

THESIS

Roland Tóth

2004

University of Veszprém, Hungary
Department of Automation
Faculty of Information Technology

THESIS

**Implementation of a Speed Sensorless
Induction Motor Control on TMS320F243**

Roland Tóth

Supervisor: Dénes Fodor Dr.

2004





UNIVERSITY OF VESZPRÉM

**Department of Automation
Faculty of Information Technology**

H-8200 Veszprém, 10 Egyetem St., Hungary

Tel.: +36(88)422022

Fax: +36(88)427633

24th March 2004, Veszprém

DIPLOMA TOPIC

For **Roland Tóth**

3rd year Electrical Engineering student

Title of the thesis:

Implementation of a Speed Sensorless Induction Motor Control on TMS320F243

Induction motors are widely used in the industry due to their simple structure, low cost, and high reliability. Although they are the horsepower of industry, their control is significantly more challenging than of dc motors. Nowadays, therefore, there is a great interest in developing high performance control devices to enhance the operation of induction drives in all fields of applications. Especially, these efforts concentrate on controllers that do not need speed sensors to operate, which greatly reduces costs and maintenance. The goal of this diploma thesis is to implement a speed sensorless controller on a TMS320F243 DSP based VSI-fed induction motor drive.

Thesis work assignments:

- study the existing speed sensorless techniques (observer based, MRAC, DTC, Sliding mode) and the modern inverter related control technologies (space vector PWM, sync. PWM,...)
- present the selected speed sensorless algorithm and the mathematical and physical considerations of its design
- develop the implementation of the algorithm on a TMS320F243 EVM and realize the closed loop drive with the direct control of a DigitalSpectrum inverter fed induction motor.

Subject groups of the final examination:

- Analóg áramkörök: (*Analog Circuits*)
- Irányítástechnika (*Control Techniques*)
- Digitális áramkörök (*Digital Circuits*)
- Szabályozott villamos hajtások (*Control of Electrical Drives*)

.....

Dénes Fodor Dr.
Supervisor

.....

József Vass Dr.
Head of the Department

Nyilatkozat

Alulírott Tóth Roland, diplomázó hallgató, kijelentem, hogy a szakdolgozatot a Veszprémi Egyetem Automatizálás Tanszékén készítettem el főiskolai villamosmérnök diploma (Bachelor of Electrical Engineering) megszerzése érdekében.

Kijelentem, hogy a szakdolgozatban foglaltak saját munkám eredményeit, és csak a megadott forrásokat (szakirodalom, eszközök, stb.) használtam fel.

Tudomásul veszem azt, hogy a szakdolgozatban foglalt eredményeket a Veszprémi Egyetem, valamint a feladatot kiíró szervezeti egység saját céljaira szabadon felhasználhatja.

Veszprém, 2004. május 15.

.....
Tóth Roland

Köszönetnyilvánítás

Mindenekelőtt szeretném megköszönni Szüleimnek, hogy lehetővé tették számomra egyetemi tanulmányaimat.

Szeretnék továbbá köszönetet mondani témavezetőmnek, Dr. Fodor Dénesnek, valamint a Veszprémi Egyetem Automatizálás Tanszékének, amiért munkámat magas szintű technikai feltételek mellett végezhettem, továbbá Dr. Szederkényi Gábornak és Bognár Endrének a szakmai segítségért.

Külön köszönöm Salekovics Ádámnak a nyelvi nehézségek leküzdésében nyújtott kitartó segítséget.

Hálás vagyok még kedvesemnek és barátaimnak, akik biztatásukkal és türelmükkel nagyon sokat segítettek.

TARTALMI ÖSSZEFOGLALÓ

A szakdolgozat az aszinkronmotor robusztus fordulatszám-érzékelő nélküli szabályozását megvalósító rendszer implementációját mutatja be egy TMS320F243 processzor vezérelt feszültség betáplálású inverter (VSI) által üzemeltett laboratóriumi hajtáson.

A szabályozás és rendszer elmélet ezen területén fellelhető modern elméletek (mint a politopikus rendszer reprezentáción történő LPV H_∞ -szabályozó és -megfigyelő szintézis, a súlyozott érzékenységgű struktúrák, a kis erősítések tétele és a kiterjesztett Kálmán szűrők (EKF) teóriája) alapján megtervezett és bemutatásra kerülő szabályozó a motor direkt fluxus és fordulatszám szabályozását teszi lehetővé a teljes 4/4-es elméleti hajtási tartományon zajos ipari környezet hatásit is figyelembe véve. Az eredményül kapott hajtás képes alkalmazkodni a motor tengelyén fellépő nyomatékváltozásokhoz és teljesíti a manapság egyre növekvő villamos hajtásokkal szemben támasztott ipari elvárásokat, mint a fordulatszám érzékelő nélküliség (csak a fázis áramok mérése), robusztus stabilitás és a zaj érzéketlenség. A fenti elméleti ismeretek alapján tervezett zárt szabályozó kör Matlab/Simulink környezetében készült implementációját is ismertetjük melyen elvégzett szimulációk eredményei alapján a struktúra az elvárásoknak megfelelően működött.

A dolgozatban bemutatásra kerülnek a jelenleg iparilag használt fordulatszám érzékelő nélküli szabályozást biztosító megoldások, mint a referencia motormodellek, adaptív megfigyelők, MRAS, DTC, és vektor alapú technikák, melyek összehasonlításra kerülnek az általunk adott szabályozási struktúrával. Ezenkívül, a mai aszinkronmotoros hajtások elvi és gyakorlati felépítését is megvizsgáljuk a gerjesztés előállítására szolgáló impulzus szélesség modulációs (PWM) technikák bemutatása mellett.

Az implementáció alapjául szolgáló mikrokontroller vezérelt térvektor (SV)-PWM VSI betáplálású hajtásláncot részletesen elemezzük az általunk tervezett szabályzó megvalósítási lehetőségeinek függvényében, és az implementáció által a Code Composer fejlesztői környezetben létrehozott ANSI C programot valamint a hajtásban felhasznált aszinkronmotor paraméter azonosítását is ismertetjük. A dolgozatot a mérések és a szimulációk alapján kapott eredmények vizsgálata zárja a hajtás jóságára, hatékonyságára, robusztusságának és stabilitására vonatkozó következtetések és a továbbfejlesztési lehetőségek ismertetése mellett.

Kulcsszavak: aszinkronmotor, H_∞ , LPV, EKF, súlyozott érzékenység, fordulatszám-érzékelő nélküli szabályozás, TMS320F243 EVM, DTC, MRAS, SV-PWM

ABSTRACT

The thesis shows the implementation of a robust control structure for the speed sensorless vector control of the induction motor (IM) on a TMS320F243 processor controlled voltage source inverter (VSI) driven laboratory induction motor drive.

The designed robust controller whose design is based on the modern achievements of Control and Systems theory (such as the polytopic system representation based H_∞ controller and observer synthesis, the mixed sensitivity structures, small gain theorem, and the theory of extended Kalman filters (EKF)) is presented which makes possible the direct control of the flux and speed on the full 4/4 operation range of the motor with torque adaptation in highly noisy industrial environment. In this way the produced structure fulfills the recently growing expectations for industrial drives as the control without speed sensors (with the measurements only of the phase currents), robust stability, and noise attenuation. The designed closed loop control system is tested by intensive Matlab/Simulink simulations that are presented in the thesis to prove the goodness of the solution, which according to the results shows good dynamic and robust performance.

In this paper the recently used speed sensorless techniques in the industry, such as the direct reference models, adaptive observers, MRAS, DTC, and vector control based methods are shown and compared to the designed controller structure. Moreover, the theoretical and practical concepts of the modern IM drives and the pulse width modulation (PWM) based the power feed generation methods of these devices are also presented.

The microcontroller driven space vector (SV)-PWM VSI fed drive, which is used for the implementation of the controller, is analyzed in details with respect to the concepts of the controller realization. The implementation resulted ANSI C program in the Code Composer development environment is also explained with the parameter identification of the applied motor of the laboratory drive. The experienced measurement and simulation outcomes of the implementation are evaluated in terms of effectiveness, robustness, and stability and the obtained conclusions are drawn. Finally, possible improvements of the control structure and future applications are pointed out as well.

Key words: induction motor, H_∞ , LPV, EKF, mixed sensitivity, speed sensorless control, TMS320F243 EVM, DTC, MRAS, vector control, SV-PWM

Table of contents

Abbreviations.....	vi
1. Introduction.....	1
2. The mathematical model of the induction motor.....	4
2.1. Basic concepts of operation.....	4
2.2. Modeling conditions.....	6
2.3. General mathematical description.....	6
2.3.1. Voltage and flux equations.....	9
2.3.2. The dynamical motion equation.....	11
2.3.3. Approximation of rotor resistance variation.....	12
2.4. Model representation in rotating reference frame.....	13
2.4.1. General reference orientation.....	15
2.4.2. Rotor flux oriented representation.....	16
2.5. The linear parameter variant motor model.....	17
2.6. General dynamic properties of the model.....	20
3. Existing speed sensorless techniques.....	22
3.1. Constant Volts per Hertz control.....	22
3.2. Machine models.....	26
3.2.1. Direct reference models.....	27
3.3. Adaptive observers & filters.....	29
3.3.1. Full order nonlinear observer.....	29
3.3.2. Sliding mode observer.....	31
3.3.3. Extended Kalman Filter.....	32
3.3.4. H_{∞} based LPV observer.....	34
3.3.5. Model reference adaptive systems.....	36
3.4. Direct torque control.....	38
3.5. Vector control.....	44
3.6. Fuzzy control.....	46
4. Operation of induction motor drives.....	49
4.1. Classical power feed generation concepts.....	49
4.1.1. Rectifier.....	50
4.1.2. DC link.....	52
4.1.3. Inverters.....	52
4.1.4. Inverter control circuitry.....	53
4.1.5. The microcontroller.....	54
4.2. The PWM inverters.....	54
4.2.1. The insulated-gate bipolar transistor.....	54

4.2.2. The mechanism of PWM-VSI inverters	55
4.2.3. Different techniques of the 3-phase PWM generation on VSI-PWM inverters	62
4.2.3.1. Synchronous symmetric PWM.....	62
4.2.3.2. Synchronous asymmetric PWM.....	62
4.2.3.3. Asynchronous PWM.....	62
4.2.3.4. Space vector PWM	63
4.2.3.5. Overmodulated PWM.....	65
5. The experimental laboratory drive.....	67
5.1. The induction motor	68
5.2. The Spectrum Digital inverter	70
5.2.1. Specifications of the inverter module.....	70
5.2.2. Capabilities of the inverter module	71
5.2.3. Sensors	72
5.2.4. Built in protections	73
5.3. The Inverter Interface Card	73
5.3.1. Specifications of the interface module	74
5.3.2. I/O signal conditioning	74
5.3.3. Inverter Digital Interface	75
5.3.4. Shield of the drive.....	76
5.4. The TMS320F243 Evaluation Module.....	76
5.4.1. Specifications of the EVM.....	77
5.4.2. Considerations of controller implementation	78
5.4.3. Chip of many things.....	78
5.4.3.1. Speed and memory	79
5.4.3.2. Interrupts and peripherals	80
5.4.3.3. Digital I/O pins	82
5.4.3.4. Event manager (EVM2).....	82
5.4.3.4.1. Timers	82
5.4.3.4.2. PWM generation.....	84
5.4.3.4.3. Deadband unit.....	87
5.4.3.4.4. Capture units.....	87
5.4.3.4.5. Quadrature Encoder Pulse (QEP) Circuit.....	88
5.4.3.4.6. Analog to digital converter (ADC).....	89
5.4.3.5. Communication interface.....	90
5.4.3.5.1. Serial Communications Interface (SCI)	90
5.4.3.5.2. Serial Peripheral Interface (SPI).....	91
5.4.3.5.3. Controller are network interface (CAN).....	91
5.4.3.6. Watchdog.....	92

5.4.4. Heart and soul of the drive.....	92
5.5. The PC	93
5.5.1. The XMS510P Plus JTAG driver	93
5.5.2. The Code Composer environment	93
5.5.3. The operator interface.....	94
6. Implementation of the controller	95
6.1. The theoretical structure of control.....	96
6.1.1. The H_{∞} controller module.....	98
6.1.2. I/O linearization based reference transformation	99
6.1.3. Rotor flux orientation and flux angle tracking	101
6.1.4. The structure of estimation	102
6.1.5. The H_{∞} observer.....	103
6.1.6. The EKF.....	104
6.1.7. Estimation of the load torque.....	106
6.1.8. Tuning parameters	106
6.2. Simulation results	107
6.3. The identification of the motor	114
6.3.1. Method of identification	115
6.3.2. Measurements	118
6.3.3. Results of the identification	120
6.3.4. Validation of the model	121
6.4. The program of the DSP	121
6.4.1. Flow chart of the program	122
6.4.2. The hardware boot up	123
6.4.3. System initialization and hardwired constants	124
6.4.4. Control loop	125
6.4.5. Process of ADC	125
6.4.6. Processes of estimation.....	127
6.4.7. Process of control	128
6.4.8. Process of PWM generation	128
6.4.9. Additional features.....	129
6.5. Speed considerations	129
7. Conclusion	132
References.....	134

Table of notations

$n(t)$	rotor speed [revolution/sec]	R_{sx}	stator side resistance of phase x [Ω]
n_0	synchronous frequency [Hz]	R_s	stator side 3 phase resistance [Ω]
f_0	input synchronous frequency [Hz]	R_r	rotor side 3 phase resistance [Ω]
p	number of pole pairs	R_0	initial value of R_r [Ω]
$\omega(t)$	angular speed of the rotor [rad/sec]	L_s	lumped stator 3 phase induc. [H]
ω_0	sync. angular speed [rad/sec]	L_r	lumped rotor 3 phase induc. [H]
$\omega_k(t)$	angular speed of rotating reference frame [rad/sec]	L_m	lumped mutual 3 phase induc. [H]
$\omega_{\text{flux}}(t)$	ang. speed of rotor flux [rad/sec]	l_s	stator winding phase inductivity [H]
$\omega_{\text{ref}}(t)$	ref. signal of rotor speed [rad/sec]	l_r	rotor winding phase inductivity [H]
$s(t)$	slip	l_f	field inductivity [H]
$F_{sx}(\alpha, t)$	current density distribution in phase x [A/rad]	l_m	mutual inductivity [H]
$i_{sx}(t)$	stator current in phase x [A]	C	capacity [F]
$u_{sx}(t)$	stator voltage in phase x [V]	$\rho(t)$	angle between the stator fixed and the rotor fixed reference frame [rad]
Ψ	magnetic flux [Wb]	$\rho_k(t)$	angle between the stator fixed and the rotating reference frame [rad]
ϕ_i	init. ang. of stat. current vec. [rad]	$P_{\text{mech}}(t)$	mechanical power [W]
ϕ_u	init. ang. of stat. voltage vec. [rad]	$W_{\text{mech}}(t)$	mechanical energy [J]
\mathbf{a}	three-phase unity vector	$W_e(t)$	electrical input power [J]
\mathbf{x}^s	space vector in stat. fixed rep.	$W_v(t)$	resistive power loss [J]
\mathbf{x}^r	space vector in rotor fixed rep.	$W_{\text{field}}(t)$	air gap power of magnetic field [J]
\mathbf{x}^{fr}	space vect. in rotor field orient. rep.	$T_e(t)$	electromagnetic torque [Nm]
\mathbf{x}^k	space vect. in rotating ref. frame	$T_{\text{load}}(t)$	load torque [Nm]
x_d	real part of space vector x in rotating reference frame	τ_{mech}	mechanical time constant [Nm]
x_q	imaginary part of space vector x in rotating reference frame	τ_e	electrical time constant [Nm]
x_α	real part of space vector x in stator fixed reference frame	τ_s	stator time constant (constant)
x_β	imaginary part of space vector x in stator fixed refernce frame	τ_r	rotor time constant (constant)
x_{nominal}	nominal value of signal x	τ	m. coupling time constant (constant)
x_{ref}	reference signal of x	σ	leakage factor (constant)
x_{error}	error signal of x	λ	notation, def. pp. (constant)
$\mathbf{i}_{sa}^x(t)$	space vector describing current density distribution of phase x [A]	J	moment of inertia [Nm]
$\mathbf{i}_s^x(t)$	space vector describing the overall stator side cur. density distrib. [A]	$T(t)$	rotor temperature [K]
$\mathbf{i}_r^x(t)$	space vector describing the overall rotor side cur. density distrib. [A]	T_0	initial rotor temperature [K]

Table of notations

$\Psi_s^x(t)$	space vec. describing the overall stator side flux linkage distrib. [Wb]	$Q(t)$	heat [kcal]
$\Psi_r^x(t)$	space vec. describing the overall rotor side flux linkage distrib. [Wb]	c	specific heat ct. Al [J/kgK]
$\Psi_\sigma^x(t)$	space vec. describing the overall leakage flux distrib. [Wb]	m	weight of the rotor winding [kg]
$\mathbf{u}_s^x(t)$	space vector describing the overall stator side volt. density distrib. [V]	K_k	linear heat convection (constant)
$\mathbf{u}_r^x(t)$	space vector describing the overall rotor side volt. density distrib. [V]	F	fraction (constant)
$i_{eff}^r(t)$	rms value of the rotor phase cur. [A]	h	discrete time step [sec]
$i_{eff}^s(t)$	rms value of the stator p. cur. [A]	t	time [sec]
$u_{eff}^s(t)$	rms value of the stator p. volt. [V]	v, α	angle [rad]
$i_{sp}(t)$	active stator current [A]	\mathbf{K}	contr. of general problem definition
U_{bus}	inverter bus voltage	\mathbf{P}	considered system (<i>plant</i>) of the general problem definition
\mathbf{u}_x	x^{th} switching state of the inverter	\mathbf{p}	parameter vector
$\mathbf{u}_{x \text{ or } x \pm 60}$	voltage vect. of sector x (SV-PWM)	\mathbf{x}	state vector of the system
$u_{in}(t)$	control voltage of the PWM	$\mathbf{G}(s)$	frequency transition func. of the sys.
$u_{x,y}(t)$	x or y ref. voltage of the PWM	$\mathbf{x}(k k)$	estimated state vector based on k pervious estimations
$u_{out}(t)$	output voltage of the PWM	$\mathbf{P}(k k)$	covariance of the error process
U_0^{out}	amplitude of the fundamental frequency component	$\tilde{\mathbf{x}}$	error of the estimation
m_{FM}	frequency modulation value	\mathbf{y}	output vector of the system
m_{AM}	amplitude modulation value	\mathbf{u}	control/general system input
V_p	peak voltage value	\mathbf{v}	vector of measured output signals
V_{pp}	peak to peak voltage value	\mathbf{z}	system output vector (optimization)
$\mathbf{S}_x(\hat{\omega})$	complex gain factor	\mathbf{w}	input disturbances of the system
$\mathbf{f}(\mathbf{x}, \mathbf{u})$	state function (non linear state space sys. description)	\mathbf{e}	error signals of the system
$\mathbf{f}(\mathbf{x})$	state func. of an input-affine model	\mathbf{R}	covariance of measurement noises
$\mathbf{f}_d(\mathbf{x})$	discretised state function	\mathbf{Q}	covariance of system noises
$\mathbf{h}(\mathbf{x})$	output function (non linear state space sys. description)	γ	optimal H_∞ gain
$\mathbf{h}_d(\mathbf{x})$	discretised output function	v_1, v_2	virtual inputs
$\mathbf{g}(\mathbf{u})$	input func. of an input-affine model	$\hat{\mathbf{x}}$	predicted / normalized value
$\mathbf{g}_d(\mathbf{u})$	discretised input function	δ	scaling step size/Re. of comp. freq.
Ξ	sequence of measured values	T_1, T_2	Modulation times in the SV-PWM
θ	parameter values	α_u	angle of stator voltage vec. [rad]
$\vec{\mathbf{G}}_1(\cdot)$	first order gradient vector	Δ	change / error of a given variable
$\vec{\mathbf{G}}_2(\cdot)$	second order gradient vector	λ_{filter}	gain of the exponential filter

(A, B, C, D, E)

matrices of the continuous state space representation of the system

(Φ, Γ, C, D)

matrices of the discrete state space representation of the system

Abbreviations

AC	Analog Current/Voltage	MIPS	Million Instructions Per Second
ACG	Automated Code Generation	MOSFET	Metal Oxide Semiconductor Field Effect Transistor
AI	Artificial Intelligence	MRAS	Model Reference Adaptive Control
AS	Asymmetric Synchronous	NL	Nonlinear (system)
CAN	Controller Area Network	NMI	Non Maskable Interrupt
CPU	Central Processing Unit	NRZ	Non Return to Zero
DC	Direct Current/Voltage	PC	Personal Computer
DSP	Digital Signal Processor	PIC	Programmable Integrated Circuit
DTC	Direct Torque Control	PID	Proportional, Integrational, Derivational (control)
EC	External Commutation	PWM	Pulse Width Modulation
EKF	Extended Kalman Filter	QEP	Quadrature Encoder Pulse
EVM	Evaluation Module	RAM	Random-Access Memory
EVM2	Event Manager	ROM	Read-Only Memory
FIFO	First In First Out (data storage)	RTDX	Real-Time Data Exchange
FLC	Fuzzy Logic Control	SC	Self Commutation
GEL	General Extension Language	SCI	Serial Communication Interface
GP	General Purpose	SPI	Serial Peripheral Interface
IDE	Integrated Development Environment	SS	Symmetric Synchronous
IGBT	Isolated Gate Bipolar Transistor	SV-PWM	Space Vector based Pulse Width Modulation
JTAG	Joint Test Action Group (IEEE 1149.1)	TI	Texas Instruments
LPV	Linear Parameter Variant (system)	UART	Universal Asynchronous Receiver/Transmitter
LQG	Linear Quadratic Gaussian Control	UML	Unified Modeling Language
LTI	Linear Time Invariant (system)	VSI	Voltage Source Inverter
LTV	Linear Time Variant (system)	WD	Watch Dog

emf. electromagnetic force
mmf. magnetomotoric force

1. Introduction

Induction motors are widely used in the industry due to their simple structure, low cost, and high reliability. Although they have become dominant only in the past decades on the field of variable speed electrical drives, their economical importance is noteworthy with more than 12 billion US\$ world market volume, of which the annual growth rate is 15%. [18]

The construction of the first induction motor (IM) was a significant breakthrough in the field of electrical machines. In 1888, Tesla received half a million dollars from the Westinghouse for the patent of his invention. Unfortunately, this was followed by 60 years of banishment, because it quickly turned out that the control of this electrical machine is a challenging task [31]. Difficulties had risen because of its highly nonlinear nature and the low technology level of Power Electronics. The nonlinear dynamic behavior of this system, which is even effected by parameter disturbances, prevented the effective use of the analog control methods. Moreover, the extremely important strength and orientation of the magnetic field, described by the magnetic flux (Ψ), can be only expensively and inaccurately measured, so its control needs mathematical estimation. Furthermore, some of the parameters of the IMs, like the variation of the rotor resistance (R_r), introduce large uncertainties into the system behavior.

These were the reasons for the golden age of DC motors which provide easy controllability and power management with a more complicated construction [19]. By the improvement of technology, like the evolution of control and system theories and the important results in Power Electronics, as the *isolated gate bipolar transistor* (IGBT) and the cheap, reliable, and high frequency inverters, the induction drives have slowly overtaken this field and become the horsepower of the industry, providing more than 90% of today's industrial drives. The success of the IMs can be explained by their simple structure and the need of less maintenance because of the absence of graphite brushes.

Only in those fields which need simple control and high precision preserved the DC motors their leading role. Nowadays, therefore, there is a great interest in

developing high performance and robust controllers to make induction drives unbeatable in all fields of applications. Especially, these efforts concentrate on controllers that do not need speed sensors to operate, which greatly reduces costs and maintenance. Modern industrial drives also need to be insensitive for the heavy noises of the industrial environment and be able to adapt to parameter uncertainties related to manufacturing.

Motivated by this goal and based on modern mathematical methods and physical considerations, we show the implementation of a robust, speed sensorless controller on a TMS320F243 *digital signal processor* (DSP) controlled induction motor drive that gives the opportunity of fast control of the motor speed and the magnetic field associated with the rotor flux ($\Psi_r = [\Psi_{r\alpha}, \Psi_{r\beta}]^T$) on the full 4/4 operation range of the motor. The voltage feed of the IM is generated by a *voltage source inverter* (VSI), which is connected with an interface card to the DSP. The full operation of the drive is shown with the steps of the implementation of the obtained controller. Moreover, as the result of the implementation, the produced DSP program with its unique methods is presented in C as well. According to the theoretical and practical results, this realization holds the possibility of good performance and fulfills the needs of the modern industrial expectations.

To be able to explain the solution of the considered control problem, firstly the dynamic mathematical model of the IM is introduced with the self constructed *linear parameter variant* (LPV) system model of the motor. Then, the main dynamic properties of the given models are analyzed.

In the next section the existing speed sensorless techniques are presented briefly to provide good look out to the most up-to-date techniques used in sensorless IM drives. The usage and effectiveness of the showed methods are analyzed and their disadvantages and applicabilities are pointed out.

In the third section the problem is investigated from the point of view of power systems. The existing techniques to provide the power feed of the drive are presented with the mechanism of the inverters.

Then the self-assembled laboratory drive is focused on, with the description of the physical layout and by the introduction of each hardware element of the drive.

Several issues of implementation are analyzed and the specification of an applicable control structure is given.

In the sixth section the implemented control algorithm is introduced and investigated from both theoretical and practical aspects. The results of the numerically simulated performance are presented in the case of intensive and dynamic changes of the operation environment to prove the effectiveness of the algorithm. The nonlinear parameter identification of the motor model is also analyzed, based on the results of measurements and on the theory of the gradient method. Furthermore, the self-written DSP program, which makes possible the implementation of the presented algorithm in C is also given and analyzed in the point of view of optimality.

Finally, the results of the implemented drive are given and compared with other published results in this field. Furthermore, at the end, the further possibilities of improvements of the system are considered with the ongoing research efforts.

2. The mathematical model of the induction motor

2.1. Basic concepts of operation

In the the following section, the structure of the induction motors and their main features of functioning are introduced based on [6, 12, 20, 21, 26, 32].

The structure of the IMs can be divided into two parts, namely the stator and the rotor, as it can be seen in *Figure 2.1*. The stator, which is built up from good flux conducting plates, is held together by a cast-iron hull. In the other part, the voltage powered stator winding is situated in the shaft-oriented rabbets on the inner side cylinder of the plated motor hull. This stator winding is usually mathematically modeled by an infinitely fine coil continuously wrapped around the inner side cylinder of the stator. The three-phase AC voltage, which is applied to this winding, serves as the power source of electrical torque that produces the mechanical motion. The other part, which is called the rotor, has mechanical connection only through bearings with the stator and it is crafted from iron plates. The three-phase winding of the rotor is situated in similar rabbets around the surface like the stator winding. In those motors of which the rated power is less then 10kW, these rotor coils are usually built from aluminum bars, which are connected by conducting rings on the front side of the plated body. This type of design is called squirrel-cage motor and it is very typical in most of the applications of the IM. In the following section, this type of the IM is considered.

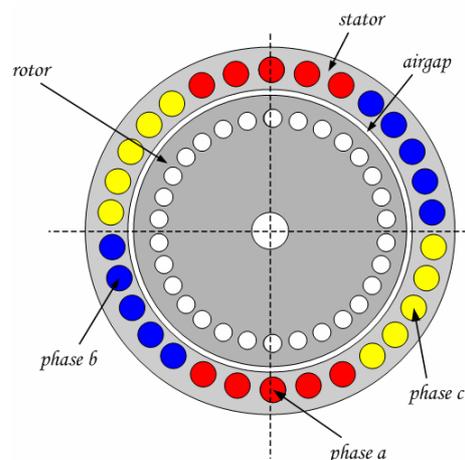


Figure 2.1. Schematics of the induction motor

2. The mathematical model of the induction motor

In *Figure 2.1*, the schematics of the motor can be seen, where an air gap separates the previously mentioned stator and rotor parts. The existence of this air gap is important, because through it, the rotor is connected only magnetically to the stator. Based on this, the two parts can be modeled by the primer and secunder sides of a rotating transformator which is described by complex quantities due to the rotation. Similarly to the properties of a transformator, the magnetic coupling through the air gap provides the flow of energy between the two electrical subsystems, and therefore has a significant role in the overall efficiency of the IM.

The mechanism of the motor is briefly the following: when three-phase AC voltage is applied to the evenly distributed coils around the cylinder of the stator, a rotating magnetic field is built up inside the machine, which produces sinusoidal current inside the rotor winding based on the induction phenomena. This inducted current also produces a rotating magnetic field around the rotor which tries to extinct the effect that brought it to existence. Thus, it tries to line up to the magnetic field of the stator in strength and in orientation as well, but because of the field of the stator rotates and the induction in the rotor lessens when the two magnetic fields approach closer, the field of the rotor can never align with the field of the stator. After the transients an equilibrium state is reached when the two fields rotate with the same speed, which is called synchronous frequency, but the field of the rotor always lags behind the field of the stator. Moreover, the interaction between the two magnetic fields produces a *electromagnetic force* (emf), whose effect is orthogonal to the shaft and forces the rotor to rotate. The speed of the rotor is noted by $n(t)$, which is usually given in angular speed: $\omega(t) = 2\pi \cdot n(t)$, and the synchronous speed of the magnetic field can be described as follows:

$$n_0 = \frac{f_0}{p} = \frac{\text{input frequency}}{\text{number of pole pairs}}, \quad (2.1)$$

which is usually given in angular frequency: $\omega_0 = 2\pi n_0$. Furthermore, the normalized difference of the two speed is the *slip* (s) of the motor:

$$s = \frac{n_0 - n}{n_0}. \quad (2.2)$$

The slip is directly related to the load torque and it describes the equilibrium state of the electromagnetical interaction. $s(t) \in (0,1]$ (in motoric operation range) Based on the direction of the energy flow, the IM can be operated as a motor and a generator as well, although it is barely used as a generator because of efficiency related issues. In this paper, only the motoric operation of the IM is going to be described.

2.2. Modeling conditions

In the next sections, the squirrel-cage type IM is considered whose physical layout parameters are already included in the synchron frequency of the magnetic field and in the mechanical time constant of the machine. Beside this, the following assumptions are made, which are widely common in mathematical investigations of this physical phenomena [32]:

- ◆ The permeability of the iron body of the rotor is assumed to be infinite with linear magnetic properties, so the saturation phenomena do not affect the magnetic coupling.
- ◆ The iron core is assumed to be homogeneous, thus circular parasite currents are not produced inside the core which would lessen the induction. Slotting effects, like deep bar and end effects are also neglected.
- ◆ The spatial distribution of the phase currents in the stator winding is considered to be sinusoidal, which describes two half moons shape density distribution rotating along the cylinder of the stator.
- ◆ The three-phase star connection of the stator coils has symmetric electrical parameters, and the input excitation is also a perfectly symmetric three-phase voltage or current feed.

2.3. General mathematical description

By applying the above mentioned assumptions and neglecting the described loss effects to the description of the motor, the instantaneous spatial distribution of one phase current can be given as *Figure 2.2*. In the following, the mathematical description of the motor is investigated through the space vector

theory introduced in [26] to be able to use the possibilities provided by this approach. If it is assumed that the current of the stator is the input excitation of the machine, than the $F_{sx}(\alpha, t)$ spatial distribution along the stator of the x phase current can be described by the $\mathbf{i}_{sx}^s(t)$ complex vector whose orientation is determined by the direction of the respective phase axis and the current polarity [18]. In the presented case the positive phase current $\mathbf{i}_{sa}^s(t)$ in stator winding phase a creates a sinusoidal current density distribution that leads the winding axis a by 90° , having therefore its maximum in the direction of the imaginary axis.

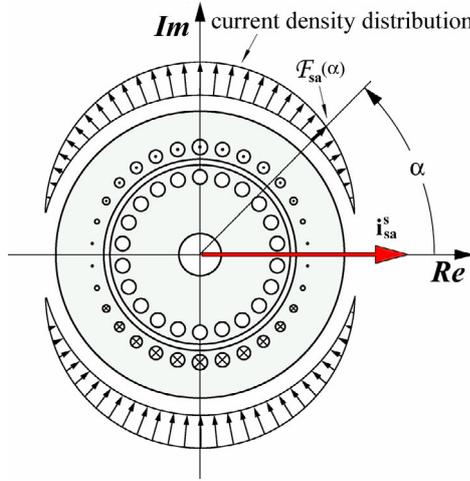


Figure 2.2. Current density distribution of phase a

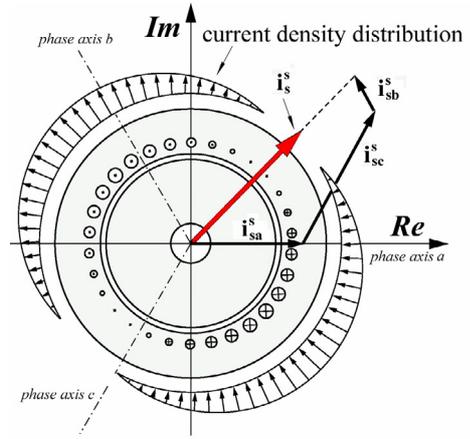


Figure 2.3. Stator side current density distribution

The total *magnetomotive force (mmf.)* inside the motor is obtained as the superposition of the current density distributions of all three phases. This produces again a sinusoidal distribution of the overall stator current density wave indicated by *Figure 2.3*. The $\mathbf{i}_s^s(t)$ space vector, which describes this overall current distribution, can be given by the superposition of the space vectors of each phase as it can be seen in *Figure 2.3*. Furthermore, $\mathbf{i}_s^s(t)$ can be also computed from the positive successive representation of the symmetric phase currents:

$$\mathbf{i}_s^s(t) = \frac{2}{3} \left(\underbrace{\mathbf{a}^0 \cdot \mathbf{i}_{sa}(t)}_{\mathbf{i}_{sa}(t)} + \underbrace{\mathbf{a} \cdot \mathbf{i}_{sb}(t)}_{\mathbf{i}_{sb}(t)} + \underbrace{\mathbf{a}^2 \cdot \mathbf{i}_{sc}(t)}_{\mathbf{i}_{sc}(t)} \right) = \sqrt{2} \cdot \mathbf{i}_{\text{eff}}^s(t) \cdot e^{j\omega_0 t + \frac{\pi}{2} + \varphi_i}, \quad (2.3)$$

$$\mathbf{i}_{sa}(t) = \text{Re}\{\mathbf{i}_s^s(t)\}, \quad \mathbf{i}_{sb}(t) = \text{Re}\{\mathbf{a}^2 \cdot \mathbf{i}_s^s(t)\}, \quad \mathbf{i}_{sc}(t) = \text{Re}\{\mathbf{a} \cdot \mathbf{i}_s^s(t)\}, \quad (2.4)$$

2. The mathematical model of the induction motor

where $\mathbf{a} = e^{j\frac{2\pi}{3}}$ is the complex unity vector that describes the phase lags and the space orientation of the current density distributions to each other along the stator. For completeness, equation (2.4) provides the inverse transformation of (2.3). So $\mathbf{i}_s(t)$, the stator current space vector, represents the sinusoidal spatial distribution of the total *mmf.* wave created inside the machine by the three phase currents that flow from the outside of the machine. The *mmf.* wave has its maximum in an angular position that leads $\mathbf{i}_s(t)$ by 90° as illustrated in *Figure 2.3*. The amplitude of this *mmf.* is proportional to $|\mathbf{i}_s(t)|$. Furthermore, the scaling factor $2/3$ presented in the equation (2.3) reflects the fact that the total density distribution is obtained as the superposition of the density distributions of the three phase windings, while the contribution of only two phase windings, spaced 90° apart, would have the same spatial effect with the phase currents properly adjusted. Therefore, it provides the scaling factor from three-phase description to the two-phase based complex representation. This approach is called energy invariant representation and it is used in the following to construct the model of the IM.

The excitation produced flux density distribution in the air gap is obtained by spatial integration of the current density wave along the cylinder of the stator. Therefore, it is also a sinusoidal wave, and it lags the current density wave by 90° as illustrated in *Figure 2.4*.

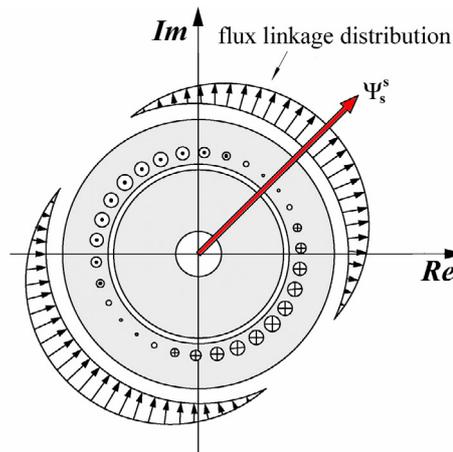


Figure 2.4. Stator side overall flux linkage distribution

It is convenient to choose the flux linkage wave, as a system variable instead of the flux density wave as the former contains added information on the winding

geometry and the number of turns of the coil. By definition, a flux linkage distribution has the same spatial orientation as the pertaining flux density distribution, therefore the stator flux linkage distribution presented in *Figure 2.4* can be described by the space vector $\Psi_s^s(t)$. Based on same reasons, the $\Psi_r^r(t)$ space vector can be also introduced to represent the rotor flux linkage distribution. In the individual stator windings, the rotating flux density wave induces voltages. Since the winding densities are sinusoidal spatial functions, therefore the induced voltages are also sinusoidally distributed in space. The same is true for the resistive voltage drop in the windings. Thus, the overall distributed voltages in all phase windings can be represented by the $\mathbf{u}_s^s(t)$ stator voltage space vector, which is a complex variable and can also be computed from the positive successive representation of the stator phase voltages:

$$\mathbf{u}_s^s(t) = \frac{2}{3} \left(\underbrace{\mathbf{a}^0 \cdot \mathbf{u}_{sa}(t)}_{\mathbf{u}_{sa}(t)} + \underbrace{\mathbf{a} \cdot \mathbf{u}_{sb}(t)}_{\mathbf{u}_{sb}(t)} + \underbrace{\mathbf{a}^2 \cdot \mathbf{u}_{sc}(t)}_{\mathbf{u}_{sc}(t)} \right) = \sqrt{2} \cdot \mathbf{u}_{\text{eff}}^s(t) \cdot e^{j\omega_0 t + \frac{\pi}{2} + \phi_u}, \quad (2.5)$$

$$u_{sa}(t) = \text{Re} \left\{ \mathbf{u}_s^s(t) \right\}, \quad u_{sb}(t) = \text{Re} \left\{ \mathbf{a}^2 \cdot \mathbf{u}_s^s(t) \right\}, \quad u_{sc}(t) = \text{Re} \left\{ \mathbf{a} \cdot \mathbf{u}_s^s(t) \right\}. \quad (2.6)$$

2.3.1. Voltage and flux equations

If we consider the primer side of the complex transformator (*Figure 2.5*) as the stator of the IM, then equation (2.7) describes the connection between the introduced stator side space vectors:

$$\mathbf{u}_s^s(t) = \mathbf{i}_s^s(t) \cdot R_s + \frac{d\Psi_s^s(t)}{dt}, \quad (2.7)$$

where $\mathbf{i}_s^s(t) \cdot R_s$ is the resistive voltage drop and $R_s = R_{sa} = R_{sb} = R_{sc}$ denotes the resistance of the symmetric stator phase windings. The induced voltage vector, represented by the last term of (2.7), is the back *electromagnetic force* (emf.).

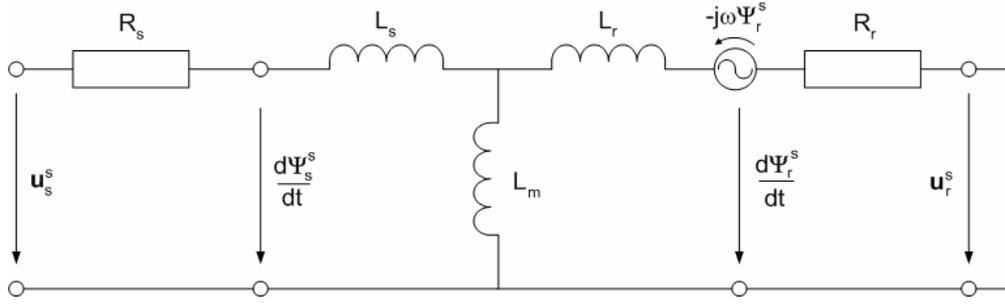


Figure 2.5. Complex transformator that describes the model of the IM

By considering the secunder side of the complex transformator, it can be concluded that the same relationship is true to the rotor side space vectors, but because of the short circuited rotor winding, in this case $\mathbf{u}_r^r(t) = 0, \forall t \in \mathbb{R}$, thus

$$\mathbf{u}_r^r(t) = \mathbf{i}_r^r(t) \cdot \mathbf{R}_r + \frac{d\Psi_r^r(t)}{dt} = \mathbf{0}. \quad (2.8)$$

Equation (2.7) and (2.8) describe the electromagnetic interaction as the connection of first order dynamic subsystems. Because four complex variables: $\mathbf{i}_s^s(t), \mathbf{i}_r^r(t), \Psi_s^s(t), \Psi_r^r(t)$ are presented in these two equations, (2.9) and (2.10) flux equations are needed to complete the relationship between them.

$$\Psi_s^s(t) = \mathbf{i}_s^s(t) \cdot \mathbf{L}_s + \mathbf{i}_r^r(t) \cdot \mathbf{L}_m \cdot e^{j\rho(t)}, \quad (2.9)$$

$$\Psi_r^r(t) = \mathbf{i}_s^s(t) \cdot \mathbf{L}_m \cdot e^{-j\rho(t)} + \mathbf{i}_r^r(t) \cdot \mathbf{L}_r, \quad (2.10)$$

where the presented $\rho(t)$ angle describes the position of the rotor compared to the axis of the stator, while $\mathbf{L}_r = \frac{3}{2}\mathbf{l}_r + \frac{3}{2}\mathbf{l}_f$, $\mathbf{L}_s = \frac{3}{2}\mathbf{l}_s + \frac{3}{2}\mathbf{l}_f$ are the three-phase inductances and $\mathbf{l}_s, \mathbf{l}_r$ are the inductances of a stator and a rotor phase winding, \mathbf{l}_f is the self inductance, and $\mathbf{L}_m = \frac{3}{2}\mathbf{l}_m$ is the mutual inductance between the stator and the rotor [32]. In order to eliminate the $\rho(t)$ angle from the model equations transformation of rotor side vectors into the reference frame of the stator is needed. Thus, by applying the following substitutions: $\mathbf{i}_r^s(t) = \mathbf{i}_r^r(t) \cdot e^{j\rho(t)}$ and $\Psi_r^s(t) = \Psi_r^r(t) \cdot e^{j\rho(t)}$, then

$$\Psi_s^s(t) = \mathbf{i}_s^s(t) \cdot L_s + \mathbf{i}_r^s(t) \cdot L_m \quad (2.11)$$

$$\Psi_r^s(t) = \mathbf{i}_s^s(t) \cdot L_m + \mathbf{i}_r^s(t) \cdot L_r \quad (2.12)$$

equations provides the flux connections in the model, while, because of the consistence of the transformations, (2.7) and (2.8) do not change.

2.3.2. The dynamical motion equation

Based on the concept of energy considerations, the electromagnetic torque of the motor can be derived easily [26]. For the mechanic power $P_{\text{mech}}(t)$ of the system, the following is true:

$$P_{\text{mech}}(t) = \frac{dW_{\text{mech}}(t)}{dt}, \quad (2.13)$$

where the mechanical energy $W_{\text{mech}}(t)$ in case of rotating systems can be given by

$$\frac{dW_{\text{mech}}(t)}{dt} = T_e(t) \cdot \omega(t). \quad (2.14)$$

However, it is also true that the mechanical energy can be presented in the form of

$$W_e(t) = W_{\text{mech}}(t) + W_v(t) + W_{\text{field}}(t), \quad (2.15)$$

where

$$\frac{dW_e(t)}{dt} = \frac{3}{2} \text{Re} \left\{ \mathbf{u}_s^s \cdot \mathbf{i}_s^{s*} + \mathbf{u}_r^s \cdot \mathbf{i}_r^{s*} \right\}, \text{ is the input electric power,}$$

$$\frac{dW_v(t)}{dt} = \frac{3}{2} \text{Re} \left\{ R_s |\mathbf{i}_s^s|^2 + R_r |\mathbf{i}_r^s|^2 \right\}, \text{ is the resistive power loss,}$$

$$\frac{dW_{\text{field}}(t)}{dt} = \frac{3}{2} \text{Re} \left\{ \frac{d\Psi_s^s}{dt} \mathbf{i}_s^{s*} + \frac{d\Psi_r^s}{dt} \mathbf{i}_r^{s*} \right\}, \text{ is the air gap power.}$$

Here, the complex conjugation was noted by $*$. By substituting the above equations into (2.14), it can be concluded that:

$$P_{\text{mech}} = T_e \cdot \omega = \frac{3}{2} \frac{L_m}{L_r} \omega \Psi_r^s(t) \times \mathbf{i}_s^s(t). \quad (2.16)$$

From (2.16) the electromagnetic torque of the motor can be expressed in case of a p polyphase machine:

$$T_e(t) = \frac{3}{2} p \frac{L_m}{L_r} \Psi_r^s(t) \times \mathbf{i}_s^s(t). \quad (2.17)$$

It must be noted that (2.17) can also be given as a product of any two of the $\mathbf{i}_s^s(t)$, $\mathbf{i}_r^r(t)$, $\Psi_s^s(t)$, $\Psi_r^r(t)$ state variables by appropriate scaling. Furthermore, for the mechanical subsystem of the motor, the dynamical motion equation is as follows:

$$\tau_{\text{mech}} \frac{d\omega(t)}{dt} = T_e - T_{\text{load}} - F\omega(t), \quad (2.18)$$

where $\tau_{\text{mech}} = J/p$ is the mechanical time constant, J is the moment of inertia, T_{load} is the load torque, and $F\omega(t)$ is the energy loss due to friction. Based on (2.18), the differential equation, describing the change of ω , the angular speed of the rotor, can be derived as follows:

$$\frac{d\omega(t)}{dt} = \underbrace{\frac{3p^2 L_m}{2J L_r}}_{1/\tau_e} \cdot (\Psi_r^s(t) \times \mathbf{i}_s^s(t)) - \frac{T_{\text{load}}(t) + F\omega(t)}{\tau_{\text{mech}}}. \quad (2.19)$$

2.3.3. Approximation of rotor resistance variation

Because the parameter uncertainties have great impact on the overall system dynamics, the most significant rotor resistance variation has to be modeled somehow for an accurate description of the system. However, it is not rewarding to give very detailed description of this change, because the value of R_r is unique for every IM and its dynamic properties strongly depend on the layout and the built-in materials of the motor and on other uncertain parameters [40]. Thus, only an approximation of the real description is needed to model the most important dynamics of the uncertain parameter variations. It is known that the resistance change to heat of a given aluminum body with linear heating characteristics is as follows:

$$\frac{R_r(t) - R_0}{T(t) - T_0} = \frac{R_0}{245 + T_0}, \quad (2.20)$$

where R_0 is the rotor resistance at T_0 normal temperature, and 245 is the specific constant of the aluminum. Furthermore, the heat produced by the electrical energy can be given by (2.21).

$$Q(t) = 0.86 \cdot \int_{t_0}^t \left(i_{\text{eff}}^r(t) \right)^2 \cdot R_r(t) dt. \quad (2.21)$$

By considering the heating properties of materials:

$$Q(t) = m \cdot c \cdot (T(t) - T_0), \quad (2.22)$$

is also true, where c is the specific heat, m is the weight of the aluminum bars. Based on equations (2.20), (2.21), and (2.22) it can be concluded that

$$\frac{dR_r(t)}{dt} = \frac{0.86 \cdot R_0}{\underbrace{(245 + T_0) \cdot m \cdot c}_{R_{T0}}} \cdot \left(i_{\text{eff}}^r(t) \right)^2 \cdot R_r(t) - K_k (R_r(t) - R_0), \quad (2.23)$$

if it is assumed, that the heat dissipation to the environment depends only on the energy leaking caused by convection, which is linear and it is described by the rate constant K_k .

2.4. Model representation in rotating reference frame

The representation of the motor model in a rotating reference frame, introduced by Blaschke, Hasse, and Leonard more than 20 years ago, has completely changed the field of controller synthesis for asynchronous motors [41]. They formed the basic idea of the space vector based direct torque control, which has been recently the most widely used method for IM drives. This basic idea is the following: introduce such a new coordinate system, which rotates with ω_k angular speed against the previously used stator fixed reference frame.

On this way, at any time instant, if the angle between the real stator axis and the real axis of the new reference frame is ρ_k , then the flux space vectors in this representation (see *Figure 2.6*) can be given in the following form:

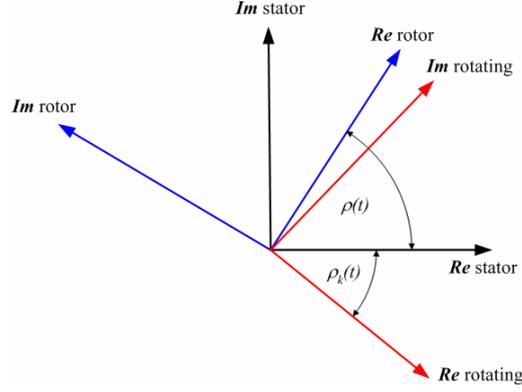


Figure 2.6. Rotating two-phase coordinate system

$$\frac{d\Psi_s^s(t)}{dt} = \frac{d(\Psi_s^k(t) \cdot e^{-j\rho_k(t)})}{dt} = \frac{d\Psi_s^k(t)}{dt} \cdot e^{-j\rho_k(t)} + j \frac{d\rho_k(t)}{dt} \cdot \Psi_s^k(t) \cdot e^{-j\rho_k(t)} \quad (2.24)$$

$$\begin{aligned} \frac{d\Psi_r^r(t)}{dt} &= \frac{d(\Psi_r^k(t) \cdot e^{j(\rho_k(t) - \rho(t))})}{dt} = \\ &= \frac{d\Psi_r^k(t)}{dt} \cdot e^{j(\rho_k(t) - \rho(t))} + j \left(\frac{d\rho_k(t)}{dt} - \frac{d\rho(t)}{dt} \right) \cdot \Psi_r^k(t) \cdot e^{j(\rho_k(t) - \rho(t))} \end{aligned} \quad (2.25)$$

where the vectors presented with index k are the transformed vectors of the rotating reference frame, $\frac{d\rho_k(t)}{dt} = \omega_k(t)$ is the angular speed of the reference rotation, and $\frac{d\rho(t)}{dt} = \omega(t)$ is the angular speed of the rotor. By applying the reference transformation to all of the space vectors, equation (2.7) and (2.8) are reformed as follows:

$$\begin{aligned} \mathbf{u}_s^k(t) &= \mathbf{u}_s^s(t) \cdot e^{-j\rho_k(t)} = \mathbf{i}_s^s(t) \cdot e^{-j\rho_k(t)} \cdot \mathbf{R}_s + \frac{d\Psi_s^s(t)}{dt} \cdot e^{-j\rho_k(t)} = \\ &= \mathbf{i}_s^k(t) \cdot \mathbf{R}_s + \frac{d\Psi_s^k(t)}{dt} + j\omega_k \Psi_s^k(t) . \end{aligned} \quad (2.26)$$

$$\begin{aligned} 0 &= \mathbf{u}_r^k(t) = \mathbf{u}_r^r(t) \cdot e^{-j(\rho_k(t) - \rho(t))} = \mathbf{i}_r^r(t) \cdot e^{-j(\rho_k(t) - \rho(t))} \cdot \mathbf{R}_r + \frac{d\Psi_r^r(t)}{dt} \cdot e^{-j(\rho_k(t) - \rho(t))} \\ &= \mathbf{i}_r^k(t) \cdot \mathbf{R}_r + \frac{d\Psi_r^k(t)}{dt} + j(\omega_k - \omega) \Psi_r^k(t) . \end{aligned} \quad (2.27)$$

By the consistency of the transformation, (2.11) and (2.12) preserve their original forms.

2.4.1. General reference orientation

Let equations (2.11), (2.12), (2.26), and (2.27) are considered in a general reference frame rotating with the angular speed of ω_k . In this case, any two of the $\mathbf{i}_s^k(t)$, $\mathbf{i}_r^k(t)$, $\Psi_s^k(t)$, $\Psi_r^k(t)$ space vectors can be chosen to state variable during the formulation of the model, but for decoupled control of the flux and speed, it is worth choosing $\mathbf{i}_s^k(t)$ and $\Psi_r^k(t)$ for the states of the system. On this way, from (2.11) and (2.12):

$$\Psi_s^k(t) = \frac{L_s L_r - L_m^2}{L_r} \cdot \mathbf{i}_s^k(t) + \frac{L_m}{L_r} \cdot \Psi_r^k(t), \quad (2.28)$$

can be formulated with

$$\frac{d\Psi_r^k(t)}{dt} = \frac{L_m R_r}{L_r} \mathbf{i}_s^k(t) - \left(\frac{R_r}{L_r} + j(\omega_k - \omega) \right) \Psi_r^k(t), \quad (2.29)$$

from (2.27) and

$$\begin{aligned} \frac{L_s L_r - L_m^2}{L_r} \cdot \frac{d\mathbf{i}_s^k(t)}{dt} = \mathbf{u}_s^k(t) - \left[\left(\frac{L_m^2}{L_r^2} R_r + R_s \right) + j\omega_k \cdot \left(\frac{L_s L_r - L_m^2}{L_r} \right) \right] \cdot \mathbf{i}_s^k(t) \\ + \left(\frac{L_m R_r}{L_r^2} - j\omega \cdot \frac{L_m}{L_r} \right) \cdot \Psi_r^k(t), \end{aligned} \quad (2.30)$$

from (2.26). Also, the exact state equation can be derived from (2.29) and (2.30). If we decompose our complex variables to their real (d) and imaginary (q) parts, and suppose that the rotor resistance varies, then the following input-affine state space equations system, which is nonlinear in ω , ω_k , and R_r , describes the model.

$$\frac{d}{dt} \begin{bmatrix} \Psi_{rd}^k(t) \\ \Psi_{rq}^k(t) \\ i_{sd}^k(t) \\ i_{sq}^k(t) \end{bmatrix} = \underbrace{\begin{bmatrix} -\frac{1}{\tau_r(t)} & (\omega_k - \omega(t)) & \frac{L_m}{\tau_r(t)} & 0 \\ -(\omega_k - \omega(t)) & -\frac{1}{\tau_r(t)} & 0 & \frac{L_m}{\tau_r(t)} \\ \frac{\tau}{\sigma \cdot \tau_r(t)} & \omega(t) \cdot \frac{\tau}{\sigma} & -\frac{(\lambda \tau_r(t) + \tau_s)}{\sigma} & \omega_k \\ -\omega(t) \cdot \frac{\tau}{\sigma} & \frac{\tau}{\sigma \cdot \tau_r(t)} & -\omega_k & -\frac{(\lambda \tau_r(t) + \tau_s)}{\sigma} \end{bmatrix}}_{A(\omega, \omega_k, R_r)} \cdot \begin{bmatrix} \Psi_{rd}^k(t) \\ \Psi_{rq}^k(t) \\ i_{sd}^k(t) \\ i_{sq}^k(t) \end{bmatrix} + \underbrace{\begin{bmatrix} 0 & 0 \\ 0 & 0 \\ \frac{1}{L_s \sigma} & 0 \\ 0 & \frac{1}{L_s \sigma} \end{bmatrix}}_B \cdot \begin{bmatrix} u_{sd}^k(t) \\ u_{sq}^k(t) \end{bmatrix} \quad (2.31)$$

where $\sigma = 1 - (L_m)^2 / (L_s \cdot L_r)$, $\lambda = (L_m)^2 / L_s$, $\tau = L_m / (L_s \cdot L_r)$, $\tau_r(t) = L_r / R_r(t)$, $\tau_s = L_s / R_s$. Besides (2.31) the

$$\frac{d\omega(t)}{dt} = \frac{\Psi_{rd}^k(t) \cdot i_{sq}^k(t) - \Psi_{rq}^k(t) \cdot i_{sd}^k(t)}{\tau_e} - \frac{T_{load}(t) + F\omega(t)}{\tau_{mech}}, \quad (2.32)$$

motion and the

$$\frac{dR_r(t)}{dt} = R_{T0} \cdot \left(\left(\frac{\Psi_{rd}^k(t) - L_m i_{sd}^k(t)}{L_r} \right)^2 + \left(\frac{\Psi_{rq}^k(t) - L_m i_{sq}^k(t)}{L_r} \right)^2 \right) \cdot R(t) - K_k (R_r(t) - R_0), \quad (2.33)$$

rotor resistance nonlinear differential equations complete the description of the system by defining the variance of the parameters of the state matrix $\mathbf{A}(\omega, \omega_k, R_r)$, from which ω and R_r are representation independent parameters of the (2.31) motor model, while ω_k defines the representation of the model. If $\omega_k = 0$, then the real part of the state variables is indexed with α and the imaginary part is with β .

2.4.2. Rotor flux oriented representation

Nowadays, there are several representations of the motor model used in the field of IM drives, based on the value chosen for ω_k . From these possible representations, only the field oriented approach is considered in the following section, because as it is going to be shown, this orientation makes possible the design of an efficient controller with ease. By this approach, a new reference frame is introduced, where the real axis is bounded to the rotating $\Psi_r^k(t)$ vector, thus $\omega_k = \omega_{\text{flux}}$. In this case, $\Psi_r^k(t)$ has no imaginary component, which fact implies the reducement of the state variables by letting out $\Psi_{rq}^k(t)$ from the model. Moreover, because at steady state all of the stator side electrical quantities (*Figure 2.7*) in the stator fixed reference frame rotate with the same synchronous frequency of the flux fields, these field-orientated variables are constants. So in this way, $\omega_{\text{flux}} = \omega_0$, and (2.31) is transformed into the following form:

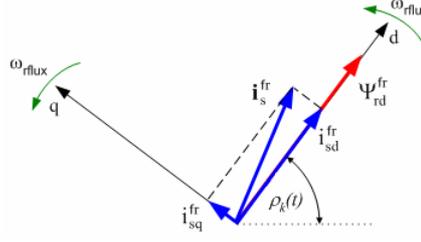


Figure 2.7. Rotor field orientation in two phase rotating reference frame

$$\frac{d}{dt} \begin{bmatrix} \Psi_{rd}^{fr}(t) \\ i_{sd}^{fr}(t) \\ i_{sq}^{fr}(t) \end{bmatrix} = \begin{bmatrix} -\frac{1}{\tau_r(t)} & \frac{L_m}{\tau_r(t)} & 0 \\ \tau & -(\lambda\tau_r(t) + \tau_s) & \omega_{rflux} \\ -\omega(t) \cdot \frac{\tau}{\sigma} & -\omega_{rflux} & -\frac{(\lambda\tau_r(t) + \tau_s)}{\sigma} \end{bmatrix} \cdot \begin{bmatrix} \Psi_{rd}^{fr}(t) \\ i_{sd}^{fr}(t) \\ i_{sq}^{fr}(t) \end{bmatrix} + \begin{bmatrix} 0 & 0 \\ \frac{1}{L_s\sigma} & 0 \\ 0 & \frac{1}{L_s\sigma} \end{bmatrix} \cdot \begin{bmatrix} u_{sd}^{fr}(t) \\ u_{sq}^{fr}(t) \end{bmatrix} \quad (2.34)$$

Because $\Psi_{rq}^{fr}(t) = 0$, thus from (2.31), it involves that:

$$\omega_{rflux}(t) = \omega(t) + \frac{L_m}{\tau_r(t)} \cdot \frac{\dot{i}_{sq}^{fr}}{\Psi_{rd}^{fr}} \quad (2.35)$$

Since (2.35) ensures the correct orientation, the form of (2.19) must be also modified.

$$\frac{d\omega(t)}{dt} = \frac{\Psi_{rd}^{fr}(t) \cdot \dot{i}_{sq}^{fr}(t)}{\tau_e} - \frac{T_{load}(t) + F\omega(t)}{\tau_{mech}} \quad (2.36)$$

In this way, the problem is reformulated brilliantly, because $\Psi_{rd}^{fr}(t)$ can be directly controlled by $i_{sd}^{fr}(t)$, while $\omega(t)$ is controlled with $i_{sq}^{fr}(t)$, without any cross-effect. This idea, which provides the decoupling of the control, is going to be used in the controller to be designed.

2.5. The linear parameter variant motor model

Today, because the LTI system theory is well worked-out and supports several efficient and robust control methods, great portion of the nonlinear control problems are solved, by attributing them to a linear time invariant (LTI) form, even in the case of highly sophisticated and reliability-needed areas, such as the aviation [28]. In the simplest cases, the transformation to LTI form is usually

done through the linearization of the system model, although the global linearization for most of the nonlinear models produces great loss in the description of the real dynamics of the system. Thus, for those cases where the system has strong dynamical properties, the easiest way to solve this dilemma is to locally linearize our system point by point on a parameter space to produce locally LTI systems. If we choose the linearization points to be infinitely close to each other, then the system description of the model can be imagined as a locally LTI system moving in a n dimensional system space. These type of systems are called LPV systems, where the model is linear with respect to the states, inputs, and outputs, and the state space matrices $\mathbf{A}(\cdot)\dots\mathbf{D}(\cdot)$ are dependent on a $\mathbf{p}(t)$ n dimensional bounded parameter vector. If we suppose that the parameters are smooth and changing slow enough, then it is possible to use the LTI control theories to design in every point of the parameter space an LTI controller for the locally linear LPV system. This method is called *gain scheduling* and it was introduced by Packard [30] and Apkarian & Gahinet [2] based on the *small gain theory*. Recently, more and more of such nonlinear control problems, which can be transformed to an LPV form, have been solved by the help of this method [7, 16, 22, 28, 31]. However, this approach does not always give better results than the common ad-hoc methods.

Because the model of the IM is highly nonlinear with strong dynamical properties, this method has to be considered at the first place. Thus, let the field oriented system description of the IM transformed into an LPV form based on the previously mentioned idea. The result of this transformation is a parameter-affine LPV model dependent on the ω , ω_{rflux} , and \mathbf{R}_r parameters:

$$\dot{\mathbf{x}} = \mathbf{A}(\mathbf{p}(t)) \cdot \mathbf{x} + \mathbf{B} \cdot \mathbf{u} + \mathbf{B} \cdot \varepsilon_{\text{inv}}(t) \quad (2.37)$$

$$\mathbf{y} = \mathbf{C} \cdot \mathbf{x} + \mathbf{C} \cdot \varepsilon_{\text{mes}}(t) \quad (2.38)$$

where

$$\mathbf{x} = \begin{bmatrix} \Psi_{\text{rd}}^{\text{fr}}(t) \\ i_{\text{sd}}^{\text{fr}}(t) \\ i_{\text{sq}}^{\text{fr}}(t) \end{bmatrix}, \quad \mathbf{y} = \begin{bmatrix} i_{\text{sd}}^{\text{fr}}(t) \\ i_{\text{sq}}^{\text{fr}}(t) \end{bmatrix}, \quad \mathbf{p}(t): \mathbb{R}_+ \rightarrow \mathbb{R}_+^3 = \begin{bmatrix} p_1(t) \\ p_2(t) \\ p_3(t) \end{bmatrix} = \begin{bmatrix} \omega(t) \\ \mathbf{R}_r(t) \\ \omega_{\text{rflux}}(t) \end{bmatrix}$$

$$\mathbf{u} = \begin{bmatrix} \mathbf{u}_{sd}^{fr}(t) \\ \mathbf{u}_{sq}^{fr}(t) \end{bmatrix} = \frac{1}{3} \underbrace{\begin{bmatrix} 3 \cos \rho(t) & \sqrt{3} \sin \rho(t) & 2\sqrt{3} \sin \rho(t) \\ -3 \sin \rho(t) & \sqrt{3} \cos \rho(t) & 2\sqrt{3} \cos \rho(t) \end{bmatrix}}_{\text{Park \& Clark transformation}} \cdot \begin{bmatrix} \mathbf{u}_{sa}(t) \\ \mathbf{u}_{sb}(t) \\ \mathbf{u}_{sc}(t) \end{bmatrix}$$

$$\mathbf{A}(\mathbf{p}(t)) = \begin{bmatrix} -\frac{p_2(t)}{L_r} & \frac{L_m p_2(t)}{L_r} & 0 \\ \frac{\tau \cdot p_2(t)}{L_r \cdot \sigma} & -\left(\frac{\lambda \cdot p_2(t)}{L_r \cdot \sigma} + \frac{\tau_s}{\sigma}\right) & p_3(t) \\ -p_1(t) \cdot \frac{\tau}{\sigma} & -p_3(t) & -\left(\frac{\lambda \cdot p_2(t)}{L_r \cdot \sigma} + \frac{\tau_s}{\sigma}\right) \end{bmatrix},$$

$$\mathbf{B} = \begin{bmatrix} 0 & 0 \\ \frac{1}{L_s \sigma} & 0 \\ 0 & \frac{1}{L_s \sigma} \end{bmatrix}, \quad \mathbf{C} = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}.$$

It can be seen that the state matrix can be given in the form of $\mathbf{A}(\mathbf{p}(t)) = \mathbf{A}_0 + \mathbf{A}_1(p_1(t)) + \mathbf{A}_2(p_2(t)) + \mathbf{A}_3(p_3(t))$, which property is called *parameter affinity*. Furthermore, the input affinity of the original model is also preserved in the LPV state equation system. In this way, the strongly nonlinear (2.34) differential equation system can be handled as a purely LTI description. New elements are introduced to the model as well, like $\varepsilon_{inv}(t)$, the *pulse width modulation (PWM)* resulted error of the inverter which produces the input voltage feed of the motor and $\varepsilon_{mes}(t)$, the measurement noise of the sensors. While $\varepsilon_{inv}(t)$ can be modeled as a band limited white noise effect, the $\varepsilon_{mes}(t)$ is perfectly described by a high frequency dominant filtered white noise.

The transformation of the three phase quantities to a two-phase coordinate system and the field orientation is calculated through the Clark and Park transformations, whose details will be explained in the later sections.

2.6. General dynamic properties of the model

The dynamical analysis of the motor model needs large spectrum of investigations which should cover another full paper in this topic. However, these investigations must be done before the true designing of the controller begins. For this reason, the full nonlinear analysis of the model has been completed in [38], but here only the main results and conclusions are mentioned.

From equation (2.31), it is clear that the model is built up from first order subsystems. Thus, by considering each of these subsystems separately, the state variables tend to their equilibrium point without any overshoot and fluctuation, as well as their transition functions have purely exponential decrease. However, the inner cross effects caused by the multiplications between the state variables introduce high nonlinearities into the system, which cause oscillations in the dynamic behavior of the model. Moreover, the whole system dynamics can be separated into two operation interval: such as the startup and the steady state, in which the behavior of the model is different in terms of stability. During startup, the system shows instability till the electromagnetic torque (T_e) has not reached a specific value called the pullout torque. This phenomenon exists, because a magnetic field has to be built up inside the motor to provide an adequate energy flow between the rotor and the stator subsystems. Thus, a high current impulse has to be provided to the motor, which quickly builds up the needed magnetic field and has enough energy to overcome the moment of inertia of the rotor. The difference in dynamic behavior of the two intervals can be seen in *Figure 2.8* [18], where the interaction between the normalized T_e and ω is plotted. It is clearly shown that during the direct startup, the function of torque / speed has large oscillations at first, which are dumped slowly till the system asymptotically reaches the equilibrium point $(\omega_{nom}, T_{e_{nom}})$ as in the case of the steady state change. This plotted function also describes the step response of the transition function between the torque and speed.

It is another important fact that the relative degree between any inputs and any outputs of the system is not greater than 3. Based on this reason, the I/O

2. The mathematical model of the induction motor

linearization of the model can be given easily in most of the cases, but for the load torque a very complicated virtual input and output functions have to be chosen for the linearization, which cause the accumulation of numerical error in practice. The effect of this property is analyzed in later sections. The zero dynamics of the system in contrast of the previous facts is stable for ω that makes possible to hold T_{load} with the motor at 0 speed and control the full 4/4 operation range of the IM drive.

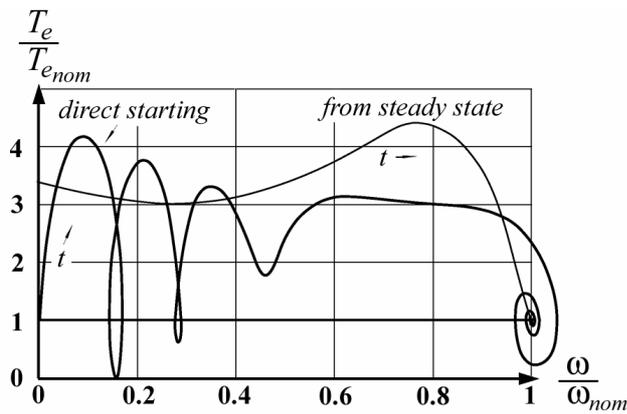


Figure 2.8. Torque / frequency transients of the motor in each dynamic interval

3. Existing speed sensorless techniques

Today, several methods are known for the speed sensorless control of the induction drives, but each of these solutions has its own disadvantage beside its strong sides. In the past, there were arduous attempts to extract the speed or the position signal and to estimate the strength and the orientation of the flux field of an induction machine [18]. However, the first attempts have been restricted to techniques which are only valid in the steady state. These can be used in low-cost drive applications, not requiring high dynamic performance. But as the raging competition of the drive manufacturing companies has shaken this field up, a lot of new methods have been introduced which are applicable for high-performance applications in vector and direct torque controlled drives.

It is a common feature of the sensorless techniques that they depend on machine parameters, like the temperature of the motor, rotor resistance variations, frequency, etc. To compensate the parameter variations, various parameter adaptation schemes have been proposed in the literature. Unfortunately, conventional techniques are not suitable to achieve stable, very low speed operation in speed sensorless drives because the rotor speed directly based on the applied torque and the estimation of this fine balance is very problematic in this case, due to the parameter mismatch and noise. This task can only be solved by very difficult models that utilize various effects like the rotor slot harmonics, saliency, etc.[18]. This is the reason why beside the existing techniques there is a great search for new methods, like the controller whose implementation is the topic of this paper, which makes possible the control with a simple model that has smaller computational load than the recently used solutions in high performance drives. In the following parts, the most popular methods will be discussed briefly to have an outlook on the field of applied control of induction drives.

3.1. Constant Volts per Hertz control

One way of dealing with the complex and nonlinear (NL) dynamics of the IM in adjustable speed drives is avoiding excitation at the eigenfrequency of the model. To this aim, a gradient limiter reduces the bandwidth of the stator voltage

frequency command as shown in *Figure 3.1*. The band limited stator frequency command signal then generates the stator voltage reference magnitude $|\mathbf{u}_{s\text{ref}}^s|$ while its integral determines the phase angle $\angle(\mathbf{u}_{s\text{ref}}^s)$.

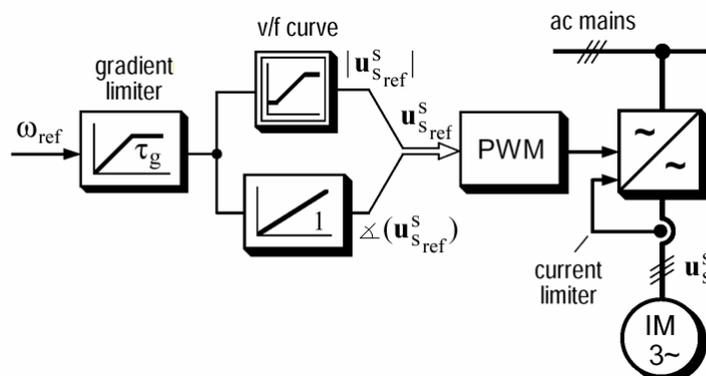


Figure 3.1. Constant volt per hertz control

By examining the real voltage / rotor speed (given in frequency) characteristic of the motor presented in *Figure 3.2*, it can be clearly seen that during a large interval of the operation range, the relationship between ω and the applied voltage is linear in the case of constant load [20].

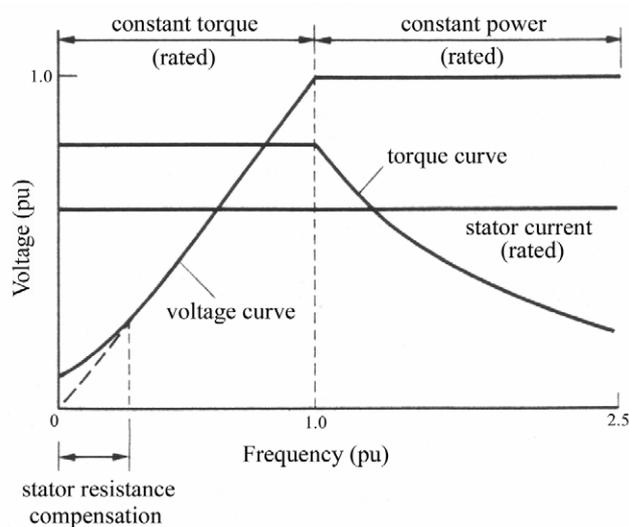


Figure 3.2. Voltage / frequency characteristic of the IM

It is also possible to effect the *mmf*. of the motor by changing the synchronous frequency of the voltage feed through the PWM modulation of the inverter. Moreover, the T_e / ω characteristics of the motor can be directly shifted along the frequency axis with the changing of the synchronous frequency, of course this is

only true theoretically, because the layout effects deform this relationship if the input frequency is too large or too low [5]. This is shown by *Figure 3.3*.

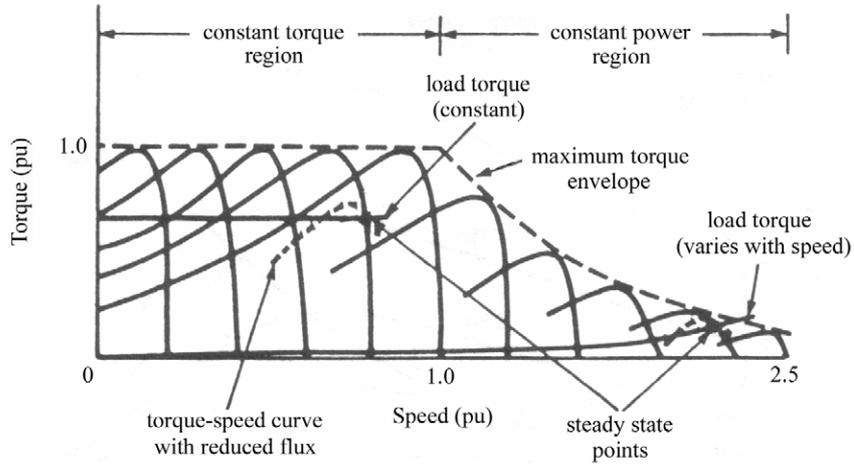


Figure 3.3. Torque / speed characteristics of the IM at different synchronous frequencies

The above mentioned properties can be derived mathematically as well. From (2.7), equation (3.1) is derived, by neglecting the resistive voltage drop $R_s \mathbf{i}_s^s$ and, in view of band limited excitation, by assuming steady state operation with $d\Psi_s^s / dt \approx \mathbf{0}$. This yields

$$\mathbf{u}_s^s(t) = j\omega_s(t) \cdot \Psi_s^s(t), \quad (3.1)$$

or $|\mathbf{u}_{s_{ref}}^s| / \omega_s = \text{const.}$ (or $v/f = \text{const.}$) when the stator flux is maintained at its nominal value in the base speed range. Field weakening is obtained by maintaining $|\mathbf{u}_{s_{ref}}^s| = u_{s_{max}} = \text{const.}$, while increasing the stator frequency beyond its nominal value. At very slow stator frequency, a preset minimum value of the stator voltage is programmed to account for the resistive voltage drop, which can be seen in *Figures 3.1 and 3.2*.

The signals $|\mathbf{u}_{s_{ref}}^s|$ and $\angle(\mathbf{u}_{s_{ref}}^s)$ are obtained to constitute the reference vector $\mathbf{u}_{s_{ref}}^s$ of the stator voltage, which in turn controls a PWM modulator to generate the switching sequence of the inverter.

Since v/f drives operate purely as feed forward systems, the produced drive is absolutely robust, however, the mechanical speed ω differs from the reference

speed ω_{ref} when the machine is loaded. The difference is the slip frequency equal to the electrical frequency of the rotor currents. The maximum speed error is determined by the nominal slip, which is most commonly 3-5% for low power machines and less for high power. To compensate this, a load current dependent scheme can be employed to reduce the speed error [1].

If the system equations are derived in the stator fixed reference frame, letting $\omega_k = 0$. The result is

$$\frac{d\Psi_s^s(t)}{dt} = \mathbf{u}_s^s(t) - \frac{1}{\sigma\tau_s} \left(\Psi_s^s(t) - \frac{L_m}{L_r} \Psi_r^s(t) \right), \quad (3.2)$$

$$\sigma\tau_r \cdot \frac{d\Psi_r^s(t)}{dt} + \Psi_r^s(t) = j\omega(t) \cdot \sigma\tau_r \Psi_r^s(t) + \frac{L_m}{L_s} \Psi_s^s(t). \quad (3.3)$$

From the (2.11) and (2.12) flux equations it can be clearly seen that

$$\mathbf{i}_s^s(t) = \frac{1}{\sigma L_s} \left(\Psi_s^s(t) - \frac{L_m}{L_r} \Psi_r^s(t) \right), \quad (3.4)$$

and the stator flux vector is directly generated by the integral of $\mathbf{u}_s^s - \mathbf{R}_s \mathbf{i}_s^s$. The key quantity of this control concept is the active stator current i_{sp} which is

$$i_{\text{sp}}(t) = \mathbf{u}_{\text{sref}}^s(t) \cdot \frac{|\mathbf{i}_s^s(t)|}{|\mathbf{u}_{\text{sref}}^s(t)|} = i_{\text{s}\alpha}^s(t) \cos(\alpha) + i_{\text{s}\beta}^s(t) \sin(\alpha), \quad \text{where } \alpha = \angle(\mathbf{u}_{\text{sref}}^s(t)). \quad (3.5)$$

It can be derived that i_{sp} is proportional to the torque, thus if at the $\omega_{r_{\text{nominal}}}$ nominal slip the magnitude of the active \mathbf{i}_s^s is known, which is $i_{\text{sp}_{\text{nominal}}}$ then the slip to be compensated can be estimated as:

$$\omega_r(t) \approx \left(\frac{\omega_{r_{\text{nominal}}}}{i_{\text{sp}_{\text{nominal}}}} \right) \cdot i_{\text{sp}}(t). \quad (3.6)$$

Based on this, the control structure demonstrated on *Figure 3.4*. provides favorable dynamic behavior by making the trajectory of the stator flux independent from of the stator current and the load.

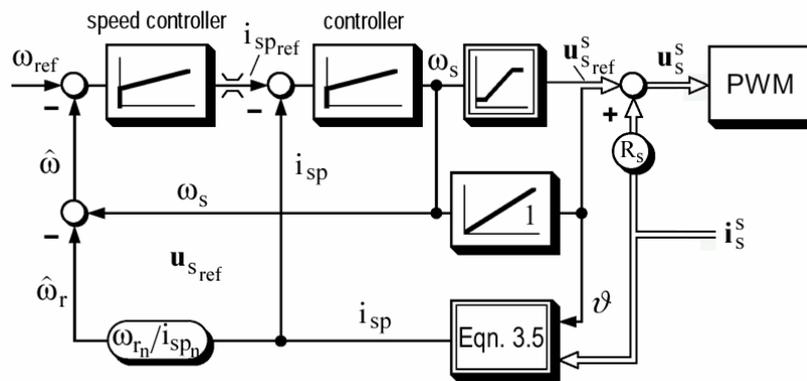


Figure 3.4. v/f controller with slip compensation

Constant v/f control ensures robustness at expense of reduced dynamic performance which is adequate for applications like pump and fan drives. Their particular attraction is their extremely simple control structure which favors an implementation by a few highly integrated electronic components. These cost saving methods are specially important for applications at power below 5kW, but at high power, when the power components themselves dominate the cost, the implementation of more sophisticated control methods becomes available.

3.2. Machine models

In practice, machine models are usually used to estimate the motor shaft speed to avoid the need of encoders or tacho generators and, in high performance drives with field oriented control, to identify the time varying angular position of the flux vector. In addition to this, the magnitude of the flux vector is estimated as well to ensure the effective operation of the drive.

Different machine models are employed for this purpose, and they are implemented in different manner to compute the unknowns depending on the problem at hand. Sometimes, by only the measured signals based solution of the differential equations of the model, with implemented real time numerical computation on a microprocessor, provides adequate performance, but in case of high expectations in optimality, these models are somehow enhanced by creating mathematical methods, called observers, which can improve the quality of the estimation of the desired values greatly. By applying these methods, the common

non speed sensorless control approaches become available to provide speed sensorless operation of the drive.

3.2.1. Direct reference models

Direct reference models are derived from the differential equations of the motor model either in stator fixed or field oriented representation. The accuracy of these models depends on the degree of coincidence which can be obtained between the model and the modeled system. Coincidence should prevail both in terms of structures and parameters. While existing analysis methods permit establishing of appropriate model structures in case of the IM, the parameters of such model are not always in good agreement with the corresponding machine data. Parameters may significantly change with temperature, or with the operating point of the machine. On the other hand, the sensitivity of a model to parameter mismatch may differ, depending on the respective parameter, and the particular variable estimated by the model, but it is always true that the slight computational load and easy structure of the reference machine models can guarantee only limited achievable performance.

Rotor models are derived from the relationship of the rotor winding either in stator (2.12) or in field coordinates (2.29). By example, in the stator fixed representation, the following equation gives possibility to compute the rotor flux:

$$\tau_r \frac{d\Psi_r^s(t)}{dt} + \Psi_r^s(t) = j\omega(t) \cdot \tau_r \Psi_r^s(t) + L_m \mathbf{i}_s^s(t). \quad (3.7)$$

If the previous differential equation is numerically solved, for example by the Euler method, then the desired value of Ψ_r^s can be obtained to provide the rotor flux orientation by computing the angle of the rotor flux trough

$$\rho(t) = \text{atan} \frac{\Psi_{r\beta}^s(t)}{\Psi_{r\alpha}^s(t)} + \begin{cases} 0, & \text{if } \Psi_{r\alpha}^s(t) \geq 0 \\ \pi, & \text{if } \Psi_{r\alpha}^s(t) < 0 \text{ and } \Psi_{r\beta}^s(t) > 0, \\ -\pi, & \text{if } \Psi_{r\alpha}^s(t) < 0 \text{ and } \Psi_{r\beta}^s(t) < 0 \end{cases} \quad (3.8)$$

and the feedback signal of the field strength with

$$|\Psi_s^s(t)| = \sqrt{(\Psi_{r\alpha}^s(t))^2 + (\Psi_{r\beta}^s(t))^2}. \quad (3.9)$$

Another important example is the stator approach, where from the (2.7) stator voltage equation (3.10) can be obtained.

$$\Psi_s^s(t) = \int_0^t (\mathbf{u}_s^s(v) - \mathbf{R}_s \mathbf{i}_s^s(v)) dv \quad (3.10)$$

Equations (2.11) and (2.12) can be used to determine the rotor flux linkage vector Ψ_r^s and the leakage vector Ψ_σ^s from which the penetrating field angle, and the magnitude of the flux linkage can be obtained:

$$\Psi_r^s(t) = \frac{L_r}{L_m} \left(\int_0^t (\mathbf{u}_s^s(v) - \mathbf{R}_s \mathbf{i}_s^s(v)) dv - \sigma L_s \mathbf{i}_s^s(t) \right) = \frac{L_r}{L_m} (\Psi_s^s(t) - \Psi_\sigma^s(t)). \quad (3.11)$$

The stator model is difficult to apply in practice, since the error in the required signals \mathbf{u}_s^s and \mathbf{i}_s^s , and the offset and drift effects in the integrating hardware will accumulate because there is no feedback from the integrator output to its input. The resulting runaway of the estimated output signal is a fundamental problem of open integration, therefore a negative low gain feedback is added to stabilize the integration and prevent the output to increase without bounds. The feedback signal converts the integrator into a first order delay having a low corner frequency $1/\tau_1$. In this way, the stator models (3.10) and (3.11) become

$$\tau_1 \frac{d\Psi_s^s(t)}{dt} + \Psi_s^s(t) = \tau_1 (\mathbf{u}_s^s(t) - \mathbf{R}_s \mathbf{i}_s^s(t)), \quad \Psi_r^s(t) = \frac{L_r}{L_m} (\Psi_s^s(t) - \sigma L_s \mathbf{i}_s^s(t)), \quad (3.12)$$

and

$$\tau_1 \frac{d\Psi_r^s(t)}{dt} + \Psi_r^s(t) = \frac{\tau_1 L_r}{L_m} \left(\mathbf{u}_s^s(t) - \mathbf{R}_s \mathbf{i}_s^s(t) - \sigma L_s \frac{d\mathbf{i}_s^s(t)}{dt} \right) \quad (3.13)$$

respectively.

Other models of course can be also derived from the system equations, but it is important to tell that the accuracy of these models greatly depends on the correct setting of the model parameters. Thus, a correct identification of these parameters has to be provided before any application of these models, or in the other case,

because of these incorrect parameter values, the models lose the relationship with reality.

3.3. Adaptive observers & filters

The accuracy of the open loop estimation models described in the previous chapter reduces as the mechanical speed of the IM lowers. The limit of acceptable performance depends on how precisely the model parameters can be matched to the corresponding parameters in the actual machine. Particularly at the lower speed range the parameter errors have significant influence on the steady-state operation and the dynamic performance of the drive system.

In this case, the robustness against parameter mismatch and signal noises can be improved by employing closed loop observers to estimate the state variables, and the system parameters. Unfortunately, because of the high nonlinearities of the IM model, usually the *linear time invariant* (LTI) model-based estimation methods do not provide much better results than the feedforward approach mentioned before, because they need the linearization of the system model, which cannot be accomplished without great loss of the described system dynamics. Therefore, mainly nonlinear methods provide that quality of performance, which is enough to implement these much more computation needed methods instead of the simple ones mentioned before.

3.3.1. Full order nonlinear observer

Estimation methods are constructed in such a way that they are capable to compute the unknown states of the system with the help of measured inputs and reference outputs of the system. The deviation from these reference signals provides the closed loop refinement of the estimation.

In the LTI case, Luenberger observers can be easily constructed to the linearized motor model, such as it is described in [33], but their performance is much worse than if the idea of this method is applied directly to the untouched nonlinear system. In this way, such a full order nonlinear observer can be constructed from the (2.7), (2.8), (2.11), and (2.12) machine equations in the

stator fixed reference frame, which are visualized in the upper portion of *Figure 3.5*.

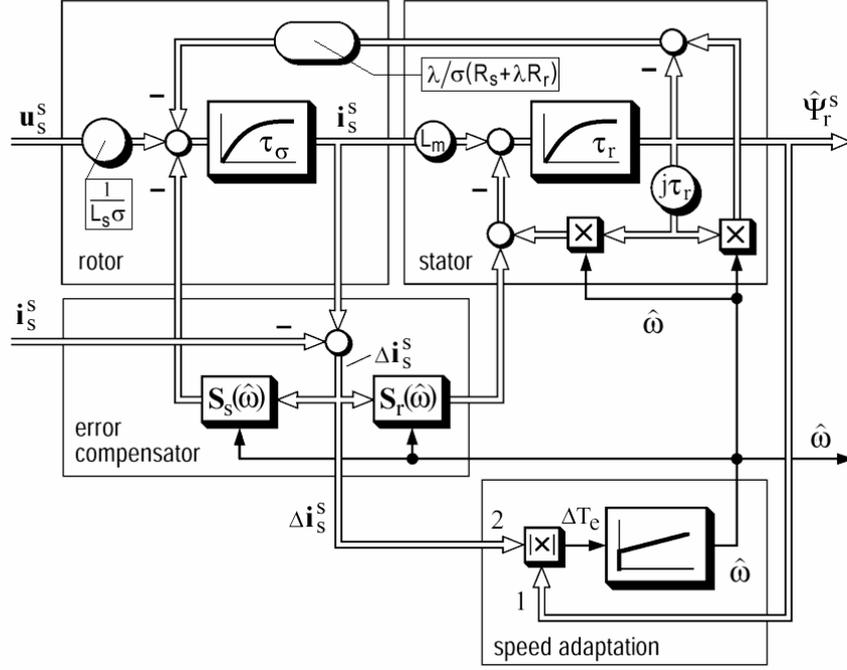


Figure 3.5. Full order nonlinear observer

The model outputs are the estimated values of $\hat{\mathbf{i}}_s^s$ and $\hat{\Psi}_r^s$ respectively. Adding an error compensator to the model establishes the observer. The error is computed by the deviation of the model current from machine current: $\Delta \mathbf{i}_s^s(t) = \hat{\mathbf{i}}_s^s(t) - \mathbf{i}_s^s(t)$. This value is used to generate correcting inputs to the electromagnetic subsystem that represents the stator and the rotor in the machine model. The equations of the full order observer are then established in accordance with the machine equations.

$$\frac{d\hat{\mathbf{i}}_s^s(t)}{dt} = -\frac{\lambda\tau_r + \tau_s}{\sigma} \cdot \hat{\mathbf{i}}_s^s(t) + \frac{\tau}{\sigma} \left(\frac{1}{\tau_r} - j\omega \right) \hat{\Psi}_r^s(t) + \frac{1}{L_s\sigma} \mathbf{u}_s^s(t) - \mathbf{S}_s(\hat{\omega}) \cdot \Delta \mathbf{i}_s^s(t), \quad (3.14)$$

$$\frac{d\hat{\Psi}_r^s(t)}{dt} = -\left(\frac{1}{\tau_r} - j\omega \right) \hat{\Psi}_r^s(t) + \frac{L_m}{\tau_r} \cdot \hat{\mathbf{i}}_s^s(t) - \mathbf{S}_r(\hat{\omega}) \cdot \Delta \mathbf{i}_s^s(t), \quad (3.15)$$

where the complex gain factors $\mathbf{S}_s(\hat{\omega})$ and $\mathbf{S}_r(\hat{\omega})$ are selected in such a way that the resulted two complex eigenvalues of the observer are the $k > 1$ real constant multiplied eigenvalues of the machine. By Kubota [27], this calibration of the gains provides good performance, because $k > 1$ scales the observer by pole

placement to be dynamically faster than the machine. The nonlinearity of the system is also considered, because the complex gains depend on the estimated mechanical speed.

The signal $\hat{\omega}$ is required to adapt the rotor structure of the observer to the mechanical speed of the machine. It is obtained through a PI-controller from the current error $\Delta \mathbf{i}_s^s$. In fact, the term $\hat{\Psi}_r^s(t) \times \Delta \mathbf{i}_s^s(t)$ represents the torque error ΔT_e , which can be verified from (2.17). If a model torque error exists, the modeled speed signal $\hat{\omega}$ is corrected by the PI controller in *Figure 3.5*, adjusting the input to the rotor model in this way. The phase angle of $\hat{\Psi}_r^s$, which estimates the true field angle of the motor, can be used for field orientation as well. The correct estimate is reached when each of the $\Delta \mathbf{i}_s^s$ and ΔT_e error signals reduce to zero.

3.3.2. Sliding mode observer

The effective gain of the error compensator can be increased by using a sliding mode controller to tune the observer for speed adaptation and for rotor flux estimation [4]. This method is proposed by Sangwongwanich and Doki [10]. *Figure 3.6* shows the dynamic structure of the error compensator.

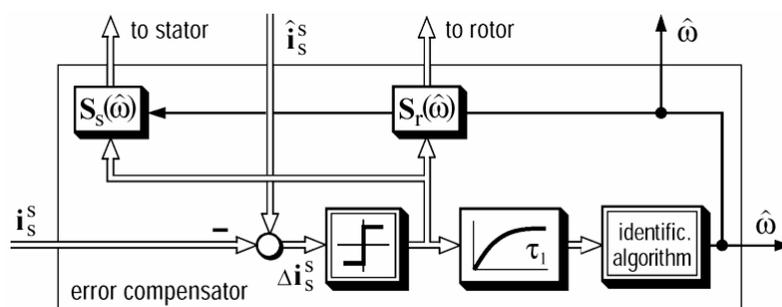


Figure 3.6. Sliding mode compensator

It is interfaced with the machine model the same way as the error compensator in *Figure 3.5*. In the sliding mode compensator, the current error vector $\Delta \mathbf{i}_s^s$ is used to define the sliding hyperplane. The magnitude of $\Delta \mathbf{i}_s^s$ is then forced zero by a high-frequency nonlinear switching controller. The switched waveform can be directly used to exert a compensating influence on the machine model, while its average value controls an algorithm for speed identification. The robustness of the

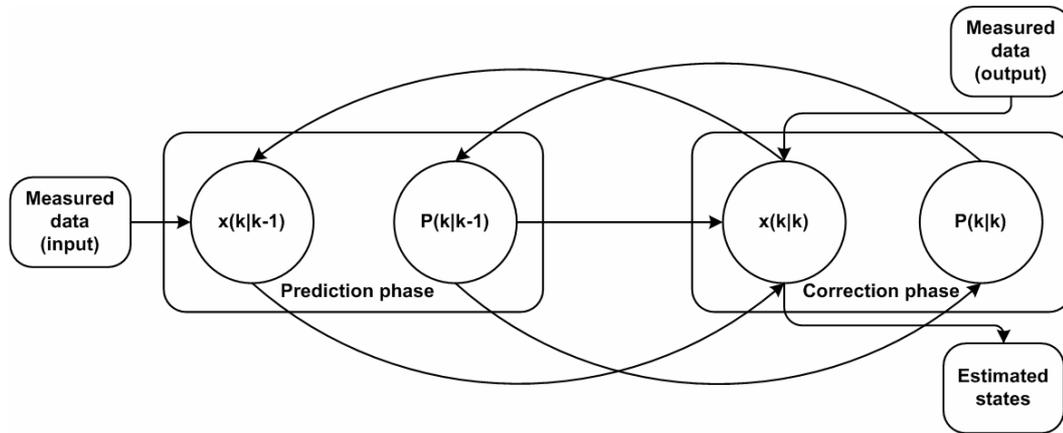
sliding mode approach ensures zero error of the estimated stator current and trough it the tracking of the rotor speed. The H_∞ approach, which theory is discussed later, used in [27] for pole placement in the observer design even minimizes the rotor flux error in the presence of parameter deviations, which provides a robust control structure. But the implementation of this method requires very fast microprocessor.

3.3.3. Extended Kalman Filter

Kalman filtering techniques are based on the complete machine model, and try to statistically refine the quality of estimation. The machine is usually modeled as a 3rd or 4th order system like in [13, 41], introducing the mechanical speed as an additional state variable.

The mechanism of the Kalman filter is represented in *Figure 3.7*, which consist of merely two steps in each control cycle. It uses the same approach as the previously mentioned observers, by taking the measured input signals and reference output signals of the system to solve the problem at hand. The basics of the algorithm is as follows [3]: In the first step the measured data is obtained, and with the new information, the differential equations of the complete model are solved by numerical approximation, which is usually done through the explicit Euler method. In this way the new $\mathbf{x}(k|k-1)$ estimated states of the system are predicted based on the previous $k-1$ estimates and the input data. Then, the predicted states are statistically corrected with respect to the expected covariance of the predicted states $\mathbf{P}(k|k-1)$ and the *a priori*-obtained information about the covariance's of the existing system and measurement noises described in the \mathbf{R} and \mathbf{Q} matrices. The expected noises of the system are white noise processes with zero mean value and constant covariance. The method provides the correctness of the estimation only in this case, which might seem to be a very strict assumption, but by considering the real life systems, the existing noises can be adequately modeled with such effects. This is true in case of the IM as well, because the measurement noise of the stator current signal and the system noise caused by the PWM modulation of the inverter can be approximated well by such a description. In the correction phase of the algorithm, not only the states are corrected to

provide $\mathbf{x}(k|k)$, but the refined value of $\mathbf{P}(k|k)$ is also obtained. After this step the new estimated system states can be used by the control algorithm, which can be implemented in several ways like in [39].



3.7. The basic mechanism of the Kalman filter algorithm

The above mentioned approach is derived from the minimization of a quadratic error function, which is evaluated on the basis of the predicted state variables, noises, and parameter deviations. In this way it reduces error sensitivity and permits also the use of models of lower order than the machine [15].

Since the model of the IM is highly nonlinear and the original method [25] for the Kalman filter is given for the LTI case, an extended Kalman algorithm must be applied to solve the control problem of the machine. This algorithm consist of the similar steps as the original one, but instead of using constant matrices in the prediction phase it substitutes into the nonlinear equation system of the model. The numerical approximation usually needs much more accuracy than the Euler method provides, therefore higher order methods are used, like the Adams-Basforth 3rd order recursive method [14] in [39]. Furthermore, for the prediction of the system covariance matrix, the full partial differentials of the nonlinear equations must be computed. In this way, the extended algorithm basically linearizes the model in the actual operating point in each control cycle, which has a huge computational load. Therefore, the realization of these methods can only be imagined on a floating point high speed hardware which can handle this load.

3.3.4. H_∞ based LPV observer

In *Control Systems Theory*, nowadays new methods evolve to solve the increasing need of accurate and robust control, opening new ways to handle problems never seen before. The most promising of these methods are the H_∞ and H_2 norm based controller synthesis which have been used in several applications [9, 11, 28, 31, 39]. These theories were introduced by Zames [42] in his first paper trying to improve the bad robust properties of the *linear quadrant Gaussian* (LQG) control. To resolve this problem, he introduced an integration constraint which used the norm of the system transfer function. Based on his foundation, the H_∞ norm, which is the cornerstone of the theory, is defined for the $\mathbf{G}(s)$ transfer function of a $\mathbf{P}(\mathbf{w}(t),\mathbf{z}(t))$ system as follows:

$$\|\mathbf{G}(s)\|_\infty = \sup_{\|\mathbf{w}(t)\|_2 \neq 0} \frac{\|\mathbf{z}(t)\|_2}{\|\mathbf{w}(t)\|_2}, \quad t \in (0, \infty), \quad (3.16)$$

where $\mathbf{w}(t)$ denotes the input disturbances, reference signals, and noises effecting the system and $\mathbf{z}(t)$ refers to the output perturbations like reference tracking errors or other control aims. The essence of the H_∞ theory can be briefly explained by *Figure 3.8*. By the represented system, such a \mathbf{K} controller is searched for which stabilizes the given \mathbf{P} plant, and based on the (3.16) norm it minimizes the transfer function between $\mathbf{w}(t)$ and $\mathbf{z}(t)$.

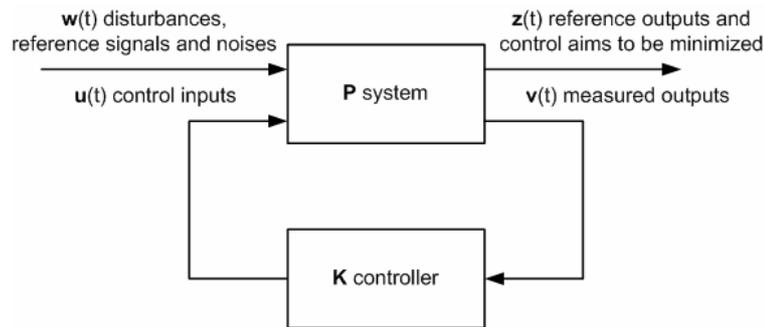


Figure 3.8. General problem formulation of H_∞ control problems

According to the *small-gain* theorem the robust stability is dependent on the overall gain between the disturbances and the output signals of the closed loop system [42]. Therefore, the minimization of this norm can be viewed as the maximization of the stability and goodness of the control. The H_∞ norm gives the worst case gain of the system therefore provides a good match to engineering

specifications and a useful description of the error and input signals of the closed loop control, because it also represents an upper and lower bound on these signals on the whole frequency range.

Instead of the direct minimization of the mentioned transfer function, which is an enormous task, of course, most commonly only a suboptimal controller is obtained, which can be computed through a recursive gamma-iteration based on the solution of Riccati equations or *linear matrix inequalities* (LMIs). The obtained controller is specified by its γ worst case gain which is equal to the closed loop H_∞ norm of the controlled system.

This theory provides both controller schemes and observer solutions for the problem, depending on how the \mathbf{P} system is defined. By clever approach, the synthesised estimators can provide very accurate results with a small computational load. With the use of frequency range describing filters in the definition of the \mathbf{P} problem even the properties of the produced algorithm can be tuned. This approach is called *mixed sensitivity* [34].

Unfortunately, this very useful theory is only given for LTI systems, but for such NL models as the IM, an extension of the theory is needed. By the use of the *linear parameter variant* LPV representation of the machine given in *Section 2.5* the NL form of the equations can be handled as LTI parameter dependent systems defined on a parameter space, on which a H_∞ controller can be synthesised. In this way, the NL problem can be handled as an LTI one but with the cost of the description of the parameter-dependent exact location of the model in a system space, which representation is called *polytopical* description. This requires higher and higher computational load based on the difficulty of the model, which can lower the economical value of this brilliant method.

Today, there are several examples of experimenting with this method to solve the control problem of the IM, using this theory for observer or controller synthesis like in [9, 11, 31, 39]. Although the obtained promising results, the industrial usage of this approach is not significant, which can be explained by the fast microprocessor need and the freshness of this solution.

3.3.5. Model reference adaptive systems

The *model reference adaptive system* approach (MRAS) makes use of the redundancy of two machine models of different structures that estimate the same state variable on the basis of different sets of input signals [35]. One of these MRAS models is called the reference model and the other one is referred as the adaptive model. The general mechanism of these methods is based on comparing the outputs of these models to estimate the desired unknowns by continuous refinement of the adaptive structure.

In the following example, both models are represented in the stator fixed reference frame. The (3.12) stator model in the upper portion of *Figure 3.9* serves as the reference model. Its output is the estimated rotor flux vector $\hat{\Psi}_r^{s \leftarrow s}$, where superscript $\leftarrow s$ indicates that the vector originates from the stator model.

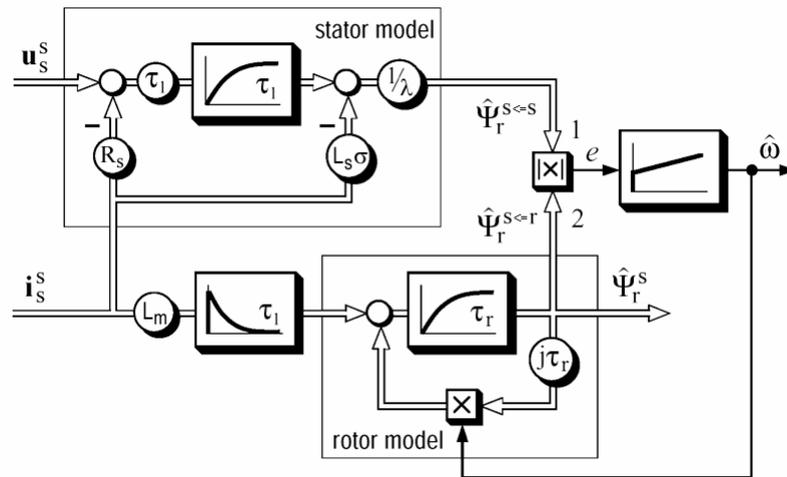


Figure 3.9. Model reference adaptive system for speed estimation, reference variable: Ψ_r^s

Additionally, the rotor model (2.29) with $\omega_k = 0$, which is the adaptive model, estimates the rotor flux $\hat{\Psi}_r^{s \leftarrow r}$ from the measured stator current and from the tuning signal $\hat{\omega}$, like it presented in *Figure 3.9*. Then, the tuning signal is obtained through a *proportional-integral* (PI) controller from a scalar error signal $e = \hat{\Psi}_r^{s \leftarrow s}(t) \times \hat{\Psi}_r^{s \leftarrow r}(t)$. As the error signal e gets minimized by the PI controller, the tuning signal $\hat{\omega}$ approaches the actual speed of the motor. The rotor model, as

Equations (3.17) and (3.18) are derived from (2.35) with the steady state solution of the rotor field oriented coordinates.

It is a particular asset of this approach, that the accurate orientation of the injected current vector is maintained even if the model value of τ_r differs from the actual rotor time constant of the IM. The reason is that the same, even erroneous value of τ_r is used both in the rotor model and in the control algorithm (3.17) and (3.18) of the speed control scheme in *Figure 3.10*, then if the tuning controller (*Figure 3.9*) maintains zero error, the control scheme exactly replicates the same dynamic relationship between \mathbf{i}_s^s and Ψ_r^s that exist in the actual motor, even in the presence of a rotor time constant error [35]. However, the accuracy of speed estimation, reflected in the feedback signal $\hat{\omega}$ to the speed controller, does depend on the error in τ_r . The speed error may be even higher than with those methods that estimate ω_r and use $\hat{\omega}(t) = \omega_s(t) - \hat{\omega}_r(t)$ to compute the speed. The reason why this approach is more robust considering the speed than the MRAS, is that the synchronous frequency of the power feed is a control input and therefore accurately known, and even if $\hat{\omega}_r$ is erroneous, its nominal contribution to $\hat{\omega}$ is small, thus it does not effect the speed estimation, unless the real speed is very low. As it could be seen MRAS offer a powerful method to realize low performance speed sensorless control even with its robustness problem.

3.4. Direct torque control

Direct torque controlled (DTC) induction drives were developed approximately 20 years ago by Dependbrock [8] and Takahashi [36], however at present, ABB is the only industrial company who have introduced a commercially available DTC drive. In a DTC-IM drive, supplied by a *voltage source inverter* (VSI), it is possible to control directly the stator flux linkage (or the rotor/magnetizing flux linkage) and T_e by the selection of optimum inverter switching modes. The selection is made to restrict the flux and torque errors within respective flux and torque hysteresis bands to obtain fast torque response, low inverter switching frequency and low harmonic losses. In the following section, such a DTC drive will be briefly shown which in addition to controlling

T_e , it also controls the stator flux linkage. However, it is also possible to have other implementations in which the rotor or the magnetizing flux linkage is controlled. Generally, DTC drives allow very fast torque responses and flexible control of the IM, although the speed control of the shaft is not provided by this approach, thus usually external control is needed to complete the drive with the use of speed estimation based on the previously mentioned observers.

By considering the (2.19) dynamic motion equation of the IM it is clear that T_e is proportional to the cross-vectorial product of the flux linkage space vector Ψ_s^s and the stator current \mathbf{i}_s^s space vector. In this way, (2.19) transformed into the following form:

$$T_e = \frac{3}{2} p \frac{L_m}{L_r} |\Psi_s^s(t)| \cdot |\mathbf{i}_s^s(t)| \cdot \sin(\underbrace{\alpha_i(t) - \alpha_\psi(t)}_{\alpha(t)}), \quad (3.19)$$

where $\alpha_i(t) = \angle(\mathbf{i}_s^s(t))$ and $\alpha_\psi(t) = \angle(\Psi_s^s(t))$. It can be shown, that by using the voltage equations of the IM, for a given value of the rotor speed, if the modulus of the Ψ_s^s is kept constant and the angle α_ψ is changed quickly, then T_e can be rapidly changed. In other words if such \mathbf{u}_s is imposed on the motor, which keeps the magnitude of stator flux constant at the demanded reference value, but which quickly rotates Ψ_s^s into the position required by the torque command, then fast torque control is performed. It follows that in the DTC drive, if the developed actual T_e is smaller than the reference value, then its increase should be achieved as fast as possible by using the greatest $d\alpha_\psi / dt$. However, when T_e is equal to its reference, the rotation is stopped. If the stator flux linkage vector is accelerated in forward direction, then positive T_e is produced, and when it is decelerated backwards, negative T_e is produced. However, the Ψ_s^s can be adjusted by using appropriate \mathbf{u}_s , which is generated through the inverter, which produces the voltage feed of the IM.

The time constant of a standard squirrel cage IM is large in comparison to the power electronic elements of the inverter, thus Ψ_r^s changes slowly compared to

Ψ_s^s . Therefore, its magnitude can be assumed to be constant. This follows from the (2.8) rotor voltage equation, if the magnitude of Ψ_s^s is assumed to be constant as well. However, if both of the magnitude of these vectors are assumed to be constant, then the electromagnetic torque equation in form of (3.20) yields that T_e can be rapidly changed by changing the slip angle $\alpha_{\text{slip}}(t) = \angle(\Psi_s^s(t)) - \angle(\Psi_r^s(t))$ in the required direction determined by the torque command.

$$T_e = \frac{3}{2} p \frac{L_m}{L_r} |\Psi_s^s(t)| \cdot |\Psi_r^s(t)| \cdot \sin(\alpha_{\text{slip}}(t)), \quad (3.20)$$

In case of field weakening, which case is not described here, the magnitude of Ψ_s^s is not constant, therefore the control of the magnitude and slip angle has to be provided, which is also possible by the switching of appropriate inverter voltages.

In contrast to a vector controlled IM drive, which is going to be mentioned later, where the stator currents are used as control quantities, in the DTC, the stator flux linkages are controlled. In both methods this is done through the \mathbf{u}_s vector. If, for simplicity's sake, it is assumed, that the stator ohmic drops can be neglected, then $d\Psi_s^s(t)/dt = \mathbf{u}_s(t)$, so the inverter-provided voltage directly impresses the stator flux, and thus the required stator-flux locus will be obtained by using appropriate switching states of the inverter. It follows from the previous concepts that for a short Δt time, when the stator voltage is applied $\Delta\Psi_s^s = \mathbf{u}_s \cdot \Delta t$ holds. Thus, Ψ_s^s moves by $\Delta\Psi_s^s$ into the direction of \mathbf{u}_s at a speed which is proportional to $|\mathbf{u}_s|$, where the magnitude of the stator voltage is proportional to the DC voltage of the inverter bus. By selecting step by step the appropriate stator voltage vectors, it is possible to change the stator flux in the required way.

By considering the layout of a six-pulse VSI shown in *Figure 3.11*, it can be noticed, that there are six possible non-zero active voltage-switching space vectors ($\mathbf{u}_1, \dots, \mathbf{u}_6$) and two zero space vectors ($\mathbf{u}_7, \mathbf{u}_8$). These are shown in *Figure 3.12*, which also presents the corresponding eight inverter states. The six inverter-switching vectors can be expressed as

3. Existing speed sensorless techniques

$$\mathbf{u}_k = \frac{2}{3} U_{bus} \cdot e^{\frac{j(k-1)\pi}{3}}, \quad k = 1, 2, \dots, 6 \quad (3.21)$$

where U_{bus} is the DC bus voltage. However, in case of $k = 7, 8$, $\mathbf{u}_k = \mathbf{0}$ holds for the two zero switching states where the stator windings are short circuited.

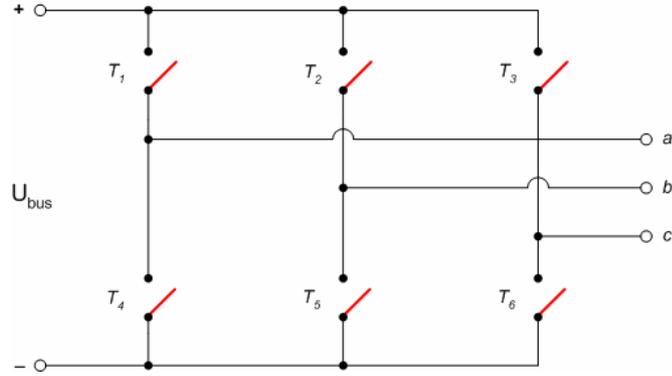


Figure 3.11. Schematic representation of a voltage source inverter

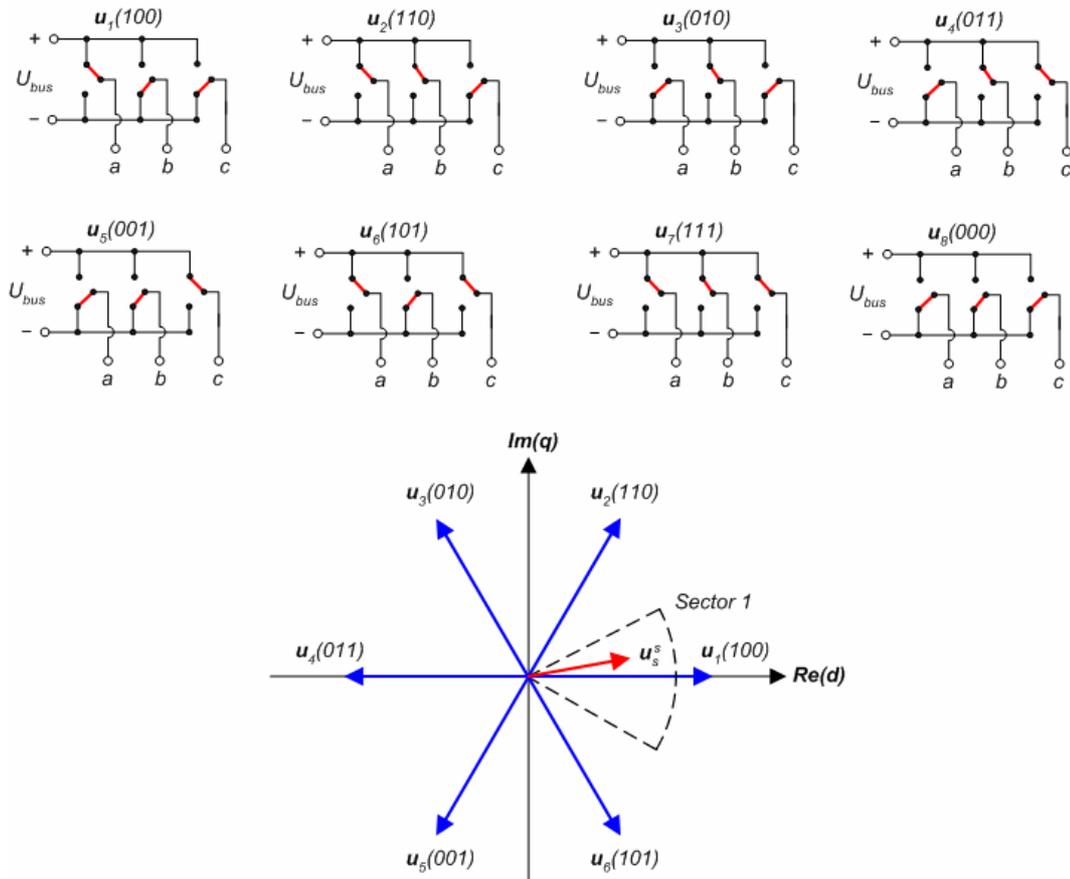


Figure 3.12. Possible switching vectors and the corresponding states of PWM-VSI inverters

Since $\Delta\Psi_s^s = \mathbf{u}_s \cdot \Delta t$, and as were previously mentioned Ψ_s^s moves fast into the direction of the applied \mathbf{u}_s , which is one of the six inverter-switching vectors, and stops in case of the two zero switching states, therefore it can be seen, that Ψ_s^s moves along a hexagonal path with constant linear speed if the applied voltage vector is continuously circulated in the order of the given switching vectors. In the DTC drive, at every sampling period, the switching vectors are selected on the basis of keeping the stator flux-linkage errors and the torque error in a given tolerance band. It is assumed that the widths of these tolerance bands are $2\Delta T_e$ and $2\Delta\Psi_s$. If the Ψ_s^s lies in the k^{th} sector (see *Figure 3.13*) its magnitude can be increased by using switching vectors $\mathbf{u}_k, \mathbf{u}_{k+1}, \mathbf{u}_{k-1}$, and decreased with $\mathbf{u}_{k+2}, \mathbf{u}_{k-2}, \mathbf{u}_{k+3}$. Obviously, the selected voltage switching vectors affect T_e as well. The speed of the Ψ_s^s is zero in case of \mathbf{u}_7 and \mathbf{u}_8 and it is possible to change its speed by changing the output ration between the zero and non-zero voltage vectors.

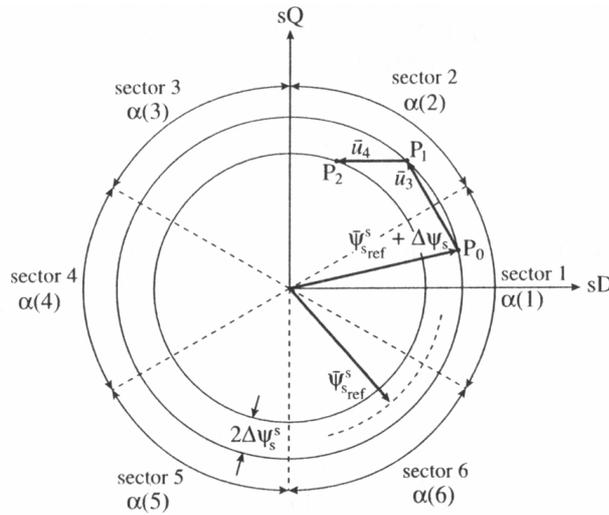


Figure 3.13. Flux linkage control by DTC

As shown above, Ψ_s^s is basically the integral of \mathbf{u}_s and it will move into the direction of the selected \mathbf{u}_k as long as it is applied. Thus, if a reduced Ψ_s^s modulus is required, then it can be achieved by applying such switching voltage vectors which are directed towards the center of the rotor, and if an increased modulus is required, then it is controlled by applying voltage vectors which are direct out

from the center of the rotor. This is illustrated in *Figure 3.13*. Therefore, by this method, the magnitude of Ψ_s^s can be kept in a $2\Delta\Psi_s$ hysteresis band. It is also true that in general, if an increase of the torque is required, then the torque is controlled by applying voltage vectors that advance Ψ_s^s in the direction of the rotation, and if decrease is required, then such vectors are selected which oppose the direction of the torque. If zero torque is required, then \mathbf{u}_7 and \mathbf{u}_8 are applied, by which the rate of torque change can be also controlled.

Based on this, such an optimal voltage switching vector look-up table can be obtained that tells which vector should be applied to reach the desired state. Several other considerations also have to be applied of course, for example the length of switch on of each vectors in the switching pattern to control the speed of torque and flux rise.

It is also important to tell that for the functioning of such a drive an estimation of the stator flux is needed which can be done in several ways mentioned before, but the optimality of the control strongly depends on the accuracy of the estimation.

Figure 3.14 shows the schematic of one simple form of the DTC IM drive, employing a VSI inverter. In this scheme the stator flux is the controlled flux, thus it is referred to be stator-flux based DTC drive.

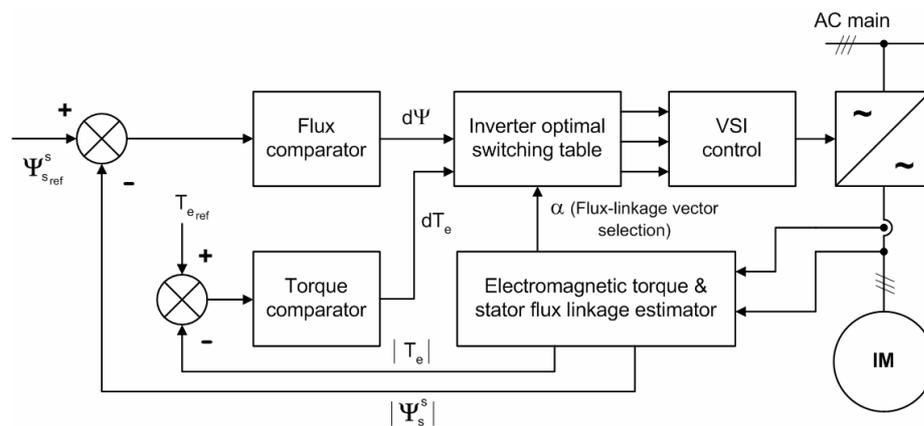


Figure 3.14. Structure of a simple stator-flux based DTC-IM drive with VSI

The main mechanism of this structure is clear from the point of view of the previous concepts and only the estimation method has to be synthesized in such a way that it can provide the accurate states of the motor. For a simple scheme, the

previously mentioned stator flux estimators can be used with the following equation to obtain T_e :

$$T_e = \frac{3}{2} p \frac{L_m}{L_r} (\Psi_{s\alpha}^s(t) \cdot i_{s\beta}^s(t) - \Psi_{s\alpha}^s(t) \cdot i_{s\beta}^s(t)). \quad (3.22)$$

In this way, the main advantages of the produced DTC drive is the absence of coordinate transformations and voltage modulator block, which are required in space vector drives. Moreover, much smaller accuracy of the flux vector is needed in the estimation because only the 60° long sector has to be determined for this method and it has also minimal torque response time. However, it has disadvantages as well, like possible problems during start up and low speed operation and the quickly changing switching frequency, which produces high ripple torque.

3.5. Vector control

The basics of vector control have been already discussed in *Section 2.4*. The main advantage of vector control is that it makes possible the representation of the system model in such ways that the control and estimation of the whole model becomes relatively simple. For example, with the introduced field oriented representation of the motor model the decoupled control of speed and flux can be achieved with optimal performance of the control, because this approach transforms the IM into a DC motor like representation. However, several other representations are known, for example the stator flux orientation, magnetizing current orientation, etc., with the known advantage and disadvantage of each of them, but most commonly the rotor field orientation is used. Unfortunately, it is generally true, that the good control abilities of the oriented model representations have to be paid in increased computational load.

The main structure of speed sensorless vector-controlled drives with VSI inverter can be seen in *Figure 3.15*.

In every case, if the high needs of the industry have to be fulfilled, the vector control algorithm must consist of a speed and flux estimation method, which can be implemented in the previously mentioned ways.

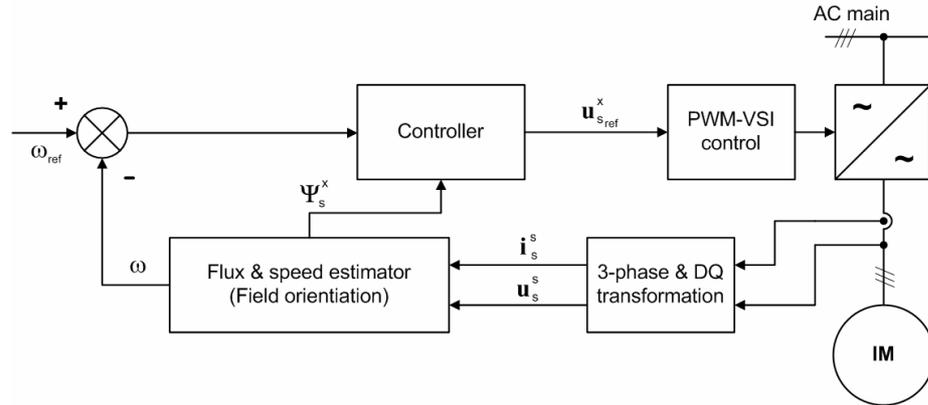


Figure 3.15. General structure of VSI driven speed sensorless, vector controlled IM drives

However, for faultless operation, very accurate estimation of speed and flux must be implemented, because in other case, the field orientation will be missed and the drive becomes instable. After this measured stator current based estimation, the obtained space vectors of electrical and magnetical quantities are aligned with the rotating reference frame. This orientation is usually computed through the *Park transformation*:

$$\begin{bmatrix} x_d^k(t) \\ x_q^k(t) \end{bmatrix} = \begin{bmatrix} \cos \rho_k(t) & \sin \rho_k(t) \\ -\sin \rho_k(t) & \cos \rho_k(t) \end{bmatrix} \cdot \begin{bmatrix} x_\alpha^s(t) \\ x_\beta^s(t) \end{bmatrix}. \quad (3.23)$$

where ρ_k is the angle between the stator fixed reference frame and the instant position of the oriented reference frame. The space vector \mathbf{x} presented in (3.23) is computed from its three-phase quantities by the *Clark transformation* method:

$$\begin{bmatrix} x_\alpha^s(t) \\ x_\beta^s(t) \end{bmatrix} = \begin{bmatrix} \frac{2}{3} & -\frac{1}{3} & -\frac{1}{3} \\ 0 & \frac{1}{\sqrt{3}} & -\frac{1}{\sqrt{3}} \end{bmatrix} \cdot \begin{bmatrix} x_a(t) \\ x_b(t) \\ x_c(t) \end{bmatrix}, \quad (3.24)$$

These transformations are usually applied for the measured stator currents in case of VSI feed. As it can be seen, only the estimation and the accurate orientation are significant tasks to be accomplished in vector controlled drives, however, these tasks is worth the effort, because in this way, decoupled torque and speed control of the IM can be obtained, which is free from the ripple torque effects of the DTC.

The control loop is usually synthesised based on the full information obtained from the previous steps. Moreover, in case of the field orientation, the steady state values of the machine states are constant, therefore even PID-based controller can be used for good performance control. Of course, to fully use the opportunities of this approach, more sophisticated control algorithms are developed, such as the H_∞ theory-based solutions of [11, 30, 39], which provide performance that can only be hardly achieved in other ways. For control signal, most commonly the stator voltage is used, which is transformed back into the stationary reference frame and then realized through the PWM-modulated voltage provided by the VSI. The exact way of this realization will be discussed in the next section.

Nowadays, high performance vector control based IM drives are produced by Siemens, which have more computational load on the built in microcontroller than other methods. Therefore, these drives cost more than conventional techniques based solutions, but above 5kW the relative value of this extra cost is insignificant to the price of power electronic devices for this magnitude of rated power.

3.6. Fuzzy control

The artificial intelligence-based vector control appeared almost 20 years ago and it was thought of the possible future control solution for IM. In the literature, there are many papers which discuss various vector drives with fuzzy-logic based speed controllers, but only a few papers discuss implementations, and the others mainly concentrate on simulations. Moreover, the commercial implementation of this type of control is non-detectable. The reason is very simple. In the case of the previously mentioned non-AI approaches, the stability of the system and its answer to any environmental effect is predictable, while in the case of the AI methods, any speculations to prove the overall stability are only divinations. In spite of the main advantage of the fuzzy control results in the reference tracking without overshoots, this has not shown enough importance to be used in real life applications.

There are many types of fuzzy-logic controllers (FLCs), but a good example is the Mamdani [29] type, shown in *Figure 3.16*.

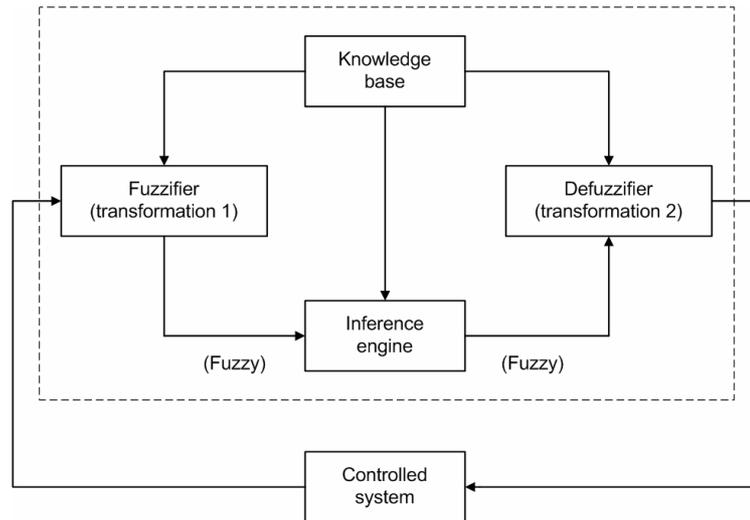


Figure 3.16. General structure of Mamdani type Fuzzy-logic controllers

In general, the fuzzy logic controller contains four main parts, two of which perform transformations. These are the following:

◆ *Fuzzifier (transformation 1)*

The fuzzifier performs measurements of the input variables and scale mapping of the obtained values into fuzzy quantities using membership functions which tell that the quantity belongs to which specific fuzzy set determined by the designer. A membership function has a value between 0 and 1, and it indicates the degree of belongingness of a quantity to a fuzzy set. For example 0.5 means that there is only 50% chance for the correct belongingness.

◆ *Knowledge base*

It consists of a data base and a linguistic-control rule base. The data base provides the information which is used to define the linguistic control rules and the fuzzy data manipulation in the FLC. The rule base specifies the control goal actions by means of a set of linguistic control rules provided by the designer. The FLC looks at the input signals and by using these *expert rules* determines the appropriate output signal as a control action. Therefore, the rule base contains a set of if-then rules which provide the control scheme.

◆ *Inference engine*

It is the kernel of the FLC and has the ability both of simulating human decision-making based on fuzzy concepts and of inferring fuzzy control actions by using fuzzy implication and fuzzy-logic rules of inference. In other words, once all the monitored input variables are transformed into their respective linguistic variables the inference engine evaluates the set of if-then rules and the result is obtained which is again a linguistic variable.

◆ *Defuzzifier (transformation 2)*

The defuzzifier yields a non-fuzzy, real control action from the inferred fuzzy control action by using membership functions. Physically this corresponds to taking a weighted average of the control action contributions from each of the fuzzy rules and then obtain the final decision.

Usually the whole FLC is designed purely on a neural system approach and has the property of self learning if during the design stage it is taught that whether good or bad action was performed. As a solution, this type of control is very unique and provides robust performance with good dynamic behavior, but as it was stated, if any event occurs which are not thought to the controller or not considered during the design of the rule set, the system behavior becomes unpredictable [41].

4. Operation of induction motor drives

Over the last 45 years, a revolution has occurred in the application of electric motors. The development of solid-state motor drive packages has progressed to the point where practically any power control problem can be solved by using them. With such solid state drives, it is possible to run DC motors from AC power supplies or AC motors from DC power supplies. It is even possible to change AC power at one frequency to AC power at another frequency, which is crucial for the realization of command signals in IM control problems. Furthermore, the cost of solid-state drive systems has decreased dramatically, while their reliability has increased. The versatility and the relatively low cost of solid state controls and drives have resulted in many new applications for the AC motors, such as the IM, in which they are doing jobs formerly done by DC machines [6].

In the following section the general power electronic solution of IM drives will be shown with practical concepts that strongly connected to the considered realization, detailed in *Chapters 5 and 6*.

4.1. Classical power feed generation concepts

As it was theoretically introduced in the previous chapters, the control of the magnitude and frequency of the power feed of the IM provides the ability to continuously guarantee the desired speed and torque of the machine. While the control of the magnitude is an easy task with the help of transformers, the frequency conversion of AC power to AC power at another frequency needs the use of complicated solid-state electronics. Traditionally, there have been two approaches to AC frequency conversion: the *cycloconverter* and the *rectifier-inverter*. The cycloconverter is a device for directly converting the frequency of AC power to AC power, while the rectifier-inverter, first converts the AC power to DC power and then converts the DC power to AC again at a different frequency. However, today the rectifier-inverter approach is used most commonly because cycloconverters need more solid-state components and their digital control is much more complicated.

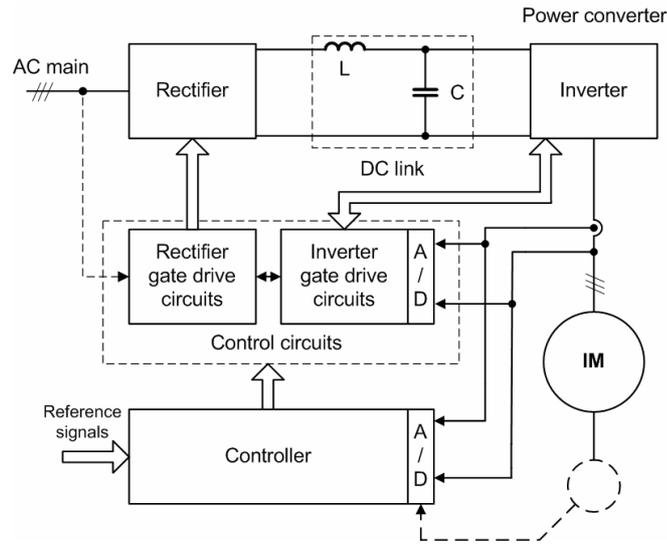


Figure 4.1. General structure of rectifier-inverter powered IM drives

In *Figure 4.1* the most general structure of a rectifier-inverter powered IM drive is presented. As it can be seen, the power processor element converts the three-phase or one-phase AC power to the desired 3-phase AC voltage defined by the microcontroller. The power processor, shortly called the inverter, consists of the following three main elements:

4.1.1. Rectifier

The basic concept of rectifying is to control the flow of energy with the help of diodes or *silicon controlled rectifiers (SCRs)* in such a way that the resulted voltage will consist of only positive voltage components producing nearly a DC signal. There are many different rectifier circuits but the most common full-wave bridge rectifiers are shown in *Figure 4.2(a)* for one-phase input, and in *Figure 4.2(b)* for three-phase input.

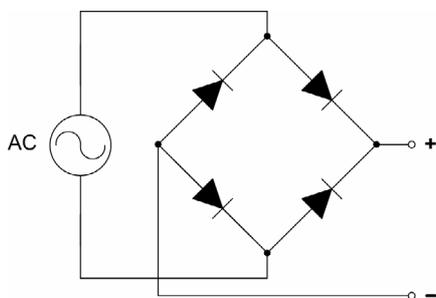


Figure 4.2(a). One-phase full-wave rectifier circuit

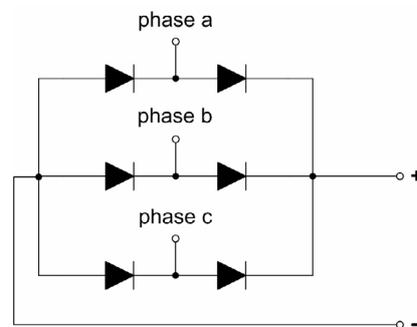


Figure 4.2(b). Three-phase full-wave rectifier circuit

The produced voltage can be seen in case in Figure 4.3(a) and 4.3(b). These circuits have one problem from a motor-control point of view; their output voltage is fixed for a given input voltage.

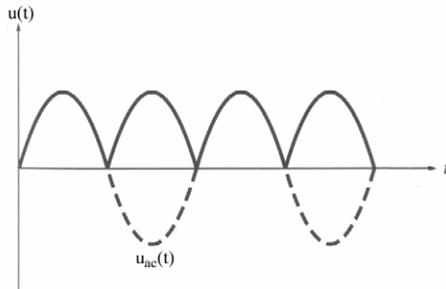


Figure 4.3(a). Output waveform of the one-phase full-wave rectifier bridge

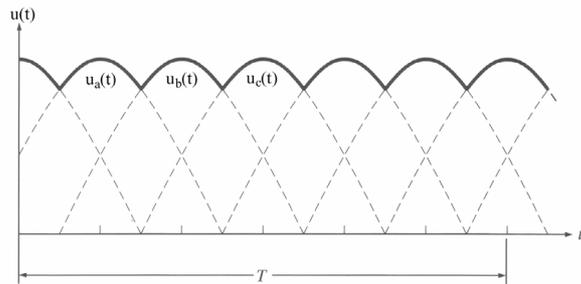


Figure 4.3(b). Output waveform of the three-phase full-wave rectifier bridge

This problem can be overcome by replacing the diodes with TRIACs or SCRs as it can be seen in Figure 4.4.

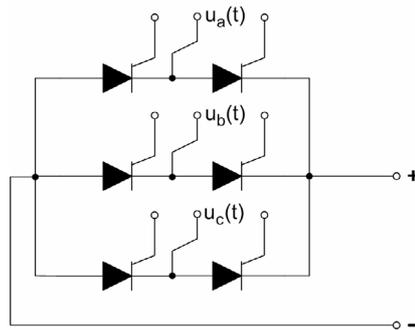


Figure 4.4. Rectifier bridge with SCRs

The average DC output voltage from this circuit depends on when the SCRs are triggered during their positive half-cycles. If they are triggered at the beginning of the half-cycle, this circuit will do the same as the three-phase full-wave rectifier. If the SCRs are never triggered, then the output voltage will be 0V. For any firing angle between 0° and 180° on the waveform, the DC output voltage will be somewhere between the maximum value and 0V. Unfortunately, by using SCRs the output voltage will have more harmonic components than with diodes, but in this way the control of the DC voltage output can be achieved and even back-feeding into the electrical network is possible during the reversal breaking of the IM. With diodes, only dynamic breaking is possible by switching an external breaking resistor to the input side of the inverter bridge with for example a relay.

The control of the triggering signals is provided by the inverter control circuit presented in *Figure 4.1*.

4.1.2. DC link

The DC link is a power capacitor circuit which has two very related purposes. Firstly, it stores energy to provide constant input of the inverter bridge, and it is also referred to as to smooth the incoming signal to filter out the unwanted harmonics. By considering these tasks, it can be seen that they mean the same. A low pass filter constructed from an L inductance and a C capacitor (as it is shown in *Figure 4.1*) fulfils the needs mentioned above, and by a clever choice of their values a very smooth DC power can be produced. In the case of VSI inverters the relative greatness of C is dominant to L, because the smoothing of the voltage is needed, but for CSI inverters a relatively much larger L than C has to be applied. Because creation of large inductances is a very expensive task, and the produced coil has a non-neglectable space need, while large capacities can be cheaply produced in small scale, thus the VSI inverters dominate the market. However, because of the current related control of the IM, CSI inverters would hold much more possibilities.

4.1.3. Inverters

Inverters are classified into two basic types with respect to the used commutation technique: *external commutation* (EC) and *self-commutation* (SC). EC inverters (*Figure 4.5(a)*) are such inverters in which energy, provided by an external power supply, is required to turn off the solid-state switch devices, while SC inverters (*Figure 4.5(b)*) are constructed in such a way that they are capable of self-turning off the solid-state switch devices by the use of stored energy in built-in capacitors or coils. If the regulation of frequency is needed for the output AC power, as in the case of the IM control, both of the two basic types can be used, however, the EC inverters provide more flexibility of AC power production.

In the following sections only the three-phase SC-VSIs will be considered, because they provide the most cost effective way to implement a IM drive. The VSIs can also be divided into two groups: the *pulse width modulation* (PWM)

based inverters and the square wave inverters. PWM inverters require more complex control circuitry and faster switching components, usually their DC bus voltage is constant, and they are capable of controlling the frequency and amplitude of the produced three-phase voltage signal, while square wave inverters require changeable DC bus voltage with controlled rectifier, because they are only able to change the frequency of the produced signal which is similar to a square wave form (hence the name is coming from). The mechanism of PWM inverters will be described in the next section.

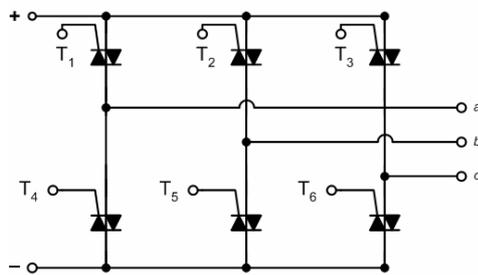


Figure 4.5(a). External commutation needed inverter with TRIACs

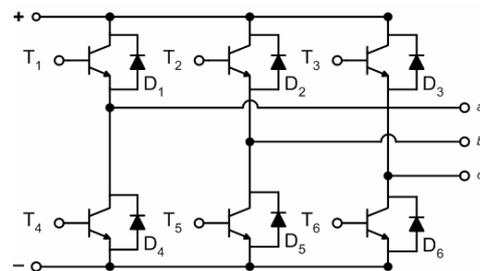


Figure 4.5(b). Self-commutation needed inverter with power transistors

The drive structure presented in *Figure 4.1*, also consists of elements which are responsible for the control of the power processor. These are the following:

4.1.4. Inverter control circuitry

The main purpose of this circuitry is to provide the triggering signals of the inverter and the rectifier bridge with guaranteeing the protection of the drive by means of built in watchdog functions, over voltage and over current protection, thermal protection, and through shoot protection. Here, the through shoot protection means that the solid-state switch devices are triggered in such a way, that the triggering of these devices in the half bridge at once, which produces short circuit, is avoided. The inverter controller device is usually is built up from analog and digital components. By the complexity of these protections and by the digital programmability of this circuits such power systems are called *intelligent power drives*.

4.1.5. The microcontroller

The microcontroller provides the computational muscle for the control algorithm implemented within the drive. It is also capable of the processing of the sensor signals coming from the controlled IM, such as the signal conditioning and the A/D conversion of this analog information. This device also generates the command signal which is realized by the power processor, and because of this, it is the heart and soul of the whole drive. Most commonly, fixed point DSP processors or *programmable integrated circuits* (PICs) are used for this purpose depending on the performance expectations of the drive.

4.2. The PWM inverters

PWM inverters are very popular because of their low cost, high reliability and capability to produce wide range of sinusoidal AC voltage signals in frequency and in amplitude as well. Because, as it will be shown, they require very fast active elements, therefore most commonly *insulated-gate bipolar transistors* (IGBTs) are used for this purpose.

4.2.1. The insulated-gate bipolar transistor

The IGBT is a relatively recent development of power electronics. Its mechanism is similar to the power transistor [6], except that it is controlled by the voltage applied to a gate rather than the current flow into the base, as in the case of the power transistor. The impedance of the control gate is very high in an IGBT, so the amount of current flowing in the gate is extremely small. The device is essentially equivalent to the combination of a *metal-oxide-semiconductor field-effect transistor* (MOSFET) (voltage type, low power demand switching) and a power transistor (low gate-emitter voltage, low power loss). The symbol of an IGBT is shown in *Figure 4.6(a)*.

The characteristics of the IGBTs is given in *Figure 4.6(b)*, from which it can be seen that similarly to the BJT it has a voltage drop in turned out mode, which is 2-3V for a 1000V-rated device. It also needs a deadband time to turn out, similarly to the thyristor, during which it needs to have negative voltage applied between the gate and the emitter till the flow of current drops to zero. If this phenomenon

is not considered, then the device does not turn out, and for example in the inverter bridge it causes a through shoot, which destroys the whole power supply.

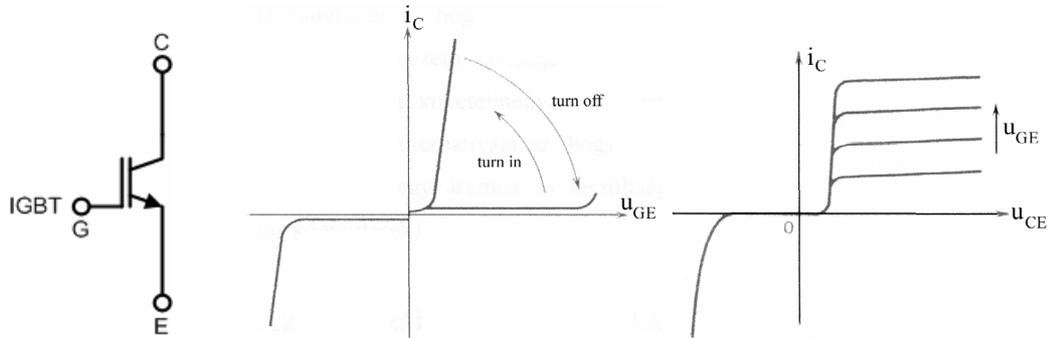


Figure 4.6(a). Symbol of the IGBT

Figure 4.6(b) characteristics of the IGBT

Since the IGBT is controlled by a gate voltage with very little current flow, it can switch much more rapidly than a conventional power transistor can. IGBTs are therefore being used in high-power high-frequency applications, but of course in very high power range, the three-wire thyristor (*also referred to SCR*) is dominant till this day because only this family is capable to conduct 3000A.

4.2.2. The mechanism of PWM-VSI inverters

Figure 4.7 shows a three-phase PWM-VSI using IGBTs as the active elements. This layout is very common, and since the IGBTs are self-commutating, no special commutation components are included in this circuit. Here, the IGBTs, which can be considered as switches, are made to conduct in the following order: IGBT₁, IGBT₆, IGBT₂, IGBT₄, IGBT₃, IGBT₅. The task is theoretically very simple, turn on the switches in such an order that the produced *a*, *b*, *c* indexed voltages become a symmetrical three-phase voltage. This task can be done in the previously mentioned conduct order which produces the output phase and line voltages shown by Figure 4.8. Additionally, in reality, deadband effects and other circuitry specific facts must be considered as well, but the basic idea is the same as it is in Figure 4.7.

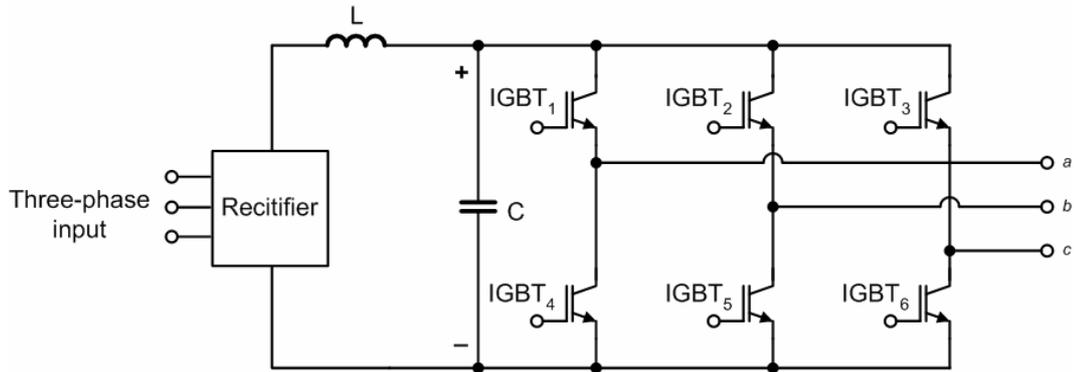


Figure 4.7. General structure of SC PWM-VSIs with IGBTs

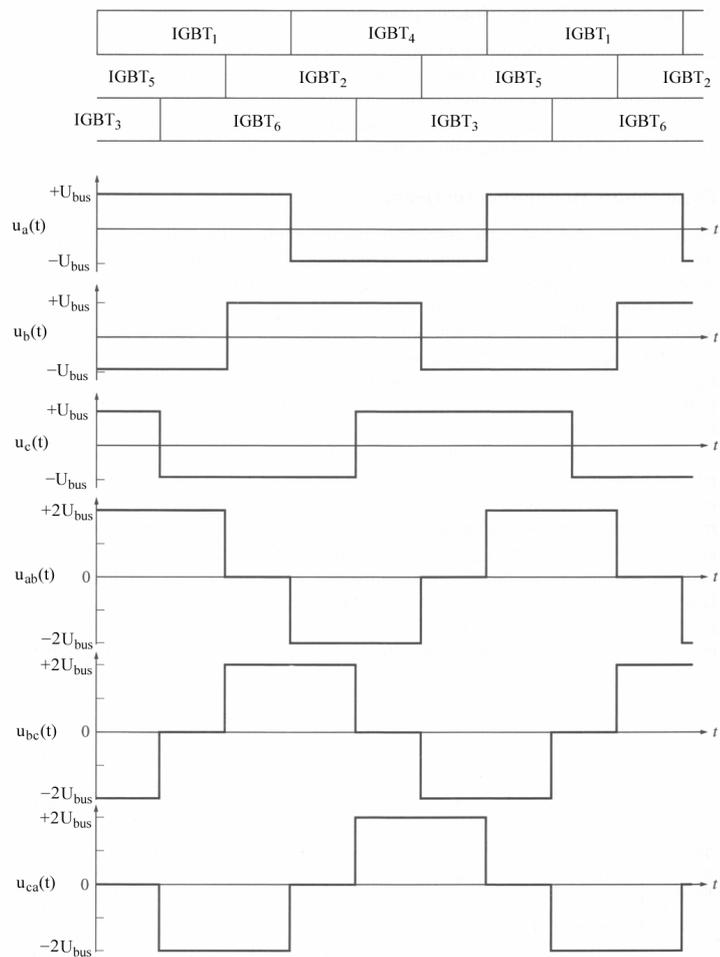


Figure 4.8. Output voltages of the 3-phase PWM-VSI

The PWM is the process of modifying the width of pulses in a pulse train in direct proportion to a small control signal; the greater the control voltage, the wider the resulting pulses become. By using a sinusoid of the desired synchronous frequency of the feed as control voltage for the PWM circuit, it is possible to

produce high-power waveform whose average voltage varies sinusoidally in a manner suitable for IMs.

The basic concepts of the PWM are illustrated by *Figure 4.8*. Basically the functionality can be described easily by a single-phase PWM inverter circuit presented in *Figure 4.9(a)* with IGBTs. The states of IGBT₁ through IGBT₄ in this circuit are controlled by the two comparators shown in *Figure 4.9(b)*.

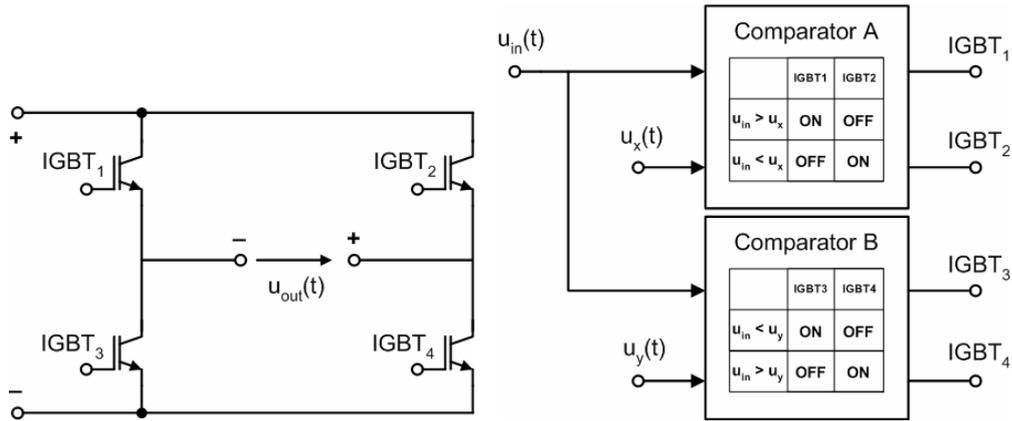


Figure 4.9(a). One-phase PWM-VSI

Figure 4.9 (b). Mechanism of PWM generation

A *comparator* is a device that compares the input voltage $u_{in}(t)$ to a reference signal and turns the IGBT switches on or off depending on the result of the test. *Comparator A* compares $u_{in}(t)$ to the reference voltage $u_x(t)$ and controls IGBT₁ and IGBT₂ based on the results of the comparison. *Comparator B* compares $u_{in}(t)$ to the reference voltage $u_y(t)$ and controls IGBT₃ and IGBT₄ based on the results of the comparison. If $u_{in}(t)$ is greater than $u_x(t)$ at any given time t , then *Comparator A* will turn on IGBT₁ and turn off IGBT₂. Otherwise it will turn on IGBT₂ and turn off IGBT₁. Similarly, if $u_{in}(t)$ is greater than $u_y(t)$ at any given time t , then *Comparator A* will turn on IGBT₃ and turn off IGBT₄. Otherwise it will turn on IGBT₄ and turn off IGBT₃. The reference voltages are shown in *Figure 4.10*. Most commonly the amplitude and frequency of reference voltages are constant and equal, so their $V_{pp_x} = V_{pp_y} = V_{pp_m}$ and $f_m = f_x = f_y$.

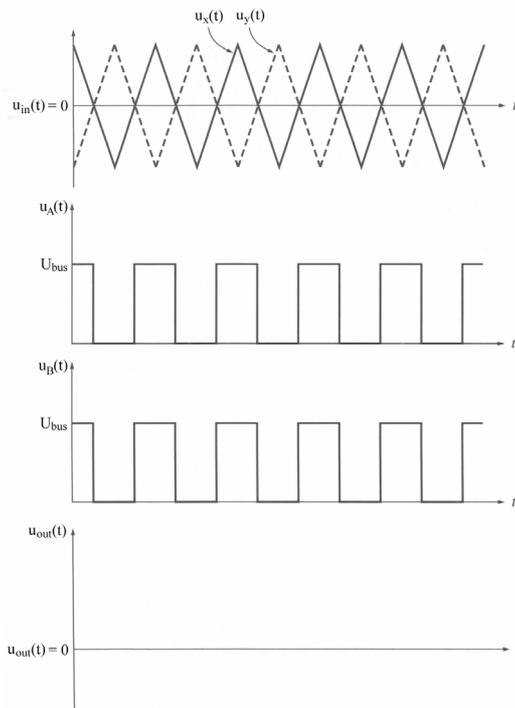


Figure 4.10. Reference signals and the outputs of the comparison in one-phase PWM generation, when $u_{in} = 0$

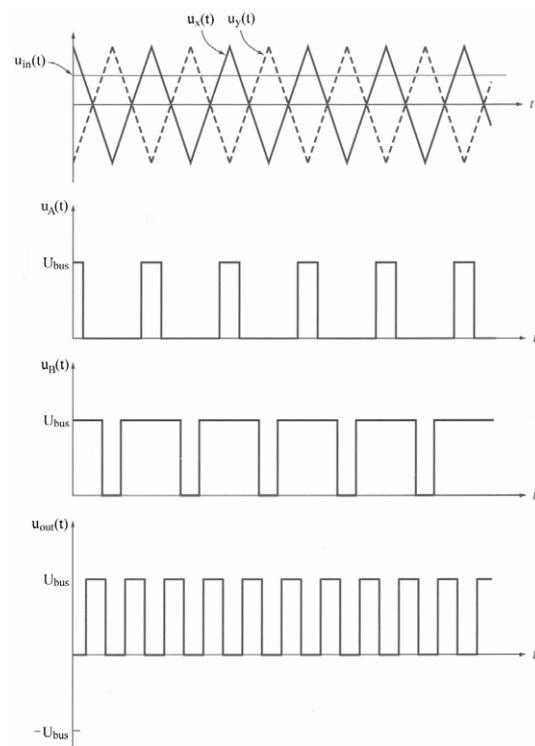


Figure 4.11. Reference signals and the outputs of the comparison in one-phase PWM generation, when $u_{in} = \frac{1}{2}V_{pp}$ of references

To understand the overall operation of this PWM inverter circuit, see what happens when different control voltages are applied. First, let assume that the control voltage is zero. Then $u_x(t)$ and $u_y(t)$ are identical, and the load voltage out of the circuit $u_{out}(t)$ is zero (see *Figure 4.10*). Next assume that a constant positive voltage with $\frac{1}{2}V_{pp}$ of the reference signals is applied to the circuit. The resulting output voltage is a train of pulses with a 50 percent duty cycle, shown in *Figure 4.11*.

Finally, assume that a sinusoidal control voltage is applied to the circuit with V_{ppin} and f_1 frequency as shown in *Figure 4.12*. The width of the resulting pulse train varies sinusoidally with the control voltage. The result is a high-power output waveform whose average voltage over any small region is directly proportional to the average voltage of the control signal in that region.

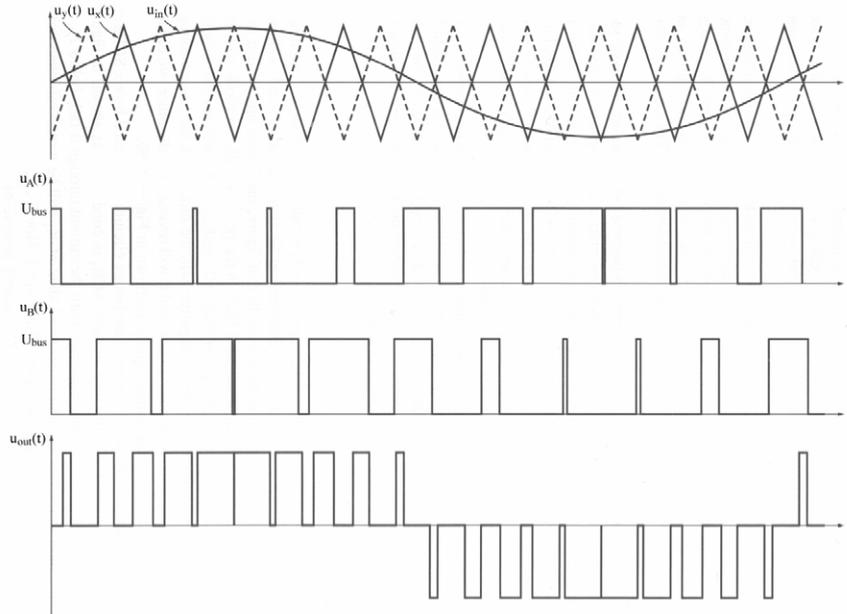


Figure 4.12. Reference signals and the outputs of the comparison, when u_{in} is sinusoidal

Moreover, the *fundamental frequency* of the output wave form is the same as the f_1 frequency of the input control voltage. It is also true, that there are harmonic components in the output voltage, but they are not usually a concern in motor-control applications. To be able to describe the properties of the modulation, the following rating values are introduced:

$$\text{Amplitude modulation value: } m_{AM} = \frac{V_{PPin}}{V_{PPm}}$$

$$\text{Frequency modulation value: } m_{FM} = \frac{f_m}{f_1}$$

With these rating values, the relationship between the command and reference signals can be investigated from the point of view of produced harmonics. In case of $m_{AM} = 0.8$ and $m_{FM} = 15$ the normalized spectrum of produced output voltage of the modulation can be seen in *Figure 4.11*.

By considering this figure the following can be concluded: The U_0^{out} peak amplitude of the fundamental frequency component is

$$U_0^{out} = \frac{m_{AM} \cdot U_{bus}}{2}, \quad \text{if } V_{PPin} \leq V_{PPm} \quad (4.1)$$

Equation (4.1) also yields, that m_{AM} must be in the range of $0 \leq m_{AM} \leq 1$ if sinusoidal output signal is needed to be generated, because of the constant U_{bus} .

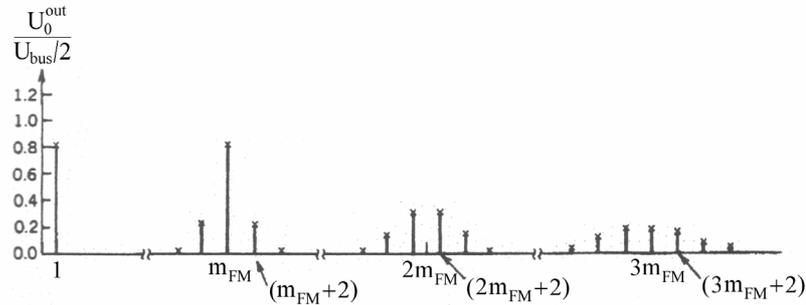


Figure 4.13. Frequency spectrum of a PWM signal with $m_{AM} = 0.8$ and $m_{FM} = 15$

In the produced voltage signal, the upper harmonics cluster in side bands around the $2m_{FM}, 3m_{FM}, \dots$, and the h^{th} harmonic component on the j^{th} m_{FM} frequency in the k^{th} side band has the frequency of

$$f_h = (j \cdot m_{FM} \pm k) \cdot f_1 \quad (4.2)$$

Based on this, if m_{FM} is chosen to be an odd natural number, then the produced voltage will be oddly symmetric and the even harmonic components are eliminated from the spectrum.

In practice, however, the harmonic components may cause additional heating in the motor being driven by the inverter, but this extra heating can be compensated for either by buying a specially designed motor or *derating* an ordinary motor, like running it at less than its full rated power.

So in the above mentioned way it can be clearly seen that with the peak-to-peak magnitude and frequency of the control voltage signal, which only theoretically exist for the digitally computed comparators, the desired voltage signal can be applied to the motor. The quality of the realization is strongly depends on the frequency of the reference signals which is most commonly 20 kHz to provide very fine modulation which is above the human hearing spectrum.

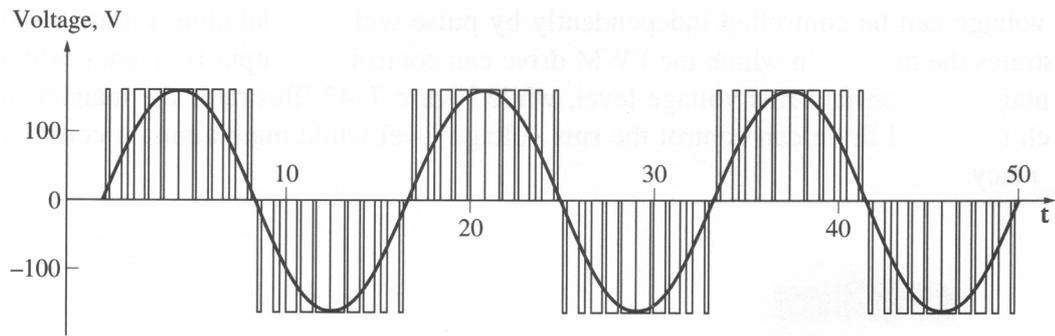


Figure 4.14(a). PWM output wave form of 60Hz, 120V_p sinusoid control signal

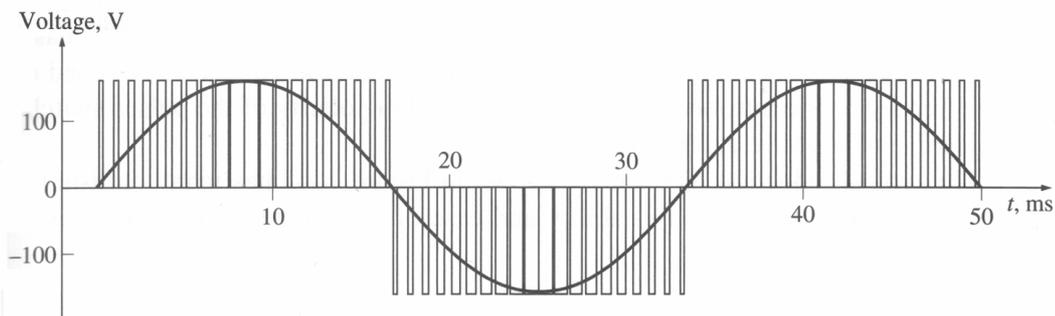


Figure 4.14(b). PWM output wave form of 30Hz, 120V_p sinusoid control signal

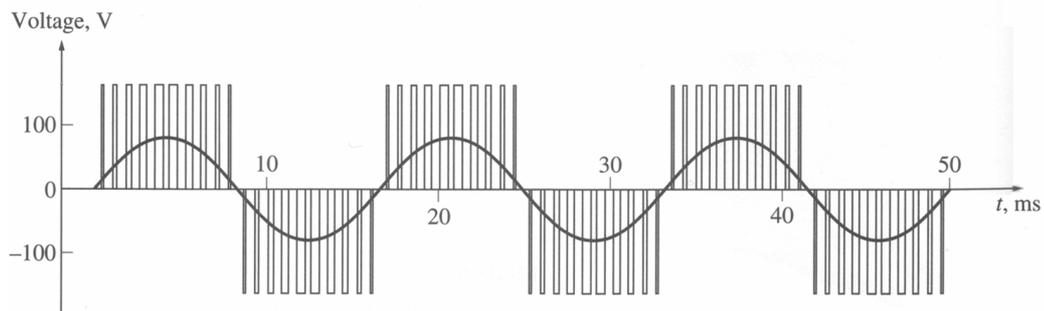


Figure 4.14(c). PWM output wave form of 60Hz, 60V_p sinusoid control signal

A complete three-phase inverter would consist of three of the single-phase inverters described above with control voltages consisting of sinusoids shifted by 120° between phases. Of course in practice, a more economical solution of three-phase compact inverters is used as it is shown in *Figure 4.7*. Frequency control in a PWM inverter of this sort is accomplished by changing the frequency of the input voltages similarly to the previous case. *Figure 4.14(a) and 4.14(b)* illustrates the manner in which the PWM drive can control the output frequency while maintaining a constant rms voltage level while *Figure 4.14(a) and 4.14(c)* illustrates how the control of the rms voltage level can be achieved while maintaining a constant frequency.

4.2.3. Different techniques of the 3-phase PWM generation on VSI-PWM inverters

Based on the previously mentioned concepts several utilizations of the PWM technique exist [17, 37]:

4.2.3.1. Synchronous symmetric PWM

In this case, m_{FM} is an odd integer number, thus the reference signals must be synchronized with the control signal. Moreover, the *synchronous symmetric* (SS) PWM has the advantage over its asymmetric cousin as it has two inactive zones of the same duration: at the beginning, and at the end of each period. This symmetry and the clever choice of m_{FM} have shown to cause fewer harmonics than in the following case. For this type of modulation, the effective value of the line voltages are $\sqrt{3/8} \cdot m_{AM} \cdot U_{bus}$.

4.2.3.2. Synchronous asymmetric PWM

The asymmetric (AS) generation of synchronous PWM is very closely related to its SS cousin except that the reference signals are saw-tooth like, so the active period is triggered only at the end of each PWM cycle [37]. Although in this case, it is much easier to produce the triggering signals of the IGBTs, the performed output will be heavily loaded with harmonics.

4.2.3.3. Asynchronous PWM

For asynchronous PWM, the value of m_{FM} is not an integer number, therefore the produced voltage signal will have much more additive frequency components than in the previous cases. This technique does not need the synchronization of the command and reference signals, thus it is used in analog logic-driven inverters, which are employed rarely today. In this case, if very large $m_{FM} \geq 21$ is used, then the harmonic components become significant, producing such large current cuffs which can easily damage the motor or the power system and overload the electric network in an industrial application.

4.2.3.4. Space vector PWM

Space vector (SV) PWM generation is very closely related to the concepts mentioned in DTC and it generates the minimum harmonic distribution in voltage and currents for the IM [43]. As it was mentioned in Section 3.2.4, for PWM-VSIs eight possible combinations of the resulted voltages can be obtained by the on and off switching of the IGBTs presented in Figure 3.12. The derived eight combinations of the switching states and the corresponding motor line-to-line and phase voltages in terms of U_{bus} are shown in Table 4.1, where 0 = off, 1 = on.

a	b	c	$U_a(U_{bus})$	$U_b(U_{bus})$	$U_c(U_{bus})$	$U_{ab}(U_{bus})$	$U_{bc}(U_{bus})$	$U_{cd}(U_{bus})$
0	0	0	0	0	0	0	0	0
0	0	1	-1/3	-1/3	2/3	0	-1	1
0	1	0	-1/3	2/3	-1/3	-1	1	0
0	1	1	-2/3	1/3	1/3	-1	0	1
1	0	0	2/3	-1/3	-1/3	1	0	-1
1	0	1	1/3	-2/3	1/3	1	-1	0
1	1	0	1/3	1/3	-2/3	0	1	-1
1	1	1	0	0	0	0	0	0

Table 4.1. Possible switching states of PWM-VSI inverters

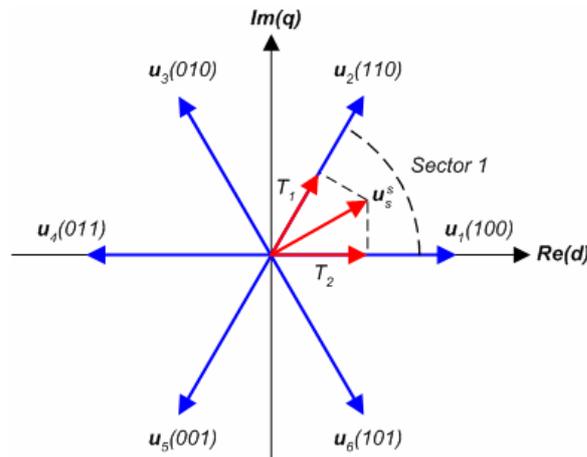


Figure 4.15. Six possible non-zero voltage vector states of SV-PWM VSIs

Mapping these voltages on the d-q complex vector plane, by performing the (3.23) Park transformation, results in the six non-zero vectors and two zero vectors (see Figure 4.15) in consistence with Section 3.2.4. However, this approach uses different sector locations, as it is presented in the figure.

The objective of the SV-PWM is to approximate the control voltage represented by the \mathbf{u}_{in} space vector with a combination of the eight possible switching patterns of the six IGBTs. One way to achieve this, requires that for any small interval of the T_{PWM} PWM period, the average inverter output be the same as the average reference voltage as shown in (4.3).

$$\frac{1}{T_{PWM}} \int_{n \cdot T_{PWM}}^{(n+1) \cdot T_{PWM}} \mathbf{u}_{out}(t) dt = \frac{1}{T_{PWM}} \cdot (T_1 \cdot \mathbf{u}_x + T_2 \cdot \mathbf{u}_{x \pm 60}), \quad n \in \mathbb{N} \quad (4.3)$$

where \mathbf{u}_x and $\mathbf{u}_{x \pm 60}$ corresponds to the voltage levels of the \mathbf{u}_x and $\mathbf{u}_{x \pm 60}$ states which are the basic space vectors of the sector containing \mathbf{u}_{in} . Note that T_1 and T_2 , are the respective durations for which switching states corresponding to \mathbf{u}_x and $\mathbf{u}_{x \pm 60}$ are applied. However, if we assume that the change in the command signal is tiny within T_{PWM} , then equation (4.3) can be approximated with (4.4).

$$\mathbf{u}_{out}(t) = \frac{1}{T_{PWM}} \cdot (T_1 \cdot \mathbf{u}_x + T_2 \cdot \mathbf{u}_{x \pm 60}), \quad T_1 + T_2 \leq T_{PWM}. \quad (4.4)$$

Therefore, it is critical that T_{PWM} be small with respect to the speed of change of \mathbf{u}_{out} . In practice, this is guaranteed by the 20 kHz modulation frequency whose period time is 50 μ sec. Equation (4.4) means that for every PWM period \mathbf{u}_{out} can be approximated by applying the switching states \mathbf{u}_x and $\mathbf{u}_{x \pm 60}$ for the T_1 and the T_2 time durations respectively. Since $T_1 + T_2 \leq T_{PWM}$, the inverter needs to be in the \mathbf{u}_7 or \mathbf{u}_8 zero states for the rest of the period, therefore for the complete space description

$$T_{PWM} \cdot \mathbf{u}_{in} = T_1 \cdot \mathbf{u}_x + T_2 \cdot \mathbf{u}_{x \pm 60} + T_0 \cdot \mathbf{u}_{7 \text{ or } 8}, \quad T_1 + T_2 + T_0 = T_{PWM}, \quad (4.5)$$

applies, which guarantees that $\mathbf{u}_{in}(t) \approx \mathbf{u}_{out}(t)$. From (4.5) the corresponding times can be computed by vector inversion like in (4.6) or by trigonometric bases like in (4.8) if the angle between \mathbf{u}_{in} and \mathbf{u}_x is α .

$$\begin{bmatrix} T_1 & T_2 \end{bmatrix}^T = T_{PWM} \cdot \begin{bmatrix} \mathbf{u}_x & \mathbf{u}_{x \pm 60} \end{bmatrix}^{-1} \times \mathbf{u}_{in} \quad (4.6)$$

$$T_1 = \sqrt{2} \cdot T_{\text{PWM}} \cdot \|\mathbf{u}_{\text{in}}\| \cdot \cos(\alpha + 30^\circ) \quad (4.7)$$

$$T_2 = \sqrt{2} \cdot T_{\text{PWM}} \cdot \|\mathbf{u}_{\text{in}}\| \cdot \cos(\alpha) \quad (4.8)$$

Depending on the specific application, the SW-PWM can be done either with (4.6) or (4.7) and (4.8). Method (4.6) is sector dependent, however the matrix inverse can be calculated off-line for each sector and obtained via a look-up table during on-line calculation, which needs less space and gives more accurate performance than the look-up table based cos and sine calculation.

This modulation technique also provides a more efficient use of the supply voltage in comparison with the other techniques and it makes possible a $m_{\text{AM}} \cdot U_{\text{bus}} / \sqrt{2}$ amplitude for u_{out} with a rms phase voltage of $m_{\text{AM}} \cdot U_{\text{bus}} / \sqrt{6}$ which is $2/\sqrt{3}$ times greater than any type of sinusoidal PWM can achieve.

Because of its easy digital computation and most optimal performance this SV-PWM method is used in most cases of vector control.

4.2.3.5. Overmodulated PWM

In case of $m_{\text{AM}} \geq 1$, the amplitude of the fundamental harmonic component can be increased above the relative value of 1, but with the cost of more upper harmonics as it is shown in *Figure 4.16*. It is also important to state, than if $m_{\text{AM}} \geq 1$ then U_0^{out} is no more changing linearly in respect to m_{AM} as it is shown in *Figure 4.17*. Because of the loss effects of high harmonics, this technique is only used for IM drives, and just only in that case, where the rated voltage of the motor is much more than $\frac{1}{2}U_{\text{bus}}$.

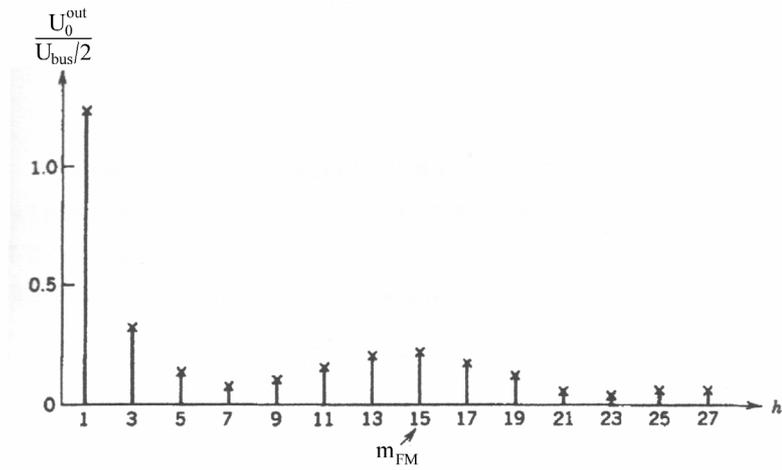


Figure 4.16. Frequency spectrum of an overmodulated PWM signal with $m_{AM} = 0.8$ and $m_{FM} = 15$

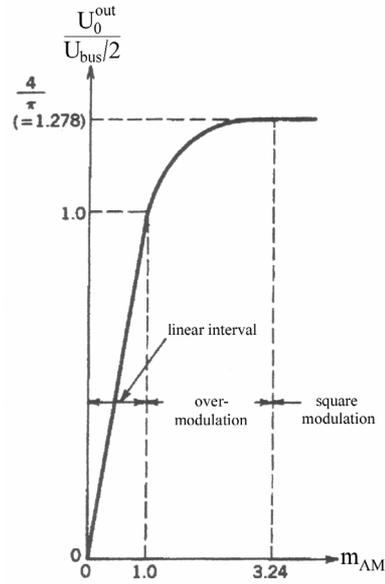


Figure 4.17. Amplitude change of the fundamental harmonic component via m_{AM}

5. The experimental laboratory drive

Based on the previously introduced theories of control methods and on the physical characteristics of the inverters such controller algorithm was designed which, by a successful implementation, would fulfill the recent expectations, for modern industrial IM drives. But before any design of an algorithm can be started, at first, the available technical equipment must be examined to know the physical environment of the implementation. Thus, in the following section, the Digital Spectrum motion control development kit powered experimental laboratory IM drive, whose heart is a TMS320F243 evaluation module (EVM), is going to be presented. This drive was self-assembled and supplemented to be able to provide speed sensorless control of an IM. In *Figure 5.1* the above mentioned drive is presented whose mechanism is explained by *Figure 5.2*.



Figure 5.1. The assembled laboratory drive

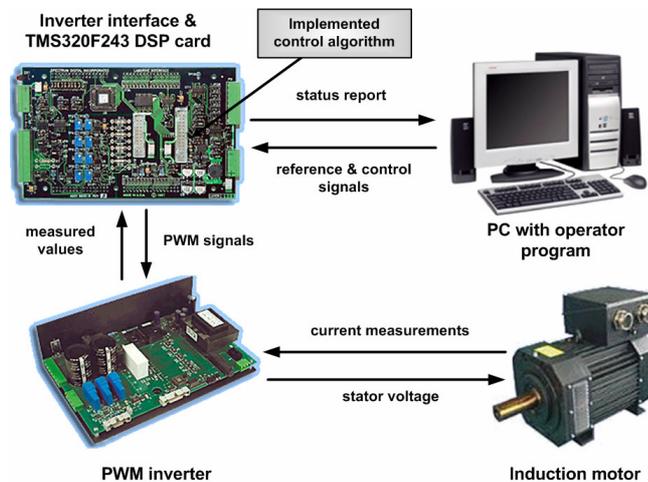


Figure 5.2. The mechanism of the considered IM drive

The drive consists of 4 main parts:

- ◆ A 3 phase, 4 poles induction motor whose control is needed to be achieved for flux and speed.
- ◆ A PWM-VSI inverter, which is the power element of the system and provides measurements on the currents and on the inverter bus voltage.
- ◆ An inverter interface card and a TMS320F243 EVM, which are connected to each other in a sandwich like design and make possible the running of implemented algorithms and through them the digital control of the whole drive based on the measured data.
- ◆ A personal computer (PC) which makes possible the debugging and monitoring of the program and also provides a user interface for operating the drive.

Each of these system parts are going to be explained in the next sections:

5.1. The induction motor

The considered 4 poles induction motor, seen in *Figure 5.2*, is a low power and low speed construction mainly used for experiments and laboratory work. Its rated power is 90W with 230/300 V and 0.59/0.49 A rated input in Y/ Δ connection. Its nominal feeding frequency is 50Hz, with 5.3% nominal slip, but of course, it can be excited on lower frequencies as well. Because of its small capabilities and its large stator winding resistance which is $38,1\Omega$ / half winding, it is only a shadow of true industrial IMs. Although, it provides a good test subject for control algorithm implementation because if fast and dynamic control of this motor can be achieved than for more powerful IMs the same algorithm with slight modifications will provide useful control solution. The whole wiring between the motor and the inverter was self-constructed to provide a connection with 230V/5A rms capability by using 1.5 type cables.

In order to successfully design a high performance controller, the parameters of this motor have been identified through measurements and test signals provided by a special program written for this task and implemented on the considered

Ladrive system. The exact procedure of this identification is going to be shown later with the obtained parameter values and with their verification as well.

To be able to complete the identification procedure and to verify the functioning of the controller a connectable encoder was built on the shaft with a DC motor (see *Figure 4.1*) to provide speed measurements with a capability of load torque control. The opto-encoder which, has a 60 slots disc fixed to the shaft of the IM, uses two pairs of 5V-powered phototransistors and LEDs to generate impulses during the rotation of the disc. Impulses are generated based on that whether there is a slot between the transistor and the LED, if it is, then the transistor is triggered by the photon stream provided energy of the LED, or in the other case, it is switched off by the absence of the light. The time base of these impulses is the 1/60 of the time period of each rotation, thus the 1/60 of the frequency of the impulse train is equal to the frequency of the rotor speed. Using two pairs of LEDs and phototransistors not only the speed but the direction of rotation can be obtained by calibrating them to generate impulse trains which differ in phase. This can be seen in *Figure 5.3*. If the leading signal is from *Sensor A* then, when the direction of rotation changes and the previously lagging *B* will have a phase shift. The 90° phase difference of these signals can be easily obtained by the use of edge sensitive circuits such as the Schmitt trigger connection. In practice, usually a third pair is used in the encoder which observes a lower region of the disc with only one slot. This is called the index slot and it is important when the incoming signals are used to obtain the position of the shaft. In high precision applications, this index signal is used for calibration, because it means the 0 angle position, and by sensing it the first time, the controller can be sure than from this point precise tracking of the angle position can be achieved.

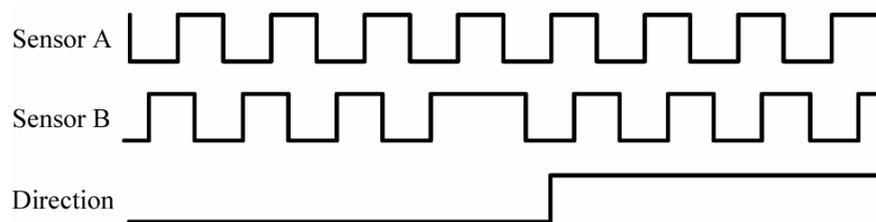


Figure 5.3. Encoder signals for speed and direction sensing

The connected DC motor, whose role is to provide load on the shaft is a 24V and 15.6W construction. Its main purpose is to test the stability and performance of the controller during the operation of the drive with applying load torque changes on the shaft. The importance of these tests will be described in the later sections.

5.2. The Spectrum Digital inverter

The direct three-phase voltage feed of the motor is generated through a Spectrum Digital PWM-VSI. This inverter is a very intelligent power system with the ability of direct digital control of the triggering signals of each built in IGBT, dynamic breaking, and several built in sensors to provide a good solution for the power element of any sensorless IM drive [24].

The inverter module is part of the Spectrum Digital Motor Development System and it is designed to be used with the TMS320F243 EVM and the Labdrive Interface card. The Labdrive interface module piggybacks the EVM and uses cables to connect to the Inverter module, so the direct control of the inverter is provided via this interface module. Because the inverter has the logic to drive 3-phase AC induction and 3-phase DC brushless motors as well, therefore the triggering signals of the built in 10A capable IGBTs, can be separately provided. Moreover, because of the hybrid design, no built in protections are available to avoid shoot through effects. This protection is provided by the interface module.

This device also contains interface circuitry which is common to most motor control systems. This means that the motor specific electronics can be isolated to one board, the inverter module.

5.2.1. Specifications of the inverter module

The board outline of the inverter is shown below in *Figure 5.4*.

The device consists of two main parts, the control logic and the power circuitry, which have separated power inputs on P1 and on P5. The power circuitry has a common design with a diode rectifier-bridge which is rated to 230/110V AC input. The output of the rectifier bridge is smoothed by a RC-low pass filter with 2 parallel connected 400V, 470 μ F electrolyte condensators. This

provides a rated bus voltage of 350VDC & 10A and through a relay, it is also able to input externally provided DC voltage on P5 for the inverter bus by disconnecting the rectifier bridge. Because of the non-controlled rectifier, only dynamic braking can be achieved by switching a braking resistor to the inverter bus to dissipate the backfed energy coming from the motor. This is done through an optically isolated transistor.

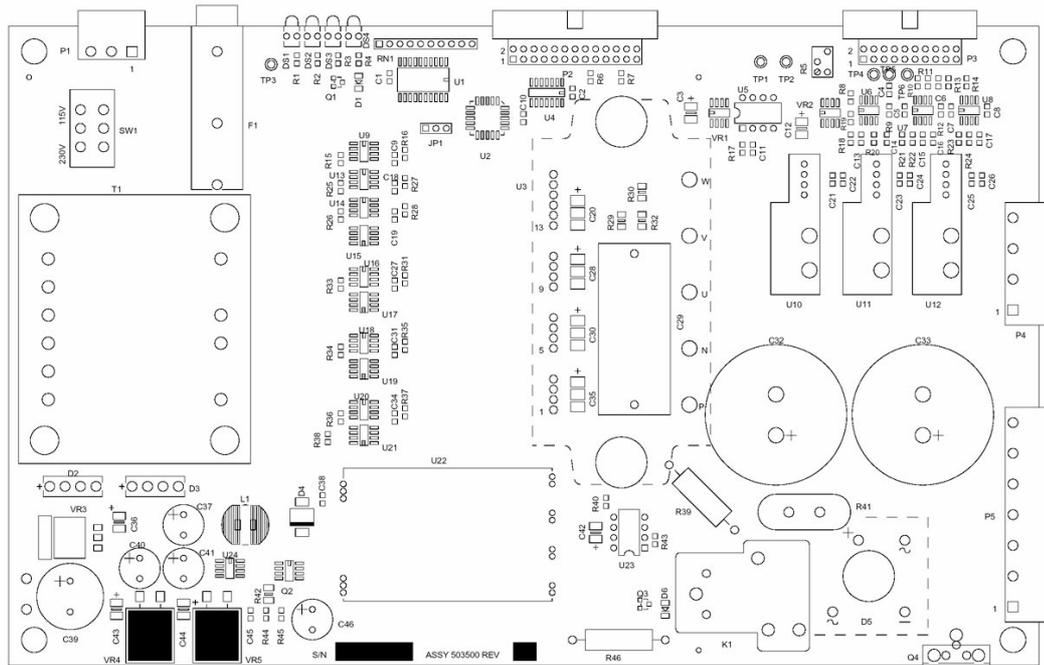


Figure 5.4. Board outline of the Digital Spectrum PWM-VSI

The dead time of the used IGBTs are 2 μ sec but because the high and the low side drivers are optically isolated, thus with the 1 μ sec transient time of the optoisolation, the triggering signals must be interleaved with a 3 μ sec deadband.

5.2.2. Capabilities of the inverter module

The control logic of the inverter, which has an onboard separated power supply, provides intelligent control of this power element. It is connected through the P2 and P3 connections to the interface module on which communication is provided on analog signals. This logic makes up several important features of the inverter:

As it was mentioned, the inverter is capable of dynamic braking through an externally provided resistor on the P5 connector. This resistor is switched on the

bus with a 30A IGBT, triggered by an optically isolated driver. Because in the case of this type of breaking there is a rapid voltage rise on bus with high current, a large power capable consumer is needed. For this purpose a 100W bulb is connected to the appropriate pins with a serial connected high power resistor.

The IGBT bridge side of the inverter bus is separated through a relay, which is controlled through a digital logic implemented as a state machine. This logic enables the bus voltage for the IGBTs in case of faultless operation, and if the appropriate signal is provided from the inverter card. If this signal breaks, the logic assumes that the microcontroller missed its operation and therefore it disables the bus, preventing the damage of the drive. The same functionality is provided if any kind of built in protection is activated.

The PWM signal control is made possible through digital connection with the inverter interface card. On this connection, the 6 triggering signals for the IGBTs, which are 0-4V digital signals, are given by the interface module. Unfortunately, the inverter logic provides no protection on the PWM signals, so by incautious usage short circuit can be easily generated. To avoid this, the power connections were fused by me.

The control logic can also be powered through the interface connections. Moreover, in case of any sensed fault, by raising the Fault signal on the connection line, the microcontroller can be notified about the error in the operation.

The drive has several test points to make the measurements easy and it also has onboard LEDs to signal that the control power and the bus voltage is on or the event of braking or inverter fault has occurred.

5.2.3. Sensors

The inverter has 3 Hall effect current sensors with ± 10 A sensing range on each of the a, b, c phases of the motor feed. The outputs of these sensors are ± 4 V analog signals which are filtered by signal pole filters with a cutoff frequency of 33.8Khz.

It also contains an optically isolated bus voltage sense whose signal is divided by resistors with 1766 division rate. This divided voltage is then applied to the input of an optically isolated amplifier which has a nominal gain of 8. The maximum input voltage of the isolation amplifier is 200mV, which gives a maximum bus voltage of 353V_{DC}. The amplifier stage also converts the differential output of the isolated amplifier to single ended. This amplifier stage has a gain of 1 and incorporates a single pole filter with a cutoff frequency of 33.8 Khz. A second amplifier is also provided with an adjustable gain through a potentiometer, to specify the desired sensitivity range of the voltage sense.

5.2.4. Built in protections

The module contains internal drive and protection circuitry. Protection is provided for under voltage, lockout, over current, over temperature, and shoot-through. The fault pin in the interface card connection is used to detect the type of fault. In case of over current and shoot through faults, the fault signal is pulled low for a short time and for over temperature, the fault line is low until the unit cools. Besides the signaling of the fault, the logic does nothing to avoid further faulty operation. This is done by the interface card.

5.3. The Inverter Interface Card

The next component in the control chain (*Figure 5.2*) is the Labdrive interface card which is the bridge between the microcontroller, the inverter, and other external sensors and also supports the control of AC induction, DC brushless, and switched reluctance motors [23]. The main purpose of this element is the signal conditioning and protection, therefore it contains interface circuitry that is common in most motor control environments such as inputs for optical encoders, limit switches, and control signals. There are also two headers for an optional piggy back board for additional user circuitry, for instance a resolver interface.

Two cables are used to interface to the previously mentioned inverter module. One cable drives the PWM's and digital signals, while the second cable interfaces to the analog sensors of the inverter.

5.3.1. Specifications of the interface module

A board outline of the Labdrive interface module is shown below in *Figure 5.5*.

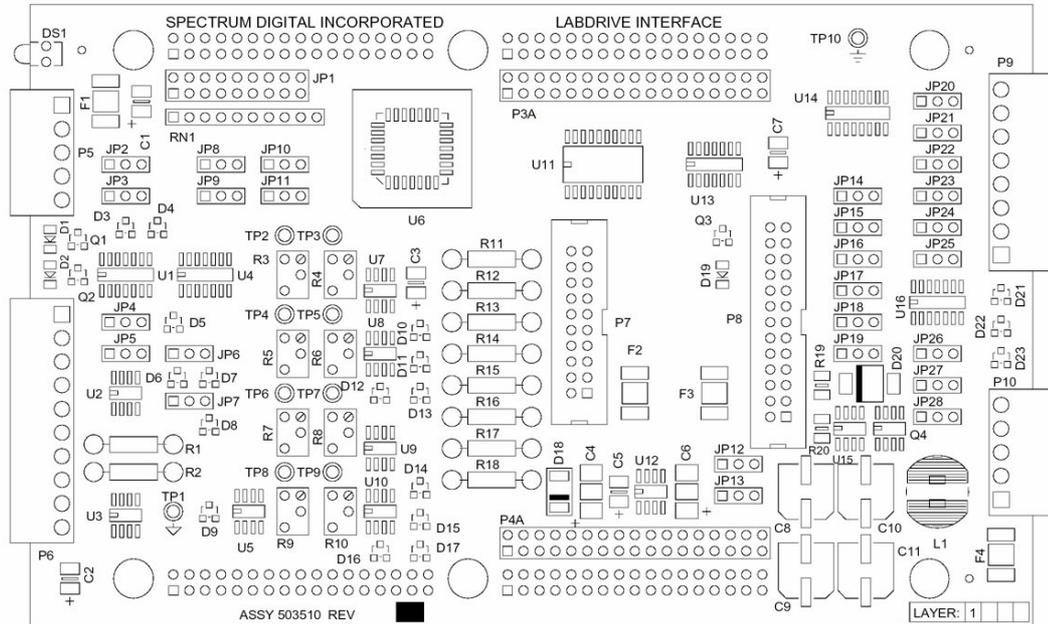


Figure 5.5. Board outline of the Labdrive interface module

This board is directly connected to the TMS320F243 EVM and as an extension provides additional, 28 jumper-defined capabilities to the EVM to make it able to fulfill the control of the drive. Moreover, each of the incoming signals is limited by the card and several of them are converted to make the EVM able to use them for further processing. The following signal conditionings and connections are important to mention:

5.3.2. I/O signal conditioning

The card provides an optical encoder interface with differential A, B, and an Index channel. Additionally, it is also able to receive single ended signals of the encoder and these signals can be mapped to the appropriate input pins of the EVM. The card also contains a single-ended Tachometer input, which can be also used for speed measurements.

The 3 inputs of the Hall effect current sensors and the bus voltage sense coming from the inverter can be conditioned through a potentiometer controlled offset and an adjustable gain to transform the received $\pm 4V$ signals into the 0-5V

input range of the ADC of the EVM. This is done through the presented simple connection with the use of an operational amplifier in *Figure 5.6*. The correct configuration of the potentiometers was adjusted manually in experiments to reach the most optimal current measurements. Because only 1.45 gain and 1.34 V offset could be achieved by the implemented circuits, and because of the low power capabilities of the motor, therefore the sensed 0-5 A ($\pm 2V$) range was calibrated to the 0-5V input range of the ADC. There is also a built in filter on each channel with a cut off frequency of 32 kHz.

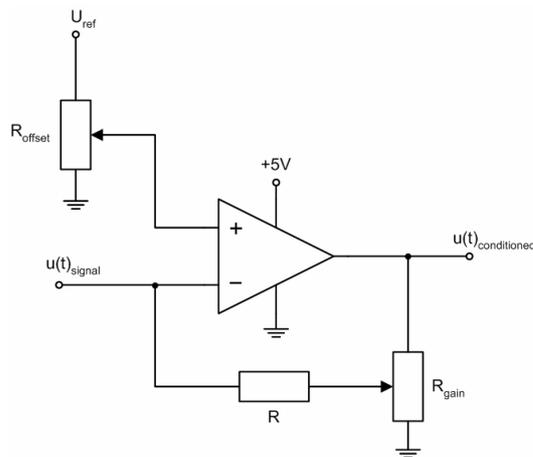


Figure 5.6. Logical structure of the signal calibration circuitry in the interface module

Moreover, the card provides 2 external limit-switches inputs, 1 differential/single-ended input for control voltage, and 1 driver enable input to make possible the control by external reference signals generated through for example potentiometer-divided voltage or limiters. In this way, with a simple circuit, the reference for speed signal can be externally generated. These connections are protected and have jumper defined termination polarities as well. The board also has unipolar analog outputs in 0-5V range for further purposes.

5.3.3. Inverter Digital Interface

This interface directly communicates with the inverter and its main purpose is to provide the correct and faultless control flow between the EVM and the VSI, with also paying attention to the protection of the drive. To be able to fulfill this, the board provides the EVM generated 9 PWM drive outputs to the 6 triggering channels of the inverter. Each of these incoming signals can also be manually

disabled by jumpers and even remapped. The 9 signals are generated by the EVM because not only IM control but also DC motor control capability is provided, thus the appropriate channels has to mapped to the 6 inverter connections. The polarity of these channels can be also specified by jumpers, which are needed to be oppositely configured for the high and low side of the inverter bridge. In any case of phase fault, the card can be configured to disable the PWM inputs of the inverter to avoid shoot-throughs.

The card is also able to provide the break, driver enable, and driver reset signals for the inverter and through them the control of the built-in logic. It also receives the fault signal of the inverter which is pulled down if any kind of protection is activated. If this happens, the card disables all the PWM outputs and the drive enable signal, which disconnects the inverter bus from the power source. In this way, it ensures that in case of faulty operation the drive will not be damaged. Additionally, the connection with the EVM is provided through four 32bit wide channels as I/O, address, data, and control expansion busses.

5.3.4. Shield of the drive

As it was previously mentioned, the interface card is very important in the control chain, because it consists of those watchdog and protective circuits that make possible the protection of the EVM and the inverter side as well. In any case of over voltage on any I/O channels or fault detection from the inverter, the two elements are totally separated and protected until the hard reset of the EVM.

5.4. The TMS320F243 Evaluation Module

This microcontroller-containing board is the heart and soul of the whole system with lots of dedicated hardware providing a very effective way of motion control. Its speed performance makes possible the implementation of an algorithm with a large computational load and it is also capable of generating the triggering signals of the IGBTs of the inverter directly by the use of SV-PWM. In the following sections, the main capabilities of this card are going to be presented.

5.4.1. Specifications of the EVM

The family of '24x DSP fixed point controllers is designed to meet the needs of control-based applications. By integrating the high performance of a DSP core and the on-chip peripherals of a microcontroller into a single-chip solution, the '24x series yields a device that is an affordable alternative to traditional microcontroller units like the PIC and expensive multichip designs [37]. With 20 million instructions per second (MIPS), the F243 DSP controller offers significant performance over traditional 16-bit microcontrollers and microprocessors. The 16-bit fixed-point DSP core provides designers a digital solution that does not sacrifice the precision and performance of their systems. In fact, system performance can even be enhanced through the use of advanced control algorithms such as adaptive control, Kalman filtering, and vector control. To achieve this, the given '24x architecture is suited for processing control algorithms. It uses a 16-bit word length along with 32-bit registers for storing intermediate results, and it has two hardware shifters available to scale numbers independently of the CPU. This combination minimizes quantization and truncation errors, and increases processing power. Therefore, The F243 offers reliability and programmability, and as a digital microcontroller, in contrast to analog control systems, is not effected by performance degradation due to aging, component tolerance, and drift.

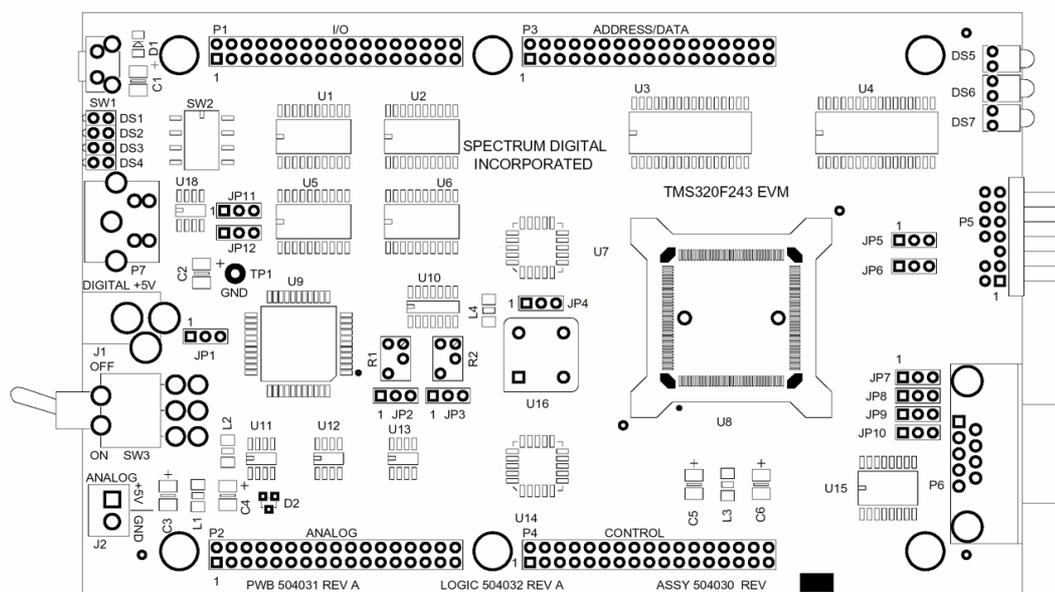


Figure 5.7. Board layout of the TMS320F243EVM

The instruction set of these DSP controllers, which incorporates both signal processing instructions and general-purpose control functions, coupled with the extensive development support and programming environment available for the '24x devices by *Texas Instruments* (TI), which greatly reduce development time.

The single DSP chip is supplemented by a peripheral board, and the hardwired connection of these elements produces the TMS320F243 EVM, which board layout is presented in *Figure 5.7*.

5.4.2. Considerations of controller implementation

It can be seen from the specifications of this device, that it offers a great possibility to realize algorithms with even high computational load. For speed sensorless operation, the incoming interface card conditioned current and bus voltage signals from the inverter must be digitized and their values obtained for the control algorithm. This task can be handled with some carefulness by a built in ADC converter. It is an also important task to realize the control voltage signal, which can be done through an *event manager* (EVM2) module. For communication with the monitoring PC, a wide range of possibilities are also presented (SPI, SCI, CAN). Of course all of these features have their specific way of use and the advantages of each of them are always followed by some stalking drawbacks. Therefore, the above mentioned important tasks and their way of implementation on the TMS320F243 is going to be briefly explained in the point of view of their applicability for the speed sensorless motion control problem at hand. The description of the hardware is also needed to clarify the implemented algorithm and to make the produced program more understandable.

5.4.3. Chip of many things

The logical structure of the EVM is presented in *Figure 5.8*, which elements and the integrated devices of DSP are going to be described in the following parts.

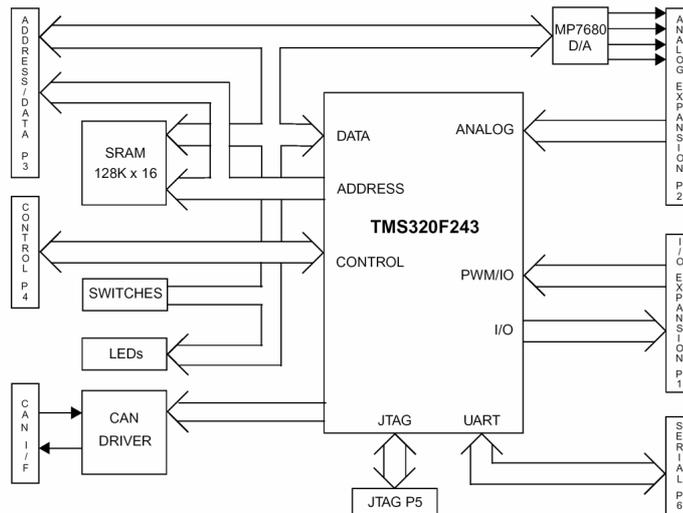


Figure 5.8. Logical structure of the TMS320F243EVM

5.4.3.1. Speed and memory

As it was mentioned, the F243 is a 16 bit address and data wide fixed point microcontroller which is also able to operate in microcomputer mode if the built-in 8K flash ROM of the chip is activated. Because for motion control, purely a fast algorithm running is needed which consists of a linear control cycle, thus only the microcontroller operation is considered. The design of the DSP processors is based on the Harvard architecture, which in contrast with the Neumann architecture of commercial PCs, has separated program, data, and I/O space memory. This structure is important, because mainly very simple programs are running on these devices, which prefer fast execution instead of easy programming capability. With this design, the fetching of instructions from the program space and the manipulation of the data can be done separately and parallel, which allows making very sophisticated pipeline structures with a possibility of 8 parallel multiplications, like in the case of some highly advanced DSPs. The F243 also profits from this structure and even if its speed only 20Mhz, which seems to be very small in comparison to today's 2-3Ghz processors of the PCs, it can totally use fully this speed for computation. This is also supported by the RISC instruction set of the DSPs, which in contrast with the CISC approach of the PCs, makes possible the 1 instruction / 1 machine cycle for any kind of operation. This makes the whole algorithm more predictable and faster, of course with a slightly more complicated programming.

In the case of the F243, 64K word of program, data, and I/O memory is presented whose configuration can be seen in *Figure 5.9*. As it is shown, the registers of the processor are available form the data space, while the input and output communication is restricted to I/O space with the control of specific data space registers. Program space is inhabited by the user code with the interrupt vector table which describes the code segment of the event specific service routines. This structure provides enough computational power to realize complicated algorithms.

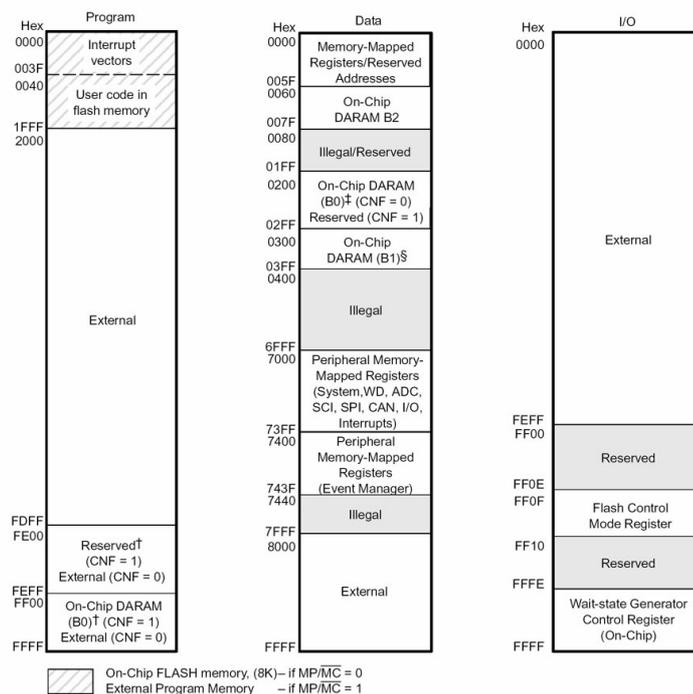


Figure 5.9. Memory architecture of the TMS320F243 DSP

5.4.3.2. Interrupts and peripherals

The '24x CPU supports one nonmaskable interrupt (NMI) and six maskable prioritized interrupt requests. The '24x devices have many peripherals, and each peripheral is capable of generating one or more interrupts in response to many events. Because the 'C24x CPU does not have sufficient capacity to handle all peripheral interrupt requests, a centralized interrupt controller (PIE) seen in *Figure 5.10* is required to arbitrate the interrupt requests from all the different sources.

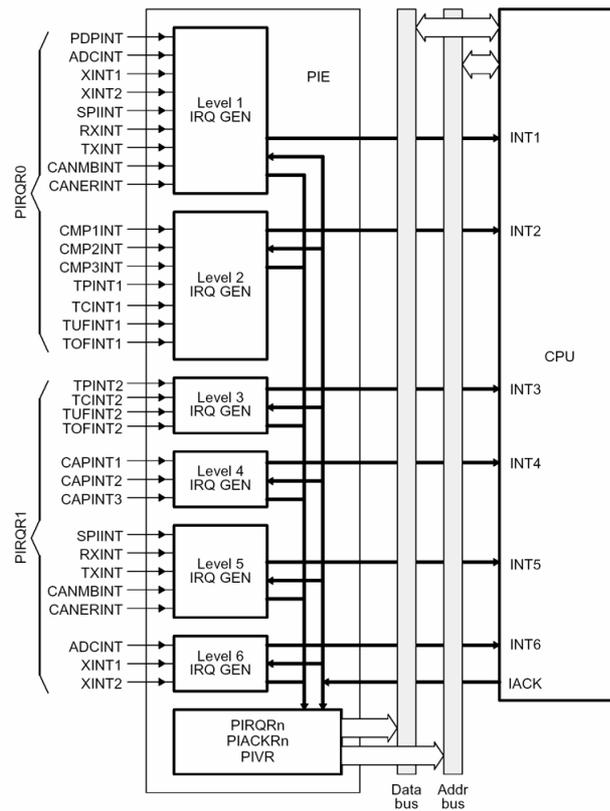


Figure 5.10. Peripheral interrupt controller of the TMS320F243 DSP

As it can be seen, the number of available interrupt requests is expanded by having two levels of hierarchy in both the interrupt request/acknowledge hardware and in the interrupt service routine software. In any case of interrupt event, the specific low group signals to the main interrupt handling logic, that a INTx level interrupt is pending. In response, the higher level logic decides which INTx interrupt is going to be serviced, and if the processor core acknowledges this, the current running code is interrupted and from the vector table defined specific program address, the first instruction is fetched. In this way, the interrupt handling routine only knows that a INTx group interrupt generated a call. To be able to obtain which of the interrupts in this group is active, the PIE register of the logic must be read. After finding out which pending interrupt in this group has the highest priority, the appropriate service routine can be called from the interrupt handling method. At each level, every INT can be masked except some global interrupts, which are responsible for the faultless operation. In this way, several events can be serviced through interrupts, which provides a very good monitoring

feature of the device. This is crucial for motion control, because in the case of fault events or reference changes, the drive has to react instantly.

5.4.3.3. Digital I/O pins

The device has multiplexable I/O pins to switch between the main functions and the channels provided by the extension busses. Through these pins the interface conditioned signals can be received and output functions, such as the control of the inverter enable signal, can be provided.

By the use of data space registers, the functionalities of the processor pins are defined with the data direction as well. This feature makes possible the use of the interface module and therefore it is crucial to provide the implementation of the IM drive. I/O capabilities also enhanced with a MP7680 four channel digital-to-analog converter.

5.4.3.4. Event manager (EVM2)

The EVM2 module provides a broad range of functions and features that are particularly useful in motion control and motor control applications. The block main functionalities are described as follows:

5.4.3.4.1. Timers

The module consists of two general purpose (GP) timers which can be used as independent time bases in applications such as:

- ◆ The generation of sampling period in a control system.
- ◆ Providing a time base for the operation of the integrated quadrature encoder pulse circuit (QEP) and the capture units.
- ◆ Also providing a time base for the operation of the integrated compare units and the associated PWM circuits to generate PWM outputs.

These timers are 16-bit up- and up/down-counters with programmable period and a variable counting speed which can be the scaled CPU speed or external impulsive signal. The block diagram of the timers is given on *Figure 5.11*.

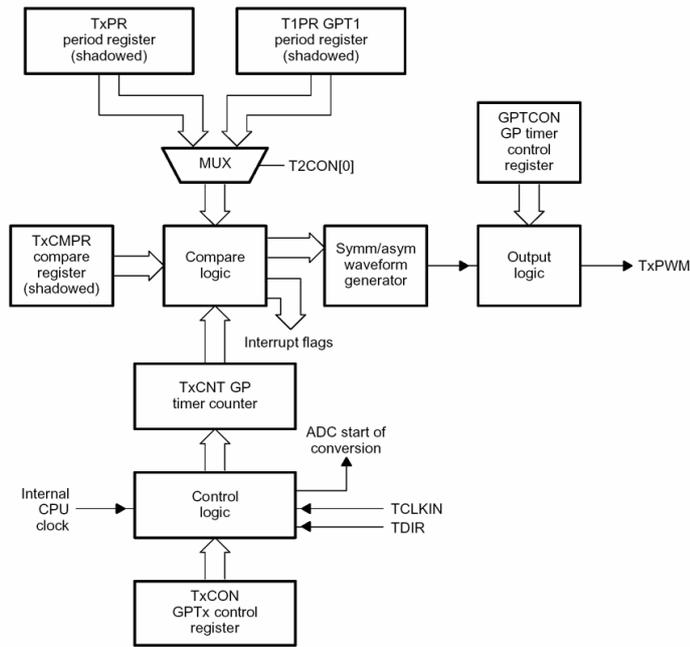


Figure 5.11. Block diagram of the implemented GP timers

Each timer also has a compare register which generates an interrupt if the counting value is equal to it and also at each underflow, overflow or period hit event the same interrupt generating feature is provided. All the main control registers can be shadowed, so in case of any functionality change, the previous operation will be completed before the new one. The two counters can also be synchronized with each other, and the start of an A/D conversion can also be generated by the reach of a specific value.

The timers have the following operation modes:

- ◆ Continuous up counting, where as it is shown in *Figure 5.12(a)* the timer is continuously counting up till it reaches the period register value and then starting the counting from zero again.
- ◆ Directional Up/Down-Counting Mode, where as it is shown in *Figure 5.12(b)* the timer counting based on the direction flag to count up or down till it reaches the period or 0, when the counting continues from 0 or the period value.
- ◆ Continuous Up/Down-Counting Mode, where as it is shown in *Figure 5.12(c)* the timer is counting upward from 0 to the period value and then downward till it reaches zero again. This is repeated continuously.

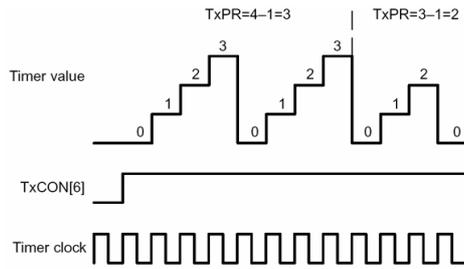


Figure 5.12(a). Continuous up counting mode of the timers

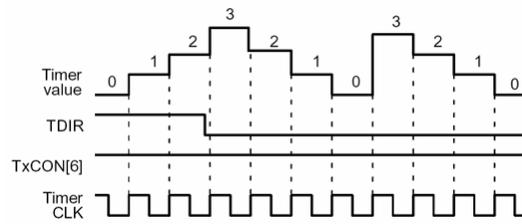


Figure 5.12 (b). Directional up-down counting mode of the timers

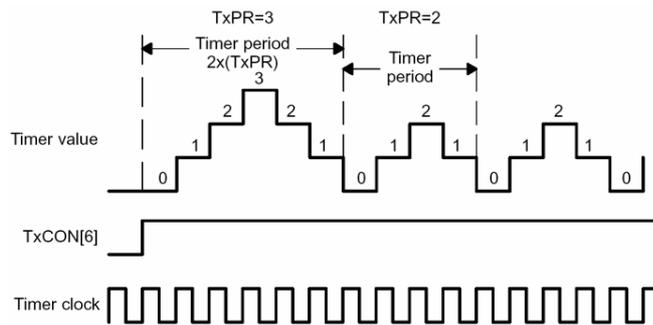


Figure 5.12 (c). Continuous up-down counting mode of the timers

While the continuous up counting can be used for several time base measurements, the other modes are important mainly for PWM generation.

5.4.3.4.2. PWM generation

The hardware-supported PWM generation feature of this device is extremely crucial for the inverter control. It provides several ways to realize the previously mentioned PWM techniques, even the SV-PWM with a dead time compensating ability, which makes the implementation of vector control very efficient.

With the use of the built in compare units of the GP timer 1, SA/SS type PWM triggering signals can be obtained by ease. Because 3 registers are associated with the compare unit structure presented in *Figure 5.13*. All the 6 PWM triggering signals can be generated. The compare units are directly connected to the PWM circuit, which generates a transition on the appropriate outputs towards the inverter, when any counter-hit on the compare registers occurs. Even interrupts can be connected to these events.

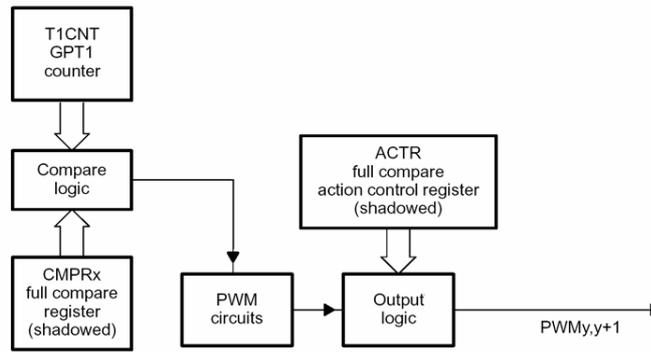


Figure 5.13. Compare unit structure of GP timer 1

The block diagram of the PWM circuit is shown in *Figure 5.14*. The PWM circuit associated with the compare units, makes possible the generation of six PWM output channels with programmable deadband and output polarity. The unit is capable of on-the-fly change of PWM carrier frequency and pulse width, however, because these registers can be shadowed it also supports the period restricted changes as well.

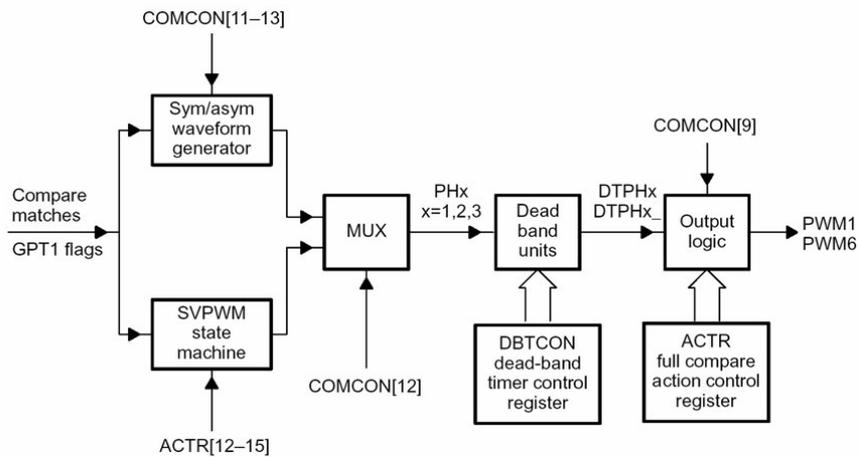


Figure 5.14. Block diagram of the PWM circuit

If the generation of AS-PWM is needed then by configuring the GP1 timer to continuous up counting with a period equal to the time period of the modulation and by setting up the compare units to the software determined values, the AS PWM can be achieved through the PWM circuitry that sets up the triggering signals based on the compare register hits. This is presented in *Figure 5.15*.

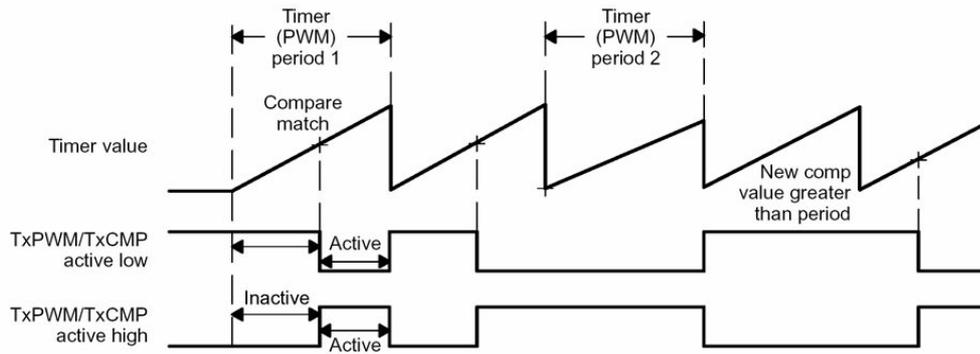


Figure 5.15. Asymmetric synchronous PWM generation with the EVM2

In case of SS PWM, the previous hardware elements can be used, but the timer is calibrated to continuous up-down counting which results the triggering signals seen in *Figure 5.16* to realize the symmetric PWM.

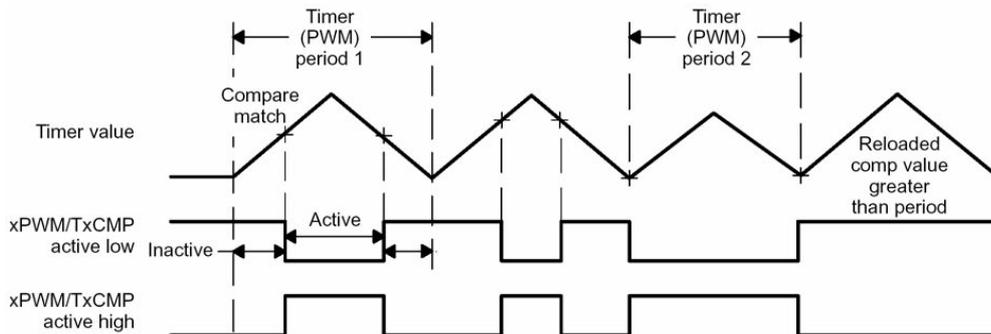


Figure 5.16. Symmetric synchronous PWM generation with the EVM2

SV-PWM can be also generated, but for this dedicated hardware logic is implemented as seen in *Figure 5.14*. The GP1 timer is calibrated to continuous up-down counting with a period match equal to the PWM period. This provides the time base. Two of the compare registers are used to obtain the transitions at $\frac{1}{2}T_1$ and $\frac{1}{2}(T_1 + T_2)$, because the switching between the base vectors is done symmetrically to reduce the unwanted upper harmonics (see *Figure 5.17*). The hardware also needs the index of one of the base vectors, which are defined by the sector of the needed output stator voltage vector. The calculation of T_1 and T_2 is done through one of the presented methods of *Section 4.2.3.4* by a software implemented algorithm. Because for correct functionality, only the T_1 and T_2 time bases have to be calculated, this hardware element provides a very fast and reliable realization of the desired output triggering signal, not mentioning that the

SV-PWM is the most efficient and low computation needed method to provide the voltage feed of the IM.

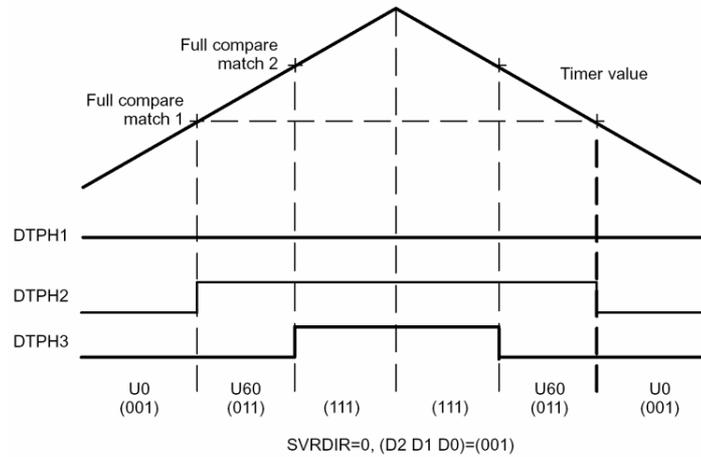


Figure 5.17. Space vector PWM generation with the EVM2

5.4.3.4.3. Deadband unit

The deadband unit is very important to provide the dead time in the triggering signals to achieve faultless operation of the IGBTs. It is implemented with three 4-bit down counting timers which knock out the active triggering at the transitions in each PWM pair. The clock of these timers is the scalable signal of the CPU clock, therefore the minimum dead-band duration is one device clock cycle, and the maximum can even be defined as 96μsec.

5.4.3.4.4. Capture units

Capture units enable logging of transitions on capture input pins. There are three capture units in the F243 and they can be used mainly in motion control applications to process the signals of encoders. Their mechanism, represented by *Figure 5.18*, is the following: by choosing one of the timers as a time base, whenever a transition happens on the associated input pin, the counter value of the input pin, the counter value of the timer is stored in a 2 level deep 16 bit wide FIFO. In this way if the output of an encoder is connected to these pins through the interface card and the associated timer is in continuous up-counting mode, then by the difference of the stored time stamps in the FIFOs, the exact elapsed time, for example between two rising edges of the incoming signal can be calculated, which will be proportional to the

frequency of the shaft, based on the resolution of the encoder. The input pins are Schmitt-triggered and they are synchronized with the CPU clock, therefore in order for a transition to be captured, the input must hold at its current level to meet the two rising edges of the device clock. Moreover, the type of detected transition can be defined to one of the followings: rising edge, falling edge, or both edges. To the defined transition, interrupt generation can be also connected.

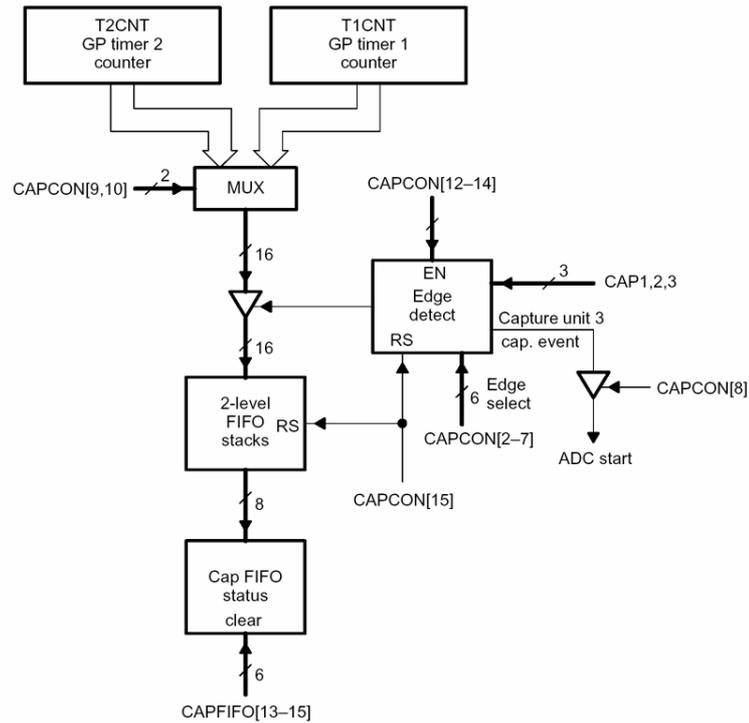


Figure 5.18. Capture unit of the EVM2

With three of these capture units, a bidirectional encoder with index slots can be connected to the drive and in this way, even position-dependent control of the rotor can be achieved. For one, this feature seems to be unnecessary for speed sensorless design, although for the well-functioning of such a drive, firstly the identification of the motor must be completed, which needs speed measurements. Therefore, during any controller design process, this ability is heavily needed.

5.4.3.4.5. Quadrature Encoder Pulse (QEP) Circuit

The F243 EVM also offers a different way of processing the signals of a position encoder. This is provided by the built-in QEP circuit presented in Figure 5.19. When this feature is enabled, it decodes and counts the input pulses from the

encoder on the appropriate input pins. The time base for the QEP circuit is provided by GP timer 2, which must be put in directional up/down-count mode, with the QEP circuit as the clock source, for correct operation. The direction detection logic of the QEP circuit in the EV2 module determines which one of the A or B sequences leads in phase. It then generates a direction signal as the direction input to GP timer 2. Thus, the timer counts up if signal A is the leading sequence, and counts down if B leads in phase.

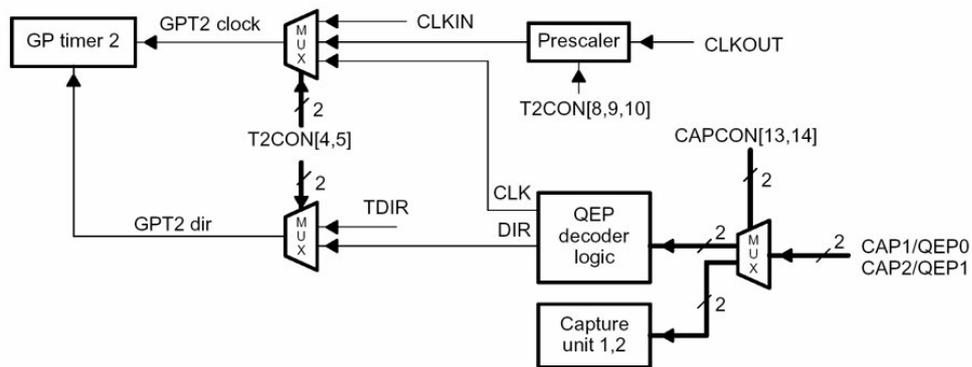


Figure 5.19. QEP circuit of the EVM2

In this way, the fast tracking of angle position can be achieved, although the speed of the rotor can not be obtained, because the elapsed time between the counts is not stored. However, this type of tracking is enough for step like operation of the IM, which is needed in high precision fields.

5.4.3.4.6. Analog to digital converter (ADC)

The F243 has a built-in 600ns, 10-bit converter with switched capacitor string providing an inherent sample-and-hold function to digitally obtain the analog signals, such as the inverter provided current and bus voltage measurements for the use of the implemented control algorithm. However, because of various synchronization delays, a single conversion takes about 1 μ s, so for the four incoming signals, minimum 4 μ s delay in the control loop has to be provided.

The converter has 8 analog inputs and it is also capable to almost simultaneous conversion on paired input channels. The two consecutive conversions are performed 850ns apart, for a total simultaneous dual-conversion time of 1700ns. This operation is called pseudo-dual ADC.

The conversions can be started by the software, an internal event, and/or an external event and the results are placed in two-level-deep FIFOs that contain the digital values of completed conversions. The mechanism of the conversion is based on successive approximation which means that the bits of the result are obtained in MSB-LSB direction with comparing the analog signal to the represented values of the bits to decide whether it is less or greater than that. The result of the conversion is approximated by the following equation:

$$\text{Digital result} = 1023 \cdot \frac{U_{\text{ADC_IN}} - U_{\text{REF_LOW}}}{U_{\text{REF_HIGH}} - U_{\text{REF_LOW}}}, \quad (5.1)$$

where $0 \leq U_{\text{REF_LOW}} \leq U_{\text{ADC_IN}} \leq U_{\text{REF_HIGH}}$, and as it can be seen, the reference signals can be arbitrarily changed to obtain the desired sensitivity on the specified amplitude region. At the end of each conversion, interrupt notification can be requested, but because it would require much time, these actions are serviced after each other.

This type of fast conversion gives the ability to quickly fetch the required data in each control cycle, which is very important for the design of high performance applications that needs a small control loop time.

5.4.3.5. Communication interface

The communication possibilities are also significant if the drive wanted to be controlled from a PC or integrated into a large industrial control chain, where incoming reference signals and the outgoing status report of the drive are provided through this interface. The F243 EVM offers three possibilities to solve these tasks and make the drive able to function as an intelligent system.

5.4.3.5.1. Serial Communications Interface (SCI)

The SCI offers the universal asynchronous receiver/transmitter (UART) communications mode for interfacing with many popular peripherals which uses the standard NRZ (non-return-to-zero) format through the serial port of the device. The asynchronous mode requires two lines to interface with many standard devices such as terminals and printers that use RS-232-C formats. The

SCI's receiver and transmitter are double-buffered, and each has its own separate enable and interrupt bits, thus both may be operated independently or simultaneously in full-duplex mode. To ensure data integrity, the SCI checks received data for break detection, parity, overrun, and framing errors. The bit rate (baud) is programmable to over 65,000 different speeds through a 16-bit baud-select register with a maximum of 38400.

With the help of this interface the well known RS-232 protocol-based communication with a PC can be achieved to provide the reference signal changes and control of the drive on-line, which is needed to provide the ability to integrate the produced IM drive into larger systems.

5.4.3.5.2. Serial Peripheral Interface (SPI)

The SPI is another communication mode provided through the serial port, but in contrast with the previously mentioned SCI, it is a synchronous type of serial I/O communication. It allows a serial bit stream of programmed length (one to sixteen bits) to be shifted into and out of the device at a programmed bit-transfer rate. The SPI is normally used for communications between the DSP controller and external peripherals or another controller; therefore it has a master-slave type of protocol. Typical applications of it include external I/O or peripheral expansion via devices such as shift registers, display drivers, and ADCs. Therefore, it is useful if other equipments have to be connected to the controller to provide the operation.

5.4.3.5.3. Controller are network interface (CAN)

Today, the CAN has a growing importance not only as the protocol of distributed control system in the industry but also as a most widely used communication method between the subsystems of cars and airplanes.

The CAN uses a serial multimaster communication protocol that efficiently supports distributed real-time control with a very high level of data integrity, and communication speeds of up to 1 Mbps. The CAN bus is ideal for applications operating in noisy and harsh environments, such as in the automotive and other industrial fields that require reliable communication. Prioritized messages of up to

eight bytes in data length can be sent on a multimaster serial bus using an arbitration protocol and an error-detection mechanism for a high level of data integrity. Thus, in a real life application this type of communication should be used to interface other control elements for example in an industrial control chain. However, for a stand alone drive and for simplicity, instead of this feature, only the use of the SCI is satisfying.

5.4.3.6. Watchdog

The watchdog (WD) timer periphery monitors software and hardware operations, and implements system reset functions upon CPU disruption. If the software goes into an improper loop, or if the CPU becomes temporarily disrupted, the WD timer overflows to assert a system reset and in this way prevents the continuing of the faulty operation.

This feature is provided through a 8-bit counter that generates a system reset upon overflow and it is fed by a 6-bit free-running counter to provide the appropriate prescaling. To avoid the overflow of the counter, it needs to be reseted with a specific reset key register that clears the WD counter when the correct combination of values are written in it, and it generates a system reset in case of an incorrect value.

5.4.4. Heart and soul of the drive

As it can be seen, the TMS320F243 EVM provides a very useable hardware solution to control the operation of the hardware elements by an implemented digital control software in its program space. For the considered speed sensorless problem at hand, it is perfectly suitable, and additionally, even the identification process of the motor can be directly accomplished with its help. From every aspect, it can be called the heart and soul of the drive, because every main operation is performed by itself and the other elements in the control chain are only carrying out its orders. Therefore, by the considerations of its mentioned capabilities the implementation of a precise control algorithm with high computational load can be completed on it. This algorithm will be shown in the next section.

5.5. The PC

At last, the drive consists of an operator program running on a PC that monitors the operation of the drive and provides the user given reference signals, while it also receives the status reports of the controller via serial RS232 connection based on the SCI interface. This PC is also the developing platform of the code and through a parallel connected JTAG driver the debugging and monitoring of the uploaded code execution can be performed.

5.5.1. The XMS510P Plus JTAG driver

The main connection to the controller is provided by the Spectrum Digital XMS510P Plus JTAG driver, which allows the full register scan of the CPU through the JTAG status pin and the read /modification of the memory spaces of the DSP card. It is connected to the LPT port of the PC by a parallel cable and its driver integrates into the Code Composer environment of TI. In this way, it is the bridge between the code developing software and the real life controller, providing such a flexibility which is experienced in code development for PCs.

5.5.2. The Code Composer environment

Texas Instruments has a great tradition with the Code Composer Studio (CCS) which allows fast and dependable code generation for extremely wide range of DSPs. This graphical integrated development environment (IDE) is also provided for the TMS320F243 EVM. This software supports Assembly and C based development, with lots of useful features like the tightly integrated editor, the visual project management system, watch window on memory locations and registers, break points and step like debugging, probe points for variable statistics and graphs, code profiling, and user friendly interface with easy programmable reconfiguration. Its C/C++ compiler also includes support for UML and Matlab which could be used for automated code generation (ACG). The real-time debugging tools, including the graphing tool (see *Figure 5.20*), have had beneficial effects on the process of development and testing.

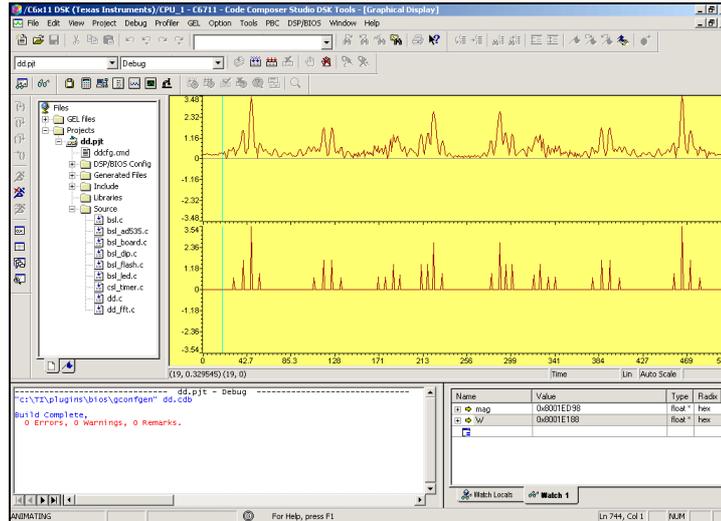


Figure 5.20. Built-in graphing tool of the CCS

CCS also supports RTDX (Real-Time Data Exchange) between the host and the target, interactive profiling and advanced graphical signal analysis, and GEL (General Extension Language) function support for automation activities such as regression testing, automated testing, and customization via scripting, and multi-processor debugging. Therefore, this IDE provides an ideal tool for the implementation process.

5.5.3. The operator interface

The operator interface is a self-written program running on the PC that is connected to the EVM via RS232 serial connection. The program main purpose is to provide the speed and flux reference signals to the controller and in case of faulty operation, it notifies the operator about the malfunction. It also provides the estimated speed of the rotor as back up about the state of the drive. The program is going to be briefly shown in the later section with the synthesised code of the implemented controller.

6. Implementation of the controller

As it was clearly shown the implementation of a speed sensorless IM drive is not an easy task. In *Section 3* several approaches were presented to show solutions of the problem at hand, but it was stated that each of them has its own drawbacks even if it is able to handle the control. By investigation of the presented classical power feed generation approaches and the available self-assembled drive, several issues were discussed that how these hardware elements can and should be used to realize such a complicated device.

Based on these, to be able to build a drive that fulfils the recent industrial requirements that were mentioned in the first section, and based on the presented theoretical control approaches, such a unique speed sensorless controller structure was aimed to be designed that makes possible the independent control of the flux and speed on the full 4 quadrant of operation range of the motor. With the previously described drive, this operation is needed to be provided through the use of the three-phase stator voltages as the control inputs and by the measurements of the stator currents only. Moreover, the controller to be designed must consider the variation of R_r to heat, in order to decrease the effect of parameter uncertainties which were mentioned in *Section 2.3.3*. Because the three-phase voltage feed of the motor is provided by a PWM-VSI, thus during the design, it was unavoidable to pay attention to the error of the inverter, caused by the finite modulation bandwidth. This inverter noise can be modeled as a medium frequency range dominant noise on the inputs of the IM. Beside this, it is also worth considering the measurement errors of the stator currents sensors, which are high frequency noises with most commonly 0.5% relative amplitude in comparison to the measured quantities. In order to prevent any reference tracking error in speed and in flux, the controller was also made able to online adapt to the load torque rising on the shaft. Usually, the needed reference signal of the flux is a constant value during the operation, which describes the most optimal point of the magnetic saturation curve. By the set point control of rotor flux to this *a priori* information defined operation point, the maximal effectiveness of the IM can be guaranteed.

In contrast with this, the reference signal for speed is expected to be as it is pleased, with the occurrences of even uncontinuous step changes. Furthermore, the control has to be provided on that way, which makes possible the most accurate reference tracking with the least possible time constant, but with the awareness that the voltage signal used as a control input, should never cause the rise of such a great current that would damage the motor. The designed controller also considers the specific behaviors and properties of the available digital and analog hardware that is going to be used for the implementation.

6.1. The theoretical structure of control

After careful theoretical and practical design presented in [39], the control structure shown in *Figure 6.1* has been synthetised for the previously defined problem.

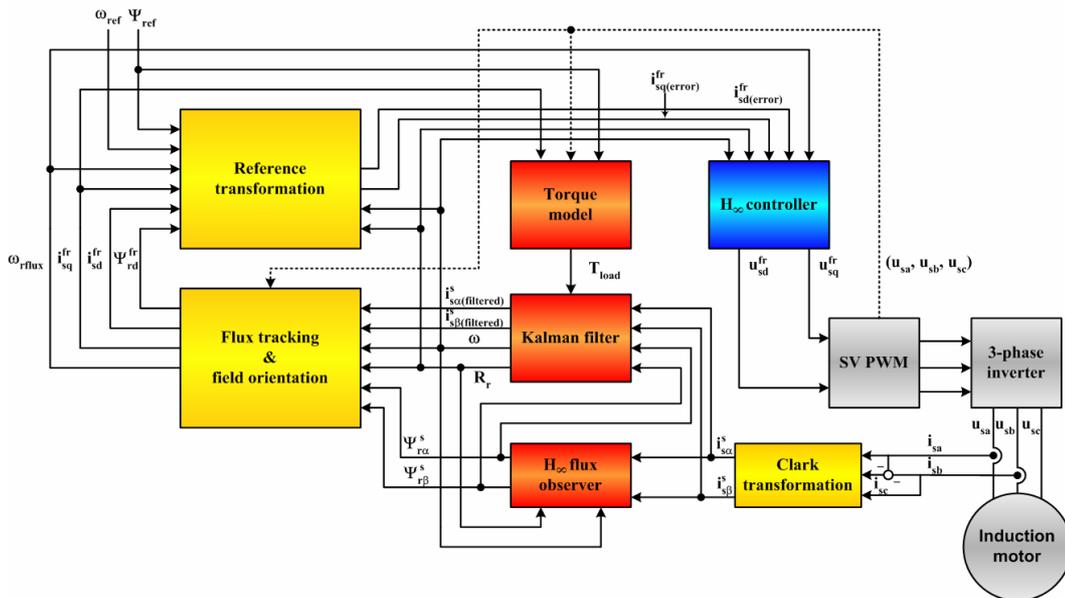


Figure 6.1. The designed control structure

For the functioning of the system presented in *Figure 6.1*, which fulfils the given strict requirements, only the measurements of two stator phase currents are needed, because of the symmetry of the phases, $i_{sc} = -(i_{sa} + i_{sb})$, which is approximately true even in real life applications. Thus, the overall vector can be accurately obtained only from two measurements, which makes possible to save up one ADC per control cycle. Furthermore, because the input voltage feed is generated directly by the system, thus the accurate \mathbf{u}_s^s is known without

measurements, however, for accurate PWM modulation, the DC bus voltage always has to be monitored, which prevents the saving of another ADC. Based on theoretical examinations presented in [38], the relative degrees of the motor model yields, that in the case of known \mathbf{i}_s^s and \mathbf{u}_s^s , each of the state variables presented in equations (2.19), (2.23), and (2.31) can be expressed with them or taken into direct connection via input-output linearization. Thus, these vectors consist of all the information about the operation of the motor, therefore the speed sensorless control of the considered IM drive can be achieved.

The mechanism of the control system is briefly the following:

With the help of the (3.24) Clark transformation and from the two measured stator phase currents, the two phase representation based $i_{s\alpha}^s$ and $i_{s\beta}^s$ components are calculated, which are equal to the d and q components of the stator space vector represented in a stator fixed, stationary ($\omega_k = 0$) reference frame. Then, by a complicated estimation method, which is the connection of an EKF and an H_∞ observer, the unknown $\Psi_{r\alpha}^s(t)$, $\Psi_{r\beta}^s(t)$, ω , and R_r state variables are obtained with the filtering of the stator currents. In this complicated estimation structure, the H_∞ observer, whose design is based on the (2.34) LPV stator oriented flux subsystem of the model, provides the stator oriented estimation of the rotor flux, while the EKF estimates the speed and rotor resistance variations with the filtering of the currents. The application of the EKF is strongly supported by the fact, that its computational load is much less than for a input/output linearized model based full state H_∞ observer [16].

The stability of the above mentioned structure is guaranteed by the highly accurate computation and tuning of the EKF and by the extremely low γ gain of the H_∞ observer. This estimation system is supplemented by a trivial calculation based, however, as it is going to be shown, very important torque estimation module, which provides the similarly unknown value of the load torque. All the information which is provided by the estimation structure is transformed into the rotating reference frame fixed to the rotor flux with the help of the field orientation module. It is important to mention that this orientation can only be

done effectively if the value of $\Psi_{r\alpha}^s$ and $\Psi_{r\beta}^s$ is accurately known. This was the reason why the estimation methods were designed in the stator fixed orientation. With the help of the freshly transformed quantities and the flux and speed reference signals of the system, the $i_{sd_ref}^{fr}$ and $i_{sq_ref}^{fr}$ current reference signals, which are needed to achieve the tracking of the references, are calculated, and the tracking errors of these quantities are received by the H_∞ controller. This structure is very important, because only the (2.37) system model can be given in a useful LPV form, thus efficient H_∞ controller can also be designed only for this case. Moreover, the current regulation is the fastest way of the control of the produced T_e and Ψ . The goodness of this approach is supported by [41] as well. The controller given u_{sd}^{fr} and u_{sq}^{fr} signals, are realized through the SV-PWM hardware module of the TMS320F243, which provides the correct triggering sequence of IGBTs of the VSI.

It is important to note, that because the SV-PWM uses stator fixed reference frame, thus the field oriented control voltage signal provided by the controller has to be transformed back by the inverse of (3.23) into this reference frame. For the correct computation of this transformation, the angle of the rotor flux must be known, which is tracked by the previously mentioned field orientation module.

In conclusion, the above described system theoretically solves the given problem with the use of the presented hardware, but before its direct implementation is described, each of its submodules has to be explained and analyzed from the point of view of how they can handle their task in terms of quality and reliability. This is shown step by step in the following sections.

6.1.1. The H_∞ controller module

The controller module was realized through the (2.37) and (2.38) field oriented LPV model-defined subsystem with mixed sensitivity, to provide robust and as accurate as possible reference tracking of i_{sd}^{fr} and i_{sq}^{fr} with the use of the vector control approach. The exact mathematical design of the produced LPV controller with ω , R_r , and ω_{flux} *scheduling parameters*, for a identified motor

experimental motor can be found in [39], where full validation of the performance by simulations is shown. The controller was synthesised to eliminate the tracking error of the currents and to provide a fast and dynamical control of the motor in a noise filtered environment. This filtering is guaranteed by the later discussed estimation methods. In the design, the external tuning ability of the controller performance was provided to make the easy adjustment and calibration possible. By a careful tuning in Matlab, simulations provided 0.35% steady state tracking error for 0 to 1 step like change in reference signals, with a rising time of 45ms, which is nearly the time constant of the identified motor. This result was obtained when $T_{load} = 0.5\text{Nm}$ was applied to the motor.

By the mixed sensitivity approach, large robust control interval was provided for the whole controller which has shown stability for even 15% parameter variation with a less than 1% tracking error in the 0-10% variation interval.

By different stability checks of [39] it was proved that the presented controller provides extremely rapid and accurate control of the currents and through them, as it was mentioned previously, the control of the whole IM. Moreover, the produced controller is a $2^3 = 8$ corner system described polytopical regulator, which has 5 states, 2 inputs, and 2 outputs. Thus, the controller matrices are obtained through 8 times of 45 multiplications and additions in each control phase, and the method needs additional 45 multiplications and additions to obtain the new controller states and outputs. Fortunately, by the mathematical structure of the model, the polytopical input matrices are identical in each dimension, so the overall operation need is 325 multiplications and additions in each control cycle. However, this computational load might seem to be huge, the F243 DSP can handle it with ease.

6.1.2. I/O linearization based reference transformation

Because the designed controller uses the current subsystem for regulation, the user provided flux and speed reference signals have to be transformed somehow to current reference signals. This task is solved by the I/O linearization of the (6.1) flux and (6.2) dynamic motion equations. The exact process of this linearization can be found in [39].

$$\frac{d\Psi_{rd}^{fr}(t)}{dt} = -\frac{R_r(t)}{L_r} \Psi_{rd}^{fr}(t) + \frac{L_m R_r(t)}{L_r} i_{sd}^{fr}(t) \quad (6.1)$$

$$\frac{d\omega(t)}{dt} = \frac{3p^2 L_m}{2J L_r} \cdot (\Psi_{rd}^{fr}(t) \cdot i_{sq}^{fr}(t)) - \frac{p}{J} \cdot (T_{load}(t) + F\omega(t)) \quad (6.2)$$

From these equations the following simple method of reference transformation can be obtained:

$$i_{sd_{ref}}^{fr}(t) = \frac{L_r}{L_m R_r(t)} \cdot v_1(t) + \frac{1}{L_m} \Psi_{rd}^{fr}(t), \quad (6.3)$$

$$i_{sq_{ref}}^{fr}(t) = \frac{2J L_r}{3p^2 L_m} \cdot \frac{v_2(t)}{\Psi_{rd}^{fr}(t)} + \frac{2L_r}{3p L_m} \cdot (T_{load}(t) + F\omega(t)). \quad (6.4)$$

In this way the input of the controller arises from the result of $i_{sd_{hiba}}^{fr}(t) = i_{sd_{ref}}^{fr}(t) - i_{sd}^{fr}(t)$ and $i_{sq_{hiba}}^{fr}(t) = i_{sq_{ref}}^{fr}(t) - i_{sq}^{fr}(t)$. It is important to mention that because $\Psi_{rd}^{fr}(t)$ is always equal to the value of $|\Psi_r^x(t)|$ in any x reference frame, the division by this quantity can always be accomplished expect at $t = 0$. To avoid the division by zero, in this time moment, the flux amplitude is substituted by a small, nearly zero value which provides continuous reference characteristics. Unfortunately, in practice this would cause a horribly high $i_{sq_{ref}}^{fr}(t)$ at start up, which must be avoided in order to prevent the controller to realize the corresponding large input signal for this reference, thus in the first time moments, till $\Psi_{rd}^{fr}(t)$ does not exceed 0.01, it is worth substituting its value with 1 in the division. In this way, it can be achieved that (6.4) would force the controller for smooth exponential rise till it reaches the border of its unstable dynamical interval, and finally the whole error signal can be received by the controller. In practice, the above mentioned task can be easily implemented with no more than 1 division, 4 multiplications and 4 additions.

Of course the reference change must be also limited to avoid aggressive responses of the controller to step like changes. This is achieved by implemented first order filters with a cutting frequency of 15Hz that force the controller to a smooth exponential rise.

6.1.3. Rotor flux orientation and flux angle tracking

To use the power of field-oriented vector control for the systems mentioned in *Section 6.1.1 and 6.1.2*, it is crucial that the stator currents measured on the motor and the later shown estimation method provided rotor flux is transformed to the $\omega_k = \omega_{\text{flux}}$ defined reference frame. This orientation is done through the following, purely mathematical procedure. Let it be assumed, that the accurate values of $\Psi_{r\alpha}^s(t)$ and $\Psi_{r\beta}^s(t)$ are known. Then, for the complex $\Psi_r^s(t)$ space vector the following equations hold:

$$\cos \rho(t) = \frac{\Psi_{r\alpha}^s(t)}{\sqrt{(\Psi_{r\alpha}^s(t))^2 + (\Psi_{r\beta}^s(t))^2}}, \quad \sin \rho(t) = \frac{\Psi_{r\beta}^s(t)}{\sqrt{(\Psi_{r\alpha}^s(t))^2 + (\Psi_{r\beta}^s(t))^2}}. \quad (6.5)$$

In this way, by using the calculated trigonometrical values of the flux angle the field orientation can be done with ease with the previously mentioned (3.23) Park transformation method. It is needless to calculate the exact value of $\rho(t)$, because for the back transformation of the $u_{sd}^{\text{fr}}(t)$ and $u_{sq}^{\text{fr}}(t)$ control inputs, only the knowledge of $\cos \rho(t)$ and $\sin \rho(t)$ is enough, therefore the whole orientation can be computed without the calculation of any trigonometrical function. Based on this, the task at hand can be completed with relative few operation if the Newton-Raphson method [14] is used for the calculation of the square root operation. This yields to 14 multiplications and additions with 6 divisions for the whole orientation method, because as it follows from *Section 2.4.2*,

$\Psi_{rd}^{\text{fr}}(t) = \sqrt{(\Psi_{r\alpha}^s(t))^2 + (\Psi_{r\beta}^s(t))^2}$ and $\Psi_{rq}^{\text{fr}}(t) = 0$, so only the current needs the Park transformation.

It also notable, that in the first time moment, when $|\Psi_r^s(t)| = 0$, the (6.5) calculation method is not useable, thus $\rho(t) = 0$ assumption is made for this case. The orientation error due to this substitution rapidly vanishes when the magnetic field is built up inside of the machine.

Moreover, it is important to tell that the reference signals need limiting in change as well, to prevent the sensitive control structure to needlessly overreact

the rapid reference changes. For this reason, filter (6.6) was introduced on the reference signals.

$$\frac{15}{s+15} \quad (6.6)$$

6.1.4. The structure of estimation

The main task of the designed estimation method, presented in *Figure 6.2*, is to make possible the speed sensorless control of the IM. Because the previously introduced methods are found on the accuracy of the system information, thus the estimation method has to be able to satisfy these needs, and even be able to attenuate the measurement and the inverter caused system noises. For these expectations, such a method is perfectly suitable which consist of three parts:

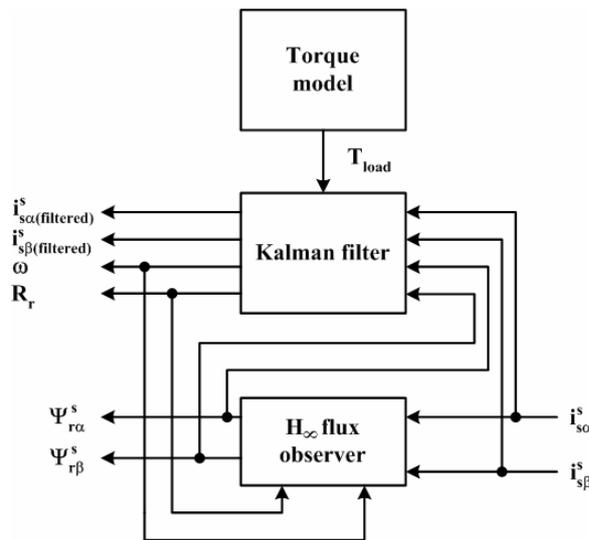


Figure 6.2. Theoretical structure of the designed estimation method

During the design of the estimators, it has quickly turned out that the equations which contain the ω , R_r , $\Psi_{r\alpha}^s(t)$ and $\Psi_{r\beta}^s(t)$ state variables cannot be transformed into a parameter-affine form, which would enable the design of only one robust LPV H_∞ observer to solve the problem. However, it is also true, that the estimation of this nonlinear problem with a EKF does not offer such accurate results than a robust H_∞ observer providing only the estimation of $\Psi_{r\alpha}^s(t)$ and $\Psi_{r\beta}^s(t)$. Thus, to ensure the precise estimation and the robustness of operation, such a unique closed loop structure was designed, which is the interconnection of

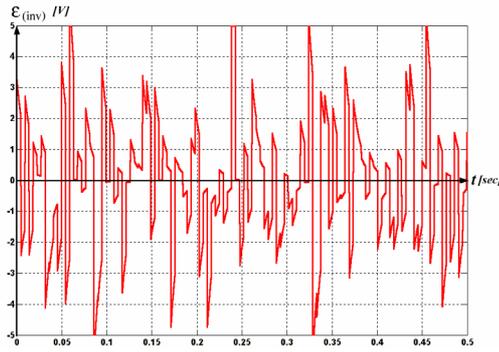
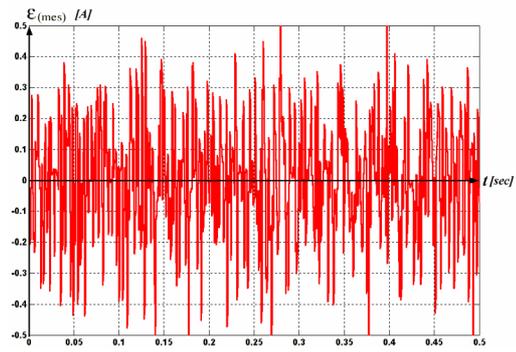
a LPV H_∞ observer and an EKF. The ω and $R_r(t)$ scheduled H_∞ observer provides the stator oriented estimation of $\Psi_{r\alpha}^s(t)$ and $\Psi_{r\beta}^s(t)$ with the help of the measured $i_{r\alpha}^s(t)$ and $i_{r\beta}^s(t)$, while the EKF calculates ω and $R_r(t)$ with the filtering of $i_{r\alpha}^s(t)$, $i_{r\beta}^s(t)$ based on the estimated values of $\Psi_{r\alpha}^s(t)$ and $\Psi_{r\beta}^s(t)$ and on the measured stator currents. This structure is completed by the dynamical motion equation based torque reference model, which gives the helping hand to guarantee the accurate ω estimation by the Kalman filter.

6.1.5. The H_∞ observer

The observer method was designed based on the (6.7) LPV stator-oriented flux subsystem of the model

$$\frac{d}{dt} \begin{bmatrix} \Psi_{r\alpha}^s(t) \\ \Psi_{r\beta}^s(t) \end{bmatrix} = \begin{bmatrix} -\frac{p_2(t)}{L_r} & -p_1(t) \\ p_1(t) & -\frac{p_2(t)}{L_r} \end{bmatrix} \cdot \begin{bmatrix} \Psi_{r\alpha}^s(t) \\ \Psi_{r\beta}^s(t) \end{bmatrix} + \begin{bmatrix} \frac{L_m p_2(t)}{L_r} & 0 \\ 0 & \frac{L_m p_2(t)}{L_r} \end{bmatrix} \cdot \begin{bmatrix} i_{r\alpha}^s(t) \\ i_{r\beta}^s(t) \end{bmatrix} \quad (6.7)$$

where the $p_1(t) = \omega(t)$ and $p_2(t) = R_r(t)$ parameters were used for scheduling. Here, also mixed sensitivity structure was used during the design to calibrate the observer for accurate estimation of the flux with robustness. The whole mathematical procedure of this is explained in details in [39]. The produced $2^2 = 4$ corner system described polytopical observer was tested in Matlab simulations as well, and it showed extreme performance with no more than 0.33% estimation error in case of 100V, 50Hz voltage feed and 0.5Nm load torque, due to its low γ worst case gain which is $8.4852 \cdot 10^{-5}$. During the simulations, modeled inverter and measurement noise effects were applied to the system. These were generated with filtering of white noise effects in such a way that for the inverter noise a 10Vpp medium frequency dominant noise presented in *Figure 6.3*, while for the sensor noise a high frequency dominant noise with a 0.5 A amplitude presented in *Figure 6.4* was introduced.


Figure 6.3. Modeled inverter noise

Figure 6.4. Modeled measurement noise

However, the current technologies, like the Labdrive inverter provided Hall sensors of the LEN company, have most commonly 0.5% sensor accuracy, which introduces much less noise effect into the system, the environment was intentionally chosen to be highly noisy to make able the designed controller to adapt to current glitches caused by the PWM modulation in the stator windings.

The robust performance was also investigated for 5% random parameter variance, which gave a relative error of 4.3% in the estimations. This can be explained based on the periodicity of the error, by an increased phase delay of the observer. Based on these, it can be concluded that a very accurate estimation of the flux was synthesised, whose implementation due to the 2 inner states, with 2 inputs and 2 outputs of the observer, can be realized through relative few operations. Because of the properties of the mathematical model, the output matrix of the observer is identical in each of its polytopical dimensions, therefore the whole computational load is no more than 44 multiplications and additions.

6.1.6. The EKF

Based on the previously shown EKF theory, the estimation method was designed for the 6 state nonlinear system constructed from equations (2.19), (2.23), and (2.31) in case of $\omega_k = 0$ orientation. Because of the strong dynamical properties of the system model, in every prediction phase the numerical solution of the equation system is calculated through the Adams-Basforth 3-step recursive approximation (for more details see [39]), while in the correction phase, it was found enough to use only the Euler method for the discretisation of the IM model, which provides the Jacobi matrix for this task. By considering the characteristics

of the model and after some try and error, the \mathbf{Q} and \mathbf{R} matrices were chosen to be the following:

$$\mathbf{Q}_{ij} = 0, \text{ except } \mathbf{Q}_{33} = \mathbf{Q}_{44} = \frac{0.0117h}{L_s\sigma}; \quad i, j \in \{0,1,\dots,6\} \quad (6.8)$$

$$\mathbf{R}_{ij} = 0, \text{ except } \mathbf{R}_{11} = \mathbf{R}_{22} = 0.0205, \quad \mathbf{R}_{33} = \mathbf{R}_{44} = 13.85; \quad i, j \in \{0,1,\dots,4\} \quad (6.9)$$

The $h/L_s\sigma$ term in (6.8), was provided to eliminate the effect of the inverter noise on the first dynamical subsystem. The measured current signals and the observer provided fluxes were chosen to be the reference state variables of the EKF, while the outputs of the estimator was constructed in the manner to provide the unknown value of ω , \mathbf{R}_r , and the filtered values of $i_{r\alpha}^s(t)$ and $i_{r\beta}^s(t)$.

The effectiveness of the designed EKF was examined for the specific model and the results of these simulations in Matlab showed that the relative greatness of the errors were 0.41% for ω and 0.08% in case of \mathbf{R}_r with an average noise attenuation of 98% on the measured $i_{r\alpha}^s(t)$ and $i_{r\beta}^s(t)$ while the previously mentioned load and excitation conditions were applied to the model.

Furthermore, the robustness of the algorithm was also tested in case of 5% variance in system parameters where the relative error was 0.8% for ω and 0.33% for \mathbf{R}_r . Unfortunately, besides of these results, it was experienced that the previously shown great noise attenuation property lost somewhat from its optimality in an affordable manner.

It is also important to mention, that the designed method has a large computational load because of the huge matrix operations with even matrix inverse calculation, thus it is the weak point of designed control structure. However, from several approaches, this technique provides the fastest way to realize such an accurate estimation method for the strongly NL equations of the IM. This is proved by [16] also. Fortunately, with the large speed capability of the F243 the designed EKF is expected to be calculated fast if its implementation is carefully accomplished.

6.1.7. Estimation of the load torque

Nowadays, several published papers [9, 11, 31] which describe the design of H_∞ controller for the IM, ignore the estimation of the load torque, either by concealing this or assuming it to be known in the system. This can be accepted in such applications where the torque really does not change, only lightly fluctuates around a constant load during the operation. However, in case of the more drastic change in the load, these methods are turned out to be worthless. Unfortunately, the measurement of the load torque is highly expensive and if a speed sensorless control has to be achieved, then only the stator currents and voltages can be used for its estimation. Although, from the (2.19) dynamic motion equation the value of T_{load} can be calculated, it cannot be forgotten, that ω and the flux presented also in this equation were estimated by the known value of the load torque. Thus, if they would be used for the estimation of T_{load} , then such a looped estimator would be produced, which amplifies its own error instead of convergence in the estimation. Based on this, the value of $\Psi_{rd_ref}^{fr}$ and ω_{ref} are used in the calculation instead of the estimated quantities, and after some transformation the following simple equation can be used for guessing the load:

$$T_{load} = \frac{2L_r}{3pL_m} \cdot (\Psi_{rd_ref}^{fr} \cdot i_{sq}^s(t)) - F\omega_{ref} \quad (6.10)$$

It is important to mention that these reference signals cannot be used directly just only in that case when their evolution is limited by dynamical filters in a similar way to the expected behavior of the system states, or on the other hand, the estimation methods presented in *Section 6.1.4 and 6.1.5* lose their connection with reality, because of the step like errors of the torque calculation. Because of this reason, the weighting filter were chosen similarly like the speed limiting filter of the references in *Section 6.1.3*. In this way, estimation of the load torque could be achieved in the simulations with a relatively low steady state error.

6.1.8. Tuning parameters

Because of the existing errors in each levels of the interconnected structure, the looped control system is not able to perfectly solve the whole control problem,

only in that case when the modules are tuned to each other by the careful calibration of the tuning parameters. The controller was initially designed to have perfect tuning ability to achieve this goal. By several considerations mentioned in [39] the tuning was realized in the following way.

Firstly, by the amplification of the tracking errors of ω_{ref} and $\Psi_{\text{rdref}}^{\text{fr}}$ before and after the reference transformation, and by applying gains for the deviations from $i_{\text{sdref}}^{\text{fr}}$, $i_{\text{sqref}}^{\text{fr}}$, the sensitivity of control can be calibrated to achieve accurate reference tracking. However, it is not worth increasing these gains limitlessly, because in this way the controller can achieve a state, when because of the maximum 250V of the SV-PWM modulation limited control signal, it begins aggressively pulling around the stator voltage to input more energy into the system. This phenomenon produces such oscillations in the transients, which cannot be allowed. So it is better to use external offset compensation to influence the greatness of the estimated flux and load torque and making the reference tracking optimal in this way. The exact value of these compensations can be most perfectly calculated in the case of large torque and flux signals because their effect is insignificant for small values of these quantities. The exact value of these compensations will be given with the simulations in the next section.

6.2. Simulation results

Before any kind of implementation begins, it is worth investigating the system performance by numerical simulations. Therefore, the previously mentioned control structure and the continuous model of the IM based on the equations given in *Section 2.3* was modeled in the Simulink environment of Matlab. Because the implementation of the complicated control system at hand is going to be completed digitally, each of the controller blocks, except the model of the IM, were discretised with a $h = 10^{-4}$ time step, which assumes that the cycle time of the control program is no more than 100 μsec . However, this kind of assumption might seem to be strict, with careful implementation it can be fulfilled with the recently introduced DSPs. In the case of the considered F243 processor, this assumption has to be relaxed to $h = 10^{-3}$ of course, which can be achieved with a

highly optimized realization in Assembly. The measured signals of the control structure are also assumed to be digitalized with the same sampling time, to ensure the correct functioning of the discretised estimation methods. This restriction can also be accomplished with the given drive, because the 1μsec A/C conversion time of the incoming signals is relatively small to the cycle time of whole control loop. Therefore these ADCs can be assumed to be happening at once from the view of the algorithm. Because the overall dynamics of the closed loop drive are much slower than the step size of the discretisation, the discretised continuous controller and observer must approximate well the continuous operation. Moreover, for the EKF such discretisation is not needed, because it provides the estimation in discrete time.

The above mentioned discretisation is done online inside of the programmed S-function Simulink models, so from the polytopical form of the methods the instant parameter value defined

$$\dot{\mathbf{x}}_K(t) = \mathbf{A}_K(\mathbf{p}(t)) \cdot \mathbf{x}_K(t) + \mathbf{B}_K(\mathbf{p}(t)) \cdot \mathbf{u}_K(t), \quad (6.11)$$

$$\mathbf{y}_K(t) = \mathbf{C}_K(\mathbf{p}(t)) \cdot \mathbf{x}_K(t), \quad (6.12)$$

LTI system can be given in a discrete form described by the

$$\dot{\mathbf{x}}_K(k) = \underbrace{e^{\mathbf{A}_K(\mathbf{p}(k))h}}_{\Phi_k(\mathbf{p}(k))} \cdot \mathbf{x}_K(k) + \underbrace{\mathbf{A}_K^{-1}(\mathbf{p}(k)) \left(e^{\mathbf{A}_K(\mathbf{p}(k))h} - \mathbf{I} \right) \mathbf{B}_K(\mathbf{p}(k))}_{\Gamma_k(\mathbf{p}(k))} \cdot \mathbf{u}_K(k), \quad (6.13)$$

$$\mathbf{y}_K(k) = \mathbf{C}_K(\mathbf{p}(k)) \cdot \mathbf{x}_K(k), \quad (6.14)$$

equations, which, due to the h small step size, can be approximated with the

$$\Phi_k(\mathbf{p}(k)) \approx \mathbf{I} + h \cdot \mathbf{A}_K(\mathbf{p}(k)) + \frac{h^2 \cdot \mathbf{A}_K^2(\mathbf{p}(k))}{2}, \quad (6.15)$$

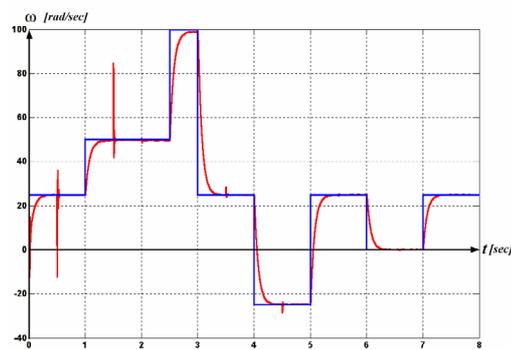
$$\Gamma_k(\mathbf{p}(k)) \approx h \cdot \mathbf{B}_K(\mathbf{p}(k)) + \frac{h^2 \cdot \mathbf{A}_K(\mathbf{p}(k)) \cdot \mathbf{B}_K(\mathbf{p}(k))}{2}, \quad (6.16)$$

matrices. In this way, the continuous controller and observer approximated in second order operates with the continuous dynamic properties in discrete time. If the second order terms are left out, then the discretisation can be implemented in a

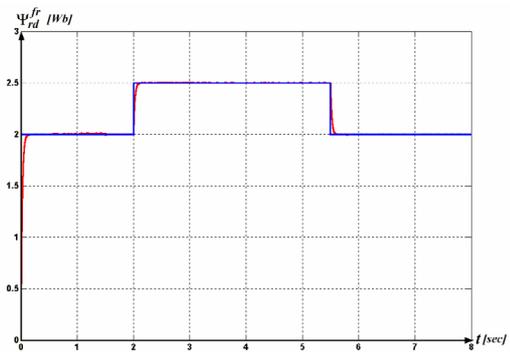
real life hardware with ease, because the operation of the discretisation can be mathematically done before the obtaining of the \mathbf{A}_k and \mathbf{B}_k matrices from the corner systems of the polytopical set defined LPV controller and observer. In this way, only the elements of the \mathbf{A} and \mathbf{B} matrices have to be multiplied by the step size, and 1 added to the diagonals of \mathbf{A} in each of their dimensions before hardwiring these values into the program. In such way, the discretisation introduces no additional operation into the system.

In the simulations by some *try and error*, the tuning parameters, described in *Section 6.1.8*, of the system were calibrated in the previously mentioned noisy environment, where the offset of the flux was affirmed to be 0.992, while the sensitivity gain of the flux tracking error was adjusted to be 45. The offset of the load torque was calibrated to 1.021 and the error sensitivity of the speed tracking was defined to have a gain of 150. The controller states were limited in the interval of ± 500 in respect to the large reference signals during the startup. The sensitivity for the stator current tracking deviation was chosen to be 35 in both of the cases. The overall system produced in this way was investigated in a noiseless environment, with such intensive reference signals which not only investigates the system behavior on the full 4/4 operation range of the IM, but also test the system on the bounds of its operation.

In this way the simulations were completed with the additive noises described in *Section 6.1.4* and the results are given in *Figures 6.5(a-p)*.



**Figure 6.5(a). Reference (red)
tracking (blue) for ω**



**Figure 6.5(b). Reference (red)
tracking (blue) for Ψ_{rd}^{fr}**

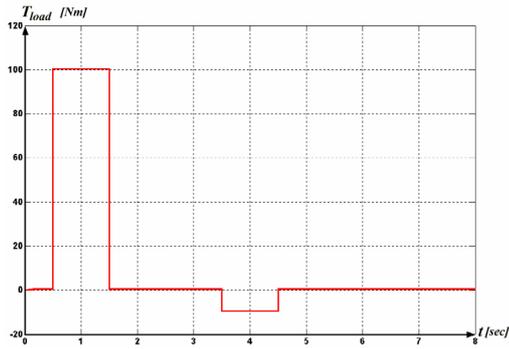


Figure 6.5(c). Change of load

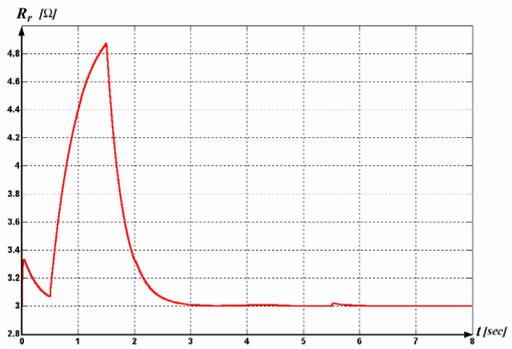


Figure 6.5(d). Change of R_r

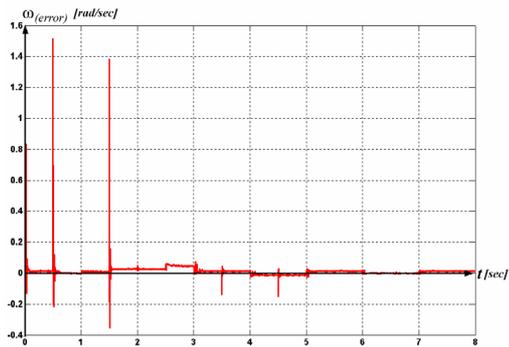


Figure 6.5(e). Estimation error of ω

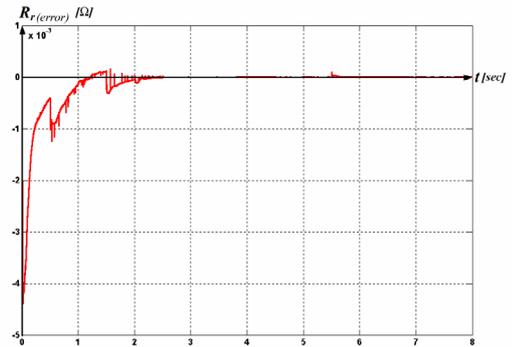


Figure 6.5(f). Filtering error of R_r

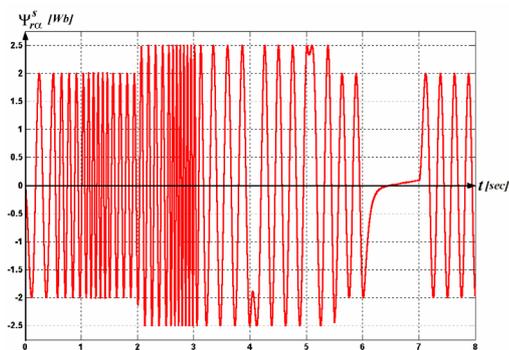


Figure 6.5(g). Change of $\Psi_{r\alpha}^s$

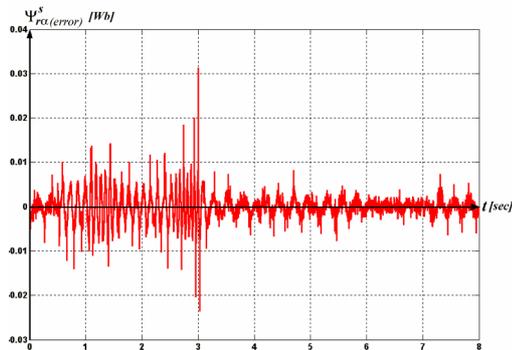


Figure 6.5(h). Estimation error of $\Psi_{r\alpha}^s$

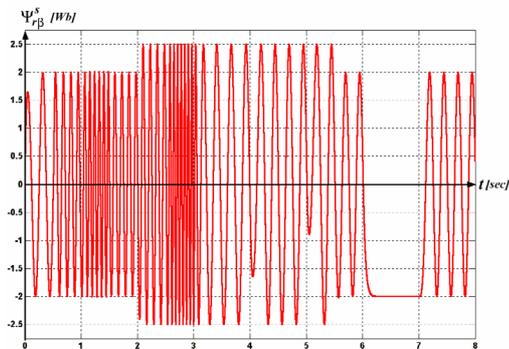


Figure 6.5(i). Change of $\Psi_{r\beta}^s$

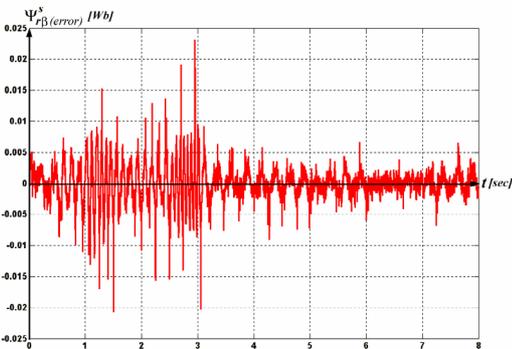


Figure 6.5(j). Estimation error of $\Psi_{r\beta}^s$

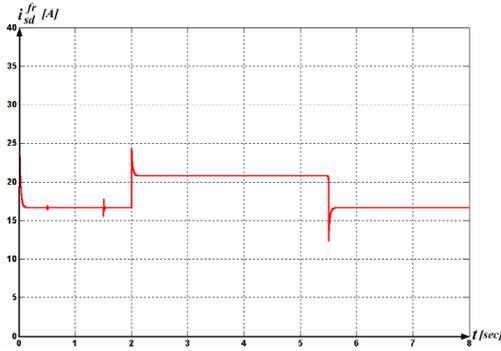


Figure 6.5(k). Change of i_{sd}^{fr}

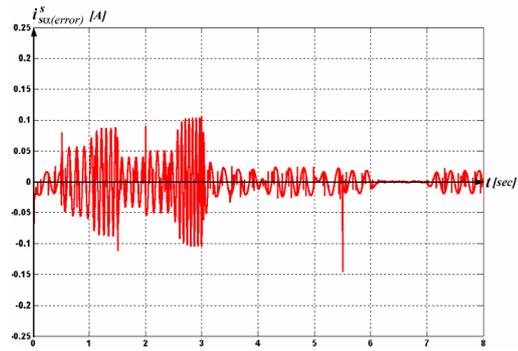


Figure 6.5(l). Filtering error of i_{scl}^s

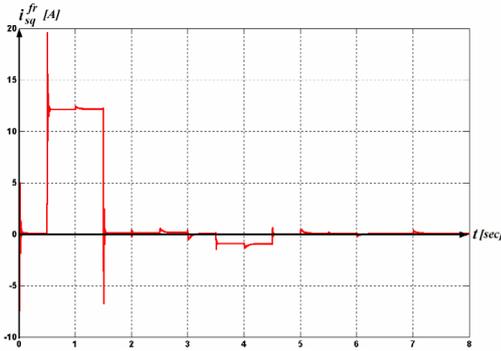


Figure 6.5(m). Change of i_{sq}^{fr}

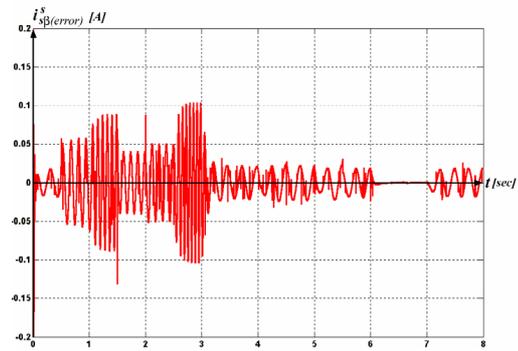


Figure 6.5(n). Filtering error of $i_{s\beta}^s$

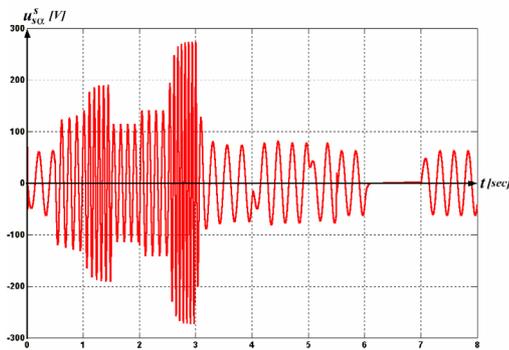


Figure 6.5(o). Change of u_{scl}^s

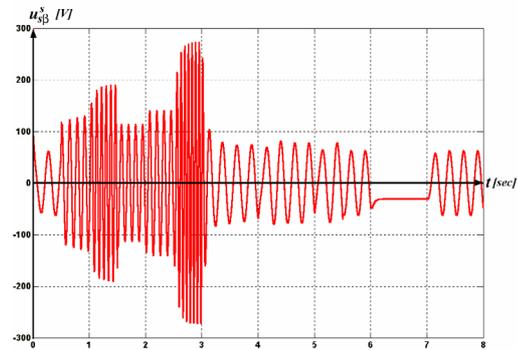


Figure 6.5(p). Change of $u_{s\beta}^s$

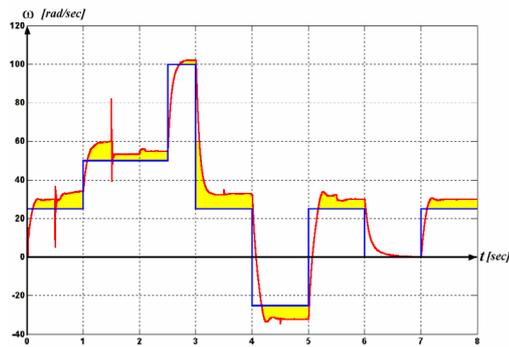
During the simulations the load torque is given with a slight clutching to the system in the first time moments (*Figure 6.5(c)*) The clutching is needed to prevent the motor to slip too much backwards at startup, because in the other case the controller would try to rise the speed in this instable dynamical interval with great voltage cuffs, which would produce oscillations of ω for almost 0.4 sec. It can be seen, that at 0.5 sec, the torque rises to 200 times of its original value, which is quickly compensated by the controller in 5msec and the motor reaches the given speed reference signal again. The reference tracking is very rapid, the

limited speed reference signal are instantly followed by the real speed of the motor (see *Figure 6.5(a)*). The changing of the direction of rotation is also tracked down quickly, in no more than 35msec, and the negative load change is also compensated with the same effectiveness. Here the IM operates as a dynamic brake, feeding energy back to the power system. Even if this property cannot be used with the presented drive because of the structure of the inverter, it is good to know, that the stability and good performance is provided for this operation range as well. At the 6. sec, the controller holds the given load with 0 speed, which is significant in the view that the most common algorithms are not capable for this because of the lack of such accurate estimation as it presented in *Figures 6.5(e-n)*.

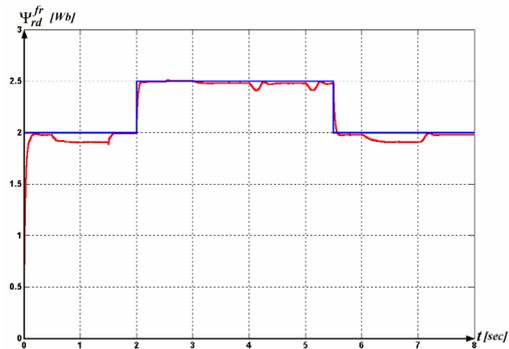
During the full operation, the tracking of the rotor flux has no significant effect on the speed tracking, although Ψ_{rd}^{fr} is crucial in the production of the electromagnetical torque. By considering the flux tracking, it can be concluded that even for extreme and rapid changes in the operation the flux remains stable (see *Figure 6.5(b)*).

For the tracking of both reference signals, the relative steady state offset error did not exceed 1.5% during this dynamic testing, thus in this way, it can be said, that the tracking capability of the controller is extremely good. Moreover, by *Figure 6.5(o) and 6.5(p)* the controller only gives out more than the maximal 500Vpp, when the rated torque of the motor is reached. During startup, the current load of the power system is also not significant, which provides economical operation of the drive.

The robustness of the simulated drive was also tested with 5% parameter uncertainty. The results of reference tracking are given in *Figure 6.6(a) and 6.6(b)* for this case. The worst error of the tracking of speed was nearly 25%, which means that the interval of robust stability is decreased in case of such heavy noises and dynamic changes, but the controller remained stable even for this large uncertainty. It must be mentioned, that by taking some performance tests on the system, this offset error can be also eliminated with ease with the help of the tuning parameters. The instability only occurs for very large, 15%-20% parameter variance, which results in the oscillations of the control signal.



**Figure 6.6(a). Reference (red)
tracking (blue) for ω**

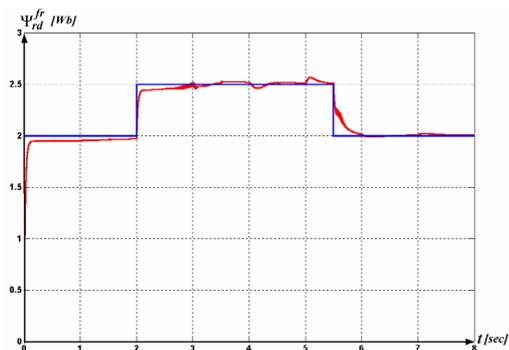


**Figure 6.6(b). Reference (red)
tracking (blue) for Ψ_{rd}^{fr}**

The calculation of the load torque, as it was stated many times in the literature, is a great theoretical challenge in this problem, thus even the open loop torque estimator system operates well, the reference tracking of the closed loop control with the use of torque estimations does not provide such good results as in the previous simulations. In this case, the simulations give the results presented in *Figure 6.7(a) and 6.7(b)* for the previously used rapid reference signals and load changes. The bad tracking ability of this case can be mainly explained by the slow rising time of the load calculation, which in case of rapid changes misleads the estimator methods, and the closed loop system can only track down the real life dynamic answers of the system with limited success.



**Figure 6.7(a). Reference (red)
tracking (blue) for ω**



**Figure 6.7(b). Reference (red)
tracking (blue) for Ψ_{rd}^{fr}**

In case of clutched, constant torque, the control structure presents nearly the effectiveness that could be seen in the previous case (*see Figure 6.8(a) and 6.8(b)*). Although this estimation method for the load torque does not provide such freedom which could be expected, it is capable of making the drive adaptable to

the load changes, if these changes are carefully applied to the system, which can be achieved during the implementation. Thus, this control structure gives more than the [9, 31] papers, because it makes possible the reference tracking without knowing the load torque. The present researches are concentrating on the development of such an estimation method which is capable of the accurate and independent estimation of the torque with the help of the phase difference between the stator currents and voltages. In this way the presented controller driven IM drive could be even applied for very dynamic tasks, for example as a mechanical power source of a vehicle.

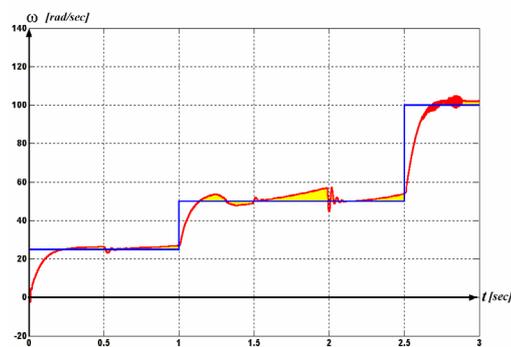


Figure 6.8(a). Reference (red) tracking (blue) for ω

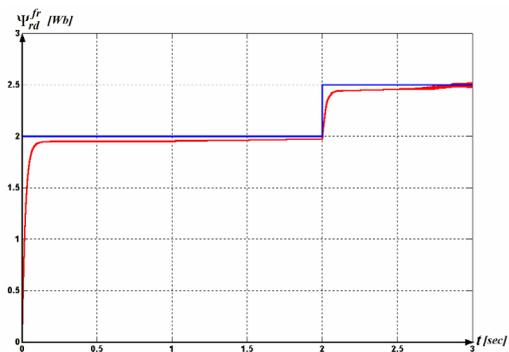


Figure 6.8(b). Reference (red) tracking (blue) for Ψ_{rd}^{fr}

6.3. The identification of the motor

To be able to apply the previously presented control structure for the real time control of the given IM of the considered drive, firstly the identification of the motor parameters has to be completed. Since the model of the IM is highly nonlinear and several of the parameters are not measurable such as L_r and R_r , the use of parameter identification methods is unavoidable. Unfortunately, even if very accurate and sophisticated methods exist for parameter identification in the LTI case, such as the linear regression, least square, maximum Likelihood, and Bayes prediction based methods, for NL systems, the solution of a global optimization problem is the only existing way, if the linearization cannot be afforded. This optimization can be solved by the *gradient method* that recursively tries to refine the estimated parameter values till the real system behavior is achieved in an adequate manner.

6.3.1. Method of identification

The parameter estimation of a NL dynamical systems, defined as an optimization problem, is most commonly given in the form of the presented algorithmic method below [13]:

Let the following be given:

- ◆ A Ξ^N is a sequence of measured values, or samples, which consist of the measured input-output pairs till the sampling time moment N, thus $\Xi^N = \{(\mathbf{y}(k), \mathbf{u}(k)) | k = 1, \dots, N\}$.

- ◆ A predictive model in the form of

$$\hat{\mathbf{y}}(k|\theta) = \mathbf{g}(k, \Xi^{k-1}, \theta), \quad (6.17)$$

where $\hat{\mathbf{y}}(k|\theta)$ is the predicted value of the k^{th} time step and the estimated parameter values given in θ while $\mathbf{g}(\cdot)$ is a NL vector function.

- ◆ A prediction error sequence, which is computed from the pervious values.

$$\varepsilon(k|\theta) = \mathbf{y}(k) - \hat{\mathbf{y}}(k|\theta). \quad (6.18)$$

- ◆ A loss function

$$V_N(\theta, \Xi^N) = \frac{1}{N} \sum_{k=1}^N \|\varepsilon(k|\theta)\|_q, \quad (6.19)$$

whit an applicable $\|\cdot\|_q$ vector norm.

It is important to note, that the selection of the most suitable norm is method dependent, but most commonly the Euclidian norm

$$\|\varepsilon\|_2 = \frac{1}{2} \sum_{i=1}^{\dim(\varepsilon)} \varepsilon_i^2 \quad (6.20)$$

is used.

If these exist, than such a parameter identification method can be searched for, that produces such a $\hat{\theta}_{\text{method}}(\Xi^N)$ estimated parameter vector form the measured values, to which (6.21) holds:

$$\hat{\theta}_{\text{method}}(\Xi^N) = \arg\left(\min_{\theta} V_N(\theta, \Xi^N)\right), \quad (6.21)$$

so the loss function is minimal at the estimated parameter values. If the previously mentioned Euclidian norm is used to define (6.19) than this equation will become:

$$V_N(\theta, \Xi^N) = \frac{1}{N} \sum_{k=1}^N \|\mathbf{y}(k) - \mathbf{g}(k, \Xi^{k-1}, \theta)\|_2, \quad (6.22)$$

and then

$$\hat{\theta}_{\text{method}}(\Xi^N) = \arg\left(\min_{\theta} \frac{1}{N} \sum_{k=1}^N \|\mathbf{y}(k) - \mathbf{g}(k, \Xi^{k-1}, \theta)\|_2\right). \quad (6.23)$$

Even if (6.23) seems to be quadratic, the presented nonlinear $\mathbf{g}(k, \Xi^{k-1}, \theta)$ usually prevents the analytical solution. However, the quadratic form generally guaranties the existence of minimum point and trough of it the solution of the problem at hand, but in most cases more than one minimum point exist. By the absence of analytical solution, a global optimization process is needed. One of the existing solutions is the gradient method, which is named about the $\vec{\mathbf{G}}(\cdot)$ gradient vector of the $\mathbf{f}(\cdot)$ function. The gradient vector of $\mathbf{f}(\cdot)$ in \mathbf{x} is defined as follows:

$$\vec{\mathbf{G}}(\mathbf{x}) = \left[\left. \frac{\partial \mathbf{f}(\mathbf{x}, t)}{\partial x_1} \right|_{\mathbf{x}}, \dots, \left. \frac{\partial \mathbf{f}(\mathbf{x}, t)}{\partial x_m} \right|_{\mathbf{x}} \right]^T, \quad \mathbf{x} = [x_1, \dots, x_m]^T. \quad (6.24)$$

For vector functions, $\vec{\mathbf{G}}(\cdot)$ is also referred as the *Jacobi matrix*. Moreover, while $\vec{\mathbf{G}}(\cdot)$ shows the tangential, the (6.25) defined $\vec{\mathbf{G}}_2(\cdot)$ gives the curvature, so the convex or concave property of the $\mathbf{f}(\cdot)$ function in \mathbf{x} .

$$\left[\vec{\mathbf{G}}_2(\mathbf{x}) \right]_{ij} = \frac{\partial^2 \mathbf{f}(\mathbf{x}, t)}{\partial x_i \partial x_j}, \quad i, j \in \{1, \dots, m\}, \quad (6.25)$$

which is also called the *Hesse matrix*. If $\mathbf{f}(\cdot)$ has a minimum in \mathbf{x}^* than the following is true:

$$\vec{\mathbf{G}}(\mathbf{x}^*) = \mathbf{0}, \quad \text{and} \quad \det(\vec{\mathbf{G}}_2(\mathbf{x}^*)) > 0 \quad (\text{positive definit}) \quad (6.26)$$

By the use of these properties, a minimum point of the error function can be reached with ε affordable error, by a recursive process started from a given \mathbf{x}_0 initial state. This process, called the gradient method, is the following:

1. Initially $i=0$, where i is the number of completed recursive steps. Let $\mathbf{x}_{i=0} = \mathbf{x}_0$ which is the initial approximation of the parameter values based on *a priori* information.
2. $\vec{\mathbf{G}}(\cdot)$ gradient vector of the loss function in the \mathbf{x}_i point is computed from the measured values.
3. If the gradient vector is small enough, such as $\|\vec{\mathbf{G}}(\mathbf{x})\|_2 < \varepsilon$, then it is assumed that the minimum is reached and the process ends.
4. In the other case, based on the sign of the gradient, the estimated parameters are refined by δ scaling step size (6.27) and with $i=i+1$ the algorithm continues from *Step 2*.

$$\mathbf{x}_{i+1} = \mathbf{x}_i - \vec{\mathbf{G}}(\mathbf{x}_i) \cdot \delta \quad (6.27)$$

In this way, the parameter identification of NL systems can be completed.

By examining this mathematical approach, it quickly turns out that the effectiveness of the method strongly depends on the \mathbf{x}_0 initial values of the parameters. Because if these chosen values are far away from the real ones, the method can quickly bump into other, even very small potholes given minimum points on the function curve and fail the identification of the real parameters. Moreover, every iteration step has a polynomial time need, therefore the initially large step size is decreased as the method approaches towards the minimum point to provide small computational time with fine results. This step size variation can be easily solved by choosing its magnitude compared to the norm of the gradient.

The previously mentioned method was implemented in Matlab by the help of the identification toolbox, to solve the parameter identification problem of the presented IM. For this purpose, a script code was written which implements the gradient method with variable step size. The method uses the precalculated first

order derivatives of the stator fixed reduced IM model and it also varies the step sizes for the parameters in a different manner. The model of the IM described in *Section 2.3* was reduced, because some parameters like the stator resistance and inductance could be computed from the results of measurements, therefore their identification is not needed. To obtain these parameters locked rotor and no load tests were completed. In this way, the following equation system could be given for the parameter identification method:

$$\frac{d}{dt} \begin{bmatrix} \Psi_{r\alpha}^s(t) \\ \Psi_{r\beta}^s(t) \\ i_{s\alpha}^s(t) \\ i_{s\beta}^s(t) \end{bmatrix} = \mathbf{A}(\theta, R_s, L_s, \omega) \cdot \begin{bmatrix} \Psi_{r\alpha}^s(t) \\ \Psi_{r\beta}^s(t) \\ i_{s\alpha}^s(t) \\ i_{s\beta}^s(t) \end{bmatrix} + \mathbf{B}(\theta, L_s) \cdot \begin{bmatrix} u_{sd}^k(t) \\ u_{sq}^k(t) \end{bmatrix} \quad (6.28)$$

where

$$\mathbf{A}(\theta, R_s, L_s, \omega) = \begin{bmatrix} -\frac{\theta_1}{\theta_2} & -\omega & \frac{\theta_1 \cdot \theta_3}{\theta_2} & 0 \\ \omega & -\frac{\theta_1}{\theta_2} & 0 & \frac{\theta_1 \cdot \theta_3}{\theta_2} \\ \frac{\theta_1 \cdot \theta_3}{L_s \cdot \theta_2^2 - \theta_2 \cdot \theta_3} & \omega \cdot \frac{\theta_3}{L_s \cdot \theta_2 - \theta_3^2} & -\frac{\theta_2 \cdot \left(\frac{L_s}{\theta_2^2} \cdot \theta_1 + R_s \right)}{L_s \cdot \theta_2 - \theta_3^2} & 0 \\ -\omega \cdot \frac{\theta_3}{L_s \cdot \theta_2 - \theta_3^2} & \frac{\theta_1 \cdot \theta_3}{L_s \cdot \theta_2^2 - \theta_2 \cdot \theta_3} & 0 & -\frac{\theta_2 \cdot \left(\frac{L_s}{\theta_2^2} \cdot \theta_1 + R_s \right)}{L_s \cdot \theta_2 - \theta_3^2} \end{bmatrix}$$

$$\mathbf{B}(\theta, L_s) = \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ \frac{\theta_2}{L_s \cdot \theta_2 - \theta_3^2} & 0 \\ 0 & \frac{\theta_2}{L_s \cdot \theta_2 - \theta_3^2} \end{bmatrix}, \quad \text{and} \quad \begin{bmatrix} \theta_1 \\ \theta_2 \\ \theta_3 \\ \theta_4 \\ \theta_5 \end{bmatrix} = \begin{bmatrix} R_r \\ L_r \\ L_M \\ J \\ F \end{bmatrix}.$$

6.3.2. Measurements

To use the previously described identification algorithm several measurements were done on the motor. Firstly, the stator resistance was measured, which turned out to be 96.2 Ω . Then by a no load test, the stator inductance was computed to

be 352,4 mH from the phase differences of the voltages and the currents and from the given rated parameters of the motor.

After this, by the help of a HP 5461B oscilloscope, several measurements were taken on the stator phase and line voltages and on the stator currents, when different feeding conditions were applied. In these tests, the speed of the rotor was also measured with the presented encoder on the shaft by the help of the QEP hardware of the DPS. The different conditions of the voltage feed were chosen to map the overall frequency answer function of the applied motor. For this purpose, a simple program was written on the TMS320F243 microcontroller, to realize the PWM stator voltage of the motor on different frequencies and amplitudes, while it also measures the rotor speed from the encoder given signals.

Some results of these measurement results are given in the following figures, with the measured analog signal of the encoder for completeness.

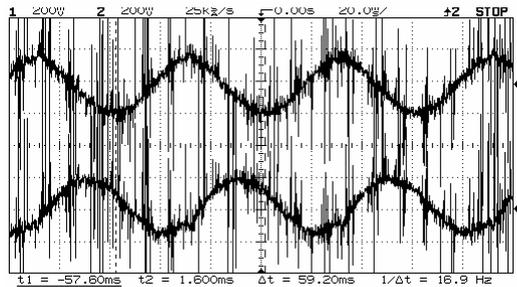


Figure 6.9(a). Measured i_{sa}^s and i_{sb}^s in case of $f_0 = 16,9\text{Hz}$ and $V_p = 100\text{V}$

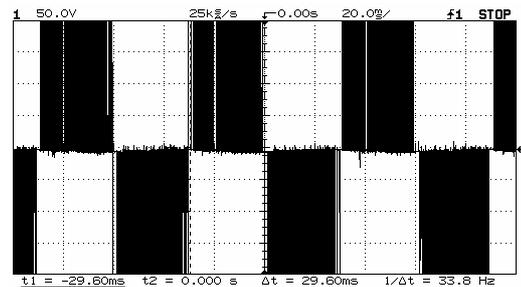


Figure 6.9(b) Measured u_{ab}^s in case of $f_0 = 16,9\text{Hz}$ and $V_p = 100\text{V}$

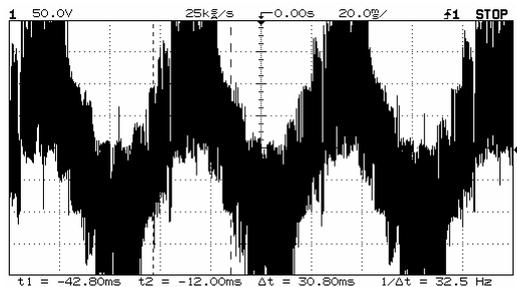


Figure 6.9(c). Measured u_a^s in case of $f_0 = 16,9\text{Hz}$ and $V_p = 100\text{V}$

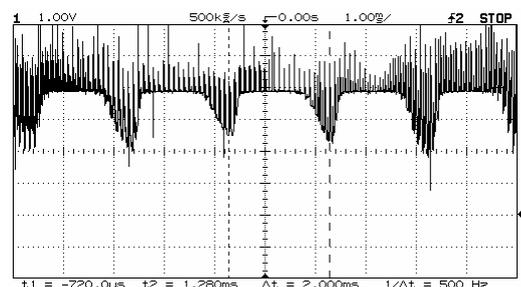


Figure 6.9(d) Measured ω in case of $f_0 = 16,9\text{Hz}$ and $V_p = 100\text{V}$

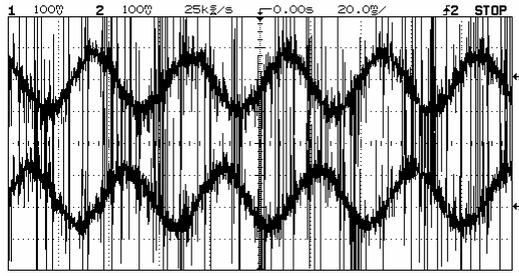


Figure 6.9 (e). Measured i_{sa}^s and i_{sb}^s in case of $f_0 = 25\text{Hz}$ and $V_p = 150\text{V}$

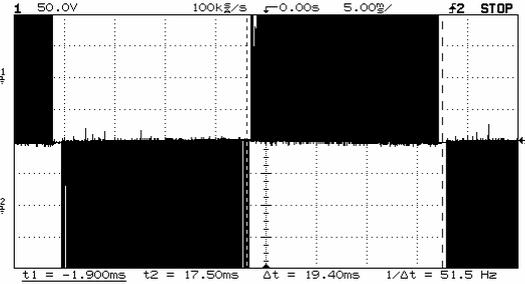


Figure 6.9(f) Measured u_{ab}^s in case of $f_0 = 25\text{Hz}$ and $V_p = 150\text{V}$

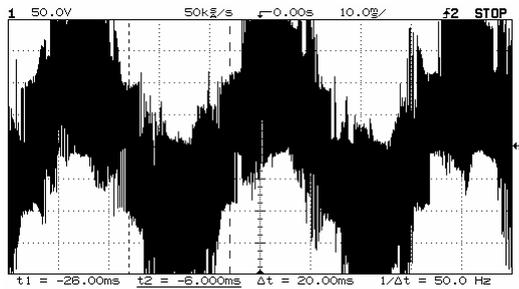


Figure 6.9(c). Measured u_a^s in case of $f_0 = 25\text{Hz}$ and $V_p = 150\text{V}$

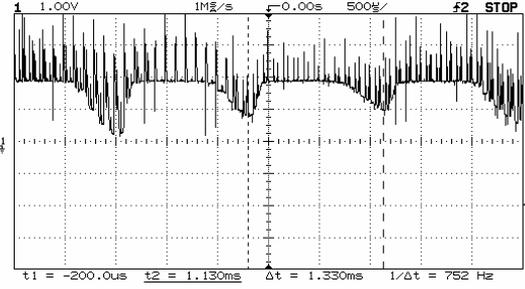


Figure 6.9(d) Measured ω in case of $f_0 = 25\text{Hz}$ and $V_p = 150\text{V}$

These results were also filtered before the identification begun to eliminate the noise effects which had risen because of the high voltage polluted environment. For the identification, the integrated signals of the measured PWM feed was used because the previously introduced mathematical modeling of the IM considers no effects of the slot and harmonic losses therefore there is no need to consider the effect of these step like changes.

6.3.3. Results of the identification

The identification algorithm was applied for each measured signal quartets (three-phase currents and the speed) of 6 different excitation frequencies and amplitudes. The computation time was large approximately 6 hours on a 950Mhz PC, and it resulted in the following parameter set which had 0,072 average error by the loss function computation.

$L_r = 172,1 \text{ mH}$	$R_s = 96,2 \ \Omega$	$J = 0.52 \text{ Nm}$	$c = 0.21 \ \frac{1}{\text{kgK}}$
$L_s = 352,4 \text{ mH}$	$R_0 = 30,1 \ \Omega$	$F = 0.04$	$m = 1,2 \text{ kg}$
$L_m = 245,3 \text{ mH}$	$p = 2$	$K_K = 3.5$	$T_0 = 18^\circ$

where K_k was given based on the scientific literature without identification, because the process which involves the obtaining of this value needs very fine transient measurements of the currents and voltages and for such measurements extremely sophisticated devices must be applied.

6.3.4. Validation of the model

The validation of the model was made in steady state by the help of the Matlab realized motor model with the obtained parameters. The investigation was performed to the case when $f_0 = 16,9\text{Hz}$ with $V_p = 150\text{V}$ was applied to the IM. By this validation process, the following result in *Figures 6.10(a-b)* could be concluded for the currents, with an average relative error of 9.09%, and a steady state relative error of 4,1% for speed. If it is considered, that the measured signals always contain noises, than these results can be evaluated to be very satisfying.

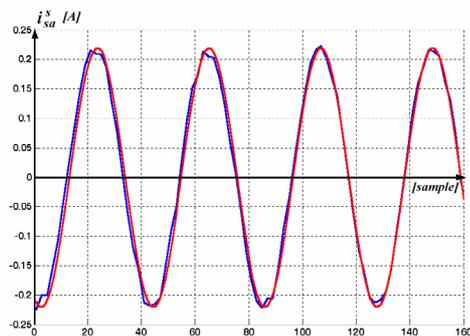


Figure 6.10(a). The real (blue) and simulated current responses of the considered motor

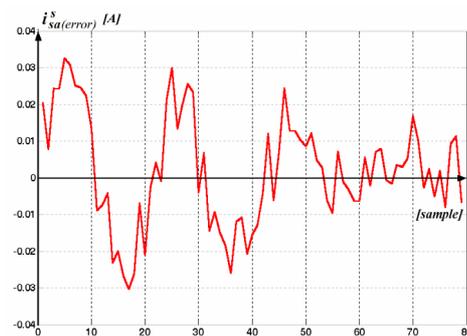


Figure 6.10(b). Relative error in current responses

In this way, it could be proved that the identified motor model describes well its real counterpart. Thus, the designed controller, which showed its effectiveness in simulations will control the motor well by the use of this information, if a fast realization of its algorithm can be given.

6.4. The program of the DSP

The previously mentioned controller structure was applied to the identified motor parameters and the redesigned structure gave similar results that were seen in *Section 6.2*. To realize this theoretical method several considerations were already given, with the brief description of the target drive. The last step that remained is to actually program the DSP of the drive to obtain run results.

Therefore, the algorithm was realized in C and in the following section the steps of this realization process is presented.

6.4.1. Flow chart of the program

The logical structure of the produced program given in *Figure 6.11*.

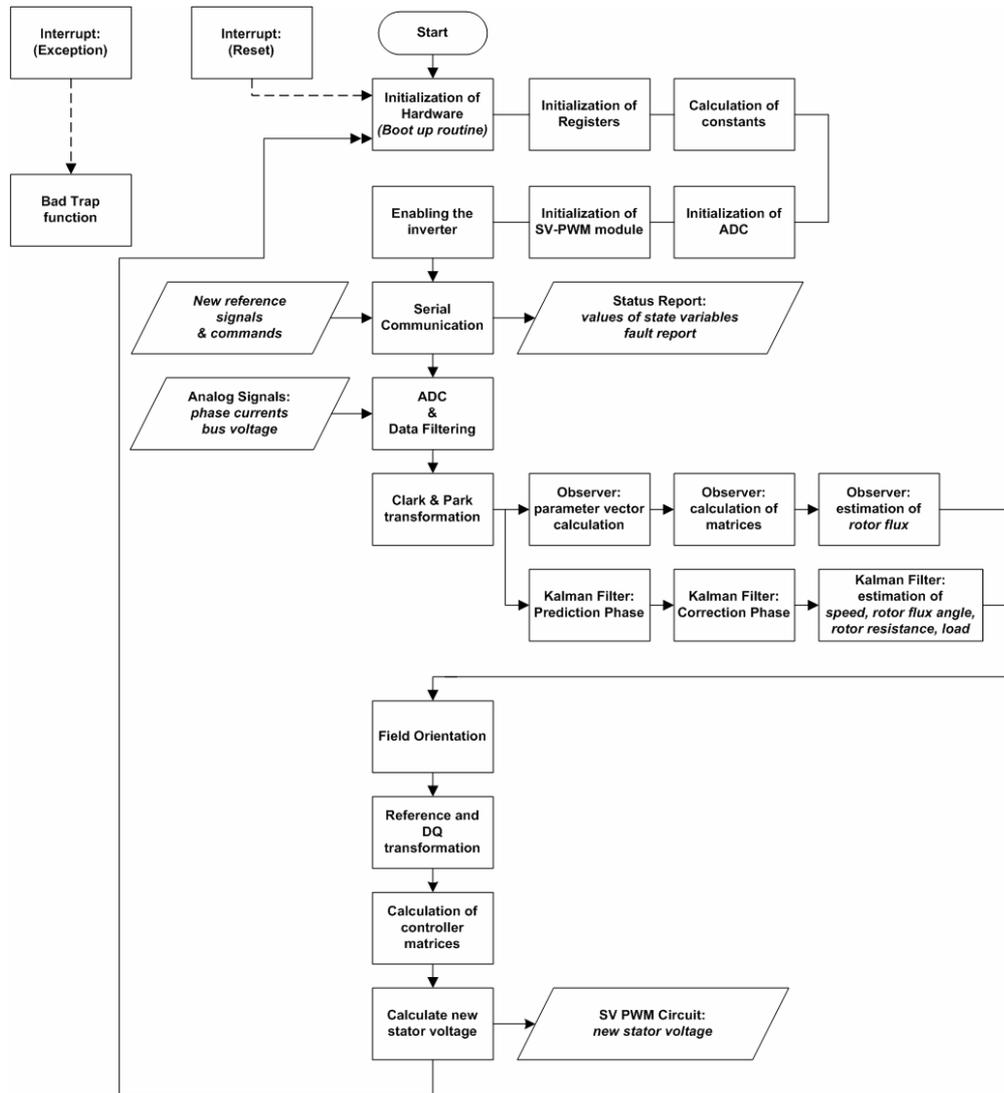


Figure 6.11. The flow chart of the implemented DSP program

This logical structure is totally identical with the structure of the already presented theoretical controller, except that it includes some initialization procedures, which are needed to ensure the correct calibration of the hardware to the task at hand. The logic blocks are realized through processes to make further modifications of the program easy and to use more sophisticated memory utilization by temporally variables. The program was also initially implemented in

such a way, that it guarantees the fastest running by saving of calculation time and using the special features of ANSI C.

The finally produced controller program is given with each of its used source files on an enclosed CD. To clarify the presented code, the functioning of the program is explained in the following parts in the viewpoint of its logical structure based on the previously given flowchart.

6.4.2. The hardware boot up

The first program block (*Boot.asm*) is responsible for the correct initialization of the hardware at every start-up. It is the modified version of the TI suggested boot-up routine, which included several other parts unnecessary for the problem at hand, and some additional code was also placed in it to realize the whole hardware set up in Assembly.

The program starts with a section defined by label *_c_int0*, which is the entry point of the code. In the vector table, given in *Vector.asm*, this label is defined as the service routine of the *reset* triggered interrupt, therefore in any case of reset event the program starts again from this point. Therefore, the drive is capable of continuous functioning, with restart handling ability.

As the program proceeds, it initializes the processor flags and disables all interrupts for the duration of the initialization process. Then it gives the correct sequence to the watch dog hardware to prevent this device generated hardware reset. After this, the processor speed, memory and I/O wait states are calibrated to their maximal affordable value to obtain the fastest machine cycle time. Then, all memory is defined to be local, the system is placed into microcontroller mode, and also every interrupt is masked and the pending interrupt requests are cleared. Then the set up of the memory stack proceeds with the copying of the program included constants into the data memory. This last step is crucial because during the compiling, all the constants included in the program are placed into the object file which is directly loaded into the program memory. Therefore, to be able to use these values, firstly they have to be copied into the data space. After this memory movement, the program calls the main procedure of the *Start.c* file which is the entry point of the C source code.

6.4.3. System initialization and hardwired constants

The C-implemented code begins with the *main* procedure which consists of the control cycle, so it must never return. If this ever happens, the abort function is called which hangs up the system. For similar reasons, if such an interrupt is acknowledged which is masked, then the service routines of these interrupts are routed to the *bad_trap* procedure which is similar to the previously mentioned one. These methods were very useful during the debugging of the code, because at any time when problem had risen, the running of program could be caught by these processes.

The main routine itself begins with some additional initialization processes. The first is *init_pre* which initializes all the key registers to zero and in this way it disables all output activity of the EVM. Then the *init_comp* is called which sets up the main constants in of the system. As it could have been seen in the previous sections, the controller is working with several constants, like the huge matrices of the H_∞ controller, the observer, etc., and of course there are other constant parameters which are needed to be obtained before the real calculation begins, to increase the performance. Therefore, these constants are defined in the first section of the program to fulfill this task and to also ease further modification of the code. Unfortunately, those constants which are calculated with the help of previously defined ones must be declared as variables, because in contrast with MS C++, the Code Composer does not support defining based on previously defined values. Because of this, all these constants are recalculated during the initialization process. This includes the step size based discretisation as well, so the calculations do not consists of any unnecessary operation. Moreover, the header files *All.h*, *Evm.h*, and *ADC.h* contains the definition of the registers allocated on specific memory addresses.

The initialization continues with *init_ADC*, which includes the initialization of the analog-to-digital converter. The device is calibrated to multi conversional mode so the needed phase currents and the bus voltage are obtained in two steps. Then the initialization of the PWM generator hardware is given with *init_PWM*. The PWM module is calibrated to 2 μ sec deadband and 10Mhz SV-PWM mode

with positive vector rotation and with GT1 as the associated timer. With *init_SCI*, the communication interface is also initialized. Finally, the program is ready for continuous operation as a controller so the inverter is enabled through the *drive_enable* routine provided output sequence. The first conversions are also started on the analog input channels and the LEDs of the EVM board are signed to inform about the end of the initialization phase.

6.4.4. Control loop

The control loop of the program runs continuously, and because all the initialization was already completed by the previous phases, it just repeats the same instructions in each of its cycle period. Since the analog signals were already converted in the initialization phase, the loop begins with *stator_oriented_curent_vec_from_3P* routine which converts the 3-phase current signal to the corresponding complex vector with the help of the (3.24) Clark transformation, then the *KalmanFilterZZ* is called which calculates the EKF. To ensure correct loop like functioning the result of the previously started ADC is read for i_{sa} and i_{sb} and the conversion for i_{sc} and U_{bus} is started. After this *Flux_obs* is called that proceeds with the estimation process by calculation of the H_∞ observer, which is directly followed by the DQ and reference transformation that are exactly implemented in the same way mentioned in *Section 6.1.3*. This is followed by the calculation of the H_∞ controller with the *H_inf_control* procedure and after this, the result of the previously started conversion is read. The conversion on the measured signals of i_{sa} and i_{sb} are also started again. Finally, the controller produced new \mathbf{u}_s^{fr} voltage control signal is transformed back to the stator-fixed reference frame and it is directly sent to the *set_voltage_pwm* process which realizes the SV-PWM generation. This is followed by the step with light on the LED row, which produces a running light spot that informs about correct functioning.

6.4.5. Process of ADC

As it was mentioned, the ADC is completed as paired conversions of the measured signals. The corresponding procedure *read_ADC(int mode)*, which at

first checks that whether the conversion was completed successfully or not, then it obtains the results from the FIFO of the ADC module. The procedure can be called in two modes, if mode 0 is selected than the read values are going to be assigned to i_{sa} and i_{sb} and the conversion of the other two signals are started. In case of mode 1, the values are assigned to i_{sc} and U_{bus} with conversion started on the other two current signals. Since the results are on the upper 10 bits of the registers, therefore shifting them by 6 bits to right is required. During the realization all the three phase current signals were used for obtaining information about the system to ensure better functioning and redundancy of data, which is important to eliminate some portion of the noise effects. However, the controller system is able to functioning with two measured currents as well. The obtaining of the bus voltage is also important, because as it is going to be showed this value is needed for the correct SV-PWM generation. Then, based on the previously mentioned offset and gain conditioning of the inverter interface card in *Section 5.3.2* the results are scaled back to get their real meaning. Because of some hardware uncertainties, even at zero voltage signal on the corresponding input pins, the obtained results do not had a zero mean value. Therefore, some offset conditioning was implemented in this process based on *try and error* to ensure correct functioning. The results are also filtered by an exponential filter with $\lambda_{\text{filter}} = 0.6$, whose Bode diagram is given in *Figure 6.12*.

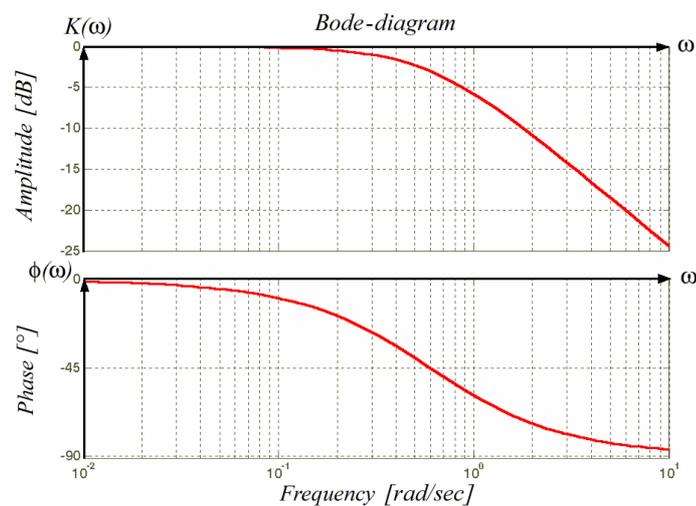


Figure 6.12. Bode-diagram of the software implemented filter of the ADC

This filter ensures the attenuation of measurement noises and other uncertainty effects of the conversion. The ADCs are also distributed to the beginning and to the end of the control loop, to guarantee them enough time to happen and to statistically provide more accurate measurements because of the redundant information stored in the three phase currents which are obtained at different time instants.

6.4.6. Processes of estimation

The EKF was implemented with great care to minimize the number of the needed operations. To achieve this, if a calculated value is used more than once during the computation then it is stored in a temporarily variable to save calculation time when it is next brought into play. Memory accesses of matrices were also reduced by using pointers, which produces indirect addressing in the interpreted ASM code. Incrementing of this pointer type index saves much more time compared to the calculation of the memory address based on the matrix indexes. With the previously given optimizations, the procedure of the EKF firstly calculates the numerical approximation of the states based on the Adams-Basforth method then it proceeds with the calculation of the Jacobbi matrix for these new states. From this, the predicted covariance matrix $\mathbf{P}(k|k-1)$ is calculated. Then for the statistical correction a \mathbf{K} matrix is computed which needs matrix inversion, which is calculated through the Gauss-Jordan elimination [14]. This elimination process is highly optimized from the view of calculation and memory access time as well. After this, the statistical correction of the predicted states and the $\mathbf{P}(k|k)$ is completed with the obtaining of the new estimated values.

This estimation is directly followed by the calculation of the H_ω observer, where at first the polytopic coordinates of the ω and R_r scheduling variables are computed. With the help of these coordinates, the new observer matrices are obtained and then the new observer states are computed as well. Because in the designed observer, the states correspond to the estimated values so, no output calculation is needed, therefore these values are directly provided. The calculation of this procedure is also optimized for speed.

6.4.7. Process of control

The controller calculation is done in a very similar manner as the calculation of the observer. The polytopic coordinates of ω , ω_{flux} , and R_r are obtained, from which the calculation of the controller matrices are done with optimized number of operation and memory accessing. Because of the mathematical structure of the controller, the input matrix is identical in each of its dimensions, only the output and state matrices must be calculated from the corner systems, provided by their polytopical description. This saves a lot of calculation time. The magnitude of the states is also checked and limited to 500 to ensure that the control does not become too aggressive. Finally, the new stator voltage is obtained, which is ready for realization.

6.4.8. Process of PWM generation

The previously calculated and then stator oriented real and imaginary part of the \mathbf{u}_s^{fr} is received by the *set_voltage_pwm* method, which task is to correctly adjust the registers of the SV-PWM hardware to produce the corresponding triggering signals of the IGBTs of the inverter. To achieve this, the calculation of T_1 and T_2 is needed, which is done with the vector based method that was mentioned in *Section 4.2.3.4*. This method also needs the calculation of the correct sector, which is obtained from the

$$\cos \alpha_{\text{sv}}(t) = \frac{u_{\text{s}\alpha}^{\text{s}}(t)}{\sqrt{(u_{\text{s}\alpha}^{\text{s}}(t))^2 + (u_{\text{s}\beta}^{\text{s}}(t))^2}}, \quad \sin \alpha_{\text{sv}}(t) = \frac{u_{\text{s}\beta}^{\text{s}}(t)}{\sqrt{(u_{\text{s}\alpha}^{\text{s}}(t))^2 + (u_{\text{s}\beta}^{\text{s}}(t))^2}} \quad (6.29)$$

values by the following logical process seen in *Figure 6.13*.

With the help of the sector code and the T_1 and T_2 values, the SV-PWM module can be correctly programmed to do its task. However it can happen that the amplitude of \mathbf{u}_s^{s} is more than the $U_{\text{bus}} / \sqrt{2}$ maximal voltage magnitude of SV-PWM method. In this case the process scales back the magnitude of the voltage vector to $U_{\text{bus}} / \sqrt{2}$, therefore it also ensures the limiting of the input voltage.

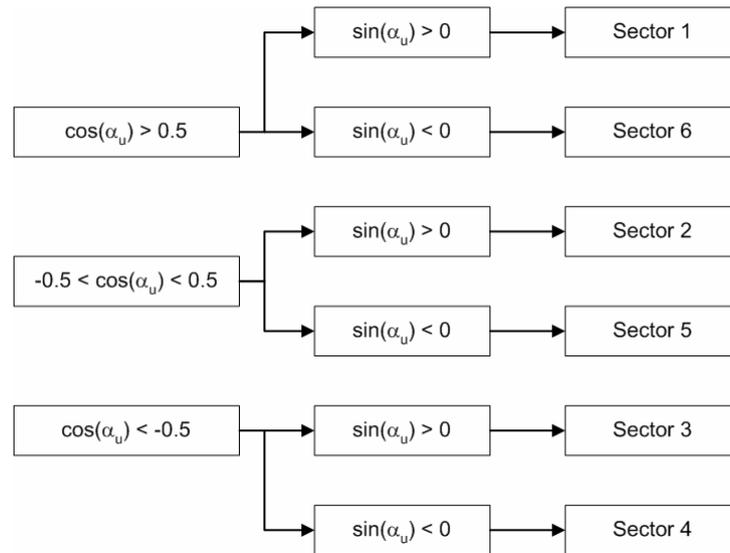


Figure 6.13. Process of sector calculation for SV-PWM

6.4.9. Additional features

The produced program is also able to communicate with the PC through the previously mentioned R232-C capability. This is solved through an interrupt handling routine, which accepts the received packets and updates the new control references. In every 100 calculation cycles, it provides information about the estimated speed and flux with the measured current and the used control voltage. The program which does this task in the PC side of the drive, is relatively simple. It only includes a port handling routine with a simple user interface to receive the new references from text windows and to represent the controller given values as well. Because of its simplicity, this program is not described in this paper.

6.5. Speed considerations

After completing the implementation and by obtaining maximally optimized code with the careful choice of optimization parameters of the Code Composer, some tests were taken to measure the running capabilities of the control loop. Unfortunately, it turned out, that the overall cycle time is 57.9 msec with all the possible optimizations. Moreover, 84.2% of the computation time is spent on the EKF, and 14% on the H_∞ controller procedure as it can be seen in *Figure 6.14*.

Sadly, this large cycle time prevents any tests to be taken on the functioning of the drive, because such great discretisation step will surely disrupt the estimation

processes and make the whole control to fail. Although, each of the blocks were tested separately to be correct mathematically and in functionality as well, finally it was concluded, the whole algorithm cannot be implemented in C for this drive.

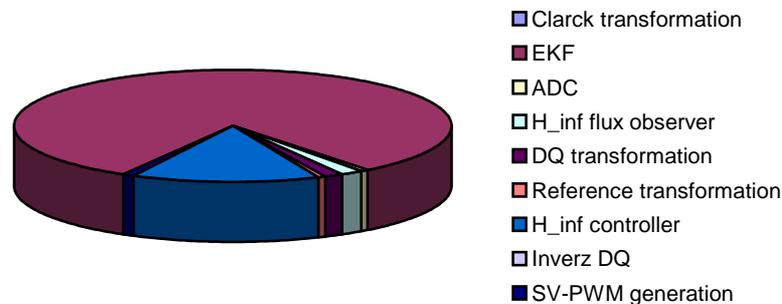


Figure 6.14. Time need of the implemented tasks in the DSP program

It is well known that, most of the microcontroller programs are written in Assembly, because obtaining the highest speed is essential for performance reasons. It is also widely accepted, that the automatic optimization of such code in C cannot overthrow the manually given optimized Assembly code, which averagely needs 50% less computational time. This can be explained by that only the designer knows exactly when the whole matrix calculation can be completed through one Assembly instruction provided memory copying for example, or which variables are needed to be left in the multiplication register for the next 10 calculation. It also worth to pay attention to that, with the scaling of the variables, which is a very laborious task and only worth the effort in Assembly, the whole calculation can be done with integers and not with fixed point values described floating point variables. This approach would surly give a boost for the whole process. In such way, the C-implemented controller is expected to reach a 6-8 msec control loop time, or even less, which would be in the acceptable range compared to the time constant of the motor. This implementation in Assembly is an ongoing process, but unfortunately still needs time to be completed, thus the real life performance of the whole controller can only be investigated when this happens.

TI has already realized that the 20Mhz speed is not enough for controllers with heavy computational load, therefore it introduced the 150Mhz speed providing TMS320F243, which is compatible in main aspects with the F243. This hardware, could even lessen the execution time, and give enough speed for the C implemented code as well. Unfortunately, its use is prevented by the enormous cost of the EVM which cannot be afforded by the University. There is also an increasing tendency in the microcontroller market to reach the performance of PCs because of the growing needs of control algorithms. Thus, TI has already introduced a microcontroller with 1,2Ghz speed and 8 parallel floating point multiplication/machine cycle, which gives the possibility to use extremely difficult methods for control. The usage of this processor is only prevented by the fact that is only sold to the US Army which uses it mainly in its new crafts, like advanced robot-airplanes.

7. Conclusion

In this thesis, the physical implementation of such a complicated control structure was aimed, which fulfills the high requirements of industrial researches on the field of induction motor drive control. Not only the creation of a speed sensorless operation capable system was tried to be achieved, but it was also guaranteed during the design, that this structure provides perfect functioning in noisy industrial environment as well.

To realize such a results, even in the first section, complicated mathematical representations were introduced to completely model the physical and dynamical properties of the induction motors. In this way, such an LPV model of the system was synthetised, which not only gives the description of the ideal operation of the machine, but it also pays attention to the parameter uncertainties of the physical system by modeling the rotor resistance variation to heat. To support our design considerations, the main dynamical properties of the motor was also investigated.

However, the difficulties of the described motor model and the aimed requirements in control have reached that point, when the common solutions of Control System Theory could not be applied ever more. Because of this, several unique and common methods were presented which are capable to solve the problem at hand with varying success.

This was followed by the thorough description of the common concepts for the power feed realization of IMs, with the mechanism of the nowadays widely used PWM inverters. The PWM realization was presented in details with several theoretical and practical considerations and with the outlining of the great possibilities of the SV-PWM approach.

After this, a self assembled Labdrive system, which heart is the TMS320F243 EVM, was investigated in both terms of functionality and concepts of controller realization. The opportunities offered by this device, which support the achievement of speed sensorless control, were mentioned with the careful analyzation of each of it sub modules.

Then, a designed control structure was introduced, which was synthesised by the help of theories presented in the previous sections. This system was investigated from several aspects in Matlab simulations. It turned out, that the self designed control and estimator structure has exceptional abilities in aspect of accuracy, robustness, and stability in noiseless and in noisy environment as well. In contrast with this, the estimation of the load torque could not be achieved in the given way with such dynamism, that would be expected in a widely used industrial application. Although, it was also showed that with constant applied torque, this control structure can be perfectly used for the implementation of an industrial drive. It is also important to note, that the accurate load torque estimation of IMs is still in the focus of ongoing researches in the scientific world.

In the last section, the implementation of the designed algorithm was explained by outlining the used features of the Labdrive system. The implementation was analyzed in details and it was completed in such a way which provides the most optimal realization of the designed controller. The motor parameters for this realization were also obtained by the help of the gradient method.

However, with the help of the parameter identification of the motor of this drive and by the writing of the code in C, the exact realization of the controller was achieved, the speed capabilities of the used DSP was not enough to obtain measured results of the performance. In this way, the goodness of the designed control system could be confirmed completely with this Labdrive device only in that case, when the whole code would be implemented in Assembly with the scaling of the variables. This type of implementation is an ongoing work and it holds great possibilities to realize the whole system perfectly.

Based on this, our future research plans concentrate on the achievement of this realization on the thoroughly analyzed Labdrive system with the synthesisation of such a rapid and accurate torque estimation method, that solves the dynamical problems of the presently used reference model and giving in this way such a sensorless control method, that can be used in wide variety of industrial applications.

References

- [1] Abbondanti A., and Brenemen M. B.: Variable Speed Induction Motor Drives use Electronic Slip Calculator Based on Motor Voltages and Currents, *IEEE Transactions on Industry Applications*, Vol. 29, No. 2, (1993), pp. 344-348.
- [2] Apkarian P., and P. Gahinet: A Convex Characterization of Gain-Scheduled H_∞ Controllers, *IEEE Transactions on Automatic Control*, Vol. 40, No. 5, (1995), pp. 856-864.
- [3] Åström K. J., and B. Wittenmark: Computer Controlled Systems, *Prentice-Hall*, New York: 1990.
- [4] Benchaib A., and C. Edwards: Nonlinear Sliding Mode Control of an Induction Motor, *International Journal of Adaptive Control and Signal Processing*, No. 14, (2000), pp. 201-221.
- [5] Bose B. K.: Adjustable Speed AC Drives – A Technology Status Review *Proceedings of the IEEE*, (1982).
- [6] Chapman S. J.: Electric Machinery Fundamentals, *WCB/McGraw-Hill*, New York: 1999.
- [7] Dettori M., and C. W. Scherer: LPV Design for a CD Player: An Experimental Evaluation of Performance, *Delft University of Technology*, The Netherlands (2002).
- [8] Depenbrock M.: Direkte Selbstregelung (DSR) für hochdynamische Drehfeldantriebe mit Stromrichterschaltung, *ETZA*, (1985), No. 7, pp. 211-218.
- [9] Ding G., X. Wang, and Z. Han: H_∞ Disturbance Attenuation Control of Induction Motor, *International Journal of Adaptive Control and Signal Processing*, No. 14, (2000), pp. 223-244.
- [10] Doki S., Sangwonwanich S., Yonemoto T., and Okuma S.: Implementation of Speed-Sensorless Field-Oriented Vector Control Using Adaptive Sliding Observers, *IECON, 16th Annual Conf. of the IEEE Industrial Electronics Society*, Asilomar/Ca. (1990), pp. 453-458.
- [11] Fodor D., L. Szalay, and K. Bíró: H_∞ Output Feedback Controller Design for AC Motor Control, *Proceedings of 10th International Power Electronics and Motion Conference, EPE-PEMC 2002*, Dubrovnik & Cavat.
- [12] Halász S.: Electrical drives (Villamos Hajtások), in Hungarian, *Műszaki Könyvkiadó*, 1993.
- [13] Hangos K., and G. Szederkényi: Parameter Identification of Dynamic Systems, *Veszprém University Press*, Veszprém: 1999.

- [14] Hartung F.: Introduction to Numerical Analysis (Bevezetés a Numerikus Analízisbe), in Hungarian, *Veszprém University Press*, Veszprém: 1998.
- [15] Henneberger G., Brunsbach B. J., and Klepsch T.: Field Oriented Control of Synchronous and Asynchronous Drives without Mechanical Sensors Using a Kalman-Filter, *EPE, European Conf. on Power Electronics and Applications*, Florence/It. (1991), pp. 3/664-671.
- [16] Hilairet M., C. Darengosse, F. Auger, and P. Chevrel: Synthesis and Analysis of Robust Flux Observers for Induction Machines, *Bd. de l'Université*, France (2000).
- [17] Holtz J.: Pulsewidth Modulation for Electronic Power Conversion, *Proceedings of IEEE*, (1995), Vol. 82 No. 8, pp. 1194-1214.
- [18] Holtz J.: Sensorless Control of Induction Drives, *Proceedings of the IEEE*, Vol. 90, No. 8, (2002), pp. 1259-1394.
- [19] Holtz J.: Sensorless Position Control of Induction Motors an Emerging Technology, *IEEE Transactions on Industrial Electronics*, Vol. 45, No. 6, (1998), pp. 840-852.
- [20] Holtz J.: The Representation of AC Machine Dynamics by Complex Signal Flow Graphs, *IEEE Transactions on Industrial Electronics*, Vol. 42, No. 3, (1995), pp. 263-271.
- [21] Jamniczky Á.: Operation of Electrical Machines (Villamos Gépek Üzemtana), in Hungarian, *Veszprém University Press*, Veszprém: 1984.
- [22] Jeon S. H., K. K. Oh, and J. Y. Choi: Flux Observer with Online Tuning of Strator and Rotor Resistances for Induction Motors, *IEEE Transactions on Industrial Electronics*, Vol. 49, No. 3, (2002), pp. 653-663.
- [23] Labdrive Interface Module, *Texas Instruments*, Technical Reference Guide: 2000.
- [24] Labdrive Inverter Module, *Texas Instruments*, Technical Reference Guide: 2000.
- [25] Kálmán R. E.: A New Approach to Linear Filtering and Prediction Problem, *Journal Basic Engineering*, Vol. 82, (1960), pp. 34-45
- [26] Kovács P.K.: Transient Phenomena in Electrical Machines, *Akadémiai kiadó*, Budapest: 1954.
- [27] Kubota H., Matsuse K., and Nakano T.: DSP Based Speed Adaptive Flux Observer for Induction Motor, *IEEE Transactions on Industry Applications*, Vol. 29, No. 2, (1993), pp. 354-358.
- [28] Lee C. H., M. H. Shin, and M. J. Chung: A Design of Gain-Scheduled Control for a Linear Parameter Varying System: An Application to Flight Control, *Control Engineering Practice*, No. 9, (2001), pp. 11-21.
- [29] Mamdani E. H.: Application of Fuzzy Algorithms for Simple Dynamic Plant, *Proceedings of IEEE*, (1979), No. 427, pp. 4585-4588.

- [30] Packard A.: Gain-Scheduling via Linear Fractional Transformations, *System Control Letters*, Vol. 22, (1994), pp. 79-92.
- [31] Premapain E., I. Postletwaite, and A. Benchaib: A Linear Parameter Variant H_{∞} Control Design for an Induction Motor, *Control Engineering Practice*, No. 10, (2002), pp. 663-644.
- [32] Rácz I.: Automated Electrical Drives (Automatizált Villamos Hajtások), in Hungarian, *Tankönyvkiadó kiadó*, 1976.
- [33] Sensorless Control with Kalman Filter on TMS320 Fixed-Point DSP, *Texas Instruments*, Application Note: 1997.
(<http://focus.ti.com/lit/an/bpra057/bpra057.pdf>)
- [34] Skogestad S., and I. Postletwaite: Multivariable Feedback Control, *John Wiley & Sons*, Chichester: 1996.
- [35] Schauder C.: Adaptive Speed Identification for Vector Control of Induction Motors without Rotational Transducers, *IEEE Industry Applications Society Annual Meeting*, San Diego Ca. (1989), pp. 493-499.
- [36] Takahashi I. and Noguchi T.: A New Quick Response and High Efficiency Control Strategy for An Induction Motor, *IEEE IAS Meeting*, Atlanta (1985), pp. 493-499.
- [37] TMS320F243/F241/C242 DSP Controllers Reference Guide, *Texas Instruments*, Technical Reference Guide: 2000.
(<http://focus.ti.com/lit/ug/spru276c/spru276c.pdf>)
- [38] Tóth, R., and G. Szederkényi: Nonlinear model analysis and analysis based control of the field oriented induction motor model, *XV. International Conference on System Science*, Wroclaw.
- [39] Tóth R. and Fodor D.: Speed Sensorless Mixed Sensitivity LPV H_{∞} Control of the Induction Motor, *11th European Conference of Power Electronics and Motion Control, EPE-PEMC*, Riga (2004).
- [40] Uray V.: Electrotechnics (Elektrotechnika), in Hungarian, *Műszaki könyvkiadó*, Budapest: 1970.
- [41] Vas P.: Sensorless Vector and Direct Torque Control, *Oxford University Press*, Oxford: 1998.
- [42] Zames G.: Feedback and Optimal Sensitivity: Model Reference Transformations, Multiplicative Seminorms and Approximate Inverse, *IEEE Transactions on Automatic Control*, Vol. 26, No. 1, (1981), pp. 301-320.
- [43] Zhenyu Yu., Space-Vector PWM with TMS320C24x/F24x Using Hardware and Software Determined Switching Patterns, *Texas Instruments*, Application Report:1999.
(<http://focus.ti.com/lit/an/spra524/spra524.pdf>)