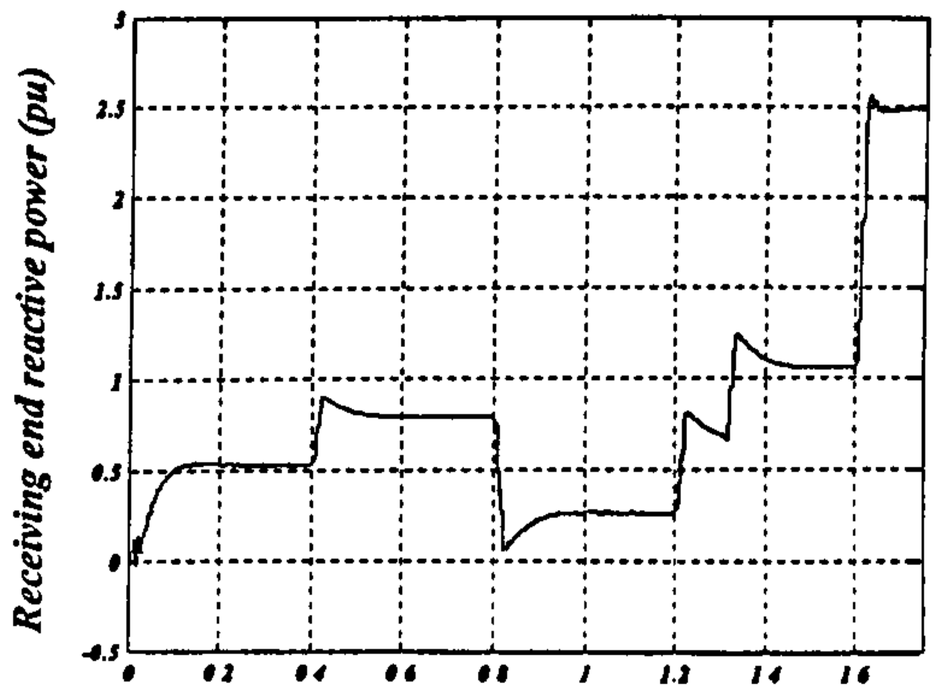
- The shunt converter angle is maintained constant and the series converter is controlled.
- Both shunt and series converters are controllable.

6.5.1 Constant shunt converter angle

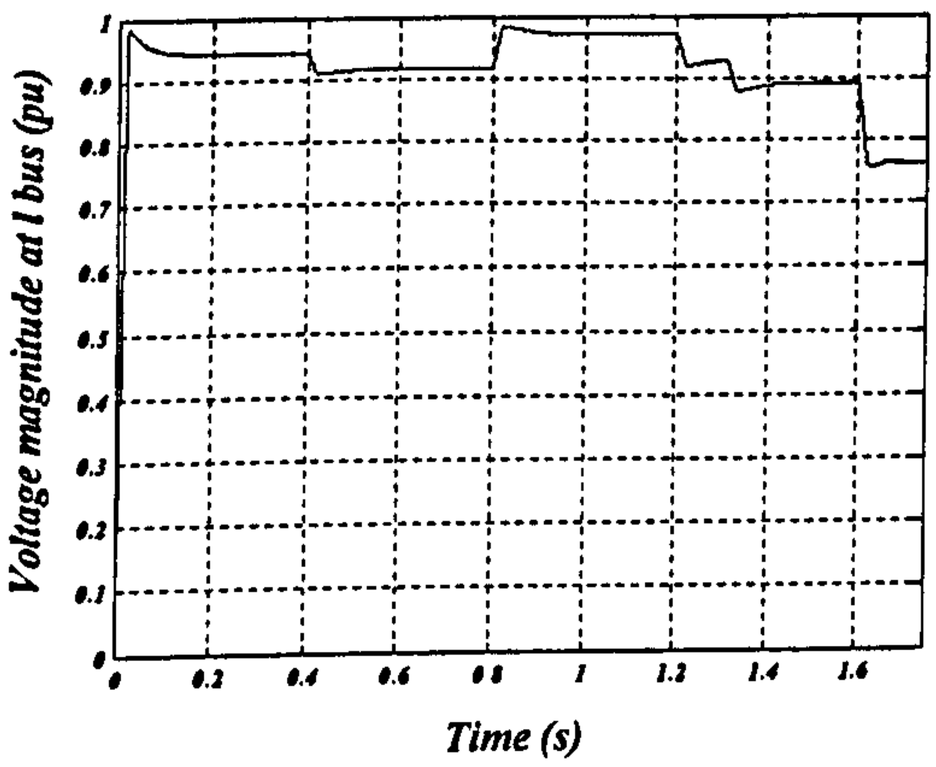
In order for the series controller to fully control the transmitted active and reactive power through the transmission line, the shunt converter is maintained at a zero displacement angle. Full control of the transmitted power is achieved as shown in figures 6.12 and 6.13.



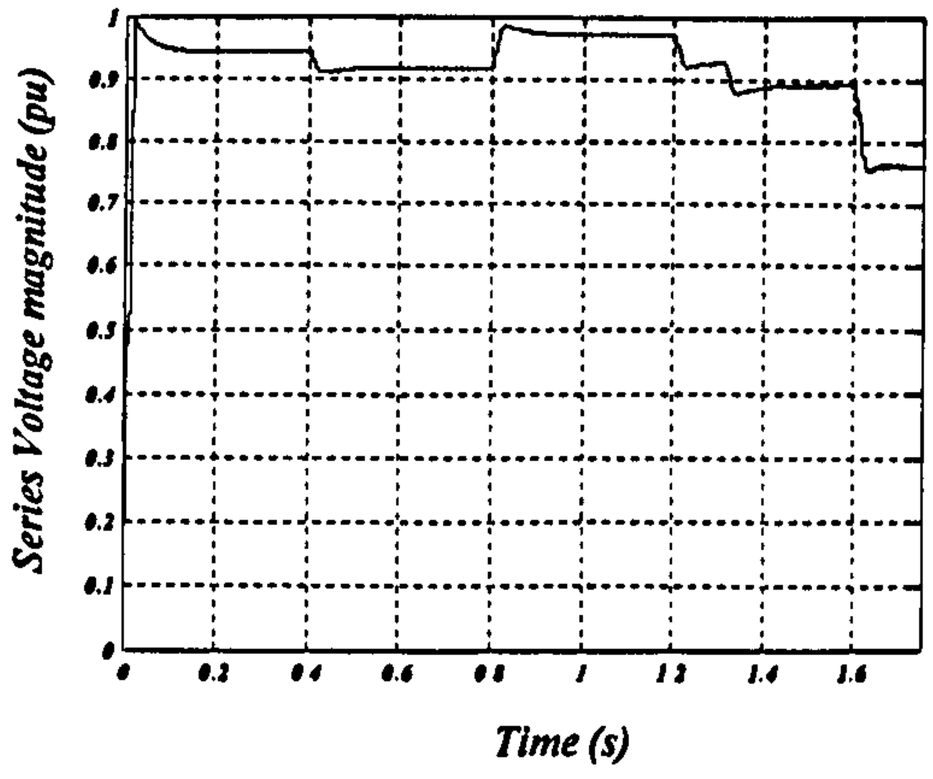
(a)



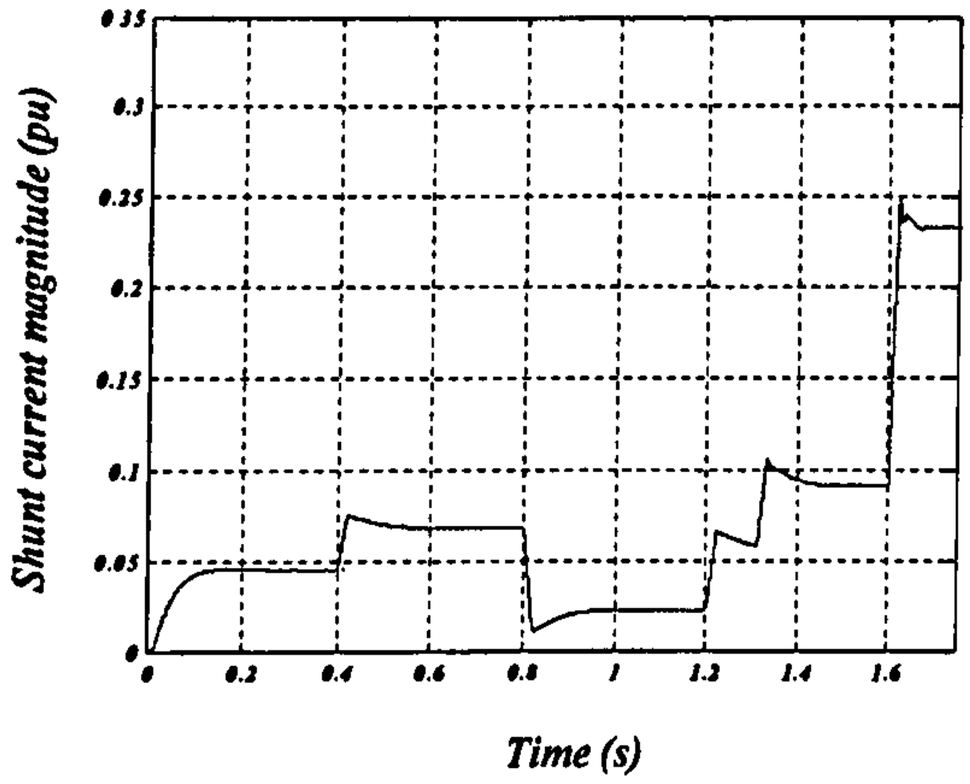
(b)

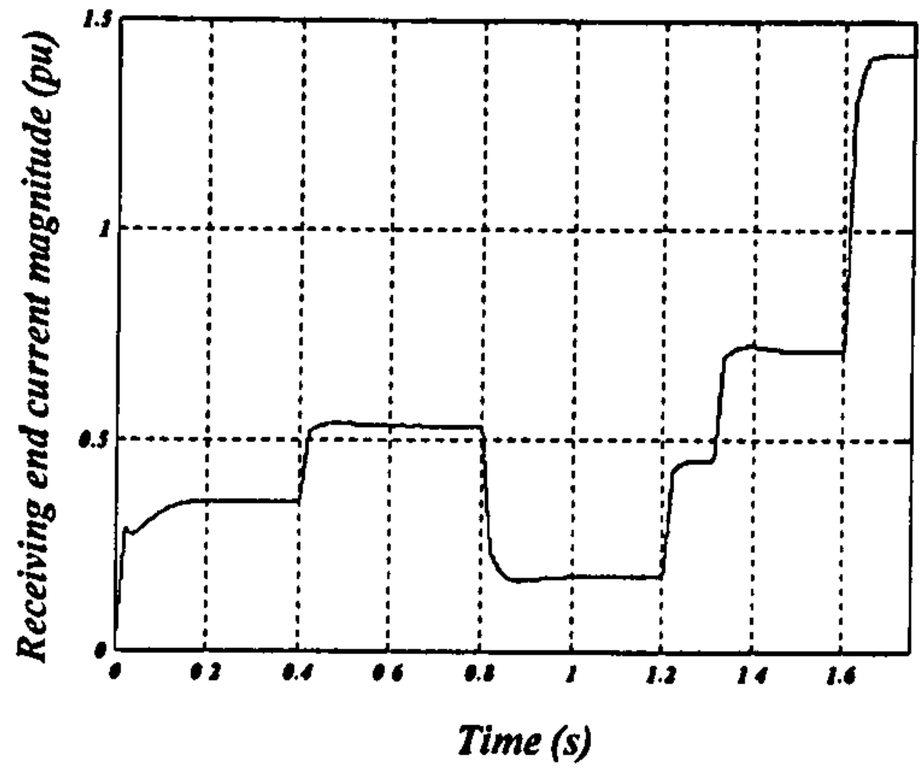


(c)



(d)



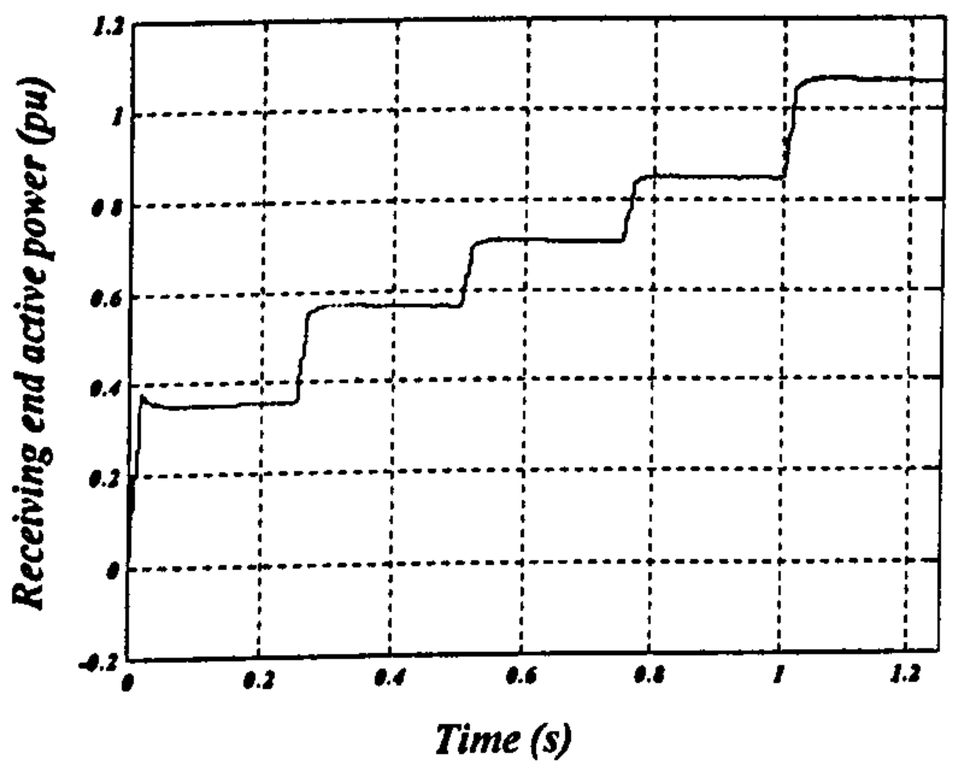


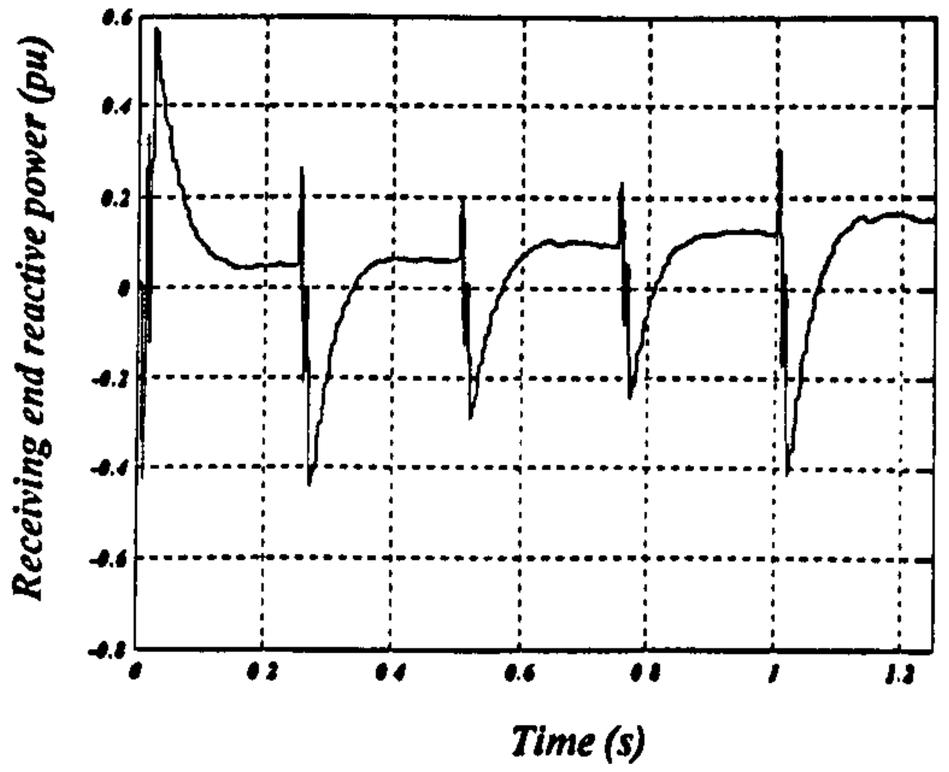
(e) (f)

Fig. 6.12 Simulated output response for a UPFC using a matrix converter at a 6.1 kHz switching frequency: (a) receiving end active power, (b) receiving end reactive power, (c) voltage magnitude at l bus, (d) series voltage magnitude, (e) shunt current magnitude, and (f) receiving end current magnitude.

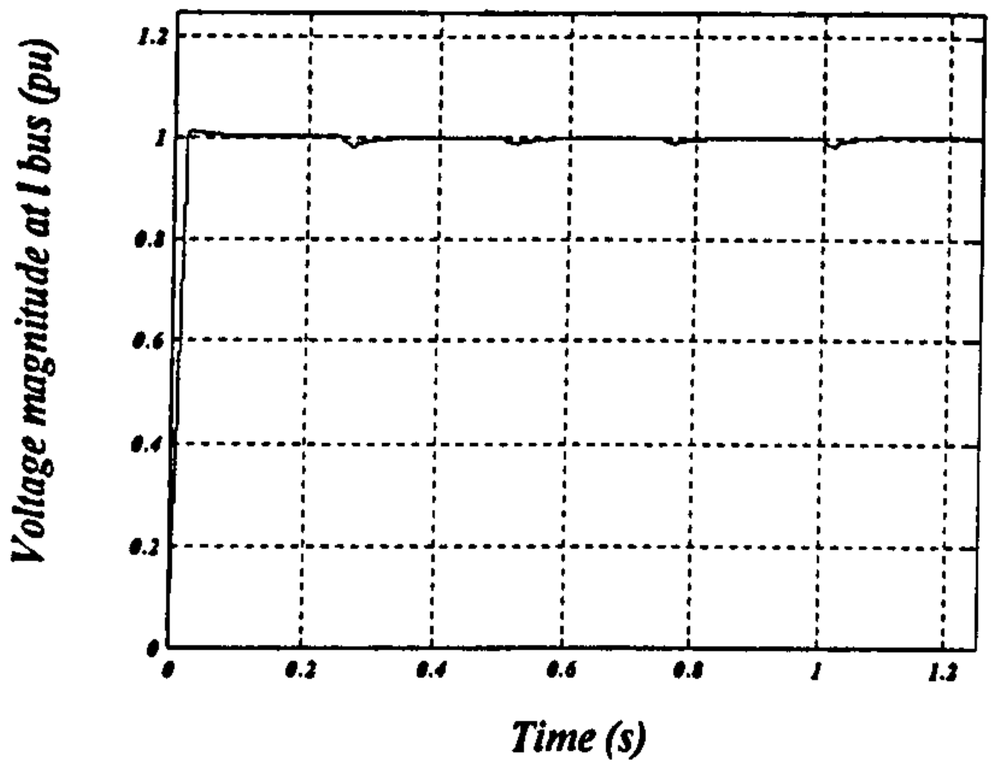
6.5.2 Shunt converter angle control

If it is required to control both the voltage magnitude at bus *l* and the transmitted power, the modified algorithm which takes into account the limitations of the shunt converter is used. The voltage can be controlled with a small shunt current as shown in fig. 6.13.

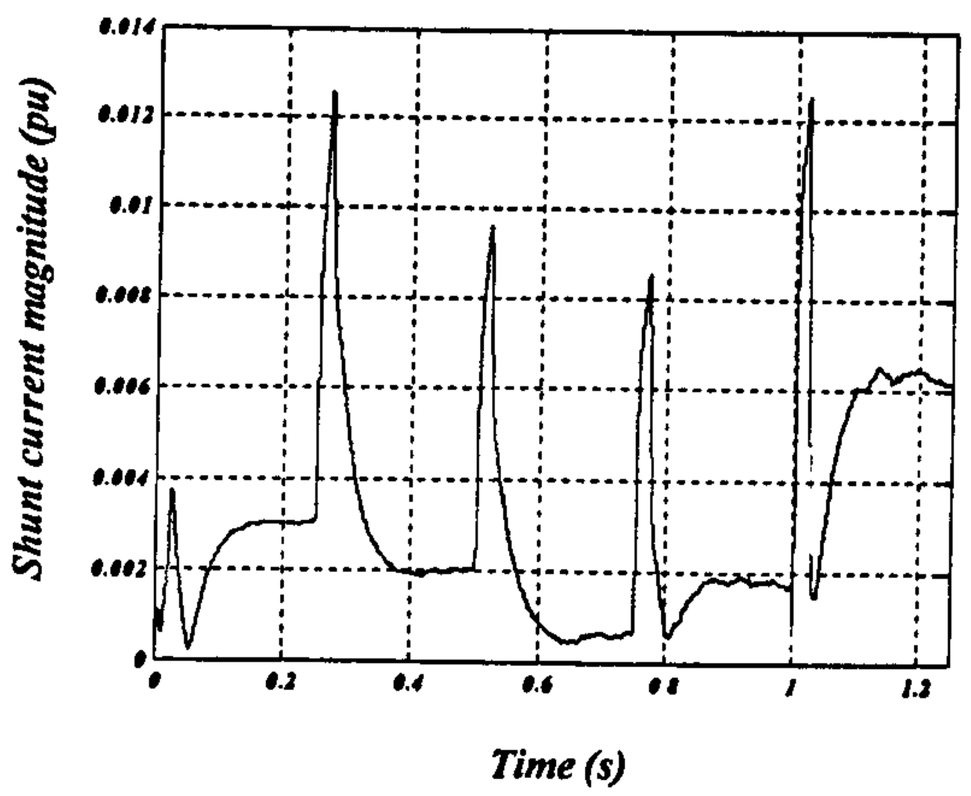




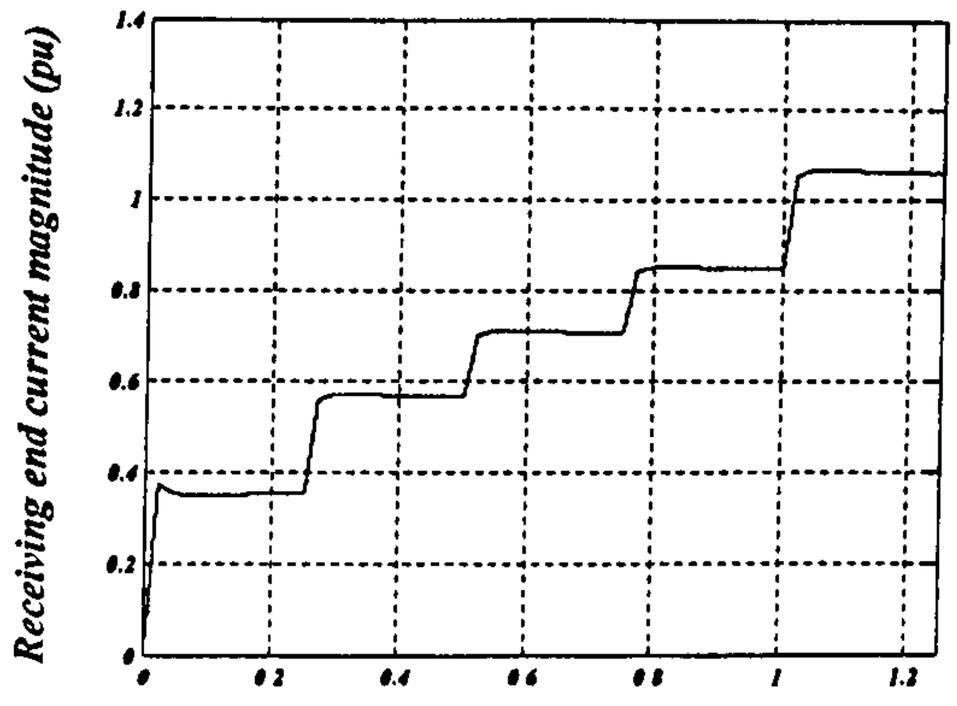
(a) (b)



(c)



(d)



(e)

Fig. 6.13 Simulated output response for a UPFC using a matrix converter at a 6.1 kHz switching frequency: (a) receiving end active power, (b) receiving end reactive power, (c) voltage magnitude at I bus, (d) shunt current magnitude, and (f) receiving end current magnitude.

It can be concluded that full control of the transmitted power can be achieved, while full control of the bus bar voltage is limited to certain values of the transmitted power.

References:

[6.1] V. K. Sood, **'HVDC and FACTS Controllers: Applications of Static Converters in Power Systems'**, Kluwer Academic Publishers (April 1, 2004), **ISBN-10:** 1402078900.

[6.2] E. Acha, **'FACTS: Modelling and Simulation in Power Networks'**, John Wiley and Sons Ltd (February 17, 2004), **ISBN-10:** 0470852712.

[6.3] B. Ooi, M. Kazerani, **'Unified power flow controller based on matrix**

converter', Power Electronics Specialists Conference, 1996. PESC '96 Record', 27th Annual IEEE, Volume: 1, 23-27 June 1996, 502 – 507.

[6.4] Strzelecki R., *et al.* **'UPFC with matrix converter'**, 9th European Conference on Power Electronics and Applications - EPE 2001. Graz 2001.

CHAPTER SEVEN

EXPERIMENTAL SETUP AND PRACTICAL RESULTS

In this chapter, the system hardware for the matrix converter is outlined. The power converter circuit, voltage and current sensing, and other elements of the matrix converter system are described. A selection of practical results for the matrix converter used as a standard ac to ac converter and as a UPFC are presented and discussed.

7.1 Hardware elements

The hardware components used in the drive system are shown in fig. 7.1 and are discussed in the following sections.

7.1.1 The Digital Controller Hardware

The digital controller hardware consists of three units:

- the Digital Signal Processor (DSP);
- the analogue expansion daughterboard; and
- a personal computer (PC).

i- Digital signal processor

The Texas Instrument digital signal processor TMS320C6711 DSK™ [7.1] shown in fig. 7.2 is used as the floating point digital controller unit.

The main features of this DSP are:

- up to 600 million floating-point operations per second (MFLOPS) and 1200 million instructions per second (MIPS) at a clock rate of 150 MHz.

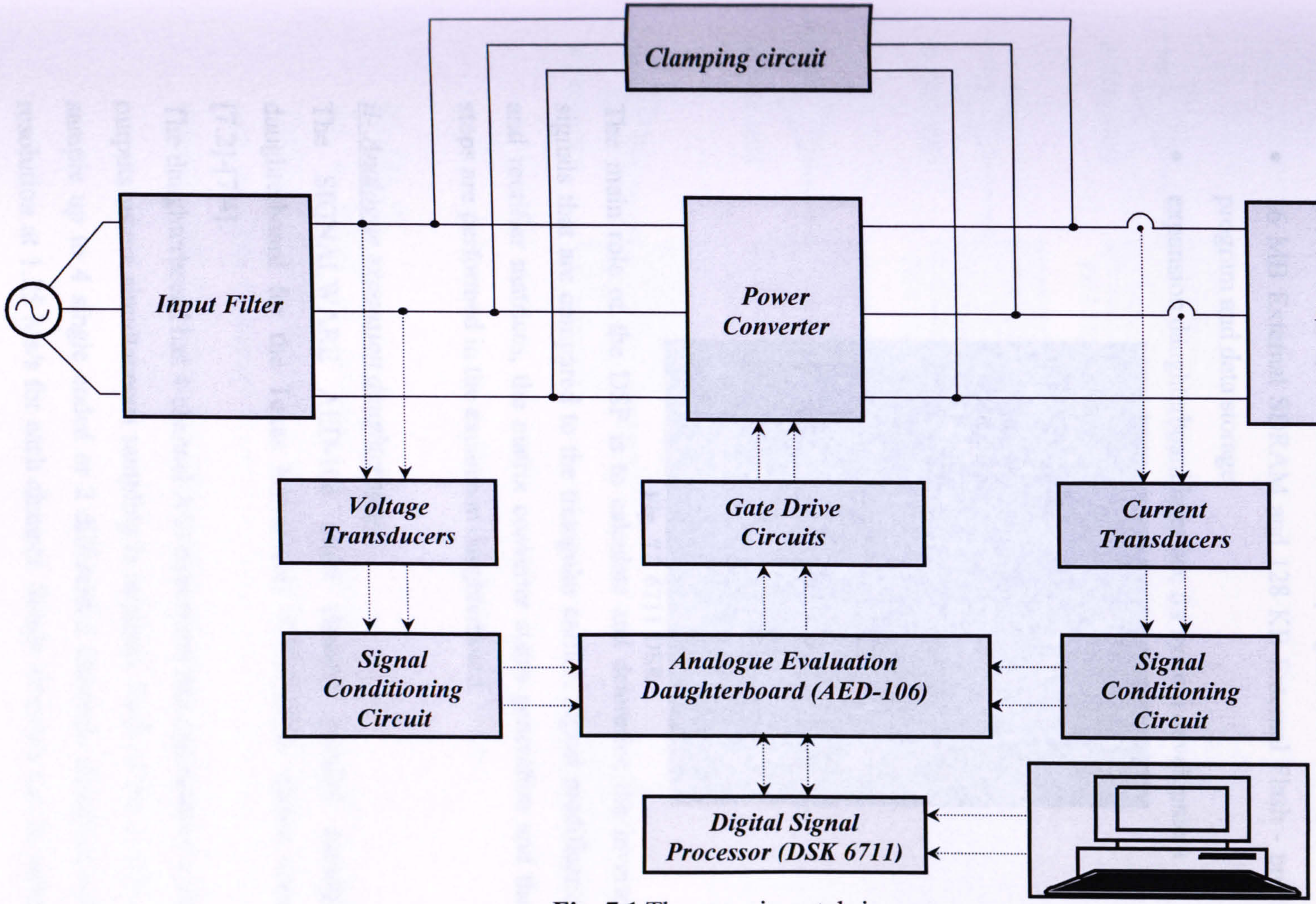


Fig. 7.1 The experimental rig.

- 32 general-purpose registers of 32-bit word length and eight independent functional units.
- two-level cache-based architecture. The Level 1 program cache (L1P) is a 32kbit direct mapped cache and the Level 1 data cache (L1) is a 32-kbit 2-way set-associative cache. The Level 2 memory/cache (L2) consists of a 512-kbit memory space that is shared as program and data space.
- 16 MB External SDRAM and 128 KB External Flash - provides additional program and data storage
- expansion daughterboard interface for system development.

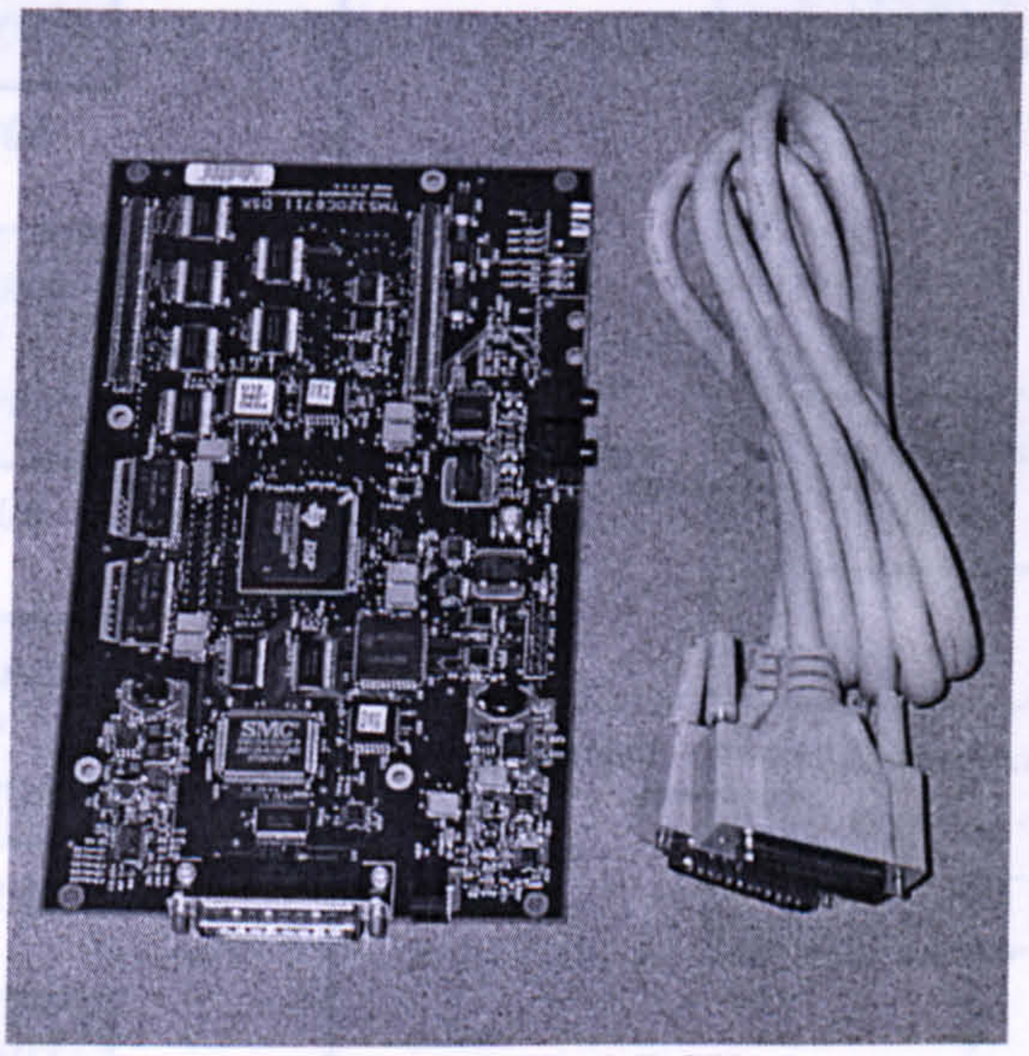


Fig. 7.2 6711 DSK.

The main role of the DSP is to calculate and determine the inverter and rectifier signals that are compared to the triangular carrier. Signal modification, the inverter and rectifier matrices, the matrix converter states generation and the commutation steps are performed in the expansion daughterboard.

ii- Analogue expansion daughterboard

The SIGNALWARE AED-106 multi channel parallel analogue expansion daughterboard for the Texas Instruments TMS320C6x allows sensor interfacing [7.2]-[7.4].

The daughterboard has 4-channel A/D converters that can monitor a variety of sensor outputs where simultaneous sampling is required. Each of the 4 A/D converters can sample up to 4 single ended or 2 differential channels simultaneously with 12 bit resolution at 1.25 Ms/s for each channel. Single channels can be sampled at up to 6

Ms/s. In addition to the analogue channels, three groups of 8 buffered digital signals can be either inputs or outputs. Preprocessing of the A/D samples and control of the converters are achieved with the field programmable gate array, which reduces the DSP load.

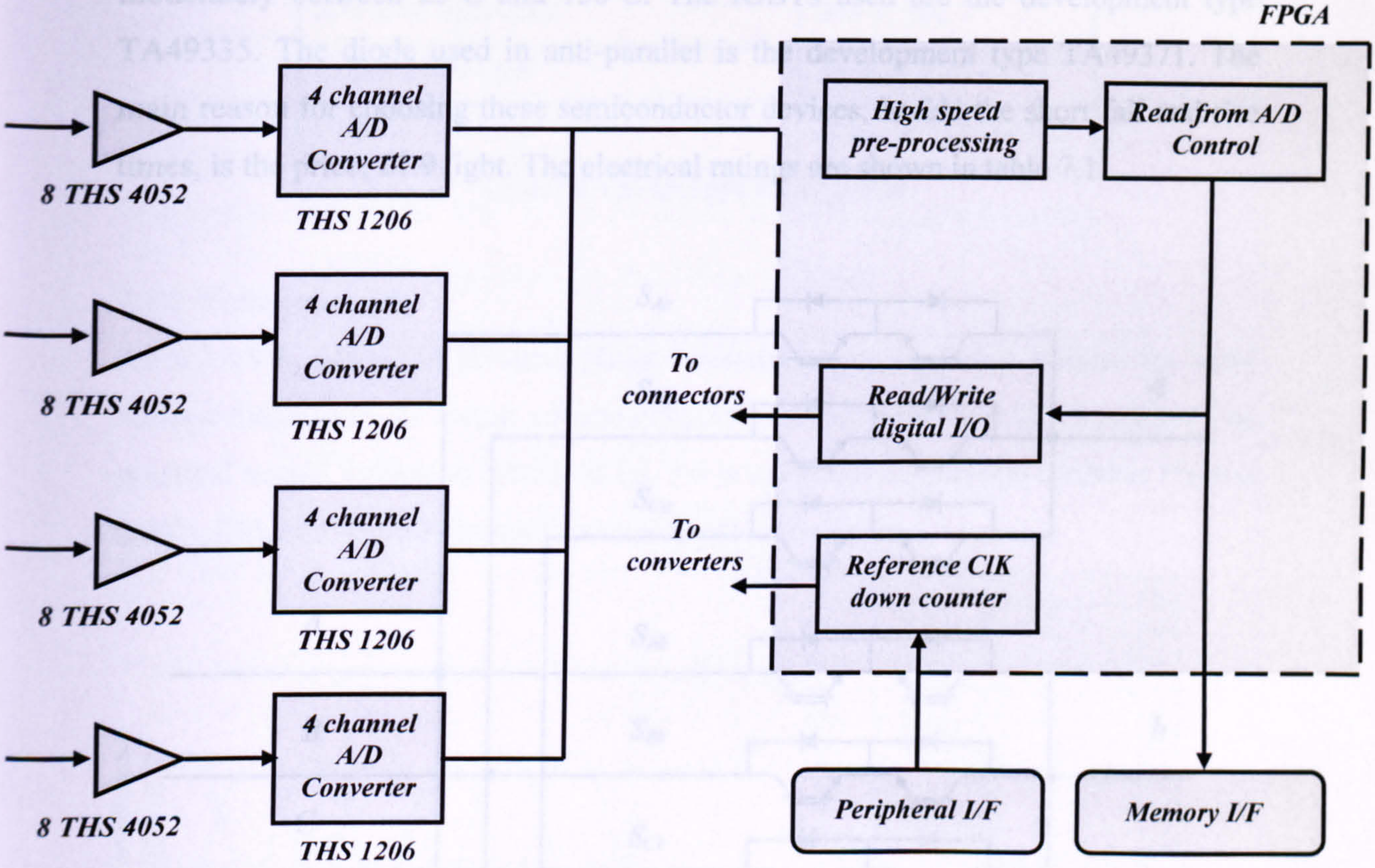


Fig. 7.3 AED-106 block diagram.

iii- The PC

As a means of accessing and programming the digital controller (DSP TMS320C6711™) in a user-friendly and non-intrusive real-time way, a PC equipped with the Code Composer Studio™ software is used. The program can be written in assembly or C languages. The latter is used for its simplicity and familiarity.

7.1.2 The power matrix converter

The conventional matrix converter topology shown in fig. 7.4 is used. The igbt semiconductors used are HGTG12N60A4D [7.5]. They are MOS gated high voltage switching devices combining the best features of MOSFETs and bipolar transistors. These devices have the high input impedance of a MOSFET and the low on-state conduction loss of a bipolar transistor. The low on-state voltage drop varies only moderately between 25°C and 150°C. The IGBTs used are the development type TA49335. The diode used in anti-parallel is the development type TA49371. The main reason for choosing these semiconductor devices, beside the short fall and rise times, is the price; £1.9 /igbt. The electrical ratings are shown in table 7.1.

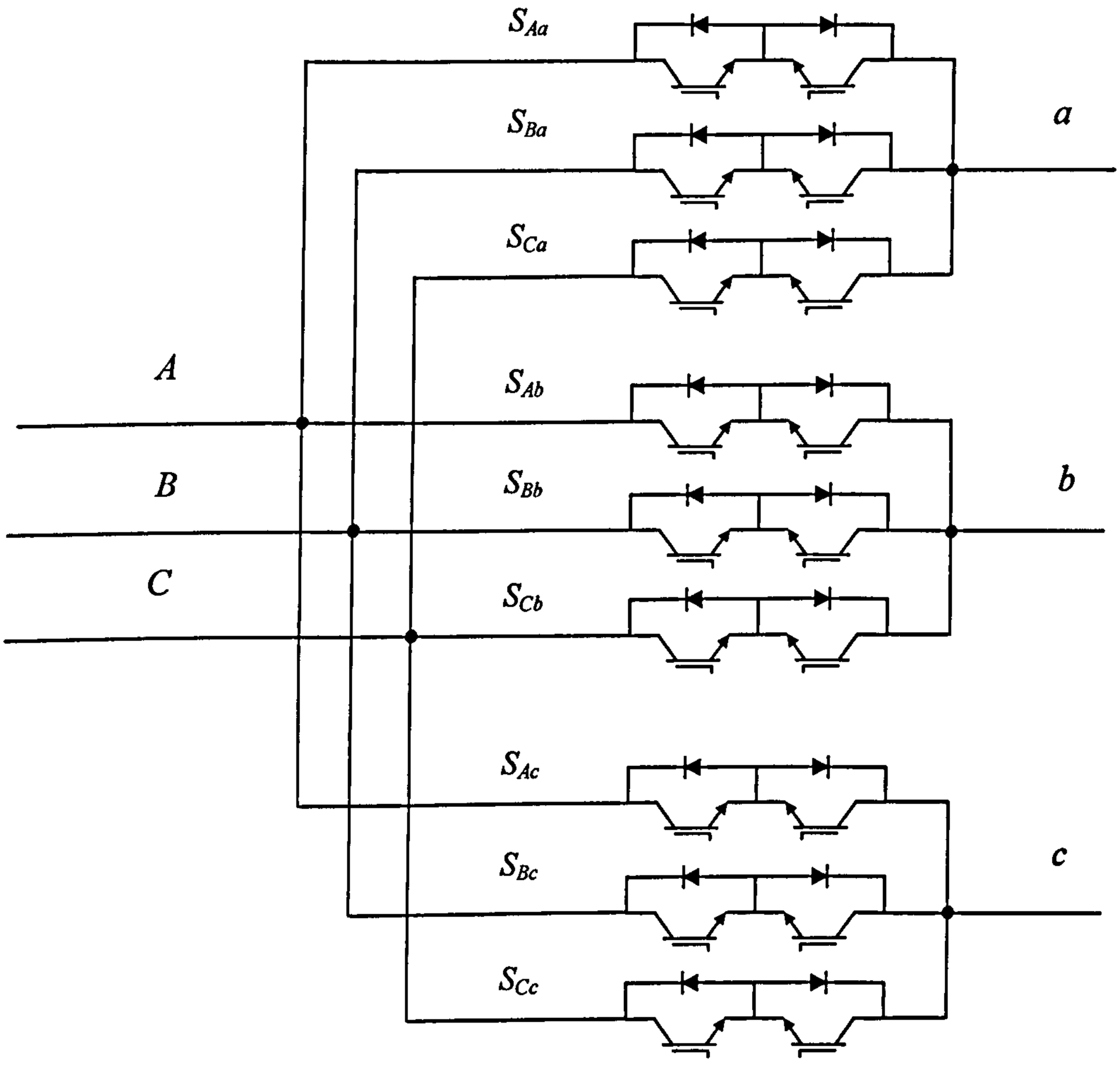


Fig. 7.4 Conventional eighteen switch matrix converter.

collector to emitter voltage	600 V
collector current continuous at $T_C = 25^oC$ at $T_C = 110^oC$	 54 A 23 A
Collector current pulsed	96 A
>100kHz operation	390V, 12A
200 kHz operation	390V, 9A
Typical fall time	70ns at $T_J = 125^oC$
Typical rise time	16ns at $T_J = 125^oC$

Table7.1 The igbt electrical ratings.

7.1.3 Voltage supply

An 8.3 kVA, 415V, 50 Hz three phase autotransformer is used to control the input voltage magnitude. Its output voltage range is 0 to 465 V on the 415V ac mains. The practical output voltage as shown in fig. 7.5 is unbalanced (6%) and contains 1% and 0.08% fifth and seventh harmonics component respectively.

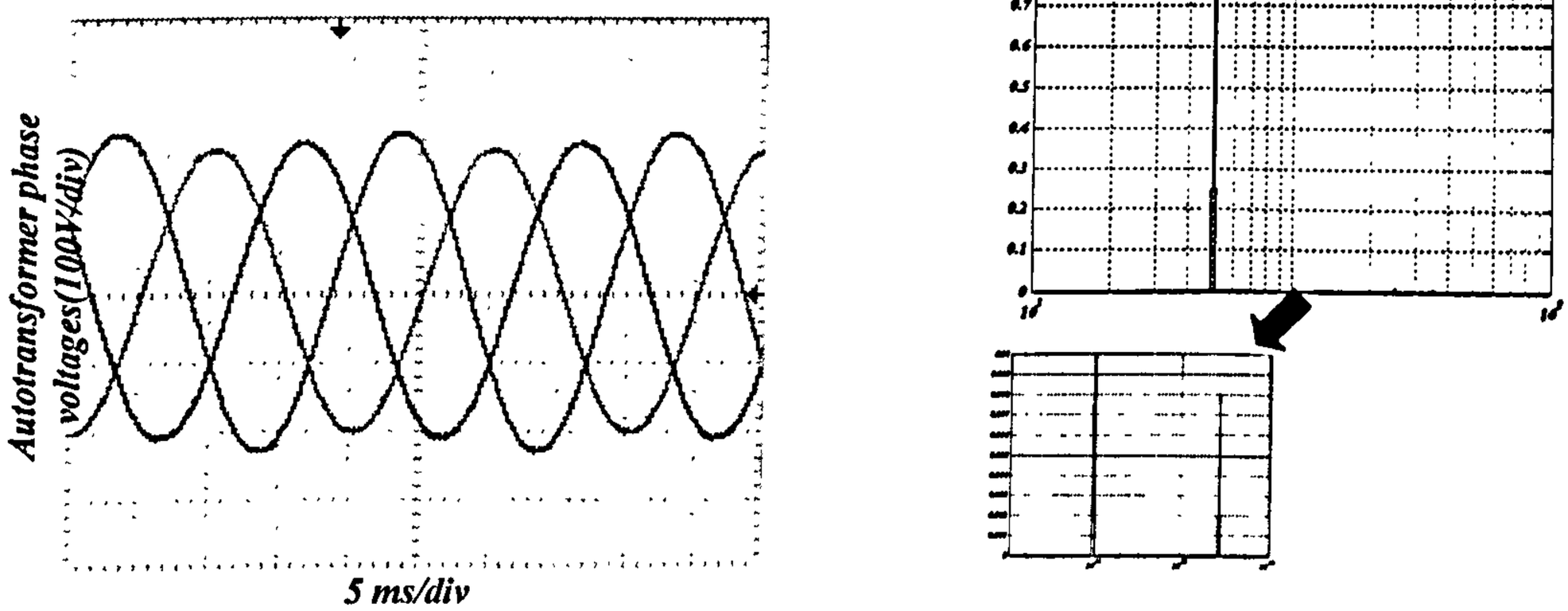


Fig. 7.5 Output voltage from the autotransformer.

7.1.4 The R-L load

A constant R-L load is used to demonstrate the novel technique presented in chapter 4. The typical load values are shown in table 7.2. These values are used in the output current controller. The measured values for the resistances are 19.8, 20.5, and 21.3 Ω

for phase a, b and c respectively. For the inductors, the measured values are 20.4, 19.6, and 18.7 mH.

R/phase	21.5Ω
L/phase	20mH

Table 7.2 Load typical Values

7.1.5 The input power filter

The main function of the input power filter is to reduce the input current ripple with minimum reactive element energy.

The design of the input filter in fig. 7.6 meets the following requirements:

- A high attenuation at the switching frequency.
- Minimized voltage drop across the filter to achieve a high overall voltage transfer ratio.
- Minimized weight and volume.
- Maximized operating power factor.

The input filter characteristics are shown in table 7.3.

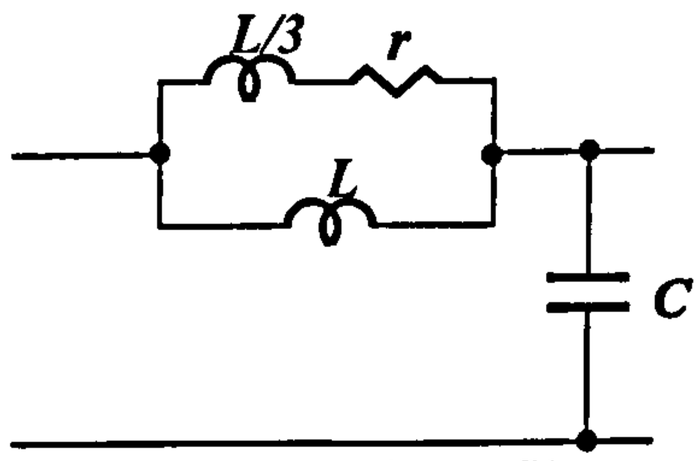


Fig. 7.6 The input filter.

L/phase	6.3 mH
C/phase	10 μF
R/phase	20 Ω
ω_c	4500 rad/s
% voltage drop at full load	0.15 %
Attenuation at the switching frequency	-26dB

Table 7.3 Input filter typical values and characteristics.

7.1.6 Current and voltage measurements

i- Current sensing unit

Phase current sensing uses two Hall-effect current transducers: LEM LA100 P [7.6] which have the following specifications:

- accuracy ± 0.45%.
- linearity <0.15%.
- thermal drift -25^{0}C....85^{0}C ± 0.05... ± 0.25 mA .
- response time < 1μs.
- bandwidth (-1 dB) dc...200kHz.
- high immunity to external interference.

For a sensed current in the range -10 to 10A, the voltage output signals of these transducers are in the range -1.5 to 1.5V, which is the A/D input voltage limits. The relationship between input current I_p and transducer output voltage V is:

$$V = 0.15 I_p \qquad 7.1$$

b- Voltage sensing unit

For an input voltage in the range -240 to 240V, the voltage output signals of these transducers should be within the range -1.5 and 1.5V which is the A/D input voltage limits. The relationship between input voltage V_i and transducer output voltage V is described by:

$$V = \frac{0.15}{24} V_i \qquad 7.2$$

Line voltage sensing was carried out using voltage transducers: LEM LV25-P [7.7] which have the following specifications:

- Accuracy ± 0.9%.
- linearity <0.2%.
- thermal drift ± 0.06... ± 0.25 mA at 0^{0}C....25^{0}C.
- response time 40μs.
- high bandwidth (L/R dependant).
- high immunity to external interference.

7.1.7 Switch commutation

Four quadrant switches are needed for the matrix converter. Four quadrant switches are able to block voltage of both polarities and to conduct current in either direction. Such a specification is realised by the connection of two diodes and two switches as shown by the combination of S_{1A}, S_{2A}, S_{1B}, and S_{2B} in fig. 7.7. The four quadrant switches in a matrix converter configuration suffer commutation difficulties. In this figure, it is assumed that the load is connected to phase 'A' (S_{1A}, S_{2A} are on and S_{1B}, S_{2B} are off). It is required to switch the connection to phase 'B'.

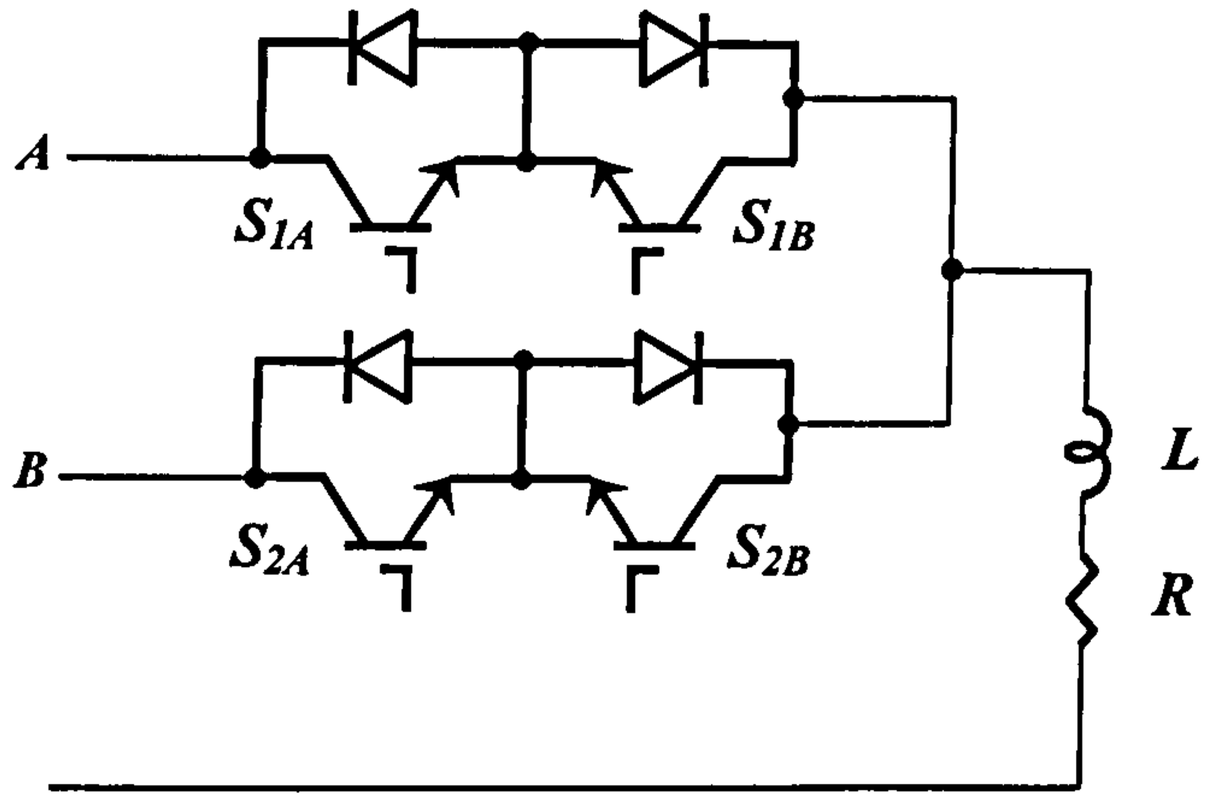


Fig. 7.7 The 2-1 matrix converter

The main constraint in the commutation process is to ensure that:

- no short circuit occurs between the input terminals A and B.
- no open circuit occurs in the inductive load circuit.

The commutation procedure used is current direction dependent. Figures.7.8 and 7.9 illustrate the commutation procedure during the load current positive and negative half cycles respectively. During the load current positive half cycle represented in fig. 7.8, S_{1A} carries the load current and no current flows in S_{2A}. So S_{2A} is turned off first. Then S_{1B} which should carry the load current is turned on. Then S_{1A} is turned off, so the current flows in S_{1B}. Finally, S_{2B} is turned on. During the negative half cycle, the procedure is the same with S_{1A}, S_{1B} interchanged with S_{2A}, S_{2B}.

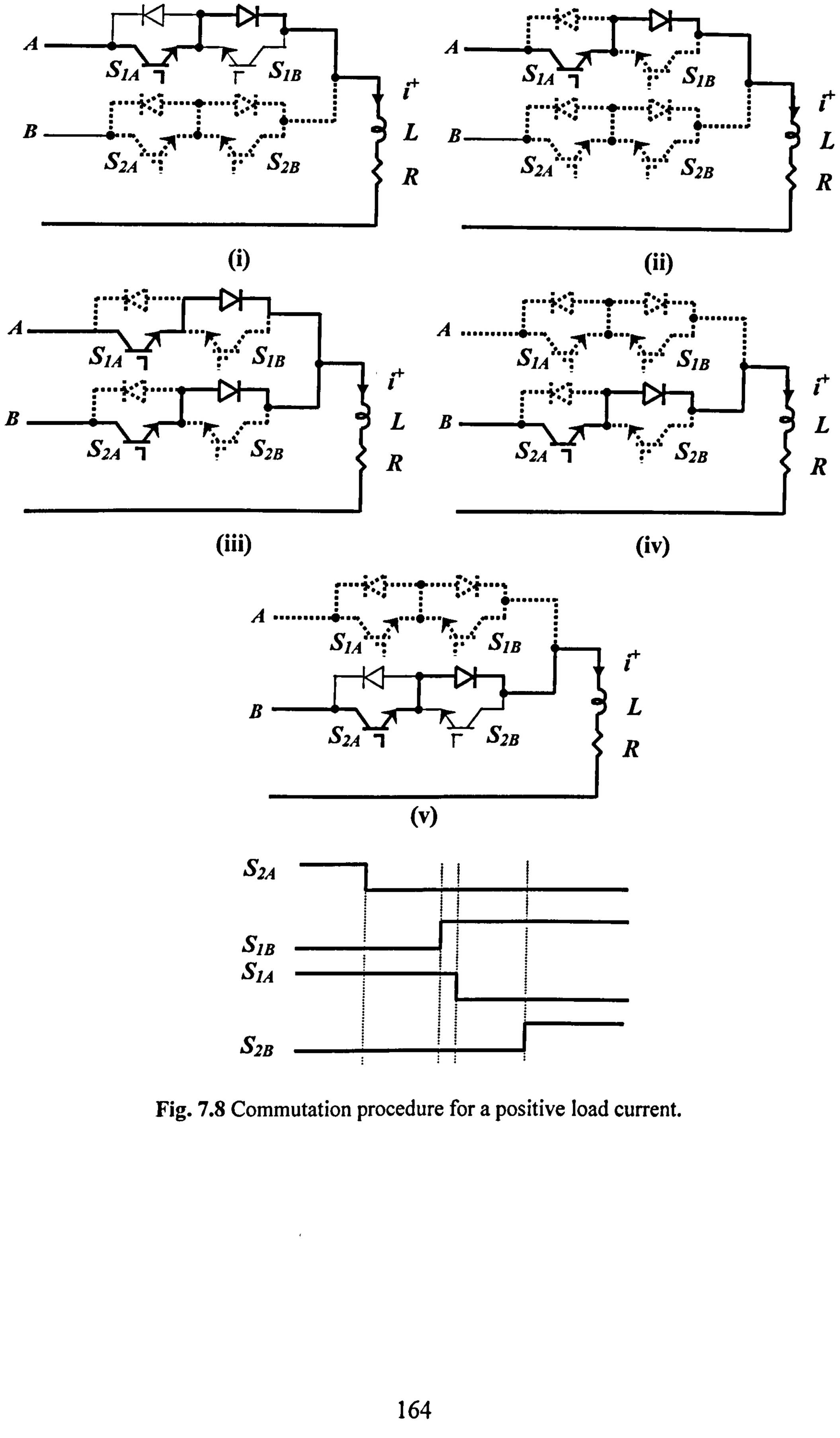


Fig. 7.8 Commutation procedure for a positive load current.

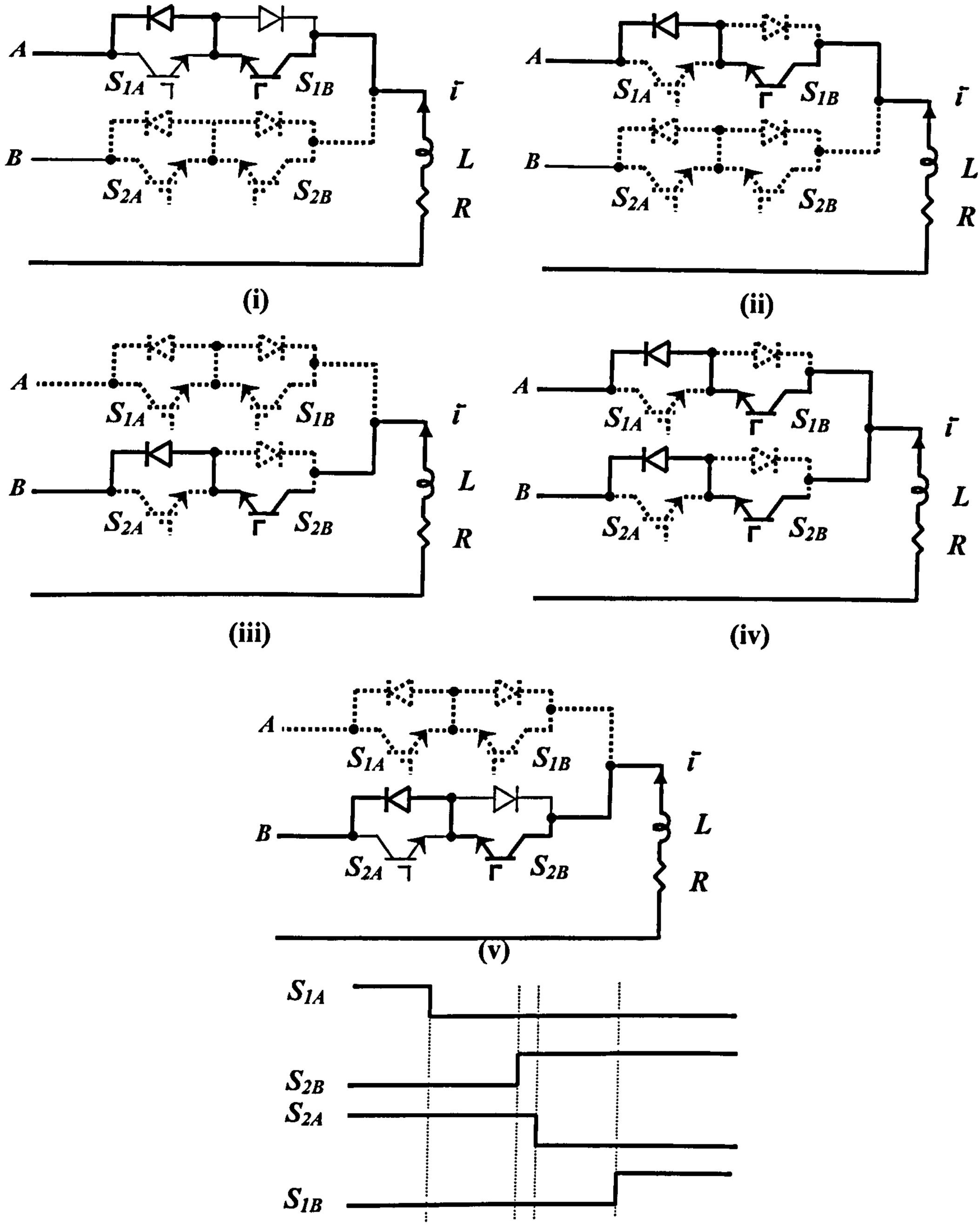


Fig. 7.9 Commutation procedure for a negative load current

The commutation logic circuitry performs two functions. The first is to set the four step commutation delays (x_1 *to* x_4). The second is to decode which of these delayed signals should be applied to each switch during commutation. The delay circuit uses four *D* flip flops, as shown in fig. 7.10.

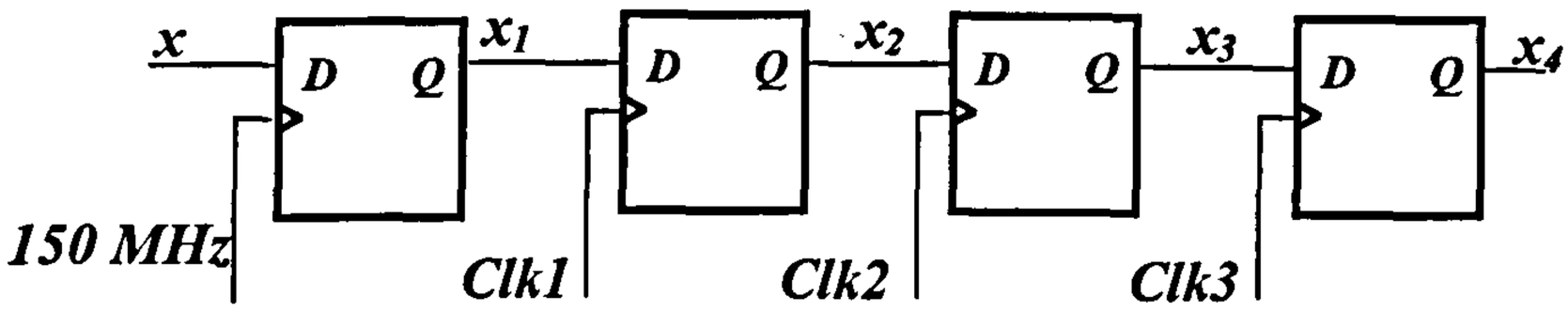


Fig. 7.10 Logic circuitry for setting the commutation delay times.

The signal selection depends on the switch state and the current direction; as to whether it should be turned on, off or unchanged. The switch state is determined by x and x_1. If x and x_1 are the same then there is no need to change the switch state. If x=1 and x_1=0 then the switch x should be turned off. If x=0 and x_1=1 then x is turned on. These cases are summarized in table 7.4.

x	x_1	*ON*	*OFF*	*No change*
0	*0*	*0*	*0*	*1*
0	*1*	*1*	*0*	*0*
1	*0*	*0*	*1*	*0*
1	*1*	*0*	*0*	*1*

Table 7.4 Truth table for the switch states and delay signals.

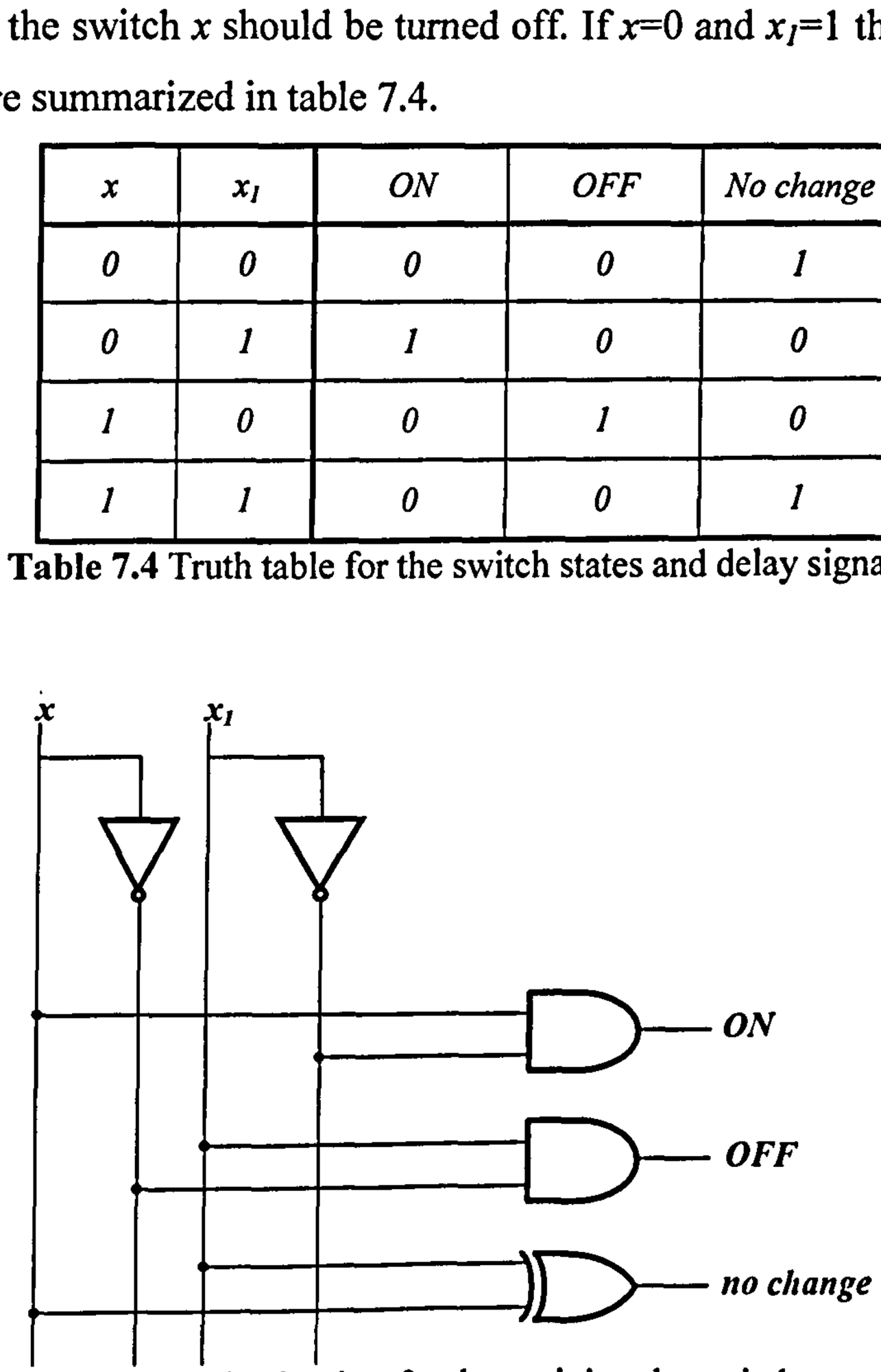


Fig. 7.11 Logic circuitry for determining the switch states.

The four delay states (x_1, x_2, x_3 *and* x_4) for each switch are summarized in table 7.5.

N.C.	*ON*	*OFF*	*DIR*	S_1	S_2
1	*D*	*d*	*d*	x_1	x_1
0	*1*	*0*	*0*	x_4	x_2
0	*1*	*0*	*1*	x_2	x_4
0	*0*	*1*	*0*	x_1	x_3
0	*0*	*1*	*1*	x_3	x_1
0	*1*	*1*	*d*	*XX*	*XX*

Table 7.5 Truth table for delay signal selection according to switch states and current direction.

7.1.8 The clamp circuit

A clamp circuit for matrix converter protection is shown in Fig. 7.12. The clamp consists of two rectifier bridges using fast-recovery diodes at the input and output. A capacitor takes the commutation energy. A parallel resistor discharges the capacitor. The design is based on the assumption that a fault has generated an error signal that immediately turns off all the converter switches. The clamp's task is to absorb the load energy thereby preventing power switch damage. The total energy stored in the three-phase inductances carrying sinusoidal balanced currents is calculated as [7.8]

$$\frac{3}{4}\hat{I}_o^2 L_{Load} = \frac{1}{2}C\left(\hat{V}_i^2 - \hat{V}_{iL}^2\right) \qquad 7.3$$

where $\hat{I}_o$: the maximum output current

L_{load} : the load inductance

$\hat{V}_i$: the maximum admissible voltage

$\hat{V}_{iL}$: the maximum input

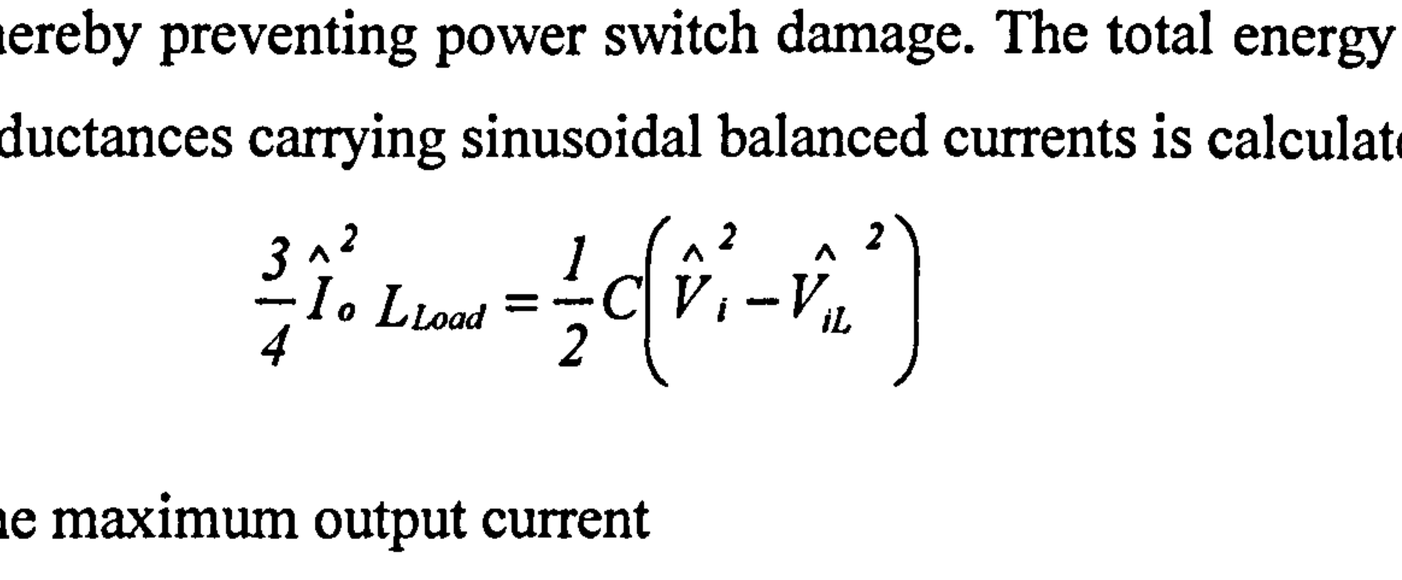


Fig. 7.12 The clamping circuit.

7.1.9 Signal filter

In order to avoid distortion, ground noise and glitches in the output from the daughterboard shown in fig. 7.13 a, HCPL-2630 dual-channel optocoupler isolation is used with the following specifications:

- high speed-10 MBit/s,
- CMR-10 kV/μs,
- Double working voltage -480V ,
- Fan-out of 8 over -40°C to +85°C,
- Strobable output, and
- Wired OR-open collector.

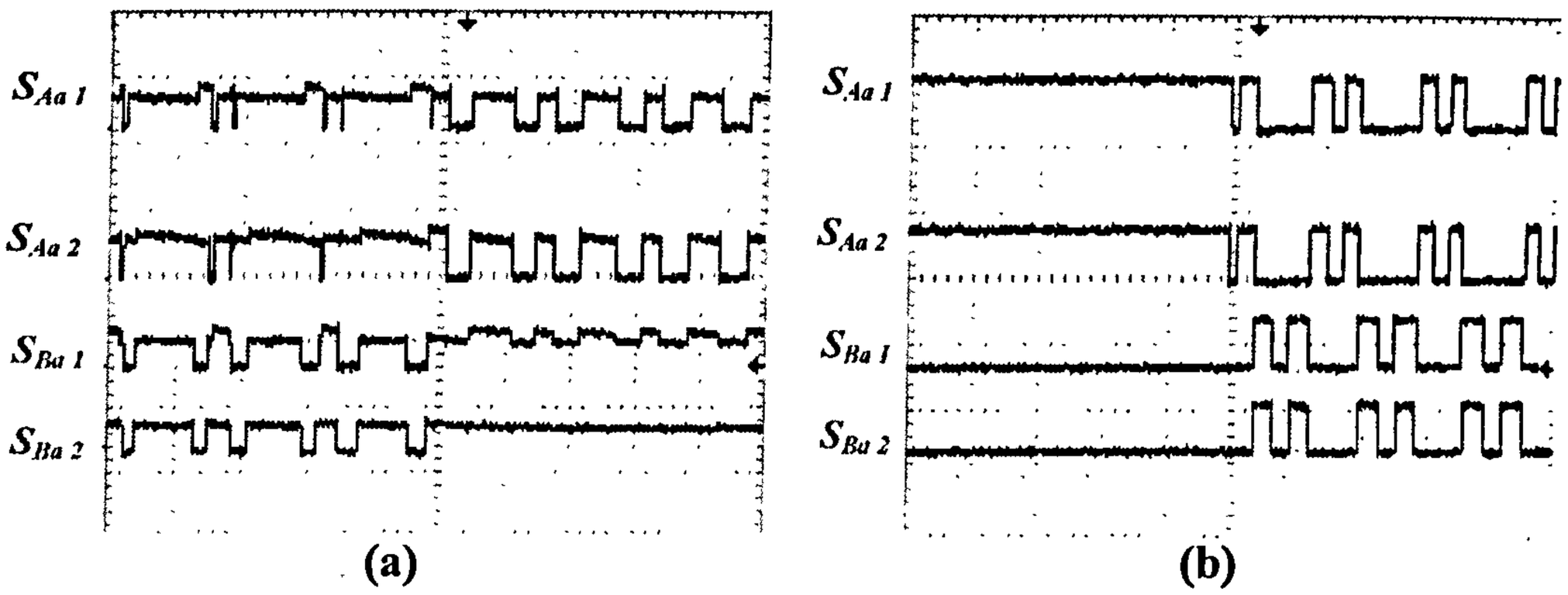


Fig. 7.13 Output to the switches from the gate drive:
(a) before the optocoupler (inverted)
(b) after the optocoupler.

The optocouplers solve the problem of distorted output signals but they introduce a delay as shown in fig. 7.14. This increases the time delay required for commutation. Normally the second and third commutation steps can be performed simultaneously. This is because the turn-off delay time of an igbt is greater than its turn-on delay time. This feature cannot be used because of the optocoupler delay. Instead of using a maximum delay of 0.5 μs for commutation, 2.125 μs is used.

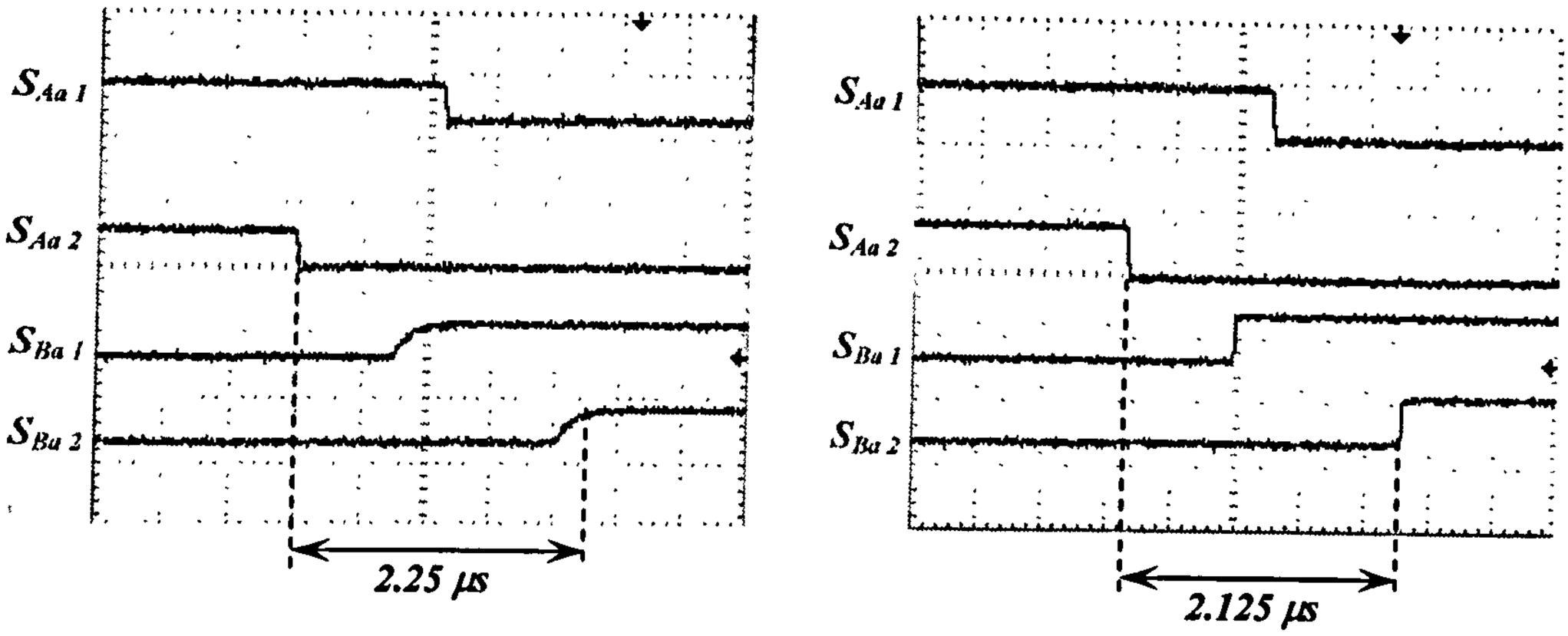


Fig. 7.14 (a) Optocouplers output signals,
(b) Gate drives output signals and
(c) Gate drives signals for phase a.

7.1.10 Digital filtering

Fig. 7.15a shows the measured ac voltage signal received by the DSP for a sinusoid input voltage. In order to reduce noise effects and avoid transients in the measured signals, they are filtered. First, the zero signals are eliminated as shown in fig. 7.15b. Second, four consecutive readings are taken instead of one per measured signal, the average of these four readings (moving average) is

$$x_k = \frac{x_1 + x_2 + x_3 + x_4}{4} \qquad 7.4$$

This reduces the noise effect as shown in fig. 7.15c.

Then the signals are low pass filtered again.

$$\frac{Y(s)}{X(s)} = \frac{\omega_0}{s + \omega_0} \qquad 7.5$$

which is

$$\frac{dy}{dt} + \omega_0 y = \omega_0 x \qquad 7.6$$

or in discrete form

$$y_k = \frac{y_{k-1} + \omega_0 T_s x_k}{(1 + \omega_0 T_s)} \qquad 7.7$$

The input voltages are filtered and the delay is compensated since only the input phase angle is required. Instantaneous output currents are needed so no filtering is performed. This reduces the robustness of the current feedback control.

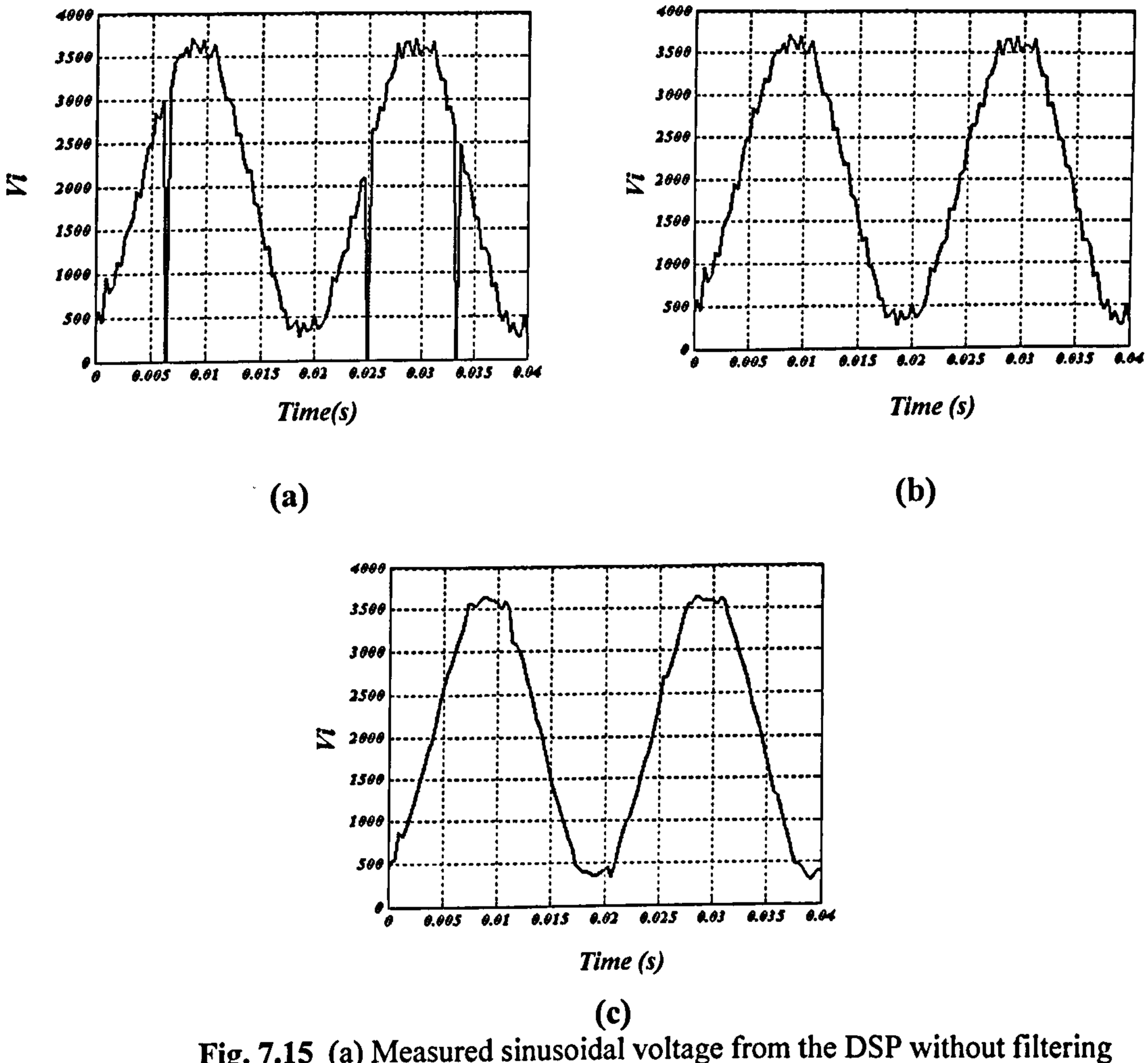


Fig. 7.15 (a) Measured sinusoidal voltage from the DSP without filtering
(b) measured signal after eliminating the zero notches and
(c) measured signal after averaging four consecutive signals.

7.2 Ac to ac matrix converter practical results

Both open loop and closed loop test are performed to show the controller's ability of controlling the output current even with asymmetrical voltage sources. Analysis of the practical results is given.

7.2.1 Open loop

Open loop control without output current feedback is investigated first. The output frequency is 80 Hz with a unity input power factor (zero displacement angle) and a 6.1 kHz switching frequency. The frequency and phase angle are estimated on the input side. Fig. 7.16 shows the flow chart of the open loop DSP program.

The practical results for the open loop tests are shown in fig. 7.17 parts a to f. The analysis focuses on the output and input current waveforms and their harmonic spectrum.

By considering the output current, the virtual dc link voltage with an unbalanced, flat-topped source is analysed to interpret the output current harmonic spectrum. The virtual dc link of an unbalanced source with its positive and negative components are shown in fig. 7.18 and is of the form

$$\begin{aligned} v_{DC} &= v_1.m_1 + v_2.m_2 \\ &= \left(\left(\hat{V}_{ip}\cos(\delta+\phi)+\hat{V}_{in}\cos(\delta-\phi+\varphi)\right)\sin\left(\frac{\pi}{3}-\delta\right)\right. \\ &\left.+\left(\hat{V}_{ip}\cos\left(\delta+\phi-\frac{\pi}{3}\right)+\hat{V}_{in}\cos\left(\delta-\phi+\varphi+\frac{\pi}{3}\right)\right)\sin\delta\right)\frac{T_{s\,eff}}{T_s} \end{aligned} \qquad 7.8$$

where φ is the angle between the negative and positive sequence components of phase a as shown in fig. 7.18.

The virtual dc link average voltage in the case of an unbalanced voltage supply is given by

$$v_{DC} = \left(\hat{V}_{ip}.\frac{\sqrt{3}}{2}\cos\phi + \hat{V}_{in}.\frac{\sqrt{3}}{2}\cos(2\omega t-\phi+\varphi)\right)\frac{T_{s\,eff}}{T_s} \qquad 7.9$$

which is shown in fig. 7.19. For this virtual dc voltage, the output current harmonic spectrum must contain two components, at $2f_i \pm f_o$. This explains the presence of the low frequency component at 20 Hz $(2\times 50Hz - 80Hz)$ and the fact that the

output current waveform given in fig. 7.17 looks as if modulated at 20 Hz (appendix b).

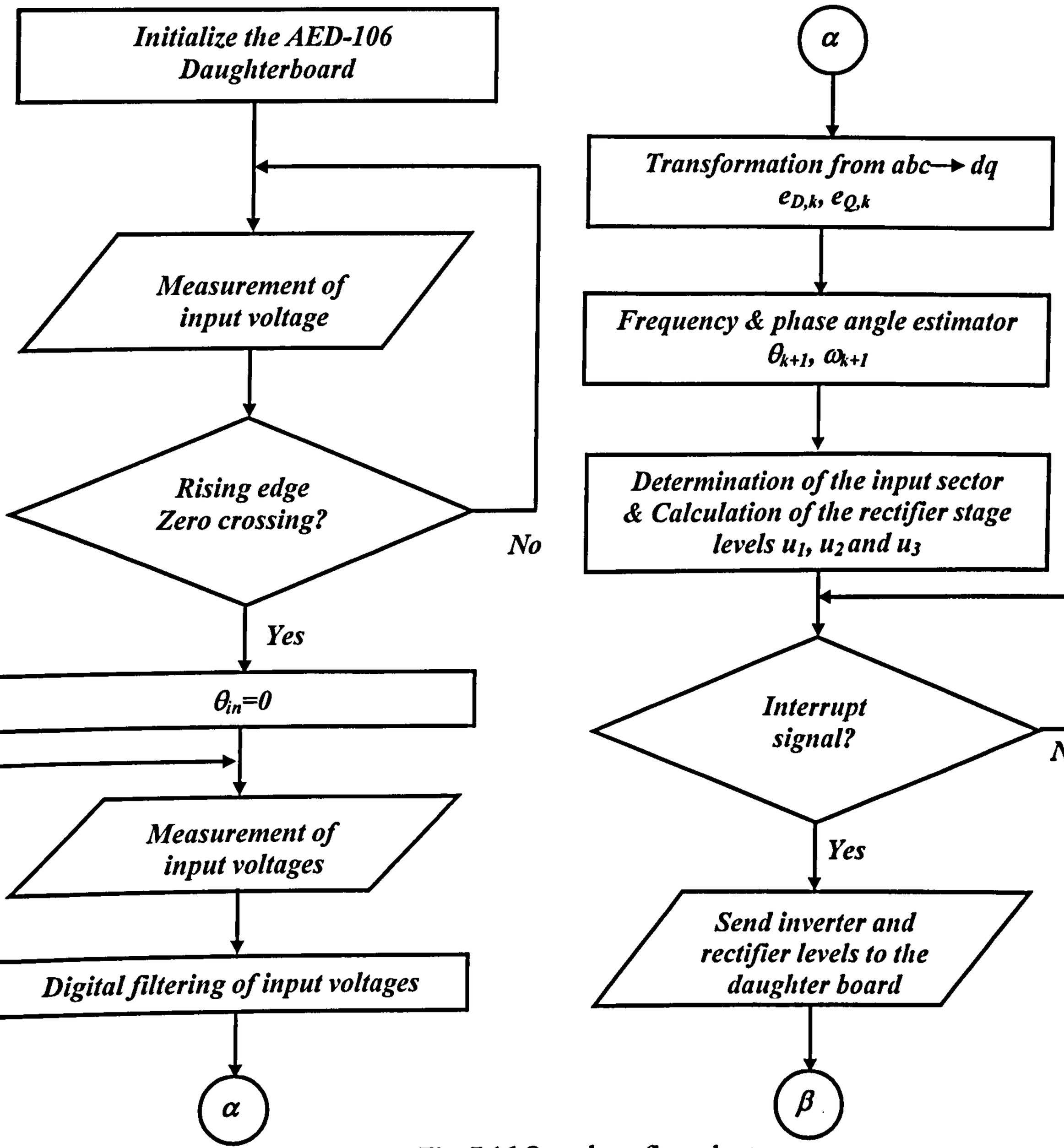
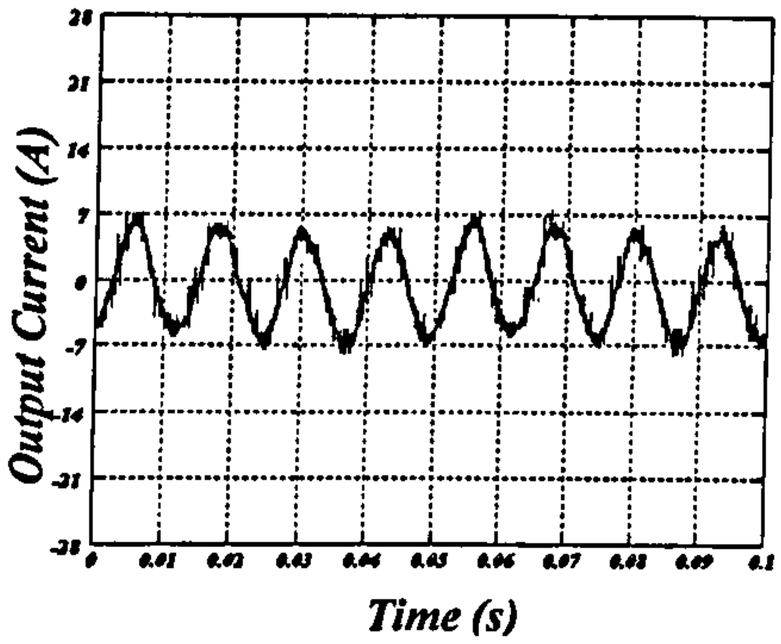
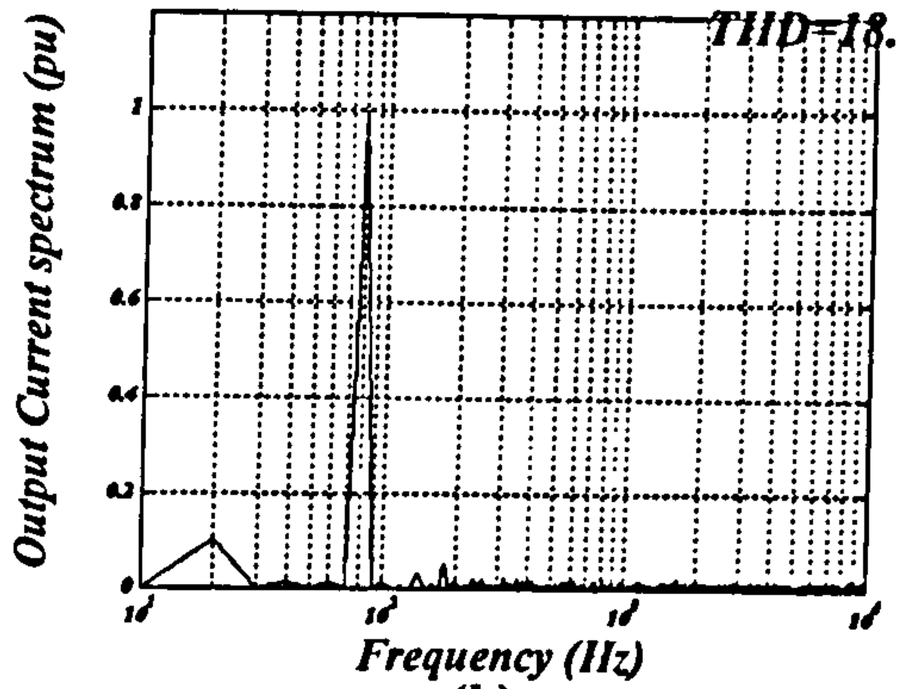


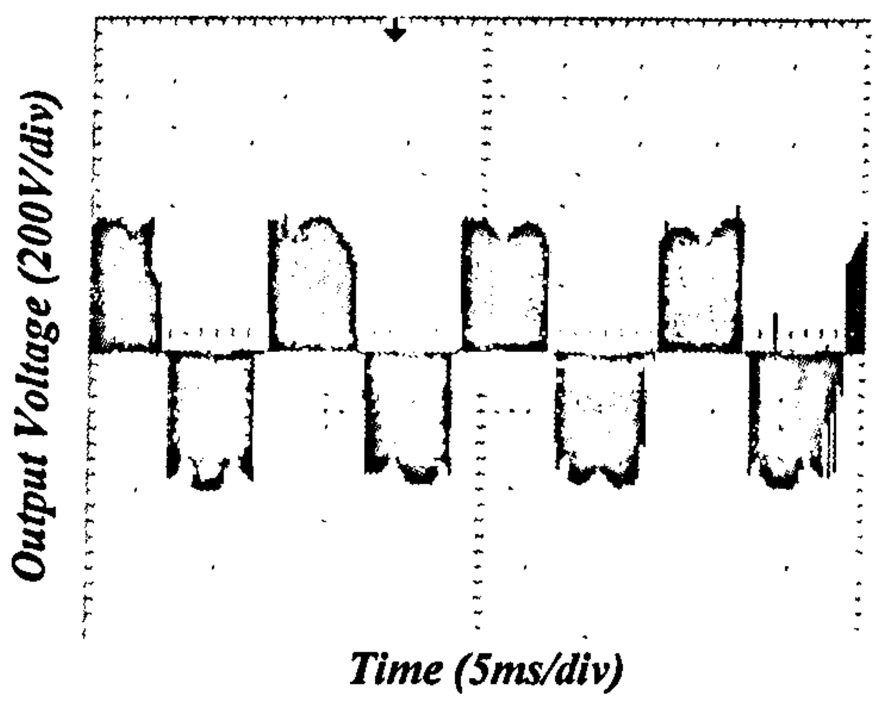
Fig. 7.16 Open loop flow chart.



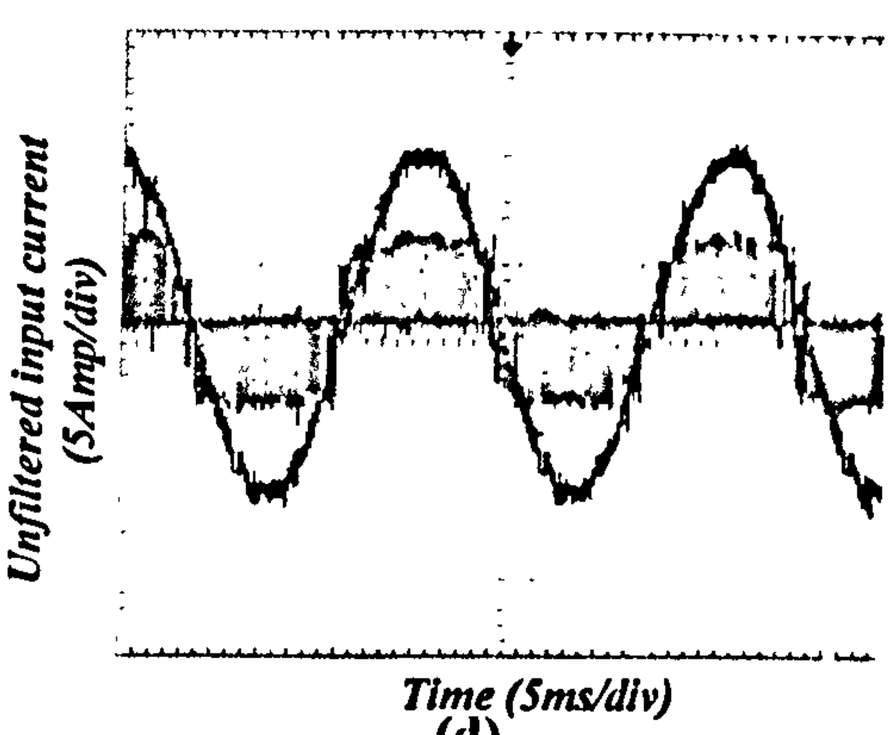
(a)



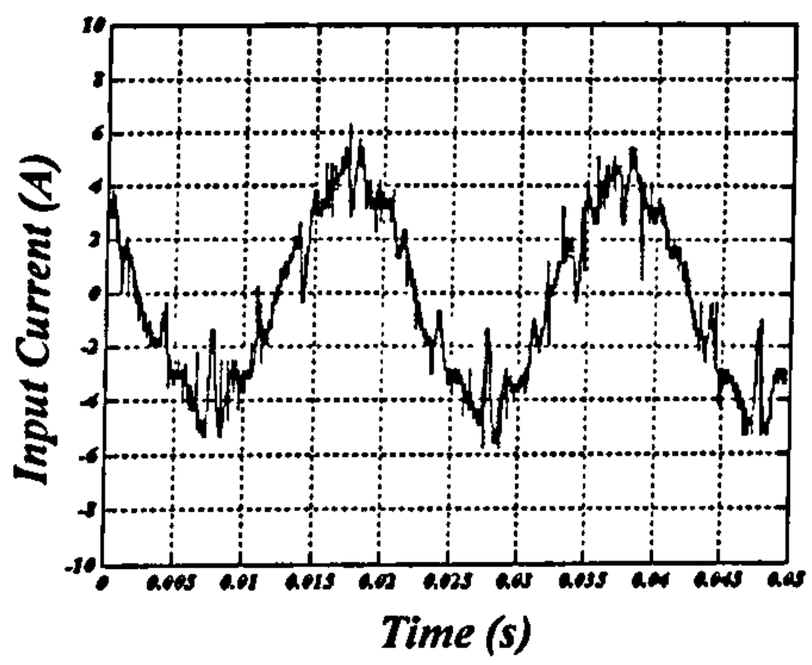
(b)



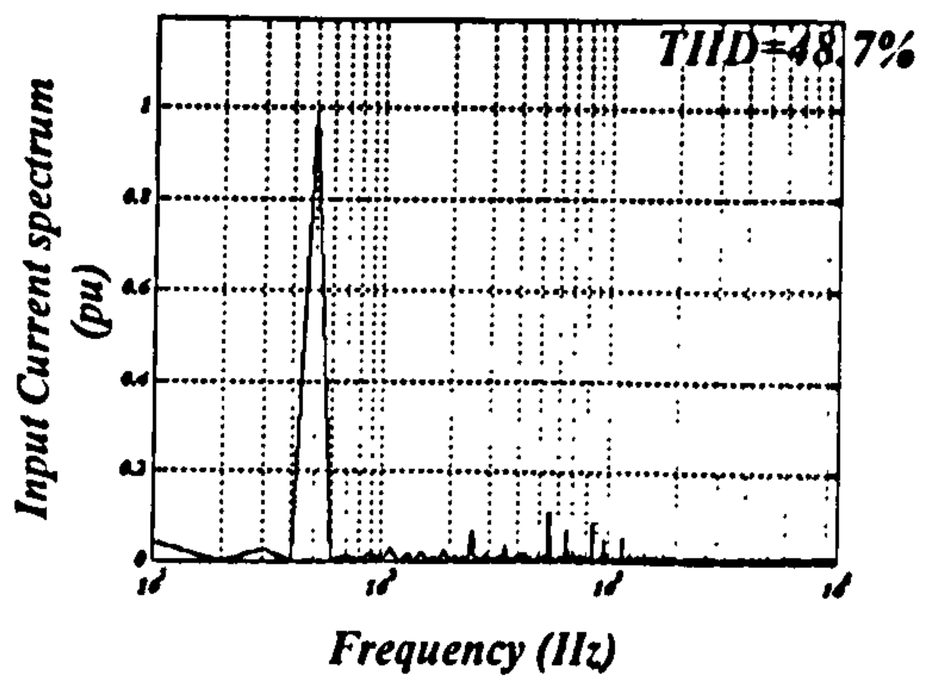
(c)



(d)



(e)



(f)

Fig. 7.17 Experimental output response for f_o=80 Hz, unity input power factor at f_s= 6.1 kHz: (a) output current, (b) output current harmonic spectrum, (c) output voltage, (d) unfiltered input current and input voltage, (e) filtered input current, and (f) input current harmonic spectrum.

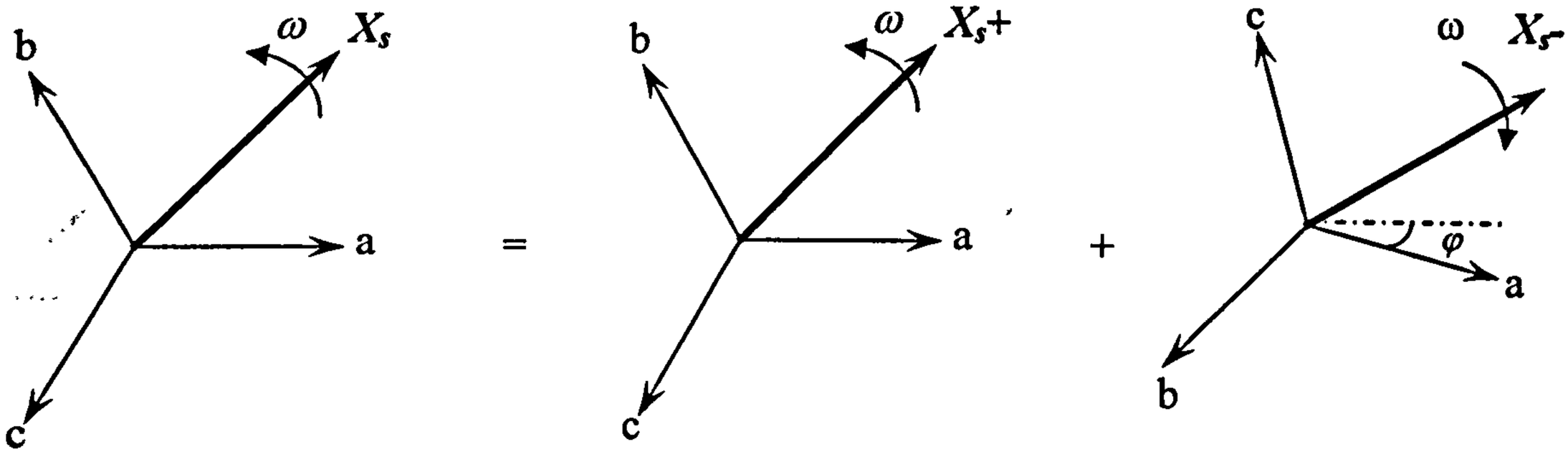
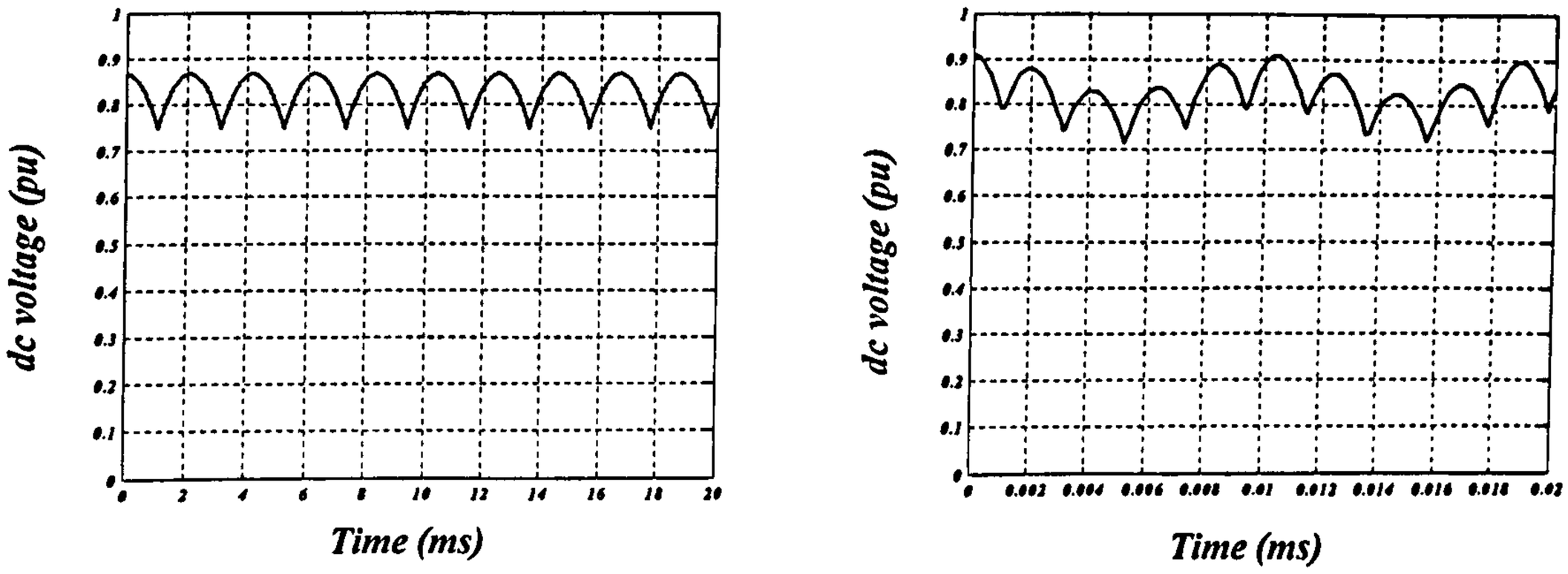


Fig. 7.18 Decomposition of signals into positive and negative sequence components.



(a) **(b)**

Fig. 7.19 The simulated virtual dc link voltage at unity modulation index and zero displacement angle for f_o=80 Hz:

(a) balanced voltage supply and (b) unbalanced voltage supply (5%).

For a flat-topped voltage source which contains fifth and seventh harmonic components as shown in fig. 7.20, the virtual dc link voltage is:

$$v_{DC} = \left(\hat{V}_{i1} . \frac{\sqrt{3}}{2} \cos\phi + \hat{V}_{i5} . \frac{\sqrt{3}}{2} \cos(6\omega t - \phi + \varphi_5) + \hat{V}_{i7} . \frac{\sqrt{3}}{2} \cos(6\omega t - \phi + \varphi_7) \right) \frac{T_{s\,eff}}{T_s} \qquad 7.10$$

The virtual dc link voltage in the case of a flat-topped source is shown in fig. 7.21. The output current contains harmonic components at $6 f_i \pm f_o$.

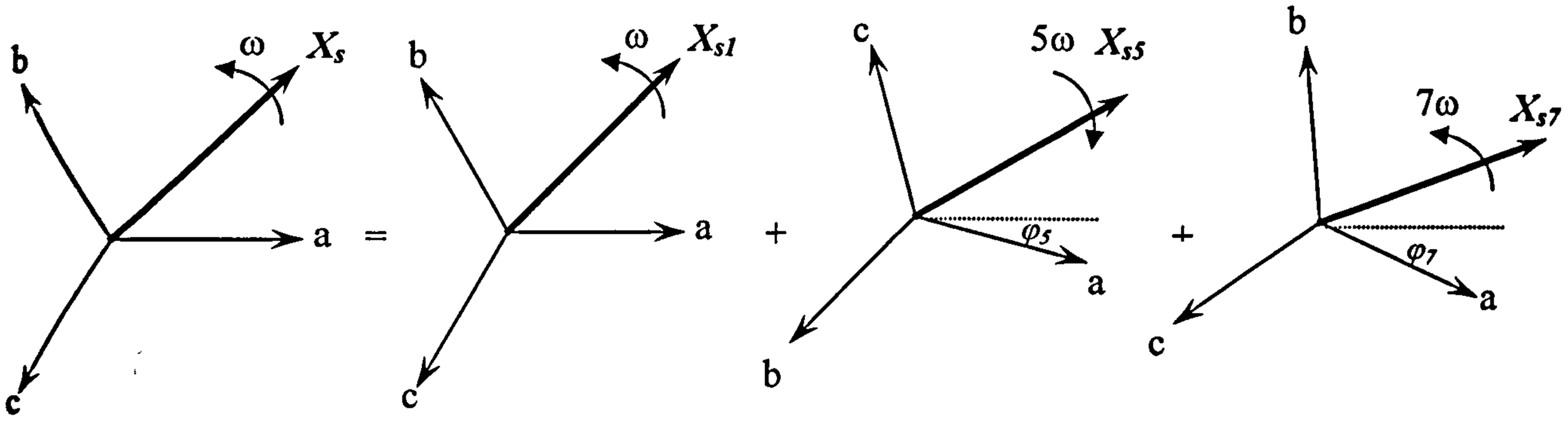
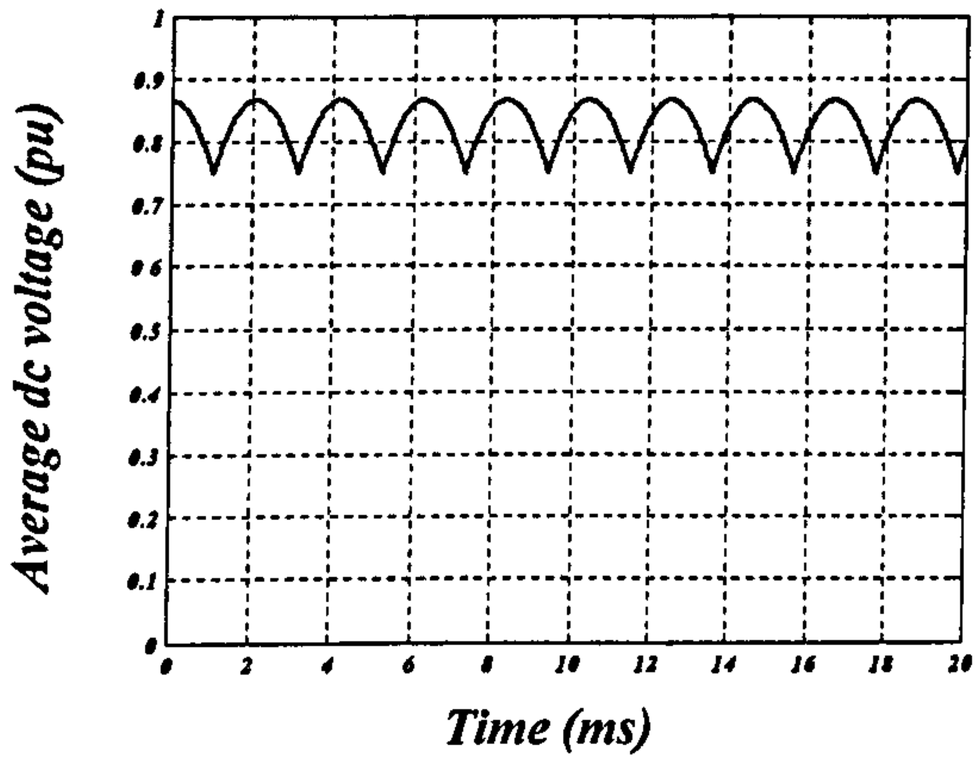
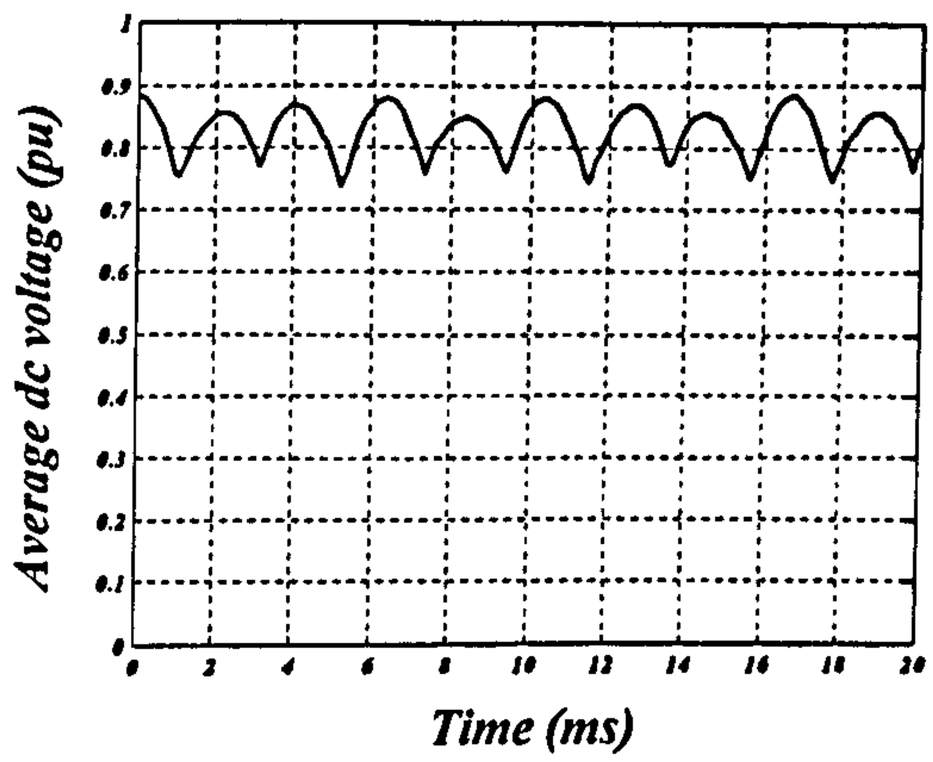


Fig. 7.20 Flat topped source fundamental, fifth, and seventh harmonic components.



(a)



(b)

Fig. 7.21 The simulated virtual dc link average voltage at unity modulation index and zero displacement angle for f_o=80 Hz:

(a) balanced voltage supply and

(b) flat-topped voltage supply (1% fifth and seventh harmonics).

Analysis of the virtual dc link current can explain the input current harmonic spectrum. The inverter stage uses space vector modulation. By neglecting the effect of the rectifier stage zero vectors and considering the output phase voltage has only a fundamental component for simplicity, the output current will be unbalanced due to the link imbalance reflected on the load. This means that the output current has positive and negative components. The virtual dc link current is no longer a constant value, instead it will contain an oscillatory component at double the output frequency due to the output current negative sequence component as follows

$$i_{DC} = \left(\hat{I}_{op} . \frac{\sqrt{3}}{2} \cos\phi_L + \hat{I}_{on} . \frac{\sqrt{3}}{2} \cos(2\omega_o t - \phi_L + \sigma) \right) \qquad 7.11$$

where ϕ_L is the load power factor angle of the positive sequence component, and σ is the angle between the positive and negative components of phase a. Each of the output current harmonic components ($6f_i \pm f_o, 2f_i \pm f_o$) will also have positive and negative sequence components. The input current related to any harmonic component ω_x will be:

$$i_i = \left(\hat{I}_{opx} \cdot \frac{\sqrt{3}}{2} cos((\omega_i \pm (\omega_o - \omega_x))t - \phi_x) + \hat{I}_{onx} \cdot \frac{\sqrt{3}}{2} cos(\omega_i \pm ((\omega_o + \omega_x))t + \phi_x) \right) \quad 7.12$$

A constant virtual dc current is necessary to obtain sinusoidal balanced input currents at the supply frequency similar to a normal current source inverter as in equation 3.13. Any oscillatory components in the virtual dc link will be reflected in the input current. This means that the input current contains components at $f_i \pm f_o \pm f_x$. These components affect the input current quality hence the input voltage to the matrix converter.

Another factor affecting the input current quality is the effect of the input filter Q and its corner frequency. Frequencies in the spectrum near the corner frequency will be amplified, not attenuated as shown in Fig. 7.22.

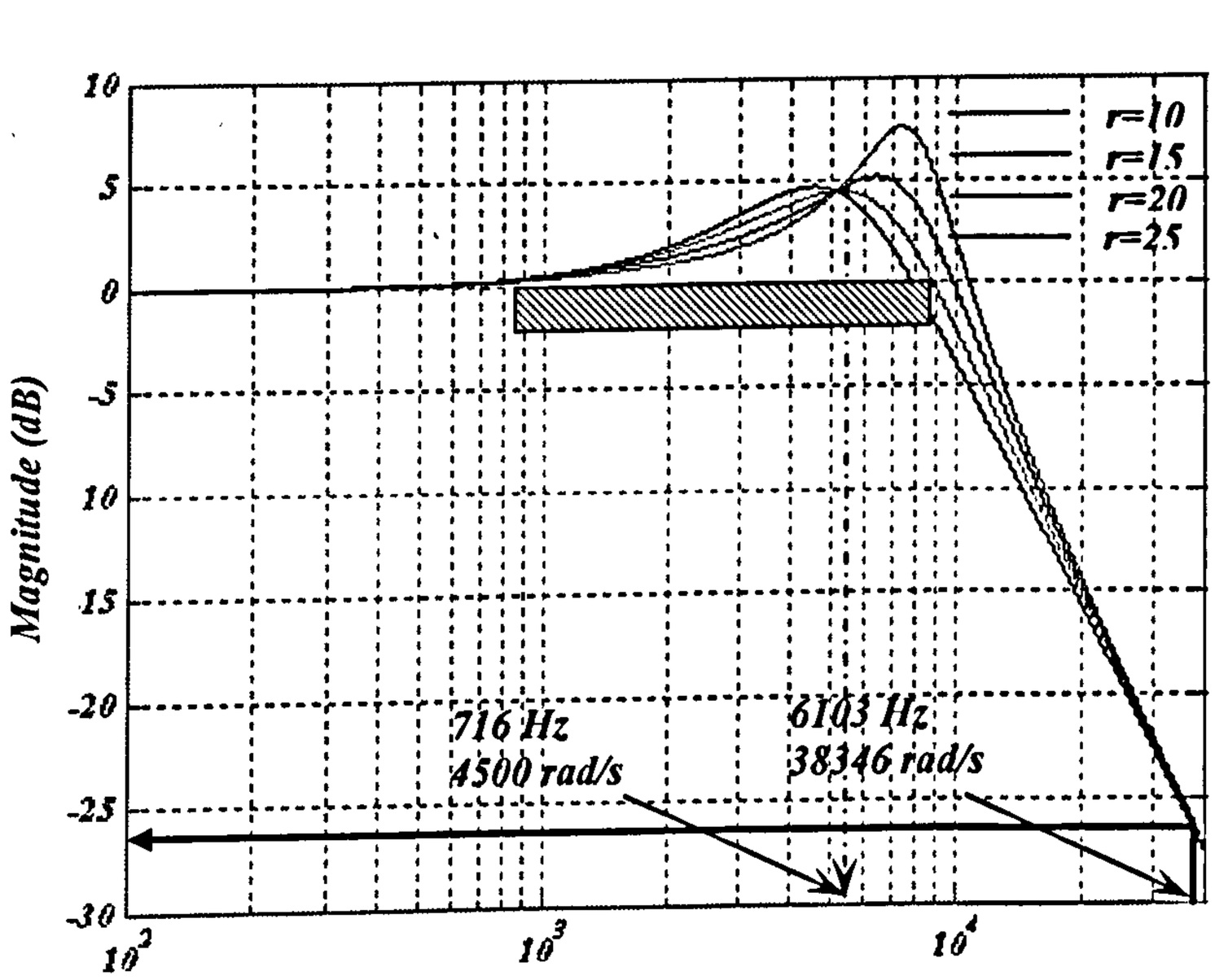


Fig. 7.22 Input filter frequency response.

7.2.2 Closed loop control

Closed loop control of the output side current is performed using the PI controller explained in chapter 4 with the same controller gains $k_p = 34, k_i = 5.125$. The study uses 20 and 80 Hz output frequencies, unity input power factor (zero displacement angle), and at a 6.1 kHz switching frequency. For the input side, the frequency and phase angle are estimated. Fig. 7.23 shows a flow chart of the closed loop DSP program (appendix c).

The closed loop practical results are shown in figures 7.24 and 7.25 parts a to f. The practical results focus on the output and input current waveforms and their harmonic spectrum.

For the 20 and 80 Hz output currents, the current quality is improved when compared with the open loop case in fig. 7.17, as indicated in the harmonic spectrum. The low order harmonic that appears in the open loop case has been reduced. The quality of the current is poorer than the simulation results due to:

- The effect of noise on the measured signals.
- The long switch commutation times are not accounted for in the simulation.
- Imbalance in the output load since the controller transient response is dependant on the load parameters.

The input current is slightly improved, but sinusoidal output current is at the expense of the input current quality due to the absence of any internal storage elements in the matrix converter. The input current at a 20 Hz output frequency is better than that at 80Hz. This is due to the effect of the input filter on the frequencies near its corner frequency.

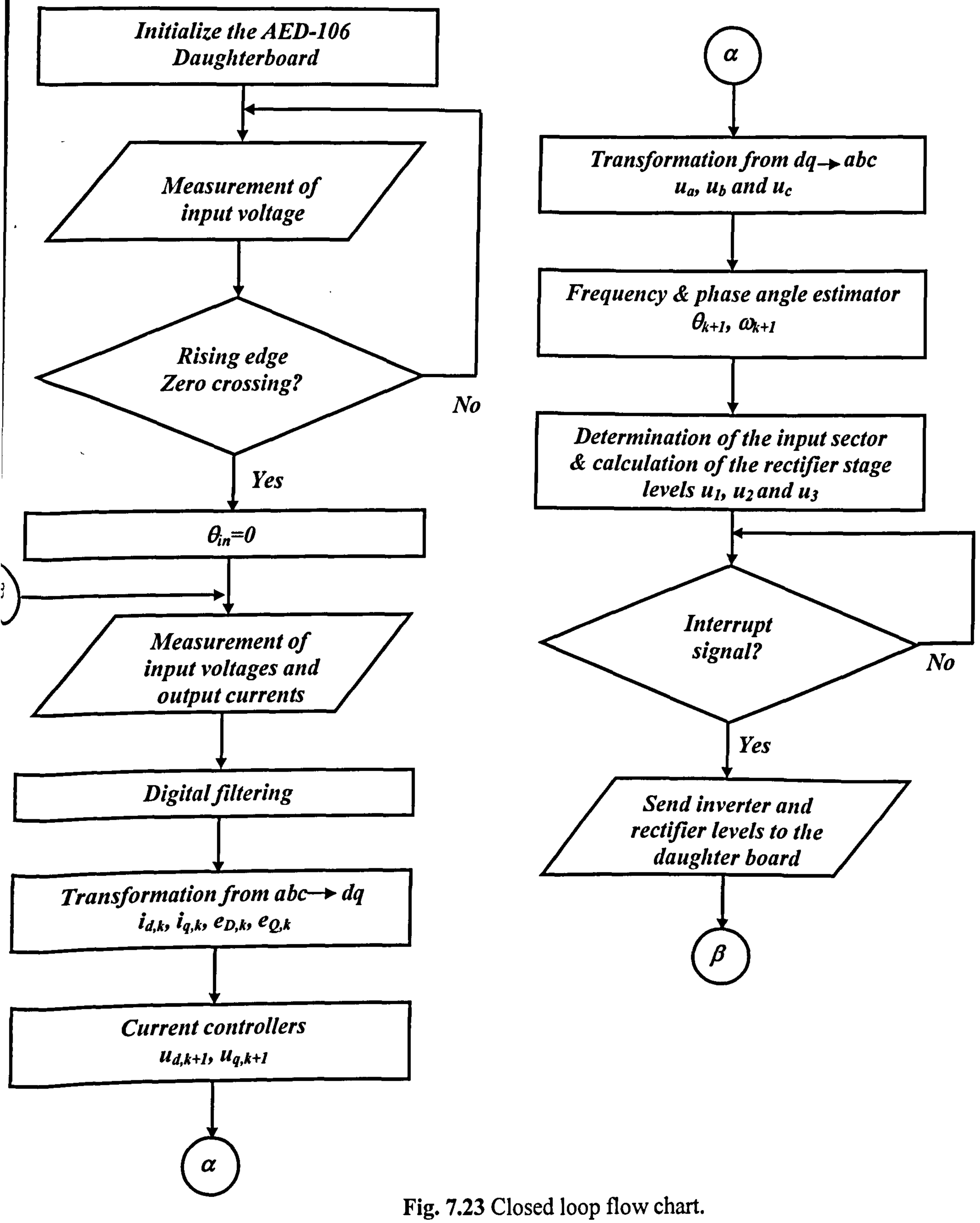


Fig. 7.23 Closed loop flow chart.

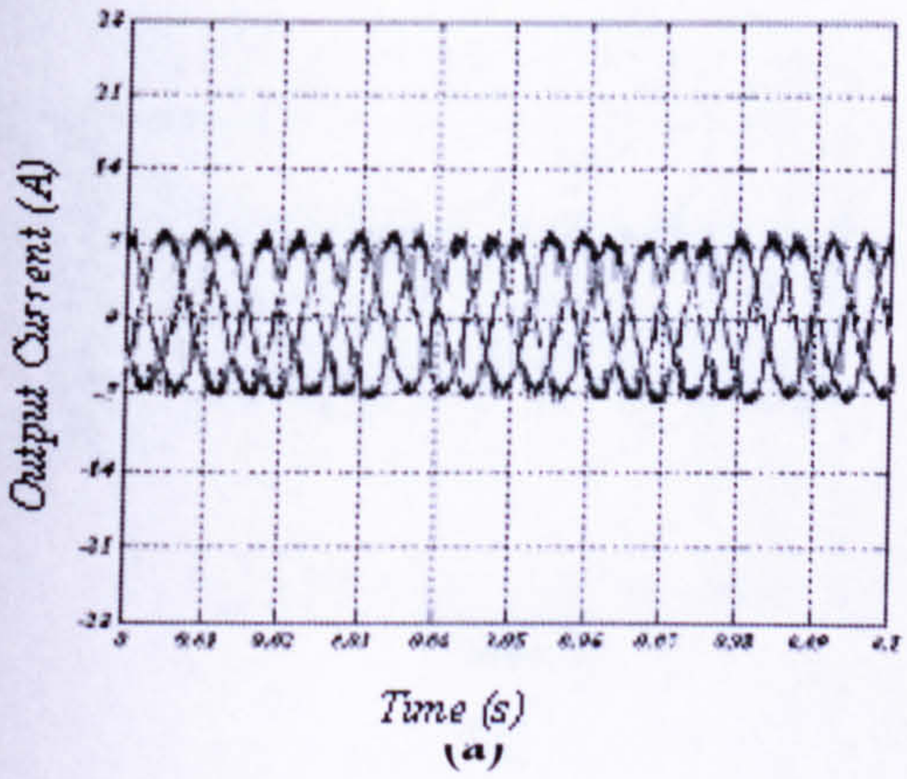
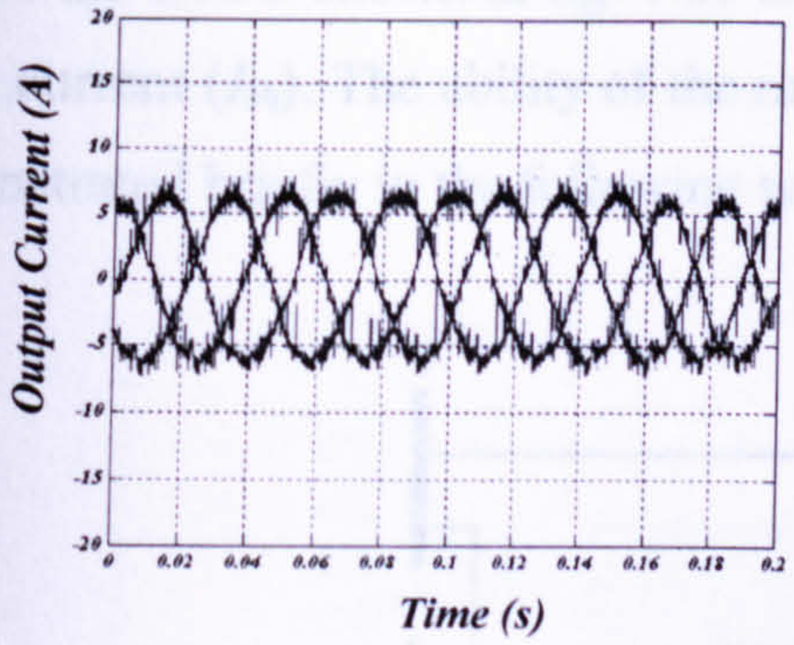
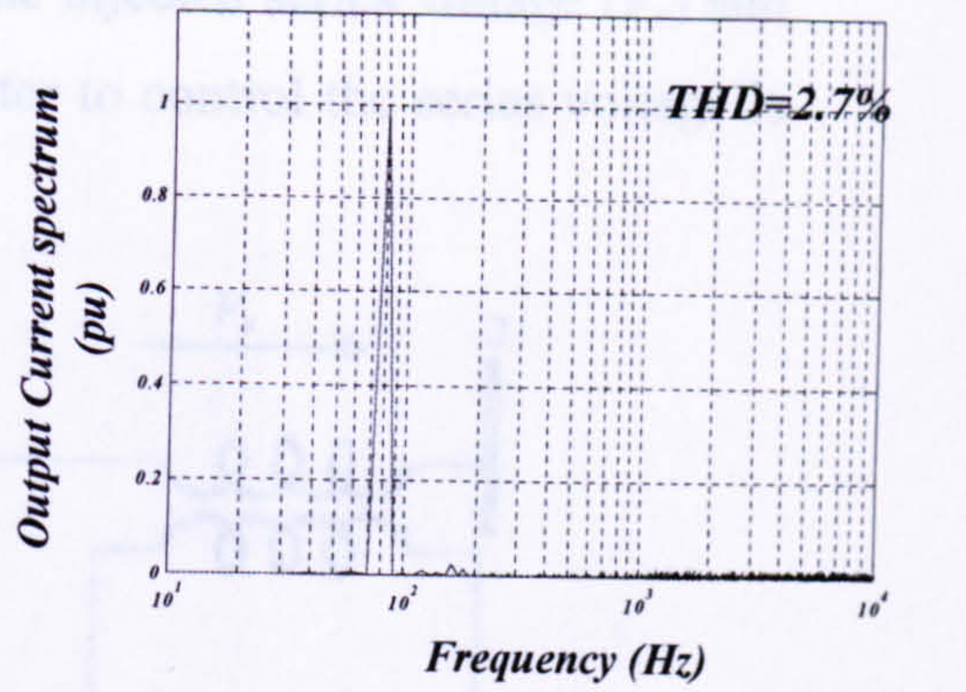


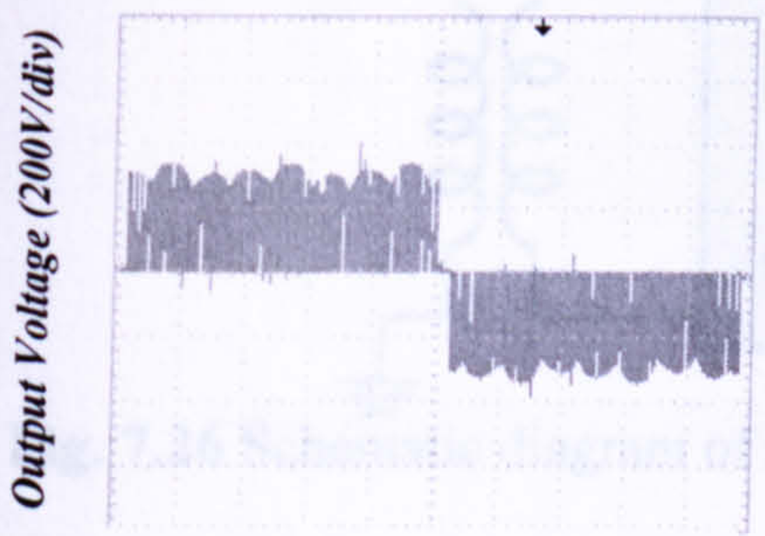
Fig. 7.24 Practical output response for f_o=80 Hz, unity input power factor at f_s= 6.1 kHz for closed loop control: (a) output current, (b) output current harmonic spectrum, (c) output line voltage, (d) filtered input current and input voltage, (e) filtered input current, and (f) input current harmonic spectrum.



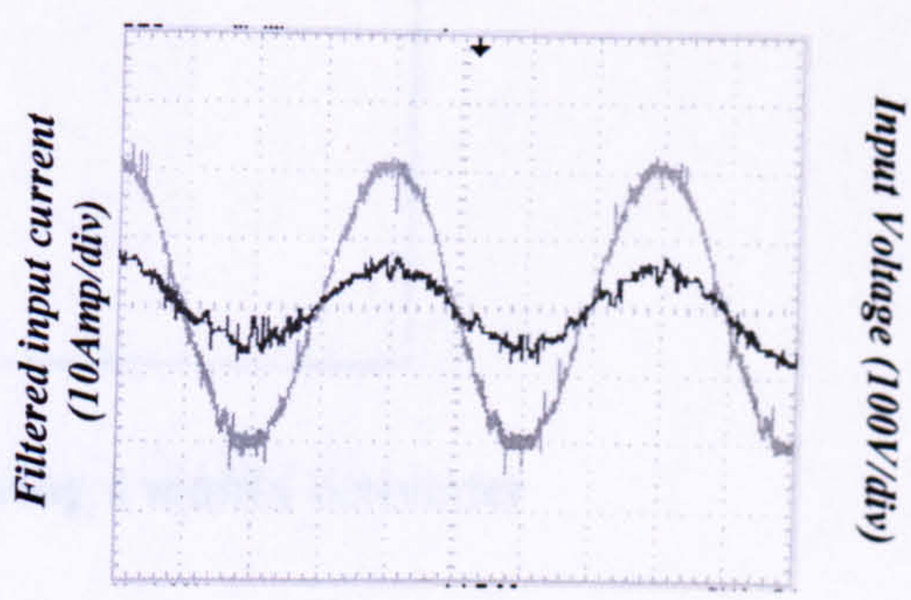
(a)



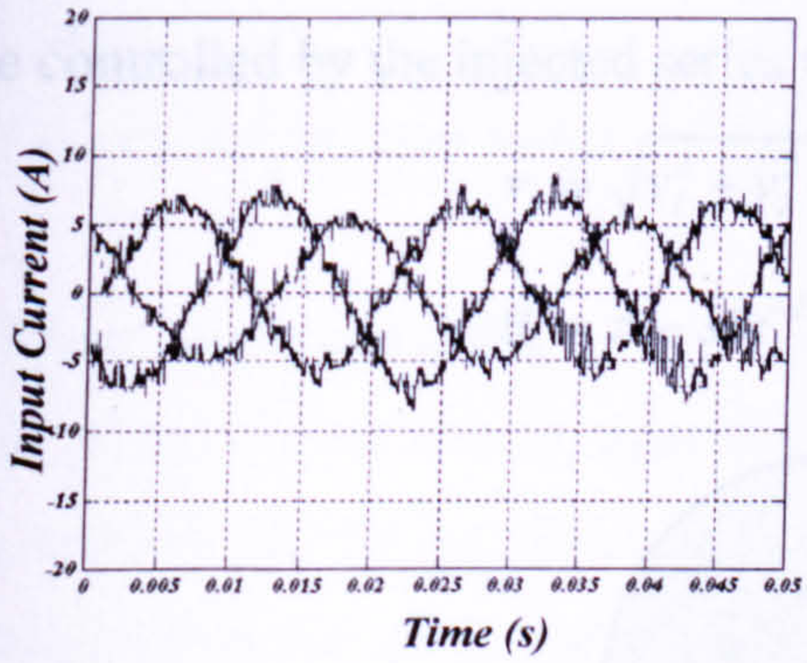
(b)



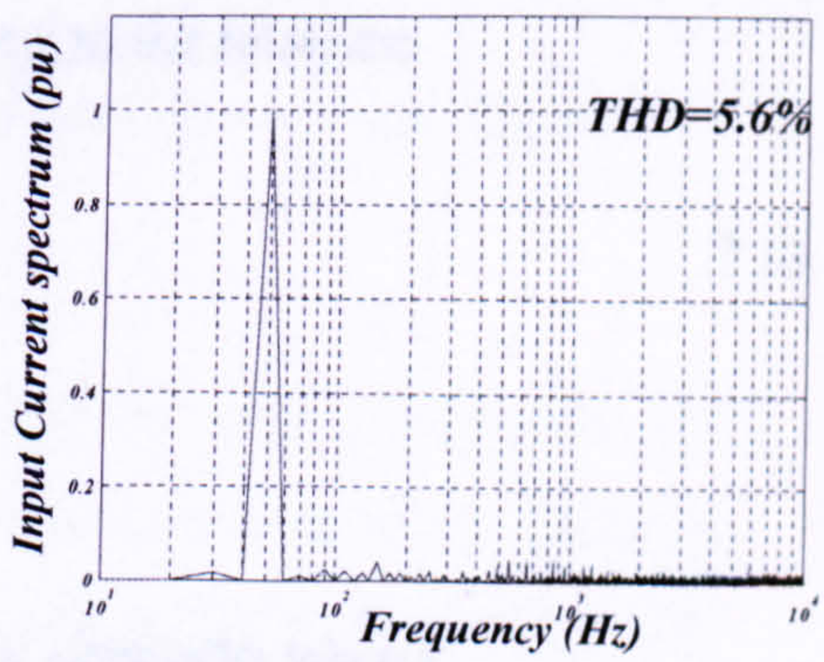
(c)



(d)



(e)



(f)

Fig. 7.25 Practical output response for f_o=20 Hz, unity input power factor at f_s= 6.1 kHz for closed loop control: (a) output current, (b) output current harmonic spectrum, (c) output line voltage, (d) filtered input current and input voltage, (e) filtered input current, and (f) input current harmonic spectrum.

7.3 Unified power flow controller

The matrix converter when used for a unified power flow controller relies on a symmetrical ac voltage supply. Any asymmetry in the voltage will be reflected in either the input or output currents, and deteriorates the line current quality. The basic

idea of the UPFC shown in fig. 7.26 is to control the injected series voltage (V_s) and shunt current (I_{sh}). The ability of the matrix converter to control the series voltage is demonstrated briefly in the following section.

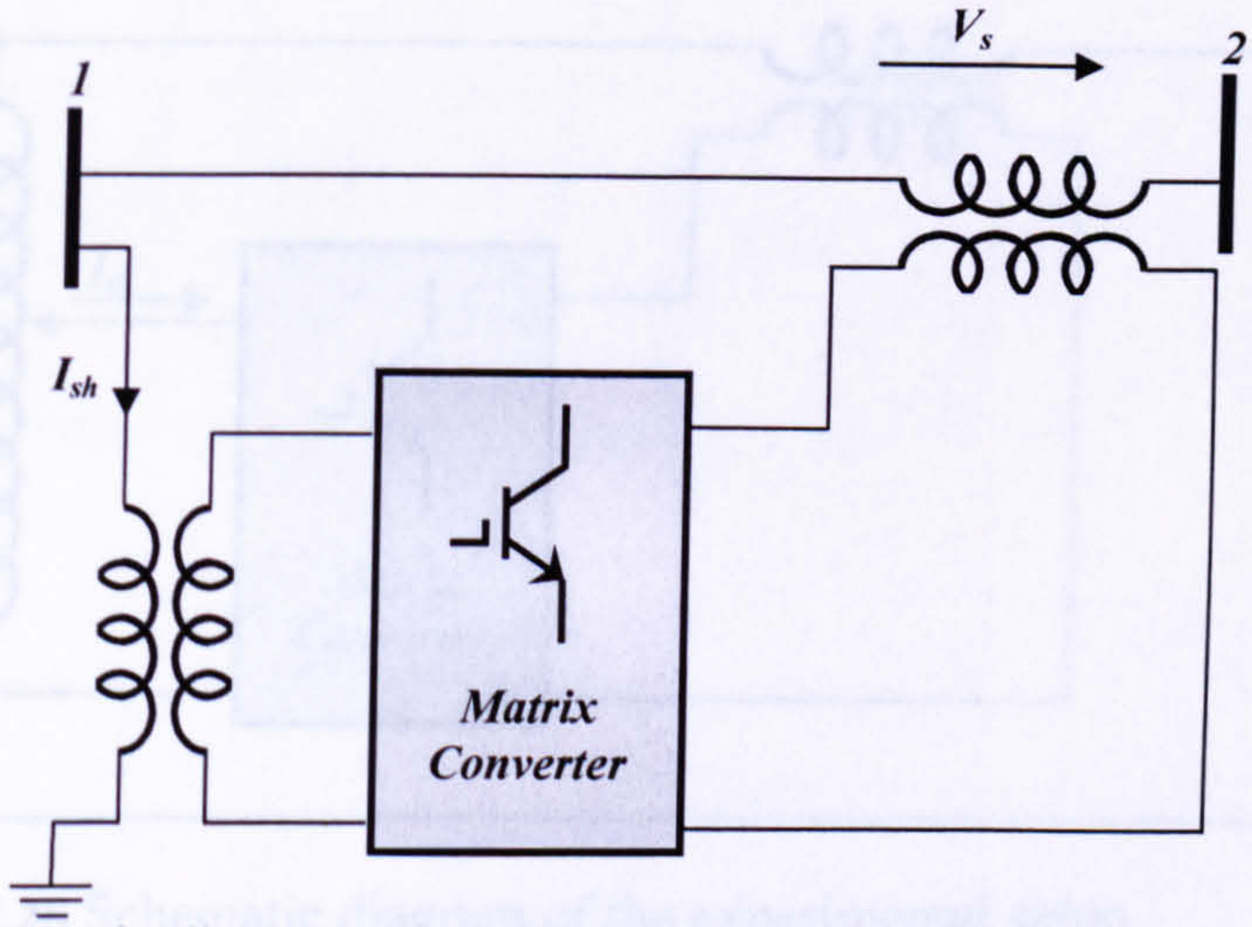


Fig. 7.26 Schematic diagram of a UPFC employing a matrix converter

The main function of the series converter is to control the transmitted power through a transmission line. This can be achieved by regulating the voltage at bus 2 which can be controlled by the injected series voltage according to the relations

$$v_s = \sqrt{v_1^2 + v_2^2 - 2v_1 v_2 \cos\theta}$$

$$\theta_v = \pi - \sin^{-1}\left(\frac{v_2}{v_s}\sin\theta\right) \qquad 7.13$$

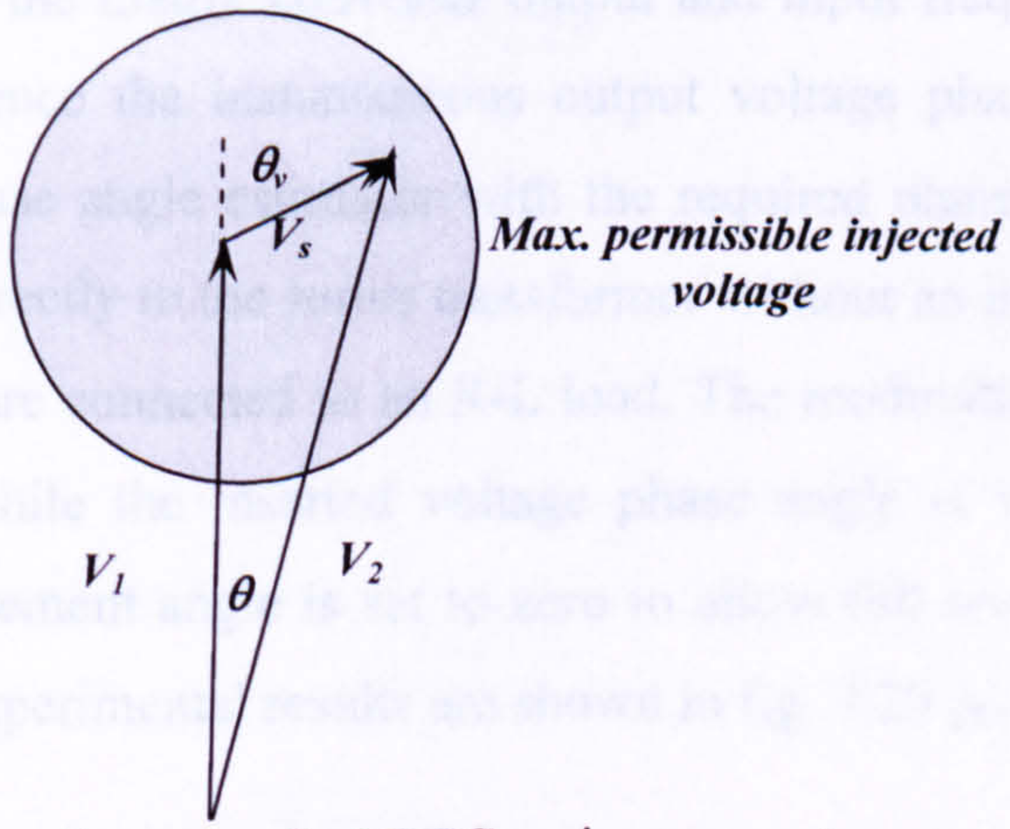


Fig. 7.27 Phasor diagram of a UPFC series converter

The circuit shown in fig. 7.28 is used to demonstrate the ability of the matrix converter to control the injected series voltage. The input side is connected to the

secondary of an autotransformer, while the output side is connected to the line through three single phase transformers.

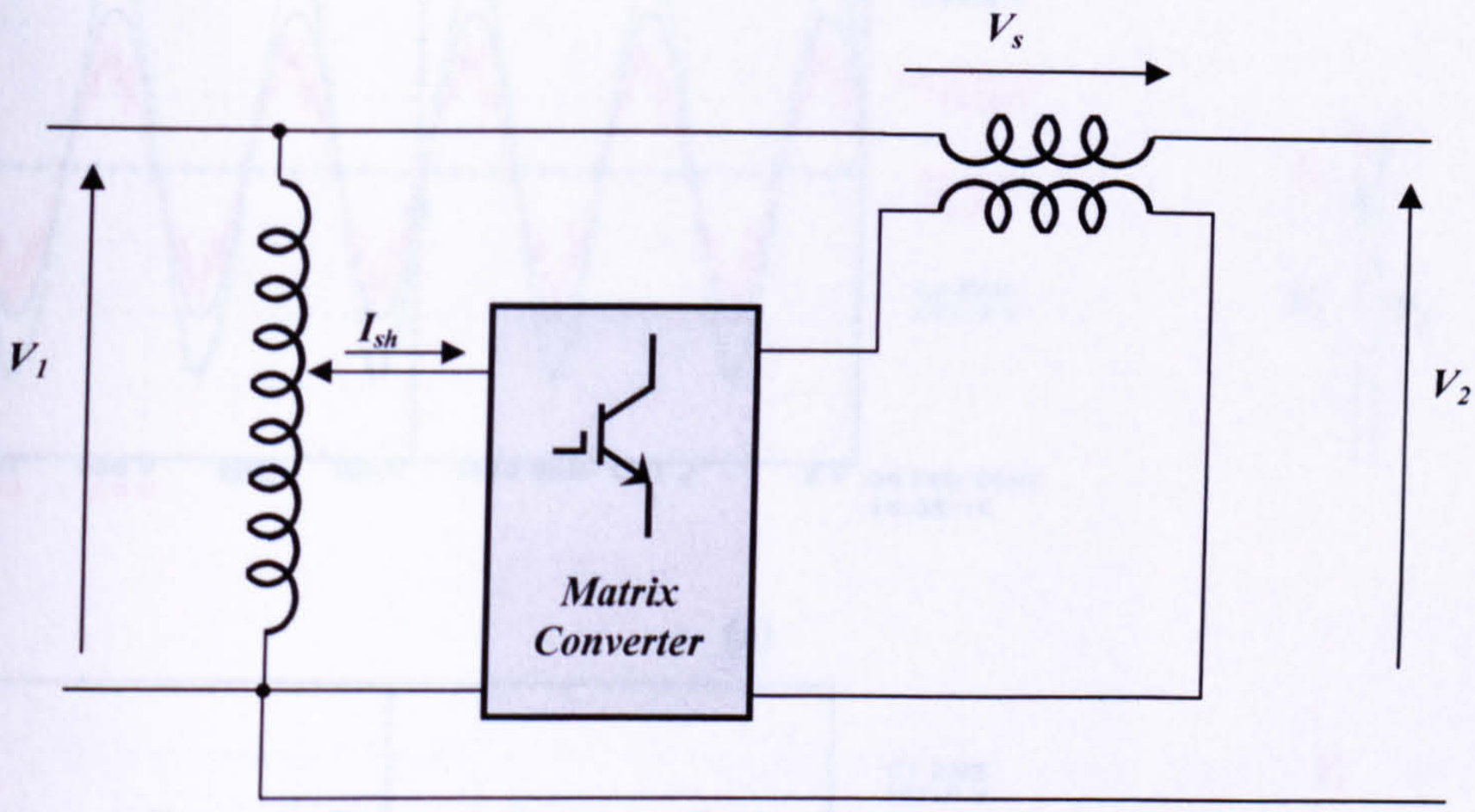


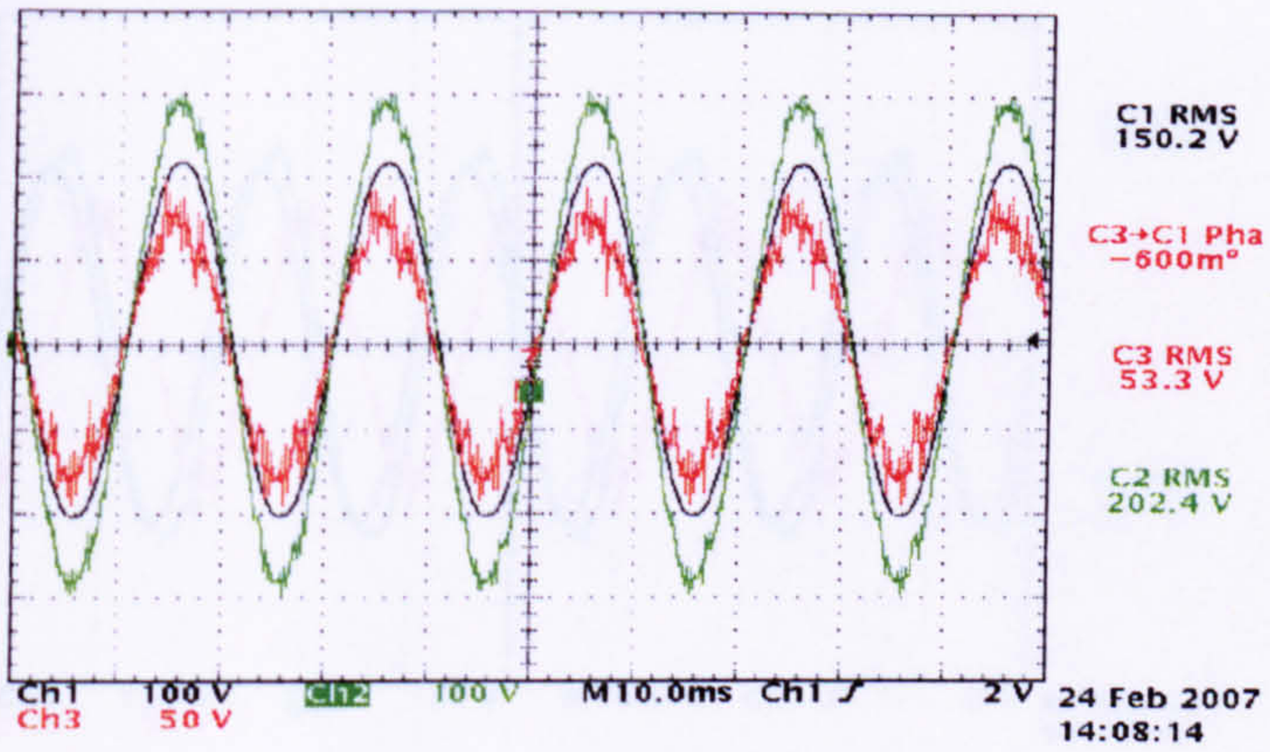
Fig. 7.28 Schematic diagram of the experimental setup.

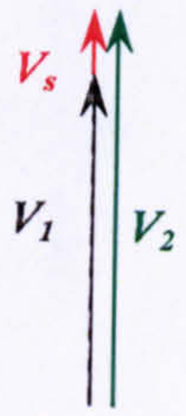
The experiment performance involves two steps (appendix d):

- Maintain the injected series voltage magnitude constant and vary its phase angle.
- Keep the injected series voltage phase angle constant and vary the magnitude.

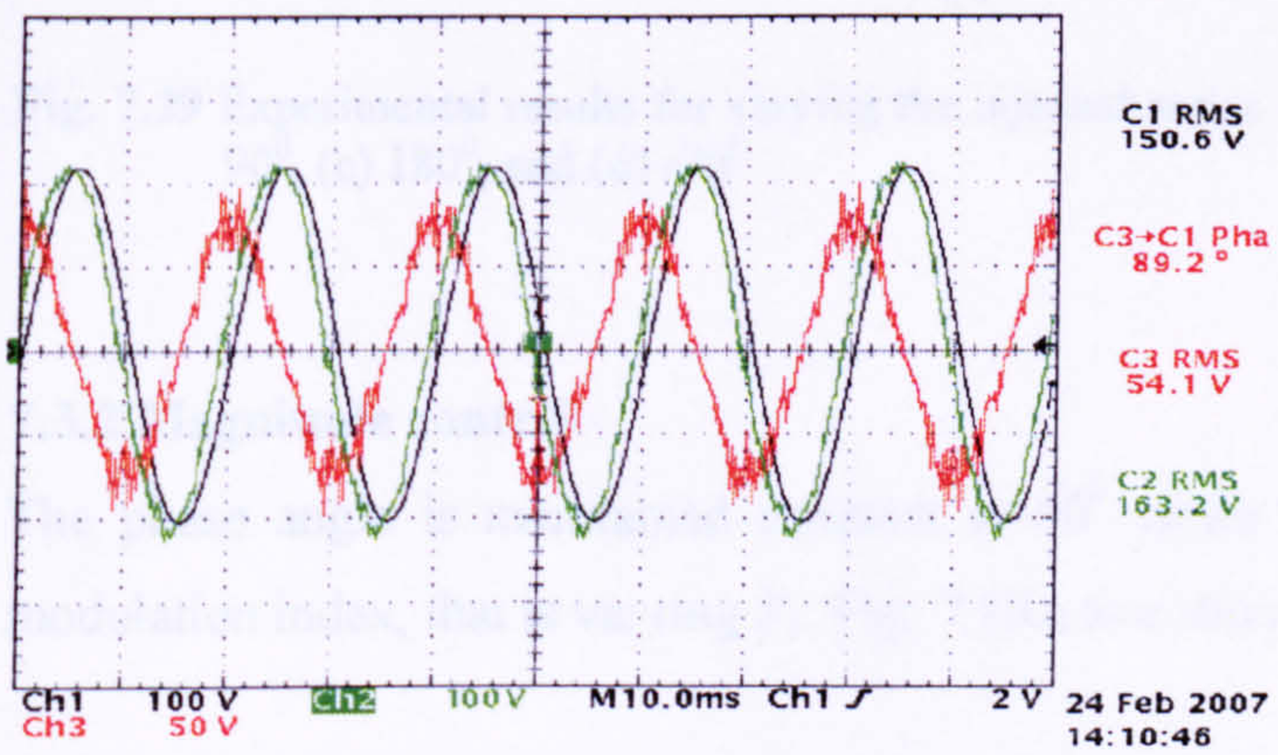
7.3.1 Phase angle control

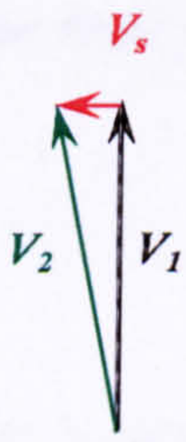
For the UPFC circuit shown, the matrix converter output and input frequencies are equal, around the 50 Hz. Hence the instantaneous output voltage phase angle is calculated using the input phase angle estimator with the required phase shift. The output voltage is connected directly to the series transformer without an intermediate filter, and the load terminals are connected to an R-L load. The modulation index is maintained constant at 0.6 while the inserted voltage phase angle is varied. The matrix converter input displacement angle is set to zero to allow full control of the output voltage. The selected experimental results are shown in fig. 7.29 parts a to d.


C1 RMS
150.2 V
C3→C1 Pha
−600m°
C3 RMS
53.3 V
C2 RMS
202.4 V
Ch1 100 V Ch2 100 V M10.0ms Ch1 ƒ 2 V
Ch3 50 V
24 Feb 2007
14:08:14

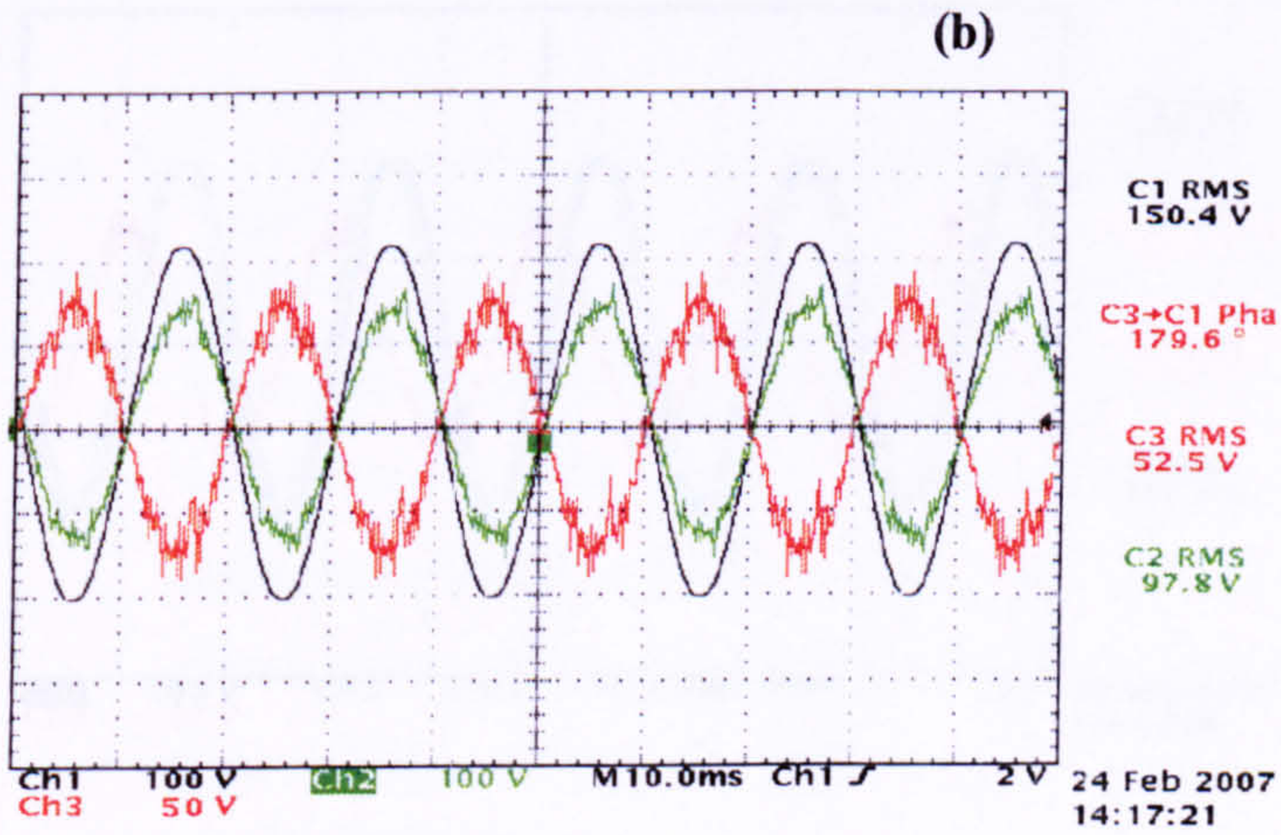

V_s
V_1
V_2

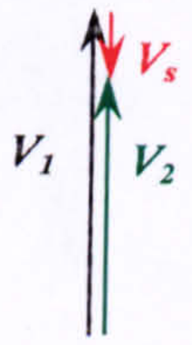
(a)


C1 RMS
150.6 V
C3→C1 Pha
89.2 °
C3 RMS
54.1 V
C2 RMS
163.2 V
Ch1 100 V Ch2 100 V M10.0ms Ch1 ƒ 2 V
Ch3 50 V
24 Feb 2007
14:10:46

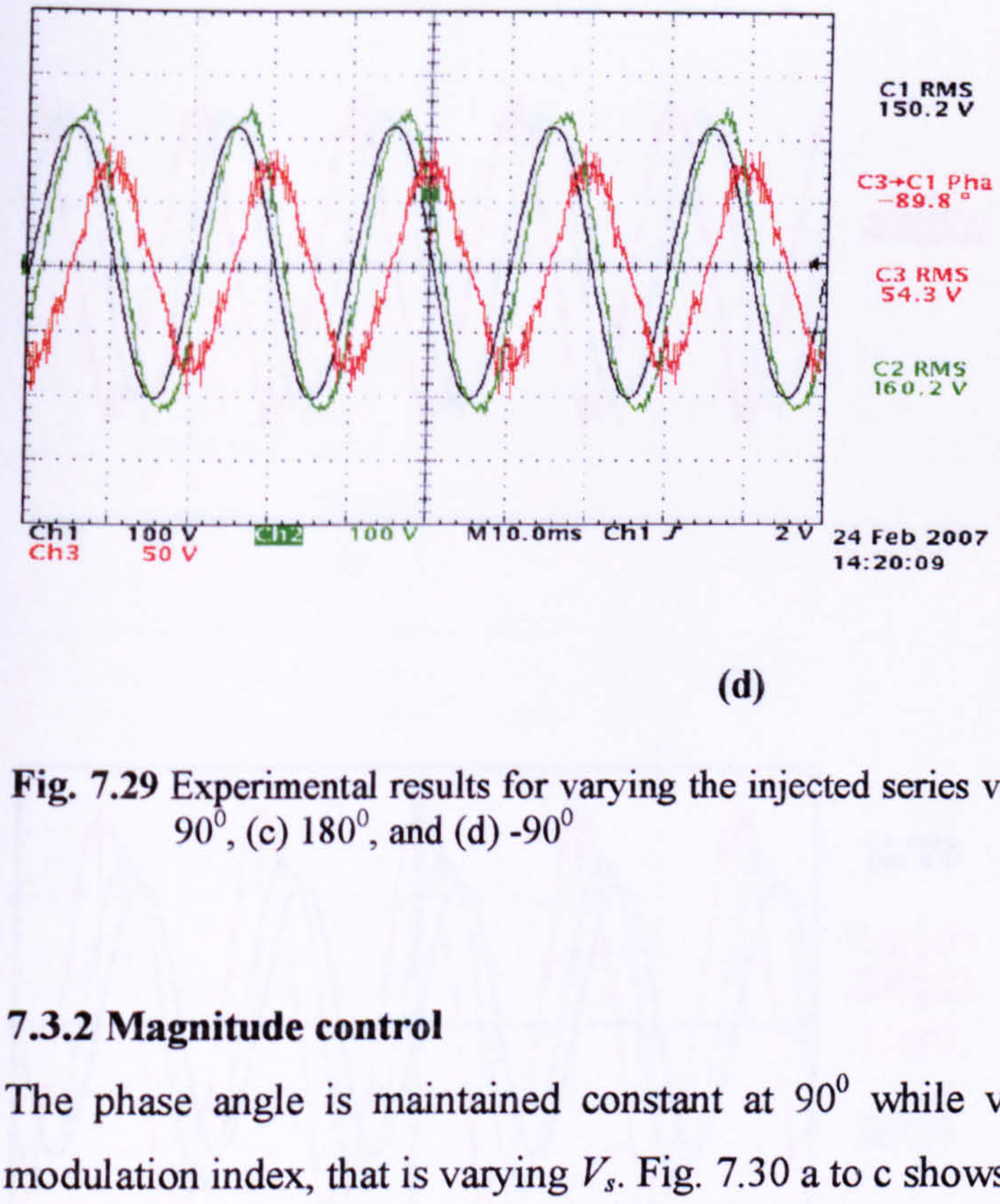

V_s
V_2
V_1

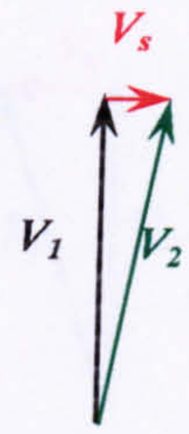
(b)


C1 RMS
150.4 V
C3→C1 Pha
179.6 °
C3 RMS
52.5 V
C2 RMS
97.8 V
Ch1 100 V Ch2 100 V M10.0ms Ch1 ƒ 2 V
Ch3 50 V
24 Feb 2007
14:17:21


V_s
V_1
V_2

(c)



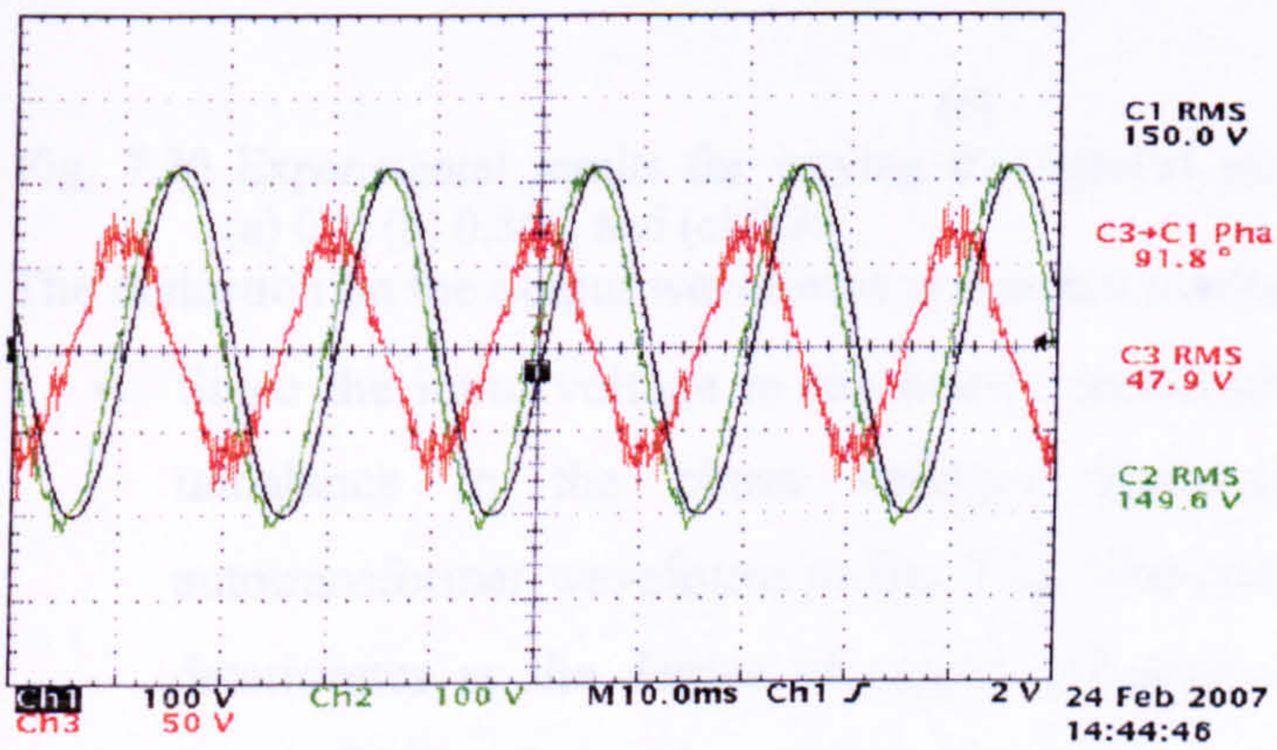


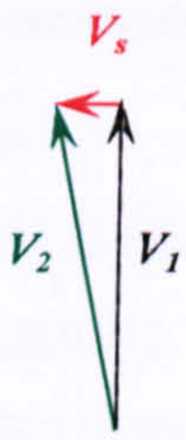
(d)

Fig. 7.29 Experimental results for varying the injected series voltage phase angle: (a) 0^0, (b) 90^0, (c) 180^0, and (d) -90^0

7.3.2 Magnitude control

The phase angle is maintained constant at 90^0 while varying the output voltage modulation index, that is varying V_s. Fig. 7.30 a to c shows the experimental results.





(a)

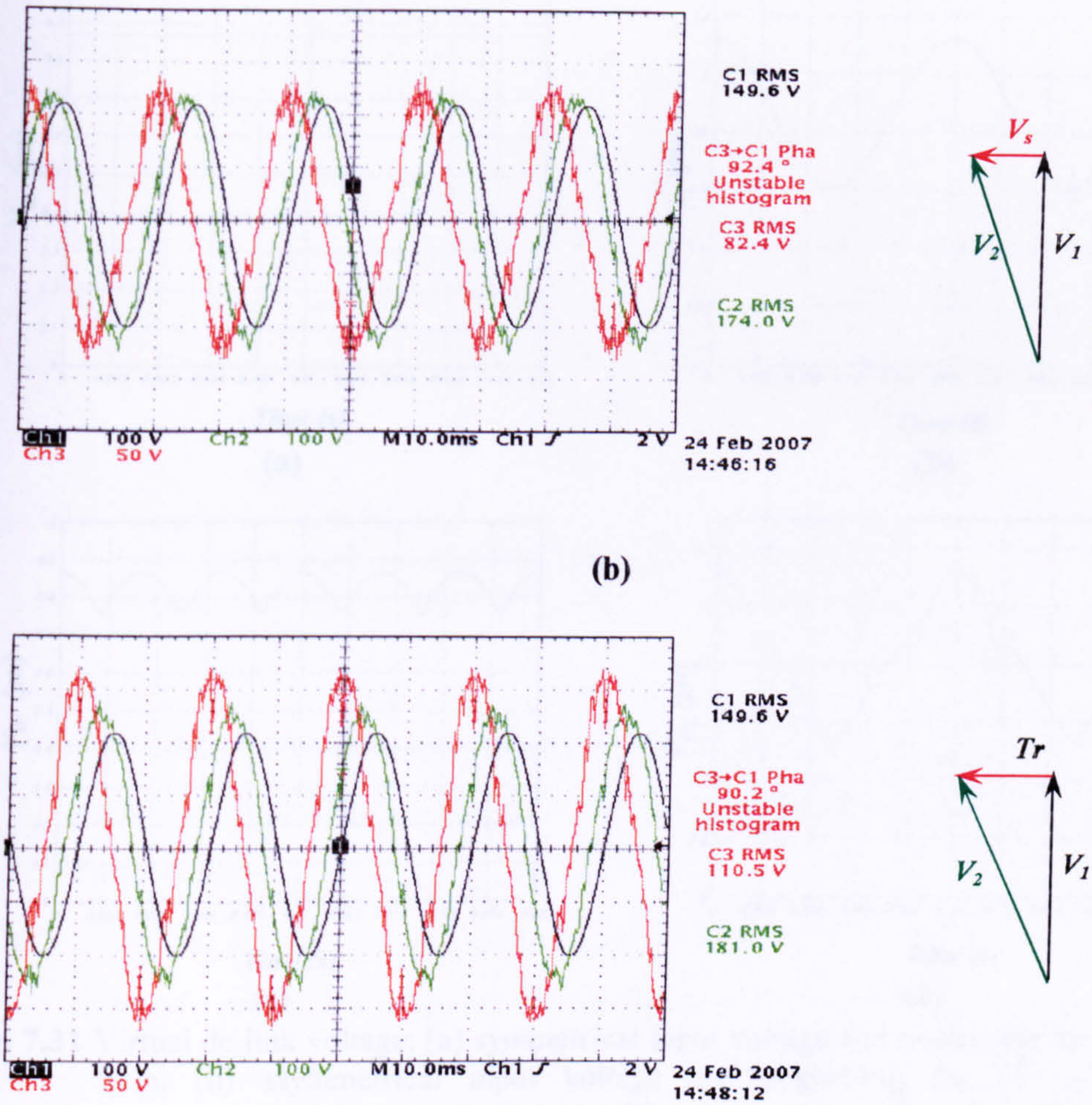


Fig. 7.30 Experimental results for varying the injected series voltage modulation index: (a) 0.5, (b) 0.866, and (c) 0.95.

The distortion on the output waveforms is due to a number of factors:

- Since the input voltage to the matrix converter is reduced, the percentage unbalance in the phase voltages increases as illustrated by the autotransformer waveforms in fig. 7.31. The constancy of the virtual dc link deteriorates as the degree of supply imbalance increases as shown in the figure. This reduces the maximum operating voltage transfer ratio, which means that the matrix converter is likely to be operating in the over modulation region.

$$v_{DC} = \left(\hat{V}_{ip} \cdot \frac{\sqrt{3}}{2} \cos\phi + \hat{V}_{in} \cdot \frac{\sqrt{3}}{2} \cos(2\omega t - \phi + \varphi) \right) \frac{T_{s\,eff}}{T_s} \qquad 7.14$$

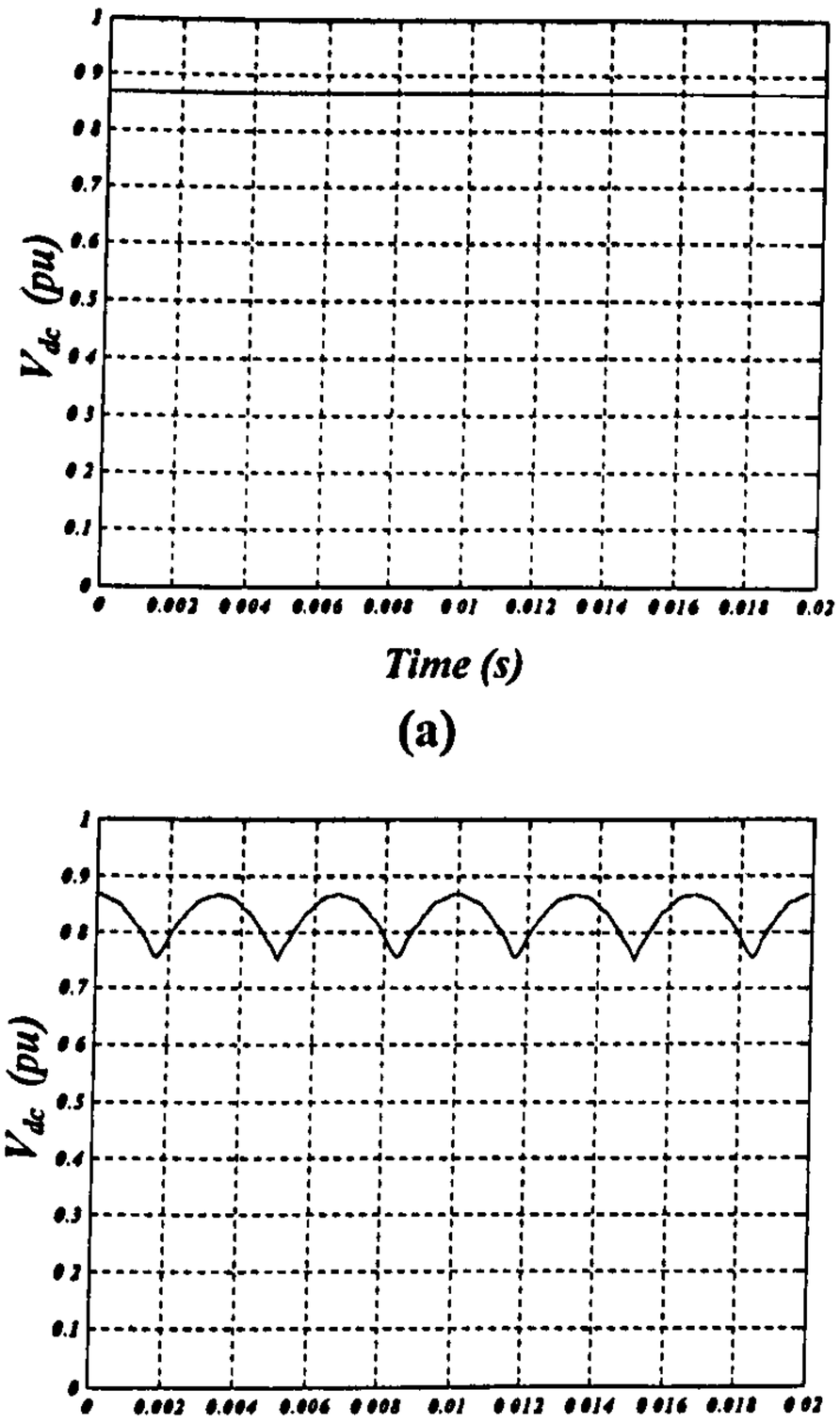


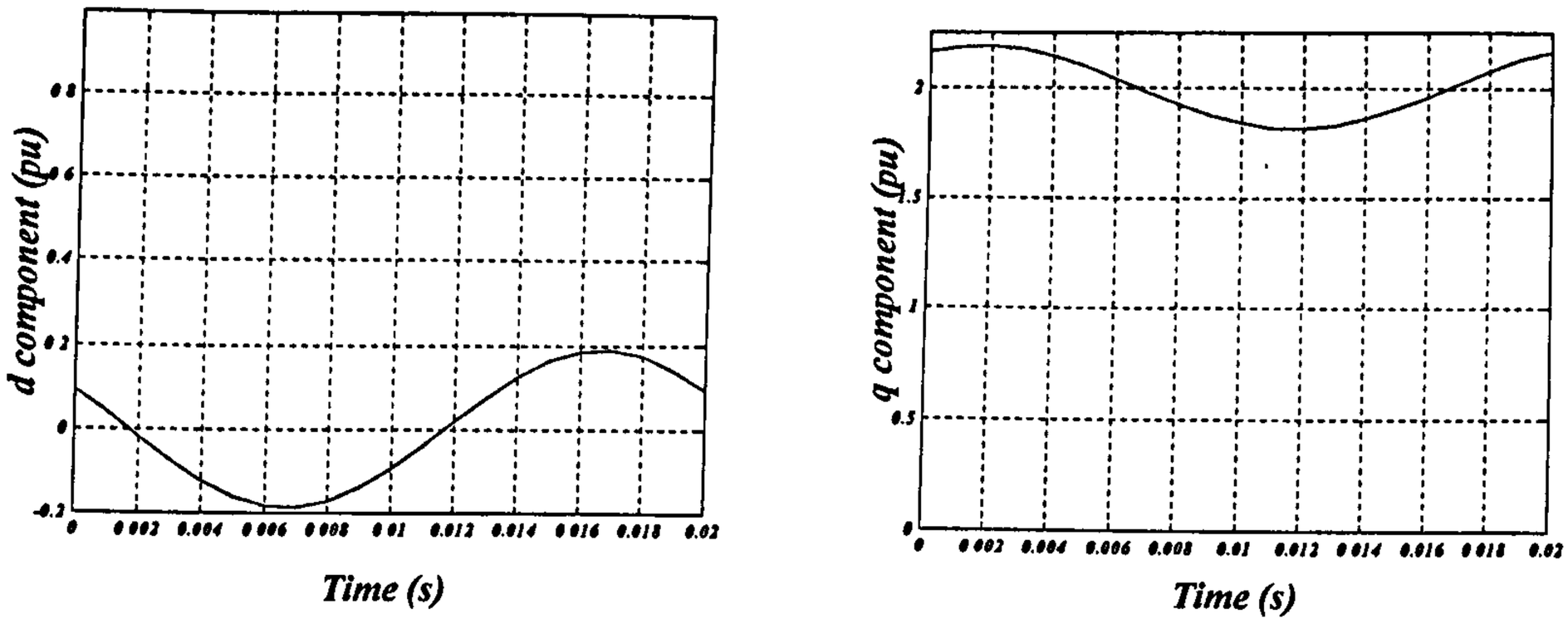
Fig. 7.31 Virtual dc link voltage: (a) symmetrical input voltage and neglecting the $(T_{s\,eff}/T_s)$ term, (b) asymmetrical input voltage and neglecting the $(T_{s\,eff}/T_s)$ term, (c) symmetrical input voltage and taking into account the $(T_{s\,eff}/T_s)$ term, and (d) symmetrical input voltage and taking into account the $(T_{s\,eff}/T_s)$ term (30% unbalance).

- The measurement transducer offset effects at low voltage is more dominant.

  Considering the *dq* components of three-phase ac signals affected by dc offsets

$$v_i(t) = V_i \begin{bmatrix} a + cos(\omega_i t) \\ b + cos\left(\omega_i t + {}^{2\pi}/_{3}\right) \\ c + cos\left(\omega_i t + {}^{4\pi}/_{3}\right) \end{bmatrix} \qquad 7.15$$

The *dq* components in this case are a combination of two signals: a dc component due to the sinusoidal part and a sinusoid signal at 50 Hz due to the offsets.



(a) (b)

Fig. 7.32 Transducer offset effect:
(a) d component in case of 10% offset in the transducers.
(b) q component in case of 10% offset in the transducers.

- The noise effect on the measured signal is more dominant at low input voltages. Transducers should be operated as close as possible to full scale deflection.
- The filter for the series injected voltage used is an L-filter. Using an LCL filter will reduce the effect of the distortion in the output waveforms.

It should be noted that the matrix converter in the second case is operating as a series VAr compensator by adjusting the injected series voltage magnitude. For series compensation, the matrix converter input displacement angle is null, and the matrix converter output voltage phase angle is set to 90^0, and its magnitude is calculated from

$$V_s = V_1 \tan\theta \qquad 7.16$$

Since the output voltage phase angle is maintained at 90^0, the matrix converter output active power is zero, hence its input power is equivalent to its internal losses (semiconductor and component losses), hence the input current is low. Also when the output voltage is uncontrolled, its magnitude is given by:

$$V_2 = \sqrt{V_1^2 + V_s^2} \qquad 7.17$$

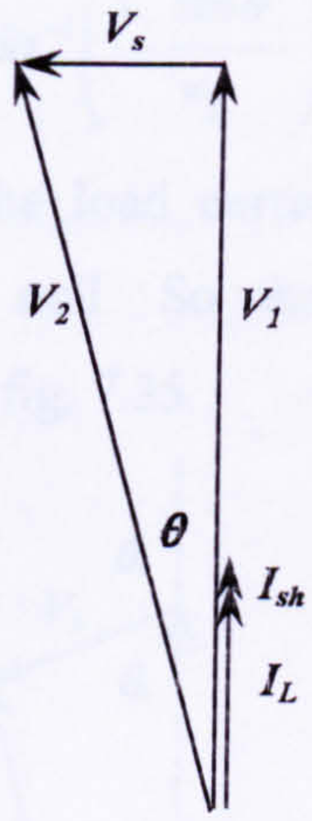


Fig. 7.33 Series compensator phasor diagram.

These relations are valid for a symmetrical voltage supply. In case of imbalance in the supply voltages, closed loop control is necessary to adjust the series injected voltage. The input voltage imbalance will be reflected in the input current even with closed loop control due to the absence of any energy storage element. Iteration gives the required phase displacement as achieved in fig. 7.34.

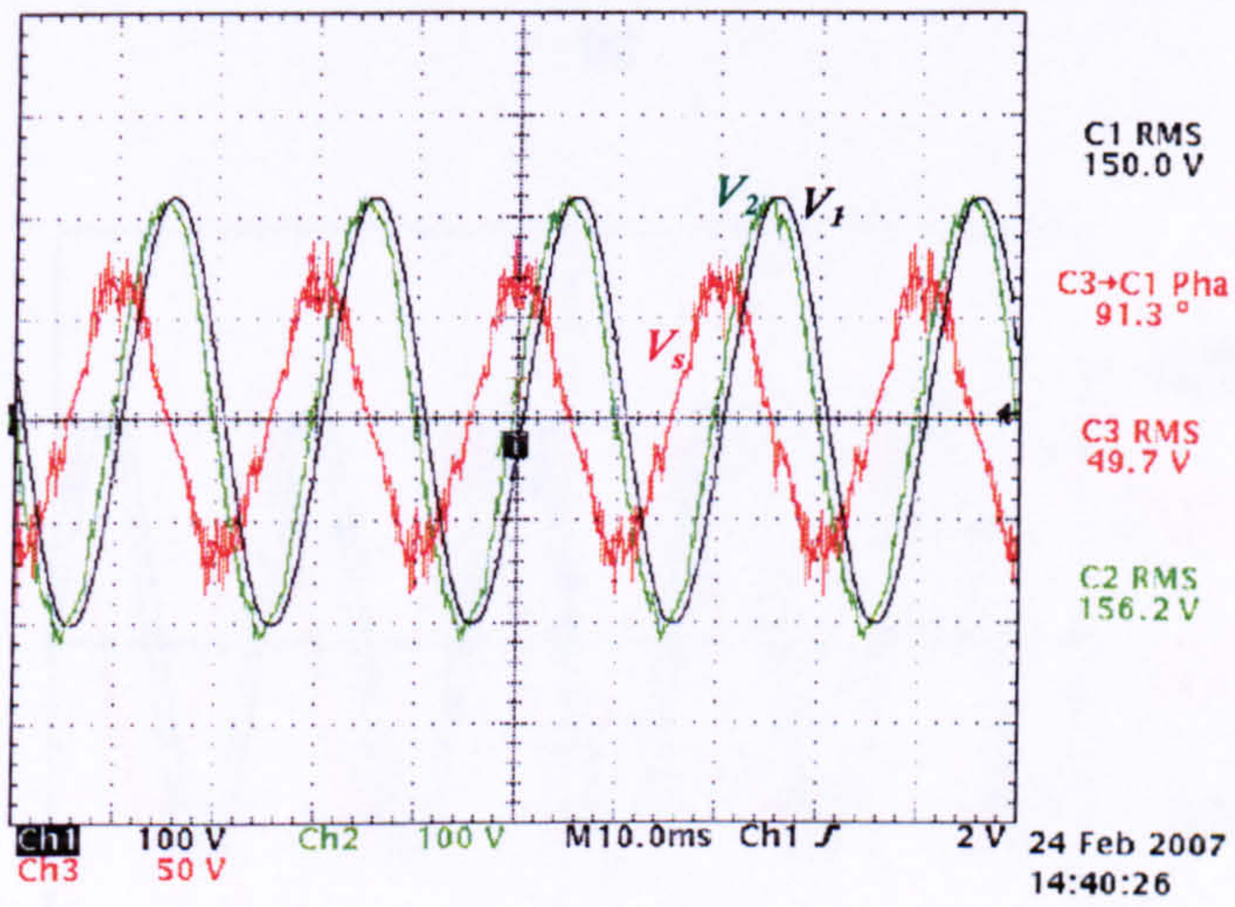


Fig. 7.34 Experimental results for a series compensator.

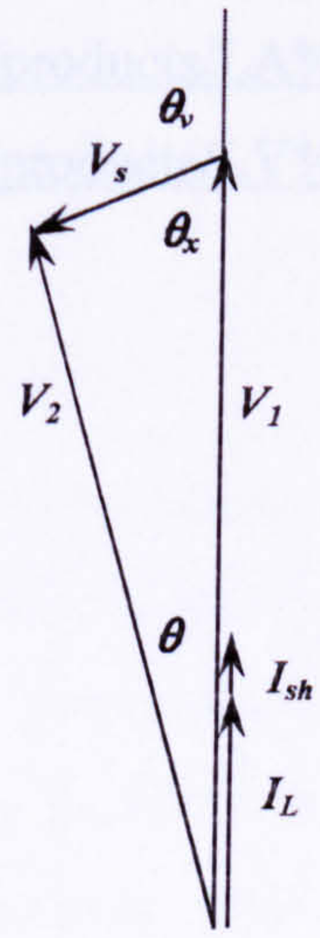
In order to control the load voltage magnitude as well, both the injected series voltage angle and magnitude should be controlled. The series voltage magnitude is calculated from

$$v_s = \sqrt{v_1^2 + v_2^2 - 2v_1 v_2 \cos\theta} \qquad 7.18$$

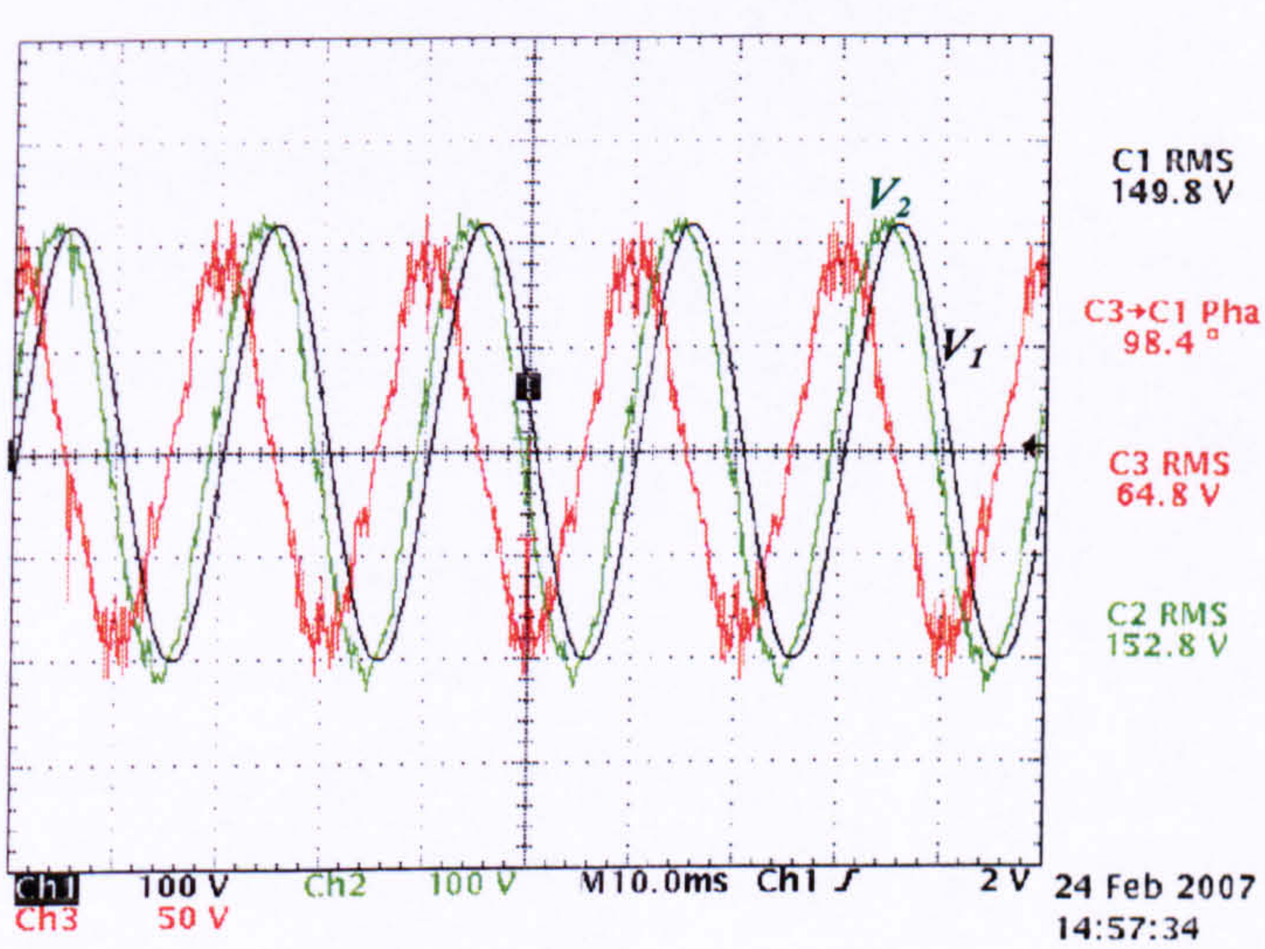
and its phasor angle is

$$\theta_v = \pi - \sin^{-1}\left(\frac{v_s \sin\theta}{v_2}\right) \qquad 7.19$$

Since the injected voltage and the load current are not orthogonal, the matrix converter output power is not null. So the input current is not minimal. Compensation results are shown in fig. 7.35.



(a)



(b)

Fig. 7.35 Modified series compensator phasor diagram:
(a) phasor diagram and
(b) experimental results.

References:

[7.1] http://focus.ti.com/docs/toolsw/folders/print/tmds320006711.html.

[7.2] http://www.signalware.com/dsp/AED_106a.htm.

[7.3] http://focus.ti.com/lit/ds/symlink/ths1206.pdf.

[7.4] http://focus.ti.com/lit/ds/symlink/ths4052.pdf.

[7.5] http://www.fairchildsemi.com/ds/HG%2FHGTG12N60A4D.pdf

[7.6] http://www.lem.com/docs/products/LA%20100-P.pdf

[7.7] http://www.lem.com/docs/products/LV%2025-P%20E.pdf

CHAPTER EIGHT

CONCLUSION

8.1 General conclusions

The matrix converter offers a practical replacement for back-to-back converters with less weight and size due to the absence of the intermediate stage storage elements. Its virtual dc link voltage magnitude is proportional to the output modulation index. This gives higher resolution and better performance at low modulation indices for a symmetrical ac voltage supply, when compared with back-to-back converters.

An approach has been proposed in chapter three which directly converts from back-to-back VSI switch states to the switch states of the matrix converter and vice versa. The approach offers explanation of matrix converter operation in terms of back-to-back converters operation, and vice versa.

The absence of any intermediate storage elements, which is seen as the main advantage of the matrix converter, has some drawbacks. The input and output active powers are not decoupled; this means that any transient in either side of the converter is reflected directly to the other side. The effective dc link voltage is less than that of back-to-back converters; hence the maximum output voltage margin is lower. This margin is also affected by the input displacement angle. Furthermore, an asymmetrical ac voltage supply reduces the matrix converter maximum transfer ratio and is detrimental to its performance. This is not the case for back-to-back converters.

With an asymmetrical ac voltage supply and without feedback control on both sides of the converter, the voltage oscillation on the virtual dc link deteriorates the performance of both the input and output. For this reason, fast controllers are required to produce symmetrical currents at either the input or the output.

For the conventional matrix converter, it was shown in chapter four, that it is not possible to obtain symmetrical currents at both the input and output ends. If symmetrical input and output currents are required, a fictious intermediate storage element is implied to decouple the input and output active power.

The bidirectional properties of the matrix converter can be used in many power system applications such as in embedded generation and for a unified power flow controller. Even under asymmetrical grid voltage connection for embedded generation, it was shown in chapter five that the matrix converter can provide balanced output current, but this produces oscillatory active power at both the input and output. Using dual vector current controllers, non-oscillatory active power transfer can be achieved.

The active and reactive power transmitted through the transmission system can be controlled using the matrix converter as a UPFC. It can also control the voltage magnitude at any point of a transmission system. Both voltage magnitude and power control using the matrix converter can be achieved with certain limitations, being restricted by the lack of any intermediate energy storage element, as considered in chapter six.

These conclusions are summarised in table 8.1.

	Conventional Matrix Converter	Back-to-Back Converters
Dc link voltage	• Max. 0.866 V_{im}. • The dc voltage constancy is affected by the asymmetry in either the input or output side voltage. • Proportional to the output modulation index, which gives better performance at low modulation indices.	• No max. limit on the voltage magnitude. • The dc voltage constancy depends mainly on the capacitance. • Not dependent on the output modulation index. Same performance at high and low modulation indices requires changing the link capacitance.

Asymmetrical i/p supply	• Coupling between the input and output active power. • Both input and output currents are deteriorated without closed loop control. • Either symmetrical input or output currents can be achieved and it requires fast controller.	• Decoupling between input and output active power can be easily achieved. • Symmetrical input and output currents can be obtained without closed loop control.
Embedded generation applications	• Under symmetrical supply, high current performance can be achieved at both ends at any modulation index. • Asymmetrical grid voltage requires dual current controller to achieve non oscillatory output power, hence non oscillatory input power.	• Under symmetrical supply, high current performance can be achieved at high modulation indices. The performance reduces at low modulation indices. • Asymmetrical grid voltage does not require any current controller modification since the output and input powers are decoupled. Constant output power requires dual current controller.
UPFC applications	• Either the transmitted power through the line or voltage magnitude control can be achieved. • Control of both voltage and power transmitted is restricted. • The performance is affected by grid voltage asymmetry.	• Control of both voltage and power transmitted can be achieved without any limitations. • The performance is not affected by grid voltage asymmetry.

Table 8.1 Suitability of matrix and back to back converters to power system applications.

8.2 Author's Contribution

A new matrix representation approach to control the matrix converter has been introduced in chapter three. The approach depends mainly on output current control feedback and the use of logic circuitry to perform the matrix converter functions. The introduction of a virtual dc link has been used to explain output voltage magnitude limitations, and the relationship between input and output current harmonics. Logic circuitry is used for most functions and for the conversion from a back-to-back converter to a matrix converter. This reduces the DSP computations required.

In addition, as considered in chapter three, the decoupled matrix approach exposes possibilities for reducing the average number of the switching commutations per cycle which increases converter efficiency and reduces device stresses. Furthermore, the matrix approach offers a basis for alternative explanation and interpretation of matrix converter operation.

The technique enables the removal of asymmetrical voltage effects from the output current within certain limits, depending on the robustness of the output current controller and the level of input asymmetry whilst lowering the achievable modulation index. In chapter four, matrix converter operation has been analysed under asymmetrical input voltage conditions, for both unbalanced and a flat-topped input ac voltage supply. For the first time, it has been proved mathematically, that under asymmetrical input voltage conditions, it is not possible to obtain symmetrical currents at both converter ends, without an intermediate energy storage element. Either obtaining symmetrical input or output currents require a fast robust controller, at the expense of the current quality at the uncontrolled end.

In chapter five, the performance of the matrix converter connected to the grid through an L-filter under balanced and unbalanced grid conditions was evaluated. Non-oscillatory output power is obtained for embedded generation even with asymmetrical grid voltages, by using a dual current controller to control both the positive and negative sequence current components. Extraction of the components delays the *dq* components by 90^0.

The ability of the matrix converter to control either the voltage magnitude or the transmitted active and reactive power through the transmission line has been demonstrated in chapter six. The matrix converter limitation to control both the voltage magnitude and the transmitted power simultaneously as a unified power flow controller due to the coupling between the input and output active powers has been explored.

8.3 Future research

The complete rectifier stage can be implemented using logic circuitry (FPGA) including the phase displacement angle and frequency estimator. This represents a

final step towards transforming from the six switch states of the standard VSI to the eighteen switches of the matrix converter.

In the case of back-to-back converters, a study of the smallest capacitance required to decouple the input and output active power and its relationship with the degree of asymmetry in the input ac voltage supply could be undertaken.

Switch commutation sequences can be imbedded into the rectifier and inverter matrices before transforming them to the eighteen switch states of the matrix converter. This approach can combine and embed dead banding and overlap into the transition matrices. If the transition is in the inverter matrix, the sequence of the inverter matrices during commutation will be:

$$a=\begin{bmatrix}1 & 0\\ 0 & 1\\ 1 & 0\end{bmatrix}, b=\begin{bmatrix}0 & 0\\ 0 & 1\\ 1 & 0\end{bmatrix}, c=\begin{bmatrix}1 & 1\\ 0 & 1\\ 1 & 0\end{bmatrix}, d_1=\begin{bmatrix}0 & 1\\ 0 & 1\\ 1 & 0\end{bmatrix}, \text{and } d_2=\begin{bmatrix}0 & 1\\ 0 & 1\\ 1 & 0\end{bmatrix}$$

If the rectifier matrix is:

$$e=\begin{bmatrix}1 & 0 & 0\\ 0 & 1 & 0\end{bmatrix}$$

the matrices of the commutation sequence will be as in tables 8.2 and 8.3.

	S_1 S_2
Initial state	$\begin{bmatrix}1 & 0 & 0 & 1 & 0 & 0\\ 0 & 1 & 0 & 0 & 1 & 0\\ 1 & 0 & 0 & 1 & 0 & 0\end{bmatrix} = [a.e \quad a.e]$
1	$\begin{bmatrix}1 & 0 & 0 & 0 & 0 & 0\\ 0 & 1 & 0 & 0 & 1 & 0\\ 1 & 0 & 0 & 1 & 0 & 0\end{bmatrix} = [a.e \quad b.e]$
2	$\begin{bmatrix}1 & 1 & 0 & 0 & 0 & 0\\ 0 & 1 & 0 & 0 & 1 & 0\\ 1 & 0 & 0 & 1 & 0 & 0\end{bmatrix} = [c.e \quad b.e]$
3	$\begin{bmatrix}0 & 1 & 0 & 0 & 0 & 0\\ 0 & 1 & 0 & 0 & 1 & 0\\ 1 & 0 & 0 & 1 & 0 & 0\end{bmatrix} = [d_1.e \quad b.e]$
4	$\begin{bmatrix}0 & 1 & 0 & 0 & 1 & 0\\ 0 & 1 & 0 & 0 & 1 & 0\\ 1 & 0 & 0 & 1 & 0 & 0\end{bmatrix} = [d_1.e \quad d_2.e]$

Table 8.2 Matrix converter commutation incorporated into the inverter matrix

In case of transition in the rectifier matrix, the sequence of the rectifier matrices will be:

$$e=\begin{bmatrix}1 & 0 & 0\\ 0 & 1 & 0\end{bmatrix},\ f=\begin{bmatrix}1 & 0 & 0\\ 0 & 0 & 0\end{bmatrix},\ g=\begin{bmatrix}1 & 0 & 0\\ 0 & 1 & 1\end{bmatrix} \text{ and } h=\begin{bmatrix}1 & 0 & 0\\ 0 & 0 & 1\end{bmatrix}$$

and the inverter matrix is:

$$a=\begin{bmatrix}1 & 0\\ 0 & 1\\ 1 & 0\end{bmatrix}$$

	S_1 $\quad$ S_2
Initial state	$\begin{bmatrix}1 & 0 & 0 & 1 & 0 & 0\\ 0 & 1 & 0 & 0 & 1 & 0\\ 1 & 0 & 0 & 1 & 0 & 0\end{bmatrix}=\begin{bmatrix}a.e & a.e\end{bmatrix}$
1	$\begin{bmatrix}1 & 0 & 0 & 1 & 0 & 0\\ 0 & 1 & 0 & 0 & 0 & 0\\ 1 & 0 & 0 & 1 & 0 & 0\end{bmatrix}=\begin{bmatrix}a.e & a.f\end{bmatrix}$
2	$\begin{bmatrix}1 & 0 & 0 & 1 & 0 & 0\\ 0 & 1 & 1 & 0 & 0 & 0\\ 1 & 0 & 0 & 1 & 0 & 0\end{bmatrix}=\begin{bmatrix}a.g & a.f\end{bmatrix}$
3	$\begin{bmatrix}1 & 0 & 0 & 1 & 0 & 0\\ 0 & 0 & 1 & 0 & 0 & 0\\ 1 & 0 & 0 & 1 & 0 & 0\end{bmatrix}=\begin{bmatrix}a.h_1 & a.f\end{bmatrix}$
4	$\begin{bmatrix}1 & 0 & 0 & 1 & 0 & 0\\ 0 & 0 & 1 & 0 & 0 & 1\\ 1 & 0 & 0 & 1 & 0 & 0\end{bmatrix}=\begin{bmatrix}a.h_1 & a.h_2\end{bmatrix}$

Table 8.3 Matrix converter commutation incorporated into the rectifier matrix

Tuning of the controller parameters and redesign of the input filter can reduce the input current distortion. This will facilitate the practical implementation of the matrix converter as a unified power flow controller and its use in embedded generation systems.

The use of an LCL filter interface instead of the L filter used can improve the quality of the output current. The controller equations will be:

$$v^{*}_{cd,k} = e_{d,k} + R_1 \sum_{n=0}^{k}\left(i^{*}_{1d,n} - i_{1d,n}\right) + \left(\frac{L_1}{T_s} + \frac{R_1}{2}\right).\left(i^{*}_{1d,k} - i_{1d,k}\right) - \omega L_1\left(\frac{i^{*}_{1q,k} + i_{1q,k}}{2}\right)$$

$$v^{*}_{cq,k} = e_{q,k} + R_1 \sum_{n=0}^{k}\left(i^{*}_{1q,n} - i_{1q,n}\right) + \left(\frac{L_1}{T_s} + \frac{R_1}{2}\right).\left(i^{*}_{1q,k} - i_{1q,k}\right) + \omega L_1\left(\frac{i^{*}_{1d,k} + i_{1d,k}}{2}\right) \quad 8.1$$

$$i^{*}_{2d,k} = i_{1d,k} + \left(\frac{C}{T_s}\right).\left(v^{*}_{cd,k+1} - v_{cd,k+1}\right) - \omega C\left(\frac{v^{*}_{cq,k+1} + v_{cq,k+1}}{2}\right)$$

$$i^{*}_{2q,k} = i_{1q,k} + \left(\frac{C}{T_s}\right).\left(v^{*}_{cq,k+1} - v_{cq,k+1}\right) + \omega C\left(\frac{v^{*}_{cd,k+1} + v_{cd,k+1}}{2}\right) \quad 8.2$$

$$v^{*}_{d,k+1} = v_{cd,k} + R_2 \sum_{n=0}^{k}\left(i^{*}_{2d,n} - i_{2d,n}\right) + \left(\frac{L_2}{T_s} + \frac{R_2}{2}\right).\left(i^{*}_{2d,k} - i_{2d,k}\right) - \omega L_2\left(\frac{i^{*}_{2q,k} + i_{2q,k}}{2}\right)$$

$$v^{*}_{q,k+1} = v_{cq,k} + R_2 \sum_{n=0}^{k}\left(i^{*}_{2q,n} - i_{2q,n}\right) + \left(\frac{L_2}{T_s} + \frac{R_2}{2}\right).\left(i^{*}_{2q,k} - i_{2q,k}\right) + \omega L_2\left(\frac{i^{*}_{2d,k} + i_{2d,k}}{2}\right) \quad 8.3$$

Another area of investigation is the examination of the effects of dips and flickers in the grid voltage on the output and input current quality with embedded generation using the matrix converter. Also, the size of the smallest external energy storage element that can be used to overcome these effects should be determined and its optimum connection position in the system assessed.

Application of the UPFC based on either matrix or back-to-back converters to control the output voltage magnitude and the power from an induction generator directly connected to the grid can be investigated.

Parameters, load, and input voltage variations have not been considered. It is important to investigate if the system transient performance will be improved by using a non-linear controller or by using sliding mode control which is more robust to parameter variations.

APPENDIX A

DSP HEADER FILES

/*commonheader.h file*/

```
#include <stdio.h>
#include <string.h>
#include <dma.h>
#include <emif.h>
#include <intr.h>
#include <common.h>
#include <math.h>
#include <board.h>
#include "fpga.h"
#include "EMIF_.h"
#include "DSP_CNTRL_REG.h"
#define pi 3.14285714285714285714285714285714
interrupt void increment(void);
```

```
/* fpga.h file*/

/*Addresses of reading and writing from FPGA*/

#define READ_FPGA_ADDR      0x01600000
#define WRITE1_FPGA_ADDR    0x01500000
#define WRITE2_FPGA_ADDR    0x01500400
#define WRITE3_FPGA_ADDR    0x01500800
#define WRITE4_FPGA_ADDR    0x01500c00

/* registers operation of reading and writing addresses*/

#define READ_FPGA       *(unsigned int*)READ_FPGA_ADDR
#define WRITE1_FPGA     *(unsigned int*)WRITE1_FPGA_ADDR
#define WRITE2_FPGA     *(unsigned int*)WRITE2_FPGA_ADDR
#define WRITE3_FPGA     *(unsigned int*)WRITE3_FPGA_ADDR
#define WRITE4_FPGA     *(unsigned int*)WRITE4_FPGA_ADDR
```

```
/* EMIF_.h file*/

/*define common emif settings*/

#define DEF_EMIF_CE0_CTRL          0x00000040
#define DEF_EMIF_CE2_CTRL          0x00000030
#define DEF_EMIF_CE3_CTRL          0x00000030
#define DEF_EMIF_SDRAM_CTRL        0x07227000
#define DEF_EMIF_SDRAM_REF               0x000004e1
#define EMIF_GCTRL_VALUE           0x00003078
#define EMIF_CE1_CTRL_VALUE              0xffff3f23
```

```
/*DSP_CNTRL_REG.h*/

/*addresses of dsp mapped control/status registers*/

#define DSP_CNTRL_ADDR          0x01780000
#define DSP_STAT_ADDR                 0x01780004
#define DSP_DIPOPT_ADDR         0x01780008
#define DSP_DIPBOOT_ADDR        0x0178000c
#define DSP_DSPOPT_ADDR         0x01780010
#define DSP_DSPBOOT_ADDR        0x01780014
#define DSP_FIFOSTAT_ADDR       0x01780018
#define DSP_SDCNTRL_ADDR        0x0178001c
```

APPENDIX B

OPEN LOOP PROGRAM

```
/*~FILE: Openloop.c

 ~HEADER:
        Application program for open loop control.
 ~!

 #include <stdlib.h>

 #if defined(CHIP_6711) || defined(CHIP_6211) || defined(CHIP_6211X)
 #include "6x11DSK.h"
 #include "regs.h"
 #else
 #include <emif.h>
 #endif

 #include "AED.h"
 #include "AED_DMS.h"
 #include "AED_Appl.h"
 #include <commonheader.h>

 #if AED_PRINT /* Must be placed after AED.h */
   #include <stdio.h>
 #endif

 /*-------------------------------------------------------------

 /* application may change these defines */
 #define AED_BOARD        "106"
 #define DMS_MODE         DMA_FS_MODE
 #define RECORD_SKIP_POWER (7)      /* NO_RECORDS to skip for speed*/
 #define DIVIDE_POWER     (0)      /* NO_RECORDS = 2^DIVIDE_POWER */
 #define NO_RECORDS       (1<<DIVIDE_POWER)      /* records/frame */
 #define NO_FRAMES        1                  /* frames/block  */
 #define RECLEN           8          /* length of record in words */
 #define ELEMENTSIZE_CODE  DMA_ESIZE32
 #define SAVE_RECORDS     1                  /* size of printout */
 #define ITERATIONS       1                /* size of averages */

 /* Globals for diagnostics of termination */
 unsigned int debug_times;
  int debug_bufs_proc;
  int debug_buf_count;
  int debug_prev_buf_count;
  int debug_DMS_err;
  int debug_appl_term_code;
 unsigned int debug_FIFO_ovfl;
 unsigned int debug_DMS_count;
 ApplBlockType data_buffer;
   unsigned int ADC[16],x[150],YYY[150],constant;
```

```
 int vvv=0,iii,kk;
  int index_ioa,index_iob,index_ioc,index_t1,index_t2;
 double vvvvv;
 int id,iq;
 int index_t2x;
 int thetaip2,sec2;
 int ialpha,ibeta;
 int sum2,mmmmm;
 int i,j,lx,s,nnn,mmmm,nx=0,mx=0;
 float Tx;
 unsigned int
INVERTER,CONVERTER,SECTOR,CURRENT_DIRECTION,LA,LB,LC,t0_,t1_,t2_,t3_,t4_,t5_,se
c;
 unsigned int da,db,dc;
 int iaref,ibref,icref,imref;
 int index_ioa90;
 int iaact1=0,ibact1=0;
 float thetaip,thetaop,stepip,stepop,fs;
 int peak,fo,index_io,iarefint,ibrefint,icrefint,iaerr,iberr,icerr,iaref2,ibref2,icref2,max,min;
 int iaact,ibact,icact;
 int zxc=0;
 short int flgy;
 int vab1=0,vbc1=0;
 int a1,a2;
 float errvd,vd1,vq1;
 float nnnnn;
 int iaerrx, iberrx, icerrx;
 int ax;
 int ccccc;
 int ud,uq;
 int ualpha,ubeta;
 int errd,errq;
 int idref=0,iqref;
 int iaerrn,iberrn,icerrn;
 unsigned int  xxx,xxxx;
 unsigned int av,k;
 unsigned int flgx=1,asd=1;
 unsigned int kkkkk;
 unsigned int index_sec;
 float valpha,vbeta,vd,vq;
 int vab,vbc;
 float w0,wk,th0,thk;
 int sum=0;
 int sss;
 int iaaaa,ibbbb;
 int index_ip,t1,t2,t0,index_2;
 float   ia_delta, ib_delta, ic_delta;
 int current_transformation,ai_1, bi_1, ci_1, ai, bi, ci,ap, bp, cp;
 ApplBlockType ptr;


 ApplBlockType save_data;

 #if(!TALK_TO_FPGA)
   far unsigned int test_data[NO_RECORDS*NO_FRAMES*RECLEN];
 #endif

 unsigned long fpga_io_reg = 0x70000001;
```

```
Dma_channel fpga_chan;
int input_count;

/*---------------------------------------*/
/* the following code is application dependent */

static int loop_count;

unsigned int A_value[ITERATIONS];
unsigned int B_value[ITERATIONS];
unsigned int C_value[ITERATIONS];
unsigned int D_value[ITERATIONS];
unsigned int E_value[ITERATIONS];
unsigned int F_value[ITERATIONS];
unsigned int G_value[ITERATIONS];
unsigned int H_value[ITERATIONS];
unsigned int I_value[ITERATIONS];
unsigned int J_value[ITERATIONS];
unsigned int K_value[ITERATIONS];
unsigned int L_value[ITERATIONS];
unsigned int M_value[ITERATIONS];
unsigned int N_value[ITERATIONS];
unsigned int O_value[ITERATIONS];
unsigned int P_value[ITERATIONS];

/* end of application dependent code */
/*---------------------------------------*/

/*****************************************************************
 appl_parms - Defines mode and size of block for DMS transfers
  Parameters:  OUT frames  - number of frames in the block
               OUT records - number of records in the frame
               OUT reclen  - number of transfer elements in record
               OUT esize   - transfer element size code (AED_Brd.h)
               OUT mode    - transfer mode code (AED_DMS.h)

 Notes:

 1) This routine is always called from main prior to allocation
 of the block for data transfer.  This function returns the
 specifics for the allocation and the DMS setup.

 2) The DMS can transfer the data in different widths up to the
 EMIF bus size.  The code selects the size independent of DSP.

 3) The DMS mode codes include both word and frame synchronization.
 Synchronization is the pulses sent on a wire connecting the AED
 with the DSP used by the DMS to determine when to read or write
 to the AED.

 4) The DMS can be implemented with any of the facilities available
 on any particular DSP.  Interrupts are available on all DSPs so
 the AED_DSM_intr support can always be used, although it is slow.
 The other implementations depend on what the DSP has for data
 transfer hardware.

*****************************************************************/
void appl_parms(unsigned int *frames,
```

```
                unsigned int *records,
                unsigned int *reclen,
                unsigned int *esize,
                unsigned int *mode)
{
 /*---------------------------------------*/
 /* the following code may be application dependent or
    additional code may be required */

/* AED106 Standard EMIF settings
 Parm        6x01   6x11
 Write setup  4      3
 Write strobe 3      2
 Write hold   3      2
 Read setup   13     5
  Read strobe  2     7
  Read hold    0     0
*/
#if defined(CHIP_6711) || defined(CHIP_6211) || defined(CHIP_6211X)
  REG_WRITE (EMIF_CE2, 0x30A50727); /* CE2 control, 32bit async*/
  #if (TALK_TO_FPGA)
    REG_WRITE (MAR0,  0x1);    /* Enable cache for SRAM     */
    REG_WRITE (L2CFG,  0x80000002);  /* 2-Way cache & EDMA Pri */
  #endif
#else
  EMIF_CE1_CTRL = 0x40FD0220;
#endif

#if(!TALK_TO_FPGA)
  appl_test_data.uword = test_data;
#endif

*frames = NO_FRAMES;
*records = NO_RECORDS;
*reclen = RECLEN;
*esize = ELEMENTSIZE_CODE;
*mode = DMS_MODE;
 peak=127;
 imref=10;
 fs=6103;        //switching frequency
Ts=1/fs;
 m=Ts/2/peak;
 fo=80;           //output frequency
 Tx=Ts;       //program dependant
stepop=720*fo/fs;
index_ioa=0;
thetaop=0;
current_transformation=40;   //2x
/**********************write to FPGA*********************************/
iaref2=0;
ibref2=0;
icref2=0;
sec=1;
t0_=0;
t1_=0;
t2_=0;
da=0;
db=0;
```

```
 dc=0;
 ai_1=0;
 bi_1=0;
 ci_1=0;
 th0=0;
  errd=0;
  iqref=(-122875);
  errq=0;
 w0=50*2*pi;
 flgy=0;
  /*******inverter values:******/
 LOAD_FIELD(&INVERTER,iaref2,0,8);
 LOAD_FIELD(&INVERTER,ibref2,8,8);
 LOAD_FIELD(&INVERTER,icref2,16,8);
 LOAD_FIELD(&INVERTER,sec,24,3);
 LOAD_FIELD(&INVERTER,0,27,5);
 /*********************************/

 /*******converter values:******/
 LOAD_FIELD(&CONVERTER,t0_,0,8);
 LOAD_FIELD(&CONVERTER,t1_,8,8);
 LOAD_FIELD(&CONVERTER,t2_,16,8);
 LOAD_FIELD(&CONVERTER,0,27,5);
 LOAD_FIELD(&CONVERTER,da,24,1);
 LOAD_FIELD(&CONVERTER,db,25,1);
 LOAD_FIELD(&CONVERTER,dc,26,1);
 WRITE1_FPGA=INVERTER;
 WRITE2_FPGA=CONVERTER;
 /*********************************/


 /************************INTERRUPT*********************************/
   intr_map(CPU_INT7,ISN_EXT_INT4);
   INTR_GLOBAL_ENABLE();
   INTR_ENABLE(CPU_INT_NMI);
   INTR_ENABLE(CPU_INT7);
   intr_hook(increment,CPU_INT7);

  /*****************************************************************/

  /* end of application dependent code */
  /*---------------------------------------*/

  write_32b_reg(get_cntl_addr(), fpga_io_reg);

#if AED_PRINT

  printf("\n*** AED " AED_BOARD " TEST PROGRAM STARTED ***\n");
#endif

} /* end appl_parms */


/*****************************************************************
 appl_init - Performs buffer init before data transfer
   Parameters:  OUT data_block - pointer to beginning of entire input
                       block (all frames)
```

```
            IN block_bytes - number of bytes in block
            IN dma_chan - DMA channel allocated by main for input
   Note: This routine is autmatically called from main before the
         data transfer is intiated, to initialize data buffers.
*****************************************************************/
void appl_init(ApplBlockType data_block,
            unsigned int  block_bytes,
            Dma_channel   dma_chan)
{
 /*--------------------------------------*/
 /* the following code may be application dependent or additional
    code may be required */
 int i, j, k;
 unsigned int *ptr1;

 save_data.byte
  = malloc(SAVE_RECORDS*RECLEN*byte_size(ELEMENTSIZE_CODE));
 if (!save_data.byte) {
  error_flashing(AED_FLASH_APPL_SAVE_BUFFER_MALLOC_ERROR);
 } /* end if */
 fpga_chan = dma_chan;

 ptr1 = data_block.uword;
 for (i=0; i<NO_FRAMES; i++) {
  for (j=0; j<NO_RECORDS; j++) {
    for (k=0; k<RECLEN; k++) {
      *(ptr1++) = 0;
    }
  }
 }

 ptr1 = save_data.uword;
 for (j=0; j<SAVE_RECORDS; j++) {
  for (k=0; k<RECLEN; k++) {
    *(ptr1++) = 0;
  }
 }

 for (j=0; j<ITERATIONS; j++) {
  A_value[j] = 0;
  B_value[j] = 0;
  C_value[j] = 0;
  D_value[j] = 0;
  E_value[j] = 0;
  F_value[j] = 0;
  G_value[j] = 0;
  H_value[j] = 0;
  I_value[j] = 0;
  J_value[j] = 0;
  K_value[j] = 0;
  L_value[j] = 0;
  M_value[j] = 0;
  N_value[j] = 0;
  O_value[j] = 0;
  P_value[j] = 0;
 } /* end for */

 loop_count = 0;
```

```
  input_count = 0;

#if AED_PRINT
  printf("Begin application processing (Block size = %d bytes)\n",
          block_bytes);
#endif
  /* end of application dependent code */
  /*-------------------------------------------*/
  /* the following code line is removed when daughterboard is
     added */

} /* end appl_init */
```

```
/*****************************************************************
  appl_process - Processes 1 frame of buffer data
    Returns:     user defined termination code, 0 is no termination
    Parameters:  IN data_buffer - pointer to beginning of frame of
                                  data just received from FPGA
                 IN buf_number - number of buffer just received
    Note: This routine is autmatically called from main when a full
          buffer of data has been transferred from the daughterboard.
*****************************************************************/
int  appl_process(ApplBlockType data_buffer,
                  int buf_number)
{

 /*-------------------------------------------*/
 /* the following code is application dependent */

 int xx=0;
```

```
kk=0;
xxx=0;
j=0;

index_sec=0;
nnn=0;
mmmm=0;
xxxx=0;
lx=0;
k=0;
 while (XSTAT0==0)
{;}
        do
        {
        ptr=data_buffer;
```

```
ADC[1]= (unsigned int)*(ptr.uhword++);
ADC[2]= (unsigned int)*(ptr.uhword++);
ADC[0]= (unsigned int)*(ptr.uhword++);
ADC[11]= (unsigned int)*(ptr.uhword++);
ADC[4]= (unsigned int)*(ptr.uhword++);
ADC[5]= (unsigned int)*(ptr.uhword++);
ADC[6]= (unsigned int)*(ptr.uhword++);
ADC[7]= (unsigned int)*(ptr.uhword++);
ADC[8]= (unsigned int)*(ptr.uhword++);
```

```
 ADC[9]= (unsigned int)*(ptr.uhword++);
 ADC[10]= (unsigned int)*(ptr.uhword++);
 ADC[3]= (unsigned int)*(ptr.uhword++);
 ADC[12]= (unsigned int)*(ptr.uhword++);
 ADC[13]= (unsigned int)*(ptr.uhword++);
 ADC[14]= (unsigned int)*(ptr.uhword++);
 ADC[15]= (unsigned int)*(ptr.uhword++);
 sum+=ADC[3];
 k++;
 if (k==8)
{
 av=sum/8;
 k=0;
 sum=0;
if ((av<1980)&&(av>1960))
{

        nnn=1;

}
}
if (nnn==1)
   {
   if (ADC[3]>s)
    {
    sec=6;
    thetaip=84;
    mmmm=1;
    vvv=1;
    index_ip=24;
          }
        if (ADC[3]<s)
         {
         sec=3;
         thetaip=84;
         mmmm=1;
         vvv=0;
         index_ip=24;
         }
        }
      s=ADC[3];
     }
 while (mmmm==0);

 while (xx==0)
{

/*****************************************************************/

 kkkkk++;
 iii=kkkkk/4;
 if (iii>140)
 { kkkkk=0;
  if (flgy==0)
   flgy=1;
  else flgy=0;
}
```

```
 FPGA_stop();

        FPGA_start();

/*************************************************************/

/*******************READ FROM FPGA*************************/
j=0;
sum=0;
iaaaa=0;
ibbbb=0;
sum2=0;
ax=0;
        do
        {

                while (XSTAT0==0)
                {
                };
                ptr=data_buffer;
                flgx=1;

ADC[1]= (unsigned int)*(ptr.uhword++);
ADC[2]= (unsigned int)*(ptr.uhword++);
ADC[0]= (unsigned int)*(ptr.uhword++);
ADC[11]= (unsigned int)*(ptr.uhword++);
ADC[4]= (unsigned int)*(ptr.uhword++);
ADC[5]= (unsigned int)*(ptr.uhword++);
ADC[10]= (unsigned int)*(ptr.uhword++);
ADC[7]= (unsigned int)*(ptr.uhword++);
ADC[8]= (unsigned int)*(ptr.uhword++);
 ADC[9]= (unsigned int)*(ptr.uhword++);
 ADC[6]= (unsigned int)*(ptr.uhword++);
 ADC[3]= (unsigned int)*(ptr.uhword++);
 ADC[12]= (unsigned int)*(ptr.uhword++);
 ADC[13]= (unsigned int)*(ptr.uhword++);
 ADC[14]= (unsigned int)*(ptr.uhword++);
 ADC[15]= (unsigned int)*(ptr.uhword++);
                sum+=ADC[3];
                iaaaa+=ADC[10];
                ax+=ADC[11];
                ibbbb+=ADC[14];
                j++;
}
while (j<4);
sss=sum/4;
ADC[3]=sss;
ADC[10]=iaaaa/4;
        ADC[14]=ibbbb/4;
        ADC[11]=ax/4;

/*************************Vdq*********************************/

a1=720-(thk*360/pi);
a2=a1-180;
if (a2>720) a2-=720;
if (a2<0) a2+=720;
```

```
 vab=(ADC[3]-1990)*1000;
 vbc=(ADC[11]-1975)*1000;
          vab=(vab+4*vab1)/5;
          vbc=(vbc+4*vbc1)/5;
          vbc1=vbc;
          vab1=vab;
          vab=vab/1000;
          vbc=vbc/1000;
 valpha=(3*vab)/2;
 vbeta=(0.86602540378*(-vab-2*vbc));
 vd=(((valpha*sine[a2])+(vbeta*sine[a1]))/8192);
 vq=((((-1)*valpha*sine[a1])+(vbeta*sine[a2]))/8192);
 vd=(vd+99*vd1)/100;
 vq=(vq+99*vq1)/100;
 vd1=vd;
 vq1=vq;
 if (vq1==0) vq=0.000001;
 wk=w0+(AA*vd1/vq1);
 if (wk>51)
         wk=51;
 elseif (wk<49)
         wk=49;
 end;
 thk=th0+(wk/fs)+ (BB*vd1/vq1);;
 while (thk>(2*pi)) thk-=(2*pi);
 while (thk<0) thk+=(2*pi);
 w0=wk;
 th0=thk;
 thetaip2=thk*360/pi-360;
 if (thetaip2<0) thetaip2+=720;
 if (thetaip2>=720) thetaip2-=720;
 sec2=(thetaip2/120)+1;
  if (sec2>6) sec2-=6;
  index_t2=thetaip2-((sec2-1)*120);
  index_t1=120-index_t2;


 /**********************dq controller**********************/
/**********************integral**************/

ki=0;
kc=0;
errd+=(ki*(idref-id));
errq+=(ki*(iqref-iq));
ud=((kp*(idref-id))+errd+(kc*((iq+iqref)/2)))/8192;
uq=((kp*(iqref-iq))+errq-(kc*((id+idref)/2)))/8192;
ualpha=(((ud*sine[index_ioa90])-(uq*sine[index_ioa])));
ubeta=(((ud*sine[index_ioa])+(uq*sine[index_ioa90])));
iaerr=((int)(0.8164965809*ualpha))/8192/32;
iberr=((int)((-0.4082482905*ualpha)+(0.7071067812*ubeta)))/8192/32;
icerr=-iaerr-iberr;
/*************************************************************/

/*******************current direction*******************/
//implemented on FPGA

//needs to be revised
iaact=current_transformation*(ADC[10]-1955);
```

```
 ibact=current_transformation*(ADC[14]-1987);

 if (iaact>0)
   da=0;
 else
   da=1;

 if (ibact>0)
   db=0;
 else
   db=1;

 if (icact>0)
   dc=0;
 else
   dc=1;

 /*****************************************************************/

/********************** t1, t2 & t0*****************************/

t1=((peak)*sine[index_t1])/8192;
t2=((peak)*sine[index_t2])/8192;
t0=(peak -t1-t2)/2;

/************************time calculations*********************/
 t0_=(t0);
 t1_=t0_+(t1);
 t2_=t1_+(t2);

 /*****************************************************************/

 /**********************Prepare data*********************************/


LOAD_FIELD(&INVERTER,iaerr,0,8);
LOAD_FIELD(&INVERTER,iberr,8,8);
LOAD_FIELD(&INVERTER,icerr,16,8);
LOAD_FIELD(&INVERTER,sec,24,3);
LOAD_FIELD(&INVERTER,0,27,5);

 /*******************************/
 /*******converter values:******/

 LOAD_FIELD(&CONVERTER,t0_,0,8);
 LOAD_FIELD(&CONVERTER,t1_,8,8);
 LOAD_FIELD(&CONVERTER,t2_,16,8);
 LOAD_FIELD(&CONVERTER,0,27,5);
 LOAD_FIELD(&CONVERTER,da,24,1);
```

```
  LOAD_FIELD(&CONVERTER,db,25,1);
  LOAD_FIELD(&CONVERTER,dc,26,1);
  xxxx++;
  /*******************************/



          while ((flgx==1))
          {
          zxc++;

          };
}
  return 0;
} /* end appl_process */


/*****************************************************************************
  appl_idle - Performs background processing
    Returns:   user defined termination code, 0 is no termination
    Note: This routine is automatically called from main when no other
          processing is required, but may not be called regularly.
*****************************************************************************/
int  appl_idle(void)
{
 /*-------------------------------------*/
 /* the following code is application dependent */
 loop_count++;
 /* end of application dependent code */
 /*-------------------------------------*/
  return 0;
} /* end appl_idle */


/*****************************************************************************
  appl_end - Final processing before termination
    Parameters:  IN times         - main program loop cycles executed
                 IN bufs_proc     - number of buffers processed by
                                    appl_process
                 IN buf_count     - number of buffers received
                 IN prev_buf_count - last buffer given to appl_process
                 IN DMS_err       - error code from DMS
                 IN appl_term_code - user termination code from either
                                    appl_process or appl_idle
                 IN FIFO_ovfl     - indication that the FIFOin the
                                    FPGA has overflowed
                 IN DMS_count     - number of frames remaining to be
                                    received in the block
    Note: This routine is automatically called from main at program
          termination.
*****************************************************************************/
void appl_end(unsigned int times,
              int       bufs_proc,
              int       buf_count,
              int       prev_buf_count,
              int       DMS_err,
```

```
            int       appl_term_code,
            unsigned int FIFO_ovfl,
            unsigned int DMS_count)
{
 /*-----------------------------------------*/
 /* the following code is application dependent */


 /* end of application dependent code */
 /*-----------------------------------------*/
} /* end appl_end */



/*****************************************************************************
  appl_test - Fills separate test block to simulate data from FPGA
   Parameters:  OUT fill - pointer to beginning of test block
              IN frame_bytes - number of bytes in each frame
              IN frames - number of frames in block
   Note: This routine is autmatically called from main to initialize
       the test buffer with data.
*****************************************************************************/
#if(!TALK_TO_FPGA)
#define SAMPLE_RATE_KHZ  10000L  /* Sample rate for timer setup */
unsigned long  appl_test(ApplBlockType fill,
                  int frame_elements,
                  int frames)
{
 /*-----------------------------------------*/
 /* the following code is application dependent */
 int i, j;
 int samples_per_element = (byte_size(ELEMENTSIZE_CODE)/sizeof(short));
 int frame_samples = frame_elements*samples_per_element;

 /* period is in 0.5 microseconds units */
 /* for a time between frames (frame sync)
   period = (frame_elements/SAMPLE_RATE)/(0.5e-6 us) */
unsigned long period = (2000L*(long)frame_elements)/SAMPLE_RATE_KHZ;

 /* fill can be changed to accomodate testing */
 for (i=0; i<frames; i++) {
  for (j=0; j<frame_samples; j++)  {
    fill.uhword[i*frame_samples+j]
     = ((i+1)*0x1000)+(j+1);
  } /* end for */
 } /* end for */
 /* end of application dependent code */
 /*-----------------------------------------*/
 return period;
} /* end appl_test */
#endif

interrupt void increment(void)
{
    WRITE1_FPGA=INVERTER;
    WRITE2_FPGA=CONVERTER;

    thetaop+=stepop;
```

```
        if (thetaop>720 ) thetaop=thetaop-720;
        index_ioa=(int)(thetaop);
        index_iob=index_ioa-240;
         if (index_iob<=0) index_iob+=720;


        flgx=0;

        return;
}
```

APPENDIX C

CLOSED LOOP PROGRAM

```
/*~FILE: closedloop.c

 ~HEADER:
        Application program for closed loop control.

#include <stdlib.h>
 #if defined(CHIP_6711) || defined(CHIP_6211) || defined(CHIP_6211X)
 #include "6x11DSK.h"
 #include "regs.h"
 #else
 #include <emif.h>
 #endif

 #include "AED.h"
 #include "AED_DMS.h"
 #include "AED_Appl.h"
 #include <commonheader.h>

 #if AED_PRINT /* Must be placed after AED.h */
  #include <stdio.h>
 #endif

 /*-------------------------------------------------------------
  In documentation in this module, the term "buffer" is used to refer
  to either a "block" or a "frame".  "Block" is the whole memory area
  allocated to transferring data from the AED's (daughterboard) FPGA
  to the DSP's memory via DMS through the EMIF bus.  The "block" is
  divided into one or more "frames".

  The DMS, Data Movement Service, is implemented in several forms for
  use in various DSPs.  Access to the DMS is through a common header
  module (AED_DMS.h) which serves all implementations.  The most
  elementary implementation is the use of the CPU to tranfer the data
  (AED_DMS_intr.c).  This implementation supports DSP with not data
  movement hardware, but it is slow and not recommended if another
  implementation can be used.  Other implementations are available
  for DMAs and EDMAs on various DSPs.

 In most modes of operation, one frame of data is processed at a
 time.  The DMS is setup to give an interrupt at the completion of
 transfer of each frame.  Upon the receipt of the interrupt, but
 not in the interrupt service routine, the frame last transfered
 is presented to appl_process as the "buffer" for processing.

 However, in block processing modes, interrupts occur only once for
 the entire block, and the entire block is presented as the "buffer"
 for processing in the appl_process function.

 The mode, number of frames in the block, number of records in the
```

```
  frame, and the size of the records are selected in appl_parms.
  Records are a application dependent subdivision of the frame.
  Frequently, multiple records, identical in format, a tranfered
  in a frame in order to reduce the number of interrupts.
 ------------------------------------------------------------*/

/* application may change these defines */
#define AED_BOARD       "106"
#define DMS_MODE        DMA_FS_MODE
#define RECORD_SKIP_POWER (7)     /* NO_RECORDS to skip for speed*/
#define DIVIDE_POWER     (0)      /* NO_RECORDS = 2^DIVIDE_POWER */
#define NO_RECORDS       (1<<DIVIDE_POWER)      /* records/frame */
#define NO_FRAMES        1                      /* frames/block  */
#define RECLEN           8         /* length of record in words */
#define ELEMENTSIZE_CODE  DMA_ESIZE32
 #define SAVE_RECORDS     1              /* size of printout */
 #define ITERATIONS       1             /* size of averages */

 /* Globals for diagnostics of termination */
 unsigned int debug_times;
 int debug_bufs_proc;
 int debug_buf_count;
 int debug_prev_buf_count;
 int debug_DMS_err;
 int debug_appl_term_code;
unsigned int debug_FIFO_ovfl;
unsigned int debug_DMS_count;
ApplBlockType data_buffer;
unsigned int ADC[16],x[150],YYY[150],constant;
   int vvv=0,iii,kk;
    int index_ioa,index_iob,index_ioc,index_t1,index_t2;
  double vvvvv;
    int id,iq;
    int ialphar,ibetar;
    int index_t2x;
    int thetaip2,sec2;
    int ialpha,ibeta;
    int sum2,mmmmm;
    int i,j,lx,s,nnn,mmmm,nx=0,mx=0;
    float Tx;
   unsigned int
INVERTER,CONVERTER,SECTOR,CURRENT_DIRECTION,LA,LB,LC,t0_,t1_,t2_,t3_,t4_,t5_,se
c;
   unsigned int da,db,dc;
   int iaref,ibref,icref,imref;
   int index_ioa90;
   int iaact1=0,ibact1=0;
   float thetaip,thetaop,stepip,stepop,fs;
   int peak,fo,index_io,iarefint,ibrefint,icrefint,iaerr,iberr,icerr,iaref2,ibref2,icref2,max,min;
   int iaact,ibact,icact;
   int zxc=0;
   short int flgy;
   int vab1=0,vbc1=0;
   int a1,a2;
   float errvd,vd1,vq1;
   float nnnnn;
   int iaerrx, iberrx, icerrx;
   int ax;
```

```
int ccccc;
int kp,ki,kc;
int ud,uq;
int ualpha,ubeta;
int errd,errq;
int idref=0,iqref;
int iaerrn,iberrn,icerrn;
unsigned int  xxx,xxxx;
unsigned int av,k;
unsigned int flgx=1,asd=1;
unsigned int kkkkk;
unsigned int index_sec;
float valpha,vbeta,vd,vq;
int vab,vbc;
float w0,wk,th0,thk;
int sum=0;
int sss;
int iaaaa,ibbbb;
int index_ip,t1,t2,t0,index_2;
float   ia_delta, ib_delta, ic_delta;
int current_transformation,ai_1, bi_1, ci_1, ai, bi, ci,ap, bp, cp;
ApplBlockType ptr;



ApplBlockType save_data;

#if(!TALK_TO_FPGA)
 far unsigned int test_data[NO_RECORDS*NO_FRAMES*RECLEN];
#endif

unsigned long fpga_io_reg = 0x70000001;
Dma_channel fpga_chan;
int input_count;

/*-----------------------------------------*/
/* the following code is application dependent */

static int loop_count;

unsigned int A_value[ITERATIONS];
unsigned int B_value[ITERATIONS];
unsigned int C_value[ITERATIONS];
unsigned int D_value[ITERATIONS];
unsigned int E_value[ITERATIONS];
unsigned int F_value[ITERATIONS];
unsigned int G_value[ITERATIONS];
unsigned int H_value[ITERATIONS];
unsigned int I_value[ITERATIONS];
unsigned int J_value[ITERATIONS];
unsigned int K_value[ITERATIONS];
unsigned int L_value[ITERATIONS];
unsigned int M_value[ITERATIONS];
unsigned int N_value[ITERATIONS];
unsigned int O_value[ITERATIONS];
unsigned int P_value[ITERATIONS];

/* end of application dependent code */
```

```
/*-----------------------------------------*/

/*************************************************************************
 appl_parms - Defines mode and size of block for DMS transfers
  Parameters:  OUT frames  - number of frames in the block
               OUT records - number of records in the frame
               OUT reclen  - number of transfer elements in record
               OUT esize   - transfer element size code (AED_Brd.h)
               OUT mode    - transfer mode code (AED_DMS.h)

  Notes:

  1) This routine is always called from main prior to allocation
  of the block for data transfer.  This function returns the
  specifics for the allocation and the DMS setup.

     2) The DMS can transfer the data in different widths up to the
     EMIF bus size.  The code selects the size independent of DSP.

     3) The DMS mode codes include both word and frame synchronization.
     Synchronization is the pulses sent on a wire connecting the AED
     with the DSP used by the DMS to determine when to read or write
     to the AED.

     4) The DMS can be implemented with any of the facilities available
     on any particular DSP.  Interrupts are available on all DSPs so
     the AED_DSM_intr support can always be used, although it is slow.
     The other implementations depend on what the DSP has for data
     transfer hardware.

*************************************************************************/
void appl_parms(unsigned int *frames,
                unsigned int *records,
                unsigned int *reclen,
                unsigned int *esize,
                unsigned int *mode)
{
 /*-----------------------------------------*/
 /* the following code may be application dependent or
    additional code may be required */

/* AED106 Standard EMIF settings
 Parm         6x01   6x11
 Write setup  4      3
 Write strobe 3      2
 Write hold   3      2
 Read setup   13     5
 Read strobe  2      7
 Read hold    0      0
*/
#if defined(CHIP_6711) || defined(CHIP_6211) || defined(CHIP_6211X)
  REG_WRITE (EMIF_CE2, 0x30A50727); /* CE2 control, 32bit async*/
  #if (TALK_TO_FPGA)
    REG_WRITE (MAR0,   0x1);     /* Enable cache for SRAM      */
    REG_WRITE (L2CFG,  0x80000002);  /* 2-Way cache & EDMA Pri */
  #endif
#else
  EMIF_CE1_CTRL = 0x40FD0220;
```

```
#endif

#if(!TALK_TO_FPGA)
  appl_test_data.uword = test_data;
#endif

  *frames = NO_FRAMES;
  *records = NO_RECORDS;
  *reclen = RECLEN;
  *esize = ELEMENTSIZE_CODE;
  *mode = DMS_MODE;
  peak=127;
  imref=10;
  fs=6103;        //program dependant
  Ts=1/fs;
  m=Ts/2/peak;
  fo=80;          //program dependant
  Tx=Ts;       //program dependant
  stepop=720*fo/fs;
  index_ioa=0;
  thetaop=0;
  current_transformation=40;   //2x
  /**********************write to FPGA*******************************/
  iaref2=0;
  ibref2=0;
  icref2=0;
  sec=1;
  t0_=0;
  t1_=0;
  t2_=0;
  da=0;
  db=0;
  dc=0;
  ai_1=0;
  bi_1=0;
  ci_1=0;
  th0=0;
  errd=0;
  iqref=(-122875*4);
  errq=0;
  w0=50*2*pi;
  flgy=0;

  /*******inverter values:******/
  LOAD_FIELD(&INVERTER,iaref2,0,8);
  LOAD_FIELD(&INVERTER,ibref2,8,8);
  LOAD_FIELD(&INVERTER,icref2,16,8);
  LOAD_FIELD(&INVERTER,sec,24,3);
  LOAD_FIELD(&INVERTER,0,27,5);
  /*********************************/

  /*******converter values:******/
  LOAD_FIELD(&CONVERTER,t0_,0,8);
  LOAD_FIELD(&CONVERTER,t1_,8,8);
  LOAD_FIELD(&CONVERTER,t2_,16,8);
  LOAD_FIELD(&CONVERTER,0,27,5);
  LOAD_FIELD(&CONVERTER,da,24,1);
  LOAD_FIELD(&CONVERTER,db,25,1);
```

```
 LOAD_FIELD(&CONVERTER,dc,26,1);
 WRITE1_FPGA=INVERTER;
 WRITE2_FPGA=CONVERTER;
 /*****************************/

 /***********************INTERRUPT*********************************/

 intr_map(CPU_INT7,ISN_EXT_INT4);
 INTR_GLOBAL_ENABLE();
 INTR_ENABLE(CPU_INT_NMI);
 INTR_ENABLE(CPU_INT7);
 intr_hook(increment,CPU_INT7);

/*************************************************************/
  /* end of application dependent code */
  /*---------------------------------------*/

  write_32b_reg(get_cntl_addr(), fpga_io_reg);

#if AED_PRINT

  printf("\n*** AED " AED_BOARD " TEST PROGRAM STARTED ***\n");
#endif

} /* end appl_parms */


/*************************************************************
 appl_init - Performs buffer init before data transfer
   Parameters:  OUT data_block - pointer to beginning of entire input
                      block (all frames)
                IN block_bytes - number of bytes in block
                IN dma_chan - DMA channel allocated by main for input
   Note: This routine is autmatically called from main before the
        data transfer is intiated, to initialize data buffers.
*************************************************************/
void appl_init(ApplBlockType data_block,
           unsigned int  block_bytes,
           Dma_channel   dma_chan)
{
 /*---------------------------------------*/
 /* the following code may be application dependent or additional
    code may be required */
 int i, j, k;
 unsigned int *ptr1;

 save_data.byte
  = malloc(SAVE_RECORDS*RECLEN*byte_size(ELEMENTSIZE_CODE));
 if (!save_data.byte) {
  error_flashing(AED_FLASH_APPL_SAVE_BUFFER_MALLOC_ERROR);
 } /* end if */
 fpga_chan = dma_chan;

 ptr1 = data_block.uword;
 for (i=0; i<NO_FRAMES; i++) {
  for (j=0; j<NO_RECORDS; j++) {
    for (k=0; k<RECLEN; k++) {
```

```
      *(ptr1++) = 0;
    }
   }
  }

  ptr1 = save_data.uword;
  for (j=0; j<SAVE_RECORDS; j++) {
   for (k=0; k<RECLEN; k++) {
    *(ptr1++) = 0;
   }
  }

  for (j=0; j<ITERATIONS; j++) {
   A_value[j] = 0;
   B_value[j] = 0;
   C_value[j] = 0;
   D_value[j] = 0;
   E_value[j] = 0;
   F_value[j] = 0;
   G_value[j] = 0;
   H_value[j] = 0;
   I_value[j] = 0;
   J_value[j] = 0;
   K_value[j] = 0;
   L_value[j] = 0;
   M_value[j] = 0;
   N_value[j] = 0;
   O_value[j] = 0;
   P_value[j] = 0;
  } /* end for */

  loop_count = 0;

   input_count = 0;

  #if AED_PRINT
   printf("Begin application processing (Block size = %d bytes)\n",
          block_bytes);
  #endif
   /* end of application dependent code */
   /*-----------------------------------------*/
   /* the following code line is removed when daughterboard is
     added */

} /* end appl_init */


/*************************************************************************
 appl_process - Processes 1 frame of buffer data
    Returns:    user defined termination code, 0 is no termination
    Parameters: IN data_buffer - pointer to beginning of frame of
                    data just received from FPGA
                IN buf_number - number of buffer just received
    Note: This routine is autmatically called from main when a full
        buffer of data has been transferred from the daughterboard.
*************************************************************************/
int  appl_process(ApplBlockType data_buffer,
          int buf_number)
```

```
{

 /*-----------------------------------------*/
 /* the following code is application dependent */

 int xx=0;
 kk=0;
 xxx=0;
 j=0;
 index_sec=0;
 nnn=0;
 mmmm=0;
 xxxx=0;
 lx=0;
 k=0;
 while (XSTAT0==0)
 {;}
        do
        {
        ptr=data_buffer;

 ADC[1]= (unsigned int)*(ptr.uhword++);
 ADC[2]= (unsigned int)*(ptr.uhword++);
 ADC[0]= (unsigned int)*(ptr.uhword++);
 ADC[11]= (unsigned int)*(ptr.uhword++);
 ADC[4]= (unsigned int)*(ptr.uhword++);
 ADC[5]= (unsigned int)*(ptr.uhword++);
 ADC[6]= (unsigned int)*(ptr.uhword++);
 ADC[7]= (unsigned int)*(ptr.uhword++);
 ADC[8]= (unsigned int)*(ptr.uhword++);
 ADC[9]= (unsigned int)*(ptr.uhword++);
 ADC[10]= (unsigned int)*(ptr.uhword++);
  ADC[3]= (unsigned int)*(ptr.uhword++);
  ADC[12]= (unsigned int)*(ptr.uhword++);
  ADC[13]= (unsigned int)*(ptr.uhword++);
  ADC[14]= (unsigned int)*(ptr.uhword++);
  ADC[15]= (unsigned int)*(ptr.uhword++);
  sum+=ADC[3];
  k++;
  if (k==8)
 {
  av=sum/8;
  k=0;
  sum=0;
 if ((av<1980)&&(av>1960))
 {

          nnn=1;

        }
 }
 }
 if (nnn==1)
     {
    if (ADC[3]>s)
      {
      sec=6;
      thetaip=84;
      mmmm=1;
```

```
                vvv=1;
                index_ip=24;
                      }
                    if (ADC[3]<s)
                     {
                     sec=3;
                     thetaip=84;
                     mmmm=1;
                     vvv=0;
                     index_ip=24;
                     }
                   }
          s=ADC[3];
         }
while (mmmm==0);

 while (xx==0)
{


 kkkkk++;
 iii=kkkkk/4;
 if (iii>140)
 { kkkkk=0;
  if (flgy==0)
   flgy=1;
  else flgy=0;
}

FPGA_stop();
FPGA_start();
 /*****************************************************************/

/*******************READ FROM FPGA*************************/
j=0;
sum=0;
iaaaa=0;
ibbbb=0;
sum2=0;
ax=0;
do
{

                while (XSTAT0==0)
                {
                 };
                ptr=data_buffer;
                flgx=1;
 ADC[1]= (unsigned int)*(ptr.uhword++);
 ADC[2]= (unsigned int)*(ptr.uhword++);
 ADC[0]= (unsigned int)*(ptr.uhword++);
 ADC[11]= (unsigned int)*(ptr.uhword++);
 ADC[4]= (unsigned int)*(ptr.uhword++);
 ADC[5]= (unsigned int)*(ptr.uhword++);
 ADC[10]= (unsigned int)*(ptr.uhword++);
 ADC[7]= (unsigned int)*(ptr.uhword++);
 ADC[8]= (unsigned int)*(ptr.uhword++);
```

```
ADC[9]= (unsigned int)*(ptr.uhword++);
ADC[6]= (unsigned int)*(ptr.uhword++);
ADC[3]= (unsigned int)*(ptr.uhword++);
ADC[12]= (unsigned int)*(ptr.uhword++);
ADC[13]= (unsigned int)*(ptr.uhword++);
ADC[14]= (unsigned int)*(ptr.uhword++);
ADC[15]= (unsigned int)*(ptr.uhword++);


            sum+=ADC[3];
            iaaaa+=ADC[10];
            ax+=ADC[11];
            ibbbb+=ADC[14];
            j++;
}
            while (j<6);
            sss=sum/6;
            ADC[6]=ADC[6];
            ADC[3]=sss;
            ADC[10]=iaaaa/6;
            ADC[14]=ibbbb/6;
            ADC[11]=ax/6;
            ADC[3]=sss;



/*************************Vdq*********************************/

a1=720-(thk*360/pi);
a2=a1-180;
if (a2>720) a2-=720;
 if (a2<0) a2+=720;
 vab=(ADC[3]-1990)*1000;
 vbc=(ADC[11]-1975)*1000;
          vab=(vab+4*vab1)/5;
          vbc=(vbc+4*vbc1)/5;
          vbc1=vbc;
          vab1=vab;
          vab=vab/1000;
          vbc=vbc/1000;
valpha=(3*vab)/2;
vbeta=(0.86602540378*(-vab-2*vbc));
vd=(((valpha*sine[a2])+(vbeta*sine[a1]))/8192);
vq=((((-1)*valpha*sine[a1])+(vbeta*sine[a2]))/8192);
vd=(vd+99*vd1)/100;
vq=(vq+99*vq1)/100;
vd1=vd;
 vq1=vq;
 if (vq1==0) vq=0.000001;
wk=w0+(AA*vd1/vq1);
if (wk>51)
        wk=51;
elseif (wk<49)
        wk=49;
end;
thk=th0+(wk/fs)+ (BB*vd1/vq1);;
while (thk>(2*pi)) thk-=(2*pi);
```

```
while (thk<0) thk+=(2*pi);
w0=wk;
th0=thk;
thetaip2=thk*360/pi-360;
if (thetaip2<0) thetaip2+=720;
if (thetaip2>=720) thetaip2-=720;
sec2=(thetaip2/120)+1;
if (sec2>6) sec2-=6;
index_t2=thetaip2-((sec2-1)*120);
index_t1=120-index_t2;


/*****************actual current values*******************************/
  iaact=current_transformation*(ADC[14]-1977);
  ibact=(-1)*current_transformation*(ADC[10]-1957);
k=1;
iaact1=iaact;
ibact1=ibact;
index_ioa90=index_ioa-180;
if (index_ioa90<=0)
index_ioa90+=720;
ialpha=(3*iaact/2);
ibeta=(int)(0.86602540378*(-iaact-2*ibact));
id=(((ialpha*sine[index_ioa90])+(ibeta*sine[index_ioa]))/8192);
iq=((((-1)*ialpha*sine[index_ioa])+(ibeta*sine[index_ioa90]))/8192);


 /**********************dq controller**********************/
/**********************integral**************/

  errd+=(ki*(idref-id));
  if (errd>300000) errd=300000;
  if (errd<-300000)  errd=-300000;

  errq+=(ki*(iqref-iq));
  if (errq>30000000) errq=30000000;
  if (errq<-30000000)  errq=-30000000;
  ud=((kp*(idref-id))+errd-(kc*((iqref))))/8192/4;
  uq=((kp*(iqref-iq))+errq+(kc*((idref))))/8192/4;

 ualpha=(((ud*sine[index_ioa90])-(uq*sine[index_ioa])));
 ubeta=(((uq*sine[index_ioa90])+(ud*sine[index_ioa])));
 iaerr=((int)(0.8164965809*ualpha))/8192/32;
 iberr=((int)((-0.4082482905*ualpha)+(0.7071067812*ubeta)))/8192/32;


/*****************************************************************/

/*******************current direction*******************/
//implemented on FPGA

//needs to be revised
if (iaact>0)
  da=0;
else
  da=1;
```

```
if (ibact>0)
  db=0;
else
  db=1;

if (icact>0)
  dc=0;
else
  dc=1;


/*************************************************************/
```

```
 icerr=-iaerr-iberr;


/********************** t1, t2 & t0*****************************/


t1=((peak)*sine[index_t1])/8192;
t2=((peak)*sine[index_t2])/8192;
t0=(peak -t1-t2)/2;

t0_=(t0);
t1_=t0_+(t1);
t2_=t1_+(t2);
 /*************************************************************/

 /**********************Prepare data*****************************/

LOAD_FIELD(&INVERTER,iaerr,0,8);
LOAD_FIELD(&INVERTER,iberr,8,8);
LOAD_FIELD(&INVERTER,icerr,16,8);
LOAD_FIELD(&INVERTER,sec,24,3);
LOAD_FIELD(&INVERTER,0,27,5);

/*******************************/
/*******converter values:******/

LOAD_FIELD(&CONVERTER,t0_,0,8);
LOAD_FIELD(&CONVERTER,t1_,8,8);
LOAD_FIELD(&CONVERTER,t2_,16,8);
LOAD_FIELD(&CONVERTER,0,27,5);
LOAD_FIELD(&CONVERTER,da,24,1);
LOAD_FIELD(&CONVERTER,db,25,1);
LOAD_FIELD(&CONVERTER,dc,26,1);
xxxx++;
/*******************************/
        while ((flgx==1))
        {
        zxc++;
```

```
            };
  }
  return 0;
} /* end appl_process */


/*****************************************************************
  appl_idle - Performs background processing
    Returns:   user defined termination code, 0 is no termination
    Note: This routine is automatically called from main when no other
          processing is required, but may not be called regularly.
*****************************************************************/
int  appl_idle(void)
{
  /*-------------------------------------*/
  /* the following code is application dependent */
  loop_count++;
  /* end of application dependent code */
  /*-------------------------------------*/
  return 0;
} /* end appl_idle */


/*****************************************************************
  appl_end - Final processing before termination
    Parameters:  IN times         - main program loop cycles executed
                 IN bufs_proc     - number of buffers processed by
                                    appl_process
                 IN buf_count     - number of buffers received
                 IN prev_buf_count - last buffer given to appl_process
                 IN DMS_err       - error code from DMS
                 IN appl_term_code - user termination code from either
                                    appl_process or appl_idle
                 IN FIFO_ovfl     - indication that the FIFOin the
                                    FPGA has overflowed
                 IN DMS_count     - number of frames remaining to be
                                    received in the block
    Note: This routine is automatically called from main at program
          termination.
*****************************************************************/
void appl_end(unsigned int times,
              int       bufs_proc,
              int       buf_count,
              int       prev_buf_count,
              int       DMS_err,
              int       appl_term_code,
              unsigned int FIFO_ovfl,
              unsigned int DMS_count)
{
  /*-------------------------------------*/
  /* the following code is application dependent */


  /* end of application dependent code */
  /*-------------------------------------*/
} /* end appl_end */
```

```
/*****************************************************************
  appl_test - Fills separate test block to simulate data from FPGA
    Parameters:  OUT fill - pointer to beginning of test block
                 IN frame_bytes - number of bytes in each frame
                 IN frames - number of frames in block
    Note: This routine is autmatically called from main to initialize
          the test buffer with data.
*****************************************************************/
#if(!TALK_TO_FPGA)
#define SAMPLE_RATE_KHZ  10000L   /* Sample rate for timer setup */
unsigned long  appl_test(ApplBlockType fill,
                         int frame_elements,
                         int frames)
{
/*-----------------------------------------*/
/* the following code is application dependent */
int i, j;
int samples_per_element = (byte_size(ELEMENTSIZE_CODE)/sizeof(short));
int frame_samples = frame_elements*samples_per_element;

/* period is in 0.5 microseconds units */
/* for a time between frames (frame sync)
   period = (frame_elements/SAMPLE_RATE)/(0.5e-6 us) */
unsigned long period = (2000L*(long)frame_elements)/SAMPLE_RATE_KHZ;

/* fill can be changed to accomodate testing */
for (i=0; i<frames; i++) {
  for (j=0; j<frame_samples; j++)  {
     fill.uhword[i*frame_samples+j]
       = ((i+1)*0x1000)+(j+1);
  } /* end for */
  } /* end for */
  /* end of application dependent code */
  /*---------------------------------------*/
  return period;
} /* end appl_test */
#endif

interrupt void increment(void)
{
    WRITE1_FPGA=INVERTER;
    WRITE2_FPGA=CONVERTER;

     thetaop+=stepop;

    if (thetaop>720 ) thetaop=thetaop-720;
    index_ioa=(int)(thetaop);
    index_iob=index_ioa-240;
     if (index_iob<=0) index_iob+=720;


    flgx=0;

    return;
}
```

APPENDIX E

UPFC PROGRAM

```
/*~FILE:upfc.c*/

#include <stdlib.h>

#if defined(CHIP_6711) || defined(CHIP_6211) || defined(CHIP_6211X)
#include "6x11DSK.h"
#include "regs.h"
#else
#include <emif.h>
#endif

#include "AED.h"
#include "AED_DMS.h"
#include "AED_Appl.h"
#include <commonheader.h>

#if AED_PRINT /* Must be placed after AED.h */
 #include <stdio.h>
#endif

/*-------------------------------------------------------------
 In documentation in this module, the term "buffer" is used to refer
 to either a "block" or a "frame".  "Block" is the whole memory area
 allocated to transferring data from the AED's (daughterboard) FPGA
 to the DSP's memory via DMS through the EMIF bus.  The "block" is
 divided into one or more "frames".

 The DMS, Data Movement Service, is implemented in several forms for
 use in various DSPs.  Access to the DMS is through a common header
 module (AED_DMS.h) which serves all implementations.  The most
 elementary implementation is the use of the CPU to tranfer the data
 (AED_DMS_intr.c).  This implementation supports DSP with not data
 movement hardware, but it is slow and not recommended if another
 implementation can be used.  Other implementations are available
 for DMAs and EDMAs on various DSPs.

In most modes of operation, one frame of data is processed at a
time.  The DMS is setup to give an interrupt at the completion of
transfer of each frame.  Upon the receipt of the interrupt, but
not in the interrupt service routine, the frame last transfered
is presented to appl_process as the "buffer" for processing.

 However, in block processing modes, interrupts occur only once for
 the entire block, and the entire block is presented as the "buffer"
 for processing in the appl_process function.

The mode, number of frames in the block, number of records in the
frame, and the size of the records are selected in appl_parms.
Records are a application dependent subdivision of the frame.
Frequently, multiple records, identical in format, a tranfered
in a frame in order to reduce the number of interrupts.
```

```
----------*/

/* application may change these defines */
#define AED_BOARD       "106"
#define DMS_MODE        DMA_FS_MODE
#define RECORD_SKIP_POWER (7)     /* NO_RECORDS to skip for speed*/
#define DIVIDE_POWER     (0)     /* NO_RECORDS = 2^DIVIDE_POWER */
#define NO_RECORDS       (1<<DIVIDE_POWER)      /* records/frame */
#define NO_FRAMES        1                  /* frames/block  */
#define RECLEN          8         /* length of record in words */
#define ELEMENTSIZE_CODE  DMA_ESIZE32
#define SAVE_RECORDS     1                  /* size of printout */
#define ITERATIONS       1                 /* size of averages */

/* Globals for diagnostics of termination */
 unsigned int debug_times;
 int debug_bufs_proc;
 int debug_buf_count;
 int debug_prev_buf_count;
 int debug_DMS_err;
 int debug_appl_term_code;
unsigned int debug_FIFO_ovfl;
unsigned int debug_DMS_count;
ApplBlockType data_buffer;
unsigned int ADC[16],x[150],YYY[150],constant;
   int vvv=0,iii,kk;
    int index_ioa,index_iob,index_ioc,index_t1,index_t2;
   double vvvvv;
   int id,iq;
   int index_t2x;
   int thetaip2,sec2;
   int ialpha,ibeta;
    int sum2,mmmmm;
    int i,j,lx,s,nnn,mmmm,nx=0,mx=0;
    float Tx;
    unsigned int
INVERTER,CONVERTER,SECTOR,CURRENT_DIRECTION,LA,LB,LC,t0_,t1_,t2_,t3_,t4_,t5_,se
c;
   unsigned int da,db,dc;
   int iaref,ibref,icref,imref;
   int index_ioa90;
   int iaact1=0,ibact1=0;
   float thetaip,thetaop,stepip,stepop,fs;
   int peak,fo,index_io,iarefint,ibrefint,icrefint,iaerr,iberr,icerr,iaref2,ibref2,icref2;
   int iaact,ibact,icact;
   int zxc=0;
   short int flgy;
   int vab1=0,vbc1=0;
   int a1,a2;
   float errvd,vd1,vq1;
   float nnnnn;
   int iaerrx, iberrx, icerrx;
   int ax;
   int ccccc;
   int kp,ki,kc;
   int ud,uq;
   int ualpha,ubeta;
   int errd,errq;
```

```
int idref=0,iqref;
int iaerrn,iberrn,icerrn;
unsigned int  xxx,xxxx;
unsigned int av,k;
unsigned int flgx=1,asd=1;
unsigned int kkkkk;
unsigned int index_sec;
float valpha,vbeta,vd,vq;
int vab,vbc;
float w0,wk,th0,thk;
int sum=0;
int sss;
int iaaaa,ibbbb;
int index_ip,t1,t2,t0,index_2;
float   ia_delta, ib_delta, ic_delta;
   int current_transformation,ai_1, bi_1, ci_1, ai, bi, ci,ap, bp, cp;
   ApplBlockType ptr;

 ApplBlockType save_data;

#if(!TALK_TO_FPGA)
  far unsigned int test_data[NO_RECORDS*NO_FRAMES*RECLEN];
#endif

unsigned long fpga_io_reg = 0x70000001;
Dma_channel fpga_chan;
int input_count;

/*-----------------------------------------*/
/* the following code is application dependent */

static int loop_count;

 unsigned int A_value[ITERATIONS];
 unsigned int B_value[ITERATIONS];
 unsigned int C_value[ITERATIONS];
 unsigned int D_value[ITERATIONS];
 unsigned int E_value[ITERATIONS];
 unsigned int F_value[ITERATIONS];
 unsigned int G_value[ITERATIONS];
 unsigned int H_value[ITERATIONS];
 unsigned int I_value[ITERATIONS];
 unsigned int J_value[ITERATIONS];
 unsigned int K_value[ITERATIONS];
 unsigned int L_value[ITERATIONS];
 unsigned int M_value[ITERATIONS];
 unsigned int N_value[ITERATIONS];
 unsigned int O_value[ITERATIONS];
unsigned int P_value[ITERATIONS];

/* end of application dependent code */
/*-----------------------------------------*/

/*****************************************************************
 appl_parms - Defines mode and size of block for DMS transfers
  Parameters:  OUT frames  - number of frames in the block
               OUT records - number of records in the frame
               OUT reclen  - number of transfer elements in record
```

```
OUT esize   - transfer element size code (AED_Brd.h)
OUT mode    - transfer mode code (AED_DMS.h)

Notes:

1) This routine is always called from main prior to allocation
of the block for data transfer.  This function returns the
specifics for the allocation and the DMS setup.

2) The DMS can transfer the data in different widths up to the
EMIF bus size.  The code selects the size independent of DSP.

3) The DMS mode codes include both word and frame synchronization.
Synchronization is the pulses sent on a wire connecting the AED
with the DSP used by the DMS to determine when to read or write
to the AED.

4) The DMS can be implemented with any of the facilities available
on any particular DSP.  Interrupts are available on all DSPs so
the AED_DSM_intr support can always be used, although it is slow.
The other implementations depend on what the DSP has for data
transfer hardware.

**********************************************************************/
void appl_parms(unsigned int *frames,
                unsigned int *records,
                unsigned int *reclen,
                unsigned int *esize,
                unsigned int *mode)
{
 /*-------------------------------------*/
 /* the following code may be application dependent or
    additional code may be required */

 /* AED106 Standard EMIF settings
  Parm        6x01    6x11
  Write setup  4       3
  Write strobe 3       2
  Write hold   3       2
  Read setup   13      5
  Read strobe  2       7
  Read hold    0       0
*/
#if defined(CHIP_6711) || defined(CHIP_6211) || defined(CHIP_6211X)
 REG_WRITE (EMIF_CE2, 0x30A50727); /* CE2 control, 32bit async*/
 #if (TALK_TO_FPGA)
  REG_WRITE (MAR0,   0x1);    /* Enable cache for SRAM     */
  REG_WRITE (L2CFG,  0x80000002);  /* 2-Way cache & EDMA Pri */
 #endif
#else
  EMIF_CE1_CTRL = 0x40FD0220;
#endif

#if(!TALK_TO_FPGA)
  appl_test_data.uword = test_data;
#endif

  *frames = NO_FRAMES;
```

```
*records = NO_RECORDS;
*reclen = RECLEN;
*esize = ELEMENTSIZE_CODE;
*mode = DMS_MODE;
 peak=127;
 imref=10;
 fs=6103;        //program dependant
Ts=1/fs;
m=Ts/2/peak;
fo=50;           //program dependant
Tx=Ts;       //program dependant
index_ioa=0;
thetaop=0;
thetaop=thetaop;
current_transformation=40;    //2x
 /**********************write to FPGA*******************************/
 iaref2=0;
 ibref2=0;
 icref2=0;
 sec=1;
 t0_=0;
 t1_=0;
 t2_=0;
 da=0;
 db=0;
 dc=0;
 ai_1=0;
 bi_1=0;
 ci_1=0;
 th0=0;
 errd=0;
 iqref=(-122875);
  errq=0;
  w0=50*2*pi;
  flgy=0;
/*******inverter values:******/
  LOAD_FIELD(&INVERTER,iaref2,0,8);
  LOAD_FIELD(&INVERTER,ibref2,8,8);
  LOAD_FIELD(&INVERTER,icref2,16,8);
  LOAD_FIELD(&INVERTER,sec,24,3);
  LOAD_FIELD(&INVERTER,0,27,5);
  /*******************************/

  /*******converter values:******/
 LOAD_FIELD(&CONVERTER,t0_,0,8);
 LOAD_FIELD(&CONVERTER,t1_,8,8);
 LOAD_FIELD(&CONVERTER,t2_,16,8);
 LOAD_FIELD(&CONVERTER,0,27,5);
 LOAD_FIELD(&CONVERTER,da,24,1);
 LOAD_FIELD(&CONVERTER,db,25,1);
 LOAD_FIELD(&CONVERTER,dc,26,1);
 WRITE1_FPGA=INVERTER;
 WRITE2_FPGA=CONVERTER;
 /*******************************/


/************************INTERRUPT*********************************/
```

```
  intr_map(CPU_INT7,ISN_EXT_INT4);
  INTR_GLOBAL_ENABLE();
  INTR_ENABLE(CPU_INT_NMI);
  INTR_ENABLE(CPU_INT7);
  intr_hook(increment,CPU_INT7);

 /*************************************************************************/

 /* end of application dependent code */
 /*---------------------------------------*/

 write_32b_reg(get_cntl_addr(), fpga_io_reg);

#if AED_PRINT

  printf("\n*** AED " AED_BOARD " TEST PROGRAM STARTED ***\n");
#endif

} /* end appl_parms */


/*************************************************************************
  appl_init - Performs buffer init before data transfer
    Parameters:  OUT data_block - pointer to beginning of entire input
                                  block (all frames)
                 IN block_bytes - number of bytes in block
                 IN dma_chan - DMA channel allocated by main for input
    Note: This routine is autmatically called from main before the
          data transfer is intiated, to initialize data buffers.
*************************************************************************/
void appl_init(ApplBlockType data_block,
               unsigned int  block_bytes,
               Dma_channel   dma_chan)
{
 /*---------------------------------------*/
 /* the following code may be application dependent or additional
    code may be required */
 int i, j, k;
 unsigned int *ptr1;

 save_data.byte
  = malloc(SAVE_RECORDS*RECLEN*byte_size(ELEMENTSIZE_CODE));
 if (!save_data.byte) {
  error_flashing(AED_FLASH_APPL_SAVE_BUFFER_MALLOC_ERROR);
 } /* end if */
 fpga_chan = dma_chan;

 ptr1 = data_block.uword;
 for (i=0; i<NO_FRAMES; i++) {
  for (j=0; j<NO_RECORDS; j++) {
   for (k=0; k<RECLEN; k++) {
    *(ptr1++) = 0;
   }
  }
 }

 ptr1 = save_data.uword;
```

```
for (j=0; j<SAVE_RECORDS; j++) {
  for (k=0; k<RECLEN; k++) {
    *(ptr1++) = 0;
  }
}

for (j=0; j<ITERATIONS; j++) {
  A_value[j] = 0;
  B_value[j] = 0;
  C_value[j] = 0;
  D_value[j] = 0;
  E_value[j] = 0;
  F_value[j] = 0;
  G_value[j] = 0;
  H_value[j] = 0;
  I_value[j] = 0;
  J_value[j] = 0;
  K_value[j] = 0;
  L_value[j] = 0;
  M_value[j] = 0;
  N_value[j] = 0;
  O_value[j] = 0;
  P_value[j] = 0;
 } /* end for */

 loop_count = 0;

 input_count = 0;

#if AED_PRINT
 printf("Begin application processing (Block size = %d bytes)\n",
        block_bytes);
#endif
 /* end of application dependent code */
 /*---------------------------------------*/
 /* the following code line is removed when daughterboard is
    added */

} /* end appl_init */


/*****************************************************************
 appl_process - Processes 1 frame of buffer data
  Returns:    user defined termination code, 0 is no termination
  Parameters: IN data_buffer - pointer to beginning of frame of
                       data just received from FPGA
              IN buf_number - number of buffer just received
  Note: This routine is autmatically called from main when a full
      buffer of data has been transferred from the daughterboard.
*****************************************************************/
int  appl_process(ApplBlockType data_buffer,
             int buf_number)
{

 /*---------------------------------------*/
 /* the following code is application dependent */

 int xx=0;
```

```
kk=0;
xxx=0;
j=0;

index_sec=0;
nnn=0;
mmmm=0;
xxxx=0;
lx=0;
k=0;
 while (XSTAT0==0)
{;}
        do
        {
        ptr=data_buffer;

 ADC[1]= (unsigned int)*(ptr.uhword++);
 ADC[2]= (unsigned int)*(ptr.uhword++);
 ADC[0]= (unsigned int)*(ptr.uhword++);
 ADC[11]= (unsigned int)*(ptr.uhword++);
 ADC[4]= (unsigned int)*(ptr.uhword++);
 ADC[5]= (unsigned int)*(ptr.uhword++);
 ADC[6]= (unsigned int)*(ptr.uhword++);
 ADC[7]= (unsigned int)*(ptr.uhword++);
 ADC[8]= (unsigned int)*(ptr.uhword++);
 ADC[9]= (unsigned int)*(ptr.uhword++);
 ADC[10]= (unsigned int)*(ptr.uhword++);
 ADC[3]= (unsigned int)*(ptr.uhword++);
 ADC[12]= (unsigned int)*(ptr.uhword++);
 ADC[13]= (unsigned int)*(ptr.uhword++);
 ADC[14]= (unsigned int)*(ptr.uhword++);
 ADC[15]= (unsigned int)*(ptr.uhword++);
 sum+=ADC[3];
 k++;
 if (k==8)
{
 av=sum/8;
 k=0;
 sum=0;
if ((av<1980)&&(av>1960))
{

         nnn=1;

        }
        }
if (nnn==1)
   {
   if (ADC[3]>s)
    {
    sec=6;
    thetaip=84;
    mmmm=1;
    vvv=1;
    index_ip=24;
```

```
                    }
                   if (ADC[3]<s)
                    {
                    sec=3;
                    thetaip=84;
                    mmmm=1;
                    vvv=0;
                    index_ip=24;
                    }
                  }
          s=ADC[3];
         }
 while (mmmm==0);

while (xx==0)
{



kkkkk++;
iii=kkkkk/4;
if (iii>140)
{ kkkkk=0;
 if (flgy==0)
  flgy=1;
 else flgy=0;
 }

 FPGA_stop();

        FPGA_start();

 /*************************************************************/

/*********************READ FROM FPGA*************************/
j=0;
sum=0;
iaaaa=0;
ibbbb=0;
sum2=0;
ax=0;
do
{
          while (XSTAT0==0)
          {
           };
          ptr=data_buffer;
          flgx=1;
ADC[1]= (unsigned int)*(ptr.uhword++);
ADC[2]= (unsigned int)*(ptr.uhword++);
ADC[0]= (unsigned int)*(ptr.uhword++);
```

```
ADC[11]= (unsigned int)*(ptr.uhword++);
ADC[4]= (unsigned int)*(ptr.uhword++);
ADC[5]= (unsigned int)*(ptr.uhword++);
ADC[10]= (unsigned int)*(ptr.uhword++);
ADC[7]= (unsigned int)*(ptr.uhword++);
ADC[8]= (unsigned int)*(ptr.uhword++);
ADC[9]= (unsigned int)*(ptr.uhword++);
ADC[6]= (unsigned int)*(ptr.uhword++);
ADC[3]= (unsigned int)*(ptr.uhword++);
ADC[12]= (unsigned int)*(ptr.uhword++);
ADC[13]= (unsigned int)*(ptr.uhword++);
ADC[14]= (unsigned int)*(ptr.uhword++);
ADC[15]= (unsigned int)*(ptr.uhword++);
                        sum+=ADC[3];
                    iaaaa+=ADC[10];
                    ax+=ADC[11];
                    ibbbb+=ADC[14];
                    j++;
            }
            while (j<4);
            ADC[3]= sum/4;
            ADC[10]=iaaaa/4;
            ADC[14]=ibbbb/4;
            ADC[11]=ax/4;

/*************************Vdq*****************************/

  a1=720-(thk*360/pi);
  a2=a1-180;
  if (a2>720) a2-=720;
  if (a2<0) a2+=720;
  vab=(ADC[3]-1990)*1000;
  vbc=(ADC[11]-1975)*1000;
          vab=(vab+4*vab1)/5;
          vbc=(vbc+4*vbc1)/5;
          vbc1=vbc;
          vab1=vab;
          vab=vab/1000;
          vbc=vbc/1000;
valpha=(3*vab)/2;
vbeta=(0.86602540378*(-vab-2*vbc));
vd=(((valpha*sine[a2])+(vbeta*sine[a1]))/8192);
vq=((((-1)*valpha*sine[a1])+(vbeta*sine[a2]))/8192);
vd=(vd+99*vd1)/100;
vq=(vq+99*vq1)/100;
vd1=vd;
 vq1=vq;
 if (vq1==0) vq=0.000001;
 wk=w0+(AA*vd1/vq1);
 if (wk>51)
        wk=51;
elseif (wk<49)
        wk=49;
end;
thk=th0+(wk/fs)+ (BB*vd1/vq1);;
while (thk>(2*pi)) thk-=(2*pi);
```

```
while (thk<0) thk+=(2*pi);
w0=wk;
th0=thk;
thetaip2=thk*360/pi-360;
thetaop=thetaip2-18+360-60+(90*2);
thetaip2=thetaip2;
if (thetaip2<0) thetaip2+=720;
if (thetaip2>=720) thetaip2-=720;
if (thetaop<0) thetaop=720;
if (thetaop>=720) thetaop-=720;
sec2=(thetaip2/120)+1;
if (sec2>6) sec2-=6;
thetaip2=thetaip2;
if (thetaip2<0) thetaip2+=720;
if (thetaip2>=720) thetaip2-=720;
 index_t2=thetaip2-((sec2-1)*120);
 index_t1=120-index_t2;
 index_ioa90=index_ioa-180;
 if (index_ioa90<=0)
index_ioa90+=720;

/**********actual current**********/
 iaact=current_transformation*(ADC[10]-1955);
 ibact=current_transformation*(ADC[14]-1987);


 /**********dq controller**********/
 /**********integral**********/
 ki=0;
 kc=0;
 errd+=(ki*(idref-id));
 errq+=(ki*(iqref-iq));
 ud=((kp*(idref-id))+errd+(kc*((iq+iqref)/2)))/8192;
 uq=((kp*(iqref-iq))+errq-(kc*((id+idref)/2)))/8192;
 ualpha=(((ud*sine[index_ioa90])-(uq*sine[index_ioa])));
 ubeta=(((ud*sine[index_ioa])+(uq*sine[index_ioa90])));
 iaerr=((int)(0.8164965809*ualpha))/8192/32;
 iberr=((int)((-0.4082482905*ualpha)+(0.7071067812*ubeta)))/8192/32;
 icerr=-iaerr-iberr;


/**********************************************************/

/**********current direction**********/
//implemented on FPGA

//needs to be revised
if (iaact<0)
 da=0;
else
  da=1;

if (ibact<0)
  db=0;
else
  db=1;

if (icact<0)
```

```
  dc=0;
 else
  dc=1;

 /*************************************************************/



/*********************** t1, t2 & t0*****************************/

t1=((peak)*sine[index_t1])/8192;
t2=((peak)*sine[index_t2])/8192;
 t0=(peak -t1-t2)/2;

 /*************************************************************/

 /************************time calculations********************/

t0_=(t0);
t1_=t0_+(t1);
t2_=t1_+(t2);

/*************************************************************/



/**********inverter values*********/
LOAD_FIELD(&INVERTER,iaerr,0,8);
 LOAD_FIELD(&INVERTER,iberr,8,8);
 LOAD_FIELD(&INVERTER,icerr,16,8);
 LOAD_FIELD(&INVERTER,sec,24,3);
 LOAD_FIELD(&INVERTER,0,27,5);



 /*********************************/
 /*******converter values:******/

 LOAD_FIELD(&CONVERTER,t0_,0,8);
 LOAD_FIELD(&CONVERTER,t1_,8,8);
 LOAD_FIELD(&CONVERTER,t2_,16,8);
 LOAD_FIELD(&CONVERTER,0,27,5);
 LOAD_FIELD(&CONVERTER,da,24,1);
 LOAD_FIELD(&CONVERTER,db,25,1);
 LOAD_FIELD(&CONVERTER,dc,26,1);
 xxxx++;
 /*********************************/



        while ((flgx==1))
        {
```

```
            zxc++;

            };
  }
  return 0;
} /* end appl_process */


/*****************************************************************
  appl_idle - Performs background processing
    Returns:   user defined termination code, 0 is no termination
    Note: This routine is automatically called from main when no other
          processing is required, but may not be called regularly.
*****************************************************************/
int  appl_idle(void)
{
  /*-------------------------------------*/
  /* the following code is application dependent */
  loop_count++;
  /* end of application dependent code */
  /*-------------------------------------*/
  return 0;
} /* end appl_idle */


/*****************************************************************
  appl_end - Final processing before termination
    Parameters:  IN times         - main program loop cycles executed
                 IN bufs_proc     - number of buffers processed by
                                    appl_process
                 IN buf_count     - number of buffers received
                 IN prev_buf_count - last buffer given to appl_process
                 IN DMS_err       - error code from DMS
                 IN appl_term_code - user termination code from either
                                     appl_process or appl_idle
                 IN FIFO_ovfl     - indication that the FIFOin the
                                    FPGA has overflowed
                 IN DMS_count     - number of frames remaining to be
                                    received in the block
    Note: This routine is automatically called from main at program
          termination.
*****************************************************************/
void appl_end(unsigned int times,
              int        bufs_proc,
              int        buf_count,
              int        prev_buf_count,
              int        DMS_err,
              int        appl_term_code,
              unsigned int FIFO_ovfl,
              unsigned int DMS_count)
{
  /*-------------------------------------*/
  /* the following code is application dependent */


  /* end of application dependent code */
  /*-------------------------------------*/
} /* end appl_end */
```

```
/*****************************************************************
  appl_test - Fills separate test block to simulate data from FPGA
    Parameters:  OUT fill - pointer to beginning of test block
                 IN frame_bytes - number of bytes in each frame
                 IN frames - number of frames in block
    Note: This routine is autmatically called from main to initialize
          the test buffer with data.
*****************************************************************/
#if(!TALK_TO_FPGA)
#define SAMPLE_RATE_KHZ  10000L   /* Sample rate for timer setup */
unsigned long  appl_test(ApplBlockType fill,
                         int frame_elements,
                         int frames)
{
 /*-----------------------------------------*/
 /* the following code is application dependent */
 int i, j;
 int samples_per_element = (byte_size(ELEMENTSIZE_CODE)/sizeof(short));
 int frame_samples = frame_elements*samples_per_element;

 /* period is in 0.5 microseconds units */
 /* for a time between frames (frame sync)
   period = (frame_elements/SAMPLE_RATE)/(0.5e-6 us) */
 unsigned long period = (2000L*(long)frame_elements)/SAMPLE_RATE_KHZ;

 /* fill can be changed to accomodate testing */
 for (i=0; i<frames; i++) {
  for (j=0; j<frame_samples; j++)  {
    fill.uhword[i*frame_samples+j]
       = ((i+1)*0x1000)+(j+1);
   } /* end for */
  } /* end for */
  /* end of application dependent code */
  /*-----------------------------------------*/
  return period;
} /* end appl_test */
#endif

interrupt void increment(void)
{
     WRITE1_FPGA=INVERTER;
    WRITE2_FPGA=CONVERTER;

     index_ioa=(int)(thetaop);
    index_iob=index_ioa-240;
     if (index_iob<=0) index_iob+=720;

     flgx=0;

     return;
}
```

LIST OF FIGURES