DIPLOMA THESIS

Roland Tóth

2004

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

DIPLOMA THESIS

Speed Sensorless Mixed Sensitivity
LPV H_∞ Control of the Induction Motor

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**
5th year Information Technology student

Title of the thesis:

Speed Sensorless Mixed Sensitivity
LPV H_∞ Control of the Induction Motor

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, because as a dynamical system they have a highly nonlinear nature with parameter disturbances. Nowadays, therefore, there is a great interest in developing high performance and robust controllers 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 create a robust speed sensorless controller based on the recently appeared H_∞ theory and consider its implementation on a DSP target.

Thesis work assignments:

- study the H_∞ theory and other noted modern techniques (Kalman filter, Small Gain theory) in aspect of control and estimation,
- design a robust controller with noise attenuation as a feasible improvement, based on the LPV model of the Induction Motor,
- develop simulation in Matlab of the suggested algorithm to analyze the closed loop controller from the perspective of both stability and robustness,

- report on the results and outcomes of simulation, make comparisons and draw conclusions, and consider the implementation on a TMS320F243 DSP target.

Subject groups of the final examination:

- Rendszer- és irányításelmélet: (*System and Control Theories*)
    - Számítógépvezérelt szabályozások elmélete (*Computer Controlled Systems*)
    - Digitális Jelfeldolgozás (*Digital Signal Processing*)
- Irányításelméleti módszerek (*Control Techniques*)
    - Modern szabályozótervezési módszerek (*Modern Controller Design Techniques*)
    - Irányítási algoritmusok tervezése és alkalmazása I. (*Design and Application of Control Algorithms I.*)
- Dinamikus modellezés és modell analízis (*Dynamic system modeling and analysis*)
    - Folyamatmodellezés és analízis (*Modelling and Analysis of Process Flows*)
    - Dinamikus rendszerek paramétereinek becslése (*Parameter Estimation of Dynamical Systems*)

…….	...……..
Dénes Fodor Dr.	József Vass Dr.
Supervisor	Head of the Department

Nyilatkozat

Alulírott Tóth Roland, diplomázó hallgató, kijelentem, hogy a diplomadolgozatot a Veszprémi Egyetem Automatizálás Tanszékén készítettem el mérnök-informatikus diploma (Master of Engineering in Information Technology) megszerzése érdekében.

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

Tudomásul veszem azt, hogy a diplomadolgozatban 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 diplomadolgozat az aszinkronmotor súlyozott érzékenységű, lineáris paramétervariáns (LPV) H_∞-alapokon történő, robusztus fordulatszám-érzékelő nélküli szabályozását megvalósító struktúra tervezését mutatja be, támaszkodva a manapság megjelent legkorszerűbb elméleti eredményekre.

A bemutatásra kerülő szabályozó a motor direkt fluxus és fordulatszám szabályozását teszi lehetővé a teljes 4/4-es hajtási tartományon a 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 két fázis áram mérése), robusztus stabilitás és a zaj érzéketlenség.

A dolgozatban bemutatásra kerülnek a rendszer és szabályozás elméletnek a célkitűzésünk szempontjából legfontosabbnak ítélt modern vívmányai, mint a politopikus rendszer reprezentáción történő LPV H_∞-szabályozó és -megfigyelő szintézis, a súlyozott érzékenységű struktúrák, a kis erősítések tétele és a kiterjesztett Kálmán szűrők (EKF) teóriája. Ezen túlmenően szintén ismertetésre kerülnek azok az elgondolások, melyek a szabályozási struktúrát egyedivé teszik a rendszer külső kalibrálás lehetőségével, a szabályozási állapotok korlátozásával, és az EKF pontosságának növelése érdekében a predikciós fázis numerikus approximációjának Adams-Basforth módszerrel történő megoldásával. A fenti elméleti ismeretek alapján tervezett zárt szabályozó kör Matlab/Simulink környezetében készült implementációját mutatjuk be melyen elvégzett szimulációk eredményei alapján a struktúra az elvárásoknak megfelelően működött. A szabályozórendszer tényleges megvalósítását egy TMS320F243 DSP processzort tartalmazó fejlesztői környezetben is bemutatjuk, mely által vezérelt aszinkronmotoros hajtás eredményeit is szintén prezentáljuk. 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

ABSTRACT

The thesis shows the design of a robust control structure for the speed sensorless vector control of the induction motor, based on the mixed sensitivity linear parameter variant (LPV) H_∞ control theory that have recently appeared in this field.

The designed robust controller 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, robust stability, and noise attenuation. For the operation of this structure, only the measurements of two phase currents are needed, which provides all the required information about the system behavior.

The modern achievements of Control and Systems theory, such as the politopic system representation based H_∞ controller and observer synthesis, the mixed sensitivity structures, small gain theorem, and the theory of extended Kalman filters (EKF) are presented in aspect of their usage to solve the considered control problem. The self introduced modification of these theories, like the concepts to achieve external tuning of the controller, the limiting of the control states, and the substitution of numeric approximation in the prediction phase of the EKF with the Adams-Basforth method are also showed. Based on these theories and their modifications, 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. According to the simulation results the whole control system shows good dynamic and robust performance. Implementation of the designed control structure based on a DSP TMS320F243 development platform and a experimental laboratory drive is also presented with analysis of the received results. The experienced measurement and simulation outcomes 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

Table of contents

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]
ω(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]
$\omega_{rflux}(t)$	ang. speed of rotor flux [rad/sec]	l_s	stator winding phase inductivity [H]
$\omega_{ref}(t)$	ref. signal of rotor speed [rad/sec]	l_r	rotor winding phase inductivity [H]
$\omega_{nom}(t)$	nominal rotor speed [rad/sec]	l_f	field inductivity [H]
s(t)	slip	l_m	mutual inductivity [H]
$F_{sx}(\alpha, t)$	current density distribution in phase x [A/rad]	ρ(t)	angle between the stator fixed and the rotor fixed reference frame [rad]
$i_{sx}(t)$	stator current in phase x [A]	$\rho_k(t)$	angle between the stator fixed and the rotating reference frame [rad]
$u_{sx}(t)$	stator voltage in phase x [V]	$P_{mech}(t)$	mechanical power [W]
Ψ	magnetic flux [Wb]	$W_{mech}(t)$	mechanical energy [J]
φ_i	init. ang. of stat. current vec. [rad]	$W_e(t)$	electrical input power [J]
φ_u	init. ang. of stat. voltage vec. [rad]	$W_v(t)$	resistive power loss [J]
a	three-phase unity vector	$W_{field}(t)$	air gap power of magnetic field [J]
$\mathbf{x}^s$	space vector in stat. fixed rep.	$T_e(t)$	electromagnetic torque [Nm]
$\mathbf{x}^r$	space vector in rotor fixed rep.	$T_{e_{nom}}$	nominal electromag. torque [Nm]
$\mathbf{x}^{fr}$	space vect. in rotor field orient. rep.	$T_{load}(t)$	load torque [Nm]
$\mathbf{x}^k$	space vect. in rotating ref. frame	τ_{mech}	mechanical time constant [Nm]
x_d	real part of space vector x in rotating reference frame	τ_e	electrical time constant [Nm]
x_q	imaginary part of space vector x in rotating reference frame	τ_s	stator time constant (constant)
x_α	real part of space vector x in stator fixed reference frame	τ_r	rotor time constant (constant)
x_β	imaginary part of space vector x in stator fixed refernce frame	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_σ	notation [H]
$\mathbf{i}_r^x(t)$	space vector describing the overall rotor side cur. density distrib. [A]	T(t)	rotor temperature [K]

$\Psi_s^x(t)$	space vec. describing the overall stator side flux linkage distrib. [Wb]	T_0	initial rotor temperature [K]
$\Psi_r^x(t)$	space vec. describing the overall rotor side flux linkage distrib. [Wb]	$Q(t)$	heat [kcal]
$\mathbf{u}_s^x(t)$	space vector describing the overall stator side volt. density distrib. [V]	c	specific heat ct. Al [J/kgK]
$\mathbf{u}_r^x(t)$	space vector describing the overall rotor side volt. density distrib. [V]	m	weight of the rotor winding [kg]
$i_{eff}^r(t)$	rms value of the rotor phase cur. [A]	K_k	linear heat convection (constant)
$i_{eff}^s(t)$	rms value of the stator p. cur. [A]	F	fraction (constant)
$u_{eff}^s(t)$	rms value of the stator p. volt. [V]	h	discrete time step [sec]
P	considered system (*plant*) of the general problem definition	μ	proportional error term (constant)
$\breve{\mathbf{P}}$	mixed sensitivity system	$\varepsilon_{inv}(t)$	inverter noise [V]
G(**s**)	frequency transition func. of the sys.	$\varepsilon_{mes}(t)$	measurement noise [A]
x	state vector of the system	ε	small value
$\mathbf{x}(k\|k)$	estimated state vector based on k pervious estimations	t	time [sec]
$\mathbf{P}(k\|k)$	covariance of the error process	**I**	identity matrix
$\tilde{\mathbf{x}}$	error of the estimation	**K**	contr. of general problem definition
y	output vector of the system	γ	optimal H_∞ gain
u	control/general system input	v_1, v_2	virtual inputs
v	vector of measured output signals	**p**	parameter vector
z	system output vector (optimization)	$\hat{\mathbf{p}}$	normalized parameter vector
w	input disturbances of the system	$\mathbf{X}_K(t)$	matrices/vectors of the controller/observer
r	reference signals of the system	$\mathbf{W}_x$	filter of transfer function x
n / **d**	system noises and disturbances	$\mathbf{W}_I$	filter to approximate integrator term
e	error signals of the system	**S**	sensitivity transfer func.
R	covariance of measurement noises	**T**	inverse-sensitivity transfer func.
Q	covariance of system noises	**KS**	closed loop transfer function
f(**x**,**u**)	state function (non linear state space sys. description)	**F**	state space function
f(**x**)	state func. of an input-affine model	ξ	unity value $\in \{0,1\}$
$\mathbf{f}_d$(**x**)	discretised state function	ζ	unity base vector in $\mathbb{R}^n$
h(**x**)	output function (non linear state space sys. description)	κ_i	politopic coordinate (2^n)
$\mathbf{h}_d$(**x**)	discretised output function	ϑ_i	i^{th} parameter corner vector to define politopical descrip. of system **X**(**p**)
g(**u**)	input func. of an input-affine model	$\mathbf{g}_d$(**u**)	discretised input function
Ξ	sequence of measured values	$\hat{\mathbf{x}}$	predicted / normalized value
θ	parameter values	δ	scaling step size/Re. of comp. freq.
$\vec{\mathbf{G}}(.)$	first order gradient vector	$\vec{\mathbf{G}}_2(.)$	second order gradient vector
λ_{filter}	gain of the exponential filter	T_1, T_2	Modulation times in the SV-PWM

α	angle [rad]	α_u	angle of stator voltage vec. [rad]
$\mathbf{u}_{x \text{ or } x\pm60}$	voltage vect. of sector x (SV-PWM)	$u_{out}(t)$	realized phase voltage (SV-PWM)

(**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
($\mathbf{X}_\infty$, $\mathbf{Y}_\infty$,)	solutions of the Ricatti equation
($\mathbf{A}_\infty$, $\mathbf{B}_2$, $\mathbf{C}_2$, $\mathbf{F}_\infty$, $\mathbf{Z}_\infty$, $\mathbf{L}_\infty$)	matrices of the continuous state space representation of the controller

Abbreviations

AC	Analog Current/Voltage
ACG	Automated Code Generation
CAN	Controller Area Network
CPU	Central Processing Unit
DC	Direct Current/Voltage
DSP	Digital Signal Processor
EKF	Extended Kalman Filter
EVM	Evaluation Module
EVM2	Event Manager
GEL	General Extension Language
IDE	Integrated Development Environment
IGBT	Isolated Gate Bipolar Transistor
JTAG	Joint Test Action Group (IEEE 1149.1)
LMI	Linear Matrix Inequalities
LPV	Linear Parameter Variant (system)
LQG	Linear Quadratic Gaussian Control
LTI	Linear Time Invariant (system)
LTV	Linear Time Variant (system)
MIPS	Million Instructions Per Second
NL	Nonlinear (system)
NRZ	Non Return to Zero
PC	Personal Computer
PIC	Programmable Integrated Circuit
PWM	Pulse Width Modulation
RAM	Random-Access Memory
ROM	Read-Only Memory
RTDX	Real-Time Data Exchange
SCI	Serial Communication Interface
SPI	Serial Peripheral Interface
SV-PWM	Space Vector based Pulse Width Modulation
TI	Texas Instruments
UART	Universal Asynchronous Receiver/Transmitter
UML	Unified Modeling Language
VSI	Voltage Source Inverter
WD	Watch Dog
emf.	electromagnetic force
mmf.	magnetomotoric force

1. Introduction

In the past decades, the previously hardly controllable induction motors have become dominant on the field of variable speed electrical drives. The fast takeover of this machine can be explained by the significant improvements in control and system theories and by the important results of power electronics, like the IGBT and the cheap, reliable, and high frequency inverters. In these days, the use of induction motors has become crucially important for the industry and at the beginning of the XXI. Century these machines do not only symbolize the horsepower of production, but they also mean the future of transportation and even of aviation. The economical importance of the IM drives is also noteworthy with more than 12 billion US$ world market volume of which the annual growth rate is 15%. [16]

In spite of, their simple structure, low cost, and high reliability, the main disadvantage of the induction motors is that their control is significantly more challenging than the control of their *direct current* (DC) cousins, because as a dynamical system they have a highly nonlinear nature with parameter disturbances [27]. Moreover, the measurement of the extremely important strength and orientation of the magnetic field, described by the magnetic flux (Ψ), can only be expensively and inaccurately carried out and some of their parameters, like the variation of the rotor resistance (R_r), introduce large uncertainties into the system behavior. This is the reason why the IMs are still not rival to the DC motors in a number of high precision applications. For decades DC motors had been used for electrical drives because their control is relatively simple, but with the improvement of technology they were overtaken by the more power efficient IMs which need less maintenance because of the absence of graphite brushes. Only on those fields which need simple control and high precision preserved the DC motors their leading role, although there exist new DC designs with brushless construction. Nowadays, therefore, there is a great interest in developing high performance and robust controllers to make induction drives unbeatable in all fields of applications [15]. Especially, these efforts concentrate on controllers that do not need speed sensors to operate, which greatly reduces

costs and maintenance [18, 28]. 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, we show the design steps of a robust controller for speed sensorless operation of IMs. The designed system gives the opportunity of fast control of the speed of the motor 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. This system also possesses the ability to operate in noisy environment and the online adaptation to the load torque (T_{load}), which is significant for dynamic tasks. It also estimates the change of R_r to lower the effect of uncertainties. The implemented control law is based on the *linear parameter variant* (LPV) theory of H_∞ control with mixed sensitivity (MS), which has recently appeared in this field [8, 27]. The controller is supported by an I/O linearized reference model and a complex observer synthesized from an extended Kalman filter (EKF) [2, 19] and a H_∞ observer [13, 31]. This structure needs only the measurements of the stator currents, and it shows robustness with respect to electrical parameter uncertainties, system and measurement noises. Moreover, the proposed control law is designed to be easy to tune, which holds the possibility of the online tuning of the performance.

To be able to present a solution of the above mentioned control problem, firstly the dynamic mathematical model of the IM is introduced with the self constructed LPV system model of the motor. Then, the main dynamic properties of the given models are analyzed.

In the second part of this paper, those modern control and estimation theories are introduced which were used during the design of the controller. Furthermore, the applied modifications of these theories to improve the effectiveness of the solution of the control problem are also mentioned with their investigated effect on the whole system behavior.

After this, in the next part, the designed control structure is explained and it is examined in the view of stability and robustness.

Finally, the results of intensive numerical simulations are showed with the help of models and programs made in the *Simulink* environment of the *Matlab* to qualify the effectiveness of the closed loop control. Moreover, the TMS320F243 *digital signal processor* (DSP) based implementation of the whole controller is also presented on an experimental laboratory drive with the obtained results. The outcomes of simulations and measurements are also compared with other published results in this field. Finally, the further possibilities and 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 following section, the structure of the induction motors and their main features of functioning are introduced based on [4, 10, 14, 17, 20, 29].

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.

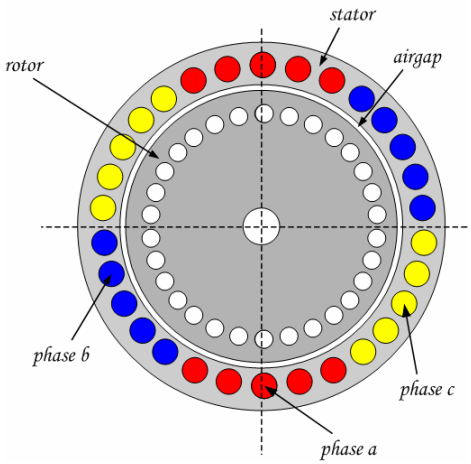


Figure 2.1. Schematics of the induction motor

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:

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

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:

$$s = \frac{n_0 - n}{n_0}. \tag{2.2}$$

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 [29]:

- 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 [20] 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 $\mathbf{i}^{s}_{sx}(t)$ complex vector whose orientation is determined by the direction of the respective phase axis and the current polarity [16]. In the presented case the positive phase current $\mathbf{i}^{s}_{sa}(t)$ 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.

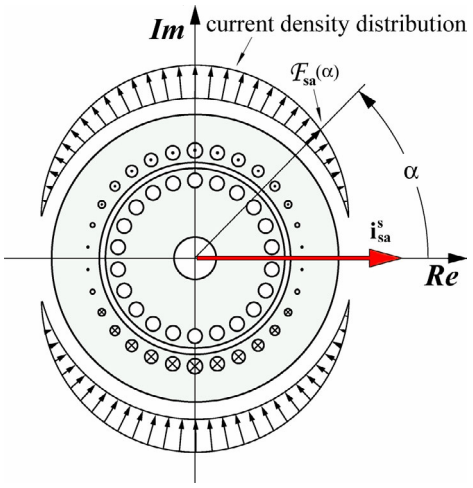


Figure 2.2. Current density distribution of phase a

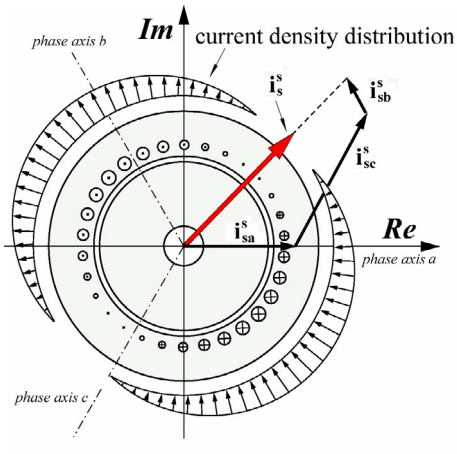


Figure 2.3. Stator side current density distribution

The total *magnetomotoric 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:

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

$$i_{sa}(t)=\mathrm{Re}\left\{\mathbf{i}^{s}_{s}(t)\right\},\quad i_{sb}(t)=\mathrm{Re}\left\{\mathbf{a}^{2}\cdot\mathbf{i}^{s}_{s}(t)\right\},\quad i_{sc}(t)=\mathrm{Re}\left\{\mathbf{a}\cdot\mathbf{i}^{s}_{s}(t)\right\}, \qquad (2.4)$$

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*.

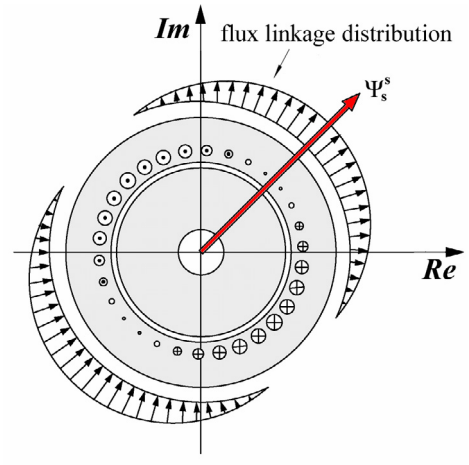


Figure 2.4. Stator side overall flux linkage distribution

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:

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

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

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:

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

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.).

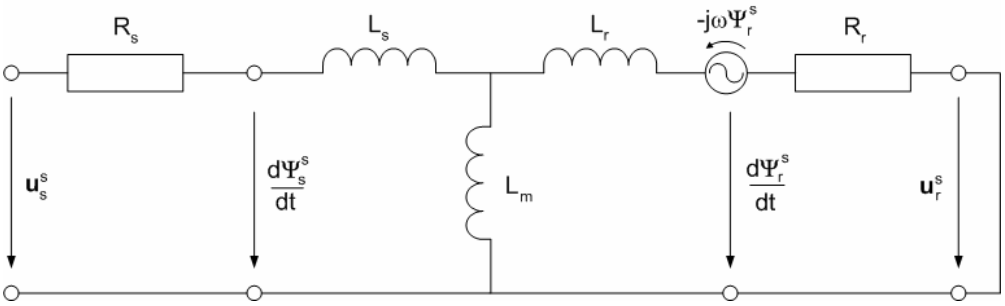


Figure 2.5. Complex transformator that describes the model of the IM

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

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

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.

$$\Psi_s^s(t) = \mathbf{i}_s^s(t) \cdot L_s + \mathbf{i}_r^r(t) \cdot L_m \cdot e^{j\rho(t)}, \tag{2.9}$$

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

where the presented $\rho(t)$ angle describes the position of the rotor compared to the axis of the stator, while $L_r = \frac{3}{2} l_r + \frac{3}{2} l_f$, $L_s = \frac{3}{2} l_s + \frac{3}{2} l_f$ are the three-phase inductances and l_s, l_r are the inductances of a stator and a rotor phase winding, l_f is the self inductance, and $L_m = \frac{3}{2} l_m$ is the mutual inductance between the stator and the rotor [29]. 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

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

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

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 [20]. For the mechanic power $P_{mech}(t)$ of the system, the following is true:

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

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

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

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

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

where

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

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

$\frac{dW_{field}(t)}{dt} = \frac{3}{2}\mathrm{Re}\left\{\frac{d\Psi_s^s}{dt}\mathbf{i}_s^{s*} + \frac{d\Psi_r^s}{dt}\mathbf{i}_r^{s*}\right\}$, 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:

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

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

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

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:

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

where $\tau_{mech} = J/p$ 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:

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

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 [34]. 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:

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

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).

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

By considering the heating properties of materials:

$$Q(t) = m \cdot c \cdot (T(t) - T_0), \tag{2.22}$$

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

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

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 synthetisation for asynchronous motors [35]. 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:

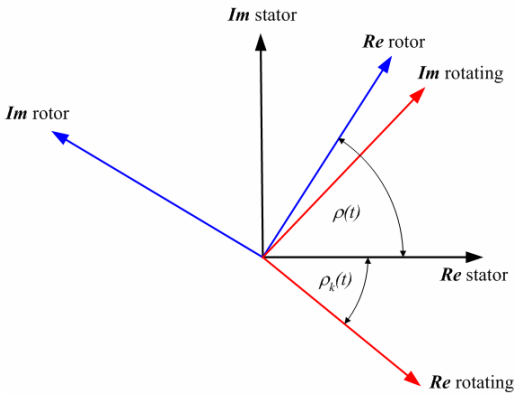


Figure 2.6. Rotating two-phase coordinate system

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

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

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:

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

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

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):

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

can be formulated with

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

from (2.27) and

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

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.

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

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

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

motion and the

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

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 $\omega_k = \omega_{rflux}$. 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, $\omega_{rflux} = \omega_0$, and (2.31) is transformed into the following form:

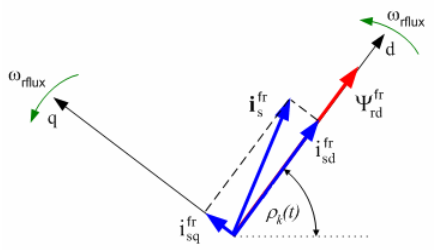


Figure 2.7. Rotor field orientation in two phase rotating reference frame

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

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

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

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

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

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 [23],. 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 **A**(.)…**D**(.) are dependent on a **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 [26] and Apkarian & Gahinet [1] 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 [5, 13, 18, 23, 27]. However, this approach does not always give better results than the common ad-hoc methods, like it is shown in [24].

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 R_r parameters:

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

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

where

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

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

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

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

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 [33], 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 $\left(T_e\right)$ 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* [16], 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 $\left(\omega_{nom}, T_{e_{nom}}\right)$ 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

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.

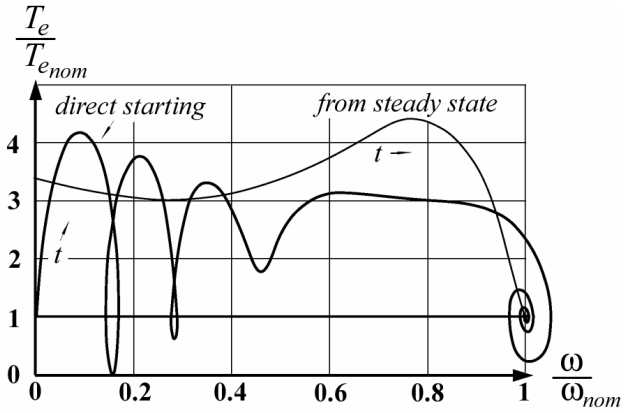


Figure 2.8. Torque / frequency transients of the motor in each dynamic interval

3. The mathematical theories and methods of the controller design

3.1. The H_∞ theory

3.1.1. The H_∞ norm

As maturing theories, H_∞ and H_2 norm-based controller synthetisations have become popular recently [6, 8, 23, 27], because with this type of feedback control, the optimal and robust control of the system can be reached. These theories were introduced by Zames [36] in his first paper trying to improve the bad robust properties of the *linear quadrant Gaussian* (LQG) control. To solve this problem, he introduced an integration constraint which was based on the norm of the system transfer function.

3.1 Definition: *For a given MIMO system, if the frequency transfer function of the* ***P****(****w****(t),****z****(t)) system is* ***G****(s) then the* H_∞ *norm of the system is defined by (3.1).*

$$\left\|\mathbf{G}(s)\right\|_\infty = \sup_\omega \left\|\mathbf{G}(j\omega)\right\|_2, \tag{3.1}$$

where for the complex frequency, $s = \delta + j\omega$ with $\delta = 0$, and $\| \ \|_2$ represents the Euclidian norm. So, the H_∞ norm gives the maximal gain of the system for the whole frequency spectrum, i.e. it gives the maximal gain of the output signal **z**(t) for any **w**(t) input signal:

$$\left\|\mathbf{G}(s)\right\|_\infty = \sup_{\|\mathbf{w}(t)\|_2 \neq 0} \frac{\left\|\mathbf{z}(t)\right\|_2}{\left\|\mathbf{w}(t)\right\|_2}, \quad t \in (0, \infty). \tag{3.2}$$

Besides, the (3.2) norm is the supreme of the structured singular value of the **G**(s) matrix as well. Because the H_∞ norm gives the worst case gain of the system, therefore provides a good match to engineering specifications. It also provides a useful description of the error and input control signals of the closed loop control, because the H_∞ norm gives an upper and lower bound on these signals on the whole frequency range. 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 [36]. Therefore, the minimization of this norm can be viewed as the maximization of the stability and goodness of the control. Thus, if any control and estimation problem can be described through the presented (3.2) norm, then it can be formulated as an H_∞ optimization problem.

3.1.2. The H_∞ optimization problem

Because there are several ways to transform the feedback-based control and estimation problems to H_∞ optimization tasks, it is important to introduce a general form of problem description by which any specific problem can be handled with ease. Let the plant model given in *Figure 3.1* is chosen for the general problem definition, where **w**(t) represents the input disturbances, noises, and reference signals of the system, while **u**(t) is the **K** controller generated input signal, **z**(t) is the vector function of output signals and control aims, and **v**(t) gives the measured information about the outputs of the **P** system.

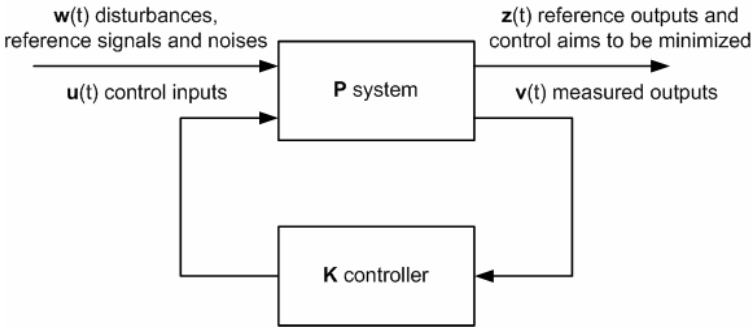


Figure 3.1. General problem definition for the H_∞ theory

The plant model presented in *Figure 3.1*, can be mathematically described by the following equation system:

$$\begin{bmatrix} \mathbf{z} \\ \mathbf{v} \end{bmatrix} = \mathbf{P}(s) \cdot \begin{bmatrix} \mathbf{w} \\ \mathbf{u} \end{bmatrix} = \begin{bmatrix} \mathbf{P}_{11}(s) & \mathbf{P}_{12}(s) \\ \mathbf{P}_{21}(s) & \mathbf{P}_{22}(s) \end{bmatrix} \cdot \begin{bmatrix} \mathbf{w} \\ \mathbf{u} \end{bmatrix} \tag{3.3}$$

$$\mathbf{u} = \mathbf{K}(s) \cdot \mathbf{v} \tag{3.4}$$

where **P**(s) is the transfer function of the open loop system. In the LTI case, this description can be given in a state space representation:

$$\mathbf{P} \overset{s}{=} \left[\begin{array}{c|cc} \mathbf{A} & \mathbf{B}_1 & \mathbf{B}_2 \\ \hline \mathbf{C}_1 & \mathbf{D}_{11} & \mathbf{D}_{12} \\ \mathbf{C}_2 & \mathbf{D}_{21} & \mathbf{D}_{22} \end{array}\right] \quad \Rightarrow \quad \begin{aligned} \dot{\mathbf{x}}(t) &= \mathbf{A}\mathbf{x}(t) + \mathbf{B}_1\mathbf{w}(t) + \mathbf{B}_2\mathbf{u}(t) \\ \mathbf{z}(t) &= \mathbf{C}_1\mathbf{x}(t) + \mathbf{D}_{11}\mathbf{w}(t) + \mathbf{D}_{12}\mathbf{u}(t) \\ \mathbf{v}(t) &= \mathbf{C}_2\mathbf{x}(t) + \mathbf{D}_{21}\mathbf{w}(t) + \mathbf{D}_{22}\mathbf{u}(t) \end{aligned} \tag{3.5}$$

Based on (3.3), the transfer function between **w**(t) and **z**(t) is as follows:

$$\mathbf{G}(\mathbf{P},\mathbf{K})(s) = \mathbf{P}_{11}(s) + \mathbf{P}_{12}(s)\mathbf{K}(s)(\mathbf{I} - \mathbf{P}_{22}(s)\mathbf{K}(s))^{-1}\mathbf{P}_{21}(s). \tag{3.6}$$

For this general description the optimization problem can be defined:

3.2 Definition: *The H_∞ optimization problem represents the search for such a optimal, robust and causal controller **K** that stabilizes the **P** system and minimizes the $\|\mathbf{G}(\mathbf{P},\mathbf{K})\|_\infty$ system norm.*

Instead of a direct minimization, which is a difficult task, most commonly we are looking for a suboptimal, robust, and stabilizing controller that fulfills (3.7).

$$\|\mathbf{G}(\mathbf{P},\mathbf{K})\|_\infty < \gamma, \tag{3.7}$$

where $\gamma \in \mathbb{R}_0^+$ is a given value. In this case the (3.7) control aim is in accordance with (3.8).

$$\frac{\|\mathbf{z}(t)\|_2}{\|\mathbf{w}(t)\|_2} < \gamma \quad \text{or} \quad L(\mathbf{w}(t),\mathbf{u}(t)) = \|\mathbf{z}(t)\|_2^2 - \gamma^2 \|\mathbf{w}(t)\|_2 < 0, \;\; t \in (0,\infty), \;\; \forall \|\mathbf{w}(t)\|_2 \neq 0 \tag{3.8}$$

This case can be reformulated to the linear-quadratic *minimax* problem, which is well known from the games theory. By this approach, we are looking for such a **K** that fulfills

$$\max_{\|\mathbf{w}(t)\|_2 \neq 0} \left\{ \min_{\mathbf{u}(t)=\mathbf{K}\cdot\mathbf{v}(t)} \{L(\mathbf{w}(t),\mathbf{u}(t))\} \right\}. \tag{3.9}$$

3.1.3. Solution of the H_∞ control problem

The recently popular solutions of the (3.9) problem are based on the work of Glover and Doyle [7]. Basically, two ways of the solutions are distinguished: one includes the methods based on solution of Ricatti-equations and the other group is built up from the linear *matrix inequalities* (LMIs) based approaches. In the following the Ricatti based solution is shown, because it perfectly presents the

logic steps of the controller synthetisation. Unfortunately, the LMI method is too complicated to be presented in this paper, because its formal description would cover up another full thesis. Let it is assumed, that the system is given in the (3.3) form, and the following are true:

(C1) $(\mathbf{A}, \mathbf{B}_1)$ is controllable and $(\mathbf{C}_1, \mathbf{A})$ is observable

(C2) $(\mathbf{A}, \mathbf{B}_2)$ is stabilizable and $(\mathbf{C}_2, \mathbf{A})$ is state detectable

(C3) $\mathbf{D}_{12}^{\mathrm{T}}\mathbf{D}_{12} = \mathbf{I}$ and $\mathbf{D}_{21}^{\mathrm{T}}\mathbf{D}_{21} = \mathbf{I}$

(C4) $\mathbf{D}_{12}\mathbf{C}_1 = \mathbf{0}$ and $\mathbf{D}_{12}\mathbf{B}_1^{\mathrm{T}} = \mathbf{0}$

The (C3-C4) conditions are needed to distinguish the incoming **w**(t) disturbances from the **v**(t) measured values of the **P** system, i.e. to separate the **x**(t) state dependent and **u**(t) controller dependent part of **z**(t). Moreover, the (C2) condition is introduced to guarantee the feedback stabilization ability of the **P** system, while (C1) guarantees the existence of the **K** controller for some $\gamma \in \mathbb{R}_0^+$ and assumes that the plant is controllable from the disturbance input and observable from the reference outputs. Of course, there exist such modified methods by which condition (C3) can be relaxed. However, the above mentioned conditions seem to be highly strict, many of the real life applications fulfill these, because with the normalization of the inputs and outputs (C3) can be easily satisfied.

Unfortunately, only such a **K** controller can be designed which fulfills (3.7) in finite time because of the solution method of the Ricatti equations. In finite time, the produced controller is a linear time variant (LTV) system that asymptotically tends to an LTI controller after the transient processes of the **P** system. Because in most applications the desired controller has to operate over time periods that are long compared to the transients, it is worthy to design the controller to achieve the minimization of only the steady-state gains. By using the steady-state gains the controller becomes time invariant which allows the use of many robustness and performance analysis techniques that are not applicable to LTV systems. Based on this, let the solution is considered only for steady-state:

3.1 Theorem: *If (C1-C4) are satisfied, then there exists such a stabilizing suboptimal controller* ***K****(s) that (3.7) is fulfilled in finite time for some* $\gamma \in \mathbb{R}_0^+$ *when the following conditions are held:*

(i) There exists a symmetric and positive (semi)definite $\mathbf{X}_\infty$ *solution of the*

$$\mathbf{A}^{\mathrm{T}}\mathbf{X}_\infty + \mathbf{X}_\infty\mathbf{A} + \mathbf{C}_1^{\mathrm{T}}\mathbf{C}_1 + \mathbf{X}_\infty\left(\gamma^{-2}\mathbf{B}_1\mathbf{B}_1^{\mathrm{T}} - \mathbf{B}_2\mathbf{B}_2^{\mathrm{T}}\right)\mathbf{X}_\infty = 0, \tag{3.10}$$

Ricatti-equation, and that $\mathbf{A} + \left(\gamma^{-2}\mathbf{B}_1\mathbf{B}_1^{\mathrm{T}} - \mathbf{B}_2\mathbf{B}_2^{\mathrm{T}}\right)\mathbf{X}_\infty$ *is stable.*

(ii) There exists a symmetric and positive (semi)definite $\mathbf{Y}_\infty$ *solution of the*

$$\mathbf{A}^{\mathrm{T}}\mathbf{Y}_\infty + \mathbf{Y}_\infty\mathbf{A} + \mathbf{B}_1\mathbf{B}_1^{\mathrm{T}} + \mathbf{Y}_\infty\left(\gamma^{-2}\mathbf{C}_1^{\mathrm{T}}\mathbf{C}_1 - \mathbf{C}_2^{\mathrm{T}}\mathbf{C}_2\right)\mathbf{Y}_\infty = 0, \tag{3.11}$$

Ricatti-equation, and $\mathbf{A} + \mathbf{Y}_\infty\left(\gamma^{-2}\mathbf{C}_1^{\mathrm{T}}\mathbf{C}_1 - \mathbf{C}_2^{\mathrm{T}}\mathbf{C}_2\right)$ *is stable.*

(iii) $\rho(\mathbf{X}_\infty\mathbf{Y}_\infty) < \gamma^2$*, the spectral radius of solutions is bounded.*

In this way, the calculated controller can be given in the following state space representation:

$$\mathbf{K}(s) \overset{s}{=} \left[\begin{array}{c|cc} \mathbf{A}_\infty & -\mathbf{Z}_\infty\mathbf{L}_\infty & \mathbf{Z}_\infty\mathbf{B}_2 \\ \hline \mathbf{F}_\infty & \mathbf{0} & \mathbf{I} \\ -\mathbf{C}_2 & \mathbf{I} & \mathbf{0} \end{array}\right], \text{ where } \begin{array}{c} \mathbf{F}_\infty = -\mathbf{B}_2^{\mathrm{T}}\mathbf{X}_\infty, \;\; \mathbf{L}_\infty = -\mathbf{Y}_\infty\mathbf{C}_2^{\mathrm{T}} \\ \mathbf{Z}_\infty = \left(\mathbf{I} - \gamma^{-2}\mathbf{Y}_\infty\mathbf{X}_\infty\right)^{-1} \\ \mathbf{A}_\infty = \mathbf{A} + \gamma^{-2}\mathbf{B}_1\mathbf{B}_1^{\mathrm{T}}\mathbf{X}_\infty + \mathbf{B}_2\mathbf{F}_\infty + \mathbf{Z}_\infty\mathbf{L}_\infty\mathbf{C}_2 \end{array} \tag{3.12}$$

By *Theorem 3.1*, the H_∞ suboptimal controller can be separated to a steady state H_∞ full information control (3.10) and a steady state H_∞ estimator which can be designed independently if their combined solution fulfills (*iii*). Naturally, the goodness of the obtained controller strongly depends on how accurately the lowest reachable overall gain, the optimal γ can be achieved. In practice, most commonly recursive algorithms are used based on try and error to get as near to this optimal value as possible. These methods are called γ-iterations.

3.1.4. Extension of the H_∞ control problem to LPV systems

Although, the previously described theory provides robust control for only LTI systems, the theoretical possibilities of this control synthetisation can be used for

LPV systems as well by introducing some modifications [1, 36]. In *section 1.5*, it has already been stated that in the LPV case, LTI controllers can be designed point by point for a given **P**(**p**) defined in an appropriate system space. This approach theoretically provides that in every point of the parameter space the control aims given for the LTI case are satisfied. For this reason, the H_∞ control structure is also applicable for such systems. However, it would be an extremely difficult task to solve equations (3.10) and (3.11) in every point of the system space. Thus, **P** must be transformed to such form which can be described with a hypercube defined by 2^n corner systems in an *n* dimensional system space, if the parameter vector is bounded in each of its coordinates and **P** is parameter-affine. This hyper cube corresponds to a convex polytope defined in a 2 dimensional system space, which representation is called the polytopic form of the LPV system. The corner points of such a polytope are those system points which are defined by all of the possible combinations of the bounds of the parameters.

Let a $\mathbf{p}(t) \in \mathbb{R}^n$ parameter vector dependent system is considered:

$$\begin{aligned} \mathbf{E}(\mathbf{p})\dot{\mathbf{x}} &= \mathbf{A}(\mathbf{p})\mathbf{x} + \mathbf{B}(\mathbf{p})\mathbf{u} \\ \mathbf{y} &= \mathbf{C}(\mathbf{p})\mathbf{x} + \mathbf{D}(\mathbf{p})\mathbf{u} \end{aligned} \tag{3.13}$$

given in state space representation. Where $\mathbf{A}(.)$ … $\mathbf{E}(.)$ are the functions of the **p** vector. The (3.13) system is called parameter-affine if all of the $\mathbf{A}(.)$ … $\mathbf{E}(.)$ system matrices can be given in a $\mathbf{X}(\mathbf{p}) = \mathbf{X}_0 + \mathbf{X}_1(p_1) + \ldots \mathbf{X}_n(p_n)$ form. Moreover, if the **p**(t) vector function is bounded in each of its coordinates, then its image is an *n* dimensional cube in $\mathbb{R}^n$ (see *Figure 3.2(a)*), which corresponds to a 2 dimensional polytope (see *Figure 3.2(b)*) that defines a 2 dimensional polytope of the LPV system (see *Figure 3.2(c)*).

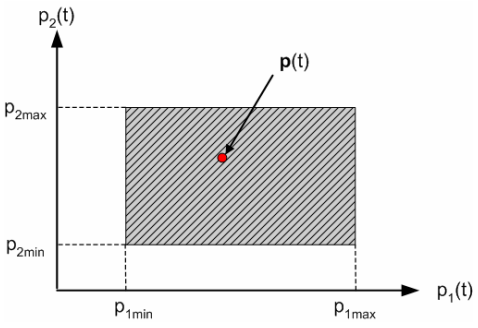


Figure 3.2(a). p(t) defined hypercube, if dim(p(t)) = 2

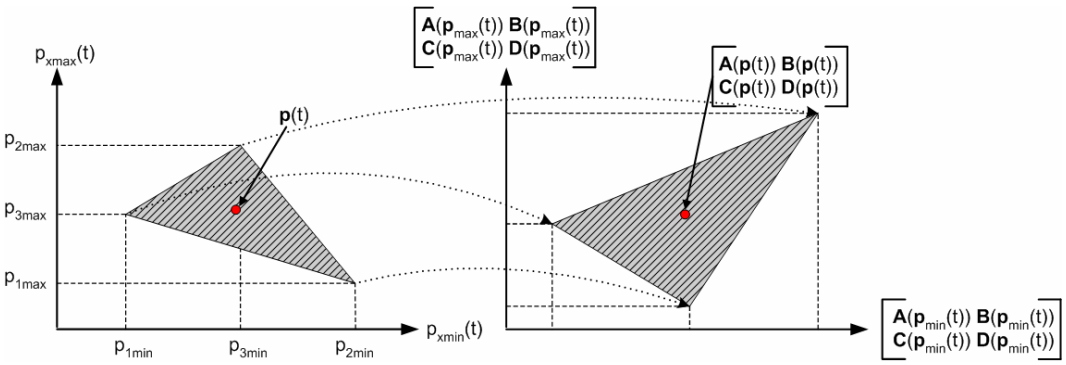


Figure 3.2(b) p(t) defined polytope, if dim(p(t)) = 3

Figure 3.2(c) p(t) defined polytopic representation of system P

If this *n* dimensional hypercube is normalized, then it is going to be a unity cube, whose corners are defined by the

$$\zeta_i = \left[\xi_1,\ldots,\xi_{k-1},\xi_k,\xi_{k+1},\ldots,\xi_n\right]^T,\ \ \xi_k \in \{0,1\},\ i = 1\ldots 2^n \tag{3.14}$$

bases vectors, and if

$$\hat{\mathbf{p}}(t) = \left[\frac{p_1(t)-p_{1\min}}{\underbrace{p_{1\max}-p_{1\min}}_{v_1(t)}},\ldots,\frac{p_n(t)-p_{n\min}}{\underbrace{p_{n\max}-p_{n\min}}_{v_n(t)}}\right]^T \tag{3.15}$$

is the normalized parameter vector, then

$$\hat{\mathbf{p}}(t) \in \sum_{i=1}^{2^n} \kappa_i(t)\zeta_i, \tag{3.16}$$

where $\left[\kappa_1(t),\ldots,\kappa_{2^n}(t)\right] = \tilde{\mathbf{v}}_n = \left[\tilde{\mathbf{v}}_{n-1}\cdot(1-v_{n-1}),\tilde{\mathbf{v}}_{n-1}\cdot v_{n-1}\right]$ and $\tilde{\mathbf{v}}_0$=1. Moreover, for the basis weights: $1 = \sum_{i=1}^{2^n}\kappa_i(t)$. Let the (3.5) problem is reformulated in the following LPV form:

$$\begin{aligned}\dot{\mathbf{x}}(t) &= \mathbf{A}(\mathbf{p})\mathbf{x}(t)+\mathbf{B}_1(\mathbf{p})\mathbf{w}(t)+\mathbf{B}_2(\mathbf{p})\mathbf{u}(t)\\ \mathbf{z}(t) &= \mathbf{C}_1(\mathbf{p})\mathbf{x}(t)+\mathbf{D}_{11}(\mathbf{p})\mathbf{w}(t)+\mathbf{D}_{12}(\mathbf{p})\mathbf{u}(t)\\ \mathbf{v}(t) &= \mathbf{C}_2(\mathbf{p})\mathbf{x}(t)+\mathbf{D}_{21}(\mathbf{p})\mathbf{w}(t)+\mathbf{D}_{22}(\mathbf{p})\mathbf{u}(t)\end{aligned} \tag{3.17}$$

In this case (3.17) can be transformed to a polytopic form, satisfying (3.13) with $\mathbf{E}(\mathbf{p}) = \mathbf{I}$:

$$\mathbf{P}(\mathbf{p}) = \left[\begin{array}{c|cc} \mathbf{A}(\mathbf{p}) & \mathbf{B}_1(\mathbf{p}) & \mathbf{B}_2(\mathbf{p}) \\ \hline \mathbf{C}_1(\mathbf{p}) & \mathbf{D}_{11}(\mathbf{p}) & \mathbf{D}_{12}(\mathbf{p}) \\ \mathbf{C}_2(\mathbf{p}) & \mathbf{D}_{21}(\mathbf{p}) & \mathbf{D}_{22}(\mathbf{p}) \end{array}\right] \tag{3.18}$$

Based on the linearity of the parameter dependence, for (3.18), the following is true:

$$\mathbf{P}(\mathbf{p}) = \sum_{i=1}^{2^n} \kappa_i(t)\mathbf{P}(\vartheta_i), \quad \text{where} \quad \vartheta_i = \zeta_i \cdot \sum_{k=1}^{n} p_{k\min}\mathbf{e}_k + (\mathbf{I}_{2^n\times 1} - \zeta_i)p_{k\max}\mathbf{e}_k. \tag{3.19}$$

Here $\mathbf{e}_k$ denotes the orthonormal bases vectors of $\mathbb{R}^n$. If $\mathbf{D}_{12}$, $\mathbf{D}_{21}$, $\mathbf{D}_{22}$ are p independent constant matrices, and for any $i \in \{1,\ldots,2^n\}$ the $\mathbf{P}(\vartheta_i)$ fulfills the (C1-C2) conditions, then it is possible to design for every $\mathbf{P}(\vartheta_i)$ LTI corner system such a $\mathbf{K}(\vartheta_i)$ H_∞ controller that satisfies (3.7) with an overall $\gamma \in \mathbb{R}_0^+$. Moreover, it is also true that for any **p**(t) vector which is represented in the form of (3.16), the interpolation of the $\mathbf{K}(\vartheta_i)$ systems produces a

$$\mathbf{K}(\mathbf{p}) = \sum_{i=1}^{2^n} \kappa_i(t)\mathbf{K}(\vartheta_i), \tag{3.20}$$

LPV controller that satisfies (3.7) with the given γ, if p(t) is a smooth function [1]. This problem can be efficiently solved through LMIs, which method is not covered by this paper.

3.1.5. Control structures with mixed sensitivity

If frequency filters are introduced to specify the frequency domains of the signals of the **P** system, then not only a more accurate description of the plant can be given, but it is also possible to mathematically define the control aims by the definition of the frequency characteristics of the sensitivity $\mathbf{S} = (\mathbf{I} + \mathbf{GK})^{-1}$, the inverse sensitivity $\mathbf{T} = \mathbf{I} - \mathbf{S}$, and the closed loop **KS** transfer functions. This mixed sensitivity approach means such a control problem (*Figure 3.3*) which, by manipulation of the introduced filters, aims the achievement of a more efficient H_∞ control with a smaller γ.

For the closed loop system in *Figure 3.3* the following state equations hold:

$$\mathbf{y}(s) = \mathbf{T}(s)\mathbf{r}(s) + \mathbf{S}(s)\mathbf{d}(s) - \mathbf{T}(s)\mathbf{n}(s), \tag{3.21}$$

$$\mathbf{u}(s) = \mathbf{K}(s)\mathbf{S}(s)\left[\mathbf{r}(s) - \mathbf{n}(s) - \mathbf{d}(s)\right], \tag{3.22}$$

$$\mathbf{e}(s) = -\mathbf{S}(s)\mathbf{r}(s) + \mathbf{S}(s)\mathbf{G}(s)\mathbf{d}(s) - \mathbf{T}(s)\mathbf{n}(s), \tag{3.23}$$

where **r** is the reference signal, **e** is the error signal, **d** is the disturbance, **n** is the measurement noise, and **y** is the output signal of the system.

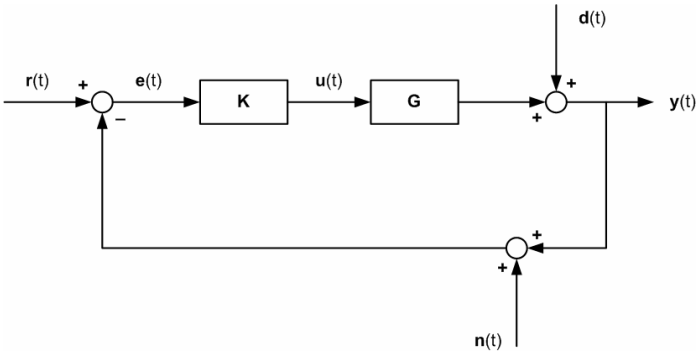


Figure 3.3. One degree of freedom control structure

These relationships determine several closed loop objectives (O1-O6), in addition to the requirement, that **K**(s) must stabilize **G**(s).

(O1) For good disturbance attenuation, $\bar{\sigma}(\mathbf{S})$ must be a small value;

(O2) For good noise attenuation, $\bar{\sigma}(\mathbf{T})$ must be a small value;

(O3) For good reference tracking, $\bar{\sigma}(\mathbf{T}) \approx \underline{\sigma}(\mathbf{T}) \approx 1$;

(O4) For control energy reduction, $\bar{\sigma}(\mathbf{KS})$ must be a small value;

(O5) For robust stability in the presence of adaptive output perturbation, $\bar{\sigma}(\mathbf{KS})$ must be a small value;

(O6) For robust stability in the presence of multiplicative output perturbation, $\bar{\sigma}(\mathbf{T})$ must be a small value;

where $\bar{\sigma}$ denotes the maximal, while $\underline{\sigma}$ denotes the minimal singular value of the matrix. Because $\mathbf{T} + \mathbf{S} = \mathbf{I}$, it can be easily seen that (O1-O6) are opposing requirements. However, these objectives can be simultaneously achieved because the typical frequency ranges of the considered signals are different. For example, disturbance signals most commonly have low frequency components while the

noises are usually dominant in the upper frequency range, thus their attenuation on the specific frequency domains is not a contradiction.

The non-contradictionary arrangement of the frequency ranges for the (O1-O6) objectives by the specification of the **K** controller reshapes the closed loop transfer function. Thus, if the frequency domain of the signals presented in *Figure 3.3.* are defined by filters, then the H_∞ controller synthetised for this problem will accomplish the specified (O1-O6) objectives.

In the following parts, it is investigated that how the previously mentioned approach can be applied to the (3.5) system. Let it is assumed, that for the **P** system a design of a such a controller is desired that guarantees the **r** reference tracking of **P** with a γ norm described performance. The general mixed sensitivity structure representing this objective can be seen in *Figure 3.4*.

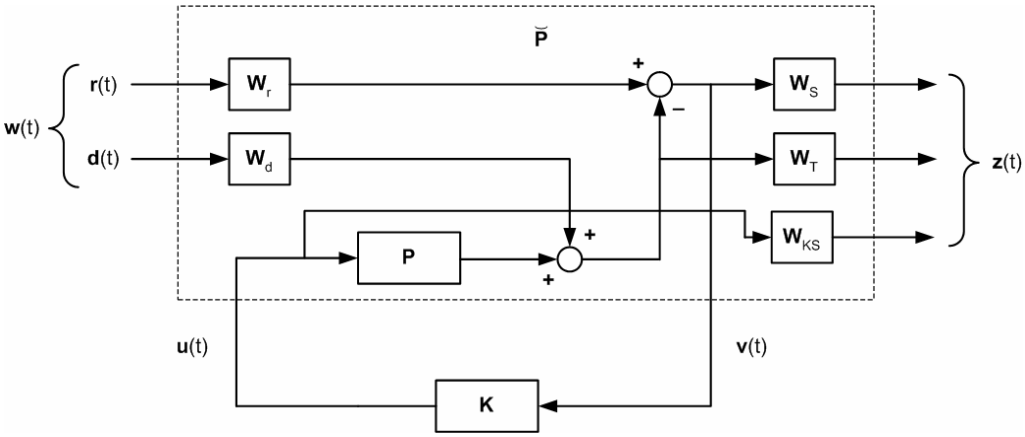


Figure 3.4. Mixed sensitivity structure for r(t) reference tracking

Here, with the help of the presented $\mathbf{W}_d$ (low pass) and $\mathbf{W}_r$ (high pass) filters, the frequency characteristics of **d** and **r** can be specified in the previously mentioned way. Additionally, with the help of the $\mathbf{W}_S$, $\mathbf{W}_T$, $\mathbf{W}_{KS}$ filters, the whole behavior of the tracking can be calibrated according to the (O1-O6) objectives. From the reformulation of the structure, a new $\tilde{\mathbf{P}}$ system is born. Based on the calibration, the designed controller to this new system will achieve the (O1-O6) objectives with a achievable γ, because in this case the normalization is done through the

$$\left\|\mathbf{G}(\mathbf{P},\mathbf{K})\right\|_\infty = \left\|\begin{bmatrix} \mathbf{W}_S\mathbf{S} \\ \mathbf{W}_T\mathbf{T} \\ \mathbf{W}_{KS}\mathbf{KS} \end{bmatrix}\right\|_\infty \tag{3.24}$$

norm. Thus, with the specification of the filters, the controller can be directly tuned to the problem. For good disturbance rejection and for the minimization of the tracking error, (O1) must be fulfilled, thus $\mathbf{W}_S$ should be a low pass filter, whose cutting frequency approximately equals to the expected maximal frequency of the noises. For the robust stability and for the reduction in control energy, (O4) and (O5) have to be satisfied, thus $\mathbf{W}_{KS}$ should be chosen to a high pass filter with a cutting frequency equal to the expected bandwidth of the closed loop. For good state tracking and for noise attenuation, (O3) must be fulfilled, thus it is worth choosing $\mathbf{W}_T$ as a high pass filter, whose cutting frequency is no more than the maximal frequency of the tolerated error [31, 38].

Let it is assumed, that such a **K** controller is needed to be designed for the **P** system which provides the estimation of the system states. For this case, the mixed sensitivity structure presented in *Figure 3.5* is applicable, where the **K** observer receives the inputs with some of the measured output signals of the system and the estimation error for reference outputs as well. Of course, for optimal tuning, filters which calibrate the designed observer according to the (O1-O6) objectives [38] are also introduced.

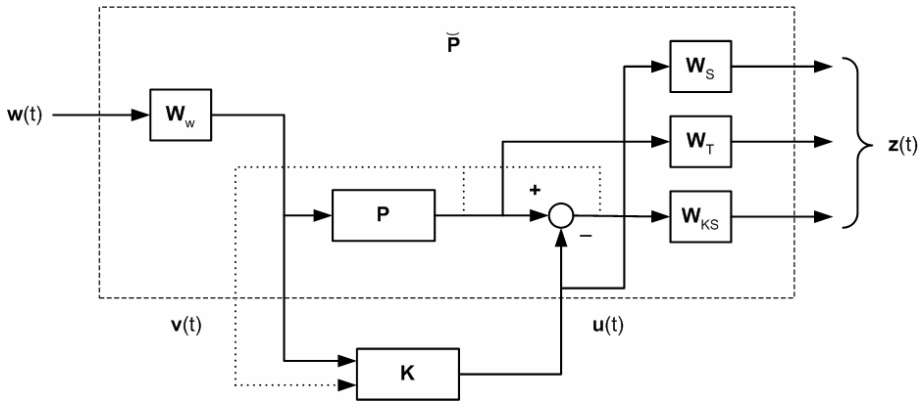


Figure 3.5. Mixed sensitivity structure for state estimation

It is also important to mention that with the introduced filters, which are responsible for the mixed sensitivity, the needed control aims can be achieved

extremely well, however, the filters also cause great increase in the number of the states of the designable H_∞ controller and, trough this, in the computational load of the whole control system [13].

3.1.6. Elimination of the offset error

Unfortunately, controllers with integral terms cannot be generated directly using H_∞ synthesis, however, they are required to remove steady state offset errors of the control. By introducing *ad-hoc* filters like (3.25), with a strong integrator behavior, between the **e** error signal and the inputs of **K**, the poles of the controller can be placed near the origin, which guarantees high DC loop gain. Altough there is no mathematical proof of the performance, this substitution has very little effect on the stability of the whole system The great disadvantage of this method is that the substitution increases the time constant of the control and makes the system liable to oscillations during the transients. Thus, in some cases, this method is neglected and the offset error is eliminated via pure proportional (P) terms, as it is going to be used in the design of the controller in this paper.

$$\mathbf{W}_I(s) = \frac{1}{s+\varepsilon}, \qquad |\varepsilon| \approx 0 \tag{3.25}$$

3.1.7. Improved controller sensitivity

It can happen that the reference tracking of the controller becomes sluggish because of the high gain or the aggressive integration of the error signal. This can be explained by the rapid change of the controller inner states caused by the high input gains. Moreover, if the sign of the error signal quickly changes, then this phenomena produces high frequency control signals, which can easily destabilize the whole system. To avoid these effects, the following solution was implemented: based on physical considerations, well-chosen bounds are introduced on the inner states of the controller to be able to achieve smooth control signals with improved sensitivity in the desired frequency range. The effect of this idea will be analyzed in the later sections.

3.1.8. Small gain theory

Let the system in *Figure 3.6* is considered. For this model, the following is true:

$$\mathbf{u}_1 = \mathbf{e}_1 - \mathbf{y}_2, \qquad \mathbf{u}_2 = \mathbf{e}_2 + \mathbf{y}_1 \tag{3.26}$$

$$\mathbf{y}_1 = \mathbf{G}_1(\mathbf{u}_1), \qquad \mathbf{y}_2 = \mathbf{G}_2(\mathbf{u}_2) \tag{3.27}$$

where $\dim(\mathbf{u}_1) = \mathrm{m}_1$ and $\dim(\mathbf{u}_2) = \mathrm{m}_2$. By combining (3.26) with (3.27) it can be shown [25] that

$$\begin{bmatrix}\mathbf{u}_1\\ \mathbf{u}_2\end{bmatrix} = \begin{bmatrix}\mathbf{e}_1\\ \mathbf{e}_2\end{bmatrix} - \begin{bmatrix}\mathbf{0} & \mathbf{I}_{\mathrm{m}_1}\\ -\mathbf{I}_{\mathrm{m}_2} & \mathbf{0}\end{bmatrix} \cdot \begin{bmatrix}\mathbf{G}_1(\mathbf{u}_1)\\ \mathbf{G}_2(\mathbf{u}_2)\end{bmatrix}, \tag{3.28}$$

$$\begin{bmatrix}\mathbf{y}_1\\ \mathbf{y}_2\end{bmatrix} = \begin{bmatrix}\mathbf{G}_1(\mathbf{u}_1) & \mathbf{0}\\ \mathbf{0} & \mathbf{G}_2(\mathbf{u}_2)\end{bmatrix} \cdot \left(\begin{bmatrix}\mathbf{e}_1\\ \mathbf{e}_2\end{bmatrix} - \begin{bmatrix}\mathbf{0} & \mathbf{I}_{\mathrm{m}_1}\\ -\mathbf{I}_{\mathrm{m}_2} & \mathbf{0}\end{bmatrix} \cdot \begin{bmatrix}\mathbf{y}_1\\ \mathbf{y}_2\end{bmatrix} \right). \tag{3.29}$$

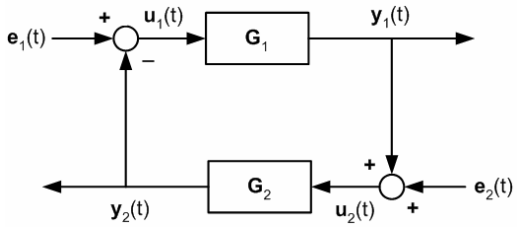


Figure 3.6. Closed loop system

If the following definitions are introduced [30]:

3.3 Definition: $\mathrm{L}_\infty(\wp) = \left\{ \mathrm{f} : \mathbb{R}_0^+ \to \wp \;\middle|\; \mathrm{f} \text{ is measureable and } \|\mathrm{f}\|_\infty < \infty \right\}$, *where* $\|\mathrm{f}\|_\infty = \sup_{t\in[0,\infty)} \left(\max_{i=1..\dim(\mathrm{f})} \mathrm{f}_i(t) \right)$.

3.4 Definition: *A* ***G*** *system is* L_∞ *stable if*

$$\forall \mathbf{u} \in \mathrm{L}_\infty(\mathrm{U}) \;\Rightarrow\; \forall \mathbf{y} \in \mathrm{L}_\infty(\mathrm{Y}), \quad \textit{where}\;\; \mathrm{U} = \mathrm{dom}(\mathbf{G}) \;\textit{and}\; \mathrm{Y} = \mathrm{img}(\mathbf{G}) \tag{3.30}$$

3.5 Definition: *A* ***G*** *system has finite* L_∞ *gain if and only if*

$$\left\| \mathbf{G}(\mathbf{u})_{[0,\mathrm{T}]} \right\|_\infty \le \gamma \left\| \mathbf{u}_{[0,\mathrm{T}]} \right\|_\infty + \mathrm{b}_\infty, \quad \forall \mathrm{T} \in \mathbb{R},\, \mathrm{b}_\infty \in \mathbb{R} \tag{3.31}$$

then the next theorem is true:

3.2 Theorem: (3.28) *is* L_∞ *stable if and only if (3.29) is also* L_∞ *stable, and (3.28) has finite* L_∞ *gain if and only if (3.29) has finite* L_∞ *gain as well. Moreover, the overall* $\mathbf{G}_1$, $\mathbf{G}_2$ *closed loop system has finite* L_∞ *gain only in the case of* $(\gamma_1 \cdot \gamma_2) < 1$.

This theory is important if the control system is built up from the serial connection of more than one H_∞ controller, because the closed loop system will be only stable if the *3.2 theory* is true. So, if a H_∞ observer is implemented to estimate the non measurable parameters of the system, then for any serially connected H_∞ controller to this observer, the overall L_∞ gain of the system must be less then 1 by the *3.2 theorem*, or the maximal overall γ gain received during the design and the stability of the control, which is provided by this gain, is not guaranteed. This holds for LPV systems as well [30].

3.1.9. Robustness and stability

In case of MIMO systems, the presence of some uncertainty in the parameters and in the state equations of the system is a common problem, because usually the physical system is not perfectly described and known. Thus, the output signals of the system are affected by fluctuations or perturbations. Such a system is presented in *Figure 3.7.*

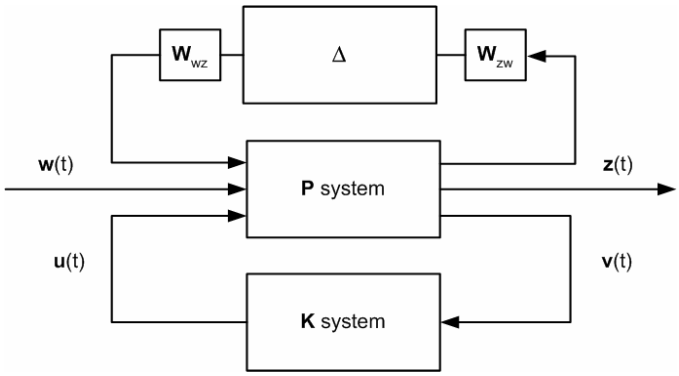


Figure 3.7. Robust control of uncertain systems

The above mentioned uncertainty of the system can be modeled in the following way: the effect of uncertainty is fed back as a reference signal into the plant, while the input uncertainty is included as a system noise in the model. Based on the work of Glover and McFarlane [9], it is well-known today that for

even such systems it is possible to apply the H_∞ theory, but the overall robust stability is guaranteed if and only if the transfer function between the input and output uncertainty has a finite L_∞ gain of no more than 1. Because this criteria is given for the full frequency spectrum, by the frequency shaping of the uncertainties with the $\mathbf{W}_{zw}$ and $\mathbf{W}_{wz}$ filters presented in *Figure 3.7*, the robust operation interval of the controller can be tuned to the dominant frequency ranges of the uncertainties of the particular problem.

3.2. The extended Kalman filters

The previously mentioned H_∞ theory makes possible the design of controllers only in the cases of LTI and LPV systems. But if the (2.23) and (2.36) equations are considered then there is no possible way to transform the state variables presented there to an LTI or LPV form, because these relationships consist of such nonlinearities that cannot be given in a useful parameter-affine form. Thus, another theory is needed, which holds the possibility to estimate these variables. Because of this, in the following section the theory of the extended Kalman filters [19], which is also applied in the designed control structure, is introduced briefly.

3.2.1. Theory of the Kalman filters

Let it is supposed that the following discrete time system is given:

$$\mathbf{x}(k+1) = \Phi \cdot \mathbf{x}(k) + \Gamma \cdot \mathbf{u}(k) + \varepsilon_{sys}(k) \tag{3.32}$$

$$\mathbf{y}(k) = \mathbf{C} \cdot \mathbf{x}(k) + \varepsilon_{mes}(k) \tag{3.33}$$

where

- $\varepsilon_{sys}(k)$ is the system noise, which is a white noise, with a **0** mean value, and with a covariance equals to $E\left\{\varepsilon_{sys}(k)\varepsilon_{sys}^T(k)\right\} = \mathbf{Q}$.
- $\varepsilon_{mes}(k)$ is the measurement noise, which is also a white noise, with a **0** mean value, and with a covariance equals to $E\left\{\varepsilon_{mes}(k)\varepsilon_{mes}^T(k)\right\} = \mathbf{R}$.

- Φ is the discrete counterpart of the **A** state matrix of the originally continuous state equation system, and $\Phi = e^{\mathbf{A}h}$ where $h \in \mathbb{R}_0^+$ is the discrete time step.
- Γ is the discrete counterpart of the **B** matrix of the originally continuous state equation system, and $\Gamma = \mathbf{A}^{-1}\left(e^{\mathbf{A}h} - \mathbf{I}\right)\mathbf{B}$.

For this system such a method is searched for which is capable to estimate the value of the non measurable **x**(k) states with the help of the **y**(k) outputs. Let the estimation process be chosen in the following form:

$$\mathbf{x}(k+1|k) = \Phi \cdot \mathbf{x}(k|k-1) + \Gamma \cdot \mathbf{u}(k) + \mathbf{K}(k) \cdot \underbrace{\left(\mathbf{y}(k) - \mathbf{C} \cdot \mathbf{x}(k|k-1)\right)}_{\text{estimation error}}. \tag{3.34}$$

In this case, the estimation error: $\tilde{\mathbf{x}}(k) = \mathbf{x}(k) - \mathbf{x}(k|k-1)$ is also equal to:

$$\tilde{\mathbf{x}}(k+1) = \Phi \cdot \tilde{\mathbf{x}}(k) + \varepsilon_{sys}(k) + \mathbf{K}(k)\left(\mathbf{y}(k) - \mathbf{C} \cdot \mathbf{x}(k|k-1)\right), \tag{3.35}$$

which is the discrete error process with a covariance defined by (3.36).

$$\mathbf{P}(k) = E\left\{\left[\tilde{\mathbf{x}}(k) - E\left\{\tilde{\mathbf{x}}(k)\right\}\right] \cdot \left[\tilde{\mathbf{x}}(k) - E\left\{\tilde{\mathbf{x}}(k)\right\}\right]^T\right\}. \tag{3.36}$$

To achieve our goal, the following task must be solved: Find a **K**(k) discrete process that minimizes the $\tilde{\mathbf{x}}(k)$ error function for any $k \in \mathbb{N}$, so that it ensures the most optimal estimation of the states. The solution of this problem can be given with the help of direct minimization, to which the $\mathbf{P}(k)$ covariance function must be given in a quadratic form. The following recursive method presents the solution for obtaining the correct **K**(k) and for accomplishing the estimation efficiently:

The recursive algorithm of the Kalman filter [2]:

1. *Prediction phase*

$$\mathbf{x}(k+1|k) = \Phi \cdot \mathbf{x}(k|k) + \Gamma \cdot \mathbf{u}(k) \tag{3.37}$$

$$\mathbf{P}(k+1|k) = \Phi \cdot \mathbf{P}(k|k) \cdot \Phi^T + \mathbf{Q} \tag{3.38}$$

2. *Correction phase*

$$\mathbf{K}(k+1) = \mathbf{P}(k+1|k) \cdot \mathbf{C}^T \cdot \left(\mathbf{C} \cdot \mathbf{P}(k+1|k) \cdot \mathbf{C}^T + \mathbf{R} \right)^{-1} \tag{3.39}$$

$$\mathbf{x}(k+1|k+1) = \mathbf{x}(k+1|k) + \mathbf{K}(k+1) \cdot \left(\mathbf{y}(k+1) - \mathbf{C} \cdot \mathbf{x}(k+1|k) \right) \tag{3.40}$$

$$\mathbf{P}(k+1|k+1) = \mathbf{P}(k+1|k) - \mathbf{K}(k+1) \cdot \mathbf{C} \cdot \mathbf{P}(k+1|k) \tag{3.41}$$

where $\hat{\mathbf{x}}(0) = \mathbf{0}$ and $\mathbf{P}(0) = \mathbf{0}$, if the considered system is started from a zero initial state. So the above mentioned algorithm is capable of the estimation of states in two steps, where it not only uses the mathematical equations of the system model for numerical approximation (*step 1*), but it also tries to attenuate the effect of noises, disturbances, and uncertainties (*step 2*). In this way the given method functions well in a noisy environment, and it is also robust, therefore it provides an excellent tool for the state estimation of real time systems. The definition of the **R** and **Q** covariance matrices is a critical part of the design, to which the stochastic properties of the existing noises have to be investigated. Because this information is usually not available, in most of the cases the optimal weights are obtained by *try and error*.

3.2.2. Extension to nonlinear systems

The previously introduced recursive algorithm of the Kalman filter provides a solution for the state estimation of only LTI systems. However, in most of the real life applications, the role of nonlinearities is significant. If the state equations of a continuous system can be discretised, and in discrete time, it can be given in the form of (3.42) and (3.43), where $\mathrm{f_d}(\mathbf{x}(k), \mathbf{u}(k))$ is partially differentiateable by $\mathbf{x}(k)$ and $\mathbf{u}(k)$, moreover, $\mathrm{h_d}(\mathbf{x}(k))$ is partially differentiateable by $\mathbf{x}(k)$, then the recursive algorithm given in the LTI case can be extended for the NL case as well. This extended algorithm is called *extended Kalman filter* (EKF) [35].

$$\mathbf{x}(k) = \mathrm{f_d}(\mathbf{x}(k), \mathbf{u}(k)) \tag{3.42}$$

$$\mathbf{y}(k) = \mathrm{h_d}(\mathbf{x}(k)) \tag{3.43}$$

Here, the method of the EKF is shown only for the basic nonlinear case like the model of the IM, thus the investigated system is the following:

$$\mathbf{x}(k) = f_d(\mathbf{x}(k)) + g_d(\mathbf{u}(k)) + \varepsilon_{sys}(k) \tag{3.44}$$

$$\mathbf{y}(k) = \mathbf{C} \cdot \mathbf{x}(k) + \varepsilon_{mes}(k) \tag{3.45}$$

Then for the problem defined by (3.44) and (3.45) the EKF is the following recursive algorithm:

1. *Prediction phase*

$$\mathbf{x}(k+1|k) = f_d(\mathbf{x}(k|k)) + g_d(\mathbf{u}(k)) \tag{3.46}$$

$$\mathbf{P}(k+1|k) = \mathbf{F}(k) \cdot \mathbf{P}(k|k) \cdot \mathbf{F}(k)^T + \mathbf{Q}, \quad \text{where} \quad \mathbf{F}(k) = \left.\frac{\partial f_d(\mathbf{x})}{\partial \mathbf{x}}\right|_{\mathbf{x}=\hat{\mathbf{x}}(k)} \tag{3.47}$$

2. *Correction phase* consists of the same (3.39), (3.40), and (3.41) defined steps.

Moreover, the EKF produced estimation fulfills the same optimality provided in the LTI case.

3.2.3. Modification of the EKF with the Adams-Basforth method

In the application of the EKF, most commonly, the previously continuous NL system is discretised by the first order explicit Euler method, which provides the following approximation:

$$\mathbf{x}(h \cdot (k+1)) = \mathbf{x}(h \cdot k) + h \cdot f(\mathbf{x}(h \cdot k)) = f_d(\mathbf{x}(k)). \tag{3.48}$$

However, the (3.48) method introduces a discretisation error into the system, whose absolute value is μh, if the Lipsich property is true for $f(\mathbf{x}(t))$, and it can be continuously partially differentiated, and μ is a given constant. But for such nonlinear systems, where the applicable step size can be compared to the time constants of the system, like the time constant of the change of the torque in the case of the IM, the value of μh will be significant. Because of this, in the prediction phase, the discretisation of $f(\mathbf{x}(t))$ through the Euler method will generate such a system level accumulating error which can be compensated with very high weight of **Q**, but this overweighting extremely increases the length of the transients of the EKF, thus it will be able to track the system states with only large time lags. In case of the sinusoidal state variables of the IM, this approach provides catastrophic estimation error. Thus, instead of the Euler method, such an

algorithm was used that has much less discretisation error, and because its recursive nature, the increase in the computational load can be neglected. This method is the Adams-Basforth, third order numerical recursive explicit method, which for our problem can be given in the following form:

$$\begin{aligned}\mathbf{x}\left(\mathrm{h}\cdot(\mathrm{k}+1)\right) = \mathbf{x}(\mathrm{h}\cdot\mathrm{k}) + \frac{\mathrm{h}}{12}\Big[5\cdot\mathrm{f}\left(\mathbf{x}\left(\mathrm{h}\cdot(\mathrm{k}-2)\right)\right) \\ -16\cdot\mathrm{f}\left(\mathbf{x}\left(\mathrm{h}\cdot(\mathrm{k}-1)\right)\right) + 23\cdot\mathrm{f}\left(\mathbf{x}\left(\mathrm{h}\cdot\mathrm{k}\right)\right)\Big]\end{aligned} \tag{3.49}$$

where $\mathbf{x}(0) = \mathbf{0}$, $\mathbf{x}(-1\cdot\mathrm{h}) = \mathbf{0}$, and $\mathbf{x}(-2\cdot\mathrm{h}) = \mathbf{0}$, if the considered system started from a zero initial state. The local discretisation error of this method is $3/8\cdot\mathrm{x}^4(\mathrm{t})\cdot\mathrm{h}^3$ which is much less than for the Euler method [12]. In this way, the estimation accuracy of the EKF is greatly sharpened with the use of two extra multiplication and addition.

4. The design of the speed sensorless LPV control structure

4.1. Formulation of the control problem

To be able to build a drive which fulfils the recent industrial requirements that were mentioned in the first section and based on the previously presented theories, such a unique speed sensorless controller structure is aimed to be designed that makes possible the independent control of the flux and speed on the full 4 quadrant operation range of the motor. This functionality is needed to be provided through the use of the three-phase stator voltages as control inputs and by the measurements of the stator currents only. Moreover, the controller must consider the variation of R_r to heat, in order to decrease the effect of parameter uncertainties. Because in practice, the three-phase voltage feed of the motor is most commonly provided by an inverter through *pulse width modulation* (PWM) of a DC bus voltage, thus it is 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 in 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 a most commonly 0.5% relative amplitude in comparison to the measured quantities. Moreover, in order to prevent any reference tracking error in speed and in flux, the controller also has to be 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, and it 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 opposition to this, the reference signal for speed can be as it is pleased, with the occurrences of even uncontinuous step changes. Furthermore, the control has to be provided in that way which makes possible the most accurate reference tracking with the least possible time constant, but with the awareness on 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 has to consider the specific behaviors and properties of the available digital and analog hardware that can be used for the implementation.

4.2. The structure of control

After careful theoretical and practical design, for the previously defined problem, the controller has been synthetised by the structure presented in *Figure 4.1*. For the functioning of this system, which fulfils the given strict requirements, only the measurements of two stator phase currents are needed. Because of the symmetry of the phases, $i_{sc} = -(i_{sa} + i_{sb})$ yields and this is approximately true even in real life applications. Thus, the overall $\mathbf{i}_s^s$ vector can be accurately obtained only from two measurements. Furthermore, because the input voltage feed is generated directly by the system, therefore the accurate $\mathbf{u}_s^s$ is known without measurements.

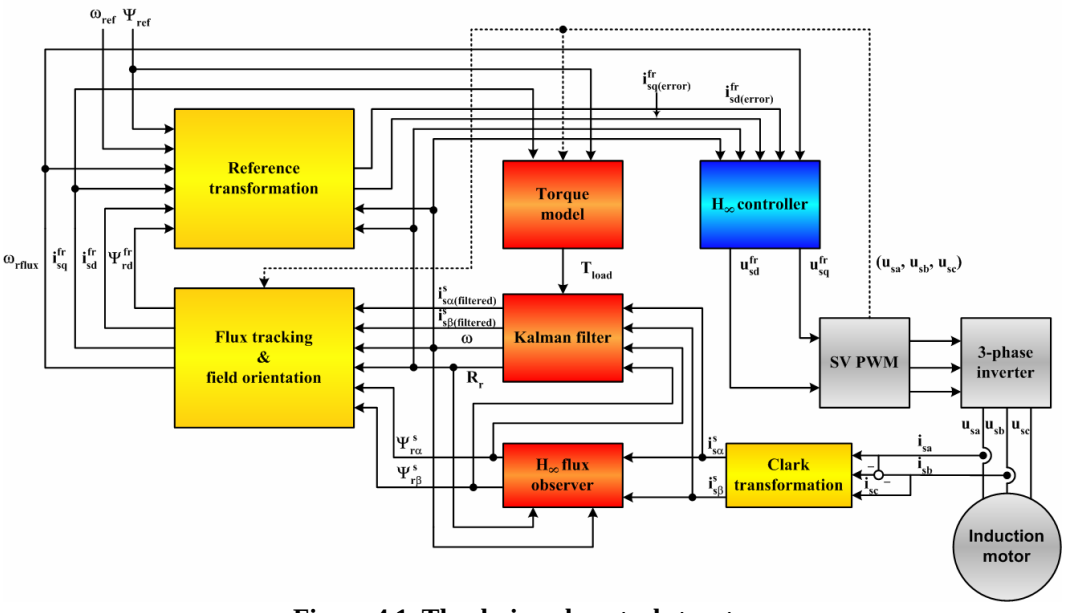


Figure 4.1. The designed control structure

By the examination of the relative degrees of the motor model, it can be concluded, 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. The mechanism of the control system is briefly the following:

With the help of the (4.1) *Clark transformation*, from the two measured stator phase currents the two phase representation based $i^s_{s\alpha}$ and $i^s_{s\beta}$ components are calculated, which are equal to the *d* and *q* component of the stator current space vector represented in a stator fixed, stationary $(\omega_k = 0)$ reference frame.

$$\begin{bmatrix} i^s_{s\alpha}(t) \\ i^s_{s\beta}(t) \end{bmatrix} = \begin{bmatrix} \frac{2}{3} & -\frac{1}{3} & -\frac{1}{3} \\ 0 & \frac{1}{\sqrt{3}} & -\frac{1}{\sqrt{3}} \end{bmatrix} \cdot \begin{bmatrix} i_a(t) \\ i_b(t) \\ i_c(t) \end{bmatrix}, \tag{4.1}$$

Then, by a complicated estimation method, which is the connection of an EKF and an H_∞ observer, the unknown $\Psi^s_{r\alpha}(t)$, $\Psi^s_{r\beta}(t)$, 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 measured current. The application of the EKF is strongly supported by the fact, that its computational load is less than for an input/output linearized model based full state H_∞ observer [13].

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 of the H_∞ observer. This estimation system is supplemented by a trivial calculation based, however, as it is going to be showed, 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 $\Psi^s_{r\alpha}$ and $\Psi^s_{r\beta}$ 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 with 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 their tracking error is 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 [35] as well. The controller given u_{sd}^{fr} and u_{sq}^{fr} signals, are realized through the SV PWM (Space Vector PWM) hardware module, which transforms these signals into a 3 bit hexagonal ignition code stream, which code corresponds to the triggering times of the IGBTs of the inverter. The correct switching sequence of these IGBTs provides the chopping of the bus voltage which realizes the 3 phase PWM stator voltage feed of the motor.

It is important to note, that for the correct computation of the ignition codes the angle of the rotor flux must be known, which is either tracked by the inverter itself, or it is provided by the previously mentioned field orientation module. In conclusion, the above described system theoretically solves the given problem, but before its direct implementation, each of its submodules has to be designed and then tuned to each other. This is shown step by step in the following sections.

4.3. The design of the H_∞ controller module

The controller module was realized through the (2.37) and (2.38) LPV model defined field oriented subsystem. During the design the following requirements were taken into consideration:

- The module has to provide as accurate as possible reference tracking of i_{sd}^{fr} and i_{sq}^{fr}, thus the optimization problem must include the minimization of the **e**(t) error signal describing the deviation from the references of these values.
- Furthermore, it has to guarantee the optimal state tracking of the system, for which the frequency definition of the outputs of the system is unavoidable.

- It is assumed that no noise or disturbance is presented in the system, because the elimination of these effects is the task of the estimation module. In this way, the value of the three *scheduling parameter*: ω, R_r, and ω_{rflux} are assumed to be fully known.
- There is no need for the minimization of the control energy, because the greatness of the voltage feed is limited by the built in protection of the control logic of modern inverters or this task can also be solved externally.

Based on these criteria, the optimization problem can be given in the form presented in *Figure 4.2*.

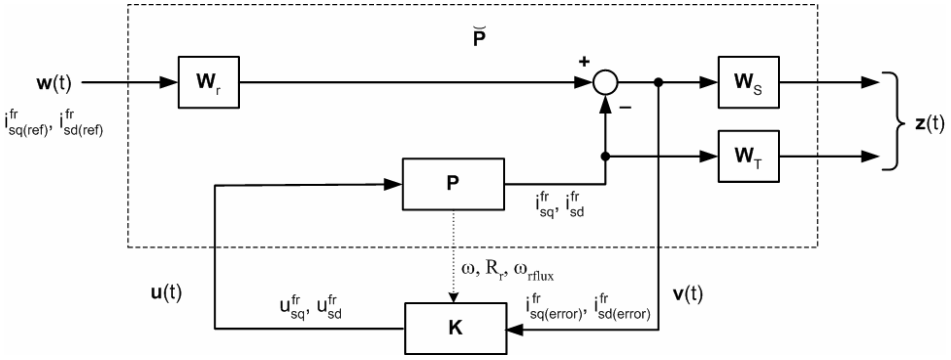


Figure 4.2. Mixed sensitivity structure for the H_∞ controller design

In the one degree of freedom structure of *Figure 4.2*, the presented $\mathbf{W}_S$ filter was chosen to be a low pass filter, based on the concepts of mixed sensitivity structures (see *Section 3.1.4*). The cutting frequency of this filter gives the minimal frequency of the acceptable error, which has to be a high value to avoid unwanted deviations. Based on this, the filter was chosen as it follows:

$$\mathbf{W}_S(s) = \mathrm{diag}\left(\frac{10}{s+10}, \frac{10}{s+10}\right) \tag{4.2}$$

The cutting frequency of filter (4.2) was calibrated to 10Hz because mainly the DC error and the small frequency oscillations were aimed to be eliminated during the control. It is important to mention that it is avoidable to choose the cutting frequency to a larger value, because it significantly raises the reachable value of γ, which is needed to be lower than one, based on the small gain theorem with respect to the connected estimation method. On the other hand, the too large

cutting frequency also makes the controller needlessly aggressive to high frequency deviations, which causes great oscillations in the control signal that cannot be tracked by the inverter. During the design, the amplification of the filter in the passing region was chosen to be 0dB, in order to provide external tuning of sensitivity independently from the controller, which even gives the possibility of an online tuning during operation. This could be an important issue for a real time application.

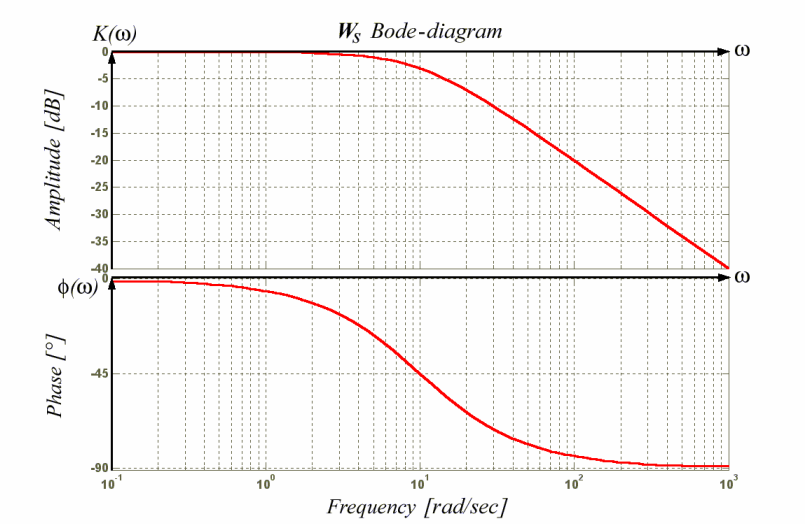


Figure 4.3. Bode-diagram of the $W_s(s)$ dynamical filter in the optimality problem of the controller

Because in the (2.37) and (2.38) defined LPV submodel, no system level noises are presented, thus if the estimations are correct and dependable, and it is assumed based on the same reasons, that i_{sd}^{fr}, i_{sq}^{fr} are also noiseless, with that the parameter uncertainties were greatly reduced by the mathematical description of the variation of R_r, then there is no need to use dynamic weighting for $\mathbf{W}_T$. Based on these concepts and the results of *try and error*, $\mathbf{W}_T$ was chosen in the form of (4.3).

$$\mathbf{W}_T(s) = \mathrm{diag}\left(0.8, 0.8\right) \tag{4.3}$$

It is also true that, because the reference signals can be arbitrary, $\mathbf{W}_r$ can be left out from the structure. The H_∞ controller defined by the previous considerations was designed with respect to the motor parameters given in *Section 5.1* and with the help of the *hinfgs* function of the LMI toolbox of the Matlab,

which transforms the polytypical H_∞ optimization problem into a Ljapunov-function synthetisation. This LMI optimization method obtained controller is such a $2^3 = 8$ corner system described polytopical current regulator, which in coincidence with *Figure 4.2* has 5 states, 2 inputs, and 2 outputs with a $\gamma = 0.6247$. Three of the controller states are directly produced by the LPV model, while the other two are gained because of the existence of the introduced filters. During the process of optimization, the bounds of the parameters are defined by the following intervals: $\omega_{max} = 100\text{Hz}$, $\omega_{min} = -85\text{Hz}$, $R_{rmax} = 6\Omega$, $R_{rmin} = 3\Omega$, $\omega_{rflux\,max} = 50\text{Hz}$, $\omega_{rflux\,min} = -50\text{Hz}$, to be able to provide the control on the full 4/4 quadrant operation range of the motor. By the testing of the resulted controller, it was experienced the method lowers the reference tracking error of $i_{sd_{ref}}^{fr}$, $i_{sq_{ref}}^{fr}$ to 62% as it was expected from γ. During the design of the $\mathbf{W}_S$, the gain of tracking error attenuation, as a tuning parameter, was lifted out from the controller, thus by the external amplification of this error the increase in the performance can be achieved.

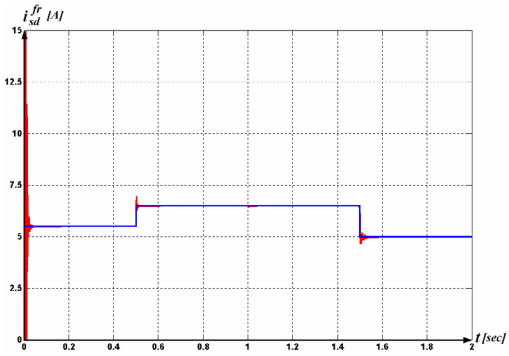


Figure 4.4(a). Reference (*red*) tracking (*blue*) for i_{sd}^{fr}

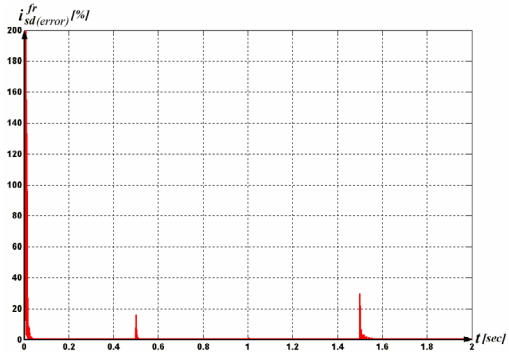


Figure 4.4(b). Relative tracking error of i_{sd}^{fr}

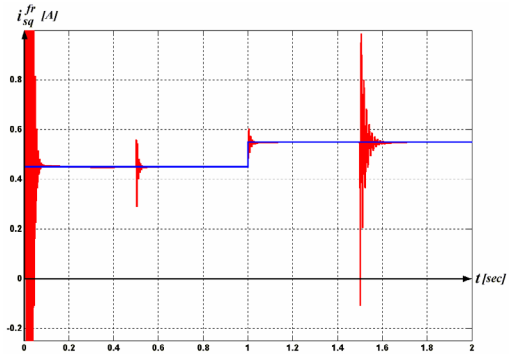


Figure 4.4(c). Reference (*red*) tracking (*blue*) for i_{sq}^{fr}

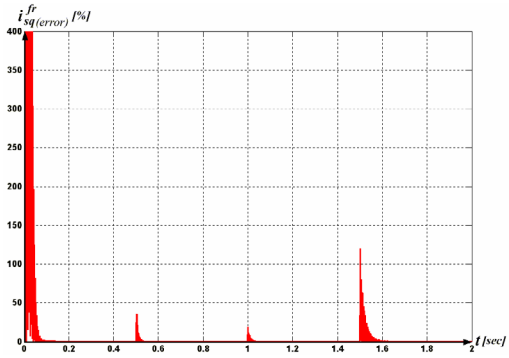


Figure 4.4(d). Relative tracking error of i_{sq}^{fr}

Because of this reason, the amplification of the error signal was calibrated to 175, which is nearly 45dB. By this approach, the reference tracking presented in *Figures 4.4(a-d)* could be achieved, which had a 0.35% tracking error in steady state for either cases when $T_{load} = 0.5\text{Nm}$ was applied to the system.

With the *hinfnorm* function of the Matlab program package, the H_∞ norm of the closed loop control was checked on the whole parameter range, and the results showed that the maximum of the deviation was 10^{-11}, so the optimization was successful on all of the possible parameter trajectories. Furthermore, the unit step answer presented on *Figure 4.5(a) and 4.5(b)* of the controller was also checked, which was generated by the *pdsimul* function of Matlab for the real life modeling parameter trajectories given in *Figure 4.5(c) and 4.5(d)* with constant $R_r = 3\Omega$. Although the overshoots were significant in each cases because of the high sensitivity of the controller, the steady state was reached in no more than 45ms.

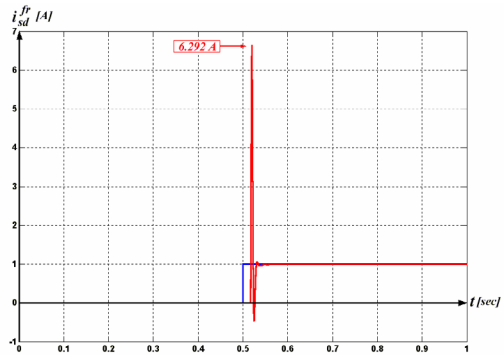


Figure 4.5(a). Closed loop unit step answer for i_{sd}^{fr}

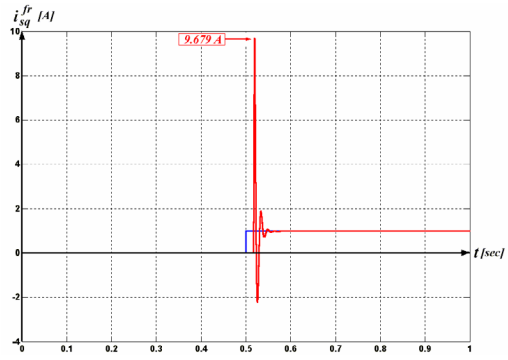


Figure 4.5(b). Closed loop unit step answer for i_{sq}^{fr}

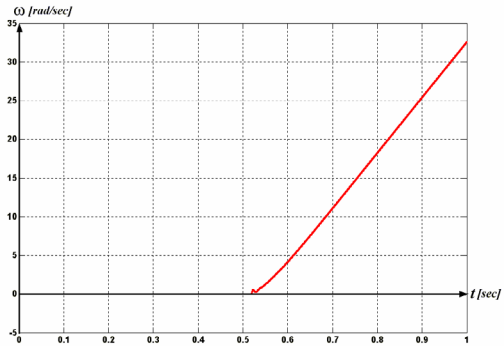


Figure 4.5(c). Trajectory of parameter ω

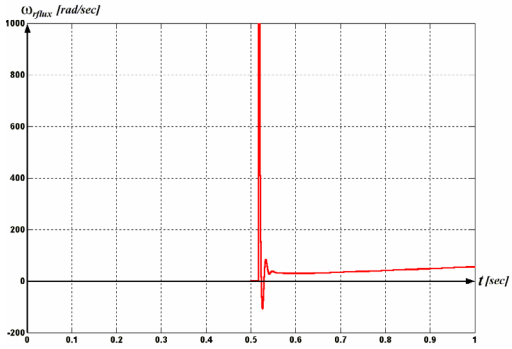


Figure 4.5(d). Trajectory of parameter ω_{rflux}

Unfortunately, the absence of the integrator term, which was left out to provide fast reference tracking, showed up in the occurrences of the previously mentioned

great overshoots during the transients. Therefore, the limiting of the states of the controller is needed because of this overshooting behavior. But the exact calibration of these bounds are rewarding only if the whole closed loop controller structure is considered to avoid making the system too sluggish.

If the parameter values of the concrete modeled motor was randomly changed in a 5% range then the reference tracking presented in *Figure 4.6(a) and 4.6(b)* has been received, which can be evaluated to be fine, because the resulted 0.4% tracking error, which is very close to excellent performance of the pervious case, guarantees the robustness of the controller.

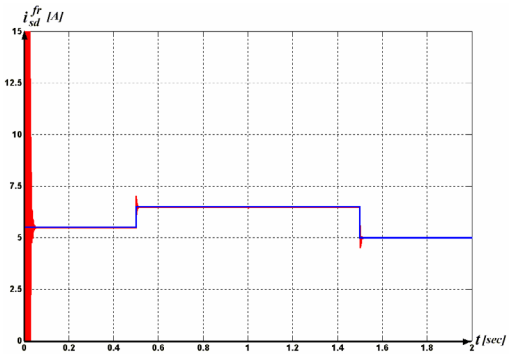


Figure 4.6(a). Reference (*red*) tracking (*blue*) for i_{sd}^{fr}

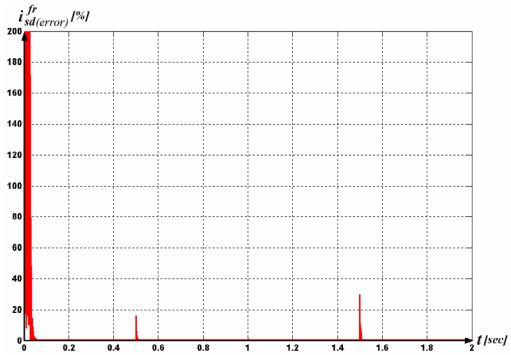


Figure 4.6(b). Relative tracking error of i_{sd}^{fr}

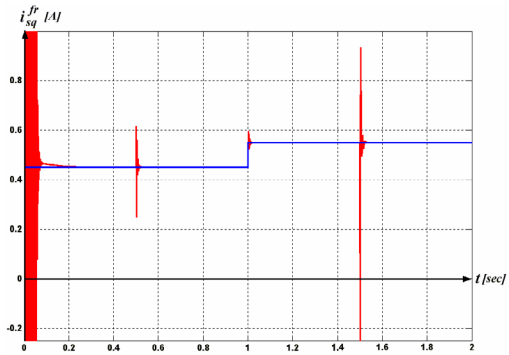


Figure 4.6(c). Reference (*red*) tracking (*blue*) for i_{sq}^{fr}

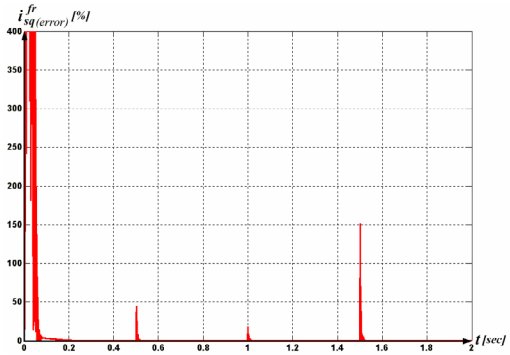


Figure 4.6(d). Relative tracking error of i_{sq}^{fr}

Moreover, the controller does not exceed the 1% tracking error for both reference signals till 15% parameter uncertainty, but after this point the control signals began to oscillate because of the 45dB gain of the error amplification. These oscillations can be detected during the real life operation and by decreasing the error gains the interval of robust stability can be greatly increased. Thus, it can be stated that the designed controller fulfills the given expectations that were

aimed. (The specific motor parameters used for these simulations are going to be given in *Section 5.1*)

4.4. I/O linearization based reference transformation

To be able to apply the designed controller, described in *Section 4.3*, for the speed and flux control of the IM, the reference tracking errors of ω and Ψ must be transformed into reference signals related to i_{sd}^{fr} and i_{sq}^{fr}. The reason of this approach is that only equation (2.34) can be given in a LPV form. If ω and Ψ would be directly used for control, then the resulted structure would not be parameter-affine, thus it cannot be given as a H_∞ optimization problem in just only that case, if the whole model is linearized somehow, or the nonlinearities are let out or algebraically eliminated. But this kind of abandonment or linearization produces great loss in the described system dynamics, which could cause the instability of the system for rapid control. Furthermore, the algebraic elimination of the nonlinearities is a challenging mathematical task, which overcomplicates the controller. Thus, if such a controller is needed which can handle the whole system dynamics, than the most clear path to achieve this leads head through the transformation of the reference signals. Let the following model equations be considered:

$$\frac{d\Psi_{rd}^{fr}(t)}{dt} = -\frac{R_r(t)}{L_r}\Psi_{rd}^{fr}(t) + \frac{L_m R_r(t)}{L_r} i_{sd}^{fr}(t) \tag{4.4}$$

$$\frac{d\omega(t)}{dt} = \frac{3p^2 L_m}{2JL_r}\cdot\left(\Psi_{rd}^{fr}(t)\cdot i_{sq}^{fr}(t)\right) - \frac{p}{J}\cdot\left(T_{load}(t) + F\omega(t)\right) \tag{4.5}$$

If we assumes that the following virtual inputs are chosen to the I/O linearization of (4.4) and (4.5):

$$v_1(t) = \frac{d\Psi_{rd}^{fr}(t)}{dt} = \Psi_{rd_{ref}}^{fr}(t) - \Psi_{rd}^{fr}(t), \tag{4.6}$$

$$v_2(t) = \frac{d\omega(t)}{dt} = \omega(t)_{ref} - \omega(t), \tag{4.7}$$

where, as it can be seen, the values of the virtual inputs are equal to the deviation from the reference at every time instant, then the I/O linearized from of this

submodel between the currents and the new virtual inputs can be given by simply inverting the equations:

$$i_{sd_{ref}}^{fr}(t) = \frac{L_r}{L_m R_r(t)} \cdot v_1(t) + \frac{1}{L_m} \Psi_{rd}^{fr}(t), \tag{4.8}$$

$$i_{sq_{ref}}^{fr}(t) = \frac{2JL_r}{3p^2 L_m} \cdot \frac{v_2(t)}{\Psi_{rd}^{fr}(t)} + \frac{2L_r}{3pL_m} \cdot \left(T_{load}(t) + F\omega(t)\right). \tag{4.9}$$

In this way, the input of the controller arises from the result of $i_{sd_{error}}^{fr}(t) = i_{sd_{ref}}^{fr}(t) - i_{sd}^{fr}(t)$ and $i_{sq_{error}}^{fr}(t) = i_{sq_{ref}}^{fr}(t) - i_{sq}^{fr}(t)$. It is important to mention that because $\Psi_{rd}^{fr}(t)$ is always equal to the value of $\left|\Psi_r^x(t)\right|$ in any *x* reference frame, the division by this quantity can be always accomplished except 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 (4.9) would force the controller for smooth exponential rise till it reaches the border of its unstable dynamical interval, and the whole error signal can be received by the controller.

4.5. Rotor flux orientation and flux angle tracking

For the systems mentioned in *Section 4.3 and 4.4*, it is crucial that the stator currents measured on the motor and the later shown estimation method-provided rotor flux is transformed to the $\omega_k = \omega_{rflux}$ 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:

$$\cos\rho(t) = \frac{\Psi_{r\alpha}^{s}(t)}{\sqrt{\left(\Psi_{r\alpha}^{s}(t)\right)^2 + \left(\Psi_{r\beta}^{s}(t)\right)^2}}, \quad \sin\rho(t) = \frac{\Psi_{r\beta}^{s}(t)}{\sqrt{\left(\Psi_{r\alpha}^{s}(t)\right)^2 + \left(\Psi_{r\beta}^{s}(t)\right)^2}}. \tag{4.10}$$

In this way, by using the calculated trigonometrical values of the flux angle, the following is true for $i^s_{r\alpha}(t)$ and $i^s_{r\beta}(t)$.

$$\begin{bmatrix} i^{fr}_{sd}(t) \\ i^{fr}_{sq}(t) \end{bmatrix} = \begin{bmatrix} \cos\rho(t) & \sin\rho(t) \\ -\sin\rho(t) & \cos\rho(t) \end{bmatrix} \cdot \begin{bmatrix} i^s_{r\alpha}(t) \\ i^s_{r\beta}(t) \end{bmatrix}. \tag{4.11}$$

The (4.11) defined method is called the *Park-transformation* in the literature. Furthermore,

$$\Psi^{fr}_{rd}(t) = \sqrt{\left(\Psi^s_{r\alpha}(t)\right)^2 + \left(\Psi^s_{r\alpha}(t)\right)^2}, \tag{4.12}$$

and the value of $\omega_{rflux}(t)$ can be obtained from (2.35). Based on this, the field orientation can be completed with relative few operations. It is needless to calculate the exact value of $\rho(t)$, because for the back transformation of the $u^{fr}_{sd}(t)$ and $u^{fr}_{sq}(t)$ control inputs only the knowledge of $\cos\rho(t)$ and $\sin\rho(t)$ is enough. In case of such a problem, where the position control of the rotor is needed (*precision drives*), the above mentioned structure is still applicable for the tracking of the magnetic field orientation because

$$\rho(t) = \operatorname{atan}\frac{\Psi^s_{r\beta}(t)}{\Psi^s_{r\alpha}(t)} + \begin{cases} 0, & \text{if } \Psi^s_{r\alpha}(t) \geq 0 \\ \pi, & \text{if } \Psi^s_{r\alpha}(t) < 0 \text{ and } \Psi^s_{r\beta}(t) > 0 \\ -\pi, & \text{if } \Psi^s_{r\alpha}(t) < 0 \text{ and } \Psi^s_{r\beta}(t) < 0 \end{cases}. \tag{4.13}$$

Unfortunately, from practical point of view, in the first time moment when still $\left|\Psi^s_r(t)\right| = 0$, the (4.10) calculation method is not useable, thus the assumption of $\rho(t) = 0$ is made for this case. The orientation error due to this substitution rapidly vanishes when the magnetic field is built up inside the machine.

4.6. The structure of estimation

The main task of the designed estimation method is to make possible the speed sensorless control of the IM. Because the previously introduced methods are founded on the accuracy of the system information, the estimation method has to be able to satisfy these needs, and even be able to attenuate of the measurement

and the inverter caused system noises. For these expectations, such a method is perfectly suitable, which consist of three parts.

During the design of the estimators, it has quickly turned out that the equations which contain the ω, R_r, $\Psi^s_{r\alpha}(t)$ and $\Psi^s_{r\beta}(t)$ 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 $\Psi^s_{r\alpha}(t)$ and $\Psi^s_{r\beta}(t)$. 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. In this structure, the $\omega(t)$ and $R_r(t)$ scheduled H_∞ observer provides the stator oriented estimation of $\Psi^s_{r\alpha}(t)$ and $\Psi^s_{r\beta}(t)$ with the help of the measured $i^s_{r\alpha}(t)$ and $i^s_{r\beta}(t)$, while the EKF calculates $\omega(t)$ and $R_r(t)$ with the filtering of $i^s_{r\alpha}(t)$, $i^s_{r\beta}(t)$ based on the estimated values of $\Psi^s_{r\alpha}(t)$ and $\Psi^s_{r\beta}(t)$ and the measured stator currents. This construction 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.

4.6.1. The design of the H_∞ observer

The design of the observer method was based on the (4.14) **P** stator oriented subsystem model.

$$\frac{d}{dt}\begin{bmatrix}\Psi^s_{r\alpha}(t)\\ \Psi^s_{r\beta}(t)\end{bmatrix}=\begin{bmatrix}-\frac{p_2(t)}{L_r} & -p_1(t)\\ p_1(t) & -\frac{p_2(t)}{L_r}\end{bmatrix}\cdot\begin{bmatrix}\Psi^s_{r\alpha}(t)\\ \Psi^s_{r\beta}(t)\end{bmatrix}+\begin{bmatrix}\frac{L_m p_2(t)}{L_r} & 0\\ 0 & \frac{L_m p_2(t)}{L_r}\end{bmatrix}\cdot\begin{bmatrix}i^s_{r\alpha}(t)\\ i^s_{r\beta}(t)\end{bmatrix} \quad (4.14)$$

where $p_1(t)=\omega(t)$ and $p_2(t)=R_r(t)$. In the output equations of **P**, the values of $\Psi^s_{r\alpha}(t)$ and $\Psi^s_{r\beta}(t)$ were assumed to be independently measurable. For such a LPV system, the following mixed sensitivity estimation structure can be given based on the contents of *Section 3.1.4.*:

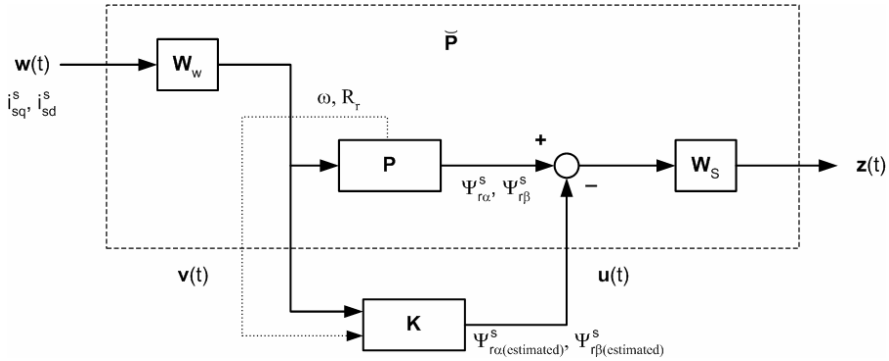


Figure 4.7. Mixed sensitivity structure of the observer synthetisation

The $\mathbf{W}_S$ filter presented in the one degree of freedom structure of *Figure 4.7* was chosen to be a low pass filter in respect to the theoretical concepts mentioned in the mixed sensitivity section, with a cutting frequency calibrated to the minimal frequency of the allowed error. Based on this, the chosen filter (*Figure 4.8*) was given in the form of (4.15).

$$\mathbf{W}_S(s) = \mathrm{diag}\left(\frac{300}{s+30}, \frac{300}{s+30}\right). \tag{4.15}$$

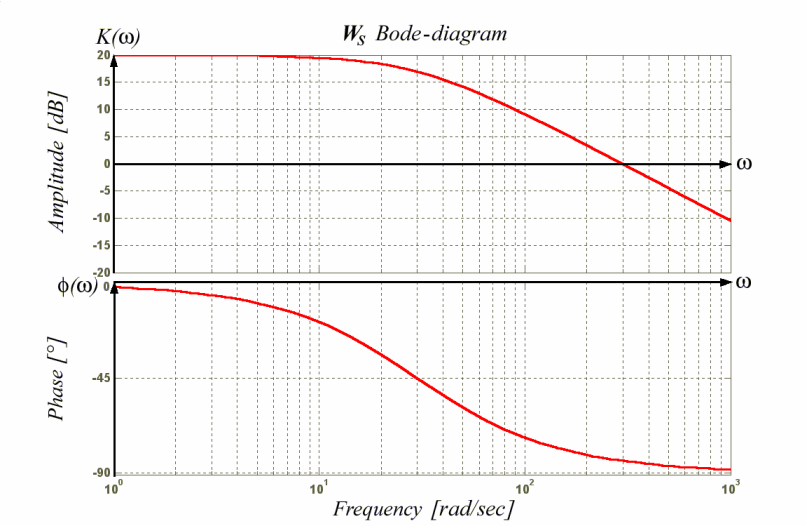


Figure 4.8. Bode-diagram of the $W_s(s)$ dynamical filter in the optimality problem of the observer

The 30Hz cutting frequency of (4.15) guarantees good disturbance filtering, while the 20dB DC gain ensures the attenuation of the error of the estimation. Furthermore, the value of $\mathbf{W}_W$ was simply given by

$$\mathbf{W}_W(s) = \mathrm{diag}(1,1), \tag{4.16}$$

because the filtering of the measurement noise is already provided by the low pass property and the robustness of the observer. For the problem given by *Figure 4.7*, a $2^2 = 4$ corner system described polytopical observer was designed with the help of the *hinfgs* function of the Matlab. The resulted system has 2 inner states, with 2 inputs and 2 outputs and its γ is $8.4852 \cdot 10^{-5}$. During the optimization the bounds of the parameters were chosen like in *Section 3.3* in consistence with the controller.

The estimation error of the observer was examined by Matlab simulations during both of the transient and steady state operation of the IM in a noisy environment. The obtained results of these simulations can be seen in *Figure 4.9(a) and 4.9(b)* in case of 100V, 50Hz voltage feed, and 0.5Nm load torque. The resulted mean error of the estimator was no more than 0.33% in these tests, when real life modeling noises were applied to the designed observer estimated system. Basically, during the operation of the motor, two kinds of noises arise: the inverter noise and the measurement noise. Because the inverter generates two corners-based PWM voltage signal, whose modulation frequency is finite, thus, even with the integrating property of the high motor inductances, the resulted feed will not be perfectly sinusoidal. This inverter-caused error can be modeled with an additive medium frequency dominant noise, as presented in *Figure 4.7*. This noise has been generated by the filtering of a white noise effect with $10V_{pp}$. The also important measurement noise (*Figure 4.10(b)*) was modeled by a high frequency dominant sensor noise with a 0.5 A amplitude which was also generated from white nose effect. However, the current technologies, like the Hall sensors of the LEN company, provide 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 implementation of the designed controller able with cheap and therefore less accurate sensors.

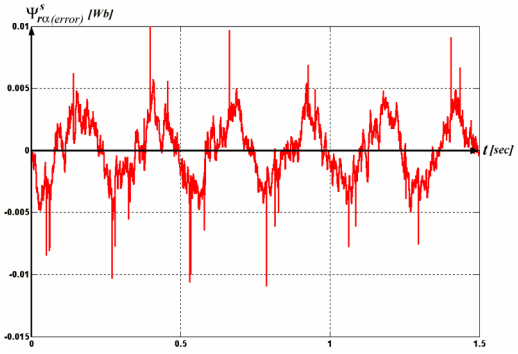


Figure 4.9(a). Estimation error of $\Psi^s_{r\alpha}$

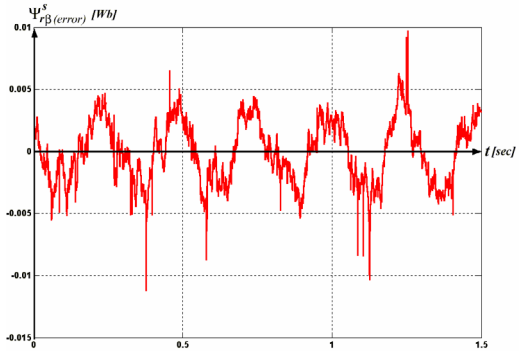
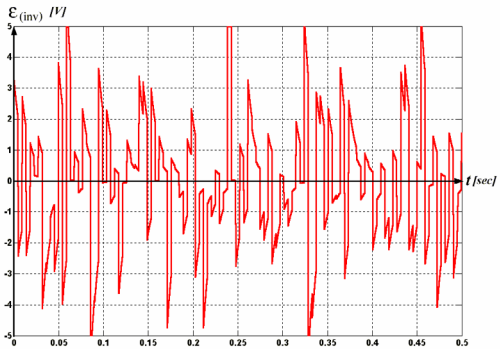


Figure 4.9(b). Estimation error of $\Psi^s_{r\beta}$



4.10(a). Modeled inverter noise

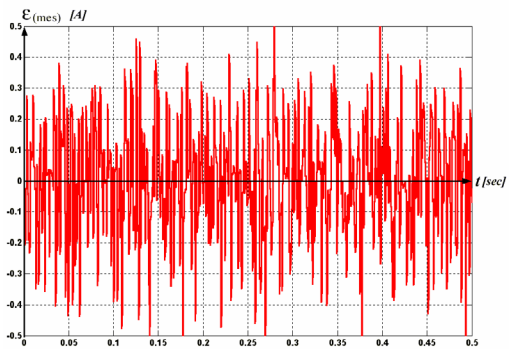


Figure 4.10(b). Modeled measurement noise

The robustness of the estimation was also investigated for 5% random parameter variance. The result of these simulations are given in *Figure 4.11(a) and 4.11(b)*.

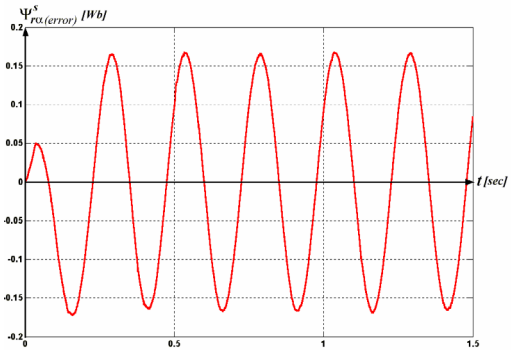


Figure 4.11(a). Estimation error of $\Psi^s_{r\alpha}$

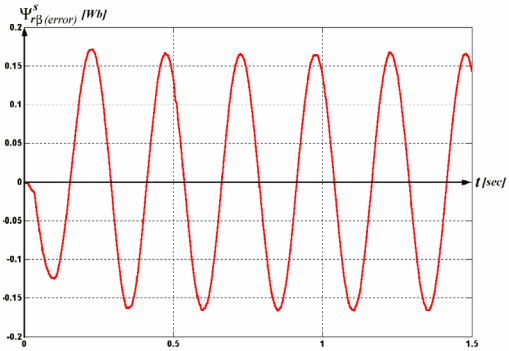


Figure 4.11(b). Estimation error of $\Psi^s_{r\beta}$

In this case, the mean relative error was 4.3%, which can be explained, based on the periodicity of the error, by an increased phase delay of the observer. With the growing of the variance of the parameters, the produced error rises, and in case of 10% it reaches significant, almost 15% relative value, which is acceptable

for this large uncertainty. Based on these results, it can be stated that the designed observer method is robust like the previously shown controller.

4.6.2. The design of the EKF

As it was mentioned in *Section 4.6*, effective LPV based method cannot be given for the joint estimation of ω and R_r with the rotor flux. Although, for the estimation of ω and R_r a parameter-affine form can be separately given with the help of (2.23) and (2.36), in which Ψ_{rd}^{fr} and $\left(i_{eff}^{r}\right)^2$ are the scheduling parameters. But this only produces parameter dependency in **B**(**p**), thus stable H_∞ state estimator cannot be designed, because the p_1 and p_2 dependent subsystems do not fulfill the (C1) and (C2) criteria given in *Section 3.1.2*. Thus, a EKF based estimator was aimed to realize the estimation of ω and R_r.

$$\frac{d}{dt}\underbrace{\begin{bmatrix}\Psi_{r\alpha}^{s}(t)\\ \Psi_{r\beta}^{s}(t)\\ i_{s\alpha}^{s}(t)\\ i_{s\beta}^{s}(t)\\ \omega(t)\\ R_r(t)\end{bmatrix}}_{\mathbf{x}}=\underbrace{\begin{bmatrix}-\frac{1}{\tau_r(t)}\Psi_{r\alpha}^{s}(t)-\omega(t)\Psi_{r\beta}^{s}(t)+\frac{L_m}{\tau_r(t)}i_{s\alpha}^{s}(t)\\ \omega(t)\Psi_{r\alpha}^{s}(t)-\frac{1}{\tau_r(t)}\Psi_{r\beta}^{s}(t)+\frac{L_m}{\tau_r(t)}i_{s\beta}^{s}(t)\\ \frac{\tau}{\sigma\cdot\tau_r(t)}\Psi_{r\alpha}^{s}(t)+\omega(t)\cdot\frac{\tau}{\sigma}\Psi_{r\beta}^{s}(t)-\frac{\left(\lambda\tau_r(t)+\tau_s\right)}{\sigma}i_{s\alpha}^{s}(t)\\ -\omega(t)\cdot\frac{\tau}{\sigma}\Psi_{r\alpha}^{s}(t)+\frac{\tau}{\sigma\cdot\tau_r(t)}\Psi_{r\beta}^{s}(t)-\frac{\left(\lambda\tau_r(t)+\tau_s\right)}{\sigma}i_{s\beta}^{s}(t)\\ \frac{\Psi_{r\alpha}^{s}(t)\cdot i_{s\beta}^{s}(t)-\Psi_{r\beta}^{s}(t)\cdot i_{s\alpha}^{s}(t)}{\tau_e}-\frac{F\omega(t)}{\tau_{mech}}\\ \left(\left(\frac{\Psi_{r\alpha}^{s}(t)-L_m i_{s\alpha}^{s}(t)}{L_r}\right)^2+\left(\frac{\Psi_{r\beta}^{s}(t)-L_m i_{s\beta}^{s}(t)}{L_r}\right)^2\right)\cdot R_{T0}\cdot R(t)-K_k\left(R_r(t)-R_0\right)\end{bmatrix}}_{\mathbf{f(x)}}+$$

$$+\underbrace{\begin{bmatrix}0&0&0\\ 0&0&0\\ \frac{1}{L_s\sigma}&0&0\\ 0&\frac{1}{L_s\sigma}&0\\ 0&0&-\frac{1}{\tau_{mech}}\\ 0&0&0\end{bmatrix}}_{\mathbf{B}}\cdot\underbrace{\begin{bmatrix}u_{s\alpha}^{s}(t)\\ u_{s\beta}^{s}(t)\\ T_{load}(t)\end{bmatrix}}_{\mathbf{u}} \qquad (4.17)$$

Based on the previously shown EKF theory, an estimation method was designed for the (4.17) 6-state NL 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 (4.17), in every prediction phase, the numerical solution of the equation system is calculated trough the Adams-Bashforth 3-step recursive approximation defined in *Section 3.2.3*.

However, the partial derivatives of the discretised form of the sate equation system, which are needed for the calculation of $\mathbf{P}(k+1|k)$ in equation (3.47) of the correction phase, are calculated by the Euler method-discretised form of the state equation system. This yields that

$$\mathbf{F}(k) = \left.\frac{\partial f_d(\mathbf{x})}{\partial \mathbf{x}}\right|_{\mathbf{x}=\hat{\mathbf{x}}(k)} = \left.\frac{\partial\left(\mathbf{I} + h \cdot f(\mathbf{x})\right)}{\partial \mathbf{x}}\right|_{\mathbf{x}=\hat{\mathbf{x}}(k)}, \tag{4.18}$$

because with the help of (4.18) a much less difficult equation can be obtained for the partial derivatives as if the derivatives of the recursive method presented in a 6×6 Jacobi matrix were used for this purpose. Furthermore, the derivatives do not need such great accuracy as the numerical approximation, because they only contribute to the success of the statistical correction, thus with the help of the **Q** matrix, their error can be eliminated. By considering the characteristics of the model, and after some *try and error*, the **Q** and **R** matrices were chosen to be the following:

$$\mathbf{Q}_{ij} = 0, \text{ except } \mathbf{Q}_{33} = \mathbf{Q}_{44} = \frac{0.0117h}{L_s\sigma}; \quad i, j \in \{0,1,\ldots,6\} \tag{4.19}$$

$$\mathbf{R}_{ij} = 0, \text{ except } \mathbf{R}_{11} = \mathbf{R}_{22} = 0.0205, \ \mathbf{R}_{33} = \mathbf{R}_{44} = 13.85; \quad i, j \in \{0,1,\ldots,4\} \tag{4.20}$$

The $h / L_s\sigma$ term in (4.19) was provided to eliminate the effect of the inverter noise on the first dynamical system. Furthermore, the reference state variables of the EKF were chosen to be the measured current signals and the observer provided fluxes. While, the outputs of the estimator were constructed to provide the unknown value of ω, R_r, and the noiseless values of $i^s_{r\alpha}(t)$ and $i^s_{r\beta}(t)$.

The effectiveness of the EKF was examined for the specific model and the results of these simulations for the estimation errors of ω and R_r are given in *Figure 4.12(a) and 4.12(b)*. The relative greatness of these errors were 0.41% for ω and 0.08% in the case of R_r. The remained noise load on $i^s_{r\alpha}(t)$ and $i^s_{r\beta}(t)$ was also investigated (see *Figure 4.12(c) and 4.12(d)*) and it was concluded that the average noise attenuation is 98%.

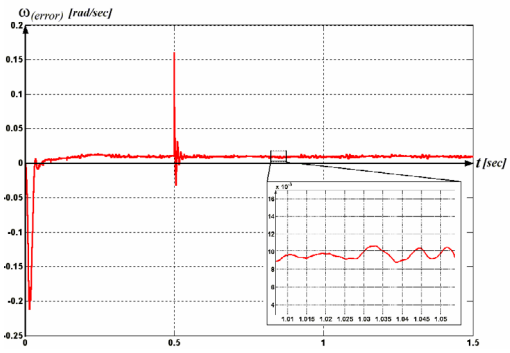


Figure 4.12(a). Estimation error of ω

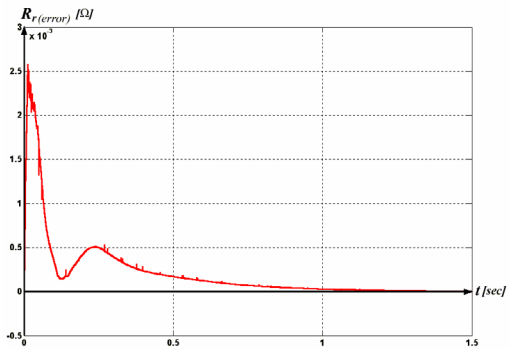


Figure 4.12(b). Estimation error of R_r

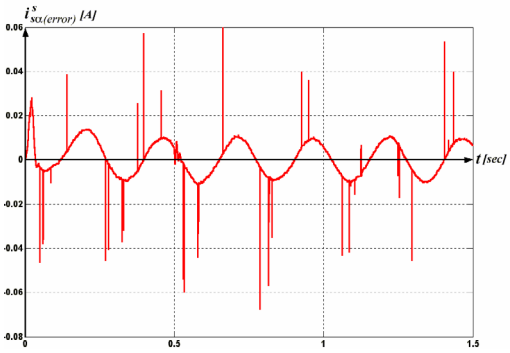


Figure 4.12(c). Filtering error of $i^s_{r\alpha}$

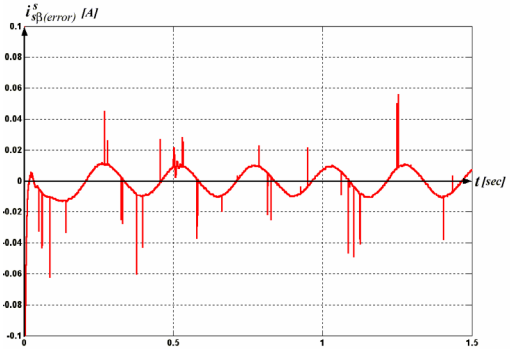


Figure 4.12(d). Filtering error of $i^s_{r\beta}$

Furthermore, the robustness of the algorithm was tested in case of 5% variance in system parameters. For this case, the estimation errors are given in *Figure 4.13(a) and 4.13(b)*. Here, the relative error was 0.8% for ω and 0.33% for R_r. Unfortunately, besides of these, the previously experienced great noise attenuation property was lost in this case (see *Figure 4.13(c) and 4.13(d)*), because the method has amplified the noise with a gain of approximately 2.6, which is relative 5% to the greatness of the stator current. This is directly in connection with the increased phase delay of the filter based on the periodicity of the resulted error, which greatness is still acceptable for this large parameter uncertainty. Based on these, it can be stated that the two designed methods have shown robustness,

effectiveness, and stability in the intensive simulations, thus their closed loop structure should have similar properties in the operation range. To prove this statement, the results of closed loop simulations are given in the next section instead of rigorous theoretical proofs.

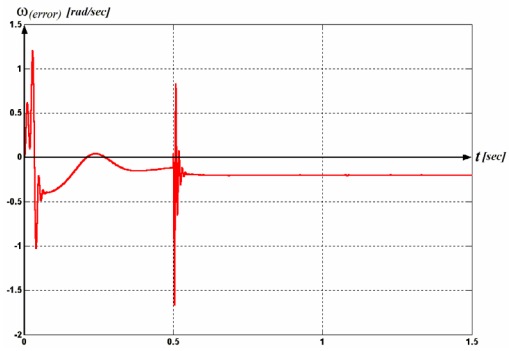


Figure 4.13(a). Estimation error of ω

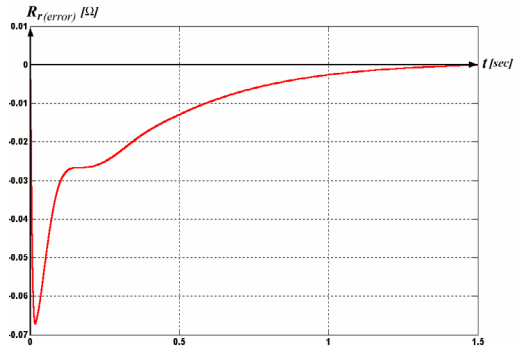


Figure 4.13(b). Estimation error of R_r

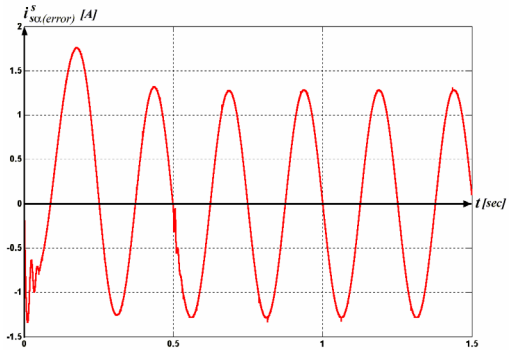


Figure 4.13(c). Filtering error of $i_{r\alpha}^{s}$

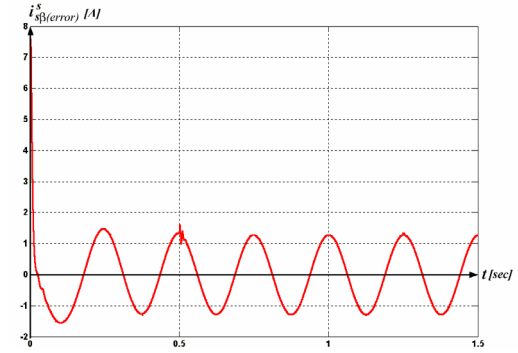


Figure 4.13(d). Filtering error of $i_{r\beta}^{s}$

4.6.3. Estimation of the load torque

Nowadays, several published papers [6, 27] which describe the design of H_∞ controller for the IM, ignore the estimation of the load torque. They either conceal this or they assume the load to be known in the system. This can be accepted in such applications where the torque does not really change and only lightly fluctuates around a constant load during the operation. However, in case of a 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 had been used for the estimation of T_{load}, then such a looped system would have been produced which amplifies its own error instead of convergence in the estimation. Based on this, the value of $\Psi^{fr}_{rd_{ref}}$ and ω_{ref} are used in the calculation instead of the estimated quantities, and after some transformations the following simple equation can be used for guessing the load:

$$T_{load} = \frac{2L_r}{3pL_m} \cdot \left(\Psi^{fr}_{rd_{ref}} \cdot i^{s}_{sq}(t) \right) - F\omega_{ref} \tag{4.21}$$

It is important to mention that these reference signals cannot be used directly just only in that case if 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 4.6.1 and 4.6.2* lose their connection with the reality because of the step like errors of the torque calculation. Because of this reason, the weighting filters were chosen similarly like the speed limiting filter of the references in the next section, because faster performance than this will not be expected from the system. Moreover, the result of the (4.21) calculation method needs further filtering as well to eliminate the glitches appearing because of other reasons. For some *try and error*, the post filtering was calibrated to be the

$$\frac{44}{s+45}, \tag{4.22}$$

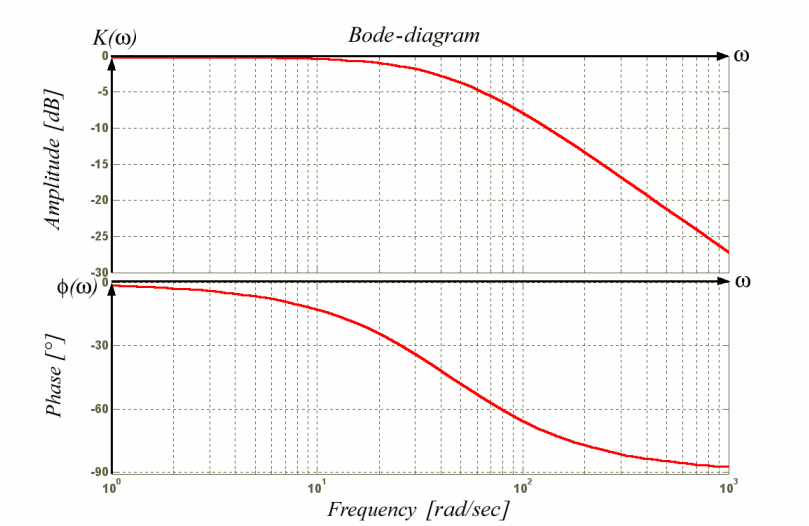
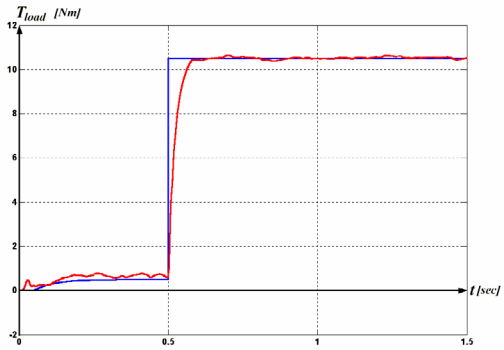
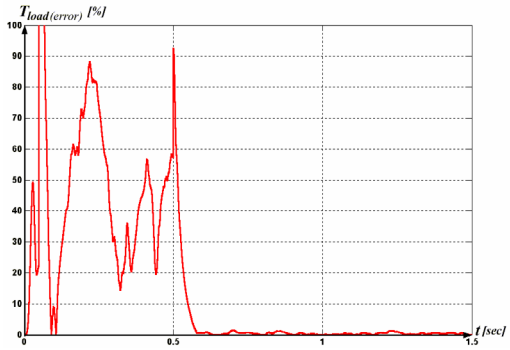


Figure 4.14. Bode-diagram of the smoothing filter of load torque estimation

dynamical filter (see *Figure 4.14*). By connecting the produced algorithm to the IM, the torque tracking produced the results given in *Figure 4.15(a). and 4.15(b)*, whose steady state error is relatively low.



4.15(a). Tracking (*blue*) of Figure the estimation (*red*) for T_{load}



4.15(b) Relative error of T_{load} estimation

4.7. Tuning parameters

Because of the existing errors in each level of the interconnected structure, the looped control system is of course 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. Even in *Section 4.3*, the amplification of the error signal was emphasized at the reference transformation method that produces the input of the controller, to calibrate the produced reference tracking error of the controller with respect to its norm, which is responsible for the remaining gain of the control error. By introducing an integrator term on the tracking error signal, this external amplification can be avoided, but as it was mentioned, this approach would make the system slower and needlessly complicated in terms of controller states. Moreover, it was experienced that it is worth amplifying the error signal in two steps. Firstly, by the amplification of the tracking errors of ω_{ref} and $\Psi^{fr}_{rd_{ref}}$ before the reference transformation, and then after the reference transformation, by applying gains for the deviations from $i^{fr}_{sd_{ref}}$, $i^{fr}_{sq_{ref}}$. In this way, more accurate reference tracking can be achieved. However, it is not worth increasing the gains limitlessly, because by this approach the controller can achieve a state when it begins aggressively pulling around the stator voltage to input more energy into the system because of the 300V limit in the

control signal. 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 to make the reference tracking optimal in this way. The exact values of these compensations can be most perfectly calculated in the case of large torque and flux signals, because the compensation effect is insignificant for small values of these signals. The exact values of these compensations will be given with the simulations in the next section. It is also 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. As it was mentioned in *Section 3.1.6*, the controller states must be bounded also to improve the adaptability and speed of control during the transients. For the speed limitation of the reference the (4.23) filter (*Figure 4.16*) was found to be the most perfect, because it was experienced that nearly half of the fastest time constant of the system, $\tau_r = L_r / R_r = 43\text{msec}$ provides fast reference tracking without noticeable overshoots.

$$\frac{15}{s+15} \tag{4.23}$$

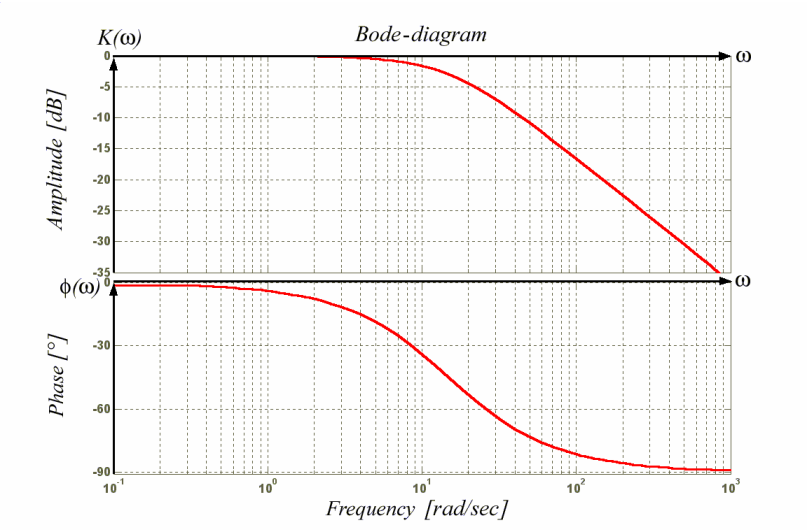


Figure 4.16. Bode-diagram of the reference speed limiting filter

5. Simulation results

5.1. Closed loop system model in Simulink

The model of the whole interconnected closed loop system was constructed in Simulink with the continuous model of the IM based on (4.17) in order to made possible investigations by Matlab simulations. The resulted motor model which is given with the built in tools is presented in *Figure 5.1*.

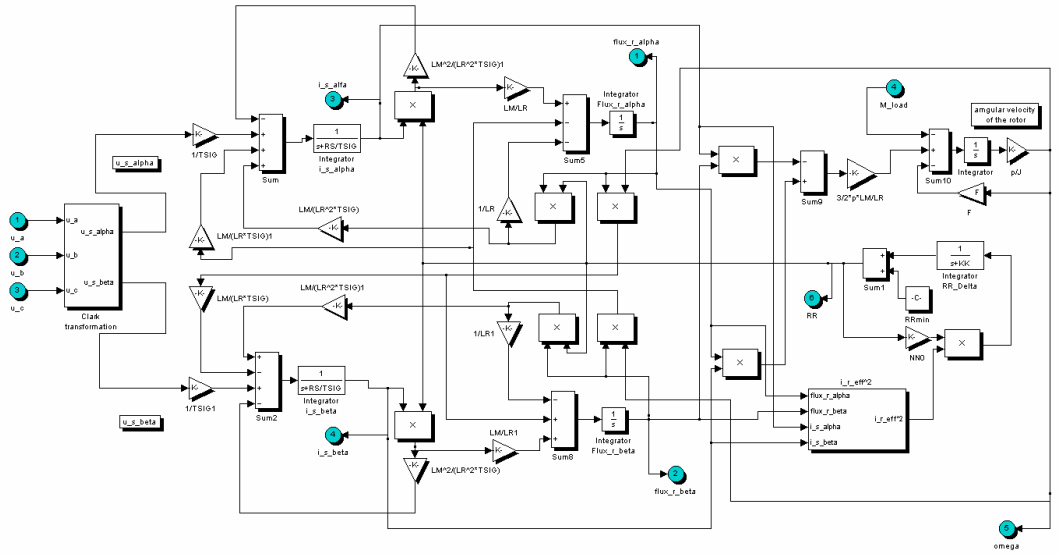


Figure 5.1. Simulink model of the induction motor

The specific motor parameters used during the simulations were as follows:

$L_r = 0.13$ H	$R_s = 1.86\ \Omega$	$J = 0.02$ Nm	$c = 0.21\ {}^{J}/_{kgK}$
$L_s = 0.13$ H	$R_0 = 3\ \Omega$	$F = 0.001$	$m = 4$ kg
$L_m = 0.12$ H	$p = 3$	$K_K = 3.5$	$T_0 = 18°$

The above presented values are the parameters of a low power and low speed experimental IM. For the simulations, the realistic parameters of even huge motors can be used of course, if these values are identified with the help of measurements and identification methods. In this case, the retuning of the controller must be done to provide satisfying results.

The model of the closed loop system was also constructed in the form of interconnected *subsystems* which can be seen in *Figure 5.2*.

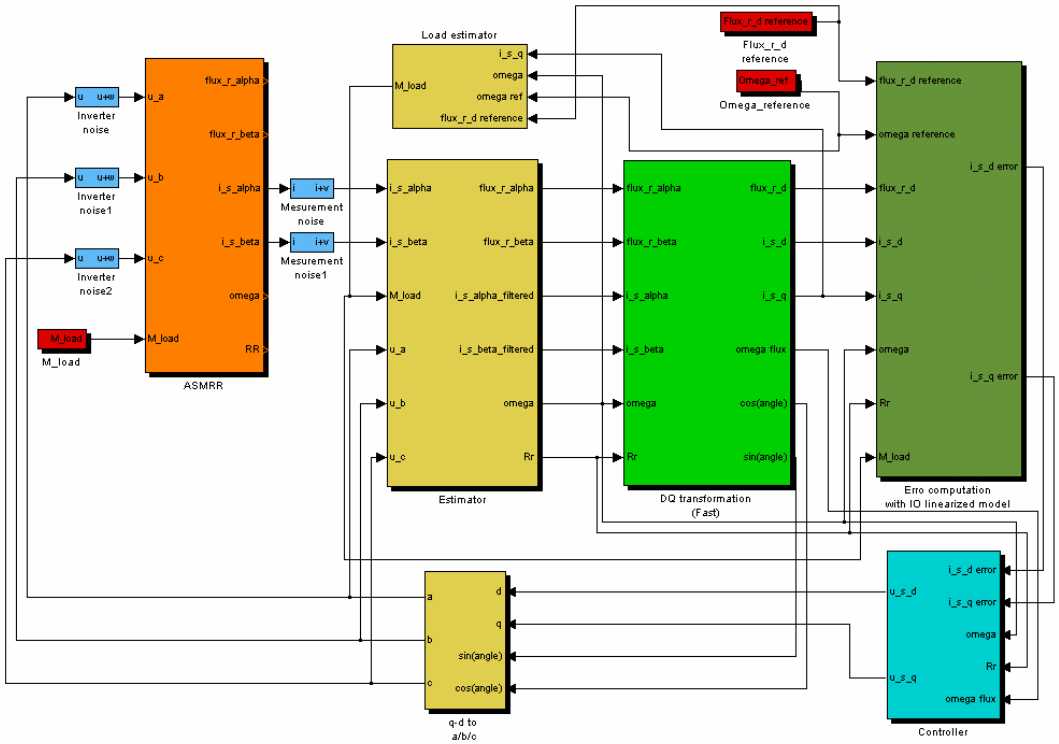


Figure 5.2. Simulink model of the closed loop controller driven drive

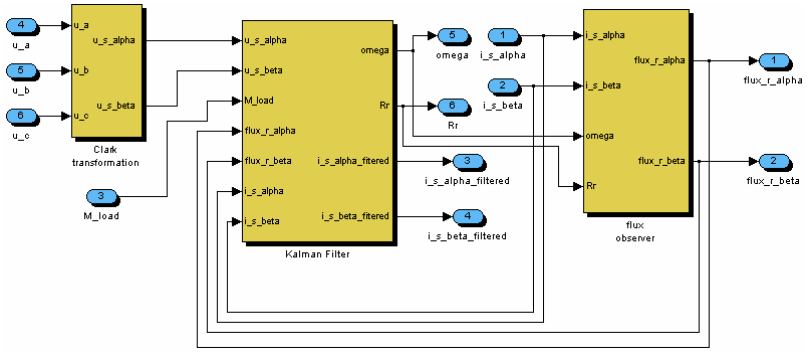


Figure 5.3. Simulink model of the interconnected estimation method

Since, such a greatly complicated system can only be implemented digitally in a real life application, each of these 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 $100\mu sec$. In the following section, it will be shown, that this cycle time can be reached in the current technology level. The measured signals are digitalized with the same sampling time, to ensure the correct functioning of the discretised estimation methods. Since the dynamics of the system are much slower than this size of time step, the discretisation has to approximate the continuous operation well. The discretised algorithms were

realized as *S-function subsystems* (*Figure 5.3-5.8*) of the *Simulink environment*, whose script code is given in *Appendix A-F*.

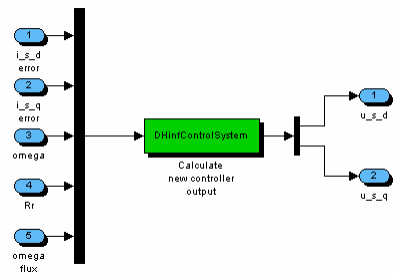


Figure 5.4. Simulink model of the H_∞ controller

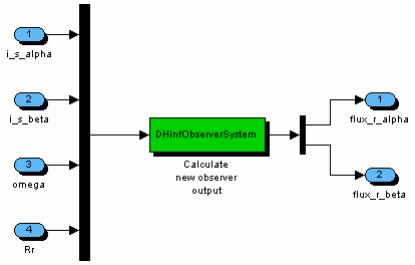


Figure 5.5. Simulink model of the H_∞ observer

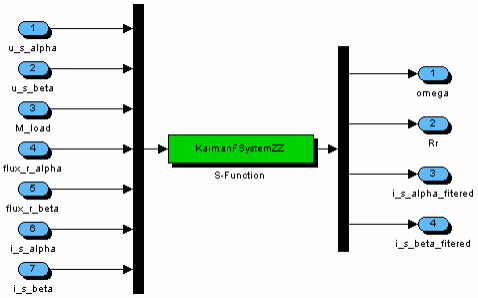


Figure 5.6. Simulink model of the EKF

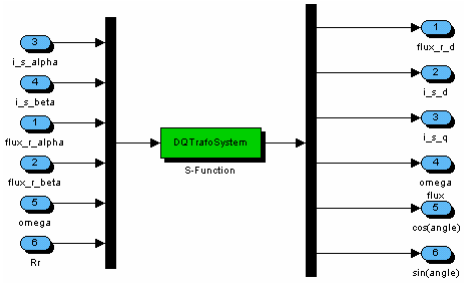


Figure 5.7. Simulink model of the field orientation model

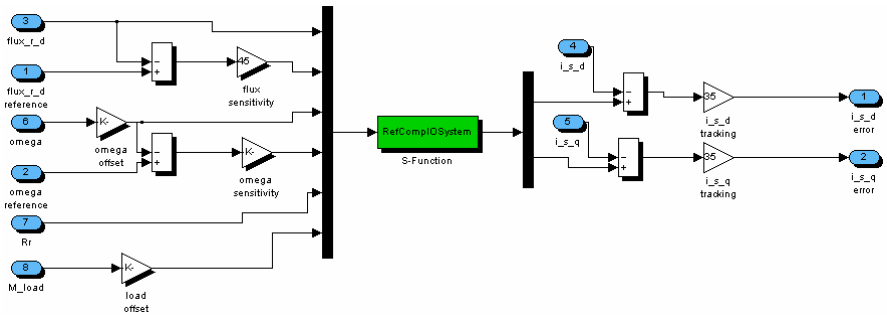


Figure 5.8. Simulink model of the reference transformation module with tuning parameters

It is important to mention that even the H_∞ LPV theory is defined only for continuous time, the controllers and estimators produced by this approach are functioning well in discrete time as well, due to the small step size of discretisation. This discretisation is done online inside of programmed Matlab S-functions. This is done as follows: from the polytopical form of the methods the instant parameter value defined

$$\dot{\mathbf{x}}_K(t) = \mathbf{A}_K\left(\mathbf{p}(t)\right)\cdot\mathbf{x}_K(t) + \mathbf{B}_K\left(\mathbf{p}(t)\right)\cdot\mathbf{u}_K(t), \tag{5.1}$$

$$\mathbf{y}_K(t) = \mathbf{C}_K\left(\mathbf{p}(t)\right)\cdot\mathbf{x}_k(t), \tag{5.2}$$

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

$$\dot{\mathbf{x}}_K(k) = \underbrace{e^{A_K(\mathbf{p}(k))h}}_{\Phi_k(\mathbf{p}(k))}\cdot\mathbf{x}_K(k) + \underbrace{\mathbf{A}_K^{-1}\left(\mathbf{p}(k)\right)\left(e^{A_K(\mathbf{p}(k))h} - \mathbf{I}\right)\mathbf{B}_K\left(\mathbf{p}(k)\right)}_{\Gamma_k(\mathbf{p}(k))}\cdot\mathbf{u}_K(k), \tag{5.3}$$

$$\mathbf{y}_K(k) = \mathbf{C}_K\left(\mathbf{p}(k)\right)\cdot\mathbf{x}_k(k), \tag{5.4}$$

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

$$\Phi_K\left(\mathbf{p}(k)\right) \approx \mathbf{I} + h\cdot\mathbf{A}_K\left(\mathbf{p}(k)\right) + \frac{h^2\cdot\mathbf{A}_K^2\left(\mathbf{p}(k)\right)}{2}, \tag{5.5}$$

$$\Gamma_K\left(\mathbf{p}(k)\right) \approx h\cdot\mathbf{B}_K\left(\mathbf{p}(k)\right) + \frac{h^2\cdot\mathbf{A}_K\left(\mathbf{p}(k)\right)\cdot\mathbf{B}_K\left(\mathbf{p}(k)\right)}{2}, \tag{5.6}$$

matrices. In this way, the continuous controller and observer approximated in second order operate with the continuous dynamic properties in discrete time.

The system also incorporates several additional modules, such as the vector transformation blocks (*Figure 5.9 and 5.10*), the inverter noise producing subsystems (*Figure 5.11*), and the modules responsible for the adding of the measurement noise (*Figure 5.12*) to the current signals.

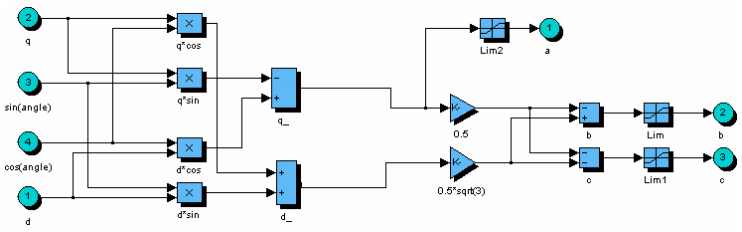


Figure 5.9. Field orientation / 3-phase transformation computing module in Simulink

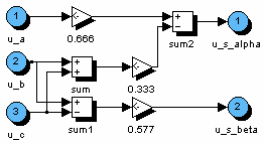


Figure 5.10. Simulink model stator-fixed orientation

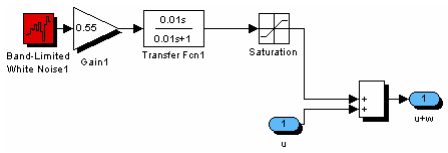


Figure 5.11. Inverter noise generating module

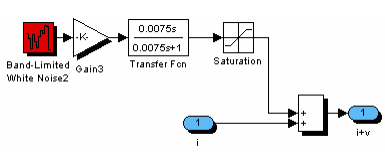


Figure 5.12. Measurement noise generating module

5.2. Reference tracking in noiseless environment

The tuning parameters of the system described in *Section 4.7* were calibrated in a noiseless 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 with respect to the large reference signals during the startup. The sensitivity for the stator 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 investigate the system behavior on the full 4/4 operation range of the IM, but also test the system on the bounds of its operation. The results are given in *Figure 5.13(a-l)*.

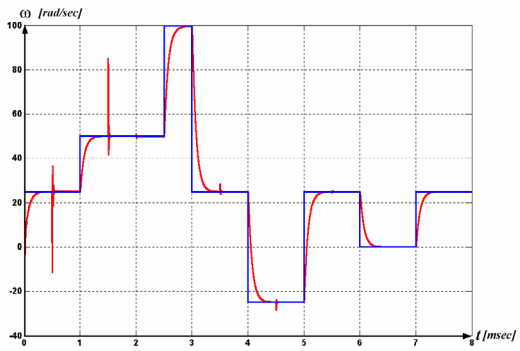


Figure 5.13(a). Reference (*red*) tracking (*blue*) for ω

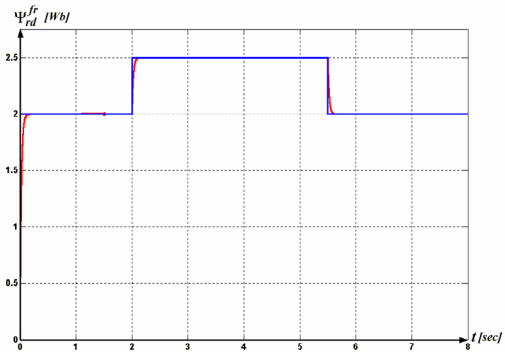


Figure 5.13(b). Reference (*red*) tracking (*blue*) for Ψ_{rd}^{fr}

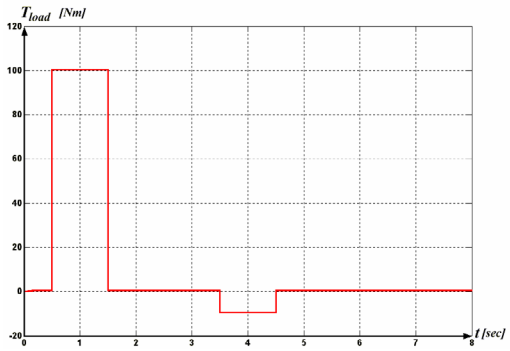


Figure 5.13(c). Change of load

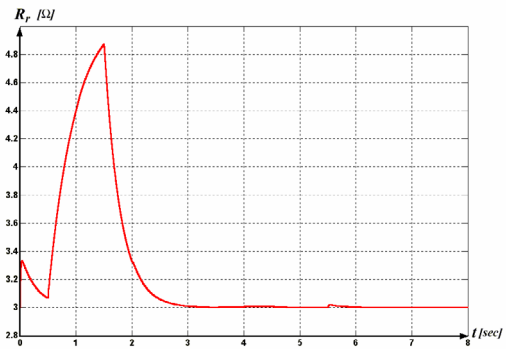


Figure 5.13(d). Change of R_r

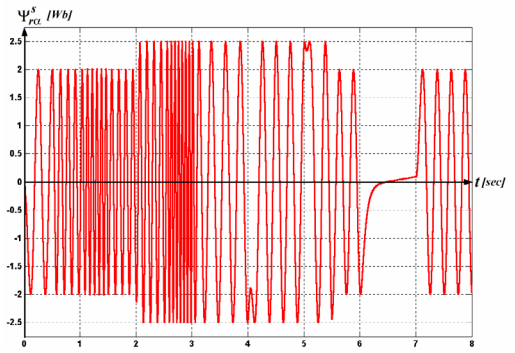


Figure 5.13(e). Change of $\Psi^s_{r\alpha}$

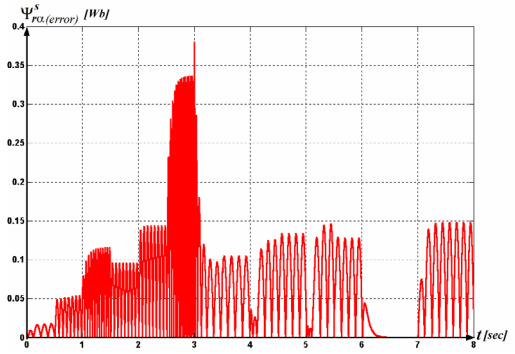


Figure 5.13(f). Estimation error of $\Psi^s_{r\alpha}$

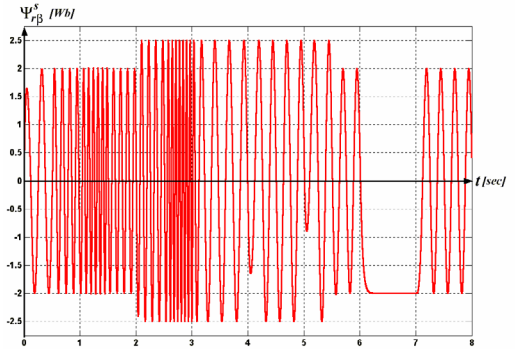


Figure 5.13(g). Change of $\Psi^s_{r\beta}$

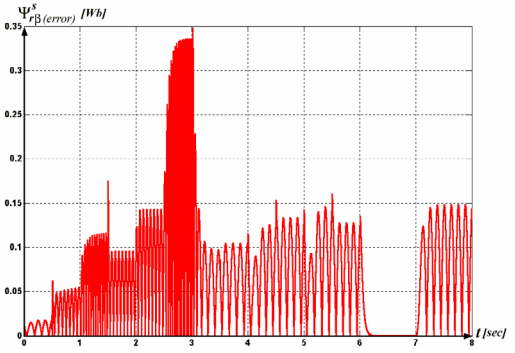


Figure 5.13(h). Estimation error of $\Psi^s_{r\beta}$

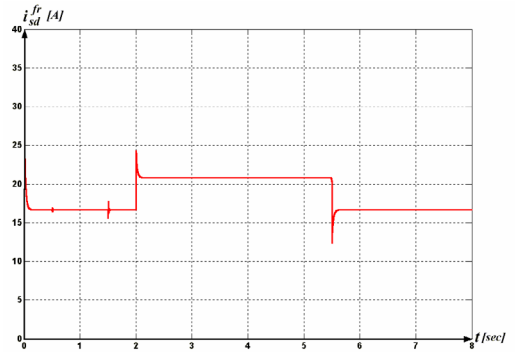


Figure 5.13(i). Change of i^{fr}_{sd}

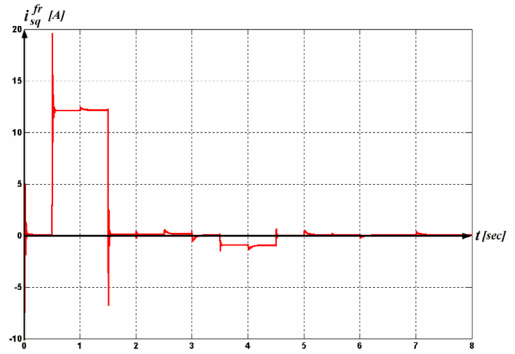


Figure 5.13(j). Change of i^{fr}_{sq}

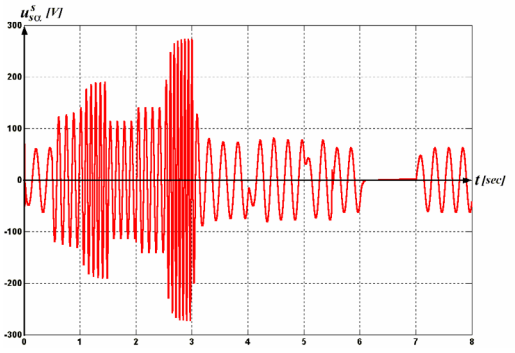


Figure 5.13(k). Change of $u^s_{s\alpha}$

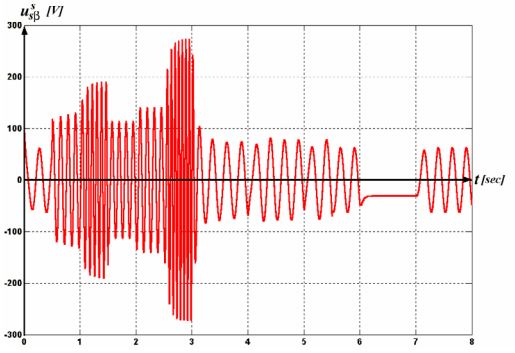


Figure 5.13(l). Change of $u^s_{s\beta}$

During the simulations the load torque is given with a slight clutching to the system in the first time moments (*Figure 5.13(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 raise 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 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 is instantly followed by the real speed of the motor (see *Figure 5.13 (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. At the 6. sec, the controller holds the given load with 0 speed. During the full operation, the speed and load changes have no significant effect on the flux tracking, although Ψ_{rd}^{fr} is crucial in the production of the electromagnetcial torque. By considering the flux tracking, it can also be concluded that even for extreme and rapid changes in the operation, the flux remains stable (see *Figure 5.13(b)*). In this way, it can be said that the tracking performance of the controller is extremely good. Moreover, by *Figure 5.13(k) and 5.13(l)*, the controller never gives out the maximal $300V_{peak}$, and during startup the current load of the power system is not significant, which provides economical operation of the drive. The robustness of the control structure was also tested in this noiseless case with the same reference signals and with 5% parameter variance. The results are given in *Figure 5.14(a) and 5.14(b)*.

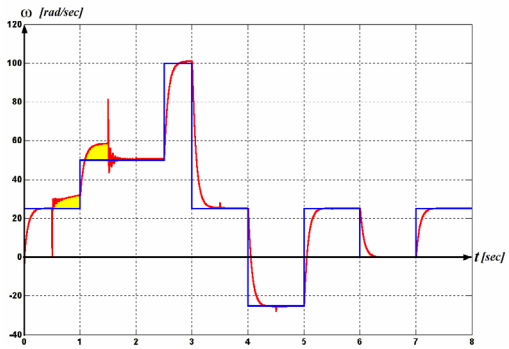


Figure 5.14(a). Reference (*red*) tracking (*blue*) for ω

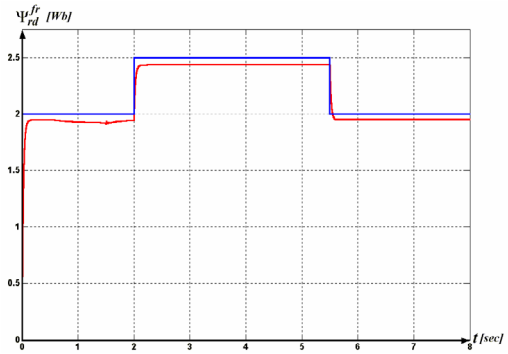


Figure 5.14(b). Reference (*red*) tracking (*blue*) for Ψ_{rd}^{fr}

The offset error, which is caused by the critical 200 times change in load, is given in yellow on *Figure 5.14(a)*. As it can bee seen, only for this case, the performance lost somewhat of its optimality, but this offset error with the tracking error of the flux can be easily eliminated by a slight calibration. This was the reason why the system was designed with external tuning capability, which makes possible to adjust the performance in case of any parameter uncertainty. Moreover, because of the absence of the integrator term, even this kind of tuning error caused reference tracking error cannot destabilize the drive. Based on these, the robustness and stability of the whole system is proved for the noiseless case. The script code of the blocks used in these simulations is given in *Appendix A-F*.

5.3. Reference tracking in noisy environment

The operation of the drive was also tested in a noisy environment. For the reference signals of the previous case the simulations were completed with the additive noises described in *Section 4.6.1*. The results of the reference tracking on *Figure 5.15(a) and 5.15(b)* are speaking for themselves.

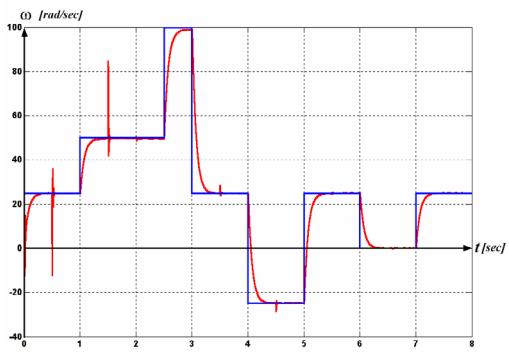


Figure 5.15(a). Reference (*red*) tracking (*blue*) for ω

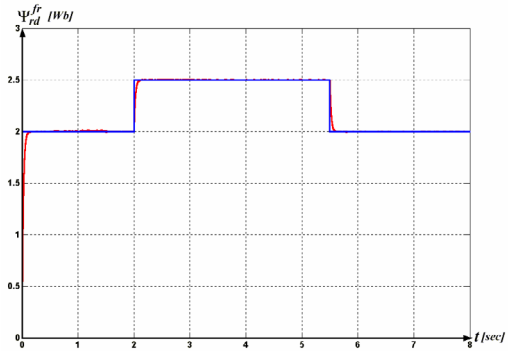


Figure 5.15(b). Reference (*red*) tracking (*blue*) for Ψ_{rd}^{fr}

For either tracking, the relative steady state offset error did not exceeded 1.5%. Since during the operation, nearly the same flux, voltage, and currents signals occurred, then, instead of presenting these again, only their estimation error is shown on *Figures 5.16(a-f)*. The estimation error of the rotor flux given in *Figure 5.16(a) and 5.16(b)* had a mean value of 0.35%, while the remaining noise error given in *Figure 5.16(b) and 5.16(c)* in the stator currents showed 88% noise attenuation. The 0.2% relative error of speed tracking presented a better result in

the closed loop control than the accuracy of the open loop estimator structure. This phenomenon can be explained by the low pass property of the H_∞ methods and by the small gain theory. The relative estimation error of R_r has remained 0.08%.

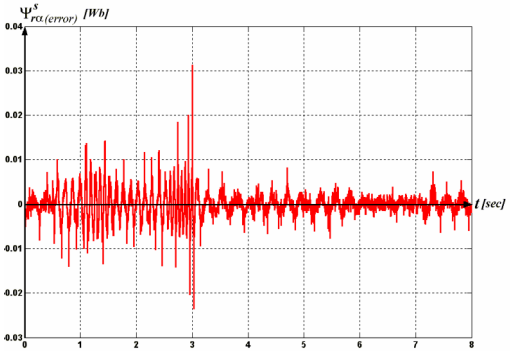


Figure 5.16(a). Estimation error of $\Psi^s_{r\alpha}$

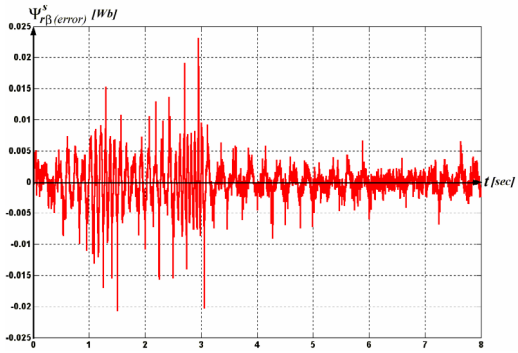


Figure 5.16(b). Estimation error of $\Psi^s_{r\beta}$

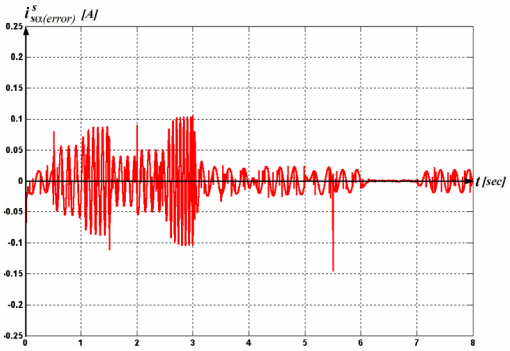


Figure 5.16(c). Filtering error of $i^s_{s\alpha}$

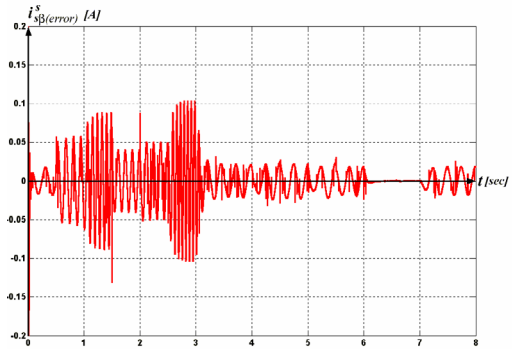


Figure 5.16(d). Filtering error of $i^s_{s\beta}$

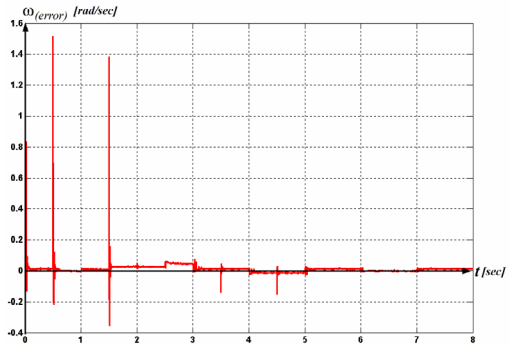


Figure 5.16(e). Estimation error of ω

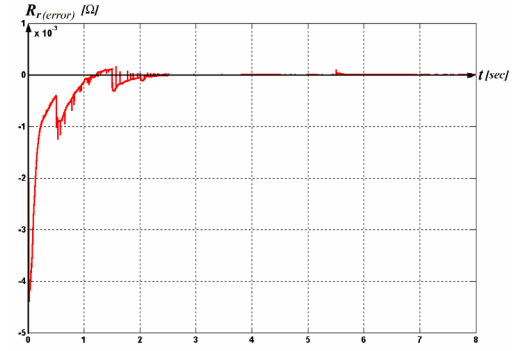


Figure 5.16(f). Filtering error of R_r

It can be clearly seen, that although the reference tracking error has increased in the noisy environment, the performance has only changed slightly in comparison with the pervious noiseless case. The robustness of the drive was also

tested with 5% parameter uncertainty. The results of reference tracking are given in *Figure 5.17(a) and 5.17(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 the case of such heavy noises. It must be mentioned that by taking some performance tests on the system, this offset error can be eliminated 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.

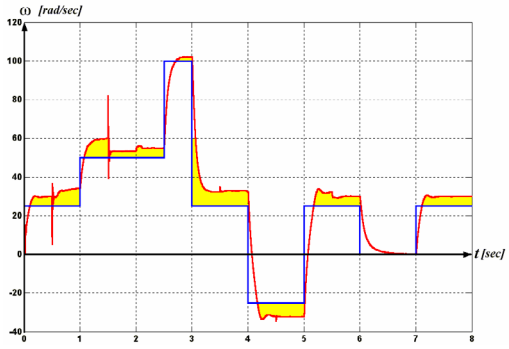


Figure 5.17(a). Reference (*red*) tracking (*blue*) for ω

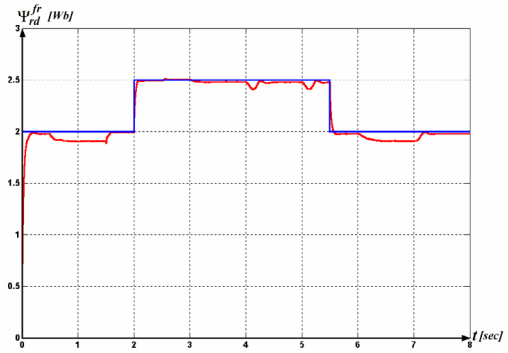


Figure 5.17(b). Reference (*red*) tracking (*blue*) for Ψ_{rd}^{fr}

The script code of the blocks used in these simulations is given in *Appendix A-F*.

5.4. Reference tracking capabilities with torque estimation

The calculation of the load torque, as it was stated many times, is a great theoretical challenge in this problem, thus even if the open loop torque estimator system (*Figure 4.15(a)*) operates well, the reference tracking of the closed loop control does not provide such good results as in the pervious cases. In this case, the simulations give the results presented in *Figure 5.18(a) and 5.18(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.

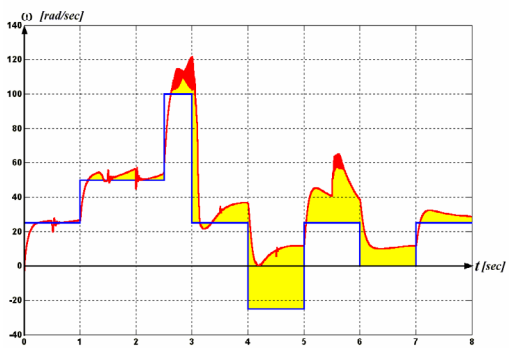


Figure 5.18(a). Reference (*red*) tracking (*blue*) for ω

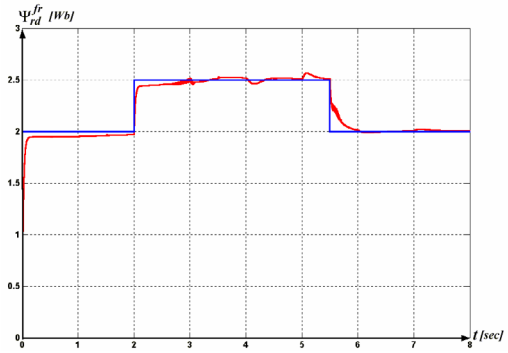


Figure 5.18(b). Reference (*red*) tracking (*blue*) for Ψ_{rd}^{fr}

In case of clutched, constant torque, the control structure presents an effectiveness that could be seen for the open loop case (*Figure 5.19(a) and 5.19(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. Thus, this control structure gives more than the [6, 27] 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.

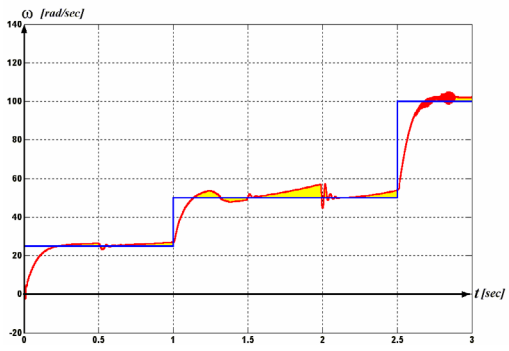


Figure 5.19(a). Reference (*red*) tracking (*blue*) for ω

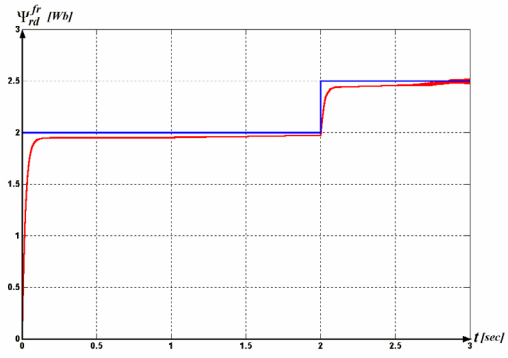


Figure 5.19(b). Reference (*red*) tracking (*blue*) for Ψ_{rd}^{fr}

The script code of the block used during these simulations is given in the *Appendix A-F*.

6. Implementation of the controller

In the previous chapters the designed controller has proved in several simulations that its structure provides a suitable solution for the control of IM drives. Till this point, only theoretical aspects of the design were considered with purely simulation results, but to show the real value of this structure the realization of the algorithm is unavoidable. Based on this, in the following section, the implementation of the designed controller is given on Digital Spectrum motion control development kit powered laboratory induction motor drive. The heart of this self-assembled device is a TMS320F243 DSP containing *evaluation module* (EVM), which serves as the microcontroller platform of the ANSI C based algorithm realization.

6.1. The experimental laboratory drive

Before any kind of realization begins, firstly the physical environment of the implementation is thoroughly investigated to get to know the possibilities and needs of the hardware which help a lot to suit the theoretical result to it. Therefore in the following parts the above mentioned laboratory drive is going to be briefly described by outlining the most important features that can help to complete the realization.

In *Figure 6.1* the above mentioned drive is presented which mechanism is explained by *Figure 6.2*.

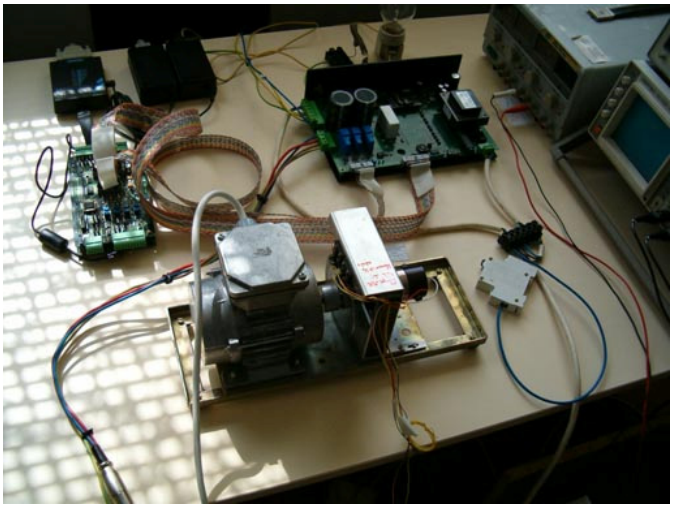


Figure 6.1. The assembled laboratory drive

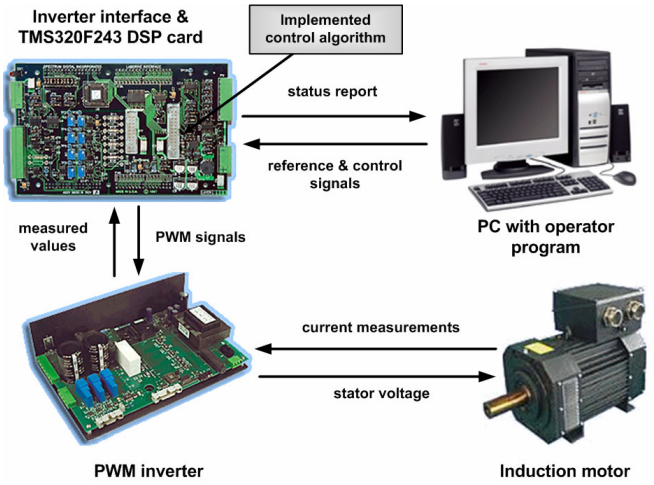


Figure 6.2. The mechanism of the TMS320F243 controlled IM drive

The drive consists of 4 main parts:

- A 3 phase, 4 poles induction motor which control is needed to be achieved for flux and speed.
- A PWM voltage source inverter (VSI), which is the power element of the system and also provides measurements on the currents and on the inverter bus voltage. (the bus voltage is needed for the PWM realization)
- 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:

6.1.1. The induction motor

The considered 4 poles induction motor, seen in *Figure 6.2*, 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 38,1Ω / 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 rating connection with 230V/5A rms capability.

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 Labdrive 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 6.1*) 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 pair of 5V powered phototransistors and LEDs to generate impulses during the rotation of the disc. Impulses are generated based on that if there is a slot between the transistor and the LED than the transistor is triggered by the photon stream provided energy of the LED, or in the other case it 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 rotor speed.

The 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. This gives the possibility to use such dynamic testing that was seen in the simulations.

6.1.2. 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 integrated sensors to provide a good solution to be the power element of any sensorless IM drive [22].

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.

The power circuitry of this inverter has a common design seen in *Figure 6.3*, 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 which provides a rated bus voltage of 350VDC & 10A. This voltage is chopped by the IGBTs in the sequence of the provided triggering signals. In this way it can be achieved, that the average voltage of the resulted high-power output waveform over any small region is directly proportional to the average voltage of the control signal in that region. This is illustrated by *Figure 6.4*.

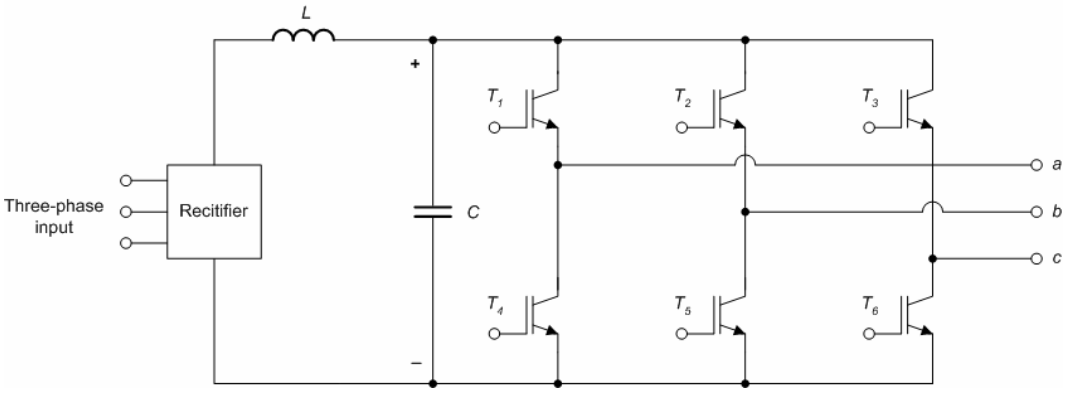


Figure 6.3. General structure of 3-phase PWM-VSI inverters with IGBTs

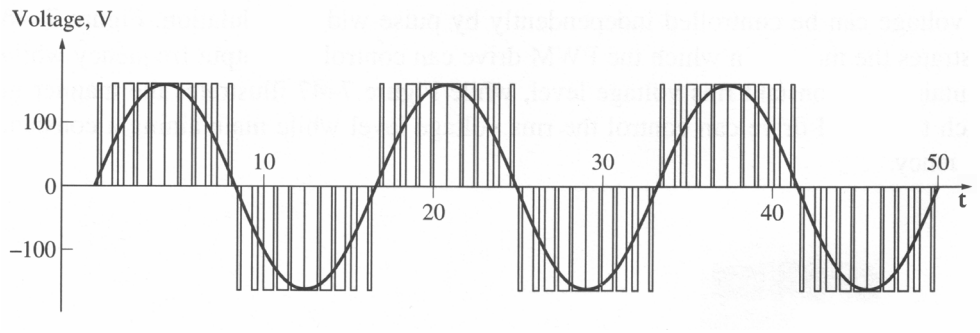


Figure 6.4. Output PWM waveform of one phase with the control signal to be realized

Moreover, because the IGBTs, as solid sate devices need time, called dead time, to turn off, during the realization of any triggering sequence this feature must be considered at first. In this power element, the dead time of the used IGBTs are 2 μsec but because the high and the low side drivers are optically isolated, thus with the 1 μsec transient time of the optoisolation, the triggering signals must be interleaved with a 3 μsec deadband.

The control logic of the inverter which has an onboard separated power supply, provides intelligent control of this power element. It is connected to the interface module by two cables on which the communications is provided on analog signals. This logic makes up several important features of the inverter.

As it was mentioned, the inverter is capable dynamic breaking, which is done through an externally provided resistor. This resistor is switched on the bus with an IGBT, triggered by an optically isolated driver. Because in 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 by a digital logic implemented as a state machine. This logic enables the bus voltage for the IGBTs in case of fault less 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 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.

Moreover, the inverter has 3 Hall effect current sensors with ±10 A sensing range on each of the three-phases of the motor feed and it also contains an optically isolated bus voltage sense. These measured signals are provided to the interface and the EVM through analog connection.

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. Besides the signaling of the fault, the logic do nothing to avoid further faulty operation. This is done by the interface card.

6.1.3. The Inverter Interface Card

The next component in the control chain (*Figure 6.2*) is the Labdrive interface card which is the bridge between the microcontroller, the inverter, and other external sensors. It supports the control of AC induction, DC brushless, and switched reluctance motors as well [21]. The main task 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. The analog sensor signals are conditioned through user adjusted offset and gain provided by built in amplifier stages with potenciometers, and each of the input channels of the card is protected against over voltage.

Moreover, 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 case any fault detection, the two element is totally separated and protected until hard reset of the EVM.

6.1.4. 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 space vector (SV)-PWM.

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 [32]. With 20 million instructions per second (MIPS), the F243 DSP controller offer 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 for additional functions. Therefore, The F243 offers reliability and programmability, and as a digital microcontroller, in contrast to analog control systems it is not effected by performance degradation due to aging, component tolerance, and drift. The instruction set of these DSP controllers, which incorporate 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 reduces development time.

The single DSP chip is supplemented by a peripheral board, and the hardwired connection of these elements produces the TMS320F243 EVM system.

6.1.4.1. 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 has its 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 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 in order to make the produced program more understandable.

6.1.4.2. 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 a 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 commercially PCs, has separated program, data, and I/O space memory. This structure is very important, because mainly simple programs are running on these DSP 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 multiplication, 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 the today's 2-3Ghz processors of the PCs, it can totally use 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 case of the F243, 64K word of program, data, and I/O memory is presented which configuration can be seen in *Figure 6.5*. 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.

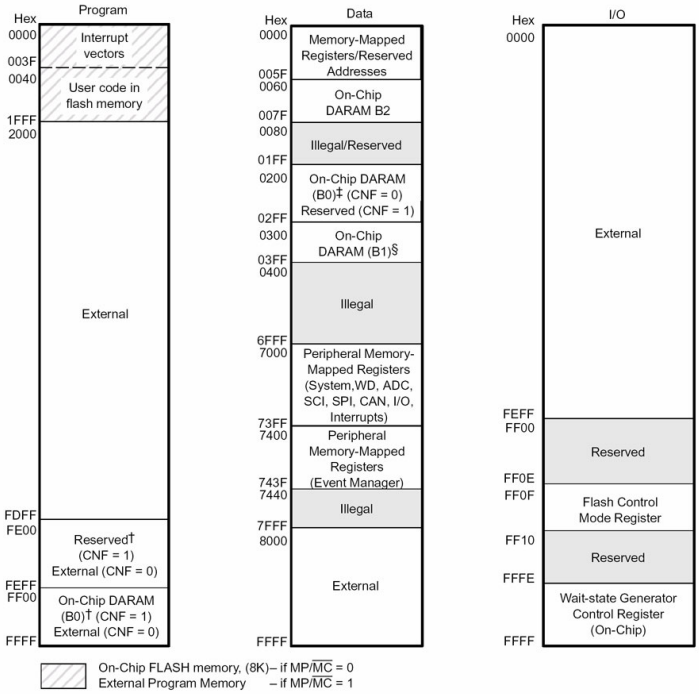


Figure 6.5. Memory architecture of the TMS320F243 DSP

6.1.4.3. Interrupts and peripherals

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 is required to arbitrate the interrupt requests from all the different sources. In this way 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 as well, which makes possible to service several events through interrupts. This provides a very good monitoring feature of the device, which is crucial for motion control, because in case of fault events or reference changes the controller have to react instantly.

6.1.4.4. 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 also provided.

6.1.4.5. 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.

- *Timers*: The EVM includes 2 timers which can be used as independent time bases in applications These timers are 16-bit up- and up/down-counters with programmable period and a variable counting speed.
- *PWM generation module*: This integrated hardware element is capable to provide symmetric/asymmetric sinusoidal and space vector based PWM generation by the use of a timer as the time base of the modulation. Through the SV-PWM, the designed algorithm provided voltage control signal can be easily realized. The exact way of this process is showed later.

- *Deadband unit*: The dead band unit is very important to provide the dead time for the triggering signals to achieve faultless operation of the IGBTs. It is implemented with three 4-bit down counting timers that knocking out the active triggering at the transitions in each PWM pair.
- *Capture units*: Capture units enable time 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.
- *Quadrature Encoder Pulse (QEP) circuit*: The F243 EVM also offers a different way of processing the signals of a position encoder by using one of these inputs as the clock of a timer. The phase difference of the other signal with the clock source signal is used as the counting direction of the timer.
- 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.

6.1.4.6. 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 capable to function as an intelligent system.

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. 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.

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. The SPI is normally used for communications between the DSP controller and external peripherals or another controllers; therefore it has a master-slave type of protocol.

Controller are network (CAN): Today, the CAN has a growing importance as the protocol of distributed control system in the industry. 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. 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.

6.1.4.7. 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 an 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, or it generates a system reset in case of an incorrect resetting sequence.

6.1.4.8. The 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 the implemented

digital control software in its program space. For the considered speed sensorless problem it is perfectly suitable and even the identification process of the motor can be directly accomplished, by the help of its capture units to measure the rotor speed, and its ADC, to obtain the signals of the current sensors. For every aspect, it can be called to 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 such as the designed control structure.

6.1.5. The PC

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

6.1.5.1. 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.

6.1.5.2. 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. The software of the CCS 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 6.6*), have had beneficial effects on the process of development and testing.

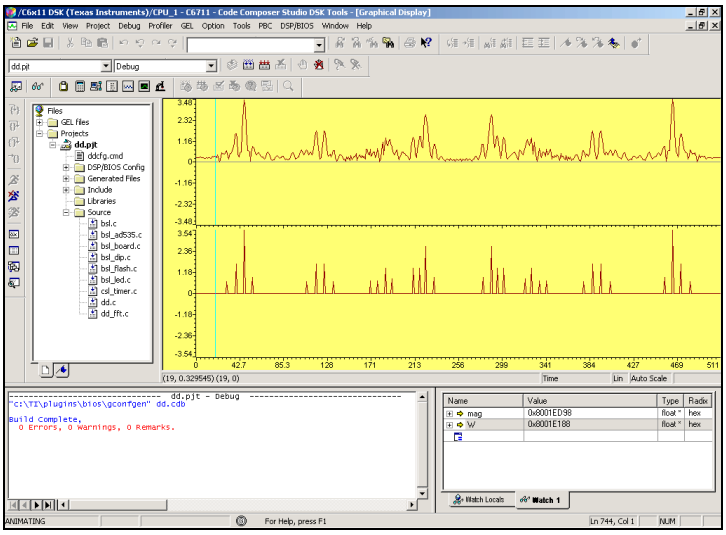


Figure 6.6. Built-in graphing tool of the CCS

CCS also supports RTDX (Real-Time Data Exchange) between the host and the target, interactive profiling, and advanced graphical signal analysis while its 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.

6.1.5.3. 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 implemented controller and in case of faulty operation it notifies the operator about the malfunction. It also provides the estimated speed of the rotor as a back up abut the state of the drive.

6.2. 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. Because the model of the IM is highly nonlinear and because 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.

6.2.1. 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 [11]:

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 $\Xi^N = \left\{ \left(\mathbf{y}(k), \mathbf{u}(k) \right) \middle| k = 1, \ldots, N \right\}$.
- A predictive model in the form of

$$\hat{\mathbf{y}}\left(k \middle| \theta\right) = \mathbf{g}\left(k, \Xi^{k-1}, \theta\right), \tag{6.1}$$

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

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

$$\varepsilon(k|\theta) = \mathbf{y}(k) - \hat{\mathbf{y}}\left(k \middle| \theta\right). \tag{6.2}$$

- A loss function

$$V_N\left(\theta,\Xi^N\right)=\frac{1}{N}\sum_{k=1}^{N}\left\|\varepsilon(k|\theta)\right\|_q, \tag{6.3}$$

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

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

$$\left\|\varepsilon\right\|_2=\frac{1}{2}\sum_{i=1}^{\dim(\varepsilon)}\varepsilon_i^2 \tag{6.4}$$

is used.

If these exist, than such a parameter identification method can be searched for, that produces such a $\hat{\theta}_{method}\left(\Xi^N\right)$ estimated parameter vector form the measured values, to which (6.5) holds:

$$\hat{\theta}_{method}\left(\Xi^N\right)=\arg\left(\min_{\theta}V_N\left(\theta,\Xi^N\right)\right), \tag{6.5}$$

so the loss function is minimal at the estimated parameter values. If the previously mentioned Euclidian norm is used to define (6.3) than this equation will became:

$$V_N\left(\theta,\Xi^N\right)=\frac{1}{N}\sum_{k=1}^{N}\left\|\mathbf{y}(k)-\mathbf{g}(k,\Xi^{k-1},\theta)\right\|_2, \tag{6.6}$$

and then

$$\hat{\theta}_{method}\left(\Xi^N\right)=\arg\left(\min_{\theta}\frac{1}{N}\sum_{k=1}^{N}\left\|\mathbf{y}(k)-\mathbf{g}(k,\Xi^{k-1},\theta)\right\|_2\right). \tag{6.7}$$

Even if (6.7) 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 $\vec{\mathbf{G}}(.)$ gradient vector of the $\mathbf{f}(.)$ function. The gradient vector of $\mathbf{f}(.)$ in $\mathbf{x}$ is defined as follows:

$$\vec{\mathbf{G}}(\mathbf{x})=\left[\left.\frac{\partial\mathbf{f}(\mathbf{x},t)}{\partial x_1}\right|_{\mathbf{x}},\ldots,\left.\frac{\partial\mathbf{f}(\mathbf{x},t)}{\partial x_m}\right|_{\mathbf{x}}\right]^T,\quad \mathbf{x}=\left[x_1,\ldots x_m\right]^T. \tag{6.8}$$

For vector functions, $\vec{\mathbf{G}}(.)$ is also referred as the *Jacobi matrix*. Moreover, while $\vec{\mathbf{G}}(.)$ shows the tangential, the (6.9) defined $\vec{\mathbf{G}}_2(.)$ gives the curvature, so the convex or concave property of the **f**(.) function in **x**.

$$\left[\vec{\mathbf{G}}_2(\mathbf{x})\right]_{ij} = \frac{\partial^2 \mathbf{f}(\mathbf{x},t)}{\partial \mathbf{x}_i \partial \mathbf{x}_j}, \qquad i,j \in \{1,\ldots,m\}, \tag{6.9}$$

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

$$\vec{\mathbf{G}}(\mathbf{x}^*) = \mathbf{0}, \text{ and } \det\left(\vec{\mathbf{G}}_2(\mathbf{x}^*)\right) > 0 \ \left(\text{positive definit}\right) \tag{6.10}$$

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 form 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. $\vec{\mathbf{G}}(.)$ 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 $\left\|\vec{\mathbf{G}}(\mathbf{x})\right\|_2 < \varepsilon$, than 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 (6.11) and with $i = i+1$ the algorithm continues from *Step 2*.

$$\mathbf{x}_{i+1} = \mathbf{x}_i - \vec{\mathbf{G}}(\mathbf{x}_i) \cdot \delta \tag{6.11}$$

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:

$$\frac{d}{dt}\begin{bmatrix} \Psi^s_{r\alpha}(t) \\ \Psi^s_{r\beta}(t) \\ i^s_{s\alpha}(t) \\ i^s_{s\beta}(t) \end{bmatrix} = \mathbf{A}\left(\theta, R_s, L_s, \omega\right)\cdot\begin{bmatrix} \Psi^s_{r\alpha}(t) \\ \Psi^s_{r\beta}(t) \\ i^s_{s\alpha}(t) \\ i^s_{s\beta}(t) \end{bmatrix} + \mathbf{B}\left(\theta, L_s\right)\cdot\begin{bmatrix} u^k_{sd}(t) \\ u^k_{sq}(t) \end{bmatrix} \qquad (6.12)$$

where

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

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

6.2.2. 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.

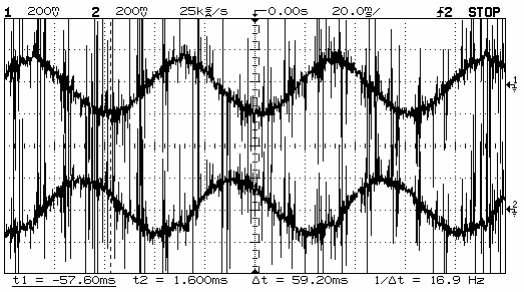


Figure 6.7(a). Measured i_{sa}^s and i_{sb}^s in case of f_0 = 16,9Hz and V_p = 100V

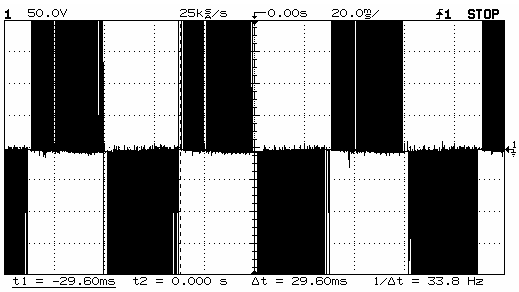


Figure 6.7(b) Measured u_{ab}^s in case of f_0 = 16,9Hz and V_p = 100V

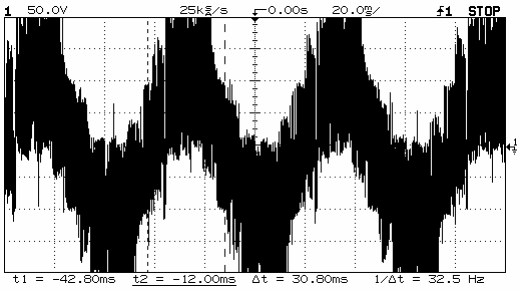


Figure 6.7(c). Measured u_a^s in case of f_0 = 16,9Hz and V_p = 100V

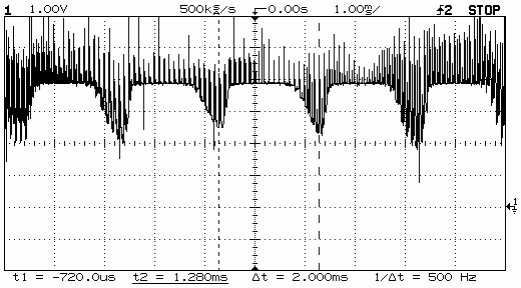


Figure 6.7(d) Measured ω in case of f_0 = 16,9Hz and V_p = 100V

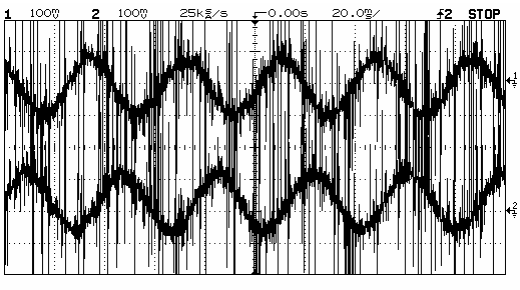


Figure 6.7 (e). Measured i_{sa}^s and i_{sb}^s in case of f_0 = 25Hz and V_p = 150V

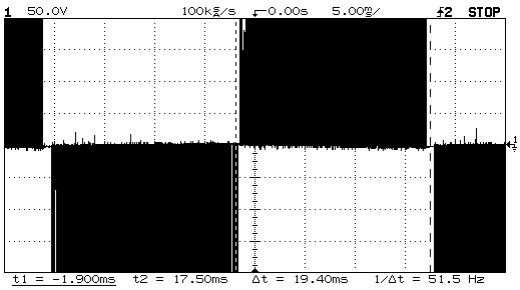


Figure 6.7(f) Measured u_{ab}^s in case of f_0 = 25Hz and V_p = 150V

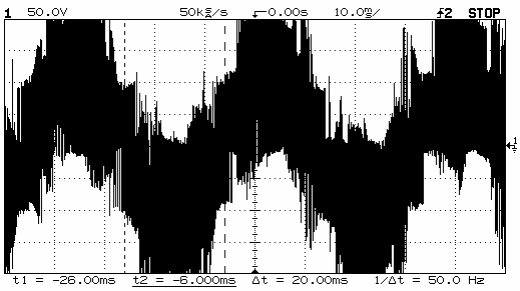


Figure 6.7(g). Measured u_a^s in case of f_0 = 25Hz and V_p = 150V

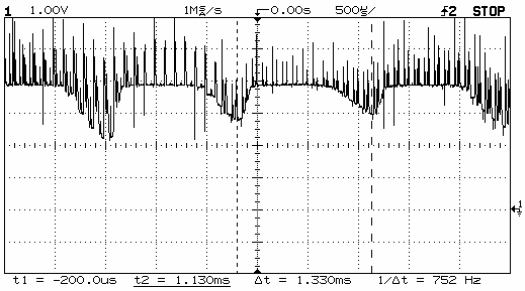


Figure 6.7(h) Measured ω in case of f_0 = 25Hz and V_p = 150V

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.

6.2.3. 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.

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

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.

6.2.4. 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 $f_0 = 16{,}9$Hz with $V_p = 150$V was applied to the IM. By this validation process, the following result in *Figures 6.8(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.

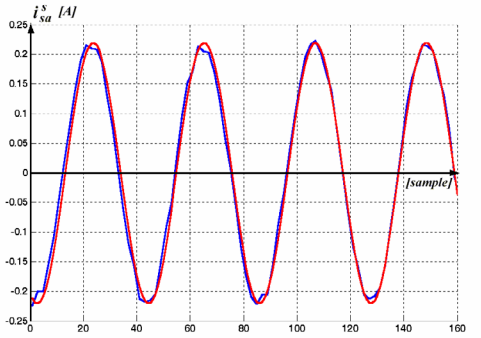


Figure 6.8(a). The real (blue) and simulated current responses of the considered motor

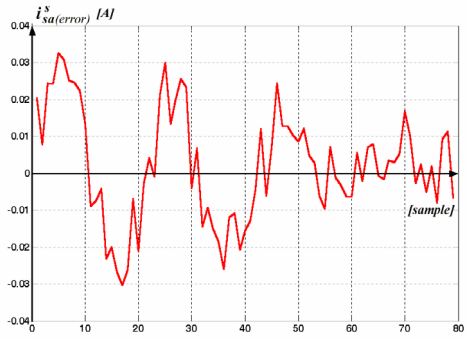


Figure 6.8(b). Relative error in current responses

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.

6.3. The constructed program

The previously mentioned controller structure was applied to the identified motor parameters and the redesigned structure gave similar results that were seen in *Section 5*. To realize this theoretical method several considerations were already given, with the brief description of the target drive. The last step that remained is 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.

6.3.1. Flow chart of the program

The logical structure of the produced program given in *Figure 6.9*.

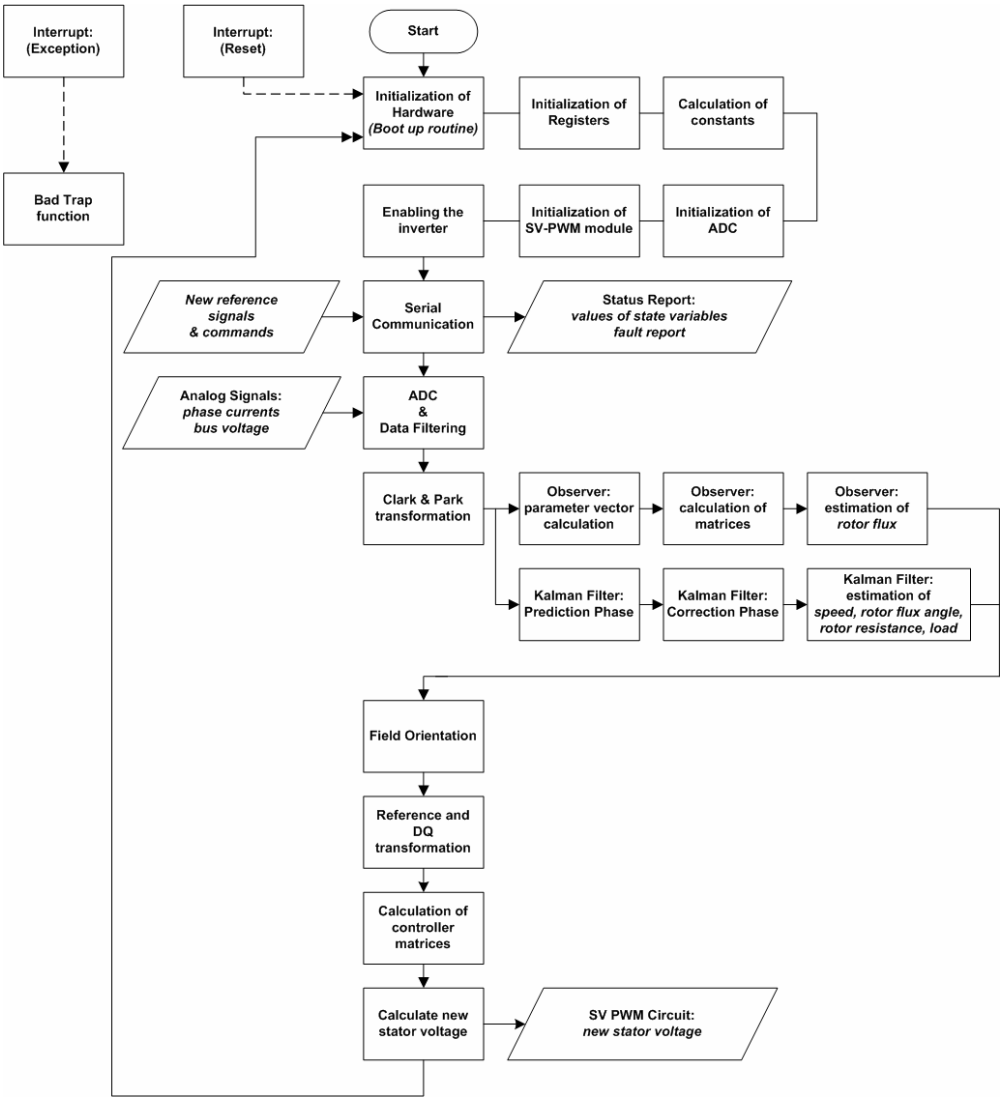


Figure 6.9. The flow chart of the implemented DSP program

This logical structure is totally identical to the structure of the already presented theoretical controller, except 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 way, that it guarantees the fastest running by saving of calculation time and using the special features of ANSI C.

To clarify the presented structure, the functioning of the program is explained in the following parts.

6.3.2. The hardware boot up

The first program block which is written in Assembly 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 which address is defined as the service routine of the *reset* triggered interrupt in the vector table, therefore at any case of reset event, the program starts again from this point. Therefore, the drive is capable of continuous functioning, with restart handling ability.

Then it initializes the whole hardware by clearing the processor flags and pending interrupts, resetting the watchdog, and calibrating the hardware to be able to functioning as a microcontroller. All interrupts are also disabled to the duration of this initialization process. After this, 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.

6.3.3. 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 an abort function is called which hangs up the system. For similar reasons, if such a interrupt is acknowledged which is masked, than the service routines of these interrupts are routed to a 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. At first, all the key registers are initialized to zero and in this way all output activity of the EVM is disabled. Then the main constants of the controller are obtained. As it could have been seen in the pervious sections, the controller is working with several constants, like the huge matrices of the H_∞ controller, observer, etc., and of course there are other constant parameters which are needed to be obtained before the real calculation begins in order to increase the performance. Therefore to fulfill this task and to also ease further modification of the code, these constants are defined in the first part of the program. Unfortunately, those constant which are calculated by 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 are not going to contain any unnecessary operation. In case of the gain scheduled polytopic matrices, the (5.5) and (5.6) discretistions were relaxed to first order, which made possible to do the discretisations before the calculation of the actual controller and observer matrices. In this way, by scaling each of the hardwired values of matrices in the program with the step size and by adding 1 to the diagonal of the state matrices, the computation of the discretisation process could be saved.

The program continues with the initialization of the ADC. The device is calibrated to a 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 completed. The PWM module is calibrated to 2μsec deadband and 10Mhz modulation frequency based SV-PWM mode with positive vector rotation. Finally, the program is ready for continuous operation as a controller so the inverter is enabled by the control logic needed specific 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.

6.3.4. Control loop

The control loop of the program runs continuously, and because all the initialization was already completed by the pervious phases, it just repeats the same instructions in each of its cycle period. Because the analog signals were already converted in the initialization phase, the loop begins with the conversion of the measured current signals to the corresponding complex vector by the help of the Clark transformation. After this, the EKF is calculated. 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 the program proceeds with the estimation process by calculation of the H_∞ observer, which is directly followed by the DQ and reference transformation. These methods are exactly implemented in the same way mentioned in *Section 4.4 and 4.5*. This is followed by the calculation of the H_∞ controller and after this task is completed, 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 process which realizes the SV-PWM generation. This is followed by moving a light on the LED row, which produces a running light spot that informs about correct functioning.

6.3.5. Process of ADC

As it was mentioned the ADC is completed as paired conversions of the measured signals. The corresponding procedure first checks that the conversion was completed successfully or not, then it obtains the results from the ADC

hardware module. The procedure can be called in two modes, if the first mode is selected than the read values are going to be assigned to i_{sa} and i_{sb} and the conversion of the i_{sc} and U_{bus} are started. In case of the other mode, the values are assigned to i_{sc} and U_{bus} with a conversion started on the other two current signals. 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. After the conversions, based on the previously mentioned offset and gain conditioning of the inverter interface card 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 does 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 $\lambda_{filter} = 0.6$, which Bode diagram is given in *Figure 6.10*.

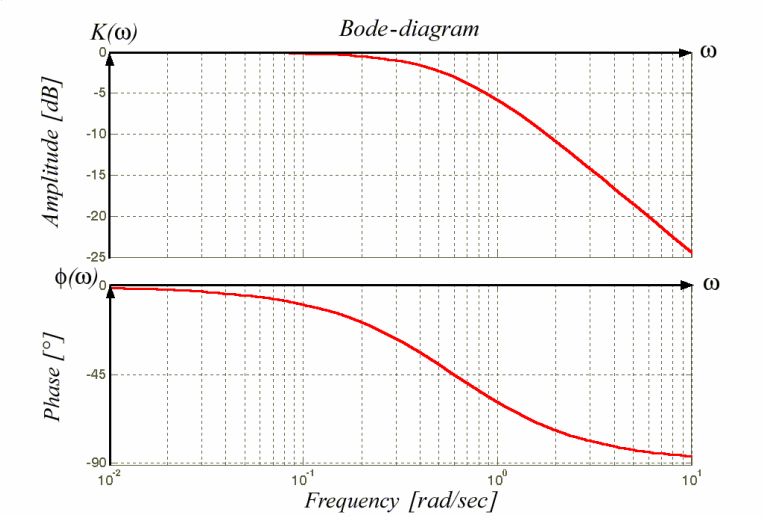


Figure 6.10. Bode-diagram of the software implemented filter of the ADC

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.

6.3.6. 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 **K** matrix is computed which needs matrix inversion, which is calculated through the Gauss-Jordan elimination [12]. 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_∞ 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.

6.3.7. Process of control

The controller calculation is done in a very similar manner to the calculation of the observer. The polytopic coordinates of ω, ω_{rflux}, 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.

6.3.8. Process of PWM generation

The previously calculated and then stator oriented real and imaginary part of the $\mathbf{u}_s^{fr}$ is received by PWM generator 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.

As a physical approach, the SV-PWM generation is very closely related to the concepts of *direct torque control*. It generates the minimum possible harmonic distribution in voltage and currents for the IM [37], by making a brilliant utilization of the eight possible combinations of the realizable voltages in the PWM-VSIs, which can be obtained by the on and off switching states of IGBT bridge presented in *Figure 6.11*.

The derived eight combinations of the switching states and the corresponding motor line-to-line and phase voltages in terms of U_{bus} inverter bus voltage are shown in *Table 6.1*, where 0 = off, 1 = on.

Mapping these voltages on the d-q complex vector plane, by performing the (4.11) Park transformation, results in the six non-zero vectors and two zero vectors (see *Figure 6.11*).

The objective of the SV-PWM modulation is to approximate the control voltage represented by the $\mathbf{u}_s^s$ space vector with a combination of these eight switching patterns of the six IGBTs. One way to achieve this, requires that for any

small period of the PWM period time T_{PWM}, the average inverter output be the same as the average reference voltage as shown in (6.13).

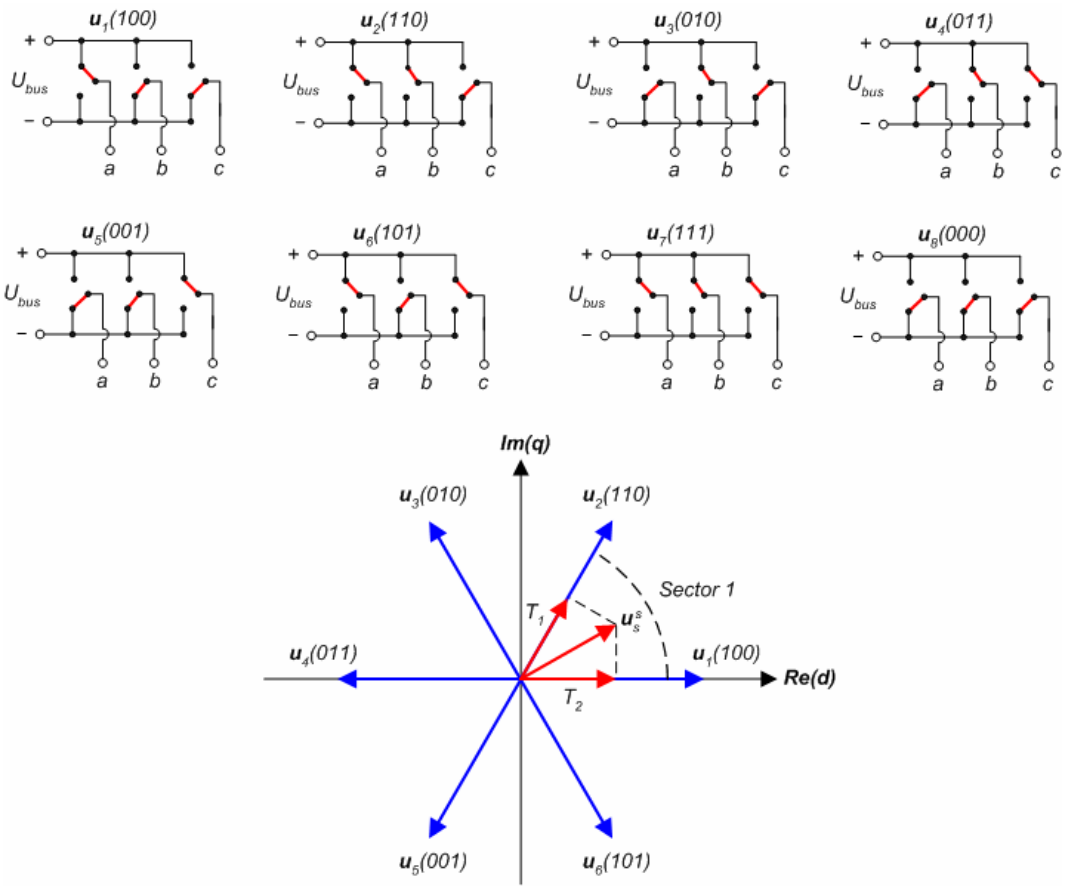


Figure 6.11. Eight possible switching states of PWM-VSIs with their corresponding stator voltage vector

a	b	c	$U_a(U_{bus})$	$U_b(U_{bus})$	$U_c(U_{bus})$	$U_{ab}(U_{bus})$	$U_{bc}(U_{bus})$	$U_{bd}(U_{bus})$
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 6.1. Possible switching states of PWM-VSI inverters

$$\frac{1}{T_{PWM}}\int_{n\cdot T_{PWM}}^{(n+1)\cdot T_{PWM}} u_{out}(t) = \frac{1}{T_{PWM}}\cdot\left(T_1\cdot u_x + T_2\cdot u_{x\pm 60}\right), \qquad n\in\mathbb{N} \tag{6.13}$$

where u_x and $u_{x\pm 60}$ corresponds to the voltage levels of the delimiter $\mathbf{u}_x$ and $\mathbf{u}_{x\pm 60}$ vectors of the considered sector and u_{out} to the realized voltage in this phase. 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. These states are the basic space vectors of the sector containing $\mathbf{u}_s^s$. However, if we assume that the change in the command signal is tinny within the T_{PWM}, then equation (6.13) can be approximated with (6.14).

$$u_{out}(t) = \frac{1}{T_{PWM}}\cdot\left(T_1\cdot u_x + T_2\cdot u_{x\pm 60}\right), \qquad T_1 + T_2 \le T_{PWM}. \tag{6.14}$$

Therefore, it is critical that T_{PWM} be small compared to the speed of the change of $\mathbf{u}_s^s$. In practice, this is guaranteed by the commonly used 20 kHz modulation frequency which period time is 50 µsec. Equation (6.14) also yields that for every PWM period u_{out} can be approximated by applying the switching states $\mathbf{u}_x$ and $\mathbf{u}_{x\pm 60}$ for T_1 and T_2 time durations respectively. Since $T_1 + T_2 \le T_{PWM}$, the inverter needs to be in the zero sates $\mathbf{u}_7$ or $\mathbf{u}_8$ for the rest of the period, therefore for the complete space description:

$$T_{PWM}\cdot\mathbf{u}_{in} = T_1\cdot\mathbf{u}_x + T_2\cdot\mathbf{u}_{x\pm 60} + T_0\cdot\mathbf{u}_{7\text{ or }8}, \qquad T_1 + T_2 + T_0 = T_{PWM}, \tag{6.15}$$

applies, which guarantees that $\mathbf{u}_s^s(t) \approx \mathbf{u}_{out}(t)$. From (6.15) the corresponding times can be computed by vector inversion like in (6.16) or by trigonometric bases like in (6.18) if the angle between $\mathbf{u}_s^s$ and $\mathbf{u}_x$ is α.

$$\begin{bmatrix} T_1 & T_2 \end{bmatrix}^T = T_{PWM}\cdot\begin{bmatrix} \mathbf{u}_x & \mathbf{u}_{x\pm 60} \end{bmatrix}^{-1}\times\mathbf{u}_{in} \tag{6.16}$$

$$T_1 = \sqrt{2}\cdot T_{PWM}\cdot\left\|\mathbf{u}_{in}\right\|\cdot\cos\left(\alpha + 30^\circ\right) \tag{6.17}$$

$$T_2 = \sqrt{2}\cdot T_{PWM}\cdot\left\|\mathbf{u}_{in}\right\|\cdot\cos\left(\alpha\right) \tag{6.18}$$

Depending on the specific application, the SW-PWM can be done either with (6.16) or (6.17) and (6.18). Method (6.16) 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 is also provides a more efficient use of the supply voltage in comparison with the other techniques and it makes possible a maximal $U_{bus}/\sqrt{2}$ amplitude for u_{out} with a maximum of $U_{bus}/\sqrt{6}$ for the rms phase voltage 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.

In the PWM hardware of TMS320F243 this method is realized as well. Therefore, for the functioning of this device the calculation of T_1 and T_2 is needed which was solved in the implemented C code by the vector based method mentioned previously. This method also needs the calculation of the correct sector, which is obtained from the

$$\cos\alpha_u(t)=\frac{u^s_{s\alpha}(t)}{\sqrt{\left(u^s_{s\alpha}(t)\right)^2+\left(u^s_{s\beta}(t)\right)^2}},\qquad \sin\alpha_u(t)=\frac{u^s_{s\beta}(t)}{\sqrt{\left(u^s_{s\alpha}(t)\right)^2+\left(u^s_{s\beta}(t)\right)^2}} \tag{6.19}$$

values by the following logical process seen in *Figure 6.12*.

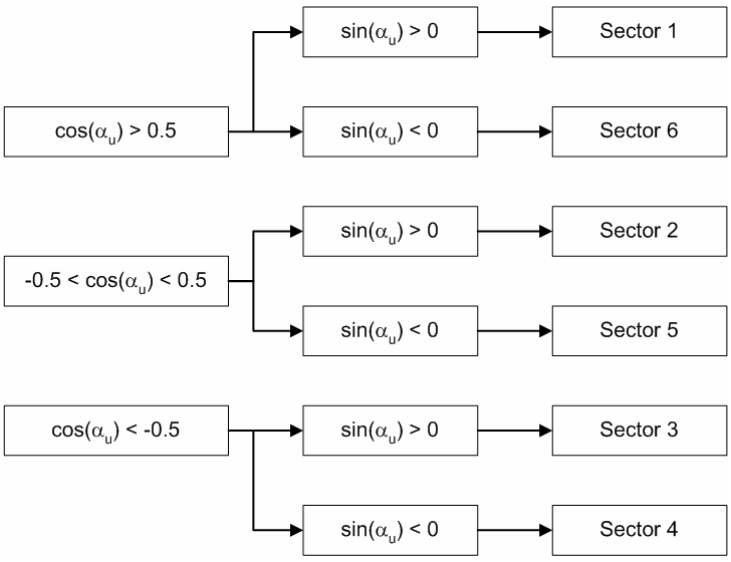


Figure 6.12. Process of sector calculation for SV-PWM

With the help of the sector code and the T_1 and T_2 values the SV-PWM module can be correctly programmed to produce all of the 6 triggering signals of the IGBT bridge. However it can happen that the amplitude of $\mathbf{u}_s^s$ is more than the $U_{bus}/\sqrt{2}$ maximal possible voltage magnitude of SV-PWM method. In this case the implemented program scales back the magnitude of the voltage vector to $U_{bus}/\sqrt{2}$, therefore it also ensures the limiting of the input voltage.

6.3.9. Additional features

The produced program is also able to communicate with the PC trough 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.

6.4. Speed considerations

After completing the implementation and by obtaining maximally optimized code with the careful choose 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 6.13*.

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

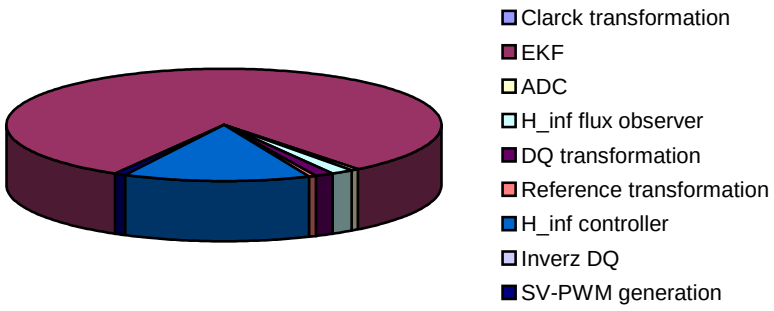


Figure 6.13. Time need of the implemented tasks in the DSP program

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 overthrown 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, by scaling of 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.

6.5. Comparison of results

If the designed control structure is compared to the similarly LPV based controllers given in recently published papers, than it can be concluded, that our system holds the possibility of more sophisticated and efficient control of IM drives than the existing references from the scientific literature. By investigating some of these publications the following comparisons can be drawn:

The solution for the IM drive control problem given in paper [27], presents the design of a H_∞ controller, which uses speed sensors to operate the motor. In this structure the flux is obtained through an LPV H_∞ observer, and moreover the load torque is assumed to be known. By examination of the given results it can be said, that our designed controller structure is not only able to achieve faster reference tracking, because of the absence of the integrator term, but its noise attenuation ability is also better. Moreover, our structure does not need the measurement of the rotor speed.

By comparing our system to the LPV based *sliding mode* controller presented in paper [3], it can be concluded that this system, which makes possible the control of the IM without the measurement of the load torque, gives similar results in tracking. Although, by the relative tracking error of the speed reference signal our controller might seems to be worse than this solution, but the reference tracking of our system is much rapid. Therefore, with a more accurate load torque estimation our controller might be more efficient than the solution given in [3].

If paper [6] is considered, than it can be said, that our designed control structure gives twice better result in the relative error of the reference tracking than the given H_∞ controller based solution of this paper.

Furthermore, in contrast with the above mentioned systems, the great advantage of our controller is that it can be easily tuned and calibrated, which can be efficiently used if the system is supplemented by a parameter identification algorithm as well. In this way, the controller could completely adapt to the parameter uncertainties of the drive, if during the startup it calibrates itself to the system at hand by the help of this parameter identification module.

It is also important to mention that because very noise polluted environment was considered during the design phase, the use of less sophisticated sensors and inverter is made possible for the implementation. Today these devices cost much more than the price of the any DSP on which the actual control could run on.

Unfortunately running results of the controller could not be compared to other implemented solutions, because the hardware turned out to be inadequate if the algorithm is realized in C. However, the ongoing implementation of this system based on Assembly holds the possibility to achieve measurement results which could give a complete proof of the efficiency of the designed system.

7. Conclusion

In this thesis, the physical considerations based design 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 and 4/4 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, the problem at hand was solved by the recently appeared, but very modern and well worked out H_∞ theory and by the help of the EKF. It was also shown, that the direct application of these theories would not give satisfying result because of some practical problems, therefore these theories were modified to fit to our control aims. In this way, such tuning parameters and computational methods were integrated in them which guaranteed a efficient solution of IM drive control.

The designed control structure, which was synthetised by the help of these theories, 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 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.

Based on this, it can be concluded that the solution of the control problem at hand was completed efficiently with high results, even if they are compared to the recent scientific publications on this field. Moreover, this solution was obtained by preferring practical concepts over mathematical considerations, which gave the possibility to realize this structure on a TMS320F243 EVM controlled drive.

In the last section, the implementation of the designed algorithm was explained by outlining the given features of the self assembled Labdrive system. By 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, however 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, if the whole code would be implemented in Assembly with variable scaling. 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 synthetisation 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 a sensorless control method that can be used in wide variety of industrial applications.

References

[1] 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.

[2] Åström K. J., and B. Wittenmark: Computer Controlled Systems, *Prentice-Hall*, New York: 1990.

[3] 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.

[4] Chapman S. J.: Electric Machinery Fundamentals, *WCB/McGraw-Hill*, New York: 1999.

[5] Dettori M., and C. W. Scherer: LPV Design for a CD Player: An Experimental Evaluation of Performance, *Delft University of Technology*, The Netherlands (2002).

[6] 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.

[7] Doyle J. C., K. Glover, P. P. Khargonekar, and B. A. Francis: State-Space Solutions to Standard H_2 and H_∞ Control Problems, *IEEE Transactions on Automatic Control*, Vol. 34, No. **8**, (1989), pp. 831-847.

[8] 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.

[9] Glover K., and D. McFarlane: Robust stabilization of normalized coprime factor plant descriptions with H_∞ bounded uncertainty, *IEEE Transactions on Automatic Control*, Vol. 34, No. **8**, (1989), pp. 821-830.

[10] Halász S.: Electrical drives (Villamos Hajtások), in Hungarian, *Műszaki Könyvkiadó*, 1993.

[11] Hangos K., and G. Szederkényi: Parameter Identification of Dynamic Systems, *Veszprém University Press*, Veszprém: 1999.

[12] Hartung F.: Introduction to Numerical Analysis (Bevezetés a Numerikus Analízisbe), in Hungarian, *Veszprém University Press*, Veszprém: 1998.

[13] Hilairet M., C. Darengosse, F. Auger, and P. Chevrel: Sysnthesis and Analysis of Robust Flux Observers for Induction Machines, *Bd. de l'Université*, France (2000).

[14] 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.

[15] Holtz J.: Sensorless Position Control of Induction Motors an Emerging Technology, *IEEE Transactions on Industrial Electronics*, Vol. 45, No. **6**, (1998), pp. 840-852.

[16] Holtz J.: Sensorless Control of Induction Drives, *Proceedings of the IEEE*, Vol. 90, No. **8**, (2002), pp. 1259-1394.

[17] Jamniczky Á.: Operation of Electrical Machines (Villamos Gépek Üzemtana), in Hungarian, *Veszprém University Press*, Veszprém: 1984.

[18] 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.

[19] Kálmán R. E.: A New Approach to Linear Filtering and Prediction Problem, *Journal Basic Engineering*, Vol. 82, (1960), pp. 34-45

[20] Kovács P.K.: Transient Phenomena in Electrical Machines, *Akadémiai kiadó*, Budapest: 1954.

[21] Labdrive Interface Module, *Texas Instruments*, Technical Reference Guide: 2000.

[22] Labdrive Inverter Module, *Texas Instruments*, Technical Reference Guide: 2000.

[23] 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.

[24] Leith D. J., and W. E. Leithead: Counter-Example to a Common LPV Gain-Scheduling Design Approach, *University of Strathclyde*, U.K. (2000).

[25] Ogata K.: State Space Analysis of Control Systems, *Prentice Hall Inc., Englewood Cliff, NJ.*, 1967.

[26] Packard A.: Gain-Scheduling via Linear Fractional Transformations, *System Control Letters*, Vol. 22, (1994), pp. 79-92.

[27] Premapain E., I. Pstletwaite, and A. Benchaib: A Linear Parameter Variant H_∞ Control Design for an Induction Motor, *Control Engineering Practice*, No. **10**, (2002), pp. 663-644.

[28] Rajashekara K., A. Kawamura, and K. Matsuse: Sensorless Control of AC Motor Drives, *IEEE Press Book*, 1996.

[29] Rácz I.: Automated Electrical Drives (Automatizált Villamos Hajtások), in Hungarian, *Tankönyvkiadó kiadó*, 1976.

[30] Rugh W. J. and D. J. Stilwell: Stability and L_2 Gain Properties of LPV Systems, *Automatica*, Vol. 38, (2002), pp. 1601-1606.

[31] Skogestad S., and I. Postletwaite: Multivariable Feedback Control, *John Wiley & Sons*, Chichester: 1996.

[32] TMS320F243/F241/C242 DSP Controllers Reference Guide, *Texas Instruments*, Technical Reference Guide: 2000. (*http://focus.ti.com/lit/ug/spru276c/spru276c.pdf*)

[33] 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.

[34] Uray V.: Electrotechnics (Elektrotechnika), in Hungarian, *Műszaki könyvkiadó*, Budapest: 1970.

[35] Vas P.: Sensorless Vector and Direct Torque Control, *Oxford University Press*, Oxford: 1998.

[36] 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.

[37] 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*)

[38] Zhou K., and J. C. Doyle: Essentials of Robust Control, *Prentice-Hall*, New-York: 1998.

Appendix

Appendix A

(Script code of the initialization routine: (DHinf.m))

```
% Definition of  the Motor Constants %

LR=130.0e-3;     % Strator inductivity %
LS=130.0e-3;     % Rotor inductivity %
LM=120.0e-3;     % Mutual inductivity %
RS=1.86;         % Stator resistance %
RRmin=3;         % Minimal rotor resistance %
RR=3;            % Rotor resistance %
p=3;             % Number of pole-pairs %
J=0.02;          % Inertia moment %
F=0.0001;        % Fraction %
KK=3.5;          % Convection %
c=0.21;          % Specific heat of the Aluminum [kcal/(kg*Celsius)] %
m=4;             % Wight of the Rotor Conductors [kg] %
Tmin=18;         % Minimal Rotor Temperature %

% Parameter ranges for politope calculation %

RRmin=3;             % Rotor resistance minimum %
RRmax=2*RRmin;       % Rotor resistance maximum %
Wmin=-85;            % Rotor speed minimum %
Wmax=100;            % Rotor speed maximum %
WFmin=-40;           % Flux speed minimum %
WFmax=50;            % Flux speed maximum %

range=[Wmin,Wmax;RRmin,RRmax;WFmin,WFmax];

% Definition of the Filters for Mixed Sensitivity Structure of the Controller %
wu1=0.8;
wu2=0.8;
we1=ltisys('tf',10,[1 10]);
we2=ltisys('tf',10,[1 10]);

% Calculation of Constants %
TSIG=(LS*LR-LM^2)/LR;
TM=-F*p/J;
TM2=3*p*p*LM/(2*J*LR);
L1=LM/LR;
L2=(L1)^2;
NN0=(0.86*RRmin)/((245+Tmin)*c*m);
RK=L2*RR+RS;

% Definition of Parameter Dependencies for the H∞ Controller %

% Parameter Dependency 0 (Constant) %
A0=[0,0,0;0,-RS/TSIG,0;0,0,-RS/TSIG];
B0=[0,0;1/TSIG,0;0,1/TSIG];
C0=[0,1,0;0,0,1];
D0=zeros(2);
E0=eye(3);
```

```
% Parameter Dependency 1 (Rotor Speed) %
A1=[0,0,0;0,0,0;-L1/TSIG,0,0];
B1=zeros(3,2);
C1=zeros(2,3);
D1=zeros(2);
E1=zeros(3);

% Parameter Dependency 2 (Rotor Resistance ) %
A2=[-1/LR,L1,0;LM/(LR^2*TSIG),-L2/TSIG,0;0,0,-L2/TSIG];
B2=zeros(3,2);
C2=zeros(2,3);
D2=zeros(2);
E2=zeros(3);

% Parameter Dependency 3 (Rotor Flux Speed ) %
A3=[0,0,0;0,0,1;0,-1,0];
B3=zeros(3,2);
C3=zeros(2,3);
D3=zeros(2);
E3=zeros(3);

% Definition of the Polytopic Form of the LPV Control Problem %
pv=pvec('box',range);
S0=ltisys(A0,B0,C0,D0,E0);
S1=ltisys(A1,B1,C1,D1,E1);
S2=ltisys(A2,B2,C2,D2,E2);
S3=ltisys(A3,B3,C3,D3,E3);
pdG=psys(pv, [S0,S1,S2,S3]);
pdDK=ltisys(zeros(2),eye(2),eye(2),zeros(2),eye(2));
pdP=aff2pol(pdG);

% Design of the Controller %
pdPP=sconnect('r(2)','e=r-G0;G0', 'K:e', 'G0:K', pdP);
pdPP=smult(pdPP,sdiag(we1,we2,wu1,wu2,eye(2)));
[gopt,pdK]=hinfgs(pdPP,[2 2]);


% Controller Formulation %
psinfo(pdK);
psinfo(pdK,'par');
for i=1:8
[AA(:,:,i), b1, BB(:,:,i), c1, CC(:,:,i), d11, d12, d21, d22]=hinfpar(psinfo(pdK,'sys',i), [2, 2]);
end

% Parameter Ranges for Observer Synthetisation %
range=[Wmin,Wmax;RRmin,RRmax];

% Definition of the Filters for Mixed Sensitivity Structure of the Observer %
we1=ltisys('tf',0.01,[1 0.01]);
we2=ltisys('tf',0.01,[1 0.01]);

% Definition of Parameter Dependencies for the H∞ Observer %

% Parameter Dependency 0 (Constant) %
A0=zeros(2);
B0=zeros(2);
C0=eye(2);
```

```
D0=zeros(2);
E0=eye(2);

% Parameter Dependency 1 (Rotor Speed) %
A1=[0,-1;1,0];
B1=zeros(2);
C1=zeros(2);
D1=zeros(2);
E1=zeros(2);

% Parameter Dependency 2 (Rotor Resistance ) %
A2=[-1/LR,0;0,-1/LR];
B2=[L1,0;0,L1];
C2=zeros(2);
D2=zeros(2);
E2=zeros(2);

% Definition of the Polytopic Form of the LPV Estimation Problem %
pv=pvec('box',range);
S0=ltisys(A0,B0,C0,D0,E0);
S1=ltisys(A1,B1,C1,D1,E1);
S2=ltisys(A2,B2,C2,D2,E2);
pdG=psys(pv, [S0,S1,S2]);
pdP=aff2pol(pdG);
pdP2=sdiag(pdP,eye(2));

% Design of the Observer %
pdPP=sconnect('d(2)','G0([1,2])-G0([3,4])', 'K: d','G0: d;K', pdP2);
[gopt,pdK]=hinfgs(pdPP,[2 2]);

% Observer  Formulation %
psinfo(pdK);
psinfo(pdK,'par');

for i=1:4
[FA(:,:,i), b1, FB(:,:,i), c1, FC(:,:,i), d11, d12, d21, d22]=hinfpar(psinfo(pdK,'sys',i), [2, 2]);
en
```

Appendix B

(Script code of the controller S-function: (DHinfControlSystem.m)

```
function [sys,x0,str,ts] = DHinfControlSystem(t,x,u,flag,AA,BB,CC,Wmax,Wmin,
                           RRmax,RRmin,WFmax,WFmin)
global h v;
switch flag,

% Initialization Phase, (setting up main system variables) %
case 0,
  sizes = simsizes;
  sizes.NumContStates  = 0;        % Number of Continuous States %
  sizes.NumDiscStates  = 5;        % Number of Discrete States %
  sizes.NumOutputs     = 2;        % Number of Outputs %
  sizes.NumInputs      = 5;        % Number of Inputs (1=i_s_d_error, 2= i_s_q_error, 3=omega,
                                                    4=RR, 5=flux_omega)
  sizes.DirFeedthrough = 0;        % Sate of Direct Feed Trough %
  sizes.NumSampleTimes = 1;        % Number of Desired Sample Times %
  sys = simsizes(sizes);
  x0  = zeros(5,1);                % Initialization of States %
  str = [];
  ts  = [0.0001 0];

   % Definition of Variables %
   h=0.0001;
   v=zeros(8,1);

% Calculation of the Controller States in Discrete Time %
case 2,

   % Calculation of Polytopic Coordinates %
   u(3)=(u(3)-Wmin)/(Wmax-Wmin);
   u(4)=(u(4)-RRmin)/(RRmax-RRmin);
   u(5)=(u(5)-WFmin)/(WFmax-WFmin);
   v=1;
   for i=3:5
     v=[v*(1-u(i)) v*u(i)];
   end

   % Calculation of Controller Matrices %
   WA=zeros(5);
   WB=zeros(5,2);
   for i=1:8
      WA=WA+AA(:,:,i)*v(i);
      WB=WB+BB(:,:,i)*v(i);
   end

   % Discretisation Process %
   WB=WB*h+WA*WB*(0.5*h^2);
   WA=eye(5)+WA*h+WA*WA*(0.5*h^2);
   y=(WA*x+WB*u(1:2));

  % Bound Limiting of States %
  for i=1:5
      if y(i)>500
         y(i)=500;
```

```
        end
        if y(i)<-500
            y(i)=-500;
        end
    end

    % Updating S-Function States %
    sys=y;

% Output Generation Phase %
case 3,

    % Calculation of Output Matrices %
    WC=zeros(2,5);
    for i=1:8
      WC=WC+CC(:,:,i)*v(i);
    end

    % Calculation of the Outputs %
    sys=WC*x;

otherwise
  sys=[];
end
```

Appendix C

(Script code of the I/O reference transformation routine: (RefCompIOSystem.m))

```
function [sys,x0,str,ts] = RefCompIOsystem(t,x,u,flag,LM,LR,TM2,F,p,KompTun)
global NB;
switch flag,

% Initialization Phase, (setting up main system variables) %
case 0,
  sizes = simsizes;
  sizes.NumContStates  = 0;          % Number of Continuous States %
  sizes.NumDiscStates  = 0;          % Number of Discrete States %
  sizes.NumOutputs     = 2;          % Number of Outputs %
  sizes.NumInputs      = 6;          % Number of Inputs %
  sizes.DirFeedthrough = 1;          % Sate of Direct Feed Trough %
  sizes.NumSampleTimes = 1;          % Number of Desired Sample Times %
  sys = simsizes(sizes);
  x0  = [];                          % Initialization of States %
  str = [];
  ts  = [0 0];
  NB=2*LR/(3*p*LM);

% Calculation of the Reference Transformation in Discrete Time %
case 3,
   y=zeros(2,1);

   % Process Transformation for i_s_d with Tuning Parameters %
   y(1)=KompTun*u(1)/LM+LR/(u(5)*LM)*u(2);

   % Checking Flux_r_d if It Needs Correction %
   WW=NB*(F*u(3)+u(6))+u(4)/TM2;
   WW2=u(1);
   if abs(WW2)<0.001
      WW2=1;
   end

   % Process Transformation for i_s_q with Tuning Parameters %
   y(2)=WW/WW2;
   sys=[y];

otherwise
  sys=[];
end
```

Appendix D

(Script code of the DQ transformation routine: (DQTrafoSystem.m))

```
function [sys,x0,str,ts] = DQTrafoSystem(t,x,u,flag,LM,LR)
switch flag,

% Initialization Phase, (setting up main system variables) %
case 0,
  sizes = simsizes;
  sizes.NumContStates  = 0;          % Number of Continuous States %
  sizes.NumDiscStates  = 0;          % Number of Discrete States %
  sizes.NumOutputs     = 6;          % Number of Outputs %
  sizes.NumInputs      = 6;          % Number of Inputs %
  sizes.DirFeedthrough = 1;          % Sate of Direct Feed Trough %
  sizes.NumSampleTimes = 1;          % Number of Desired Sample Times %
  sys = simsizes(sizes);
  x0  = [];                          % Initialization of States %
  str = [];
  ts  = [0 0];

% Calculation of the DQ Transformation and Flux Speed in Discrete  Time %
case 3,

   y=zeros(6,1);

   % Calculation of Flux_r_d %
   W=sqrt(u(3)^2+u(4)^2);
   y(1)=W;

   % Correction of Flux_r_d  for the Calculation Methods%
   if W<0.001
      W=0.01;
   end

   % COS & SIN Computation of the Rotor Flux Angle %

   CW=u(3)/W;
   if CW==0
      CW=1;
   end
   y(5)=CW;
   SW=u(4)/W;
   y(6)=SW;

   % DQ Transformation of the Stator Current %
   IW=sqrt(u(1)^2+u(2)^2);
   if IW<0.001
      IW=0.01;
   end
   CIW=u(1)/IW;
   SIW=u(2)/IW;
   K1=CW*CIW;
   K2=SW*SIW;
   y(2)=(K1+K2)*IW;
   K1=CW*SIW;
   K2=SW*CIW;
```

```
    y(3)=(K1-K2)*IW;

    % Computation of the Flux Speed %
    y(4)=LM*(y(3)/W)/LR*u(6)+u(5);

    sys=[y];
otherwise
   sys=[];

end
```

Appendix E

(Script code of the EKF: (KalmanFSystemZZ.m))

```
function [sys,x0,str,ts] = KalmanFSystemZZ(t,x,u,flag)
global LR LM LS RS RR0 p J F KK NN0 TSIG TM TM2 L1 L2 h Q R P C D MM;

% Initialization Phase, (setting up main system variables) %
switch flag,
case 0,
  sizes = simsizes;
  sizes.NumContStates  = 0;        % Number of Continuous States %
  sizes.NumDiscStates  = 6;        % Number of Discrete States %
  sizes.NumOutputs     = 4;        % Number of Outputs %
  sizes.NumInputs      = 7;        % Number of Inputs (1=u_s_alpha, 2=u_s_beta, 3=M_load,
                                       4=flux_r_alpha 5=flux_r_beta 6=i_s_alpha 7=i_s_beta) %
  sizes.DirFeedthrough = 0;        % Sate of Direct Feed Trough %
  sizes.NumSampleTimes = 1;        % Number of Desired Sample Times %
  sys = simsizes(sizes);
  x0  = zeros(6,1);                % Initialization of States %
  str = [];
  ts  = [0.0001 0];

  % Setting Up of Constants and Declaring Variables %
  x0(6)=RR0;
  X_pred = zeros(6,1);
  C=[1,0,0,0,0,0;0,1,0,0,0,0;0,0,1,0,0,0;0,0,0,1,0,0;];
  D=[0,0,0,0,1,0;0,0,0,0,0,1;0,0,1,0,0,0;0,0,0,1,0,0];
  P=ones(6)*1e-4;
  Q=zeros(6);
  Q(3,3)=0.0117*h/TSIG;
  Q(4,4)=0.0117*h/TSIG;
  R=zeros(4);
  R(1,1)=0.0205;
  R(2,2)=0.0205;
  R(3,3)=13.85;
  R(4,4)=13.85;
  MM=zeros(2,6);

% Calculation of the EKF States in Discrete Time %
case 2,

   % Measured Reference Outputs of the System %
   Y= u(4:7);

   % Measured Inputs of the System %
   U=u(1:3);

   %Prediction of  States with the Adams-Basforth Method %

   HM=-x(6)/LR*x(1)-x(5)*x(2)+L1*x(6)*x(3);
   X_pred(1)=x(1)+h/12*(5*MM(2,1)-16*MM(1,1)+23*HM);
   MM(2,1)=MM(1,1);
   MM(1,1)=HM;

   HM=-x(6)/LR*x(2)+x(5)*x(1)+L1*x(6)*x(4);
   X_pred(2)=x(2)+h/12*(5*MM(2,2)-16*MM(1,2)+23*HM);
```

```
MM(2,2)=MM(1,2);
MM(1,2)=HM;

HM=-(L2*x(6)+RS)/TSIG*x(3)+L1*x(6)/(LR*TSIG)*x(1)+x(5)*L1/TSIG*x(2)+1/TSIG*u(1);
X_pred(3)=x(3)+h/12*(5*MM(2,3)-16*MM(1,3)+23*HM);
MM(2,3)=MM(1,3);
MM(1,3)=HM;

HM=-(L2*x(6)+RS)/TSIG*x(4)+L1*x(6)/(LR*TSIG)*x(2)-x(5)*L1/TSIG*x(1)+1/TSIG*u(2);
X_pred(4)=x(4)+h/12*(5*MM(2,4)-16*MM(1,4)+23*HM);
MM(2,4)=MM(1,4);
MM(1,4)=HM;

HM=-F*p/J*x(5)+TM2*(x(1)*x(4)-x(2)*x(3))-p/J*u(3);
X_pred(5)=x(5)+h/12*(5*MM(2,5)-16*MM(1,5)+23*HM);
MM(2,5)=MM(1,5);
MM(1,5)=HM;

IEFF=((x(1)/LR-L1*x(3))^2+(x(2)/LR-L1*x(4))^2);
HM=NN0*x(6)*IEFF-KK*x(6)+KK*RR0;
X_pred(6)=x(6)+h/12*(5*MM(2,6)-16*MM(1,6)+23*HM);
MM(2,6)=MM(1,6);
MM(1,6)=HM;

X_pred=X_pred';

% Calculation of the Partial Derivatives %

DF=zeros(6);

DF(1,1)=1-h*x(6)/LR;
DF(1,2)=-h*x(5);
DF(1,3)=h*L1*x(6);
DF(1,4)=0;
DF(1,5)=-h*x(2);
DF(1,6)=-h/LR*x(1)+h*L1*x(3);

DF(2,1)=-DF(1,2);
DF(2,2)=DF(1,1);
DF(2,3)=0;
DF(2,4)=DF(1,3);
DF(2,5)=h*x(1);
DF(2,6)=-h/LR*x(2)+h*L1*x(4);

DF(3,1)=h*L1*x(6)/(LR*TSIG);
DF(3,2)=h*x(5)*L1/TSIG;
DF(3,3)=(1-h*(L2*x(6)+RS)/TSIG);
DF(3,4)=0;
DF(3,5)=h*L1/TSIG*x(2);
DF(3,6)=-h/TSIG*L2*x(3)+h*L1/(LR*TSIG)*x(1);

DF(4,1)=-DF(3,2);
DF(4,2)=DF(3,1);
DF(4,3)=0;
DF(4,4)=DF(3,3);
DF(4,5)=-h*L1/TSIG*x(1);
DF(4,6)=-h/TSIG*L2*x(4)+h*L1/(LR*TSIG)*x(2);
```

```
    DF(5,1)=h*TM2*x(4);
    DF(5,2)=-h*TM2*x(3);
    DF(5,3)=-h*TM2*x(2);
    DF(5,4)=h*TM2*x(1);
    DF(5,5)=1+h*TM;
    DF(5,6)=0;

    DF(6,1)=h*NN0*x(6)*2*(x(1)/LR-L1*x(3))/LR;
    DF(6,2)=h*NN0*x(6)*2*(x(2)/LR-L1*x(4))/LR;
    DF(6,3)=-DF(6,1)*LM;
    DF(6,4)=-DF(6,2)*LM;
    DF(6,5)=0;
    DF(6,6)=h*NN0*IEFF+(1-h*KK);

    % Calculation of the Predicted Covariance %
    P_pred=DF*P*DF'+Q;

    % Statistical Correction of the Predicted Values %
    % (This step is not performed in noiseless case) %
    K=P_pred*C'*inv(C*P_pred*C'+R);
    y=X_pred+K*(Y-C*X_pred);
    if y(6)<RR0
       y(6)=RR0;
    end
    P=P_pred-K*C*P_pred;

    % Updating S-Function States %
    sys=y;

% Output Generation Phase %
case 3,

       % Calculation of the Outputs %
       sys=D*x;

otherwise
  sys=[];
end
```

Appendix F

(Script code of the H_∞ observer: (DHinfObserverSystem.m))

```
unction [sys,x0,str,ts] = DHinfObserverSystem(t,x,u,flag,FA,FB,FC,Wmax,Wmin,RRmax,RRmin)
global hh v2;

% Initialization Phase, (setting up main system variables) %
switch flag,
case 0,
  sizes = simsizes;
  sizes.NumContStates  = 0;          % Number of Continuous States %
  sizes.NumDiscStates  = 2;          % Number of Discrete States %
  sizes.NumOutputs     = 2;          % Number of Outputs %
  sizes.NumInputs      = 4;          % Number of Inputs  (1=i_s_alpha, 2=i_s_beta, 3=omega,
                                                          4=RR) %
  sizes.DirFeedthrough = 0;          % Sate of Direct Feed Trough %
  sizes.NumSampleTimes = 1;          % Number of Desired Sample Times %
  sys = simsizes(sizes);
  x0  = zeros(2,1);                  % Initialization of States %
  str = [];
  ts  = [0.0001 0];
   hh=0.0001;
   v2=zeros(4,1);

 % Calculation of the Observer States in Discrete Time %
case 2,

   % Calculation of Polytopic Coordinates %
   u(3)=(u(3)-Wmin)/(Wmax-Wmin);
   u(4)=(u(4)-RRmin)/(RRmax-RRmin);
   v2=1;
   for i=3:4
     v2=[v2*(1-u(i)) v2*u(i)];
   end

   % Calculation of Observer Matrices %
   WA=zeros(2);
   WB=zeros(2);
   for i=1:4
      WA=WA+FA(:,:,i)*v2(i);
      WB=WB+FB(:,:,i)*v2(i);
   end

   % Discretisation Process %
   WB=WB*hh+WA*WB*(0.5*hh^2);
   WA=eye(2)+WA*hh+WA*WA*(0.5*hh^2);

   % Updating S-Function States %
   sys=(WA*x+WB*u(1:2));

% Output Generation Phase %
case 3,

   % Calculation of Output Matrices %
   WC=zeros(2);
   for i=1:4
```

```
      WC=WC+FC(:,:,i)*v2(i);
    end

    % Calculation of the Outputs %
    sys=WC*x;

otherwise
   sys=[];
end
```