NTNU

Norwegian University of
Science and Technology

Sensorless Control of Permanent Magnet Synchronous Machines

Derivation and Implementation

Eirik Haustveit

Master of Science in Electric Power Engineering
Submission date: June 2016
Supervisor: Lars Einar Norum, ELKRAFT
Co-supervisor: Anirudh Budnar Acharya, ELKRAFT

Norwegian University of Science and Technology
Department of Electric Power Engineering

Preface

This project was carried out at the Norwegian University of Science and Technology, Department of Electric Power Engineering during the spring semester of 2016.

It is a continuation of the specialization project "Laboratory platform for PM machine drive study - Adaptive Torque Estimation" that was carried out during the fall semester of 2015.

I hereby declare that this thesis titled "Sensorless Control of Permanent Magnet Synchronous Machines" and the work presented in it is my own.

Trondheim, 2016-06-09

Eirik Haustveit

Acknowledgment

I would like to thank my supervisor prof. Lars Einar Norum for suggesting this project, and for his help and guidance. Furthermore I wold like to thank my second supervisor Anirudh Budnar Acharya for his guidance and assistance during the development of the power electronics and controller algorithms. Also I wold like to thank my fellow student Yngve B. Solbakken for sharing his experience from the application of motor drives in industry.

I wold like to thank prof. Roy Nilsen for providing me with some insight into flux estimators, and their application. And I wold like to thank Bård Almås and Aksel Andreas Reitan Hanssen at the service laboratory, for helping me preparing the laboratory setup.

I would like to thank Lars Manger Ekroll at Bergen University College for providing me with a permanent magnet motor, and I would like to thank Associate Professor Trond Toftevaag for bringing the motor to Trondheim.

I wold like to thank Brage Iversen for suggesting the line receiver for the encoder, and I wold like to thank Roland Peterer and Christian Schmid for their help and cooperation during the development of the controller.

Finally i would like to thank my fellow students at NTNU for all the good times.

E.H.

Abstract

This master project is a further investigation of the topics covered in the specialization project “Laboratory platform for PM machine drive study - Adaptive Torque Estimation”.

Permanent magnet synchronous machines have seen an increased popularity in recent years. Some market analyses indicate that the industry is experiencing a gradual shift from induction machines to permanent magnet machines. The machine offer several advantages over the inductions machine. The rotor losses are minimal, as there is no excitation current in the rotor. Thus the overall efficiency is improved. The machine offers higher torque density, lower weight, and higher torque to inertia ratio, which may improve the dynamic performance. Although the permanent magnet materials are expensive, the low overall material usage suggests lower production costs.

In order to obtain the aforementioned advantages a power electronic converter is required to control the machine. Although many of the induction machines found in industry are supplied by direct grid connection, the cost savings associated with the increased efficiency of applying a motor drive is changing this trend. Thus the requirement of a motor drive should no longer be considered a disadvantage of the synchronous machine alone.

A lot of research effort has been put into developing control algorithms for electrical machines. In many cases the induction machine has been the initial study case, and later on the algorithms have been adapted to suit different machines.

In order to accurately evaluate the performance of various motor control algorithms, a laboratory platform of a permanent magnet synchronous motor drive has been designed in this project. Several practical aspect regarding cooling requirements, safety functions, gate driver design, controller and measurement interface have been considered. The platform is also intended as a base for future research and development.

Three different control algorithms, *field oriented control*, *direct torque control*, and *model predictive control* have been evaluated. Additionally different modulation techniques for the inverter switches are covered. Theoretical derivations and practical implementation is considered. The operation is evaluated by simulation and laboratory experiments.

Regardless of which motor control algorithm is selected, knowledge of the rotor flux position is required for efficient closed loop control. The use of position sensors on the motor shaft increases the cost of the drive, and reduces the reliability.

In order to address this issue, several position estimation algorithms for sensorless control have been evaluated.

Samandrag

Denne masteroppgåva er ei vidareføring av spesialiseringsprosjektet "Laboratory platform for PM machine drive study - Adaptive Torque Estimation".

Permanentmagnetiserte synkronmaskiner har auka i popularitet dei siste åra. Marknadsanalysar tydar på at industrien opplever eit paradigmeskifte når det gjeld val av motorteknologi. Der asynkronmaskina tidlegare var det opplagte valet, ser ein stadig at fordelane med permanentmagnetmaskina har gjort den meir attraktiv.

Permanentmagnetmaskina har fleire fordelar når ein samanliknar med asynkronmaskina. Det er ikkje behov for magnetiseringsstraum i rotoren. Sjølv om ein framleis har moderate kvervelstraumar, medfører dette ei betydeleg reduksjon av rotortapa. Dette medfører igjen ei stor forbetring i den totale verknadsgraden. Maskina har høgare momenttettleik, lavare vekt og høgare forhold mellom moment og tregleiksmoment. Sjølv om permanentmagnetar er kostbare, er det mykje som tydar på at dei totale produksjonskostnadane vert lavare, grunna lavare materialforbruk.

For å ha nytte av dei gode eigenskapane til maskina er ein avhengig av frekvensomformardrift. Sjølv om det framleis er vanleg at asynkronmaskina vert kopla direkte til nettspenning, er dette noko som er i ferd med å endrast. Grunna auka energieffektivitet vil installasjon av frekvensomformar ofte medføra ei kraftig kostnadsbesparing uavhengig av vald motorteknologi.

Det er gjort mykje forskning på utvikling av kontrollalgoritmar for elektriske maskiner, i mange tilfeller med asynkronmaskina som utgangspunkt. Seinare tilpassingar har gjort algoritmane tilgjengeleg for andre maskintypar.

For å oppnå eit nøyaktig og realistisk samanlikningsgrunnlag for dei forskjellige algoritmane, må ein ty til laboratorieforsøk. Eit laboratorieoppsett med omsyn på ei rekkje praktiske forhold er utarbeidd. Spesielt med tanke på kjøling, tryggleiksfunksjonar, og grensesnitt mellom styringssytemet og dei fysiske komponentane.

Utvikling og evaluering av tre ulike syringsalgoritmar er gjennomført: Feltorientert styring, direkte momentstyring og modellpredikert styring. I tillegg er det tatt omsyn til forskjellige metodar for modulering av utspenninga frå omformaren.

Uavhengig av kva styringsalgoritme ein nyttar, er ein avhengig av kjennskap til rotorens posisjon. Dette har ein tradisjonelt oppnådd ved å måla den fysiske rotorposisjonen. Bruken av posisjonssensorar aukar imidlertid kostnadane, samtidig som det er ei potensiell feilkjelde i systemet. For å bøta på dette problemet har ein i mange år forsøkt å utvikla algoritmar for estimering av posisjonen basert på

elektriske målingar. Fleire forskjellige system er i bruk i industrien, men det er framleis høve til å forbetra algoritmane, spesielt i situasjonar med lågt turtal.

Dette prosjektet evaluerer ytelsen til tre forskjellige estimeringsalgoritmar ved hjelp av simulering og laboratorieforsøk.

...implementation of an integrator for motor flux estimation is no easy task.

— Jun Hu (Excerpt from "New Integration Algorithms for Estimating Motor Flux over a Wide Speed Range", September 1998)

Contents

List of Figures

List of Tables

Listings

Part I

Theory

Chapter 1

The permanent magnet synchronous machine

The PMSM is a electric machine with permanent magnets mounted on the rotor, and an approximate sinusoidal flux distribution due to the three phase windings in the stator. It has many similarities with the Brushless DC (BLDC) motor, however the windings in the BLDC provides a trapezoidal distribution of the flux.

The PMSM offers many advantages over the alternative motor technologies [32]. There is no need for excitation currents in the rotor, which increases the efficiency when compared to the induction machine. Additionally the power density is increased. Due to the reduced amount of iron , the production costs are also decreased when compared to induction machines [45].

Due to the many advantages it is expected that the PMSM will increase in popularity, and possibly replace the induction machine as the default choice in industrial applications [45].

The PMSM is typically divided into two subcategories, the Surface permanent magnet synchronus machine (SPMSM), and the IPMSM, depending on how the magnets are located in the rotor.

The most important difference between these two machines is that the IPMSM has saliency. The inductance in the d-axis is lower than the inductance in the q-axis, i.e. $L_d < L_q$. This is caused by the fact that the permanent magnets has approximately the same reluctance as air. When the magnets are buried, some part of the rotor will exhibit higher reluctance than others. Another difference is that burying the magnets provides a more robust construction, which may support operation at higher speeds [32].

An interesting side note is that the inductance in a separately excited salient synchronous machine has the exact opposite property of the IPMSM. The direct axis inductance is higher because the rotor contains more iron in the position of the field winding, i.e. $L_d > L_q$.

Due to the saliency of the IPMSM, it offers somewhat higher flexibility in the development of algorithms for estimation of rotor position. The property of position depended inductance may be exploited.

A area of active research when it comes to PMSM, is ways to minimize torque ripple. A review of various techniques, including motor design, and drive design, is presented in [33].

1.1 Electrical model

The voltage in the d, and q-axis is expressed in (1.1) and (1.2) respectively [41].

$$v_d = R_d i_d + \frac{\mathrm{d}\lambda_d}{\mathrm{d}t} - \omega_r \lambda_q \tag{1.1}$$

$$v_q = R_q i_q + \frac{\mathrm{d}\lambda_q}{\mathrm{d}t} + \omega_r \lambda_d \tag{1.2}$$

$R_d = R_q = R_s$ is the resistance in the stator winding.

The flux linkages for the SPMSM in d, and q-axis are given by:

$$\lambda_{sd} = L_s i_{sd} + \lambda_{fd} \tag{1.3}$$

$$\lambda_{sq} = L_s i_{sq} \tag{1.4}$$

A simplified per-phase equivalent circuit of the PMSM is provided in figure 1.1, valid for balanced sinusoidal steady state operation [41].

In the figure, the induced back-electromotive force (EMF) is represented by a voltage source. The voltage vector position depend on the rotor position, and this is the basis for the position estimators covered later in the project.

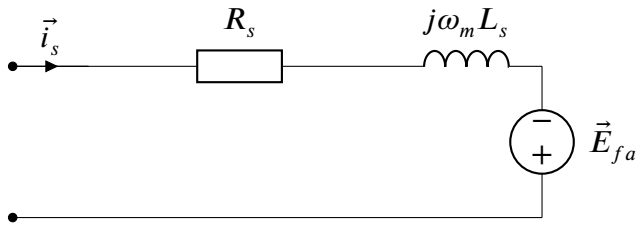


Figure 1.1: Simplified equivalent circuit for the PMSM

State space model

In order to develop a simulation model for the IPMSM it is useful to express the differential equations with the derivative of the current. The expressions for d, and q-axis are given in equation (1.5) and (1.6) respectively.

$$\frac{\mathrm{d}}{\mathrm{d}t} i_d = \frac{1}{L_d} v_d - \frac{R_s}{L_d} i_d + \frac{L_q}{L_d} \omega_e i_q \tag{1.5}$$

$$\frac{\mathrm{d}}{\mathrm{d}t} i_q = \frac{1}{L_q} v_q - \frac{R_s}{L_q} i_q + \frac{L_d}{L_q} \omega_e i_d - \frac{\Psi_f}{L_q} \omega_e \tag{1.6}$$

By integrating both sides of the equation it is possible to obtain the actual motor current.

$$\int \frac{\mathrm{d}}{\mathrm{d}t} i_d = \int \frac{1}{L_d} v_d - \frac{R_s}{L_d} i_d + \frac{L_q}{L_d} \omega_e i_q \tag{1.7}$$

$$\int \frac{\mathrm{d}}{\mathrm{d}t} i_q = \int \frac{1}{L_q} v_q - \frac{R_s}{L_q} i_q + \frac{L_d}{L_q} \omega_e i_d - \frac{\Psi_f}{L_q} \omega_e \tag{1.8}$$

As can be seen the current depends on the angular velocity of the motor. This is obtained by integrating the mechanical differential equation.

$$\frac{\mathrm{d}}{\mathrm{d}t} \omega_m = \frac{1}{J} \left(T_{em} - T_L - B \cdot \omega_m \right) \tag{1.9}$$

1.2 Torque

The resulting electromechanical torque is given by the sum of d, and q-axis torque:

$$T_{em} = T_{d,rotor} + T_{q,rotor} \tag{1.10}$$

$$T_{em} = \frac{p}{2}(\lambda_{sd} i_{sq} - \lambda_{sq} i_{sd}) \tag{1.11}$$

In some literature the factor $3/2$ is added to the electromechanical torque equation:

$$T_{em} = \frac{3}{2}\frac{p}{2}(\lambda_d i_q - \lambda_q i_d) \tag{1.12}$$

This comes from a different variation of the Clarke/Park transformation. The factor $2/3$ is used to normalize the space vector when performing the transformation, and thus it must be compensated when calculating the torque.

By substituting the flux linkages by the inductances and currents in (1.3), and (1.4), we arrive at:

$$T_{em} = \frac{3}{2}\frac{p}{2}(\lambda_f i_q + (L_d - L_q) i_d i_q) \tag{1.13}$$

Where the first element is the torque produced by the interaction between the stator and rotor magnetic field, and the second is the reluctance torque.

The induced back-EMF is given by:

$$E_{fa} = \sqrt{\frac{2}{3}} \lambda_{fd} \omega_m \tag{1.14}$$

Hence the voltage constant is given by:

$$k_e = \sqrt{\frac{2}{3}} \lambda_{fd} \tag{1.15}$$

Chapter 2

Control methods

There are several different algorithms that may be used when designing a drive for a electric machine. The preferred method depends on the performance constraints, the cost, the motor and inverter topology, as well as the skills of the engineer responsible for the design.

There are two popular methods that may be regarded as the classical control methods for motor drives. The FOC, and the DTC. However several other, simpler, and more advanced algorithms are available.

In this project the MPC has been considered as a third alternative control algorithm.

A comparison of FOC and DTC for a PMSM is presented in [18]. The paper presents simulation results comparing the performance, however laboratory experiments have not been utilized to confirm the results.

Regardless of which algorithm is selected, the knowledge of the supplied voltage and currents to the motor are required for reliable control. Additionally the rotor position is required in order to orient the magnetic flux in the most efficient manner. The controller reference is typically current/torque, speed or position, depending on the requirements of the mechanical load. This means that the speed or position information may also be required as feedback to the controllers, in addition to its usage in the orientation of the field.

2.1 Field oriented control

The consept of FOC for induction machines where originally proposed in [4, 3]. The same principles are easily applicable to synchronous machines, by using a different field alignment angle.

In FOC the magnetic field is considered as two orthogonal components that define the stator flux vector. By performing a mathematical transformation known as the Park's transformation [44], the components are synchronized to the position of the rotor.

The goal is typically to align the flux vector of the stator such that it provides maximum torque per ampere. I.e. only the component that lies $90°_e$ with respect to the flux from the permanent magnets is used. If speeds above the motor rated speed are required, a flux that is in phase with the permanent magnets, but with opposite sign is used to weaken the field.

In order to use FOC knowledge of the rotor position is required. I.e. it must either be measured mechanically or estimated based on electrical measurements.

Controller

In FOC the classical Proportional integral (PI) controller is commonly used in several cascaded loops to control the behavior of the motor. Figure 2.1 provides a overview of a typical FOC algorithm. Apart from the position and speed estimator, the implementation is straight forward, and will only be discussed briefly in this report.

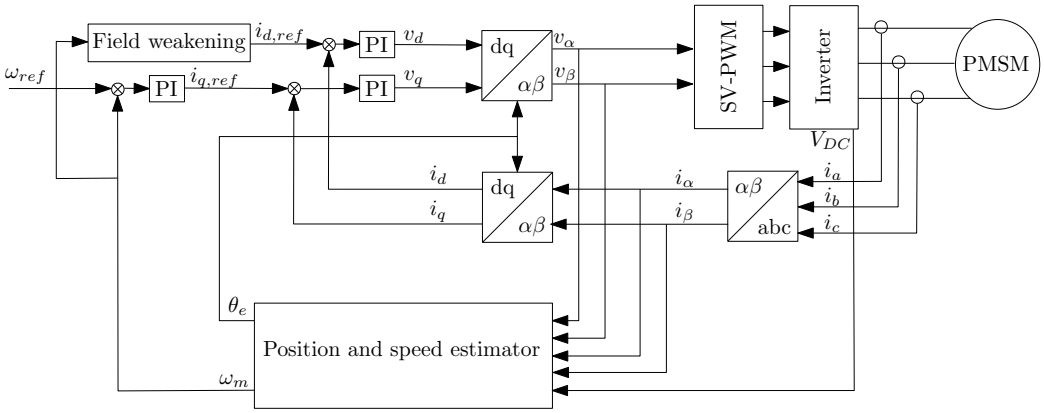


Figure 2.1: Sensorless FOC of a PMSM

The three phase currents are measured[1], and converted to $\alpha\beta$ frame using the Clarke transformation [14]. The $\alpha\beta$-currents are used in conjunction with the $\alpha\beta$-voltages to estimate the rotor flux position. The position is in turn used to transform the $\alpha\beta$-currents to the dq-frame using the Park transformation.

As the dq-frame rotates synchronous with the rotor, the currents will appear as DC quantities in steady state. This enables the use of classical PI controllers to control the currents. As the d-axis is aligned with the magnetic field of the rotor, the q-axis current will be the torque producing component of the current, and it's reference is controlled by a external speed controller.

The d-axis current reference is normally set to zero for a SPMSM, in order to obtain maximum torque per ampere operation. If the speed reference is above the rated motor speed, negative d-axis current is used to decrease the magnetic field, and in turn the induced back-EMF. This is known as field weakening, and the motor torque is decreased in inverse proportion to the speed.

The output from the PI current controllers is the dq-voltage references to the motor. These references are converted to $\alpha\beta$-frame using the inverse Park transformation, and in turn fed to a SV-PWM generator. The SV-PWM computes the switching times, and generate switches signals to the inverter.

The speed controller feedback is the estimated rotor speed as obtained by differentiation of the estimated position.

The reference values for d, and q-axis currents for a SPMSM is given as:

$$i_{d,ref} = 0 \tag{2.1a}$$

$$i_{q,ref} = \frac{T_e}{\Psi_m} \tag{2.1b}$$

Special considerations for IPMSM

In a IPMSM the relationship between q-axis current and torque is non linear. Due to the saliency there will be a given amount of reluctance torque produced by the d-axis current.

From equation 2.2 is can bee seen that a negative d-axis current will have a positive contribution on the torque.

[1] In some cases only two phases are measured, and the third is calculated by the relation $i_a + i_b + i_c = 0$, which is true for symmetrical loads.

$$T_{em} = \frac{3}{2}\frac{p}{2}(\lambda_f i_q - (L_q - L_d) i_d i_q) \tag{2.2}$$

Figure 2.2 shows how the torque characteristics is affected by variation of the current amplitude. The angle providing maximum torque is shifted to the right for increasing current [42].

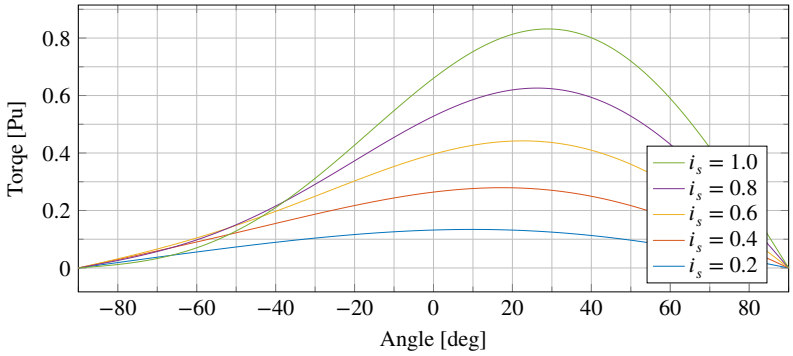


Figure 2.2: Torque as a function of the current vector angle ϕ_p. $x_d = 0.4$, $x_q = 1$ and $\Psi_m = 0.66$.

The optimal d, and q-axis currents are given by equation (2.3), and (2.4) respectively [42].

$$i_d = \frac{\Psi_m}{4(L_q - L_d)} - \sqrt{\left(\frac{\Psi_m}{4(L_q - L_d)}\right)^2 + \frac{i_s^2}{2}} \tag{2.3}$$

$$i_q = \pm\sqrt{i_d^2 - \frac{\Psi_m i_d}{L_q - L_d}} \tag{2.4}$$

Additionally, during field weakening the maximum allowable q-axis current depends on the d-axis current:

$$i_{q,max} = \sqrt{i_s^2 - (i_d + \delta i_d)^2} \tag{2.5}$$

Where:

$$i_s = \sqrt{i_d^2 + i_q^2} \tag{2.6}$$

Feed forward

The state-space model of the PMSM is given in equations (1.5), and (1.6). From these equations it is evident that there is a cross coupling between the d, and q-axis. The exact coupling terms are indicated as follows:

$$\frac{\mathrm{d}}{\mathrm{d}t} i_d = \frac{1}{L_d} v_d - \frac{R_s}{L_d} i_d + \underbrace{\frac{L_q}{L_d} \omega_e i_q}_{\text{coupling}} \tag{2.7}$$

$$\frac{\mathrm{d}}{\mathrm{d}t} i_q = \frac{1}{L_q} v_q - \frac{R_s}{L_q} i_q + \underbrace{\frac{L_d}{L_q} \omega_e i_d}_{\text{coupling}} - \underbrace{\frac{\Psi_m}{L_q} \omega_e}_{\text{coupling}} \tag{2.8}$$

By utilizing feed-forward it is possible to decouple the two equations, which in turn simplifies the control.

Controller tuning

When commissioning a new motor drive it is often considered advantageous to be able to tune the controllers automatically. Analytical formulas has been derived to compute the controller parameters, from the knowledge of the electrical and mechanical gains, and time constants of the system. A in depth analysis of this tuning is beyond the scope of this project, the following section presents the resulting equations for a popular tuning scheme [42].

Modulus optimum

Modulus optimum is commonly applied when tuning the current controller.

$$T_i = T_1 \tag{2.9a}$$

$$K_p = \frac{T_1}{2 K_s T_{sum}} \tag{2.9b}$$

Where T_i is the integration time constant, and K_p is the proportional gain. T_1 is the dominating time constant of the system, and T_{sum} is the sum of all other time constants.

In a discrete implementation the sampling time must also be considered:

$$T_i = T_1 - \frac{T_{samp}}{2} \tag{2.10a}$$

$$K_p = \frac{T_1 - \frac{T_{samp}}{2}}{K_s(2T_{sum} + T_{samp})} \tag{2.10b}$$

Where T_{samp} is the sampling time of the current controller.

Symmetric optimum

Symmetrical optimum is commonly applied when tuning the speed controller.

$$T_i = \beta T_{sum} \tag{2.11a}$$

$$K_p = \frac{T_1}{\sqrt{\beta} K_s T_{sum}} \tag{2.11b}$$

Where $\beta = 4$ is a typical tuning factor, but other factors may be considered.

In a similar manner to modulus optimum the sampling time is considered by the following equations:

$$T_i = \beta \left(T_{sum} + \frac{T_{samp}}{2} \right) \tag{2.12a}$$

$$K_p = \frac{T_1}{K_s(\sqrt{\beta} T_{sum} + T_{samp})} \tag{2.12b}$$

Improved modulation

When utilizing Sinusoidal pulse width modulation (SPWM), it can be shown that the locus of the reference voltage vector lies within a circle of radius $\frac{1}{2} V_{DC}$ [24].

Based on space vector theory, it was shown that it is possible to achieve a better utilization of the DC-link voltage[43]. The proposed modulation method was called SV-PWM.

It has been shown that the locus of the reference voltages for SV-PWM lies within a circle of radius $\frac{1}{\sqrt{3}}V_{DC}$. Which means an approximately 15% higher utilization of the DC-link voltage.

Additionally a method known as THIPWM is available with similar properties.

Voltage vectors

The possible output voltage vectors from a two level three phase inverter, in terms of V_{DC}, are given in table 2.1.

Implementation of SV-PWM

A sampling interval is divided into three sub-intervals:

$$T_s = T_1 + T_2 + T_0 \tag{2.13}$$

Where T_1 and T_2 is the time given to each of the two vectors defining the current working sector. T_0 is the time given to the zero vector.

The average output voltage vector is given as

$$V_{out} = \frac{T_1}{T_s}V_1 + \frac{T_2}{T_s}V_2 \tag{2.14}$$

If the reference is provided in $\alpha\beta$-frame, the magnitude of the reference voltage vector is given by:

$$\left|\vec{V}_{ref}\right| = \frac{2}{3}M_a\sqrt{V_\alpha^2 + V_\beta^2} \tag{2.15}$$

The angle of the reference vector is given by:

$$\theta = \arctan\left(\frac{V_\beta}{V_\alpha}\right) \tag{2.16}$$

Table 2.1: Output voltage vectors from a two level, three phase inverter [26].

Voltage	Switch state			Phase voltages			Line voltages		
Vector	A	B	C	V_{an}	V_{bn}	V_{cn}	V_{ab}	V_{bc}	V_{ca}
V_0	0	0	0	0	0	0	0	0	0
V_1	1	0	0	$\frac{2}{3}$	$-\frac{1}{3}$	$-\frac{1}{3}$	1	0	−1
V_2	1	1	0	$\frac{1}{3}$	$\frac{1}{3}$	$-\frac{2}{3}$	0	1	−1
V_3	0	1	0	$-\frac{1}{3}$	$\frac{2}{3}$	$-\frac{1}{3}$	−1	1	0
V_4	0	1	1	$-\frac{2}{3}$	$\frac{1}{3}$	$\frac{1}{3}$	−1	0	1
V_5	0	0	1	$-\frac{1}{3}$	$-\frac{1}{3}$	$\frac{2}{3}$	0	−1	1
V_6	1	0	1	$\frac{1}{3}$	$-\frac{2}{3}$	$\frac{1}{3}$	1	−1	0
V_7	1	1	1	0	0	0	0	0	0

From the angle, the sector number may be determined by:

$$\text{sector} = \text{floor}\left(\frac{\theta}{\pi/3}\right) + 1 \tag{2.17}$$

Where floor is a function rounding the number down to the nearest integer.

From this information, the dwelling time for each of the voltage vectors defining the current sector is given by:

$$T_1 = \frac{\sqrt{3}T_S\left|\vec{V}_{ref}\right|}{V_{DC}}\left(\sin\left(\frac{\pi}{3} - \theta + \frac{(s-1)\pi}{3}\right)\right) \tag{2.18}$$

$$T_2 = \frac{\sqrt{3}T_S\left|\vec{V}_{ref}\right|}{V_{DC}}\left(\sin\left(\theta - \frac{(s-1)\pi}{3}\right)\right) \tag{2.19}$$

Where s is the sector number, and T_S is the switching period.

Finally the time given to the zero vector is:

$$T_0 = T_S - (T_1 + T_2) \tag{2.20}$$

Simulink model

A Simulink® model of both the SV-PWM and the THIPWM has been implemented. The block diagram of SV-PWM is depicted in figure 2.3 and 2.4. The source code for the MATLAB function is provided in appendix E.4.

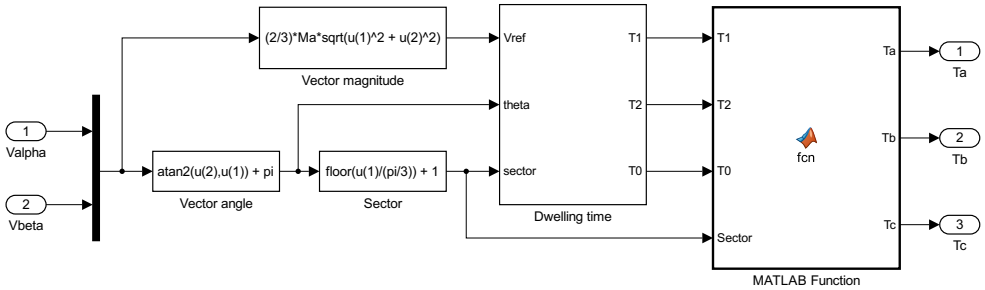


Figure 2.3: Classical SV-PWM controller.

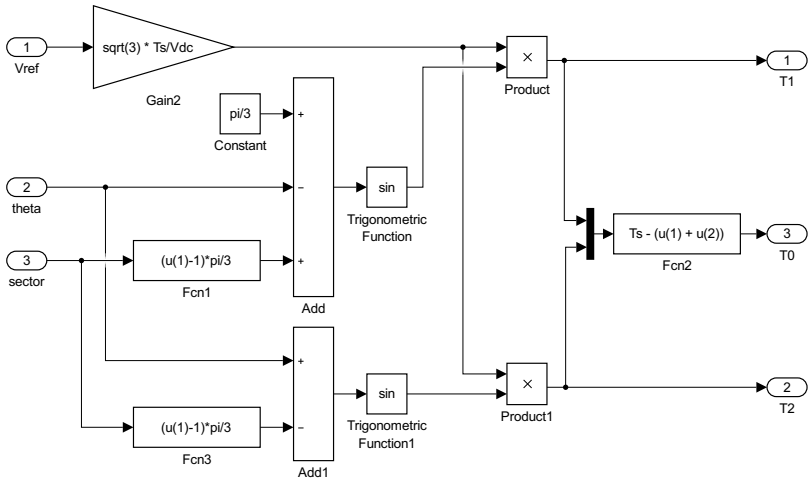


Figure 2.4: Calculation of dwelling time for SV-PWM.

The block diagram of the THIPWM-modulator is depicted in figure 2.5. The amplitude of the third harmonic is $\frac{1}{6}$ of the fundamental.

Figure 2.6 and 2.7 depicts the results from simulating a THIPWM, and a SV-PWM respectively. The input reference signal is a 50 Hz sine wave. It is observed that the different schemes have a similar effect on the modulating wave.

The THIPWM has a simpler implementation in Simulink®, however the computational intensity might be higher in a practical implementation, due to the generation of the sine waves. Additionally the SV-PWM accepts the $\alpha\beta$ voltages as reference, and thus does not require a inverse Park transformation.

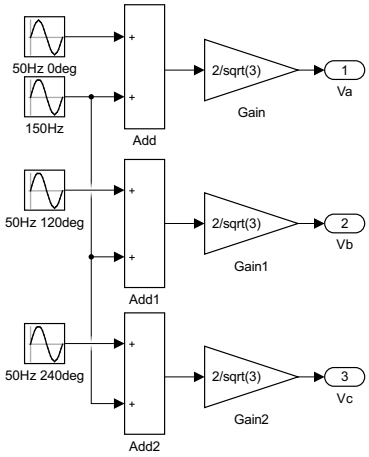


Figure 2.5: Simulink® model of a THIPWM-modulator

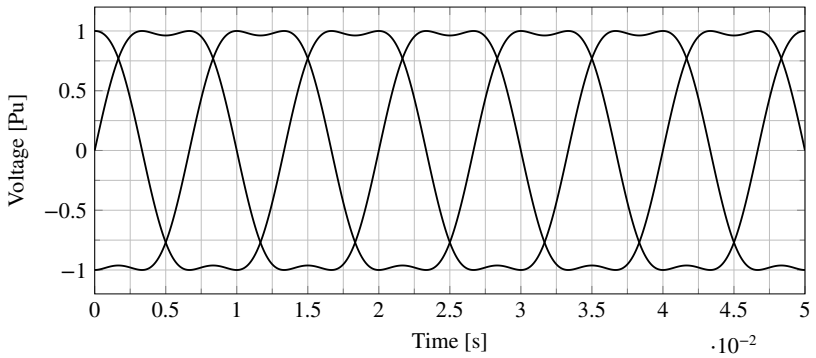


Figure 2.6: Modulating signal for a THIPWM based modulator.

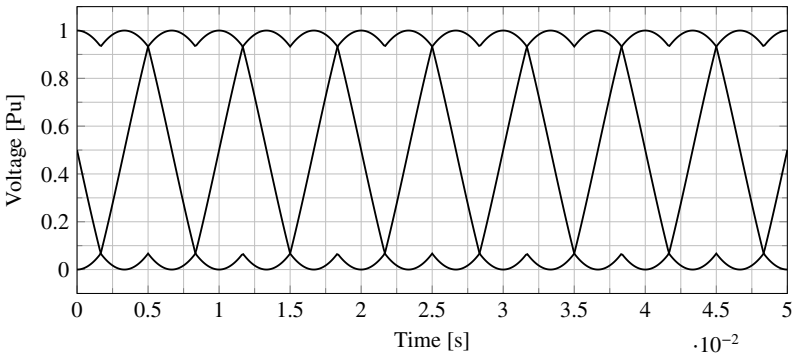


Figure 2.7: Modulating signal for a SV-PWM based modulator.

More details regarding the implementation of SV-PWM are available in [26, 29].

2.2 Direct torque control

Initially the DTC algorithm was proposed in [56] in 1986 and [9] in 1988. In DTC the flux linkage is estimated and used to compute the torque. Then instead of controlling the currents the torque is controlled directly. Originally by the use of hysteresis controllers, and lookup tables to determine the best suited voltage vectors.

DTC for PMSM was introduced in [17], in 1996. A more extensive analysis of DTC for PMSM is available in [39].

In the original DTC proposal[56], the zero voltage vectors where used when no torque change was required. It has however been suggested that the zero voltage vectors should be avoided when a PMSM is considered, and that the stator flux linkage should be kept moving with respect to the rotor flux linkage at all times [66]. The reasoning for this different control philosophy is based on the fact that the permanent magnets rotate with the rotor, the stator flux linkage will change even if zero voltage vectors are applied. Whether or not this makes sense however, is debatable [39, p. 59].

Apart from this difference the most important difference between DTC for induction machines and DTC for PMSM lies in the different ways in which flux and torque is estimated. As the initial rotor flux is zero in a induction machine, a initial voltage vector is applied in order to excite the rotor such that the flux estimator may acquire an initial estimate. For a PMSM a initial voltage vector may be used to align the rotor, such that initial rotor position is determined. Alternatively initial position estimation algorithms are available, in particular for the IPMSM [20].

A important consideration when applying the classical DTC to a PMSM is that the stator inductances are typically lower than in a induction machine. The current will respond more quickly to changes in voltage, which in turn causes more torque ripple. A possible solution to this issue is to increase the sample frequency [18].

There are however several different variations of DTC. The classical approach is to use hysteresis controllers and a lookup table[56]. A different method utilizing space vector modulation, and offering constant switching frequency is also available, originally proposed in [36], and improved in [57, 58]. Another method is the Fuzzy Direct Torque Control (FDTC) [6, 8], where a fuzzy controller replaces both the hysteresis controller and the lookup table.

Operating principle

Figure 2.8 depicts a block diagram of a typical DTC algorithm. The motor flux, torque, speed and position are all estimated, and fed back to the controller. Two hysteresis controllers are utilized for torque and flux control respectively. A lookup table is used to determine the most suitable voltage vector depending on whether the controlled variable should increase or decrease.

A external PI controller is used to compute the torque reference, based on the speed error.

Inside the nominal speed range of the motor, in order to achive maximum torque per ampere operation, the flux reference should equal the permanent magnet flux. If the requested speed-reference exceeds the ratings of the motor, a field weakening controller is used to decrease the flux in inverse proportion to the speed.

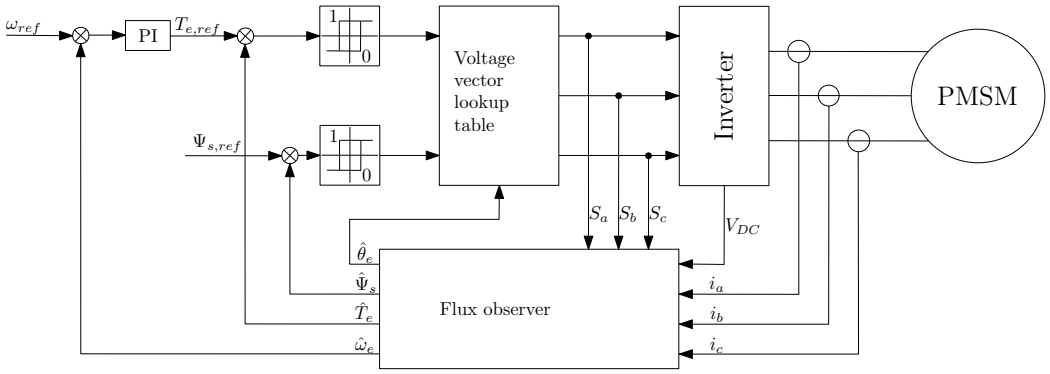


Figure 2.8: Block diagram of DTC

Figure 2.9 depicts the flux vectors of the rotor, and stator in the stator flux(xy-), rotor flux(dq-) and stationary (DQ-) reference frames.

The dq-frame is synchronized to the rotor position, i.e. the angle between the D-axis and the d-axis, is the rotor angle θ_r. The angle between rotor and stator flux linkage is known as the load angle δ.

The generated electromechanical torque depends on the load angle, and is given by equation 2.21 and 2.22, for SPMSM and IPMSM respectively [65].

$$T_{em} = \frac{3}{2}\frac{1}{L_s}p|\Psi_s||\Psi_f|\sin\delta \tag{2.21}$$

$$T_{em} = \frac{3}{4}p\frac{|\Psi_s|}{L_d L_q}(2\Psi_f L_q \sin\delta - |\Psi_s|(L_q - L_d)\sin 2\delta) \tag{2.22}$$

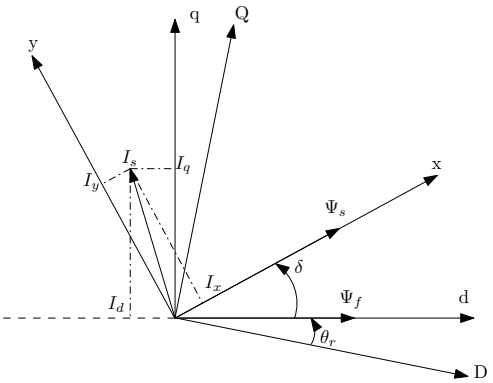


Figure 2.9: Flux and current vectors relative to different reference frames

Figure 2.10 depicts the torque as a function of load angle for a IPMSM, at several variations in stator flux linkage Ψ_s. An important observation is that in the case of $|\Psi_s| = 2\Psi_m$ there is a inverse behaviour of the torque response at the zero crossing.

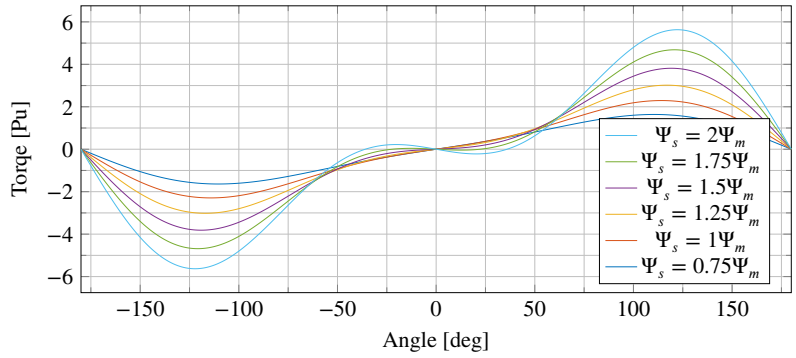


Figure 2.10: Torque as a function of the load angle δ. $\Psi_m = 0.108$ Wb, $L_d = 8.72$ mH, $L_q = 22.8$ mH

Figure 2.11 depicts the available voltage vectors from a 2-level inverter. The encircled numbers indicate the 60° sectors that define the current switching state, confined by the dashed lines.

Lookup table

Table 2.2 depicts the original DTC Lookup table (LUT), as proposed by TAKAHASHI et.al. [56]. The only modification to the table in the DTC implementation proposed in this report is that the zero voltage vectors are removed [65].

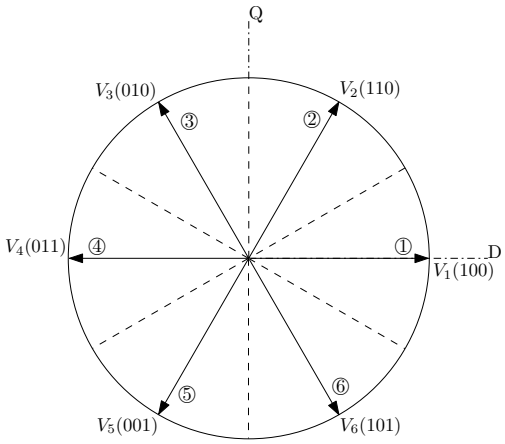


Figure 2.11: Inverter voltage vectors, and switching sectors.

Table 2.2: The original DTC switching table, as proposed in [56]

	Torque	θ_1	θ_2	θ_3	θ_4	θ_5	θ_6
$\Psi < \Psi_{ref}$	$T < T_{ref}$	$v(1,1,0)$	$v(0,1,0)$	$v(0,1,1)$	$v(0,0,1)$	$v(1,0,1)$	$v(1,0,0)$
	$T \approx T_{ref}$	$v(1,1,1)$	$v(0,0,0)$	$v(1,1,1)$	$v(0,0,0)$	$v(1,1,1)$	$v(0,0,0)$
	$T > T_{ref}$	$v(1,0,1)$	$v(1,0,0)$	$v(1,1,0)$	$v(0,1,0)$	$v(0,1,1)$	$v(0,0,1)$
$\Psi > \Psi_{ref}$	$T < T_{ref}$	$v(0,1,0)$	$v(0,1,1)$	$v(0,0,1)$	$v(1,0,1)$	$v(1,0,0)$	$v(1,1,0)$
	$T \approx T_{ref}$	$v(0,0,0)$	$v(1,1,1)$	$v(0,0,0)$	$v(1,1,1)$	$v(0,0,0)$	$v(1,1,1)$
	$T > T_{ref}$	$v(0,0,1)$	$v(1,0,1)$	$v(1,0,0)$	$v(1,1,0)$	$v(0,1,0)$	$v(0,1,1)$

Flux estimator

The traditional flux estimator is based upon equation 2.23 [56]. The open loop nature of this estimator provides a estimate that is dependent on exact knowledge of machine parameters.

$$\hat{\Psi} = \int (v_s - R_s i_s)\mathrm{d}t \tag{2.23}$$

Several more advanced estimators have been proposed in literature, some of which are covered in chapter 3, while more extensive analysis is available in [54, 25].

From the estimated values of $\alpha\beta$-flux, the flux magnitude is computed as:

$$\left|\hat{\Psi}\right| = \sqrt{\hat{\Psi}_\alpha^2 + \hat{\Psi}_\beta^2} \tag{2.24}$$

And the torque is computed as:

$$T_e = \frac{3}{2}\frac{p}{2}(\Psi_\alpha I_\beta - \Psi_\beta I_\alpha) \tag{2.25}$$

Additionally the controller needs to know the flux position, or at least in which 60°-sector it lies. This may be achieved by the following equation:

$$\theta_e = \text{atan2}(\Psi_\alpha, \Psi_\beta) \tag{2.26}$$

2.3 Model predictive control

MPC for PMSM is proposed as an alternative to the classical control alogorithms in [47] and [48], for nominal speed, and field weakening operation respectively.

MPC offers higher flexibility in the selection of control variables, when compared to the classical approaches. Additionally it avoids the requirement of a voltage modulator.

An extensive examination of the applicability of MPC to power electronic converters is presented in [50]. A comparison of MPC, deadbeat control, and FOC applied to a PMSM, is presented in [2].

A model of the system is used to predict how the system will respond to all the different possibilities in controller output. A cost function is defined for the desired control objective, and minimization of this cost function is used to determine the best output state in the current iteration.

Compared to current regulated SV-PWM, MPC is proven to have a better decoupling in the control of the $\alpha\beta$ references, a faster dynamic response to changes in the references, and a lower total harmonic distortion [50]. Basic MPC results in a variable switching frequency, but this can be improved by modification of the cost function. Another important advantage is that it is possible to use a nonlinear model in the prediction of the relevant parameters.

Equation (2.27) depicts a example of how a simple cost function is defined.

$$g = (i_\alpha^* - i_\alpha^p)^2 + (i_\beta^* - i_\beta^p)^2 \tag{2.27}$$

The value of the cost function depends on the error between reference and predicted value. Larger error gives a higher value, and thus is considered more expensive with regard to the desired control objective.

Motor control

The algorithm for MPC current control for a PMSM, or electric machines in general, may be summarized as follows:

- Obtain the reference current (from a constant or a outer control loop).
- Measure the load current.

- Predict the value of the load current in the next iteration, for each of the possible output vectors.
- Select the voltage vector that minimize the cost function (the error between the prediction and the reference).
- Generate the switching signals for the selected vector.

Figure 2.12 depicts a block diagram of typical configuration of MPC for a PMSM. The measured currents are converted to dq-frame and used along with the measured speed as parameters to the predictive model. The encoder could have been replaced by a estimator, but this is not considered a part of the MPC.

Cost function

A more application specific example of a cost function, that may be used in control of a PMSM is shown in (2.28).

$$g = (i^*_{sd} - i^p_{sd}(k+1))^2 + (i^*_{sq} - i^p_{sq}(k+1))^2 + \hat{f}(i^p_{sd}(k+1), i^p_{sq}(k+1)) \quad (2.28)$$

Where i^*_{sd} and i^*_{sq} is the d and q axis reference currents respectively. And i^p_{sd} and i^p_{sq} is the predicted d and q axis currents. $\hat{f}$ is a nonlinear function defined by (2.29), whose purpose is to limit the stator currents. If any voltage vector causes a predicted current that is higher than the maximum allowed current, the resulting cost function will go to infinity and thus will not be selected.

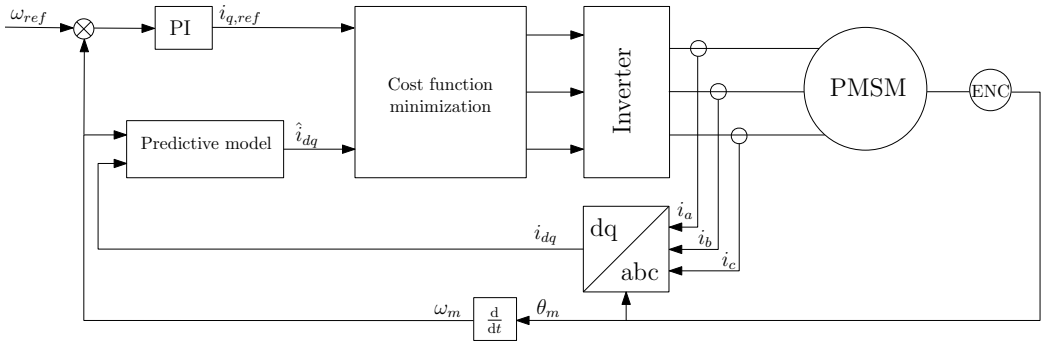


Figure 2.12: MPC used as current controller for a PMSM

$$\hat{f}(i^p_{sd}(k+1), i^p_{sq}(k+1)) = \begin{cases} \infty & \left|i^p_{sd}\right| > i_{max} \quad \text{or} \quad \left|i^p_{sq}\right| > i_{max} \\ 0 & \left|i^p_{sd}\right| \leq i_{max} \quad \text{and} \quad \left|i^p_{sq}\right| \leq i_{max} \end{cases} \tag{2.29}$$

The reference currents i^*_{sd} and i^*_{sq} are generated by a external speed control loop, e.g. by using a PI controller. The rotational direction of the motor may be changed simply by changing the sign of the q-axis reference current.

Predictive model

By using a dq reference frame oriented to the rotor magnetic field, the current in the d and q-axis may be predicted by equation (2.30a) and (2.30b).

$$i^p_{sd}(k+1) = \left(1 - \frac{R_s T_s}{L_s}\right) i_{sd}(k) + T_s \omega_r i_{sq}(k) + \frac{T_s}{L_s} v_{sd} \tag{2.30a}$$

$$i^p_{sq}(k+1) = \left(1 - \frac{R_s T_s}{L_s}\right) i_{sq}(k) - T_s \omega_r i_{sd}(k) + \frac{T_s}{L_s} v_{sq} - \psi_m \omega_r T_s \tag{2.30b}$$

Here k is a spesific sampling instant, $k+1$ is the next sampling instant and T_s is the sampling period. The rest of the variables are as defined previously. These equations are derived from the stator differential equations, by using the Euler derivative approximation depicted in (2.31).

$$\frac{\mathrm{d}i}{\mathrm{d}t} \approx \frac{i(k+1) - i(k)}{T_s} \tag{2.31}$$

The equations must be evaluated for all the possible voltage vectors, decomposed in to v_{sd} and v_{sq}. It is apparent that by using MPC, the computational intensity increases when a higher number of voltage vectors are available. E.g. by using a multilevel inverter.

Speed control

By adding speed to the cost function it is possible to avoid the external speed controller, and rely solely on MPC. The new cost function is given as:

$$\begin{aligned} g = {} & h_\omega(\omega_m^* - \omega_m^p(k+1))^2 + h_{id}(i_{sd}^* - i_{sd}^p(k+1))^2 \\ & + h_{iq}(i_{sq}^* - i_{sq}^p(k+1))^2 + \hat{f}(i_{sd}^p(k+1), i_{sq}^p(k+1)) \end{aligned} \tag{2.32}$$

Here ω_m^* is the speed reference signal, and ω_m^p is the predicted rotor speed at the next iteration. A weighting factor h is added to each of the terms in the equation, in order to account for the different units and magnitudes of the variables. The weighting factor may also be used to to consider some of the terms to be more important than others. Hence h is a design parameter used in tuning of the controller.

By considering the torque to be a constant k_t multiplied by the q-axis current, the following equation describes how the speed changes with time:

$$\frac{\mathrm{d}\omega_m}{\mathrm{d}t} = \frac{1}{J_{tot}}(pk_t i_{sq} - T_L - B\omega_m) \tag{2.33}$$

A nonlinear state space model containing all three states is given by:

$$\begin{bmatrix} \dot{i_{sd}} \\ \dot{i_{sq}} \\ \dot{\omega_m} \end{bmatrix} = \begin{bmatrix} -\frac{R_s}{L_s} & \omega_m & 0 \\ -\omega_m & -\frac{R_s}{L_s} & -\frac{\lambda_m}{L_s} \\ 0 & p\frac{k_t}{J_{tot}} & -\frac{B}{J_{tot}} \end{bmatrix} \cdot \begin{bmatrix} i_{sd} \\ i_{sq} \\ \omega_m \end{bmatrix} + \begin{bmatrix} \frac{1}{L_s} \\ \frac{1}{L_s} \\ 0 \end{bmatrix} \cdot \begin{bmatrix} v_{sd} \\ v_{sq} \end{bmatrix} \tag{2.34}$$

Where the torque constant k_t is given by:

$$k_t = \frac{3}{2}p\lambda_m \tag{2.35}$$

A modified Euler integration is used to estimate the states at each iteration:

$$\hat{x}(k+1) = x(k) + T_s\dot{x}(x(k), u(k)) \tag{2.36}$$

$$x(k+1) = x(k) + \frac{T_s}{2}\left(\dot{x}(x(k), u(k)) + \dot{x}(\hat{x}(k+1), u(k))\right) \tag{2.37}$$

Chapter 3

Flux estimation or measurement

In order to achieve efficient control of a PMSM the position of the permanent magnet flux must be known. Traditionally this knowledge has been obtained by measuring the physical position of the rotor shaft of the motor. In order to save cost and increase the reliability of the motor drive, a lot of research effort has been put into estimating the flux based on electrical measurements alone.

The estimated values may be used as a fall-back if reliability is the only concern, in many industrial applications however, the position sensor has been completely replaced by estimation techniques. The most prominent remaining issue is the performance of the estimators in the low speed region. However as there are many different estimation algorithms available this project aims at assessing the performance in several speed regions.

The speed of the motor may be extracted by differentiation of the position. Additionally the flux is required to estimate the torque, e.g. for use in DTC.

3.1 Position sensors

Rotational position sensors are typically divided into two categories: encoders and resolvers. The encoder is a digital sensor that outputs pulses depending on the position. It may function as an absolute or incremental encoder depending on how the pulses are generated. The absolute encoder provides absolute information on the postion, while the incremental encoder requires the controller to synchronise

The resolver functions as a two phase generator, providing two sinusoidal shaped voltages that are 90° out of phase. This voltage is typically converted to a dig-

ital signal by a Analog to digital converter (ADC), and further processed in order to generate position information. This process is typically performed by a Application-specific integrated circuit (ASIC) known as a by a resolver-to-digital converter, that in turn provides a digital signal to the controller.

3.2 Estimators

The observability of the rotor position of a salient PMSM, is demonstrated in [59]. It is proven that the position is observable even at zero speed operation.

The observability of the rotor position of a non-salient PMSM however is still an open issue. At zero speed there is no induced backEMF, i.e. the rotor has no effect on the electromagnetic behavior of the stator circuit. The equivalent model reduces to a static RL circuit, and hence the position cannot be observed at zero speed. Even as the speed approaches zero, the estimation is increasingly more challenging as the signal to noise ration of the observed signal increases.

Although the inobservability of the non-salient PMSM at zero speed is proven, there is one practical consideration that should be taken into account. The relative permeability of permanent magnets lies in the area of 1.05. I.e. it is close to, but not exactly the same as air. This may cause a small amount of saliency even if the magnets are surface mounted. This feature has been exploited for position estimation.

In [54, 61] a comparison of several different flux estimators for Pulse width modulation (PWM)-based DTC is evaluated. The paper [54] concludes that a estimator initially proposed in [25] is the most suited among the tested candidates.

3.3 Voltage model

The voltage model for flux estimation depends on the measurements of voltage and current. Unfortunately this model is sensitive to variations in winding resistance, as well as inaccuracies in the current and voltage measurements [61]. It is important to note that unlike the induction machine, the PMSM has a non-zero initial flux vector caused by the permanent magnets. Thus the initial rotor position needs to be determined before the voltage model is applied for flux estimation.

$$\lambda_s = \lambda_{s0} + \int_t^{t+\Delta t} (v_s - R_s i_s)\mathrm{d}t \tag{3.1}$$

One major issue with the basic integration of the back-EMF, is that the pure integration causes drift in the estimated flux. Any DC offset in the measurements, no matter how small it is, will cause drift. In a discrete implementation there will be a fixed data width for the integrator, and eventually it will saturate.

The initial value problem is another issue causing offset. If a sine wave is applied to a pure integrator one expects a cosine wave at the output. This however is only true if the wave is initially applied at the its positive or negative peak. Otherwise there will be a constant DC-offset on the output [25].

In order to overcome the issues of DC offset and drift in the estimator, it is possible to apply a Low-pass filter (LPF) to the output. It that case however, it is important to consider the phase shift and magnitude error caused by the low pass filter.

$$H(s) = \frac{1}{1 + j\tau\omega} \tag{3.2}$$

Several more sophisticated compensation schemes are proposed in [25]. The methods utilize feedback in order to compensate for drift in the integration, but are still to be regarded as open loop with respect to the motor.

3.4 Current model

The current model for flux estimation depends on the measurements of current and rotor position. This method typically offers higher accuracy than the voltage model, but the requirement of a position sensor is often considered a major drawback [61]. In addition the permanent magnet flux, and the inductances may change, depending on temperature and magnetic saturation of the machine.

The current model method for a nonsalient PMSM is given in $\alpha\beta$ frame, by equation (3.3), and (3.4) respectively.

$$\Psi_\alpha = L_s i_\alpha + \Psi_{PM} \cos(\theta) \tag{3.3}$$

$$\Psi_\beta = L_s i_\beta + \Psi_{PM} \cos(\theta) \tag{3.4}$$

Alternatively in dq reference frame the general expressions are:

$$\Psi_d = I_{sd} L_d + \Psi_{PM} \tag{3.5}$$

$$\Psi_q = I_{sq} L_q \tag{3.6}$$

Chapter 4

Flux linkage observers

In order to overcome some of the issues with the basic current and voltage model, various observer techniques may be applied to the system. In this project several observer techniques are studied in order to estimate the back-EMF, and from the back-EMF, the flux linkage and the rotor position. In particular the *Luenberger observer*, and several variations of the *Sliding mode observer (SMO)*. Additionally with the increasing performance of modern digital controllers the *Extended Kalman filter* has also been proven applicable as a flux linkage observer [5].

The following sections covers the theoretical development, and practical implementation of the observers. The simulation, and evaluation is covered in chapter 12.

4.1 Luenberger state observer

The Luenberger observer uses the difference between the observer outputs and the actual plant outputs as a compensation term. I.e the observation error is used to force the observer states to converge to the states of the plant [21].

The discrete difference equations for the observer is given by:

$$\hat{x}(k+1) = A\hat{x}(k) + L[y(k) - \hat{y}(k)] + Bu(k) \tag{4.1}$$

and

$$\hat{y}(k) = C\hat{x}(k) + Du(k) \tag{4.2}$$

Where A, B, C and D are the system matrix, input matrix, output matrix and feedforward matrix respectively. An where L is the observer gain matrix.

Derivation of the position information

If the back-EMF is sinusoidal, we have the following relationship between rotor position and EMF [21]:

$$e_\alpha(\theta_r) = -k_e\omega_r \sin(\theta_r) \tag{4.3a}$$
$$e_\beta(\theta_r) = k_e\omega_r \cos(\theta_r) \tag{4.3b}$$

We also have that:

$$k_e\omega_r = \sqrt{e_\alpha^2 + e_\beta^2} \tag{4.4}$$

The rotor position may then be expressed as a function of the back-EMF by:

$$\theta_r = \arccos\left(\frac{e_\beta}{\sqrt{e_\alpha^2 + e_\beta^2}}\right) \tag{4.5}$$

However as trigonometric calculations are computationally intensive, it is sometimes beneficial to express the sine and cosine of the angle instead. Thereby the result may be used directly in the computation of the direct and inverse park transformation, which is where the position information is commonly required:

$$\cos(\theta_r) = \frac{e_\beta}{\sqrt{e_\alpha^2 + e_\beta^2}} \tag{4.6a}$$

$$\sin(\theta_r) = \frac{e_\alpha}{-\sqrt{e_\alpha^2 + e_\beta^2}} \tag{4.6b}$$

By using the voltage equations for the PMSM it is possible to derive expressions for the EMF as function of applied voltage and current.

$$v_\alpha = R_s i_\alpha + L_s \frac{\mathrm{d}i_\alpha}{\mathrm{d}t} + e_\alpha \tag{4.7a}$$

$$v_\beta = R_s i_\beta + L_s \frac{\mathrm{d}i_\beta}{\mathrm{d}t} + e_\beta \tag{4.7b}$$

The induced back-EMF is considered as a disturbance, with the following model:

$$\frac{\mathrm{d}e_\alpha}{\mathrm{d}t} = 0 \tag{4.8a}$$

$$\frac{\mathrm{d}e_\beta}{\mathrm{d}t} = 0 \tag{4.8b}$$

By combining (4.7) and (4.8) we arrive at the state space model:

$$\begin{bmatrix} \dot{i}_\alpha \\ \dot{i}_\beta \\ \dot{e}_\alpha \\ \dot{e}_\beta \end{bmatrix} = \underbrace{\begin{bmatrix} -\frac{R_s}{L_s} & 0 & -\frac{1}{L_s} & 0 \\ 0 & -\frac{R_s}{L_s} & 0 & -\frac{1}{L_s} \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}}_{\mathrm{A}} \begin{bmatrix} i_\alpha \\ i_\beta \\ e_\alpha \\ e_\beta \end{bmatrix} + \underbrace{\begin{bmatrix} \frac{1}{L_s} \\ \frac{1}{L_s} \\ 0 \\ 0 \end{bmatrix}}_{\mathrm{B}} \begin{bmatrix} v_\alpha \\ v_\beta \end{bmatrix} \tag{4.9}$$

Based on the extended state space model, the Luenberger observer is obtained:

$$\begin{bmatrix} \dot{\hat{i}}_\alpha \\ \dot{\hat{i}}_\beta \\ \dot{\hat{e}}_\alpha \\ \dot{\hat{e}}_\beta \end{bmatrix} = A \begin{bmatrix} \hat{i}_\alpha \\ \hat{i}_\beta \\ \hat{e}_\alpha \\ \hat{e}_\beta \end{bmatrix} + B \begin{bmatrix} v_\alpha \\ v_\beta \end{bmatrix} + \underbrace{\begin{bmatrix} K_1 \\ K_2 \\ K_3 \\ K_4 \end{bmatrix} \left(\begin{bmatrix} i_\alpha \\ i_\beta \end{bmatrix} - \left(\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix} \begin{bmatrix} \hat{i}_\alpha \\ \hat{i}_\beta \\ \hat{e}_\alpha \\ \hat{e}_\beta \end{bmatrix} \right) \right)}_{\text{Correction term}} \tag{4.10}$$

The last term in the observer corrects the current state estimate by a factor proportional to the prediction error. Where the prediction error is the difference between estimated and actual current.

By proper selection of the gains in the K matrix, it is possible to have the observer states converge to the actual states of the motor.

In discrete time the difference equation used to implement this observer is given as:

$$\hat{x}(k+1) = A\hat{x}(k) + Bu(k) + K(y(k) - (C\hat{x}(k))) \tag{4.11}$$

Where k denotes that the value is valid at the current time step, and $k+1$ denotes that the value is valid at the next time step.

4.2 Sliding mode observer

This section covers the theoretical operation of a sliding mode state observer for a PMSM [21].

Figure 4.1 depicts a simplified block diagram of a SMO, where the measured current and voltage in $\alpha\beta$-frame is used as input, and where the estimated back-EMF is the output.

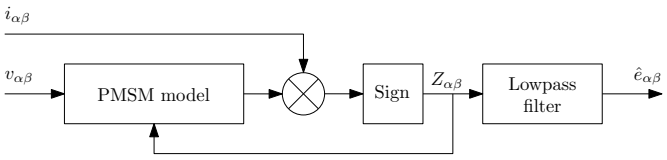


Figure 4.1: Block diagram of simple sliding mode observer.

Equation (4.12) depicts the PMSM model, that is used for position estimation.

$$\frac{\mathrm{d}i_s^p}{\mathrm{d}t} = Ai_s^p + B(v_s^* - e_s^p - z) \tag{4.12}$$

Here i_s^p is the predicted stator current, v_s^* is the stator voltage reference, e_s^p is the predicted stator back-EMF, and z is the bang-bang control signal.

A and B is the matrices depicted in (4.13), and (4.14) respectively.

$$A = \begin{bmatrix} -\frac{R_s}{L} & 0 \\ 0 & -\frac{R_s}{L} \end{bmatrix} \tag{4.13}$$

$$B = \begin{bmatrix} \frac{1}{L} & 0 \\ 0 & \frac{1}{L} \end{bmatrix} \tag{4.14}$$

R_s is the stator winding resistance, and $L = \frac{2}{3}L_m$, where L_m is the stator magnetizing inductance.

The bang-bang controller depicted in (4.15), is driven by the error between actual and predicted current, and has the goal of driving this error to zero.

$$z = k \cdot sign(i_s^p - i_s) \tag{4.15}$$

The continuous time state-space model may be converted into discrete time by applying the matrices (4.16), and (4.17) for transformation of system, and input matrix respectively [16].

$$F = e^{AT_s} \tag{4.16}$$

$$G = \left(\int_0^{T_s} e^{A(t-\tau)} \mathrm{d}\tau \right) B \tag{4.17}$$

Here A is the system matrix, and B is the input matrix in continuous time. T_s is the sampling period.

We arrive at the following discrete time equations for the SMO:

$$i_s^p(n+1) = F i_s^p(n) + G(v_s^*(n) - e_s^p(n) - z(n)) \tag{4.18}$$

$$z(n) = k \cdot sign(i_s^p(n) - i_s(n)) \tag{4.19}$$

Where the matrices F and G are given by (4.20), and (4.21) respectively.

$$F = \begin{bmatrix} e^{-\frac{R_s}{L}T_s} & 0 \\ 0 & e^{-\frac{R_s}{L}T_s} \end{bmatrix} \tag{4.20}$$

$$G = \begin{bmatrix} \frac{1}{R}\left(1 - e^{-\frac{R_s}{L}T_s}\right) & 0 \\ 0 & \frac{1}{R}\left(1 - e^{-\frac{R_s}{L}T_s}\right) \end{bmatrix} \tag{4.21}$$

By filtering the control signal z, it is possible to obtain a estimate of the induced back-EMF.

The transfer function for a first order lowpass filter is depicted in (4.22).

$$H(s) = \frac{\omega_0}{s + \omega_0} \tag{4.22}$$

Here ω_0 is given by $2\pi \cdot f_0$ where f_0 is the filter cutoff frequency. Translated into discrete time, and applied for our estimator, the same filter is depicted in (4.23).

$$e_s^p(k+1) = e_s^p(k) + \omega_0(z(k) - e_s^p(k)) \tag{4.23}$$

The estimated back-EMF is a vector with angle corresponding to the rotor position angle. In discrete time this algorithm is evaluated in two separate expressions for the α and β-axis respectively. The *arctan2* function may be used to compute the angle from the α and β values.

$$\theta_m = \arctan(-e_{s\alpha}^p, e_{s\beta}^p) \tag{4.24}$$

4.3 Improved sliding mode observer

A improved sliding mode observer, based on the *sigmoid* function is proposed in [35]. Simulations and laboratory experiments has proven higher performance when compared to the classical *signum* function.

The *sigmoid* function is defied by:

$$\frac{1}{1+e^{-t}} \tag{4.25}$$

Based upon this equation the switching functions are defined in 4.26 for α and β axes respectively.

$$H(i_\alpha) = \frac{2}{1+e^{-ai_\alpha}} + 1 \tag{4.26a}$$

$$H(i_\beta) = \frac{2}{1+e^{-ai_\beta}} + 1 \tag{4.26b}$$

The functions are scaled and biased to have a response from -1 to 1. a is a positive constant used to control the slope of the sigmoid curve. Figure 4.2 provides an example of the slope of the *sigmoid* function. The more gradual transition from -1 to 1, causes less chattering, which improves the overall performance.

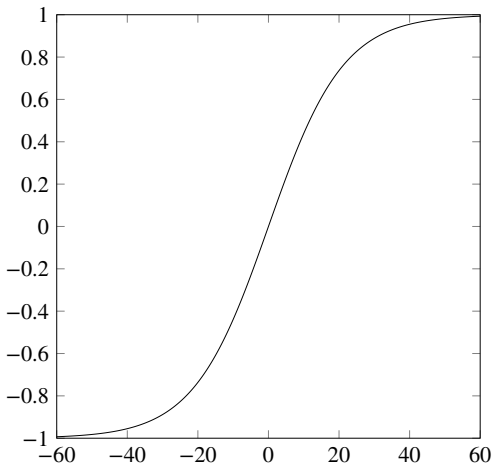


Figure 4.2: Typical slope of the sigmoid function

Phase compensation

Due to the filtering in the estimator, there is a phase shift between the estimated and actual position. The phase shift depends on the speed of the motor, and may be compensated by using the estimated speed, or by using the speed reference [15].

This compensation may be calculated by equation (4.27).

$$\Delta\theta_e = K_b + K_s \cdot \hat{\omega}_e \tag{4.27a}$$

$$\Delta\theta_e = K_b + K_s \cdot \omega_{e,ref} \tag{4.27b}$$

Where K_b and K_s is the bias, and scaling factors respectively, which depend on the system in question. $\hat{\omega}_e$ is the estimated speed, and $\omega_{e,ref}$ is the speed reference.

4.4 Extended Kalman filter flux estimator

The Kalman filter is a popular linear state estimator [21]. The Extended Kalman filter (EKF) is a nonlinear version of the Kalman filter, where the nonlinear system is linearized around the current working point at each iteration.

From a practical point of view the implementation of the Kalman filter is more complicated, and more computationally intense than the sliding mode observer discussed previously. It may however provide a better performance.

The nonlinear system may be modelled as [5]:

$$\dot{x}(t) = f[x(t)] + Bu(t) + \sigma(t) \tag{4.28}$$

$$y(t) = h[x(t)] + \mu(t) \tag{4.29}$$

Where $x(t)$, $u(t)$ and $y(t)$ is the state, input and output matrices respectively. $\sigma(t)$ and $\mu(t)$ is the system and measurement noise, each with a covariance $Q(t)$ and $R(t)$ respectively.

Based upon these equations the linearized perturbation equations are:

$$\delta\dot{x}(t) = F[x(t)]\delta x(t) + B\delta u(t) + \sigma(t) \tag{4.30}$$

$$\delta y(t) = H[x(t)]\delta x + \mu(t) \tag{4.31}$$

Where the matrices F, and H are defined as follows:

$$F[x(t)] = \left.\frac{\partial f}{\partial x}\right|_{x=x(t)} \tag{4.32}$$

$$H[x(t)] = \left.\frac{\partial h}{\partial x}\right|_{x=x(t)} \tag{4.33}$$

Kalman filter algorithm

The Extended Kalman filter algorithm is a two step recursive process [63]. It can be described by the following equations:

Step 1. *Prediction*

State estimate

$$\hat{x}_{n|n-1} = \hat{x}_{n-1} + (f[x_{n-1}] + Bu_{n-1})T_c \tag{4.34}$$

Covariance update

$$P_{n|n-1} = \Phi_{n-1} P_{n-1} \Phi_{n-1}^T + Q_d \tag{4.35}$$

Step 2. *Innovation*

Correction of state estimate:

$$\hat{x}_n = \hat{x}_{n|n-1} + K_n(y_n - H\hat{x}_{n|n-1}) \tag{4.36}$$

Correction of covariance estimate:

$$P_n = P_{n|n-1} - K_n H P_{n|n-1} \tag{4.37}$$

Kalman gain

$$K_n = P_{n|n-1} H^T [H P_{n|n-1} H^T + R]^{-1} \tag{4.38}$$

Development of PMSM model for Kalman filter

$L_s \equiv L_d = L_q$ for a nonsalient PMSM.

$$\begin{bmatrix} v_\alpha \\ v_\beta \end{bmatrix} = \begin{bmatrix} R_s + \frac{\mathrm{d}L_s}{\mathrm{d}t} & 0 \\ 0 & R_s + \frac{\mathrm{d}L_s}{\mathrm{d}t} \end{bmatrix} \begin{bmatrix} i_\alpha \\ i_\beta \end{bmatrix} + \omega_e \lambda_f \begin{bmatrix} -\sin(\theta_e) \\ \cos(\theta_e) \end{bmatrix} \tag{4.39}$$

If we choose the state variable matrix $x(t) = \begin{bmatrix} i_\alpha & i_\beta & \omega_e & \theta_e \end{bmatrix}^T$, the state space model of the PMSM in $\alpha\beta$ coordinate frame is given by:

$$\begin{bmatrix} \dot{i_\alpha} \\ \dot{i_\beta} \\ \dot{\omega_e} \\ \dot{\theta_e} \end{bmatrix} = \underbrace{\begin{bmatrix} -\frac{R_s}{L_s} i_\alpha + \frac{\lambda_f}{L_s} \omega_e \sin(\theta_e) \\ -\frac{R_s}{L_s} i_\beta + \frac{\lambda_f}{L_s} \omega_e \cos(\theta_e) \\ 0 \\ \omega_e \end{bmatrix}}_{\text{f[x(t)]}} + \underbrace{\begin{bmatrix} \frac{1}{L_s} & 0 \\ 0 & \frac{1}{L_s} \\ 0 & 0 \\ 0 & 0 \end{bmatrix}}_{\text{B}} \begin{bmatrix} v_\alpha \\ v_\beta \end{bmatrix} \tag{4.40}$$

$$\begin{bmatrix} i_\alpha \\ i_\beta \end{bmatrix} = \underbrace{\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix} \begin{bmatrix} i_\alpha \\ i_\beta \\ \omega_e \\ \theta_e \end{bmatrix}}_{\text{h[x(t)]}} \tag{4.41}$$

The Jacobian matrix is then given as follows:

$$F[x(t)] = \frac{\partial f}{\partial x}\Big|_{x=x(t)} = \begin{bmatrix} -\frac{R_s}{L_s} & 0 & \frac{\lambda_f}{L_s} \sin(\theta_e) & \frac{\lambda_f}{L_s} \omega_e \cos(\theta_e) \\ 0 & -\frac{R_s}{L_s} & -\frac{\lambda_f}{L_s} \cos(\theta_e) & \frac{\lambda_f}{L_s} \omega_e \sin(\theta_e) \\ 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix} \tag{4.42}$$

The output matrix is given as follows:

$$H[x(t)] = \frac{\partial h}{\partial x}\Big|_{x=x(t)} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix} \tag{4.43}$$

Part II

Digital controller

Chapter 5

Digital controller

In order to implement and test the algorithms previously presented a digital control platform has been designed. The platform is based upon the Xilinx Zynq®-7000 System on Chip (SoC), which is a combination of a dual ARM Cortex A9 Central processing unit (CPU) and a Field-programmable gate array (FPGA).

The FPGA offers a flexible and efficient way of offloading the more computationally intensive algorithms from the CPU [40].

The Avnet Zedboard evaluation board for the Zynq®-7000 SoC is used. However a different board utilizing the same SoC could also be a viable alternative. In particular the MicroZed™ is a cheaper alternative without compromising on the required features for motor drive applications.

This chapter presents the development environment, and some of the low-level considerations that are required when working with this controller.

5.1 Development environment

The development environment consist primarily of two software suites. Mathworks MATLAB™/Simulink® and Xilinx Vivado®. Some additional packages are required for the Matlab suite, in particular the embedded coder, HDL-conder and support package for the Xilinx Zynq-7000. For Vivado the freely available WebPack edition is sufficient.

5.2 Custom operating system

A custom GNU/Linux image for the Zynq has been developed by Roland Peterer and Christian Schmid. This system, henceforth known as *Lianx*, has some additional features to simplify the development of control algorithms on the Zynq®-7000.

In particular several reference designs are provided along with the operating system, which allows easy access to some of the peripherals of the controller.

Vivado project

Figure 5.1 depict the Vivado block diagram of the reference design, developed in close cooperation with Roland Peterer and Christian Schmid. When parts of the Simulink® model is converted to HDL-code, a option exists to automatically generate a Vivado project. By selecting the predefined reference design, a separate Intellectual property core (IP-core) is inserted in the block diagram.

This new block contains the HDL parts of the motor control algorithm. The other parts may be compiled into a Linux executable, and the two systems communicate using Advanced eXtensible Interface 4 (AXI4).

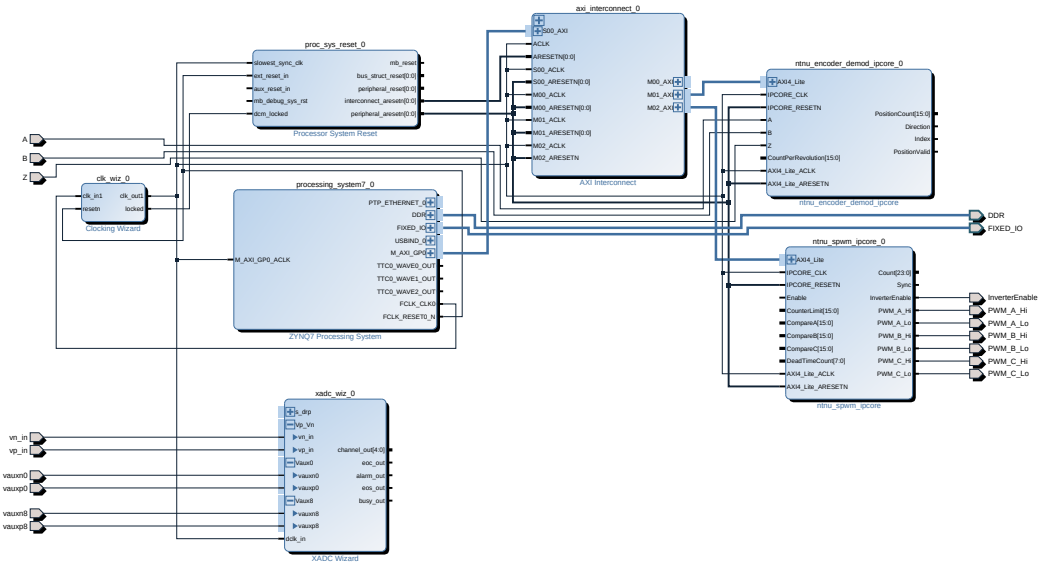


Figure 5.1: Reference design block diagram in Vivado

5.3 Basic building blocks

When developing FPGA logic using Simulink® and HDL-coder, there are some restrictions to the building blocks that may be used [1].

This section aims at providing some insight into the development process, and also to provide some useful basic building blocks that has been developed as a supplement to the blocks supplied with Simulink®.

Sample delay

The unit delay block provides a one sample delay between the input and output. This is a required building block for many of the more advanced flip-flops, detectors, counters, etc.

As an example a simple integrator is implemented by a adder and a unit delay block, as depicted in figure 5.2.

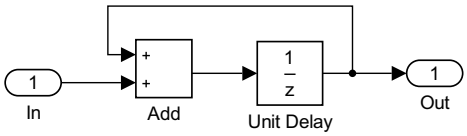


Figure 5.2: Integrator

Listing 5.1 provides an example of the automatically generated VHSIC Hardware Description Language (VHDL) code for this model [2]. The signals data types are specified in the parameter windows of the Simulink® blocks. In the example this has been translated to: `unsigned(7 DOWNTO 0)`, which specifies 8-bit unsigned integer.

The clock, reset, and enable signals are added automatically, their connections are abstracted from the Simulink® model. A process statement is used to implement the logic, where the output of the unit delay block is assigned the value of the add block at the rising edge clock, i.e. when: `clk'EVENT AND clk = '1'`.

Signal assignment is performed with the `<=` operator, i.e. `Add_out1` is always equal the sum of `Unit_Delay_out1` and `In1_unsigned`.

[1] Before starting the development the Matlab command "*hdllib on*" should be executed to ensure that any incompatible blocks are removed from the library browser.

[2] Very High Speed Integrated Circuit (VHSIC)

Listing 5.1: VHDL code for integrator.

```
1  LIBRARY IEEE;
2  USE IEEE.std_logic_1164.ALL;
3  USE IEEE.numeric_std.ALL;
4
5  ENTITY Subsystem_block IS
6  PORT( clk          :   IN    std_logic;
7  reset              :   IN    std_logic;
8  enb                :   IN    std_logic;
9  In1                :   IN    std_logic_vector(7 DOWNTO 0);
10 Out1               :   OUT   std_logic_vector(7 DOWNTO 0)
11 );
12 END Subsystem_block;
13
14
15 ARCHITECTURE rtl OF Subsystem_block IS
16
17 -- Signals
18 SIGNAL In1_unsigned            : unsigned(7 DOWNTO 0);
19 SIGNAL Unit_Delay_out1         : unsigned(7 DOWNTO 0);
20 SIGNAL Add_out1                : unsigned(7 DOWNTO 0);
21
22 BEGIN
23 In1_unsigned <= unsigned(In1);
24
25 Add_out1 <= Unit_Delay_out1 + In1_unsigned;
26
27 Unit_Delay_process : PROCESS (clk, reset)
28 BEGIN
29     IF reset = '1' THEN
30         Unit_Delay_out1 <= to_unsigned(16#00#, 8);
31     ELSIF clk'EVENT AND clk = '1' THEN
32         IF enb = '1' THEN
33             Unit_Delay_out1 <= Add_out1;
34         END IF;
35     END IF;
36 END PROCESS Unit_Delay_process;
37
38 Out1 <= std_logic_vector(Unit_Delay_out1);
39
40 END rtl;
```

Flip-flops

The unit delay block is basically a D-type flip-flop with a hidden clock input, i.e. it runs with the clock of the FPGA. Based upon this block more advanced flip-flops may be derived.

The following flip-flops are useful as long as the usage requires trigging on the system clock frequency. If different clock signals are required, a triggered subsystem may be utilized.

Toggle flip-flop

The toggle flip-flop depicted in 5.3, toggles the output at each clock cycle, as long as the input is true.

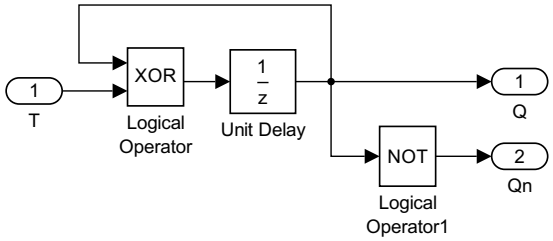


Figure 5.3: Toggle flip-flop

JK flip-flop

The JK flip-flop depicted in 5.4, is a extension to the SR flip-flop, in which all input states have meaning. The truth table is given in 5.1.

Table 5.1: Truth table for JK flip-flop

J	K	Q
0	0	Q
0	1	0
1	0	1
1	1	$\overline{Q}$

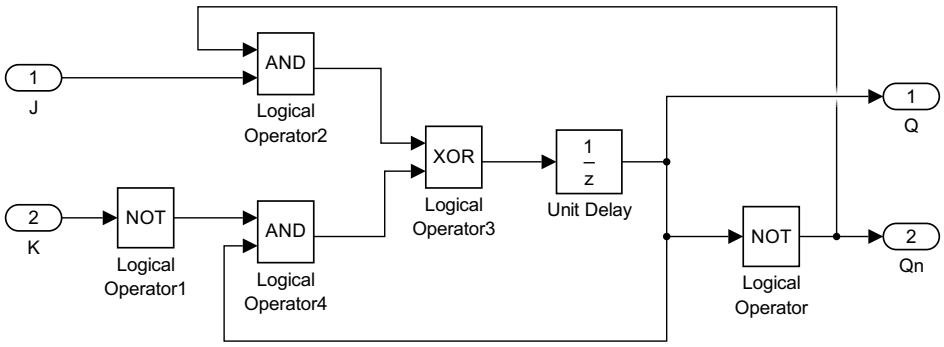


Figure 5.4: JK flip-flop

Edge triggers

It is often required to be able to detect the rising or falling edge of a digital signal. An efficient digital implementation may be achieved by comparing the current and the previous sample of the signal.

This is depicted in figure 5.5 for a rising edge detector.

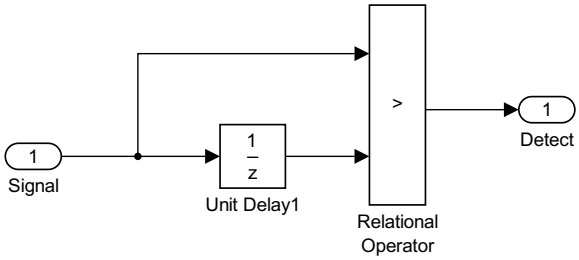


Figure 5.5: Rising edge detector

Similarly a change detector is implemented by trigging on any difference between the previous sample and the current input.

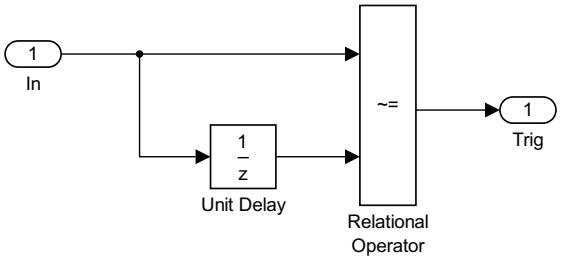


Figure 5.6: Change detector

The relational operator may be changed depending on the required behavior.

Counters

HDL-coder comes with several built in counters that run on the FPGA clock frequency. It is however sometimes desirable to count on a external signal, for instance the signal from an incremental position encoder.

Figure 5.7 depict a example of a special purpose counter designed to count the pulses from an incremental encoder. The operation of this circuit should be easy to analyse for anyone familiar with Simulink®. A external reset logic, and a rotational direction signal is required for the circuit to operate as intended.

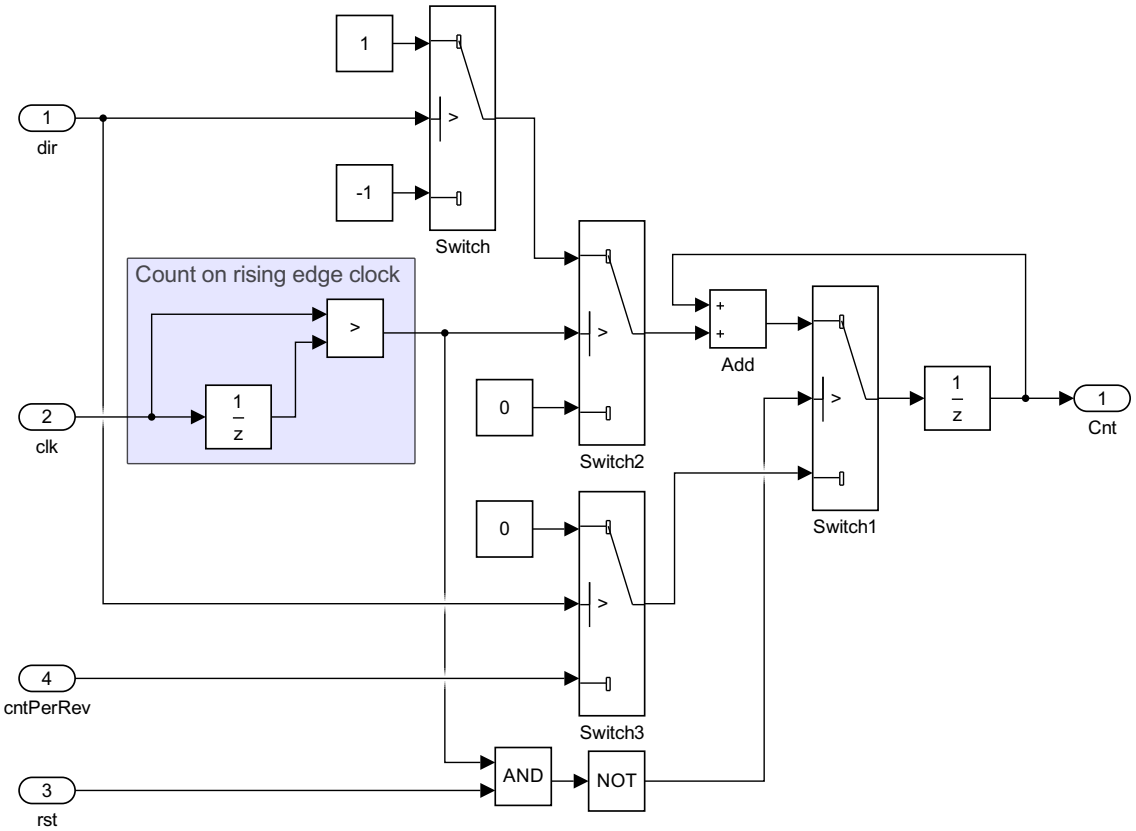


Figure 5.7: Up/down counter

1Hz clock generator

In order to have a visual indication that the FPGA is running, a 1Hz 50% duty cycle signal is generated and supplied to one of the LED's on the Zedboard.

Figure 5.8 depicts the block diagram of the generator. A counter running on the FPGA clock, is reset when it reaches a value equal to the clock frequency in Hz. I.e. it counts for one second. The generated reset signal is also used to toggle a flip-flop connected to the LED.

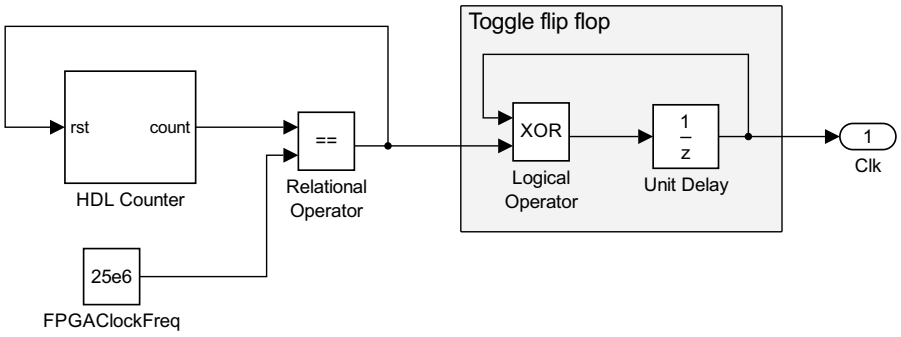


Figure 5.8: 1Hz generator

5.4 Data types

When developing a embedded system combining programmable logic with a floating point capable CPU, it is important to consider how the data is represented in various parts of the system.

Traditionally, in an environment where floating point is not supported, a Per unit (p.u.) system has been used to normalize all values. This ensures that the entire dynamic range of the variables is utilized.

With a CPU that incorporates a Floating-point unit (FPU), this is no longer a requirement. It should however be noted that a p.u. system has other advantages.

The Programmable logic (PL) does not support floating point, and the data-type of data transferred between the two systems must be converted. The developer has the choice between scaling to p.u., or adapt the data-types depending on the required dynamic range of the number that is to be represented.

Floating point data

Floating point data is a digital data type for representation of numbers approximating real numbers. The numbers are represented as a fixed number of the most significant digits, scaled to an exponent.

$$\text{number} = \text{significant digits} \cdot 10^{\text{exponent}} \tag{5.1}$$

Floating point has a very high dynamic range, when compared to other data types. This is caused by the fact that the range between two representable values increases with increasing scale.

The most commonly used standard for floating point number representation and arithmetic is IEEE 754 defined in [1].

Fixed point data

In fixed point data-representation a fixed number of bits are used for the integer and fractional part of the number. I.e. the radix point stays at a fixed position.

For any given word length, the position of the radix point must be specified in order to efficiently represent the data with maximum resolution.

A common format for fixed point number is the Q-format. Where the letter Q stems from the common notation for a rational number. The common notation for specifying a Q number is $Q.m.n$, where m is the number of integer bits, and n is the number of fractional bits.

In Simulink® the syntax for specifying fixed point data is:

fixdt(s, w, f)

Where s defines whether the number is signed or not, w is the word length, and f is the fractional length. The integer part is then given as: $i = w - f$.

Signal representation

As an example one might consider a situation where the developer needs to represent several physical signals using a 16-bit word. Suppose one signal is the position of a motor, and the other is the motor speed. If the motor has a rated speed of 1500RPM, one may consider the following domains for the signals:

$$0 \leq \theta \leq 2 \cdot \pi$$
$$-157 \leq \omega \leq 157$$

Where θ is the position in rad, and ω is the speed in rad/s.

The position may be represented using 3-bit integer, and 13-bit fraction, while the speed requires 9-bit signed integer, and only has 7-bit left for the fractional part. The resolution however, relative to the dynamic range of the number, is the same.

Alternatively if the developer chooses to use p.u. representation, only one bit may be used for the integer part. I practice however it is important to consider that some signals may exceed the ratings. In particular for this example the speed should probably support at least 2 p.u.

5.5 Analog to digital conversion

This chapter covers the software and HDL implementation of the analog to digital conversion. The hardware is described in part IV of the report. However in order to implement correct scaling and biasing of the measurements, some hardware specific information is provided here as well.

Two different methods have been utilized in this project. The built in XADC of the Zedboard, and the $\Delta\Sigma$-modulators on the custom interface card.

XADC

The built in Xilinx Analog Digital Converter (XADC) of the Zynq®-7000 supports communicating the acquired data via the AXI4 interface to the Processing system (PS), or via the Dynamic reconfiguration port (DRP), which also allows communication directly with the PL.

In this project the DRP interface is used, as it provides the required measurement for the high speed algorithm running in the PL.

A interface controller for the XADC is provided by Roland Peterer and Christian Schmid as part of their *Lianx* operating system.

The basic outline of the Simulink® model of the interface is provided in figure 5.9. The Matlab™- source for the matlab subsystem, i.e. the DRP interface, is provided in appendix E.1.

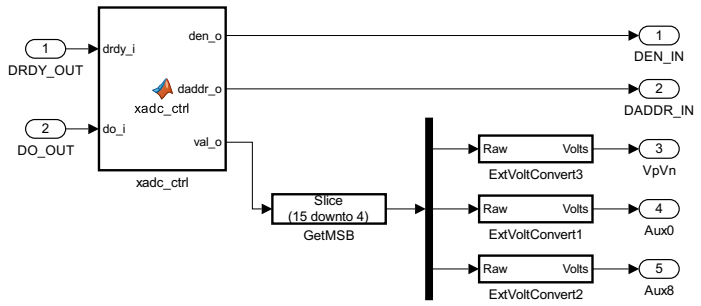


Figure 5.9: XADC DRP interface

5.6 ΔΣ-modulators

The Analog Devices AD7403 16-bit isolated Sigma-Delta modulator[10] is used on the interface card designed for the Zedboard. The most relevant parts of the datasheet are given in appendix D.2.

Sinc filter

In order to obtain the digital representation of the analog value, the pulse train must be filtered. The datasheet[10] for the modulator recommends a third order *sinc* filter for this purpose. An Verilog code example of the filter is provided in the datasheet. The data is read at the positive clock edge, and the code supports decimation rates from 32 to 4096.

Chapter 6

Custom IP cores

When developing a digital circuit using a Register-transfer level (RTL) language, it is common to package the functionality in a so called IP-core. The term comes from the licensing of the core. It is not a physical product, but rather the description of how it works that is the property of the owner [1].

This chapter covers some basic low level functionality of the inverter, that has been packaged into separate IP-cores.

[1] As the developed algorithms in this project are open source, the term *intellectual freedom core*, or *open source core* would probably be more suited.

6.1 Voltage modulator

The pulse width voltage modulator is divided into two subsystems. The first system has the task of computing the compare values depending on the desired output voltage. The second system has the task of generating the carrier wave, and switching the transistors depending on the input compare values. The output frequency is determined by the rate of change of the compare values.

Voltage to compare computation

When the controller has commanded a certain voltage, the required compare values to generate this voltage must be computed.

There is a given relationship between compare value and output voltage, depending on the DC-link voltage and on the counter limit. Zero voltage means 50% duty cycle, i.e. a bias equal to half the PWM counter limit should be added.

Sinusoidal PWM

In Sine Pulse Width Modulation (SPWM) the references to the modulator are the sinusoidally shaped voltages typically obtained from the inverse Park-Clarke transformation.

Space vector PWM

The theoretical operation of SV-PWM is provided in section 2.1.

Figure 6.1 depicts a simple implementation of the SV-PWM [26]. The implementation requires the input references to be the sinusoidal three phase voltage references.

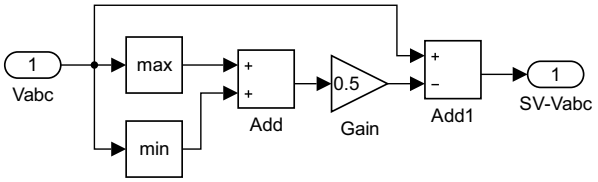


Figure 6.1: Simple SV-PWM.

Pulse width modulator

The selected pulse width modulator is of the symmetrical type, i.e. it has a triangular shaped carrier wave.

Figure 6.2 depicts the block diagram of the pulse width modulator. It consists of a up/down counter that generates the triangular wave with a peak value equal to the counter limit. The counter is running at the FPGA clock frequency, hence the relationship between counter limit and switching frequency is given by:

$$f_{sw} = \frac{f_{clock}}{\text{CounterLimit}} \tag{6.1}$$

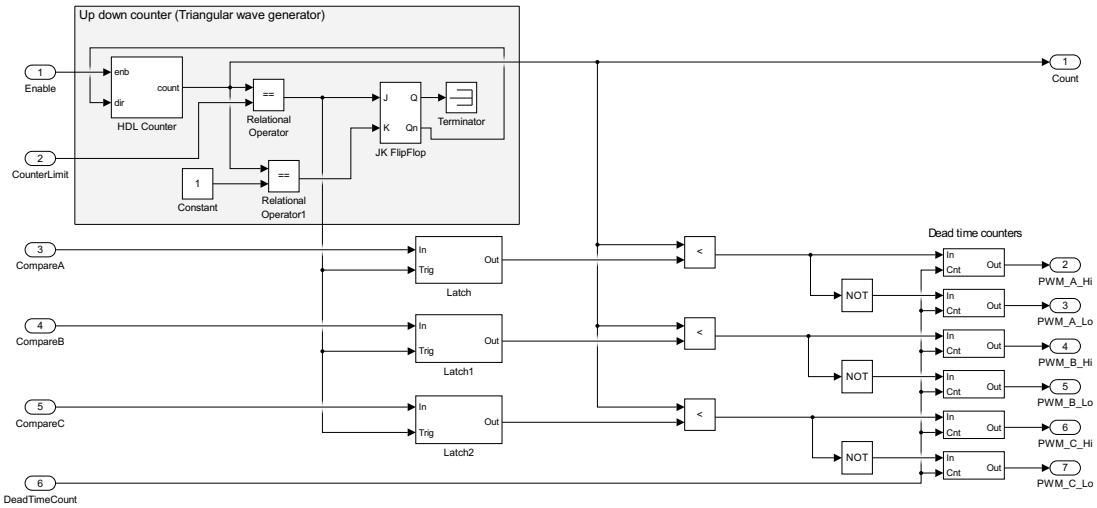


Figure 6.2: Symmetrical pulse width modulator

The counter output is compared to three reference values in order to generate a three phase switching signal. The upper transistor on each phase leg should turn on when the compare value is higher than the carrier.

The compare values are latched in at the peak of the carrier wave and the output pulses have adjustable on-delay to protect the transistors. The operation of these subsystems are explained in detail in the following sections.

Latch

The compare value to the modulator is latched in at the peak of the carrier wave. This ensures that it stays constant during the entire switching cycle.

The block diagram for the latch is depicted in figure 6.3. The *trig* input goes high when the counter has reached the limit, this causes the output to equal the input. When *trig* goes low, a unit delay block functions as a memory remembering the previous input value.

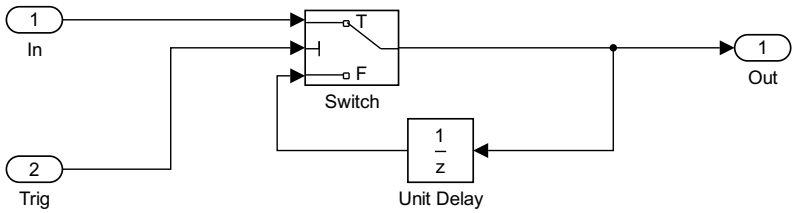


Figure 6.3: Latch

Dead time counter

The dead time counter provides a on-delay for the control signals to the transistors. When the input goes high, the counter is enabled and the output is delayed until it reaches it's final value. When the input goes low, the counter is immediately reset, such that there is no off-delay.

The correct setting for the count parameter depends on the required delay of the transistors, as well as on the clock frequency for the FPGA. Equation (6.2) depicts the relationship between the count parameter and actual delay time in seconds.

$$t_{delay} = \frac{1}{f_{clock}} \cdot \text{Count} \tag{6.2}$$

Where f_{clock} is the FPGA clock frequency in Hertz.

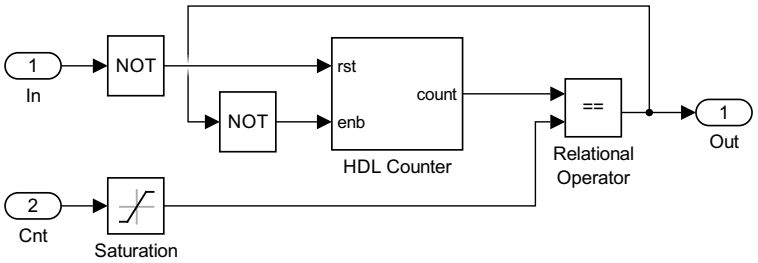


Figure 6.4: Dead time counter

Vivado

The pulse width modulator is packaged into a IP-core and used in a Vivado® block design. Figure 6.5 depicts the interconnections between the controller and the modulator. All other connections are removed for clarity.

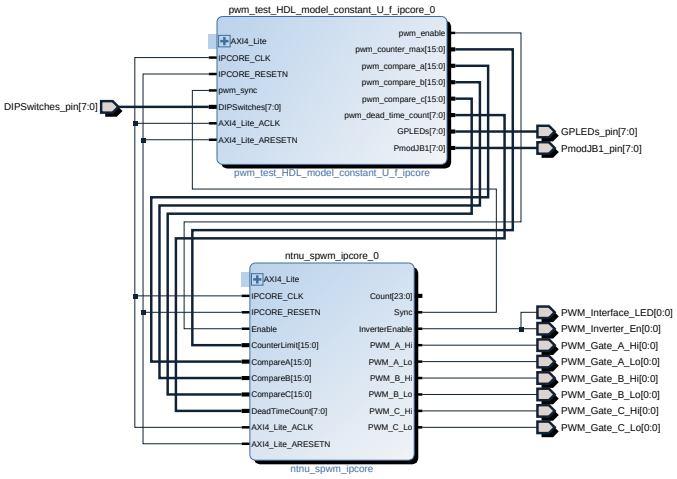


Figure 6.5: PWM IP block interconnections in Vivado

Simulation

In order to confirm the functionality of the symmetrical-PWM, a simulation using a sinusoidal modulating wave has been performed. The PWM triangular carrier, and the sinusoidal modulating wave are depicted in figure 6.6. The output pulses generated is depicted in figure 6.7.

The simulation is executed at a sampling frequency of 25 MHz. In order to clearly illustrate the operation a relatively low switching frequency when compared to the modulating wave is used. The modulating wave has a frequency of 50 Hz, and the switching frequency is set to 1000 Hz by imposing a counter limit of 25000.

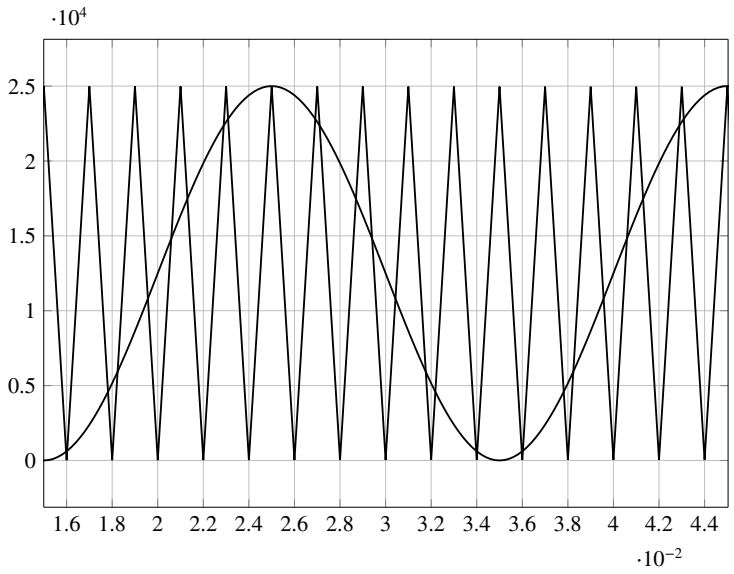


Figure 6.6: PWM triangular carrier and sinusoidal modulating wave.

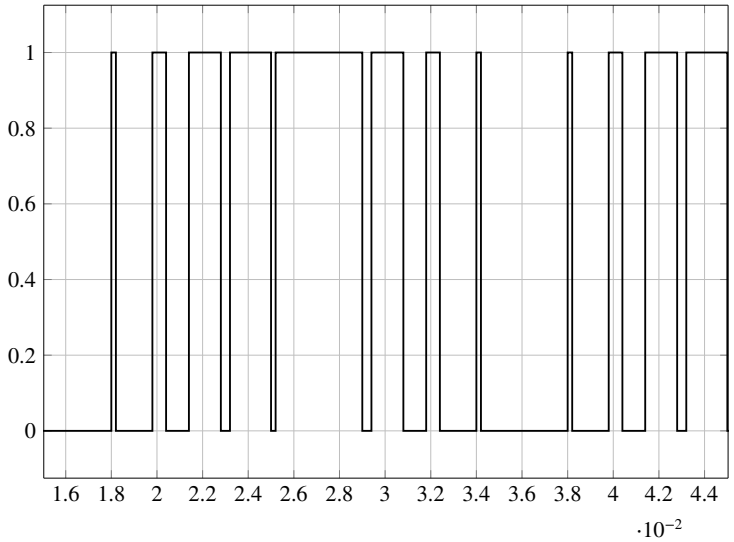


Figure 6.7: Output pulses from the modulator.

6.2 Direct gate control

In addition to the IP-core for PWM, a second simplified IP-core for direct gate control is added. Direct control over the gates is required for some variations of DTC and MPC. The only features of this core is a common enable signal, and dead time counters.

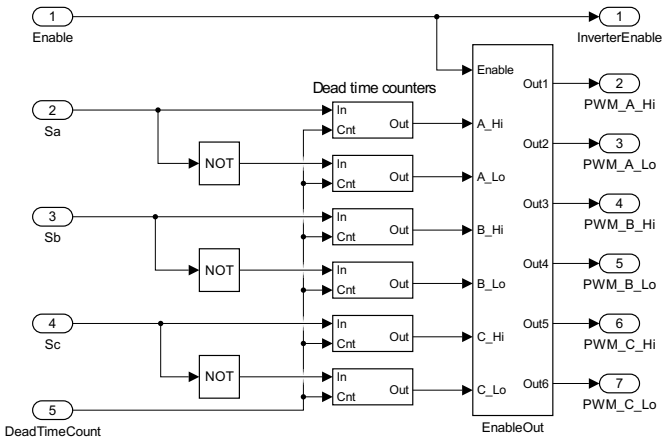


Figure 6.8: Outline of the Simulink® diagram for the direct gate control.

6.3 Encoder demodulation

In order to acquire position, rotational direction, and speed, the encoder signal must be demodulated. The following section covers the various steps involved in this process.

Debounce

When a physical signal is connected to a digital system (e.g. FPGA) with high sampling frequency, is is sometimes required to add a debounce system on the input. The reason is that the external signal might bounce back and fourth before settling on the new state. This may cause the digital system to interpret it as several actuations of the external signal.

In order to ensure that such issues are avoided, a de-bounce counter is added to the inputs from the encoder. Figure 6.9 depicts the Simulink® model of the system. If the input changes from low to high, and stays high for two sampling periods, the exclusive-or gate will release the reset of the counter. The counter starts counting until it reaches a user defied limit, and finally the output is turned on. If on the other hand the input changes during the count period, the exclusive-or will detect this and reset the counter.

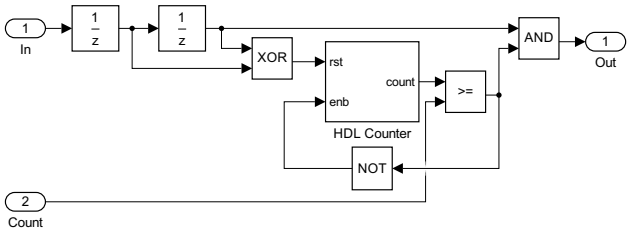


Figure 6.9: Debounce counter

In order to evaluate the necessity of a debounce counter, the pulses from the encoder interface is measured. Figure 6.10 depicts the slope of the voltage from the interface card, when the encoder outputs a pulse. It is observed that the signal has a steep slope, and that it settles in about 3 µs, without any ringing. It is therefore likely that the debounce counter will not be required in this configuration.

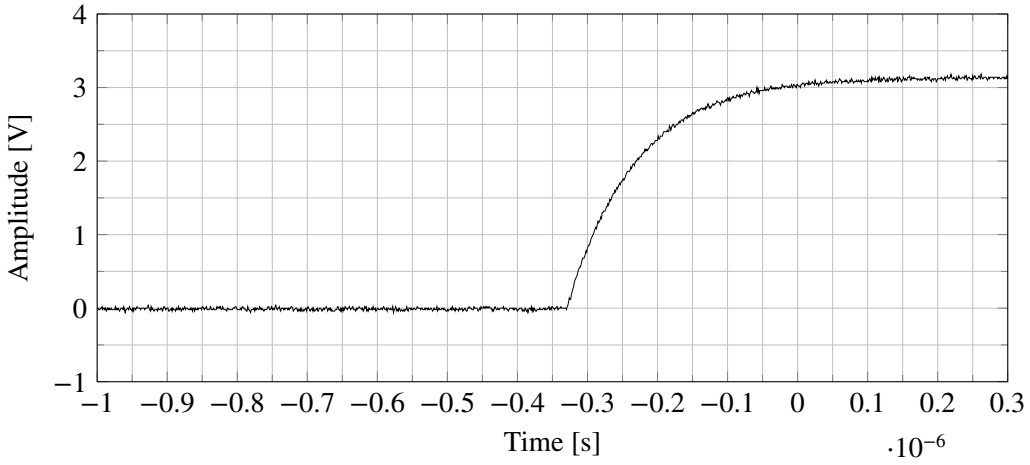


Figure 6.10: Pulse slope of the 3.3 V signal from the encoder interface.

Position counter

Figure 6.11 depicts the system used to detect the rotational direction of the encoder. The signals *A*, and *B* are 90° out of phase, and depending on which comes first the rotational direction is either one or zero. The test is performed on rising edge of signal *A*, and the direction is one if *A* comes before *B*.

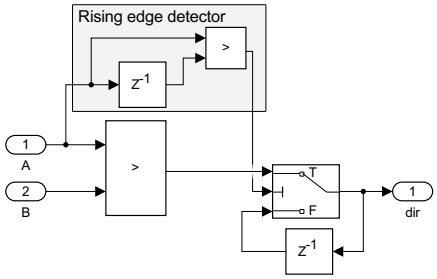


Figure 6.11: Rotational direction detector.

The encoder reset subsystem is depicted in figure 6.12. Depending on the rotational direction the counter is reset on zero, or on the counter limit.

Figure 6.13 depicts the position counter for the encoder. The direction input *dir* determines the direction of the counter, while the clock input *clk* triggers the counting. Depending on the rotational direction, the counter is either reset to zero, or to the maximum value of the encoder output, i.e. *cntPerRev*.

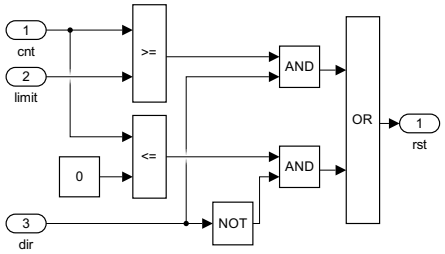


Figure 6.12: Encoder reset subsystem.

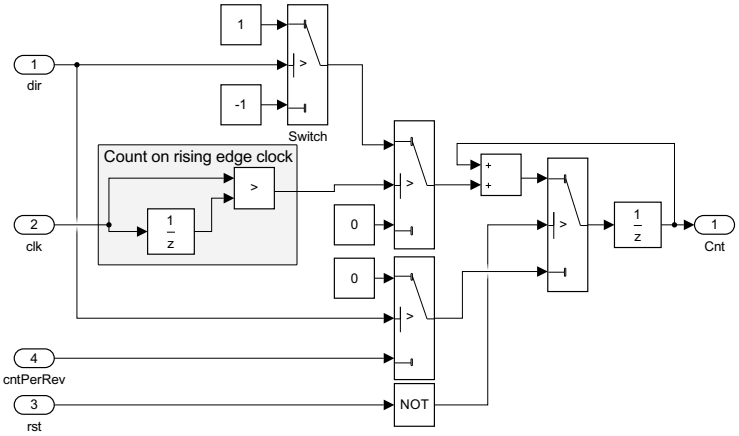


Figure 6.13: Encoder position counter.

The response of the encoder counter to a reversal of the rotational direction is depicted in figure 6.14. In the simulation a encoder with 32 pulses per revolution is used to clearly visualize the response.

Position differentiator

Figure 6.15 depicts the differentiator used to compute the angular velocity from the rate of change of the rotor position. The position delta is computed as the difference between the position at the current, and the previous sample. If the difference is to large, it means that the counter just reset, and the value is disregarded. A 10-sample buffer (figure 6.16) is used to sum the delta over a longer time period, in order to produce a less noisy signal.

The output is correctly scaled by multiplying by the sampling frequency over buffer length (6.3). Finally the output is scaled to radians per seconds by multiplying with (6.4).

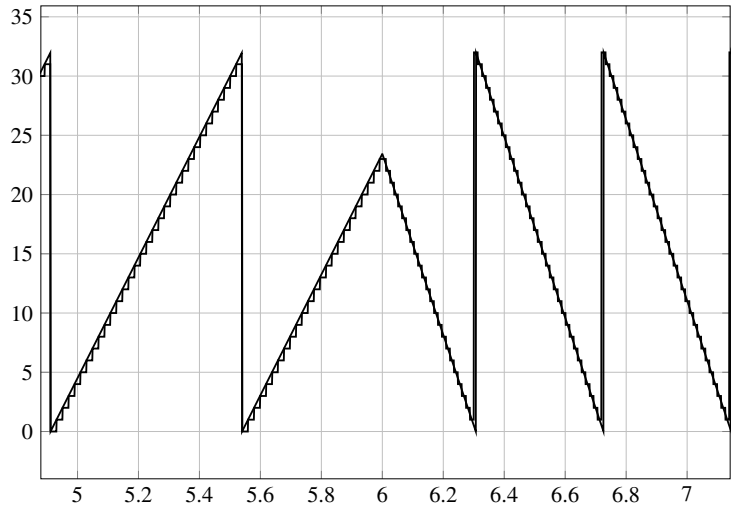


Figure 6.14: Simulation of the encoder position counter, for a reversal in the rotational direction.

$$k_{scale} = \frac{f_{samp}}{10} \tag{6.3}$$

$$k_{scale,rad} = \frac{2 \cdot \pi}{\text{MaxCount}} \tag{6.4}$$

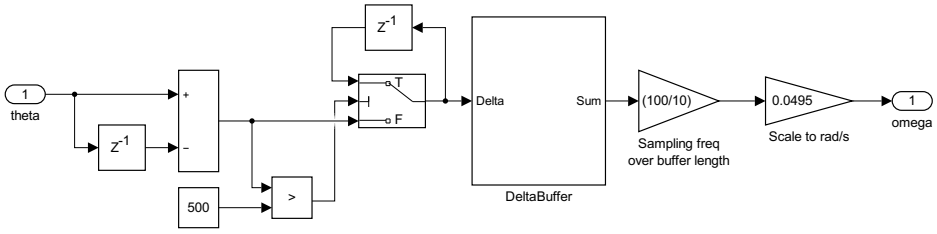


Figure 6.15: Simulink algorithm for computation of angular velocity.

The delta buffer is depicted in figure 6.16. It works by summing the position delta at the current iteration, with the deltas at the nine previous iterations.

In order to confirm the operation of the differentiator, a simulation was performed using the output from the position counter. The results are depicted in figure 6.17. The speed comes from a ideal source, and thus changes instantaneously. The gray line depict the real, while the black line depicts the computed speed. Simulation is performed at a sampling rate of 100 Hz for the differentiator, while the position

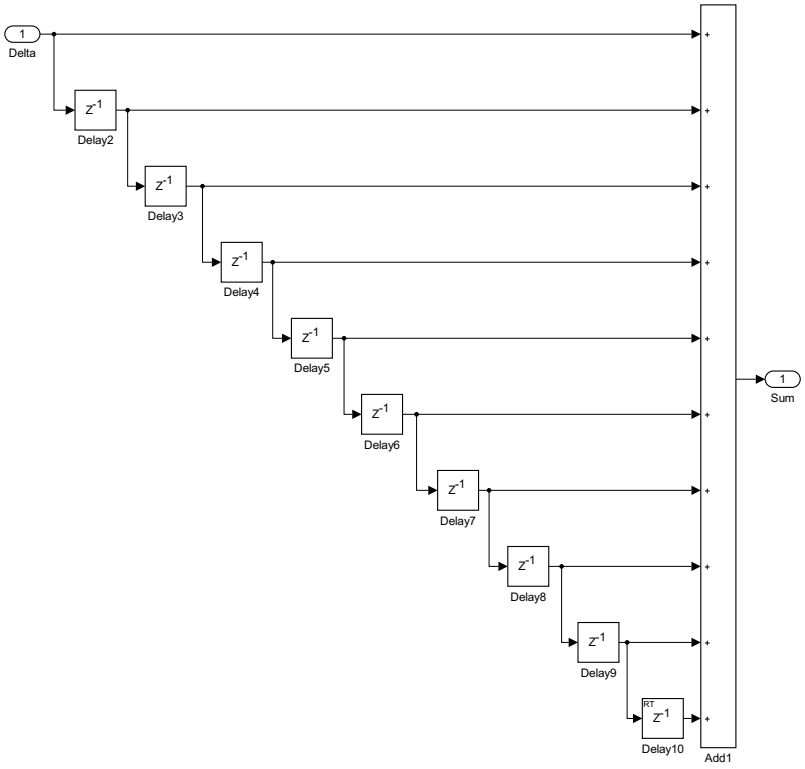


Figure 6.16: Delta buffer.

counter runs at 100 kHz. The low sampling frequency of the position differentiator indicates that is is suited for implementation in the PS. However increased sampling rate might be required depending on the dynamic requirements of the application.

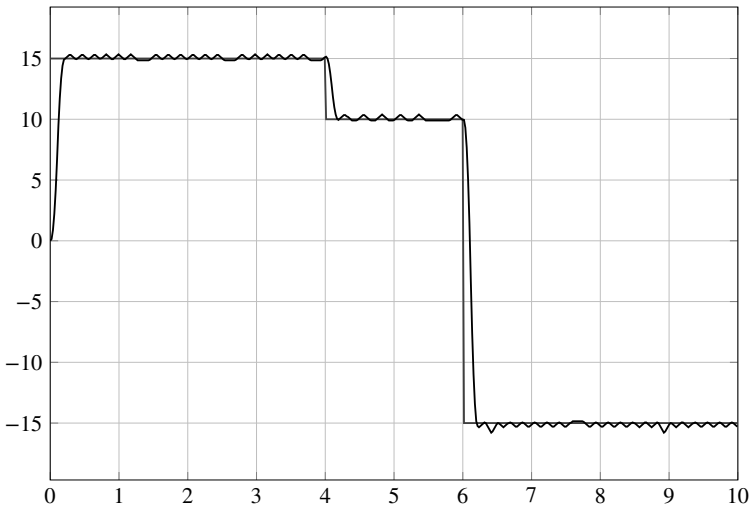


Figure 6.17: Simulation of the encoder position differentiator.

Chapter 7

Start-up routines

Before the motor drive goes into normal operation some start-up routines are performed.

7.1 Rotor alignment

In order to calibrate the encoder and position estimators, the rotor should initially be aligned to a known position. For a PMSM this is simply achieved by applying a fixed voltage vector to phase *a*.

Once the alignment is completed this position is considered the zero position. A offset is added to the demodulated signal from the encoder, in order to compensate for any deviation between the zero position and the index pulse position.

The per-unit voltage reference to the modulator is computed based on equation (7.1).

$$\begin{bmatrix} v_a \\ v_b \\ v_c \end{bmatrix} = \begin{bmatrix} 1 \\ -\frac{1}{2} \\ -\frac{1}{2} \end{bmatrix} V_{pu} \tag{7.1}$$

Where V_{pu} is the per-unit voltage, with respect to the DC-link voltage. This reference may be adapted depending on the required torque.

It should be noted that, depending on the application, this alignment might not be applicable. Some motor applications require control over the rotor position at all

times. In this project however, there are no specific constraints on the mechanical side of the system.

7.2 Pre magnetization

When running FOC of induction machines, the rotor is usually pre magnetized in order for the flux estimator to be able to obtain a initial estimate. For a PMSM however, the magnetic field is always available and no pre magnetization is needed.

A simple pre magnetization scheme has been implemented in order to offer preliminary support for induction machines. The magnetization is achieved by switching the voltage on phase b, while phase a, and c are connected to the positive side of the DC-link voltage.

7.3 Open loop velocity control

There is a fixed relationship between the applied electrical frequency to the stator, and the achieved mechanical speed of the rotor. This may be utilized in a open loop speed control of the machine.

This is common for simple induction motor drives, where the constant voltage over frequency principle is applied. For induction machines there will be slight speed variations depending on the load, i.e. the slip frequency will change. For synchronous machines however, the speed remains constant unless the machine is pulled out of synchronism.

This lack of damping torque during deviations from synchronous speed may cause stability issues when open loop control of a PMSM is considered. In order to overcome this issue the rotor of some machines contain damper windings, in a similar manner to the squirrel cage of the induction machine [46].

Generation of reference voltages

In order to generate the three phase sinusoidal voltage references to the motor, a ramp generator with adjustable frequency is used. The frequency is adjusted depending on the required speed, and the *sine* of the ramp function corresponds to the required stator voltage waveform. Two additional sine waves which are $\pm 120^\circ$ phase shifted are also generated.

The frequency of the ramp is adjusted by adjusting the clock input. This approach ensures that the bit resolution remains constant, regardless of the desired ramp frequency. The required count limit for a given modulation frequency is given as:

$$n_{count} = \frac{f_{clock}}{n_{step} \cdot f_{mod}} \tag{7.2}$$

Where f_{clock} is the clock frequency, n_{step} is the step size of the fractional counter, and f_{mod} is the desired modulating frequency.

The selected fractional counter has a step size of $\frac{1}{2048}$, i.e. the resolution is 2048 bits. In order to generate a 50 Hz sine wave, for the selected FPGA clock frequency of 25 MHz, the counter limit is given as:

$$n_{count} = \frac{25e6}{2048 \cdot 50} \approx 244$$

Figure 7.1 depicts a implementation of a three phase sine wave generator, that may be used to generate synthesizable HDL code.

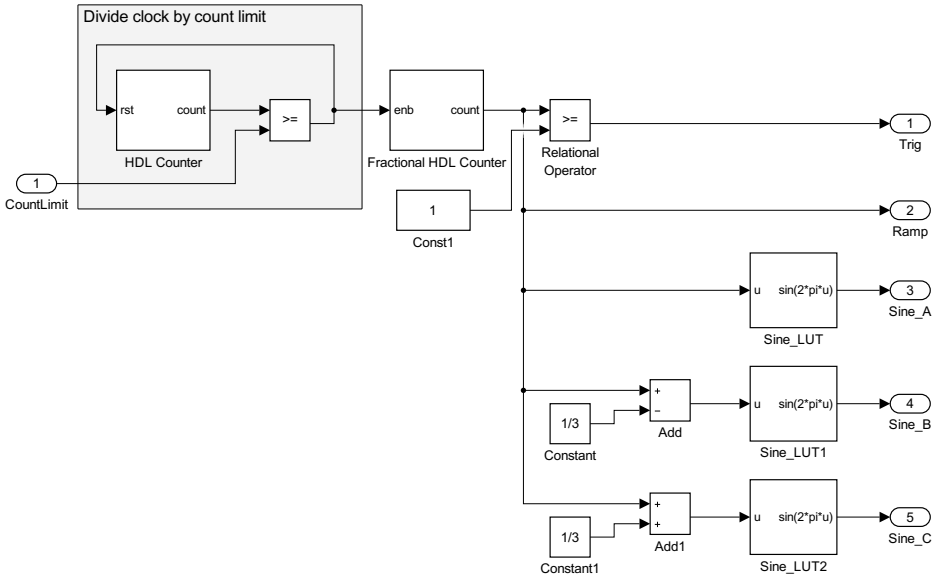


Figure 7.1: HDL synthesizable implementation of a 3 phase sine wave generator

Voltage reference

In order to obtain constant flux linkage the voltage is typically controlled in proportion to the speed.

The reactance of the machine is proportional to the applied frequency. In low speed region, the voltage drop in the winding resistance cannot be neglected, and thus the voltage needs to be boosted.

Figure 7.2 provides a example of a how the voltage is adjusted with changing frequency. The rated voltage is 230 V, and the rated frequency is 50 Hz.

In practice a simple way to implement this is by applying a upper and lower saturation on the following equation:

$$V_{out} = \frac{V_N}{f_N} f_{out} \tag{7.3}$$

Where V_N and f_N are the rated motor voltage and frequency respectively.

There are however more advanced approaches at controlling the voltage boost.

Current based voltage boost

One of the more advanced schemes is to control the voltage boost through the measured stator current [46]. Although the measured current is required as feedback, this is still a simple controller when compared to the vector control.

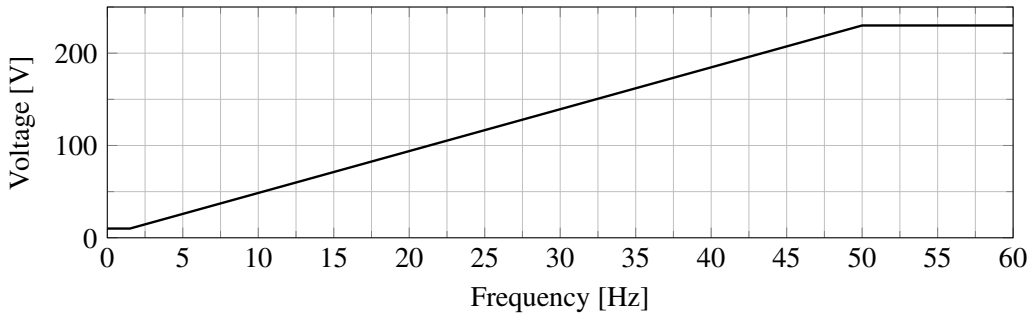


Figure 7.2: Voltage versus frequency with initial voltage boost. $U_N = 230\,\text{V}$, and $f_N = 50\,\text{Hz}$.

Additionally if the PMSM lacks damper windings, damping may be added by adjusting the frequency reference based on the applied power [46]. The power is in turn calculated by the measured current, and the voltage reference.

Part III

Simulation

Chapter 8

Simulation setup

The simulation of the estimators covered in this report are preformed using MATLAB™/ Simulink®. In order to simulate the various estimators a standard FOC inverter was selected to control the motor.

8.1 Speed reference profile

In order for the results to be easily comparable, the simulations should be subject to the same operating conditions. The speed reference profile depicted in figure 8.1 has been used in all simulations.

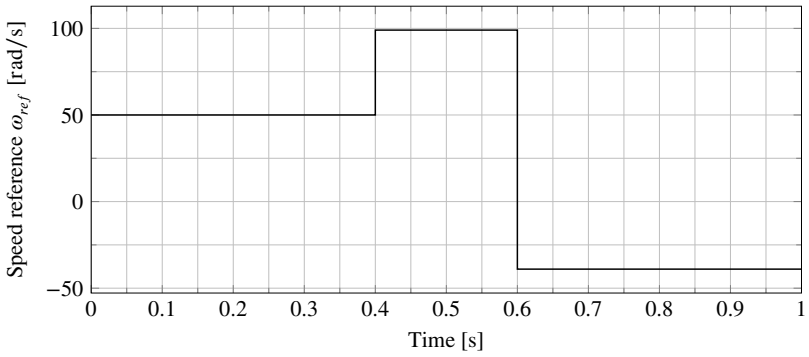


Figure 8.1: Profile of the speed reference used during simulations.

8.2 Motor model

The motor model is derived from the theory presented in chapter 1. It consist of a electrical and a mechanical subsystem.

Figure 8.2 provides a overview of the models, and their interconnections. Further the electrical and mechanical subsystems are provided in 8.4 and 8.3 respectively. The state space model in dq-frame is provided in figure 8.5.

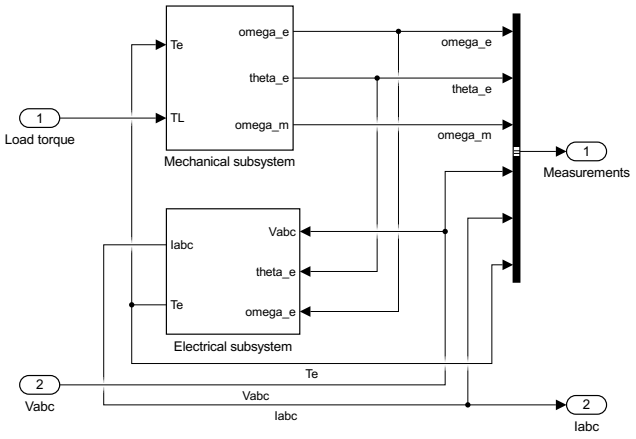


Figure 8.2: IPMSM model overview.

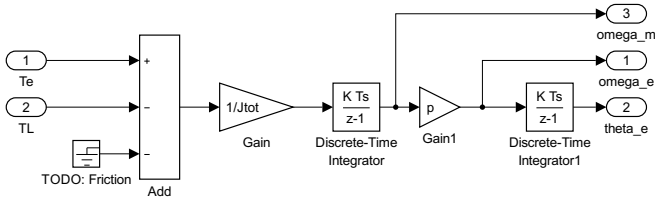


Figure 8.3: Mechanical subsystem

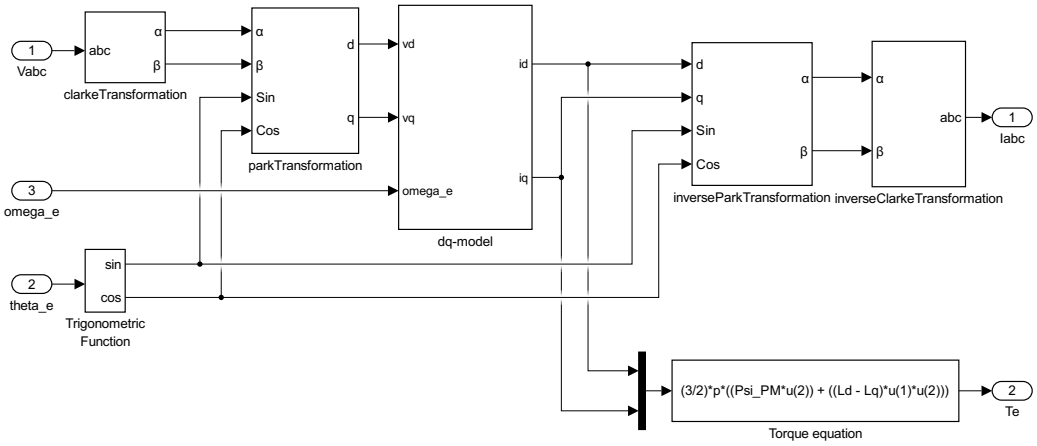


Figure 8.4: Electrical subsystem.

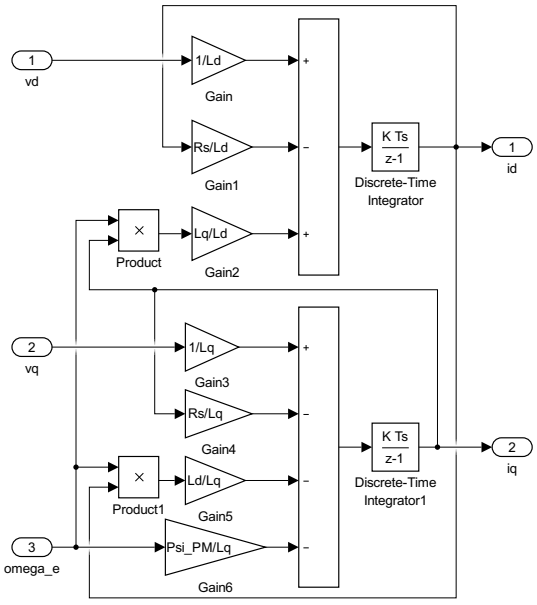


Figure 8.5: IPMSM electrical dq-model

Chapter 9

Simulation of field oriented control

This chapter presents the simulations for the FOC as presented in chapter 2.1. This controller is also used as the base when simulating the various position and speed estimators.

9.1 Field oriented control simulation model

The FOC inverter simulation model is somewhat different from the HDL synthesizable model.

The triangular wave carrier is generated by integrating the difference between two pulse trains, as depicted in figure 9.1.

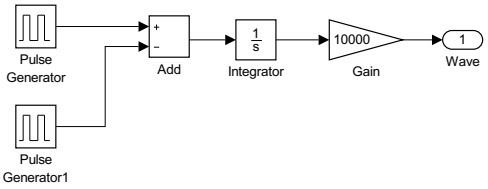


Figure 9.1: Triangular wave carrier generator

The SV-PWM is performed by the diagram in figure 9.2.

In figure 9.3 the compare values are compared to the triangular wave of figure 9.1, and output pulses are generated.

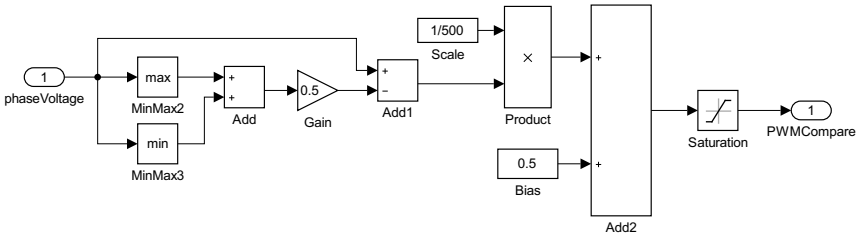


Figure 9.2: Calculate compare values for modulator

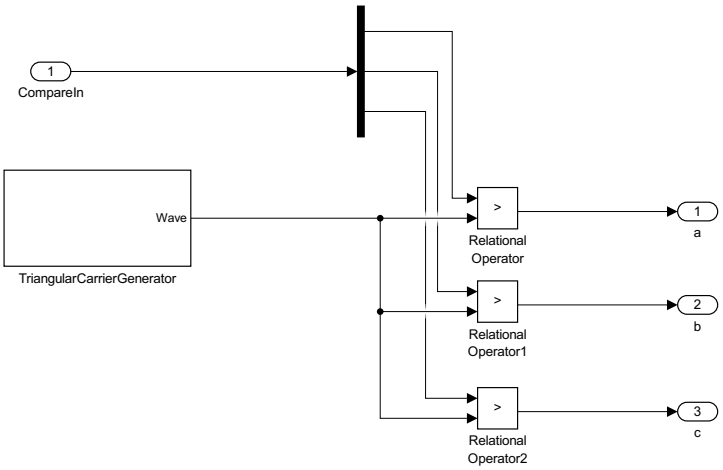


Figure 9.3: Symmetrical pulse width modulator

Figure 9.4 provides a overview of how these systems are interconnected. In addition to the gate pulses, the phase voltage reference is made available in order to avoid the need for a voltage sensor.

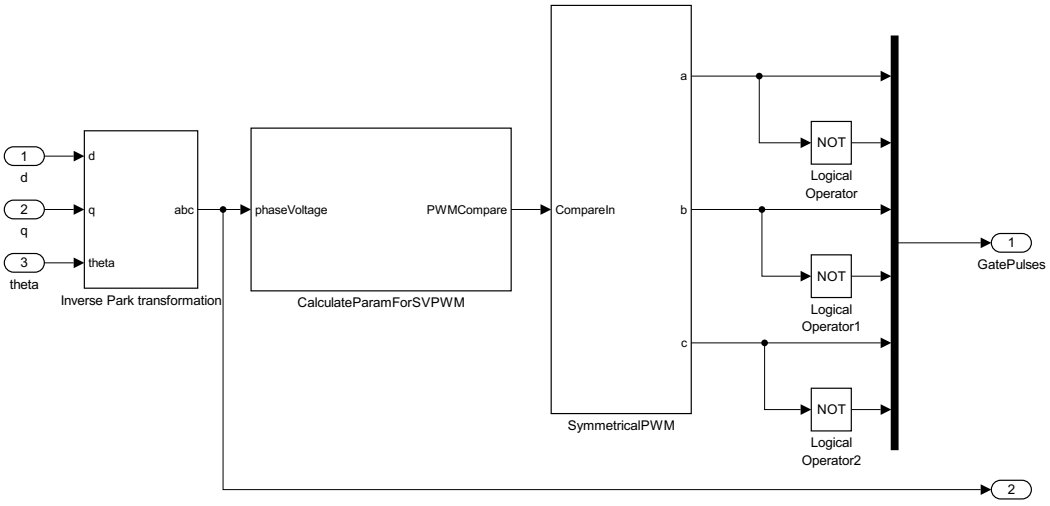


Figure 9.4: SV-PWM controller

9.2 Speed controller response

Figure 9.5 depict the response of the speed control to various changes in reference. The dashed line is the reference, while the solid line is the response. It is observed that the controller is able to achieve fast tracking of the reference with relatively small overshoots.

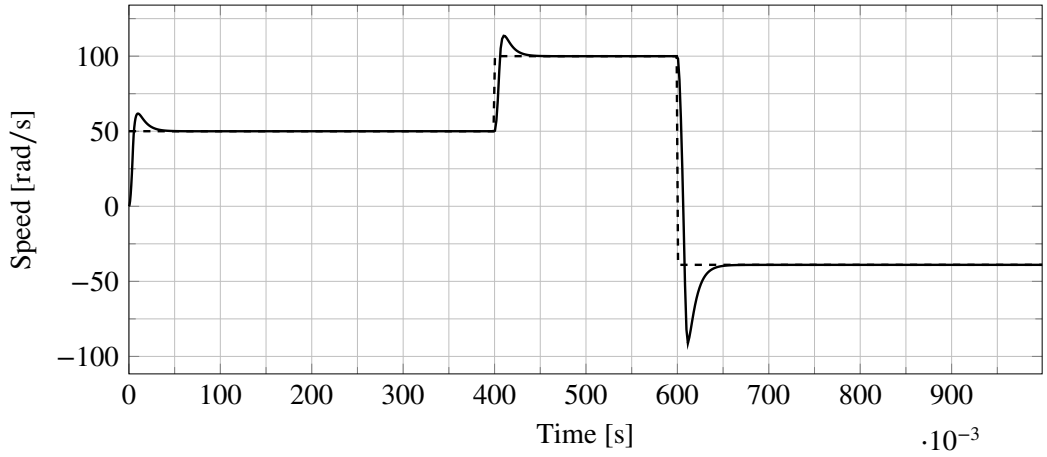


Figure 9.5: Response of the FOC speed control, to various changes in reference.

Figure 9.6 depicts the response of the q-axis current controller to the commands from the speed controller. The dashed line is the commanded current, while the solid line is the output, i.e. the voltage reference. It is observed that the controller

saturates for a small period during the transients, but also that is quickly settles at the required steady state value.

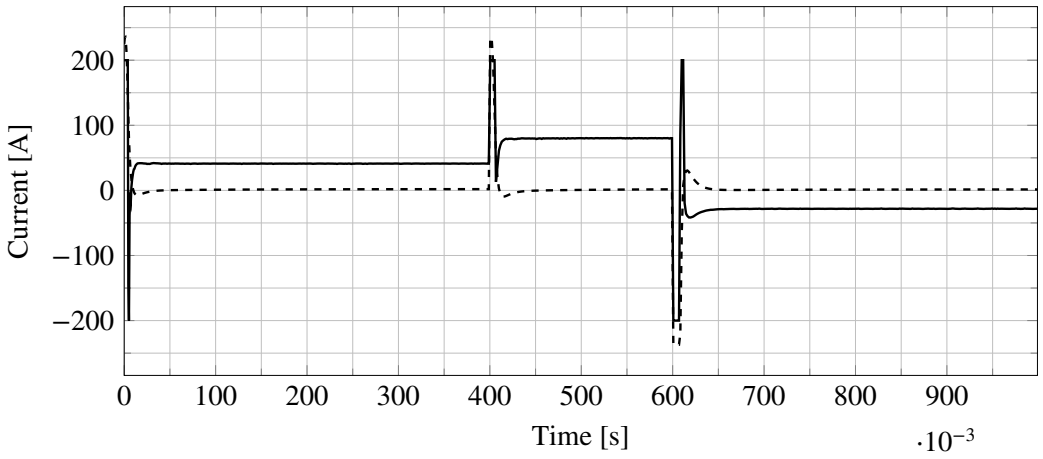


Figure 9.6: Response of the FOC q-axis current controller, to the commands from the speed controller.

Chapter 10

Simulation of direct torque control

This chapter develops the Simulink®-model, and presents the results of the simulation of the DTC algorithm presented in chapter 2.2.

10.1 Simulink model

Figure 10.1 provides a overview of the implemented DTC model. The configurations of the various subsystems are provided in the following sections.

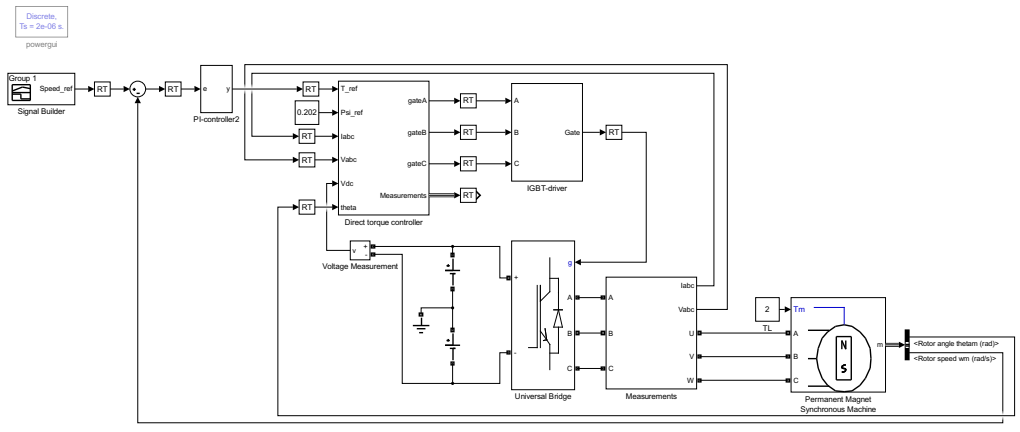


Figure 10.1: Overview of the implemented DTC model

Flux sector

In order to determine the correct flux sector for the current iteration, the *arctangent* of the estimated $\alpha\beta$-flux is typically fist applied to find the flux angle. Comparators are then utilized to obtain sector information for the lookup tables.

$$\hat{\theta} = \arctan 2(\hat{\Psi}_\alpha, \hat{\Psi}_\beta) \tag{10.1}$$

Trigonometric functions are computationally intensive, and in order to save some execution time a different approach is proposed in [55]. Figure 10.2 depict a implementation of the sector selector in Simulink®.

The truth table for the combinatorial logic is given as:

$$[5; 4; 6; 0; 0; 3; 1; 2] \tag{10.2}$$

Where the row corresponding to the value of the input vector is selected as output.

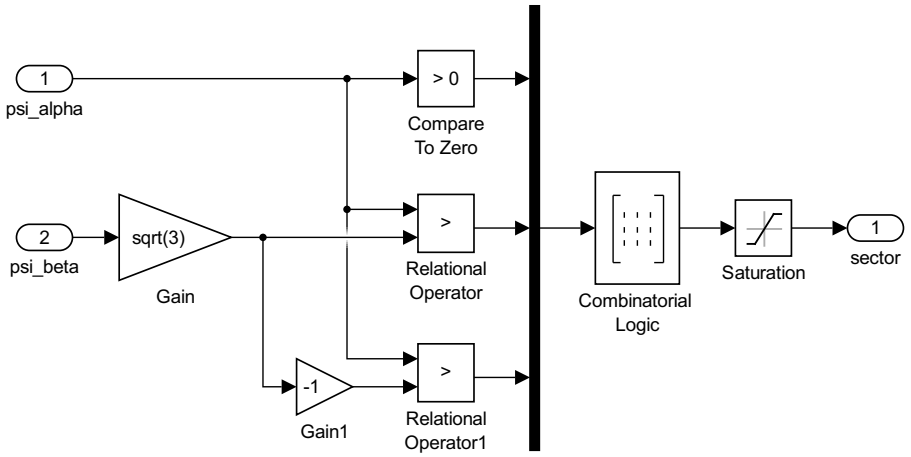


Figure 10.2: Modified flux to sector calculator

The Simulink® implementation of the lookup table for the DTC is depicted in figure 10.3. Two different lookup tables are used, one for increasing and one for decreasing flux. Eight voltage vectors are defined and selected depending on which sector and torque command that is applied to the lookup tables. An additional rotor align switch is also available. It is used to align the rotor in an initial position during start-up.

A current model based flux estimator is depicted in figure 10.4. This model requires the use of a encoder for position measurement. The model provides very

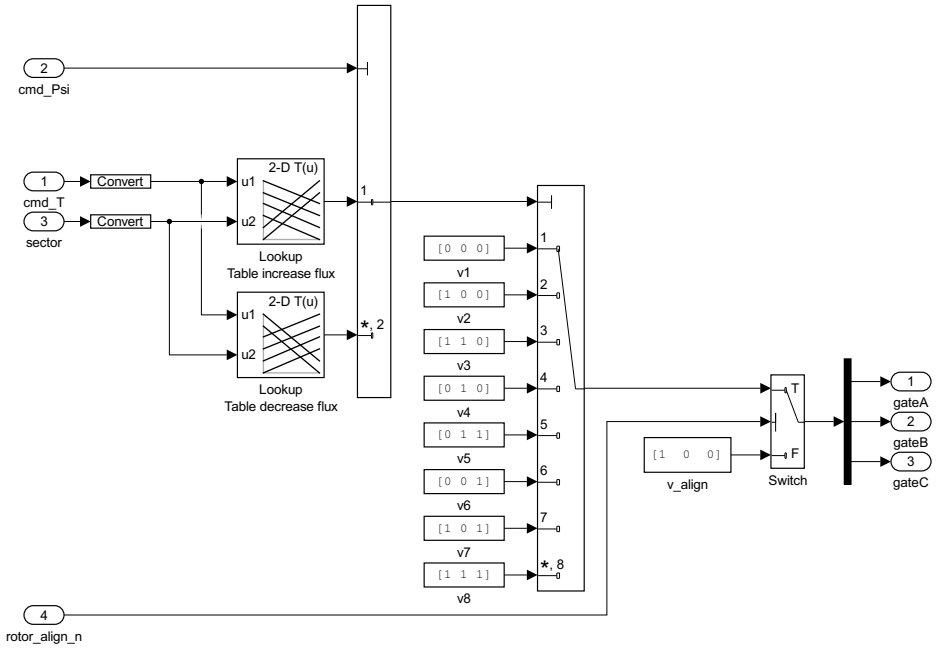


Figure 10.3: DTC lookup table

good estimation during simulation (figure 10.7), however the requirement of exact knowledge of machine parameters will likely degrade the performance during laboratory experiments.

A alternative sensorless approach is depicted in figure 10.5. This model is less dependent on knowledge of machine parameters, but the open-loop integrator will likely cause drift in the estimated values.

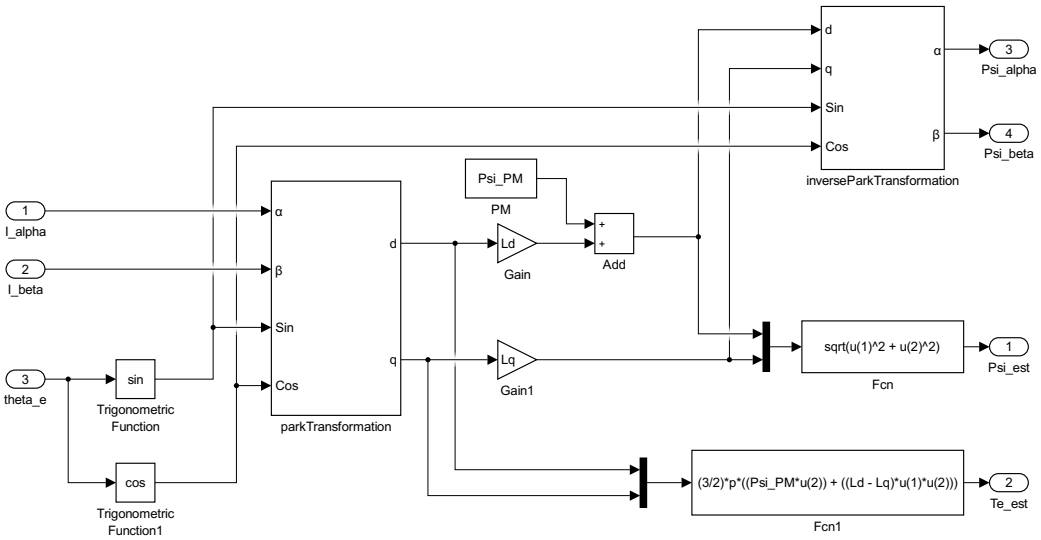


Figure 10.4: Current model based flux and torque estimator.

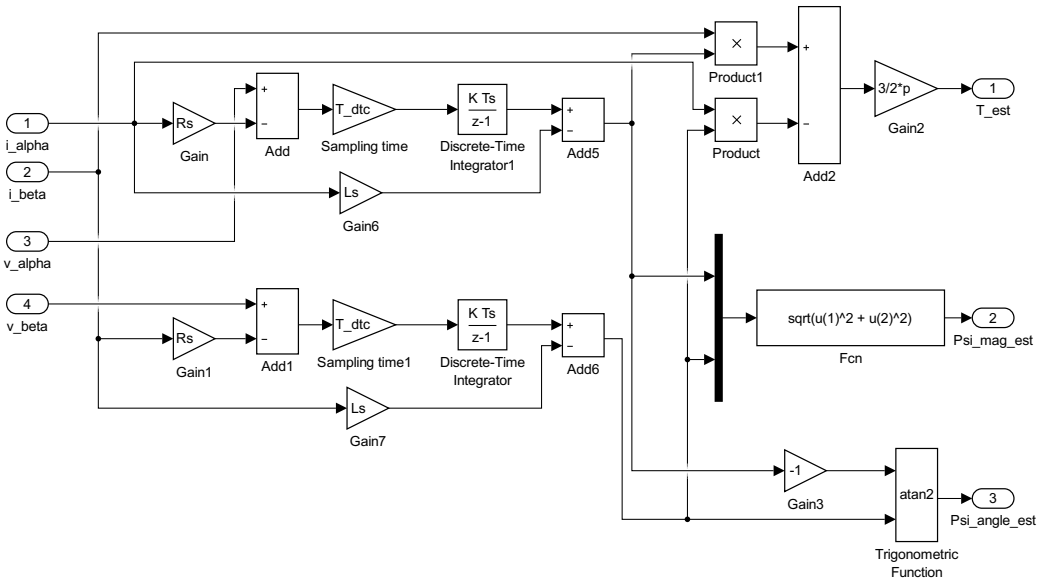


Figure 10.5: Voltage model based flux and torque estimator.

10.2 Torque response

Figure 10.6 depicts the response of the torque hysteresis controller to variatons in estimated torque. The torque reference is constant, $T_{ref} = 5\,\mathrm{N\,m}$. It is observed that although the torque has a lot of ripple, the average value is close to the reference. The hysteresis controller switches between 1 and 2, depending on whether the estimation is above or below the reference.

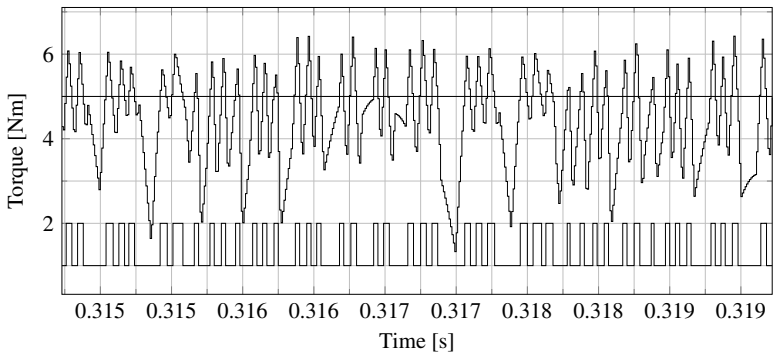


Figure 10.6: Response of torque controller to variations in estimated torque.

Figure 10.7 depicts the performance of the torque estimator, when compared to the measured torque. The solid line is the estimation, whereas the dashed line is the measured torque. It is observed that the estimation is able to follow the measurement, but that there is a delay due to the limited sampling time of the controller. Due to the discretization, there is also somewhat lower resolution in the estimated values.

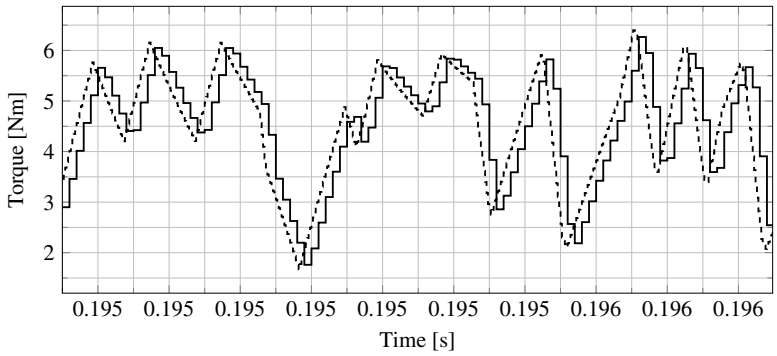


Figure 10.7: Performance of the current model based torque estimator, when compared to the measured torque.

Figure 10.8 depicts the performance of the flux controller. The flux reference is 0.202 Wb, which is the rated flux of the machine. It is observed that the controller is able to track the reference, although there is some ripple.

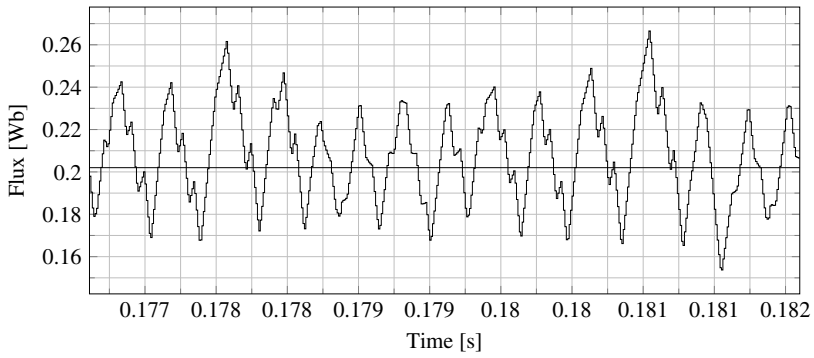


Figure 10.8: Performance of the flux controller.

Chapter 11

Simulation of model predictive control

This chapter presents the implementation, and simulation of the MPC-algorithm as presented in chapter 2.3. The implementation utilizes measured rotor position in order for it to be comparable to the other implementations.

Although model predictive speed controllers are available, this implementation only consider the currents, and utilizes a traditional PI-controller for the speed.

11.1 Algorithm implementation

The Matlab™- source for the MPC-algorithm is provided in appendix E.2, and E.3, for initialization, and control respectively.

The initialization(E.2) consists of defining the motor parameters, and the possible output voltage vectors in $\alpha\beta$ plane. Finally the switching states that corresponds to the output vectors are defined.

The control algorithm (E.3) accept the measured d, and q-axis currents, as well as their references, the rotor position and rotor speed as inputs. The outputs are the switching state to the inverter.

The voltage vectors defined in the initialization procedure are rotated to the dq-frame, by applying the Park transformation. Then the currents for the next iteration are predicted by applying the equations defined in (2.30a) and (2.30b).

The for-loop iterates over the possible output vectors to determine the vector that will minimize the cost function for the current errors. The most suited vector is selected, and the function returns.

11.2 Block diagram

Figure 11.1 provides an overview of the simulation block diagram implemented in Simulink®, while figure 11.2 depicts the interface to the controller.

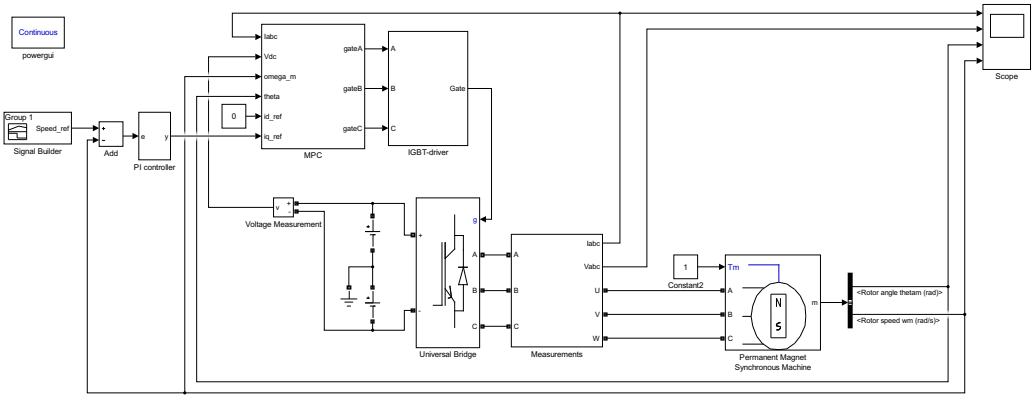


Figure 11.1: Overview of the MPC simulation model.

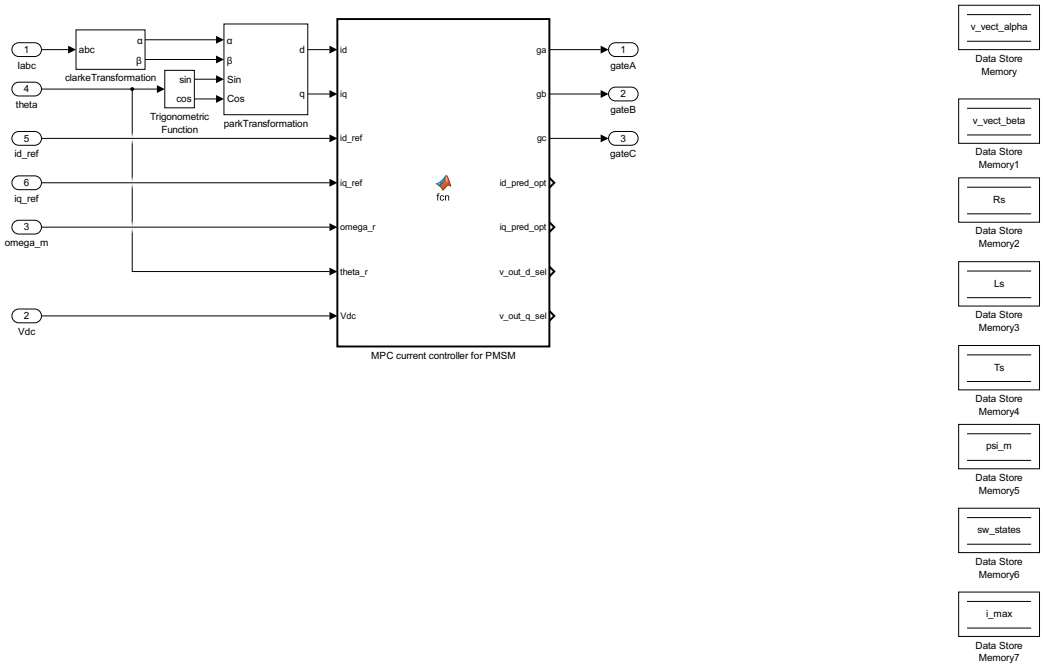


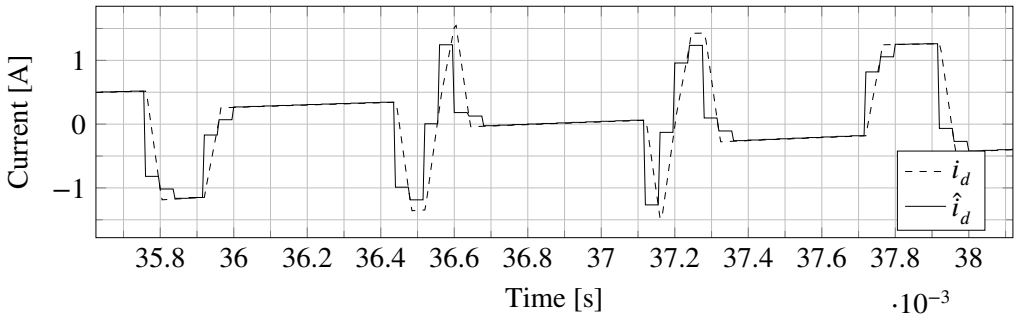
Figure 11.2: MPC interface to the controller

11.3 Current controller response

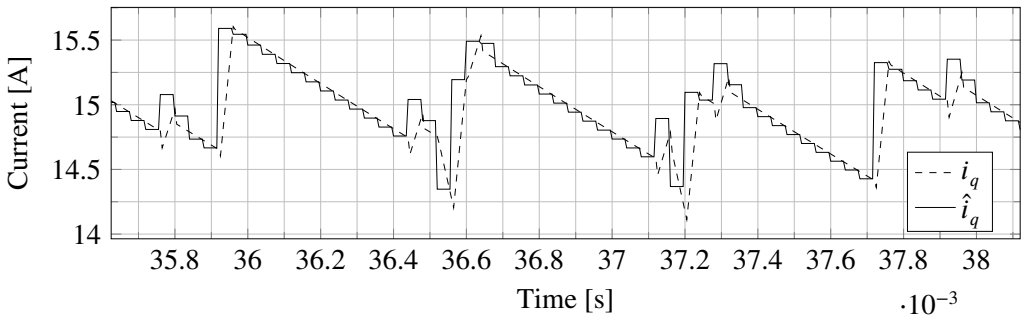
Figure 11.3 depicts how the MPC current controller is able to predict the current, and how the current follows the reference. A relatively short time period is used to clearly visualize the response. In the figure the current references $i_{d,ref} = 0\,\text{A}$, and $i_{q,ref} = 15\,\text{A}$. The dashed line is the measured current, while the solid line is the predicted current at the next iteration.

It is observed that the measured current follows the predicted current, with a small phase delay. This is expected as the prediction is of the future current response. It is also observed that the current controller is able to closely follow the reference, although there is some ripple, the average is close to the reference. The discrete nature of the MPC gives rise to the discrete steps in the prediction.

In the simulation the controller is running at a sampling time of 40 µs, which should provide a real time controller with sufficient time to compute all the predictions.



(a) Measured and predicted d-axis current



(b) Measured and predicted q-axis current

Figure 11.3: The measured and predicted dq-axis currents. $i_{d,ref} = 0$ A, $i_{q,ref} = 15$ A.

Chapter 12

Simulation of speed and position estimators

This chapter provides the results for the simulations of the back-EMF observers, performed in MATLAB™/Simulink®. The position and speed is extracted from the estimated back-EMF, and the performance is evaluated by comparison to the real values.

In order to have a fixed reference for comparison, a traditional FOC inverter model has been utilized. The inverter uses the real motor shaft position as feedback, i.e. the position is extracted directly from the motor model state variables.

12.1 Sliding mode observer

A simple sliding mode observer for a PMSM is proposed in [38]. The observer has been implemented in Simulink®, as depicted in figure 12.1.

The estimated back-EMF of the simple sliding mode observer is depicted in figure 12.2. As can be seen the curve shapes are clearly not sinusoidal, as they should be. Furthermore the locus of the EMF is depicted in figure 12.3. The expected locus shape is round, which is not the case for this observer. The performance could be improved by adding additional filtering at the output, however this will increase the complexity, and cause additional phase delay in the output.

The difference between actual, and estimated position is depicted in figure 12.5. The bias of the estimation error could likely be removed by an adaptive compen-

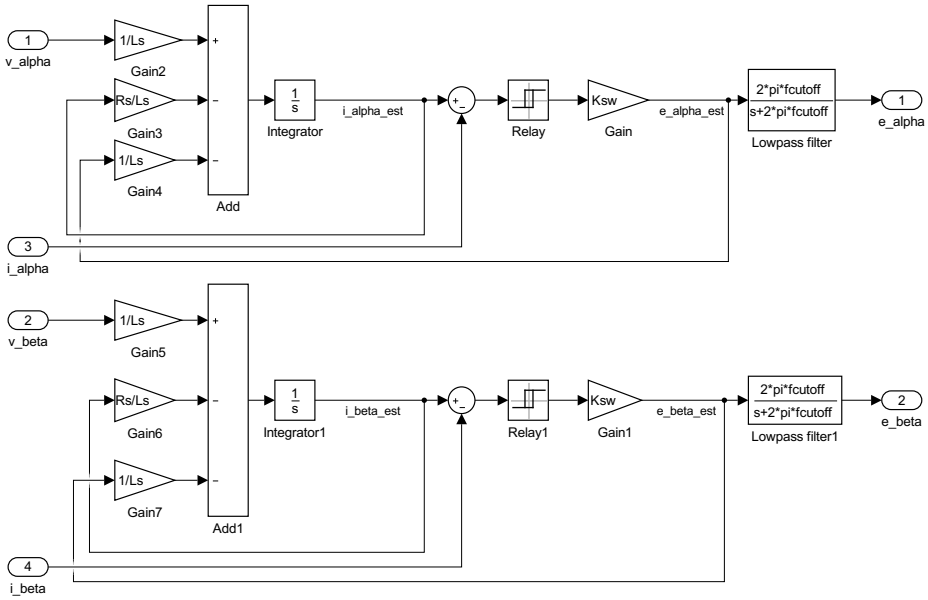
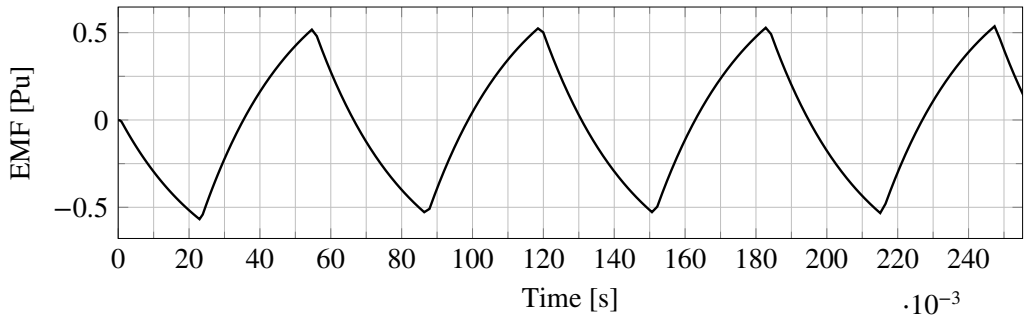
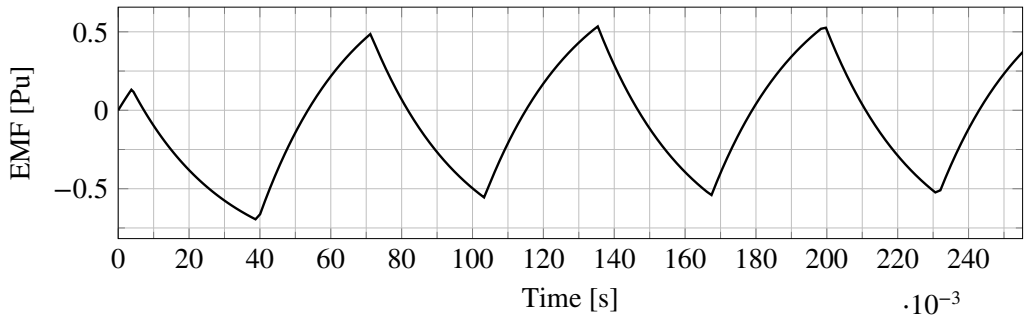


Figure 12.1: Simulink® model of the simple sliding mode observer

sation depending on the speed. This has been implemented for the more advanced algorithm in section 12.3.



(a) Alpha axis



(b) Beta axis

Figure 12.2: The estimated back-EMF of the simple SMO with a first order low-pass filter.

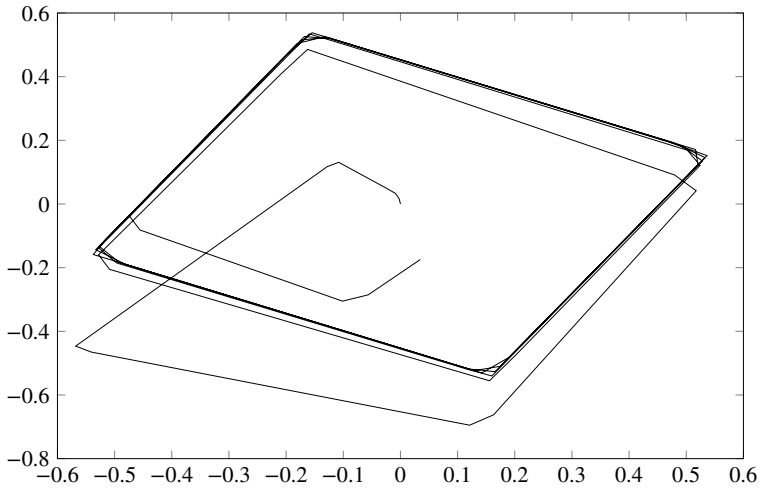


Figure 12.3: Locus of the estimated back-EMF of the simple SMO

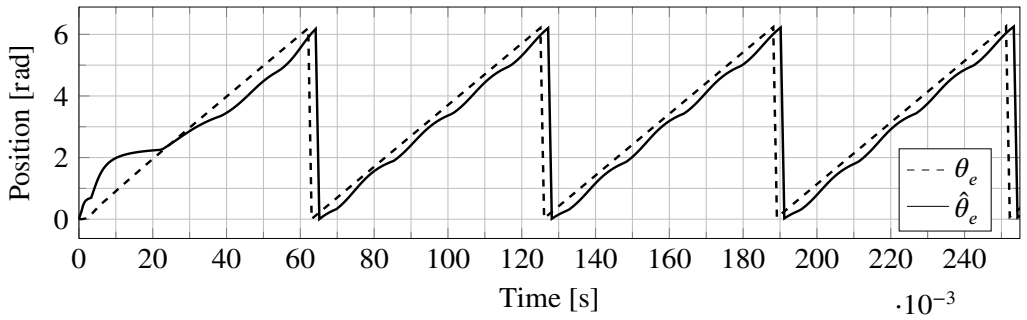


Figure 12.4: The estimated rotor angle obtained from the simple SMO. Estimated at $\omega_{e,ref}$ = 50 rad/s

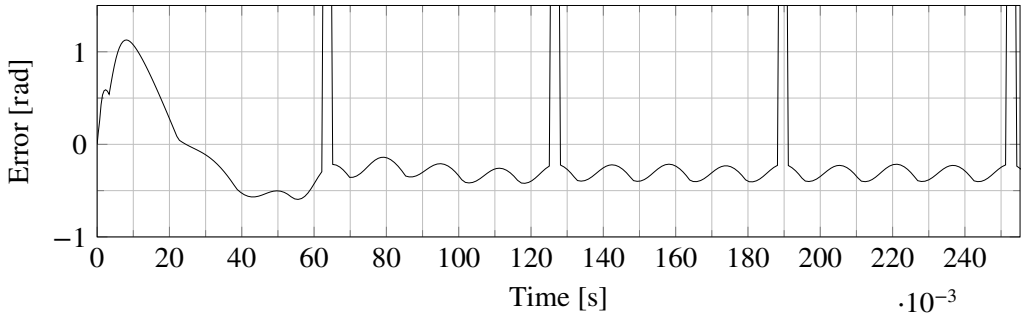


Figure 12.5: The estimation error of the simple SMO.

12.2 Sigmoid sliding mode observer

The estimated back-EMF of the sigmoid based SMO is depicted in 12.7. The curve shapes are clearly improved over the simple SMO, the phase shift is 90° as expected. The improvement is further confirmed by the locus plot in figure 12.8, which has a circular shape.

The estimated rotor position from the *sigmoid* SMO is depicted in figure 12.9a, with the real position in figure 12.9b for comparison. Apart from the initial state, it is observed an almost perfect match between the estimation, and the real value. This is further confirmed by figure 12.9c, which depicts a estimation error which on average is less than 0.25%. It is however important to note that these are simulation results. Due to factors that are neglected in the model, it will likely be hard to realize the same performance in laboratory experiments.

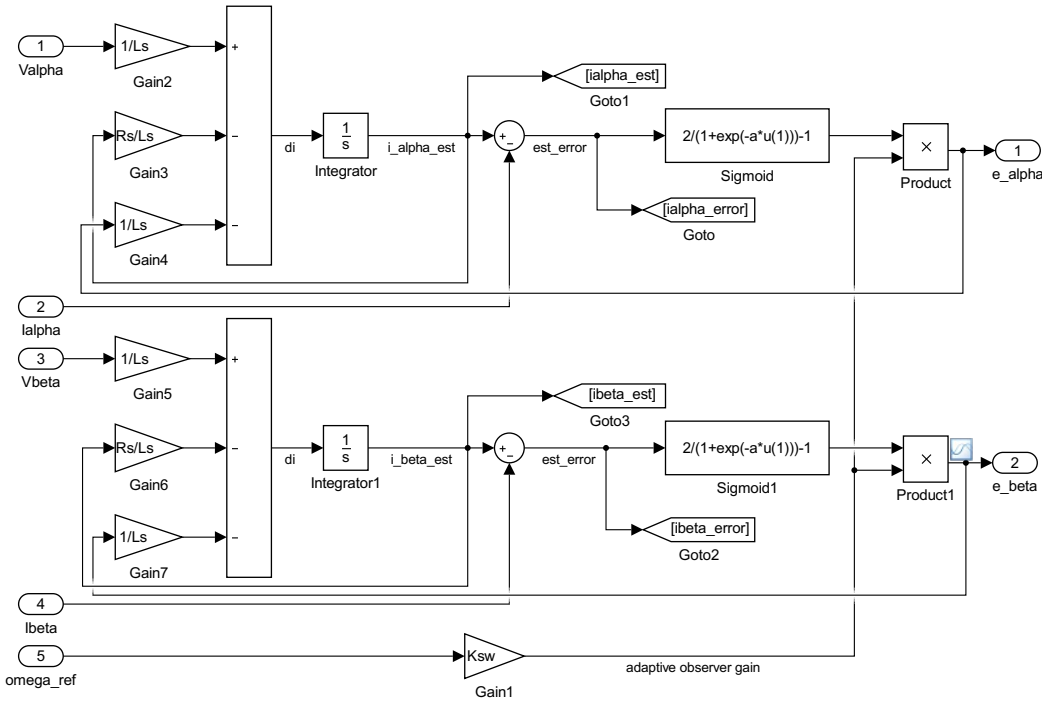
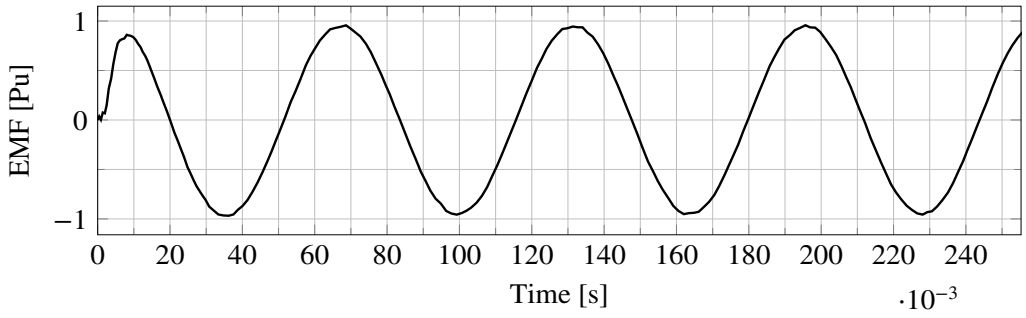
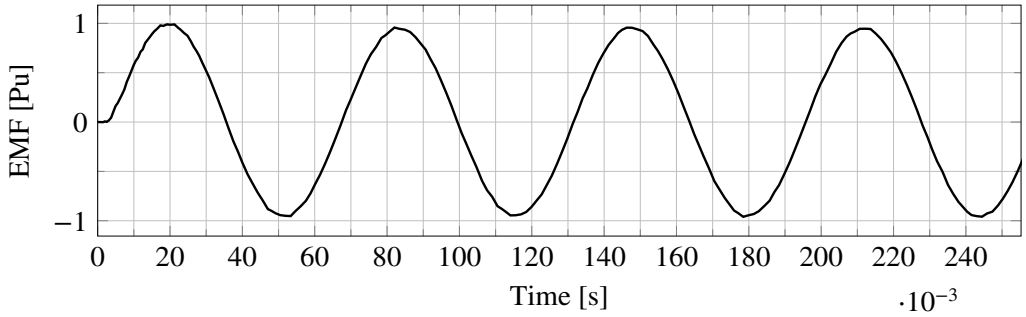


Figure 12.6: Simulink® model of the sigmoid sliding mode observer



(a) Alpha axis



(b) Beta axis

Figure 12.7: The estimated back-EMF of the sigmoid SMO

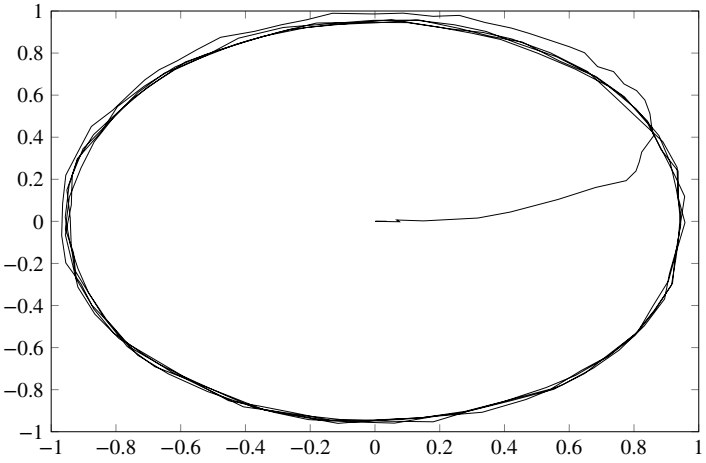
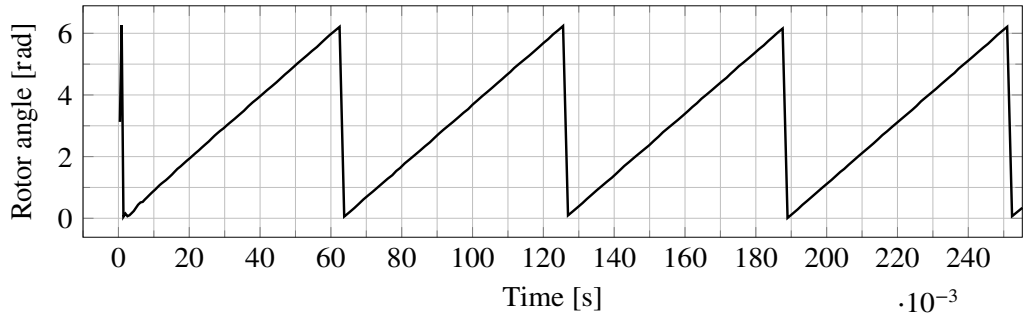
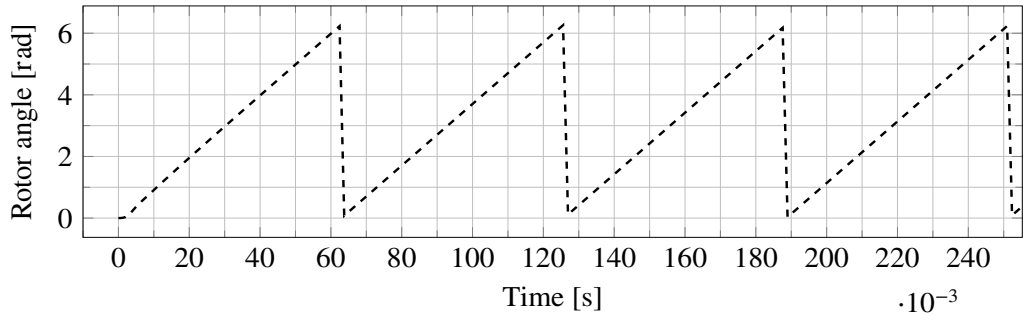


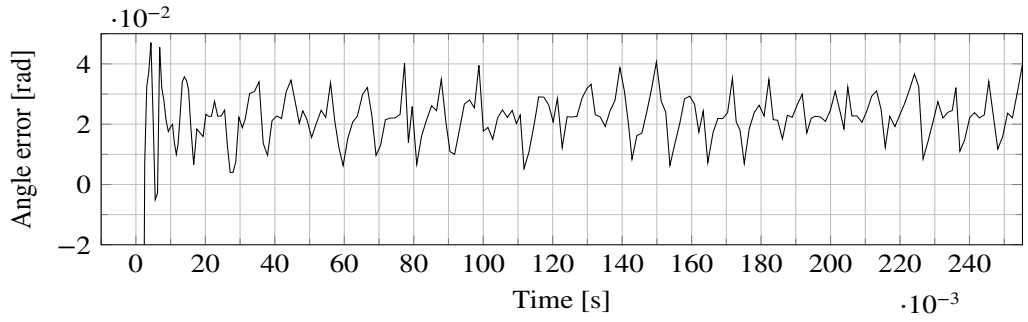
Figure 12.8: Locus of the estimated back-EMF of the sigmoid SMO



(a) Estimated rotor position



(b) Real rotor position



(c) Position estimation error

Figure 12.9: Real rotor position, estimated rotor position, and position error of the sigmoid SMO.

12.3 Combined observer

In an application note [64] Microchip Technology proposes a position estimator which combines a SMO with a linear Luenberger-type observer during small deviations from the actual states.

A Simulink® model of the proposed estimator has been developed, as depicted in figure 12.10.

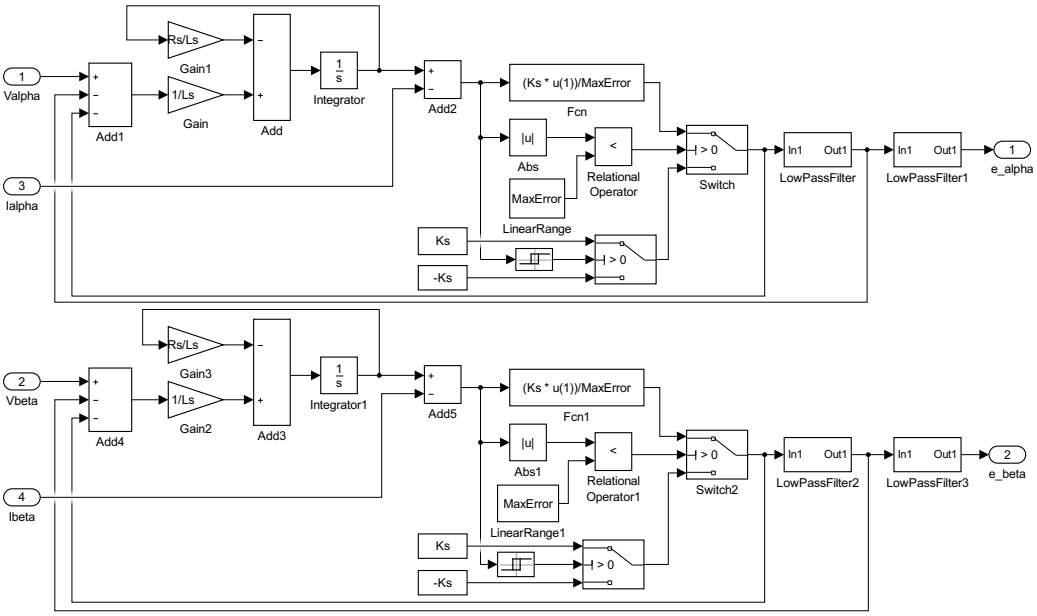
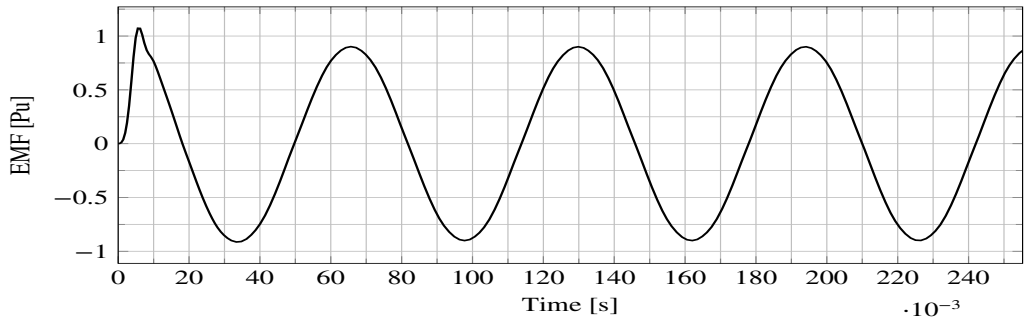


Figure 12.10: Simulink® model of the hybrid sliding mode observer

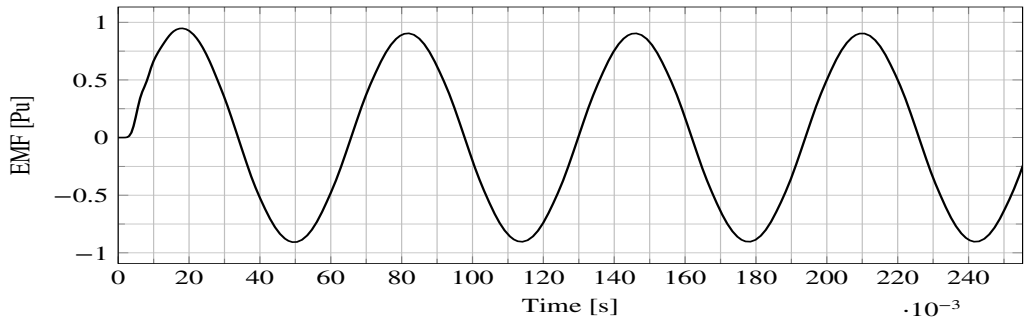
Figures 12.11 and 12.12 depict the estimated $\alpha\beta$-EMF, and EMF-locus respectively. The results indicate a even further improvement over the *sigmoid* observer.

The estimated position from the hybrid smo is depicted in figure 12.13. The dashed line depicts the real rotor position, while the solid line is the estimated value. Due to the more extensive filtering used by this algorithm, there is a profound phase shift between the estimated and the actual rotor angle. In order to correct this issue, the phase shift is compensated by a adaptive factor which depends on the estimated rotor speed.

Figure 12.15 depicts the block diagram of the adaptive phase compensator. The uncompensated position is differentiated in order to estimate the speed. The constants depend on the filtering constants of the SMO, and are determined heuristically.



(a) Alpha axis



(b) Beta axis

Figure 12.11: The estimated back-EMF of the hybrid SMO

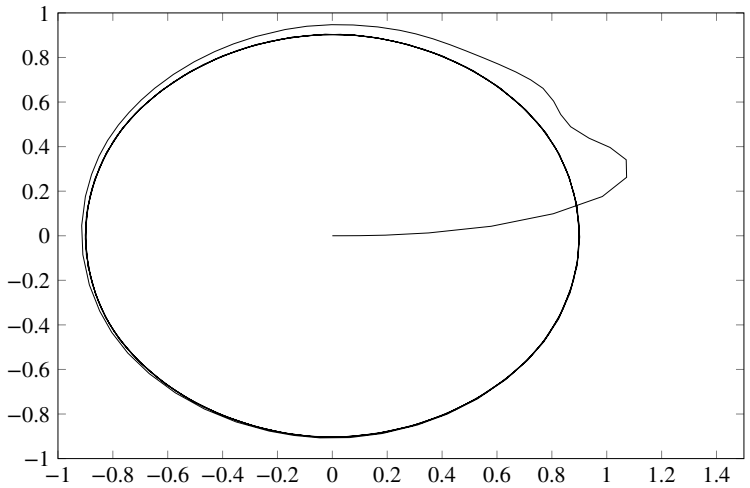


Figure 12.12: Locus of the estimated back-EMF of the hybrid SMO

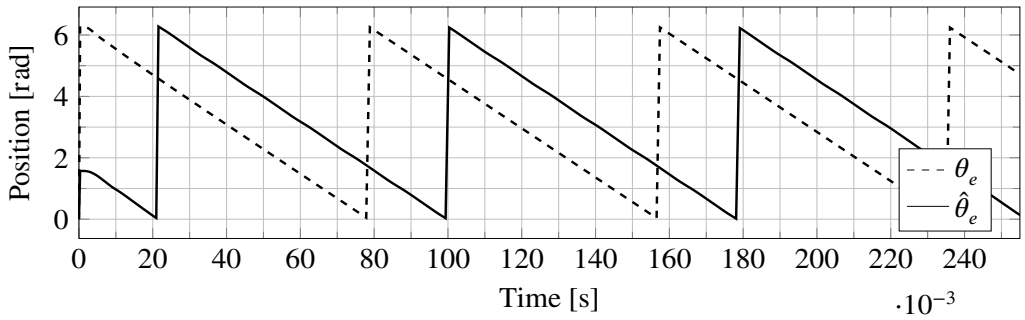


Figure 12.13: The uncompensated estimated rotor angle. Estimated at $\omega_{e,ref} = -40\,\mathrm{rad/s}$

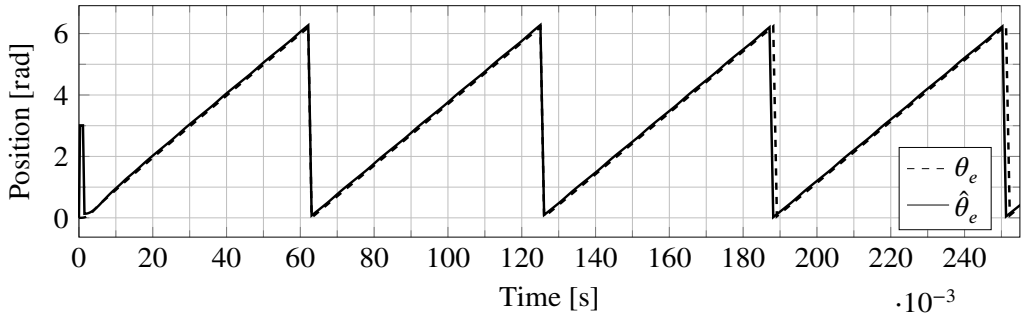


Figure 12.14: The compensated estimated rotor angle. Estimated at $\omega_{e,ref} = 50\,\mathrm{rad/s}$

Figure 12.16 depict the estimation error, i.e. the difference between estimated and real rotor angle. It is observed a small negative error. The relatively large peaks comes from the small deviation in the time in which the position is reset to zero. For a short time period the difference is close to 360°, but in reality that is the same as close to zero deviation.

The estimated speed is computed by discrete differentiation of the estimated position. When computing the speed, the phase shift of the estimated angle is irrelevant, only the slope matters. The result is depicted in figure 12.18. A certain amount of ripple is observed in the speed estimation, this may be compensated by decreasing the cut-off frequency of the low-pass filter at the output of the differentiator.

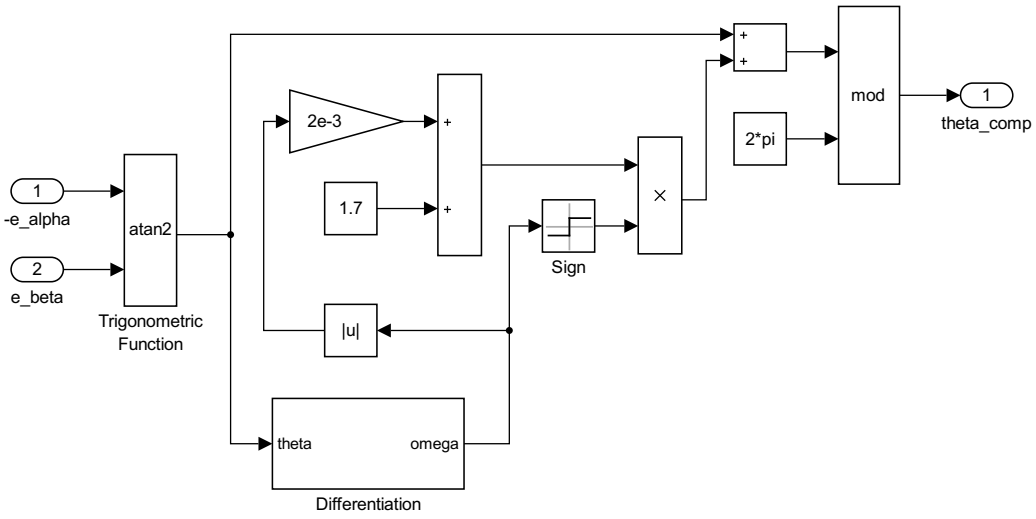


Figure 12.15: Adaptive phase compensation for the estimated position

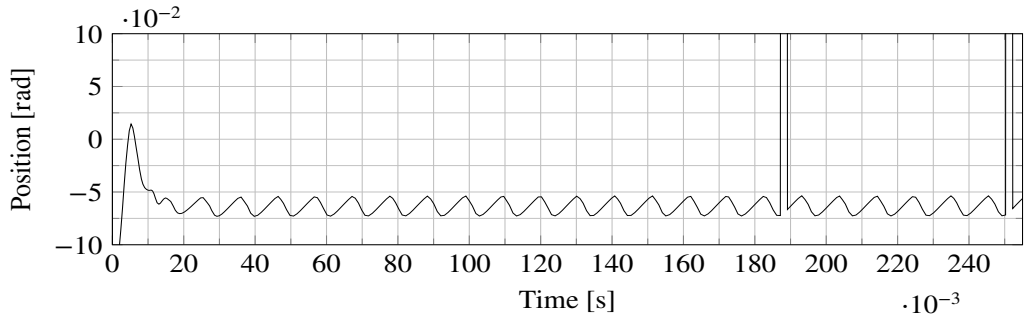


Figure 12.16: Estimation error for the hybrid SMO.

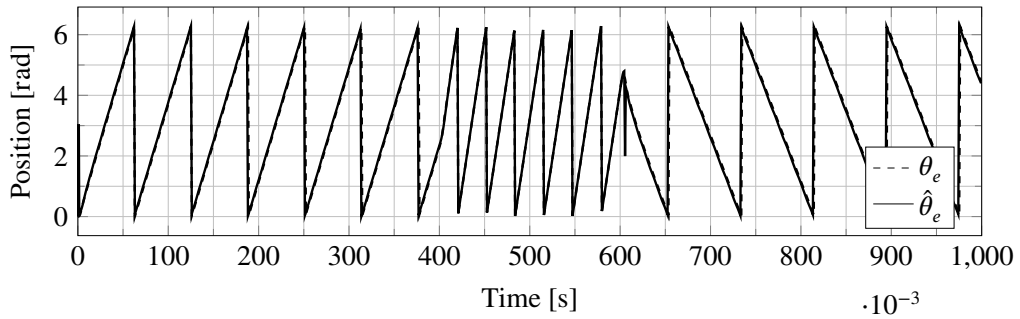


Figure 12.17: The compensated estimated rotor angle under changing conditions. $\omega_{e,ref}$ = 50 rad/s, $\omega_{e,ref}$ = 100 rad/s, $\omega_{e,ref}$ = −40 rad/s

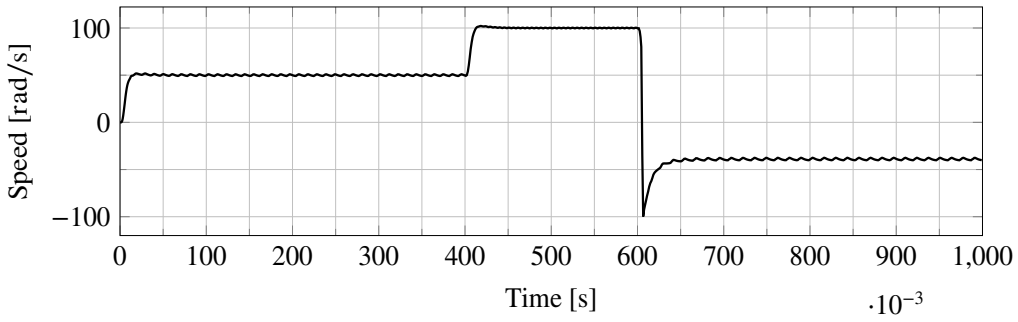


Figure 12.18: The estimated speed of the hybrid SMO. Calculated by discrete differentiation of the estimated rotor angle.

Part IV

Laboratory platform

Chapter 13

Selection of components

This chapter covers the selection of the required hardware components for the laboratory platform. Arguments for the different choices are presented, while the details of the dimensioning are left for the following chapters.

As the system is a laboratory platform, no specific requirements for the mechanical system (i.e load) are specified. A suitable motor is selected based on convenience. The converter is then designed to fit the ratings of the motor, although a certain amount of overload capability is added.

13.1 Selection of motor

The university already has access to a laboratory platform from Leybold®, where various motors are available. In order to be able to use this system, where transducers for speed and torque are also available, a motor of similar rating should be selected. The rating for the laboratory grade motors are 1 kW of mechanical power.

The voltage ratings should be in the range typically found in small industrial motors. In the Norwegian power grid the ratings of 400 V and 230 V are the most common consumer side voltages.

Another consideration to be made whether the motor should be a IPMSM, or a SPMSM. Some of the methods reviewed in the preparation project [21], require a salient machine. However the more robust methods should work independent of the selected machine.

Apart from these considerations, a high pole number might be beneficial for mechanical position control. However this is not a primary concern.

The selected motor is a three phase, 3.7 kW, surface mounted permanent magnet motor from Yaskawa Electric Corporation. The data for the motor is provided in appendix D.4.

13.2 Inverter

Although Silicon Carbide (SiC) transistors have gained popularity in recent years [34], a conventional IGBT is selected for the inverter. The primary consideration for this project is sensorless control, and this decision was made to avoid any non-relevant issues that may arise from technology that is still under active research.

The selected IGBT module is the *SKiiP 35NAB126V10* from Semikron. The module contains a three phase bridge rectifier, a brake chopper transistor, and a three phase two level inverter. The rated nominal output current is 50 A and the rated DC-link voltage is 1200 V. The datasheet[52] recommends a motor power of up to 15 kW for this inverter, i.e. it is overrated of about 400%, when compared to the selected motor. The reason for this overrating is to provide some additional flexibility if the inverter is to be used in future applications.

13.3 Digital control platform

The selected digital control platform is the Avnet ZedBoard™, based on the Zynq®-7000 SoC, from Xilinx.

Avnet also offers the MicroZed as a cheaper alternaitve to the Zedboard. It lacks some of the features, but none that are required for an inverter application.

Chapter 14

ZedBoard interface

In order to interface the Zedboard with the inverter, a application specific interface card has been designed. The card provides voltage translation for the digital signals, and analog to digital conversion for the required voltage and current measurements. It utilises a isolation interface in order to protect the Zedboard from any electrical faults in the laboaratory equipment.

The schematic, and circuit board layout for the isolation interface is provided in appendix A.

14.1 Zedboard interface card

Figure 14.1 depicts a block diagram of the signal flow on the ZedBoard interface. The dashed lines are the power supply lines.

A FPGA Mezzanine Card (FMC) connector on the Zedboard is used to interface the digital signals. The 3.3 V signals on this interface is translated to 5 V by the *ADG3308* voltage translator. The power supply for the *ADG3308* is also provided by the FMC.

Two different galvanic isolators are utilized, the *ADuM1400* Quad channel digital isolator, and the *ADuM7640* six channel isolator from Analog Devices [12, 13]. The isolators utilize magnetic coupling, as opposed to the more classical optical couplers. The isolator power supply is provided by a 5 V regulator, which steps down the 15 V from a external isolated power supply.

The *AD7403* $\Delta\Sigma$-modulator is used to convert the analog voltages from the voltage and current sensors to a digital representation [10].

The *ADuM5000* isolated power supply[11] is used to supply the *AD7403* ΔΣ-modulators. This facilitates the use of shunt resistors in the current measurements, as each ADC interface has a separately isolated ground.

The *CD40109* is used to translate the 5 V logic level of the isolation interface, to 15 V before it is sent to the gate driver of the inverter [28]. Although the gate driver is able to support 5 V logic, this translation is performed in order to improve the noise immunity.

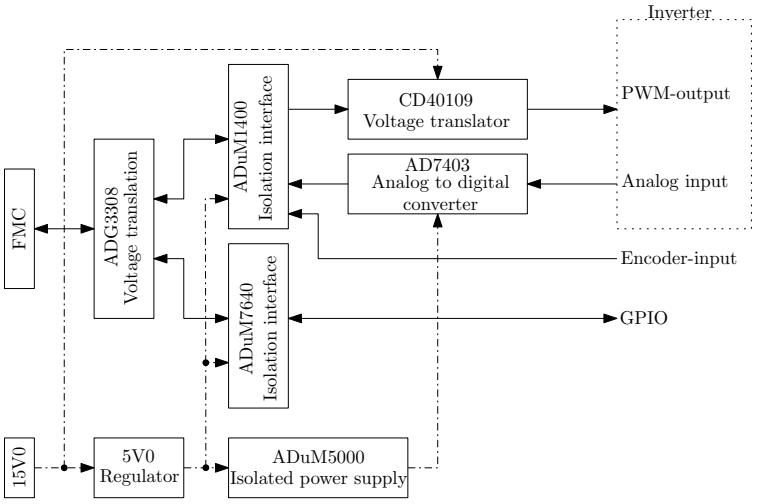


Figure 14.1: Block diagram of signal flow on Zedboard interface card. Dashed lines indicate power supply connections.

14.2 Signal conditioning

This chapter covers the practical aspects of how the various analog and digital signals in the inverter are maintained.

PWM control signals

The *Zynq 7000* digital outputs has a maximum voltage level of 3.3 V. The gate driver expects a logic voltage level of at least 5 V. When testing the inverter without using the interface card (chapter 14), a *74HCT541* octal buffer is used to boost the voltage level.

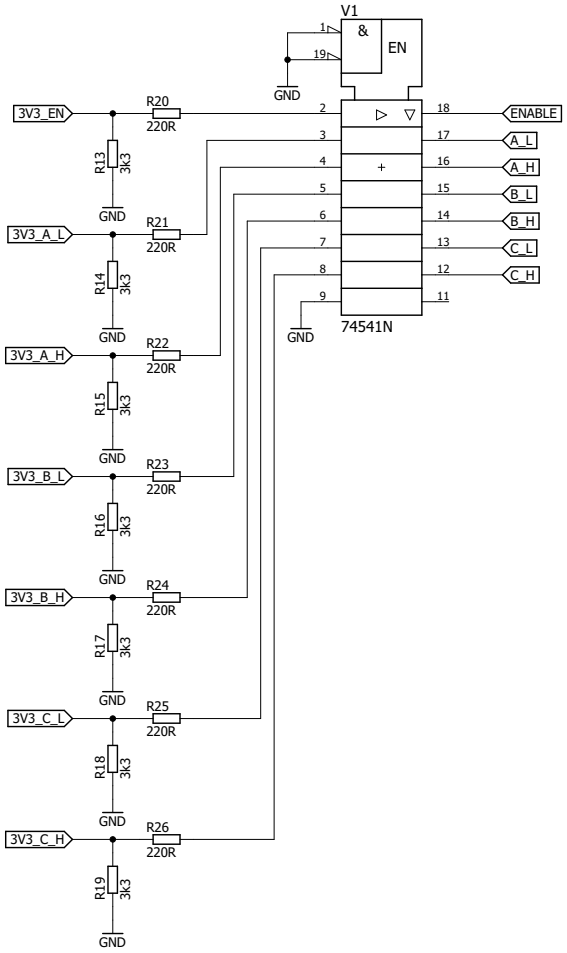


Figure 14.2: Voltage translation for the PWM signals.

The circuit diagram for the voltage buffer is depicted in figure 14.2. All inputs are pulled to ground by a 3.3 kΩ resistor. This ensures that any unconnected signal stays at zero volt.

Encoder

The selected encoder is the *ROD420-2048* from Heidenhain. It has 2048 pulses per revolution, and uses 5 V TTL outputs. A separate index pulse is provided for synchronization.

The specified cut-off frequency is 300 kHz. With a pulse count of 2048 the maximum number of revolution per second is given as:

$$\frac{300 \cdot 10^3}{2048} \approx 146$$

This is five times the rated speed of the motor, and thus should be more than sufficient. Another important parameter to consider is the angular resolution of the measurements:

$$\frac{360}{2048} \approx 0.176$$

The encoder has complementary outputs. This is done to improve the noise immunity of the interface, as any common mode noise may be removed by a differential amplifier.

The *AM26C32* Quadruple Differential Line Receiver from Texas instruments is used to interface the differential signals. The circuit diagram is depicted in figure 14.3. Termination resistors of 120 Ω are connected at the receiver, in accordance with the recommendations from the datasheet[27]. The voltage dividers at the output is used to decrease the logic levels from 5 V to 3.3 V.

Signal conditioning for ADC

The two different ADC circuits used in this design, the XADC and the ΔΣ-modulator, has a maximum differential voltage input of 1 V and 500 mV, respectively. This means that for any sensor voltage outside this range, signal conditioning is required. Ideally, in order to utilise the whole dynamic range of the ADC, the voltage should always be scaled to fit the maximum voltage window.

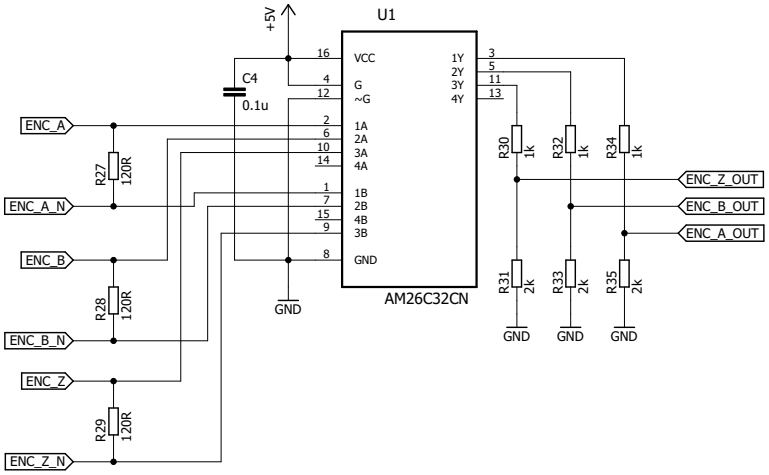


Figure 14.3: Encoder interface with voltage divider.

Simple resistive voltage dividers are typically the first thing to consider. However if the signal is biased it is difficult to have a efficient scaling, additionally there is the issue that the measured signal is attenuated by the voltage divider. For most applications operational amplifiers are a better solution.

When selecting operational amplifiers for signal conditioning, there are several important parameters that must be considered. First it is crucial to verify that the *Gain-Bandwidth Product* is sufficiently high. This parameter defines how the gain of the operational amplifier changes with changing signal frequency. It is important to have sufficient gain at the highest frequency of interest.

Anti-aliasing filter

When sampling a analog signal the sampling-theorem[53] states that we have to use a sampling frequency of at least twice the frequency of interest. If the sampling frequency is to low, this will cause aliasing in the digital domain. I.e. there will arise aliases at frequencies that differ from the actual analog signals. There is no way for the digital system to filter out these frequencies, or even to detect their presence.

In order to assure that these issues are avoided a low-pass filter is commonly applied at the input of the ADC. This filter is known as a anti-aliasing filter.

Although the ADC in the XADC is able to sample at a frequency ten times higher than the output bandwidth of the LEM-transducers, there may still be aliasing issues due to measurement noise. The Zedboard has a 10 MHz filter on the XADC

inputs, however in this project an additional RC-filter has been applied in order to reduce the bandwidth to about 200 kHz.

A simple first order RC-filter should be sufficient as the distance between sensor bandwidth and sampling rate is quite large, for more demanding applications however a more elaborated filter design might be required. Mircrochip offers the software tool FilterLab® to aid in the design of more sophisticated anti-aliasing filters.

Transducer conversion ratio

The datasheet for the voltage and current transducers are provided in appendix **??**, and D.1 respectively.

The output from the current transducer is given by:

$$V_{out} = 2.5 \pm 0.625 \frac{I_P}{I_{PN}} \tag{14.1}$$

Where $I_{PN} = 15A$ is the rated primary current, and I_P is the instantaneous primary current.

Similarly the output from the voltage transducer is give by:

$$V_{out} = I_{out} \cdot R_{measure} \tag{14.2}$$

Where the output current is given by:

$$I_{out} = \pm \frac{V_{in}}{40 \cdot 10^3} \tag{14.3}$$

E.g. if the measured input voltage is 1000 V, and the measurement resistance is 100 Ω, the output current, and voltage is given as:

$$I_{out} = \frac{1000}{40 \cdot 10^3} = 0.025\text{A}$$

$$V_{out} = 25 \cdot 10^{-3} \cdot 100 = 2.5\text{V}$$

Signal conditioning for XADC

Figure 14.4 depicts the circuit used to scale and shift the signal from the current sensors. A reference voltage of 0.5 V is generated by a voltage divider, and isolated by an operational amplifier in voltage follower connection. A second op-amp is used to attenuate the input, and shift it down to a bias of 0.5 V.

The selected op-amp is the *LF353N* from Texas instruments, which has a gain bandwidth of 4 MHz [30].

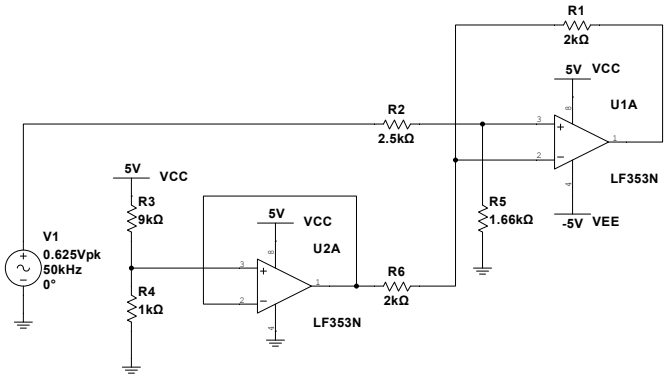


Figure 14.4: Scaling and biasing of current sensor output

The circuit has been simulated using *NI Multisim* from National Instruments, in order to verify the operation. A Stripboard has been used for the prototype.

Chapter 15

Voltage source inverter

This chapter covers the design and dimensioning of the Semikron based voltage source inverter (VSI).

15.1 IGBT module

The selected IGBT module is the Semikron *SKiiP 35NAB126V10*. Some relevant data from the datasheet is included in appendix D.3.

During initial development one of the IGBT modules where destroyed due to shoot through. A picture of the defective module is included in figure 15.1, where some annotations have been added to indicate the function of the various parts. The rectifier and chopper is still functional, while the inverter is destroyed.

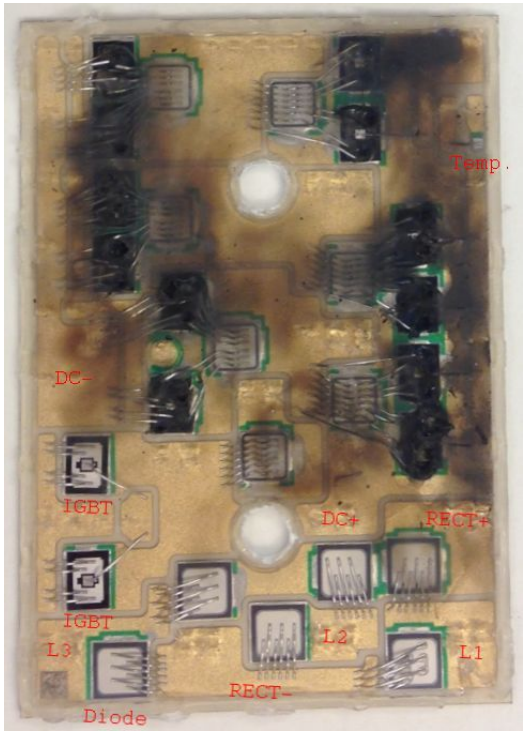


Figure 15.1: Defective IGBT module.

15.2 Thermal calculations

In order to assure sufficient cooling of the transistor-module, some initial thermal calculations are required.

Power losses

The power losses consists of switching and conduction losses, where the switching losses typically are the dominating factor. In addition a small amount of blocking losses due to lekage current will also be present, but are safely neglected in practice [19].

Hence the power losses may be expressed as:

$$P_{loss} = P_{cond} + P_{sw} + P_{block} \approx P_{cond} + P_{sw} \tag{15.1}$$

Conduction losses

The conduction losses are independent on the switching frequency, but depend on the duty-cycle of the transistor.

The voltage drop across the IGBT is a function of the collector current i_c, and may be approximated by:

$$V_{ce}(i_c) = V_{ce0} + r_c i_c \tag{15.2}$$

Where V_{ce0} is the voltage drop across the IGBT at zero current, and r_c is the IGBT on state resistance.

A similar approximation for the free-wheeling diode is given by:

$$V_D(i_D) = V_{D0} + r_D i_D \tag{15.3}$$

Then the instantaneous conduction losses for the IGBT is given by:

$$p_{cond,t}(t) = V_{ce}(t) \cdot i_c(t) = V_{ce0} \cdot i_c(t) + r_c \cdot i_c^2(t) \tag{15.4}$$

The average losses may be expressed by integrating the instantaneous losses over a switching period.

$$P_{cond,T} = \frac{1}{T_{sw}} \int_0^{T_{sw}} p_{cond,T}(t)\mathrm{d}t = \frac{1}{T_{sw}} \int_0^{T_{sw}} (V_{ce0} \cdot i_c(t) + r_c \cdot i_c^2(t))\mathrm{d}t \tag{15.5}$$

Which evaluates to:

$$P_{cond,T} = V_{ce0} I_{c,avg} + r_c I_{c,RMS}^2 \tag{15.6}$$

Similarly the instantaneous value of the conduction losses for the diodes may be expressed by:

$$p_{cond,D}(t) = u_D(t) \cdot i_D(t) = V_{D0} \cdot i_F(t) + r_D i_D^2(t) \tag{15.7}$$

Which by integration results in the following expression for the average power:

$$P_{cond,D} = V_{D0} I_{D,avg} + r_D I_{D,RMS}^2 \tag{15.8}$$

Switching losses

The switching losses in transistors and diodes are given as a product of the switching energy and the switching frequency.

$$P_{sw,t} = (E_{on,T} + E_{off,T}) \cdot f_{sw} \tag{15.9}$$

The turn-on energy in the diode is approximately equal to the reverse recovery energy:

$$E_{on,D} \approx E_{on,Drr} = \frac{1}{4} Q_{rr} U_{D,rr} \tag{15.10}$$

The switch-off losses in the diode are normally neglected, hence:

$$P_{sw,D} = (E_{on,D} + E_{off,D}) \cdot f_{sw} \approx E_{on,D} \cdot f_{sw} \tag{15.11}$$

The specified turn-on and turn-off energies from the datasheet are typically provided at rated operating conditions. In order to normalize the computed losses the following equation is typically applied:

$$P_{T,sw}(t) = k_T \cdot V_{DC} \cdot I_{0,avg} \cdot f_{sw} \tag{15.12}$$

Where:

$$k_T = \frac{E_{on,t} + E_{off,t}}{U_N \cdot I_N} \tag{15.13}$$

Calculations

The following design criteria, that affect the losses have been specified:

- Switching frequency $f_{sw} = 10\,\text{kHz}$
- Load current $I_C = 50\,\text{A}$
- Gate resistance $R_G = 22\,\Omega$
- DC-link voltage $V_{DC} = 540\,\text{V}$
- Output voltage $V_{out} = 400\,\text{V}$

- Output motor frequency $f_N = 100\,\text{Hz}$
- Motor displacement power factor $cos\phi = 0.85$
- Motor equivalent stator inductance $L = 10.4\,\text{mH}$

According to the previous derivations the conduction losses depend on both the average and the RMS value of the currents. Additionally the displacement of current with respect to voltage, i.e. the displacement power factor must also be taken into account.

Due to the symmetrical structure of the inverter all transistors and diodes exhibit the same amount of power loss. It is therefore sufficient to compute the losses in one transistor and one diode, and then multiply the result by the number of transistors and diodes.

First the average and RMS of the transistor current should be computed.

Bu assuming a sinusoidal shaped current, the relationship between the peak and the average current is given by:

$$I_{avg} = \frac{2}{\pi}\hat{i} \tag{15.14}$$

The modulation index m_a is given as the ratio of peak phase output voltage to half the DC-link voltage.

$$m_a = \frac{\hat{V}_{out}}{\frac{V_d}{2}} \tag{15.15}$$

The equations for conduction and switching losses have been adapted to work with the parameters of the pulse width modulated inverter.

$$P_{cond,T} = V_{ce0}(T_j) \cdot \hat{I}_o \left(\frac{1}{2\pi} + \frac{m_a \cos\phi}{8} \right) + r_c(T_j) \cdot \hat{I}_o^2 \left(\frac{1}{8} + \frac{m_a \cos\phi}{3\pi} \right) \tag{15.16}$$

$$P_{sw,T} = f_{sw}(E_{on,T} + E_{off,T}) \frac{\sqrt{(2)}}{\pi} \frac{I_o}{I_{ref}} \left(\frac{V_{cc}}{V_{ref}} \right)^{K_V} \cdot (1 + K_{TSW} \cdot (T_J - T_{ref})) \tag{15.17}$$

Cooling requirements

The datasheet[52] of the *SKiiP 35NAB126V10* specifies a absolute maximum junction temperature T_J of 150 °C. Additionally a thermal conductivity $R_{th(j-s)}$ of 0.55 K/W and 1 K/W in given per IGBT and diode respectively for the inverter and chopper. Finally the rectifier diodes have a specified thermal conductivity of 0.85 K/W.

The actual temperature of the IGBT-module affects the characteristics, and it should be limited somewhat when compared to the absolute maximum ratings. Semikron recommends that the devices should never be used at a temperature above 125 °C.

The absolute maximum ratings for the currents are specified at a temperature of 25 °C and 70 °C. In order to be able to use these ratings, and in accordance with common practice, a dimensioning temperature of 70 °C has been selected for the inverter at rated load conditions.

Semisel - Simulation

Semikron offers the tool Semisel® that simplifies calculation of losses and temperatures in the inverter. Simulations for a 15 kW motor are provided in appendix C.1.

A heat sink is selected based upon these calculation results.

15.3 IGBT gate driver

The gate driver selected for this project is the Semidriver® *SKHI 71 R* from Semikron. The driver has seven channels that are physically separated and isolated. Six channels are used for the inverter, while the seventh is used for the brake chopper.

The driver supports both 5 V and 15 V logic levels at the control input. In addition to the seven control signals, there is a common error input. The signal is active low, and should be pulled to 5 V in normal operation [51].

The driver outputs are bipolar, i.e. positive voltage on the gate with respect to the emitter is used to turn on the IGBT, while negative voltage is used to turn it off. The turn on threshold voltage $V_{GE(th)}$ of the IGBT-module is specified as 5.8 V. The use of negative turn-off voltage reduces the risk of parasitic turn-on of the

IGBT. Additionally the negative voltage decreases the turn-off time, which also decreases the switching losses [23].

A circuit board has been designed to support this driver. The schematic and board layout are available in appendix B.

IGBT drive signals

There are five terminals available for each transistor interface. The *gate*, and *emitter* terminals are required to drive the transistor. The *collector* terminal is used for short circuit protection, by measuring the voltage drop across the transistor, i.e. V_{CE}. The last two terminals are used to adjust the V_{CE} threshold.

A gate resistor of at least 10 Ω has to be connected between the driver and the gate of the transistor. This limit is imposed by the maximum current capability of the driver circuit.

Initial testing

If the gate driver is to be tested without connection to the transistors, the V_{CE} monitoring must be suppressed. This is achieved by shorting the collector, pin(9) to the emitter, pin(3). Additionally all the control signals must remain low for at least 9 µs in order to reset the error memory of the driver.

Figure 15.2 shows the measured response of the gate driver output for a 10 kHz 50% duty cycle square wave input signal. The selected dead-time for the gate-driver is 4 µs.

V_{CE} threshold, and blanking time

In order to detect short circuit, the voltage drop across the transistors is measured. If the voltage drop exceeds an adjustable threshold for a certain adjustable time period, the protection circuit disables all outputs.

The default turn-off threshold for the short circuit protection is 5.8 V. This voltage may be adjusted by connecting a resistor between the V_{CET1} and V_{CET2} pins of the interface.

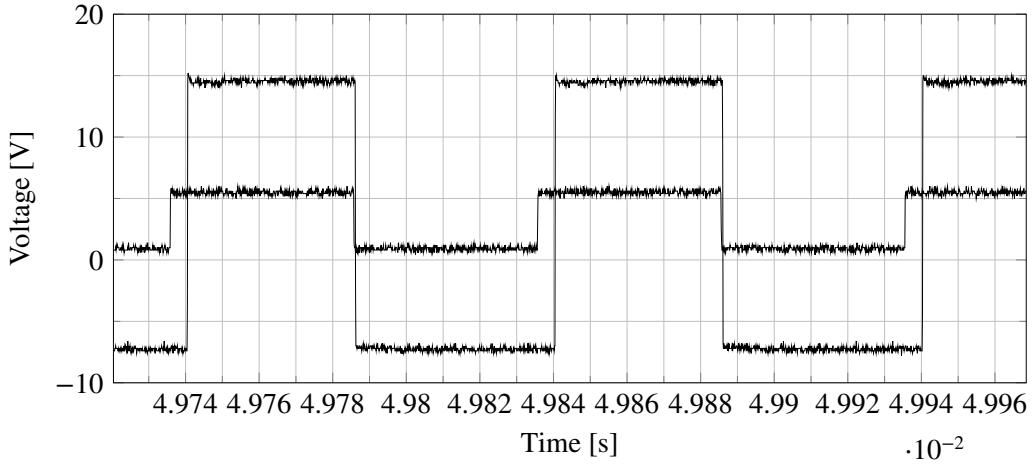


Figure 15.2: Response of the gate driver output to a square wave control signal.

The correct value for the resistor, depending on the desired V_{CE} limit, is given by equation (15.18). It should be noted that the threshold value cannot be increased, i.e. 5.8 V is the maximum value.

$$R_{VCE}[k\Omega] = \frac{11.86}{5.4 - 0.93V_{CE}} - 4.75 \tag{15.18}$$

In addition to the turn-off threshold, the blanking time should also be adjusted. This is achieved by connecting a capacitor between V_{CET2} and the emitter.

The value for the capacitor, depending on the desired blanking time is calulated by equation (15.19). As the capacitor value depends on the selected R_{VCE}, this adjustment must always follow when a new threshold has been selected.

$$C_{VCE}[nF] = \frac{t_{blank}[\mu s] \cdot (72.75 + R_{VCE}[k\Omega])}{(R_{VCE}[k\Omega] + 4.75) \cdot 36.08} - 0.1 \tag{15.19}$$

If desirable the V_{CE} monitoring may be disabled by connecting the collector pin to ground, i.e. to the emitter.

Interlock delay time

The interlock delay time is adjustable from 1 µs to 4 µs by the pins *TDT1*, *TDT2* and *SEL* on the primary pin array. The default delay is 4 µs if no connections are made [51].

A method for calculation of dead time requirements is presented in [62].

By connecting *SEL* to ground, the interlock time is disabled. This is useful if delay time is to be implemented in the controller. During development and initial testing however, the delay time should be included as an extra safety measure.

Gate resistor and switching frequency

The maximum switching frequency is calculated by equation (15.20). However the absolute maximum frequency is 50 kHz due to limitations within the electronics of the gate driver.

$$f_{max} = \frac{2 \cdot 10^4}{Q_{GE}[nC]} \tag{15.20}$$

Is some cases it is necessary to consider the power losses in the gate resistor [22]. By assuming a sawtooth shaped current waveform, the effective RMS value of the gate current may be approximated by:

$$I_{G,RMS} = I_{G,peak}\sqrt{\frac{t_p \cdot f_{sw}}{3}} \tag{15.21}$$

Where t_p is the turn on time for the transistor, f_{sw} is the switching frequency and $I_{G,peak}$ is the peak gate current.

$$P_G = I_{G,RMS}^2 \cdot R_G \tag{15.22}$$

The resistor must be able to dissipate this power. In addition the resistor must have a certain peak power capability in order to withstand the peak power during charge and discharge of the gate capacitor.

$$P_{peak} = I_{G,peak}^2 \cdot R_G \tag{15.23}$$

15.4 Calculation of component values

According to the output characteristics in the datasheet of the IGBT module, the V_{CE} saturation voltage at nominal operating condition lies in the range 1.7 V to

2.1 V. This implies that the default short-circuit threshold of 5.8 V might be a bit to high, and should be adjusted.

As a starting point 4 V is selected. In order to avoid false tripping of the protection 4 µs of blanking time is also addded.

According to the datasheet, a gate resistance of 12 Ω is the minimum. Then the typical turn on and turn off times for the IGBT are 85 ns and 430 ns respectively. Thus a interlock delay of 2 µs should be more than sufficient to avoid shoot through.

Initially, in accordance with common design practice a gate resistor of 22 Ω is selected, order to limit the rise and fall times even further. This reduces the likelihood of over voltages, due to parasitic elements in the components. Once laboratory experiments has confirmed the operation, the gate resistor may be reduced gradually, in order to obtain the best compromise between voltage transients, and switching losses.

For the selected inverter, gate-driver combination used in this project the following limitations are thus imposed:

- V_{CE} threshold 4 V
- Blanking time 4 µs
- Interlock delay time 2 µs

Based on this information the components for threshold and blanking time are calculated:

$$R_{VCE} = \frac{11.86}{5.4 - 0.93 \cdot 4} - 4.75 = 2.3[k\Omega]$$

$$C_{VCE} = \frac{4 \cdot (72.75 + 2.3)}{(2.3 + 4.75) \cdot 36.08} - 0.1 = 1.08[nF]$$

To summarize the following component values has been selected:

- Gate resistor R_G 22 Ω
- Threshold adjustment resistor R_{VCE} 2.3 kΩ
- Blanking time capacitor C_{blank} 1.08 nF

15.5 Dynamic test board

Semikron supplies a dynamic test board to aid in the connection of the inverter modules.

The maximum continuous current for the evaluation board is specified to 30 A. This is lower than the selected MiniSKiiP® IGBT module, which has a rating of 50 A.

15.6 Power circuit board

The dynamic test board is not intended for a final production design. A additional power circuit board should be designed to be used in the final laboratory platform. A Eagle® footprint for the inverter module has been prepared, and is available in the accompanying files of the project.

A safety turn off resistor of 10 kΩ should be added between gate and emitter in order to ensure that the IGBT is turned off if the gate driver is disconnected. The new board should facilitate simpler access from the collectors of the transistors, to the gate driver. Finally simple access to measurement of the DC-link current should be available.

15.7 Brake chopper controller

The brake chopper controller is designed to run independently of the main controller. This separation is done to assure that this safety system is unaffected by errors in the controller.

The DC-link voltage is measured and compared to a reference voltage. If the measured value exceeds the reference the controller sends a *on* signal to the gate driver of the brake chopping transistor. The switching frequency is limited by a certain hysteresis between the turn on, and turn off voltages.

The DC-link voltage is measured by a *LV 25-1000* voltage transducer from LEM®. The details of the sensor are provided in the datasheet [37].

In no-load conditions the sensor output will be as follows:

$$V_{sens} = 2.5 \cdot \frac{V_{DC}}{1500} = 2.5 \cdot \frac{540}{1500} = 0.9[V]$$

The absolute maximum DC-link voltage is limited by the voltage rating of the capacitors. The two series connected capacitors has a resulting rating of 800 V, however increased lifetime may be achieved by imposing a lower limit.

A threshold of 600 V is selected, giving the following reference voltage for the comparator:

$$V_{ref} = 2.5 \cdot \frac{600}{1500} = 1.0[V]$$

The design of op-amp based voltage comparators is presented in [7].

Figure 15.3 depicts the resulting design. The voltage source models the *LV 25-1000* voltage transducer. The amount of positive feedback should be adjusted depending on the resulting switching frequency.

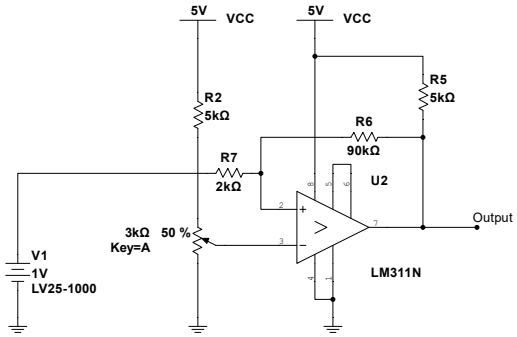


Figure 15.3: Brake chopper comparator.

Brake chopper resistor

The brake chopper should be connected to a resistor of suitable value, and it should be able to dissipate the braking power of the motor.

The lower limit for the brake resistor is determined by the current handling capability of the brake-chopper transistor [60].

$$R_{min} = \frac{540}{50} = 10.8[\Omega]$$

The instantaneous dissipated power for this resistor is given as:

$$P_R = 50^2 \cdot 10.8 = 27[kW]$$

This is the maximum braking power. Smaller values might be sufficient however, depending on the mechanical load, and on the braking requirements [60].

In order to calculate the required braking power some knowledge of the mechanical system is required. In the laboratory platform no specific mechanical constraints have been imposed, and thus there is no premise to calculate anything but the maximum braking power. The following calculation procedure however, is included for completeness.

The kinetic energy of a rotating mass is given by:

$$E = \frac{1}{2} J \omega_m^2 \tag{15.24}$$

Where J is the inertia, and ω_m is the angular velocity.

The energy difference due to a change in speed may then be expressed as:

$$\Delta E = \frac{1}{2} J (\omega_{m,1}^2 - \omega_{m,2}^2) \tag{15.25}$$

Where $\omega_{m,1}$ is the initial velocity, and $\omega_{m,2}$ is the final velocity.

The average power that must be dissipated by the brake resistor is given as:

$$P = \frac{\Delta E}{t} \tag{15.26}$$

The required braking torque is given by:

$$T_{brake} = J \frac{\mathrm{d}\omega}{\mathrm{d}t} \tag{15.27}$$

The required braking time depends on the torque, and on the required speed change, as given by equation (15.28). This is important as the sizing of the resistor depends on the energy, not the instantaneous power. As a typical rule of thumb the resistor should be rated for continuous power handling, if the braking time exceeds 1 minute [60].

$$\Delta t = J \frac{\mathrm{d}\omega}{\mathrm{d}T_{brake}} \tag{15.28}$$

The motor used in this project has a rated power of 3 kW. To make things simple a resistor has been selected that is able to continuously dissipate this power. The

brake chopper is as much a safety system as it is a feature of the inverter. If a bug in the control algorithm causes excessive braking, the chopper should protect the DC-link from over-voltage.

15.8 Pre charge of DC-link

In order to limit the inrush current when voltage is applied, a resistor is initially connected in series with the capacitors in the DC-link. Once the capacitors reaches nominal voltage, the resistor is bypassed by a electromechanical relay. The the most relevant parts of the datasheet for the selected capacitors are provided in appendix **??**.

The initial capacitor current for an idealised circuit is given by equation (15.29). In a practical system however there will be inductances and resistances limiting this current [49].

The average no-load voltage output from a three phase rectifier is given by (15.30).

$$i_{inrush} = i_c = C\frac{\mathrm{d}V}{\mathrm{d}t} \tag{15.29}$$

$$V_{dc} = V_{avg} = \frac{3\sqrt{3}\hat{V}}{\pi} = \frac{3\hat{V}_{L-L}}{\pi} \tag{15.30}$$

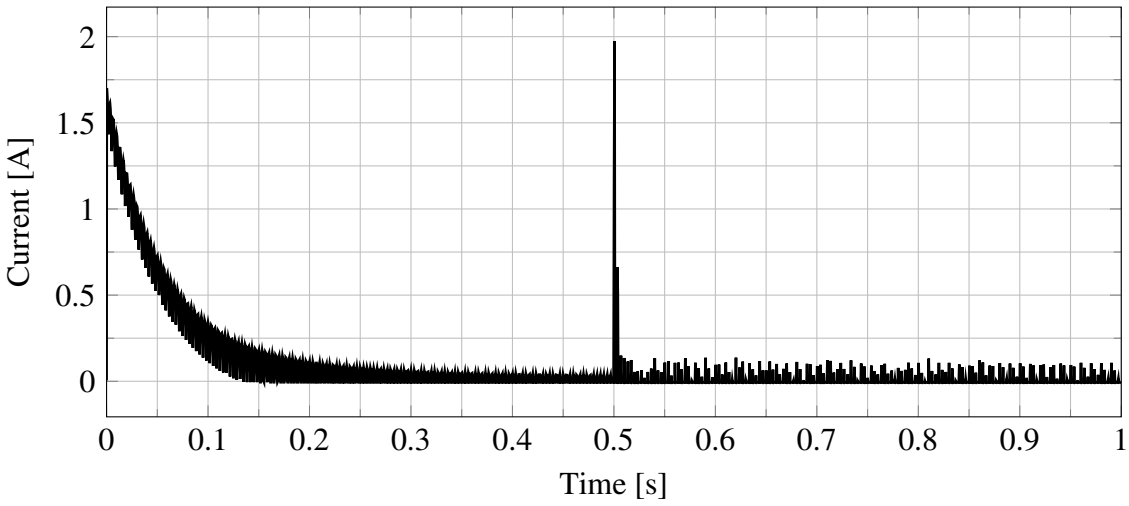
Based upon recommendations from the manufacturer, the selected resistor has a value of 330 Ω, and the net resulting capacitance is 165 µF. The supplied three phase line to line voltage on the primary side of the rectifier is 400 V.

The average rectified voltage is calculated as:

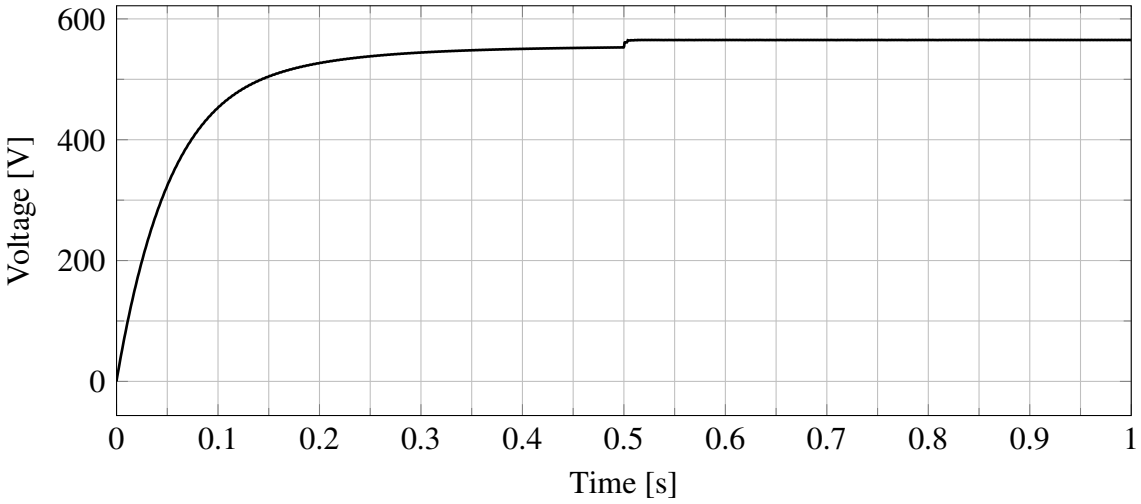
$$V_{dc} = \frac{3 \cdot 400 \cdot \sqrt{2}}{\pi} = 540.2[V]$$

Simulation of DC-link charge

Figure 15.4a and 15.4b shows the resulting current and voltage respectively, from a simulation of the charging of the DC-link capacitor. A 330 Ω resistor is connected in series with the DC-link during initial charge, and bypassed at $t = 0.5$ s. It is observed that the current is limited to a safe operating range.



(a) Capacitor charge current



(b) DC-link voltage

Figure 15.4: Current and voltage during initial charge of the DC-link capacitor.

Laboratory verification

Figure 15.5 depicts the measured voltage output from the 6-pulse rectifier, along with the input voltage for one of the phases. It is possible to observe a small deviation between the two curves, during the conduction state. This is due to the forward voltage drop in the diodes.

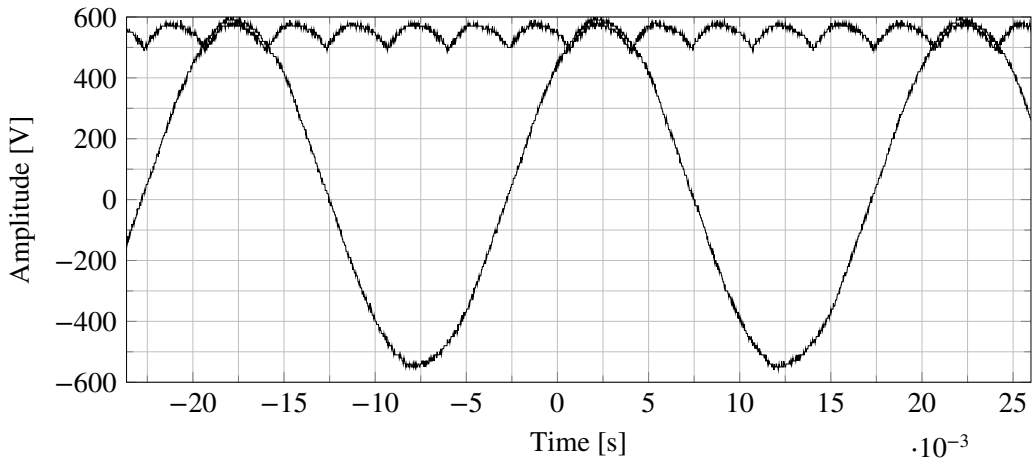


Figure 15.5: Measured voltage output from the 6-pulse diode rectifier.

Figure 15.6 displays the measured charge current and capacitor voltage, when power is initially applied. It is observed that the current decays at approximately the same rate as the simulations suggested. The circuit board does not provide easy access to measurement of the DC-link current, and thus the AC-side current is measured instead.

Pre-charge controller

A simple pre charge controller has been designed to automatically bypass the charge resistor once the voltage reaches a specified threshold.

The voltage output from the transducer is compared to a adjustable reference voltage by use of a simple analog comparator. A SR flip-flop is activated once the threshold is reached, and the resistor remains disconnected until the system is reset by a power cycle, or by the reset switch.

As the capacitors are charged the DC-link voltage approaches 540 V, and the sensor voltage approaches:

$$V_{sens} = 2.5 \cdot \frac{540}{1500} = 0.9[V]$$

The comparator reference voltage is adjusted to a level somewhat below this expected sensor output.

The circuit is depicted in figure 15.7.

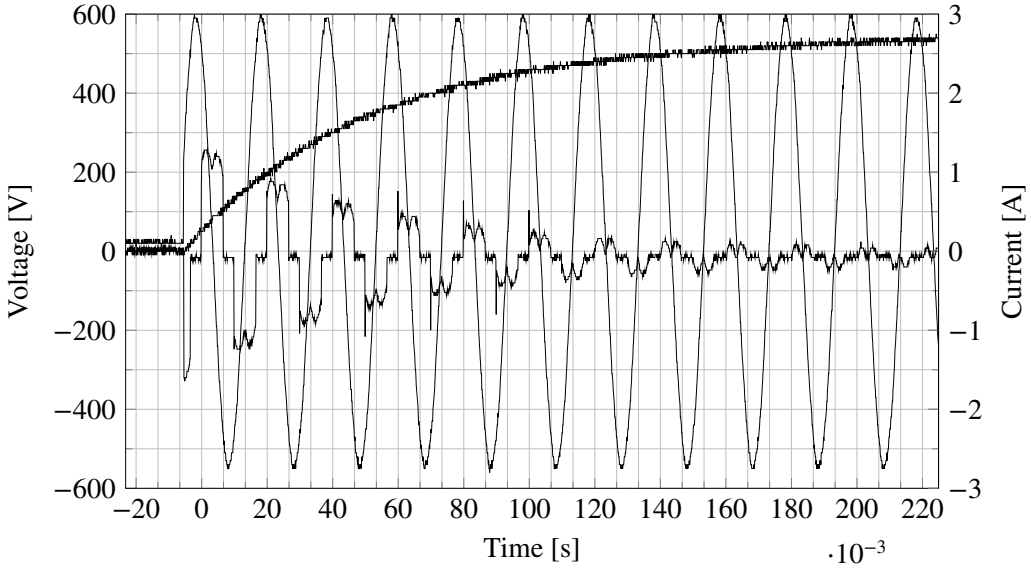


Figure 15.6: Charging of the DC-link capacitor.

The SR flip-flop is not able to drive the relay directly, and thus a transistor is used to amplify the output.

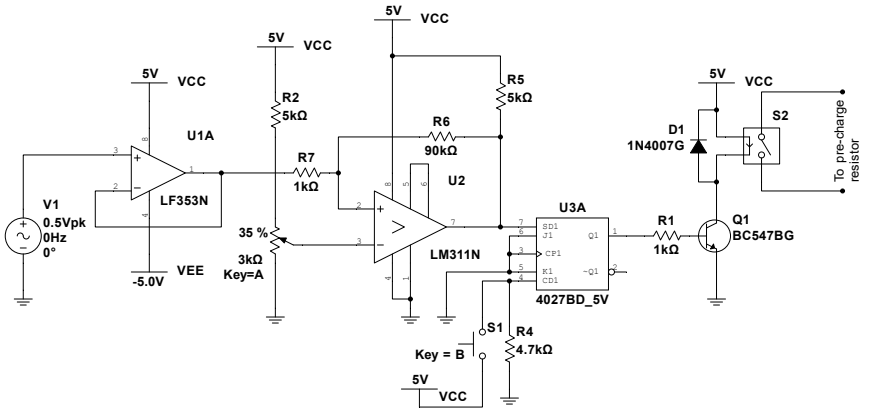


Figure 15.7: Pre-charge controller

15.9 Temperature monitoring

The *SKiiP 35NAB126V10* has a built in resistive temperature sensor. According to the datasheet, the resistance is 1000 Ω at 25 °C and 1670 Ω at 100 °C.

As the relationship between temperature and resistance is not linear, the datasheet does not provide sufficient information to calculate the actual resistance.

The MiniSKiip® Technical Explanations manual[67] provides the following equation to approximate the resistance:

$$R_{ts}(T) = 1000\Omega \cdot (1 - A(T - 25) + B(T - 25)^2) \tag{15.31}$$

Where A equals $7.635 \times 10^{-3}/°C$ and B equals $1.731 \times 10^{-5}/°C^2$.

Figure 15.8 depicts the circuit diagram for the implemented temperature monitor. By using a voltage divider, and a analog comparator[31], a error signal is generated when the temperature exceeds the ratings of the inverter. This error signal is fed to the *_ERRIN* external error signal input of the gate driver. A signal on this connector disables the gate driver, and outputs negative voltage to all IGBT-gates. The input is active low, and thus a inverting buffer is added to the output from this circuit.

In order to have some hysteresis between the turn on and turn off, some positive feedback is applied to the amplifier. The positive feedback fraction of the voltage is given by:

$$\beta = \frac{R_7}{R_6 + R_7} \tag{15.32}$$

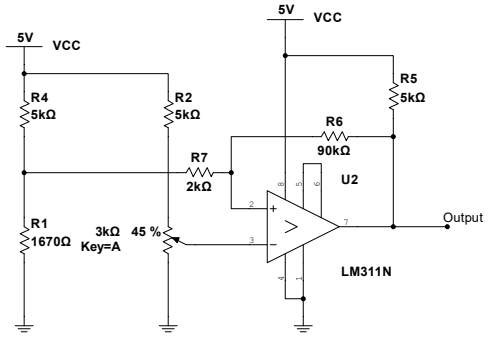


Figure 15.8: Voltage comparator for thermistor

Chapter 16

Laboratory experiments

This chapter covers the final laboratory setup, based upon the components covered in the previous chapters.

16.1 Laboratory setup

Figure 16.1 provides a overview of the laboratory setup. The SPMSM in front, and all the required components of the drive.

Figure 16.2 provides a closer look at the three *LEM LTS 15-NP* current sensors. The sensors are mounted in a box that provides screening, as well as protection from the potential hazardous voltages. A internal 5 V power supply from *Traco Power* is used to supply the three sensors.

Figure 16.3 depicts the connections between the gate driver and the inverter. The gate driver is the white box to the right, while the green circuit board is the main power board. The DC-link capacitors are visible at the top left.

Figure 16.4 depicts the connections to the Zedboard. Power and ethernet is connected at the top. Encoder input and PWM output at the bottom. Finally the XADC interface at the top right.

The laboratory test equipment used in this project is listed in appendix F. The data for the SPMSM is provided in appendix D.4.

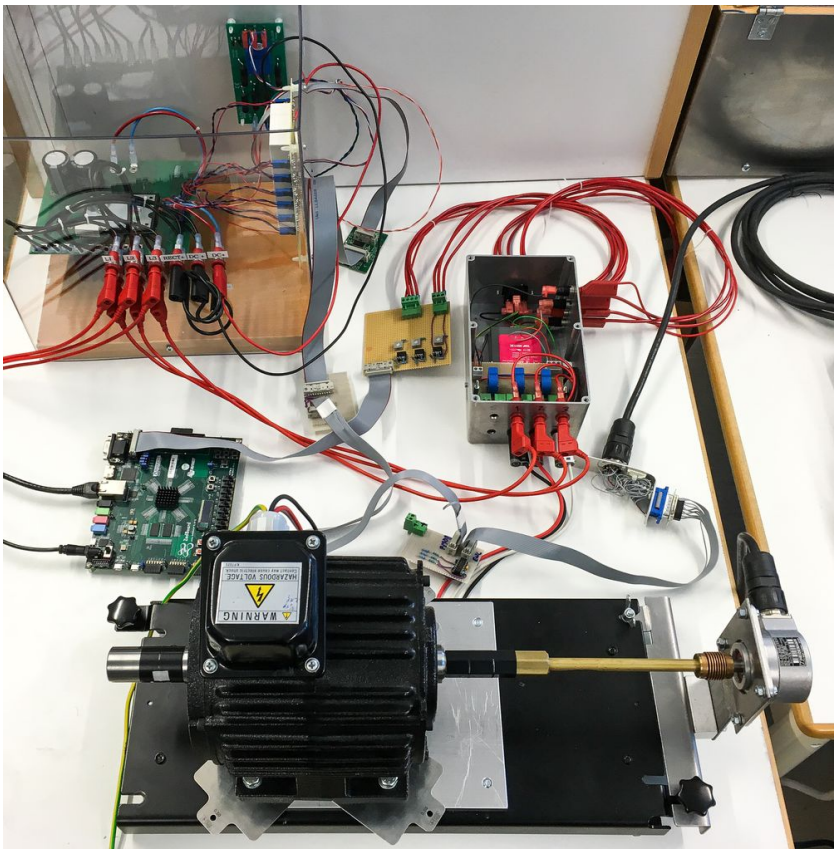


Figure 16.1: Overview of laboratory setup.

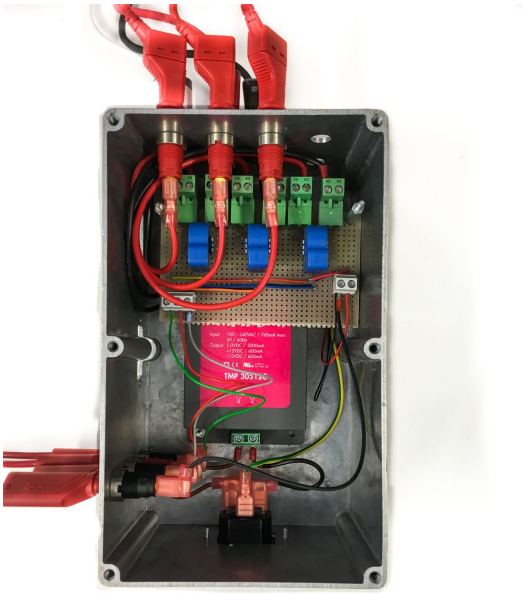


Figure 16.2: Box with three LEM LTS 15-NP current transducers. The primary current is connected at the top, wile the measurement signal is at the left hand side.

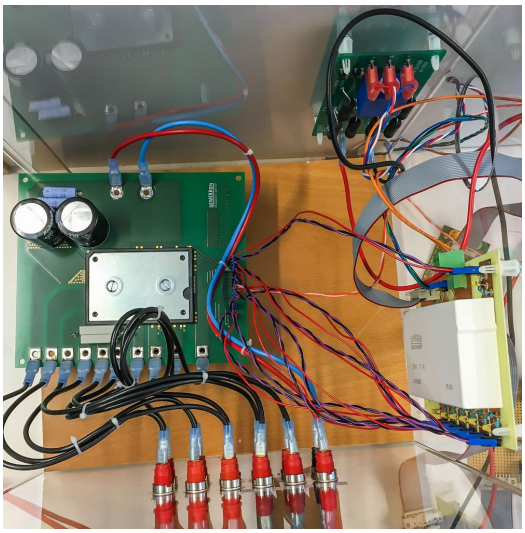


Figure 16.3: Inverter power board with DC-link capacitors, DC-link voltage transducer, and IGBT-driver board.

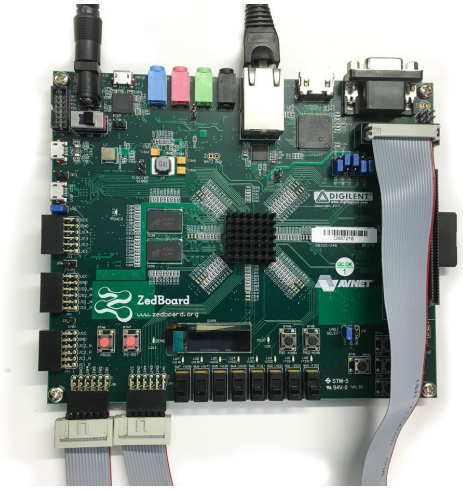


Figure 16.4: Zedboard with interface to the motor drive.

16.2 Laboratory testing

This chapter covers the results from the laboratory experiments performed on the PMSM-drive. In order to be comparable, the experiments have used the same speed reference profile as the simulations.

Figure 16.5 depicts the IGBT response to a 8 kHz control signal. The dead time is adjusted to 4 µs. Although the upper transistor is turned off immediately, when the control signal goes low, the voltage remains constant during the short duration before the lower transistor turns on. This is likely due to stray capacitance, and the fact that this test was conducted during no-load conditions.

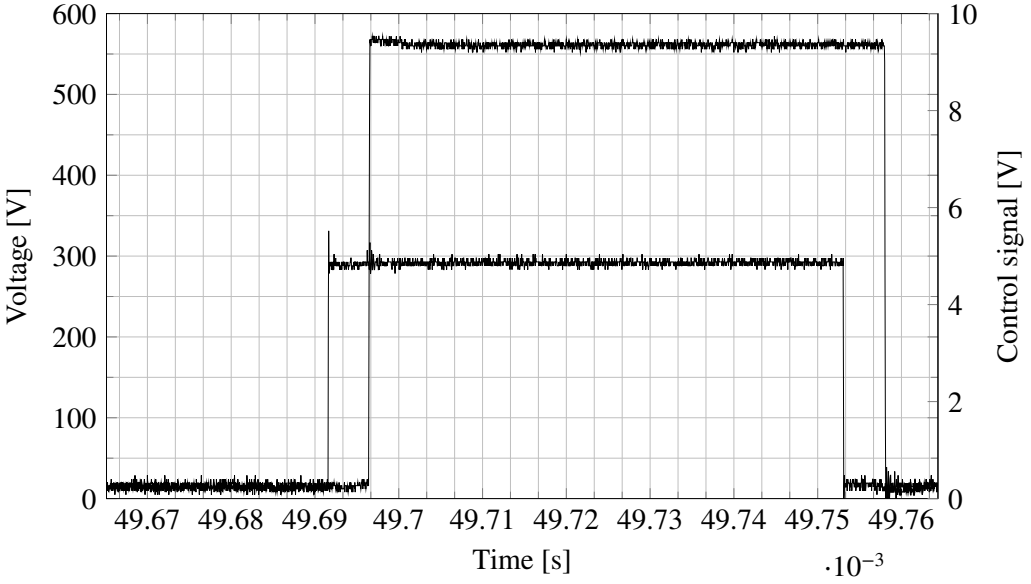


Figure 16.5: IGBT response to a turn on, and turn off command.

Figure 16.6 depicts the measured inverter line to line output voltage. The DC-link voltage is 540 V. It is observed that the pulse width modulator that is running in the FPGA is operating as intended.

Figure 16.7 depicts the output voltage from the LEM current transducers. The conversion ratio of the transducers is covered in chapter 14.2. To summarise there is a 2.5 V bias, and a scaling factor of 24. The voltages are obtained during 5 Hz operation.

Figure 16.8 depicts the three phase currents, acquired by removing the bias and scaling the measured voltage from the transducers. Figure 16.9 depicts the locus

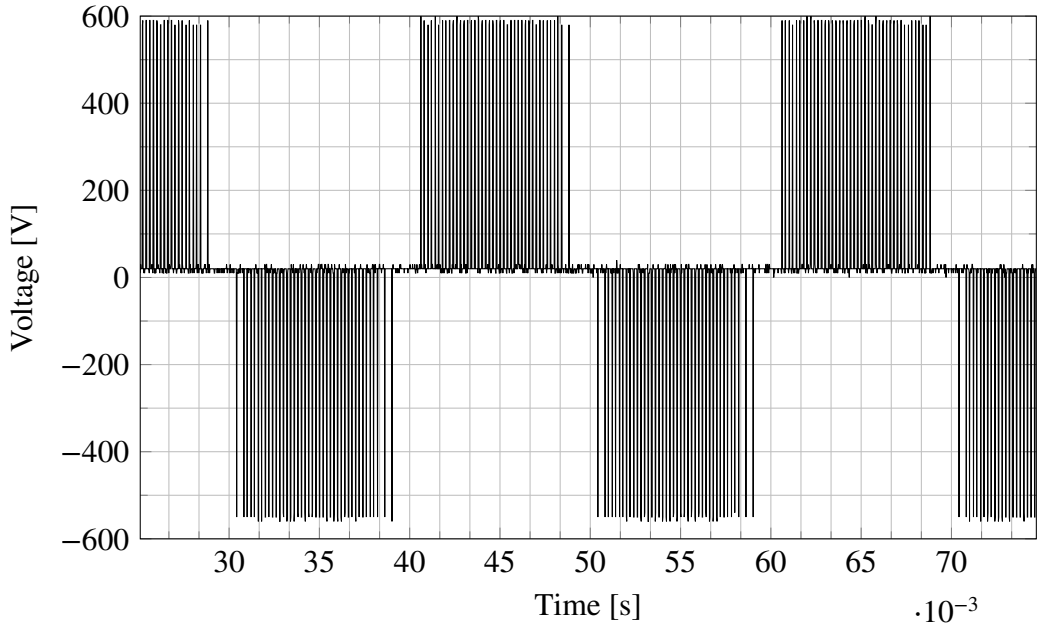


Figure 16.6: Measured line to line output voltage of inverter. $f_{sw} = 5\,\text{kHz}$

of the current, obtained by transforming the currents to $\alpha\beta$-frame. It is observed a circular shape, although the signal is relatively noisy.

Figure 16.10 depicts the estimated position from the three SMO based position estimators. The estimation is performed under the same operating conditions, which corresponds to the current depicted in figure 16.9. It is observed that the performance of the real system is in close correspondence to the result indicated by the simulations.

Although adaptive phase compensation could improve the performance of the estimator in figure 16.10a, this has not been included. The performance of this estimator is considered unsatisfactory anyhow. Adaptive compansation is however included in the estimator in figure 16.10c. It is observed that the estimator in figure 16.10b, has almost the same performance as 16.10c, without requiring compensation.

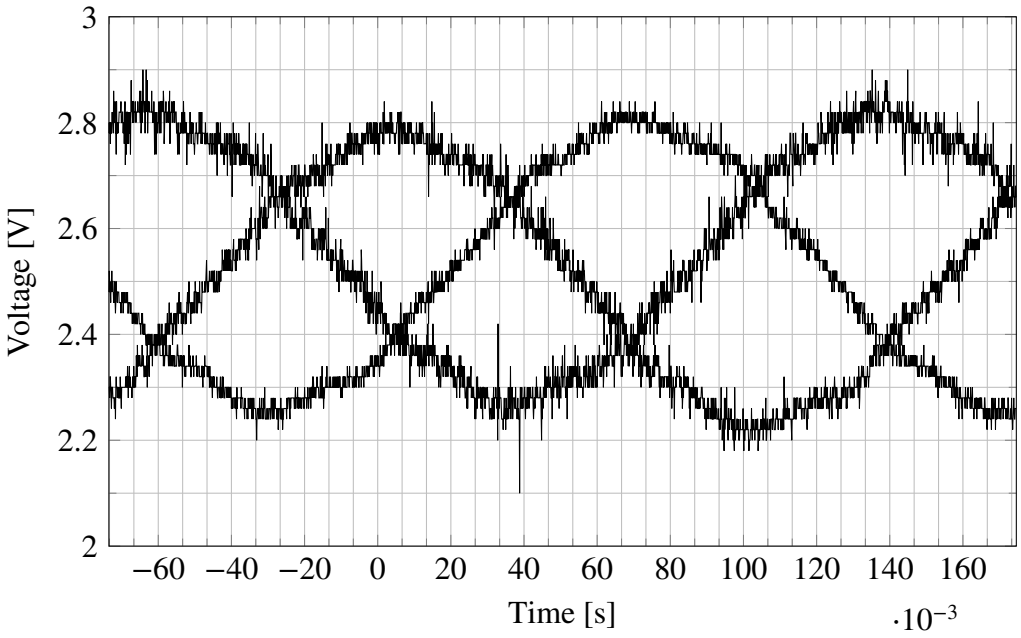


Figure 16.7: Measured output voltage of current transducer.

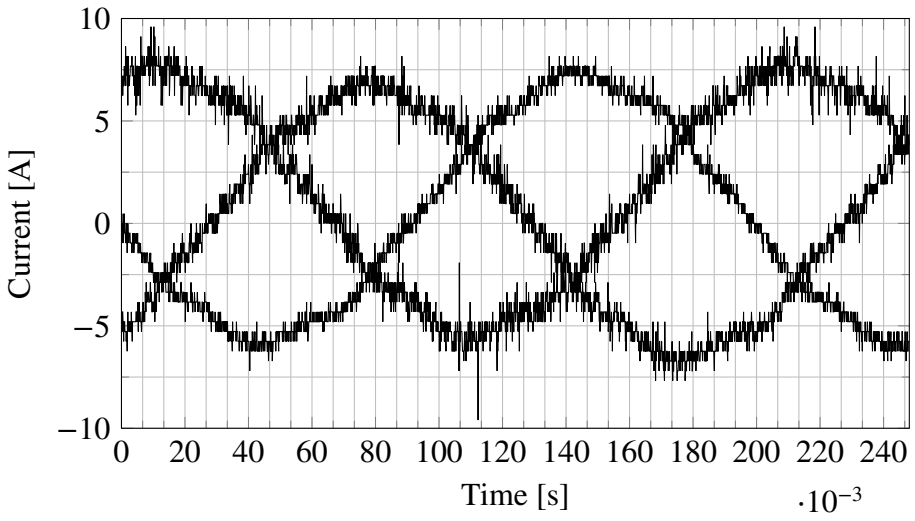


Figure 16.8: Measured motor current without bias, and scaled to real value. The time axis is also adjusted.

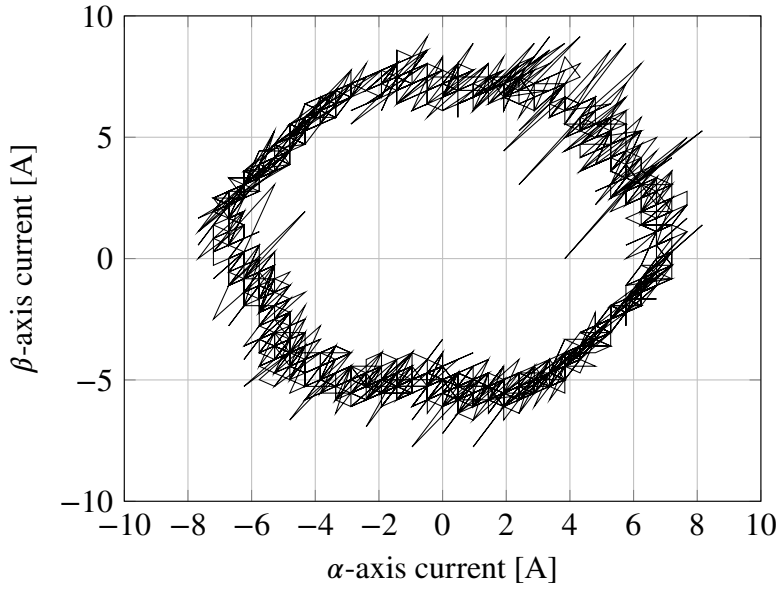
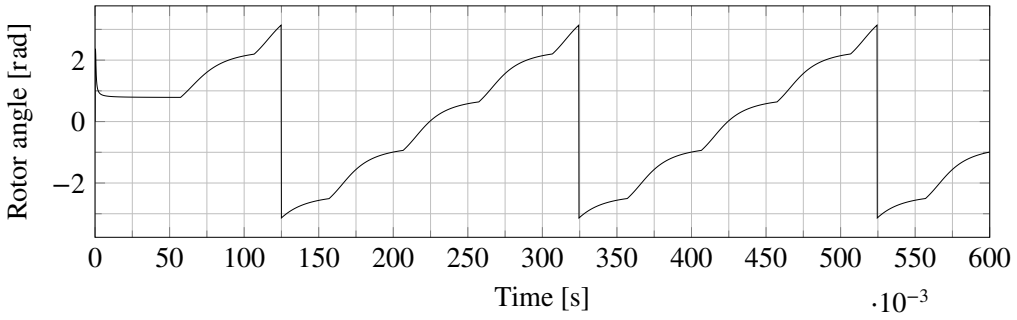
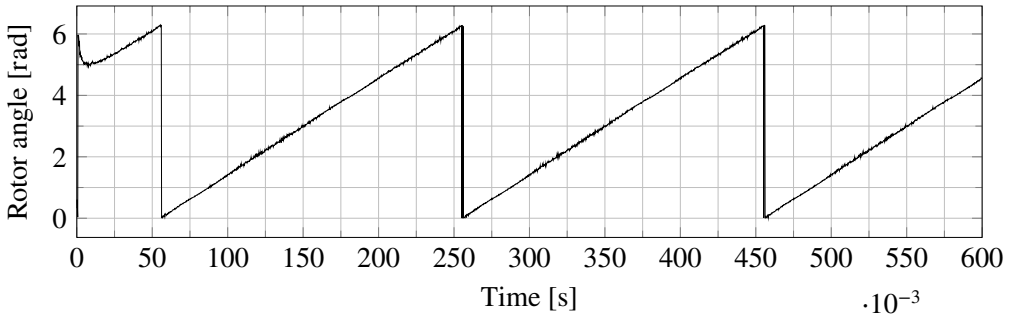


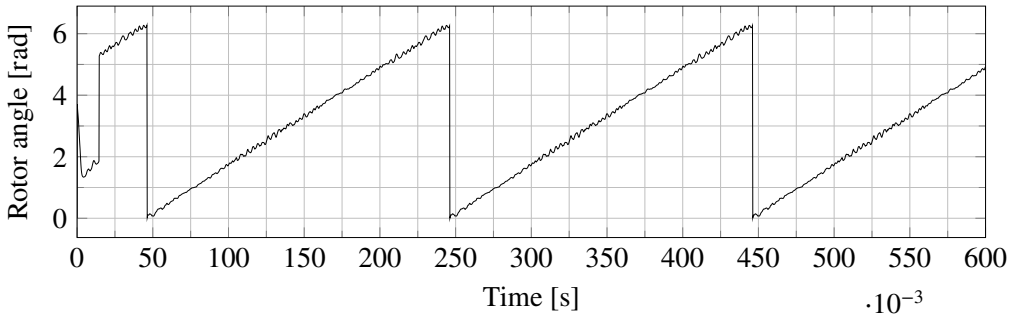
Figure 16.9: Locus of the measured current.



(a) Simple SMO



(b) Sigmoid SMO



(c) Hybrid SMO

Figure 16.10: Estimated rotor position from three different position estimators.

Discussion and conclusion

This master thesis has examined several hardware and software aspects of the implementation of permanent magnet synchronous machine drives. The main topics has been the theoretical development, and practical implementation of several control algorithms and position estimators. As well as the practical aspects of designing a power electronic converter for motor control applications.

Simulations have been carried out to verify the operation and performance of the proposed algorithms. The models have been developed in a manner that facilitates generation of synthesizable HDL code, or compilable C++ code depending on the application.

Emphasis has been put on eliminating the requirement of a position sensor on the rotor shaft, and rather rely on the estimation of rotor position. Simulation of the proposed estimation algorithms is performed, and their performance is compared.

Three different variations of the sliding mode observer are simulated. It is observed that despite the simple implementation, the *sigmoid* based observer has the best performance. Especially since it avoids the requirement of filtering, which in turn avoids the requirement of adaptive phase compensation. Laboratory experiments are used to confirm the operation.

A laboratory platform for a permanent magnet synchronous machine drive has been developed. Algorithms for pulse width modulation, and encoder demodulation has been implemented and packaged in separate IP-cores. A controller has been selected and the electrical interface between the controller and the inverter has been designed.

Measurement of current and voltage has been evaluated and methods of signal conditioning has been considered. Two methods of analog to digital conversion is considered, with different properties with regard to bandwidth and resolution.

The selected inverter topology is based on insulated gate bipolar transistors as the switching element. The selection of the correct gate driver for the inverter,

dimensioning of the gate resistors, and blanking time has been considered. The dimensioning and control of the brake chopper is considered. Thermal calculations and dimensioning of heat sinks to support the cooling requirements is performed.

Although it is attempted to cover as many practical aspects as possible for the considered implementation of the motor drive, it is important to emphasise that there is still a lot of open issues remaining. Additionally more laboratory experiments are required to verify the operation of the proposed algorithms for different operating scenarios.

It is the authors' hope that the developed software, and hardware will be a useful base platform for future research and development.

Suggestions for further work

There are several improvements that could be made on the hardware side of the motor drive. A simple op-amp based temperature monitor, pre-charge controller, and brake chopper are suggested in this project. These safety functions should remain separate from the main controller, but could be improved by using a microcontroller based design. In particular the increased flexibility, and the ability to log and timestamp any events, could aid in future research and development.

The power circuit board supplied by Semikron does not support the full load capability of the inverter. A new board including a DC link filter for rated operation, should be designed. Although regulated DC power supplies are available for laboratory experiments, it is important to test the drive under the same operating conditions as it will experience in industry.

The use of active front end to replace the brake chopper could be investigated. And whether it should be implemented on a separate controller for safety reasons, or if it could run on the same controller as the motor drive.

The use and influence of snubber circuits on the inverter should be investigated.

A graphical user interface could be designed to allow easy interaction from the user to the implemented controller. This is particularity useful if the platform is to be used in the regular education. The users will be able to focus on evaluating the performance of different algorithms, without having to consider all the low level aspects of the implementation. This could preferably be based upon the work of a separate project (chapter 5.2).

The evaluated motor is of the surface mounted magnet type, and thus it does not exhibit saliency. A comparison with the salient pole machine could be made, and additionally the salient pole machine will make different estimation algorithms viable.

Although the drive has been designed with permanent magnet synchronous motors in mind, only software changes are required in order to implement drives for several

other common motor types. Of particular interest is the induction motor, and the synchronous reluctance motor. Some of the algorithms proposed in this project could be adapted to work on those motors, and many others are available. Several ways to achieve this is proposed in literature, some of which is included in the references for the respective chapters.

On the software side there is also several estimation algorithms that could be investigated further. The methods based on signal injection has not been covered, and the estimation of the initial rotor position is important in applications where the load does not permit initial rotor alignment.

The difference in performance of direct torque control utilizing, and not utilizing the zero voltage vectors could be investigated. More advanced model predictive control, including speed control, and switching frequency limitation could be investigated.

Part V

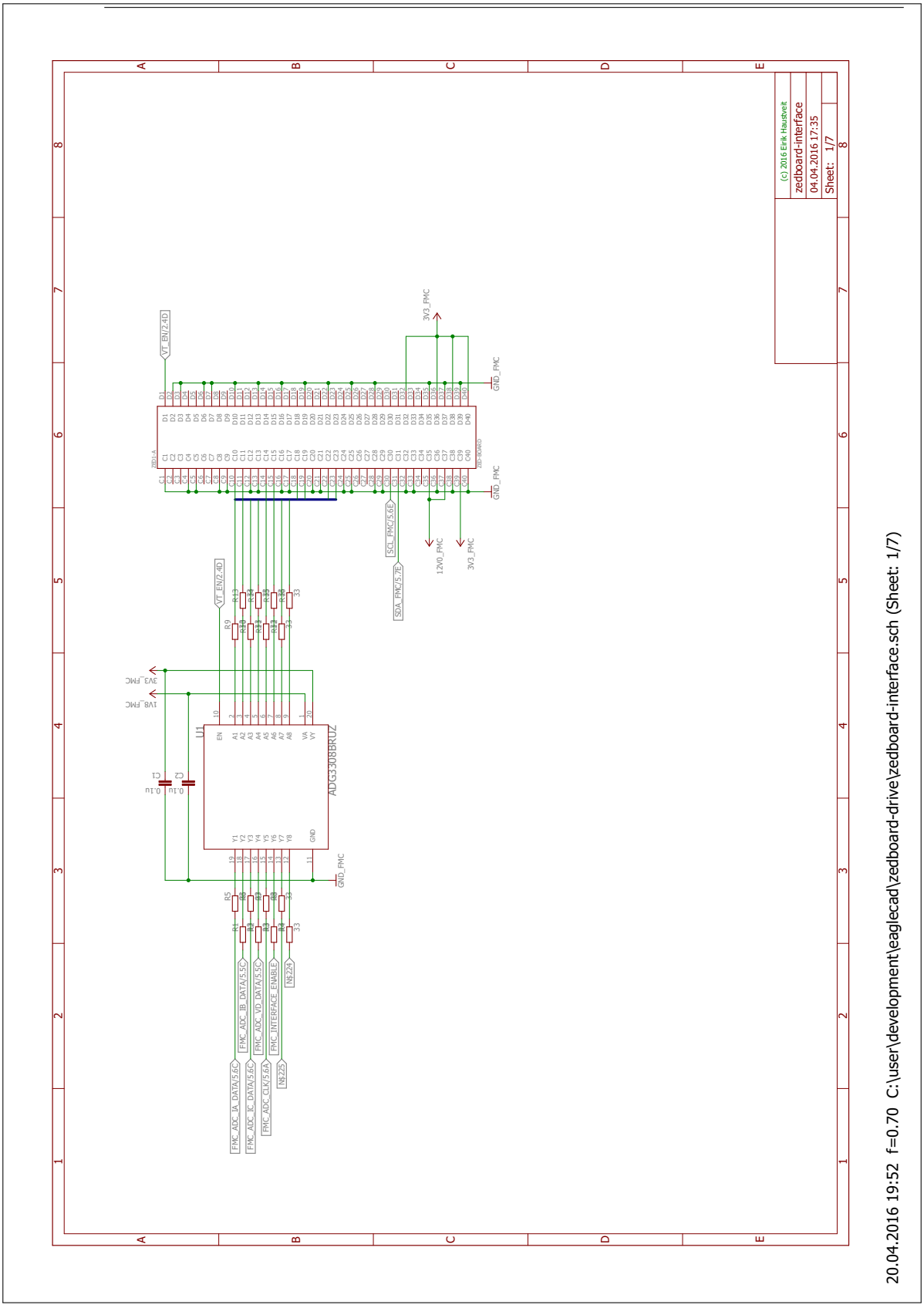
Appendix

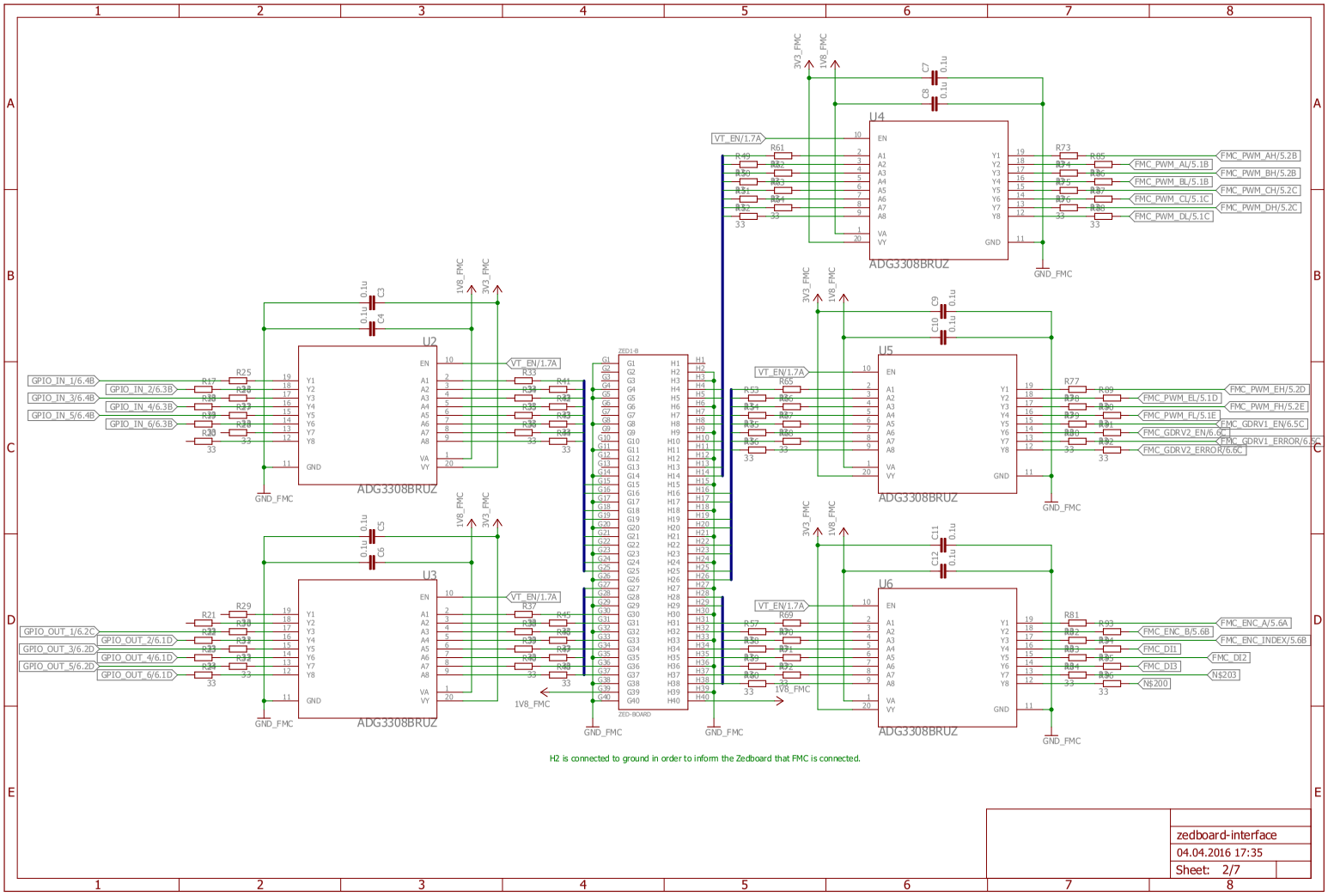
Appendix A

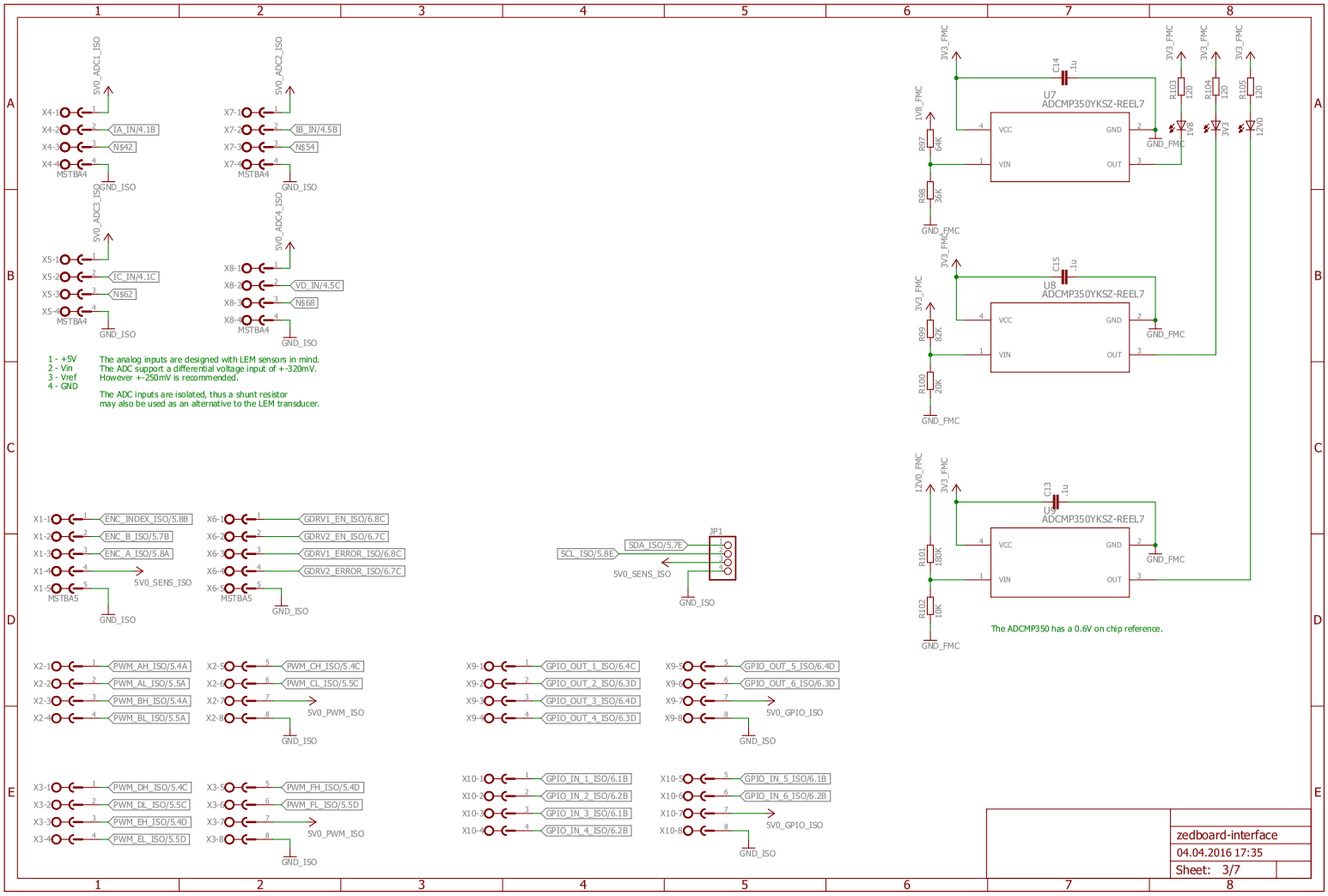
Zedboard isolation interface

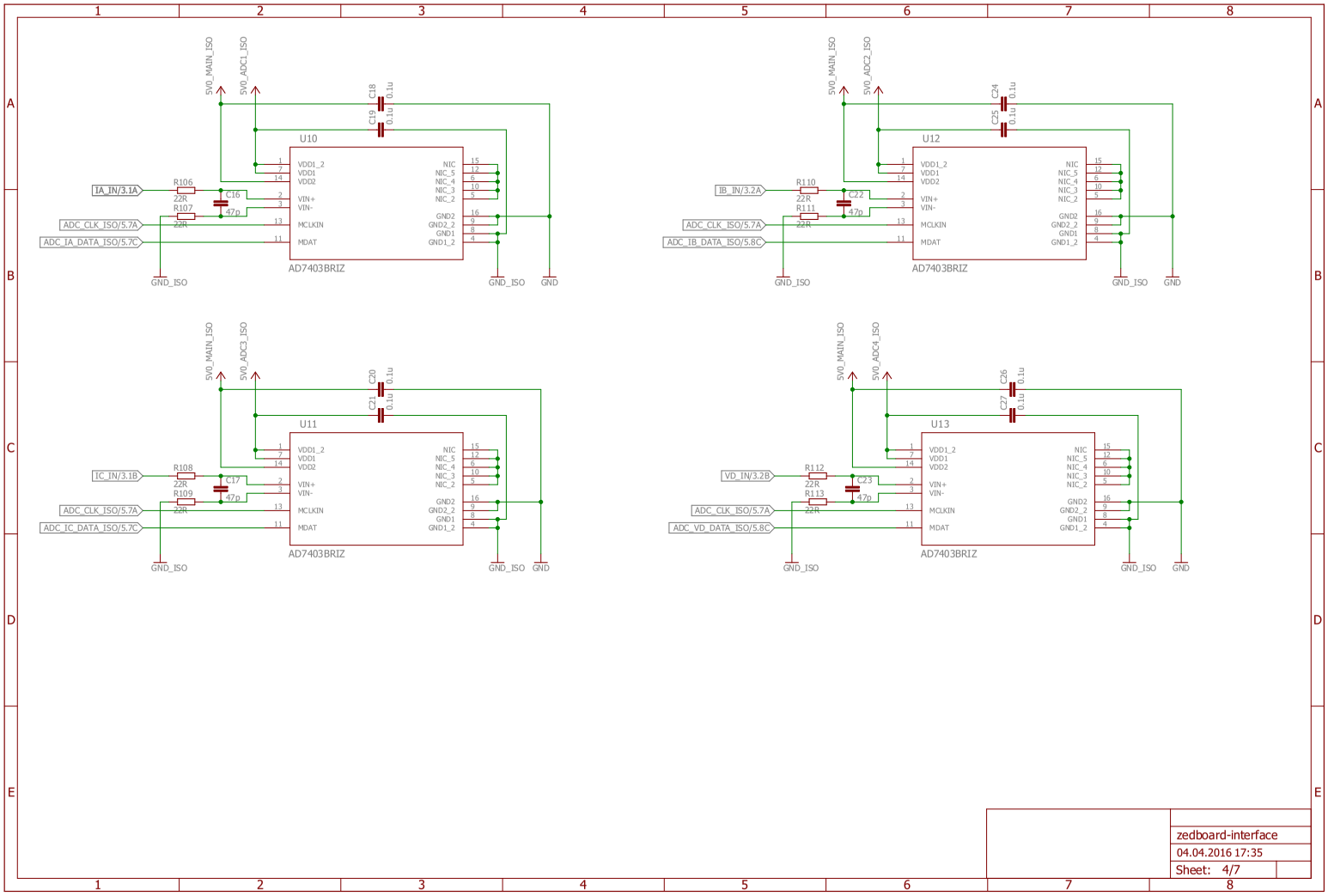
The following appendix contains the design of the Zedboard isolation interface.

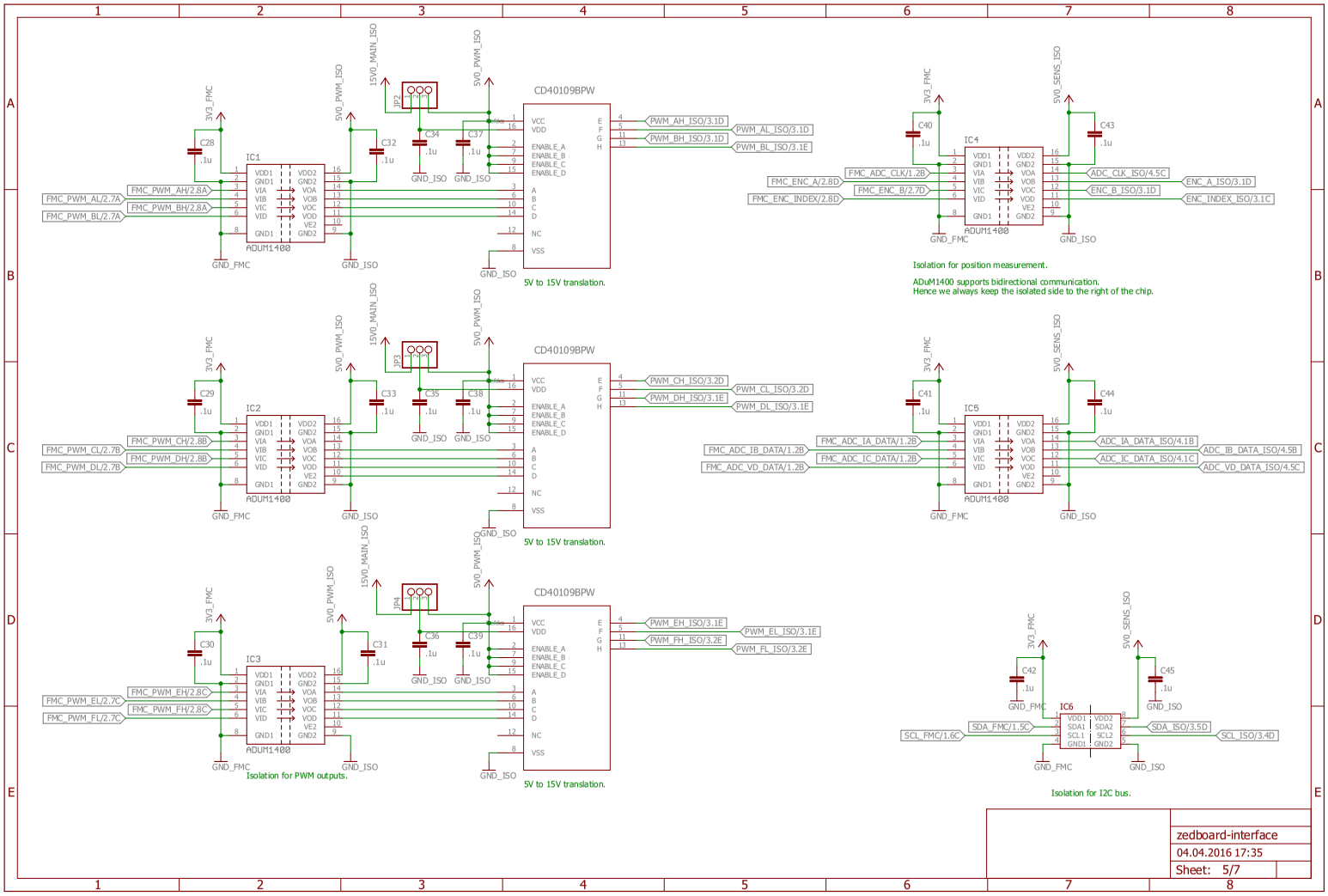
A.1 Isolation interface schematic

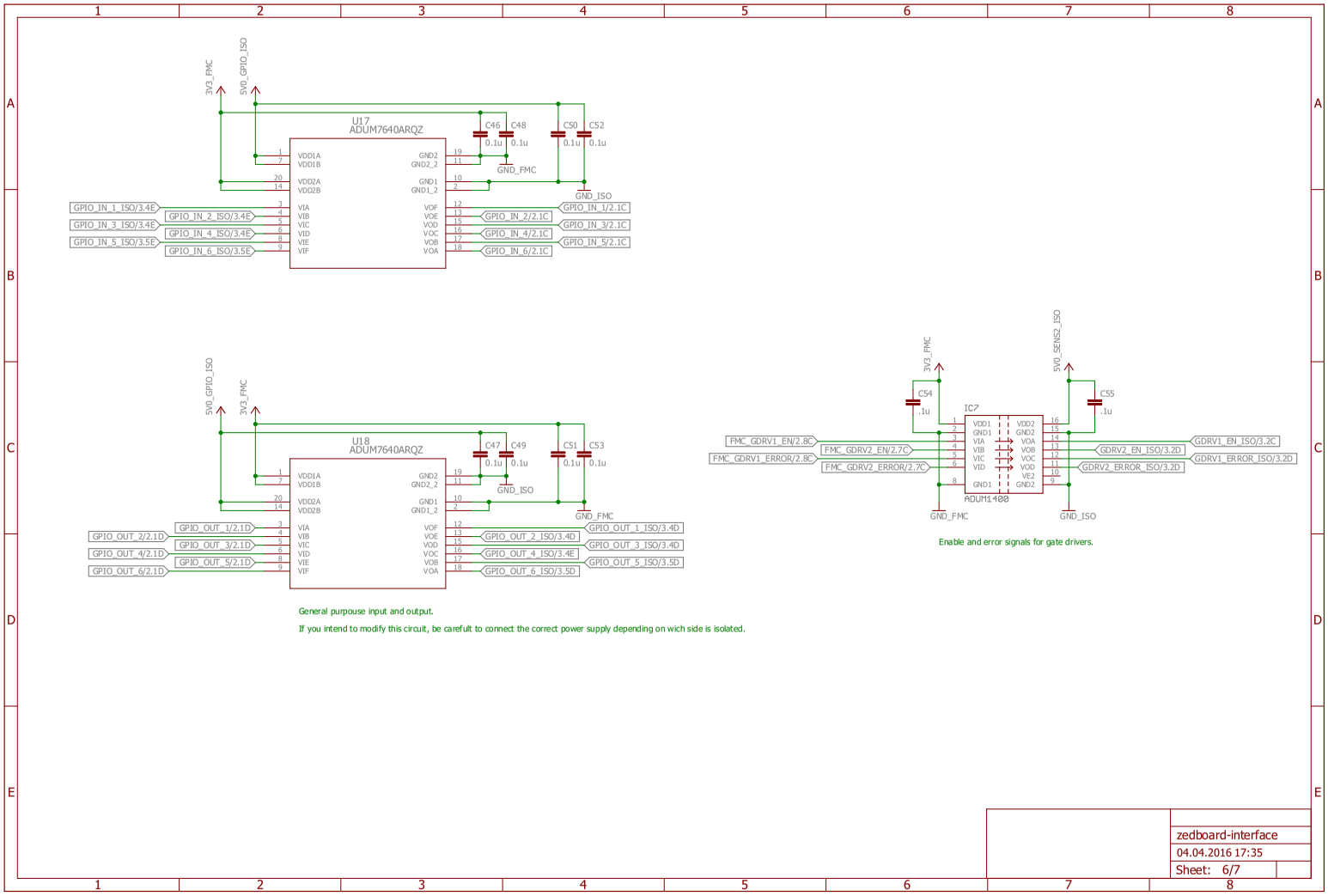


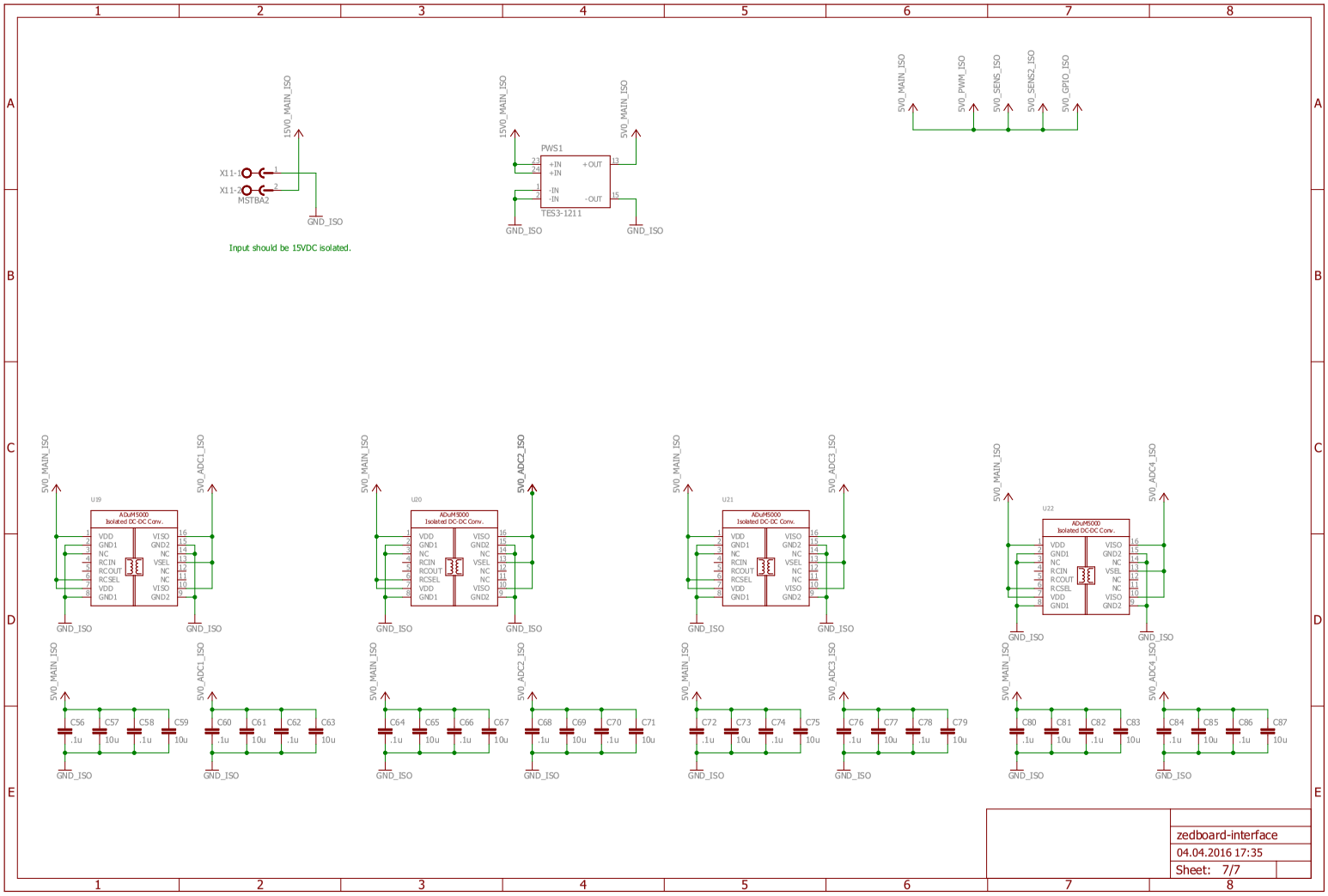

U2
U3
U4
U5
U6
ADG3308BRUZ
ZED1-B
ZED-BOARD
VT_EN/1.7A
1V8_FMC
3V3_FMC
GND_FMC
GPIO_IN_1/6.4B
GPIO_IN_2/6.3B
GPIO_IN_3/6.4B
GPIO_IN_4/6.3B
GPIO_IN_5/6.4B
GPIO_IN_6/6.3B
GPIO_OUT_1/6.2C
GPIO_OUT_2/6.1D
GPIO_OUT_3/6.2D
GPIO_OUT_4/6.1D
GPIO_OUT_5/6.2D
GPIO_OUT_6/6.1D
FMC_PWM_AH/5.2B
FMC_PWM_AL/5.1B
FMC_PWM_BH/5.2B
FMC_PWM_BL/5.1B
FMC_PWM_CH/5.2C
FMC_PWM_CL/5.1C
FMC_PWM_DH/5.2C
FMC_PWM_DL/5.1C
FMC_PWM_EH/5.2D
FMC_PWM_EL/5.1D
FMC_PWM_FH/5.2E
FMC_PWM_FL/5.1E
FMC_GDRV1_EN/6.5C
FMC_GDRV2_EN/6.6C
FMC_GDRV1_ERROR/6.5C
FMC_GDRV2_ERROR/6.6C
FMC_ENC_A/5.6A
FMC_ENC_B/5.6B
FMC_ENC_INDEX/5.6B
FMC_DI1
FMC_DI2
FMC_DI3
N$200
N$203
H2 is connected to ground in order to inform the Zedboard that FMC is connected.
zedboard-interface
04.04.2016 17:35
Sheet: 2/7


1 2 3 4 5 6 7 8
A B C D E
5V0_ADC1_ISO
X4-1 X4-2 X4-3 X4-4
IA_IN/4.1B
N$42
MSTBA4
GND_ISO
5V0_ADC2_ISO
X7-1 X7-2 X7-3 X7-4
IB_IN/4.5B
N$54
MSTBA4
GND_ISO
5V0_ADC3_ISO
X5-1 X5-2 X5-3 X5-4
IC_IN/4.1C
N$62
MSTBA4
GND_ISO
5V0_ADC4_ISO
X8-1 X8-2 X8-3 X8-4
VD_IN/4.5C
N$68
MSTBA4
GND_ISO
1 - +5V
2 - Vin
3 - Vref
4 - GND
The analog inputs are designed with LEM sensors in mind.
The ADC support a differential voltage input of +-320mV.
However +-250mV is recommended.
The ADC inputs are isolated, thus a shunt resistor
may also be used as an alternative to the LEM transducer.
3V3_FMC
C14 .1u
U7
ADCMP350YKSZ-REEL7
VCC GND VIN OUT
R103 120 1V8
R104 120 3V3
R105 120 12V0
1V8_FMC
R97 64K
R98 36K
GND_FMC
C15 .1u
U8
ADCMP350YKSZ-REEL7
R99 82K
R100 20K
C13 .1u
U9
ADCMP350YKSZ-REEL7
12V0_FMC
R101 180K
R102 10K
The ADCMP350 has a 0.6V on chip reference.
X1-1 X1-2 X1-3 X1-4 X1-5
ENC_INDEX_ISO/5.8B
ENC_B_ISO/5.7B
ENC_A_ISO/5.8A
5V0_SENS_ISO
MSTBA5
GND_ISO
X6-1 X6-2 X6-3 X6-4 X6-5
GDRV1_EN_ISO/6.8C
GDRV2_EN_ISO/6.7C
GDRV1_ERROR_ISO/6.8C
GDRV2_ERROR_ISO/6.7C
MSTBA5
GND_ISO
SDA_ISO/5.7E
SCL_ISO/5.8E
JP1
5V0_SENS_ISO
GND_ISO
X2-1 X2-2 X2-3 X2-4
PWM_AH_ISO/5.4A
PWM_AL_ISO/5.5A
PWM_BH_ISO/5.4A
PWM_BL_ISO/5.5A
X2-5 X2-6 X2-7 X2-8
PWM_CH_ISO/5.4C
PWM_CL_ISO/5.5C
5V0_PWM_ISO
GND_ISO
X9-1 X9-2 X9-3 X9-4
GPIO_OUT_1_ISO/6.4C
GPIO_OUT_2_ISO/6.3D
GPIO_OUT_3_ISO/6.4D
GPIO_OUT_4_ISO/6.3D
X9-5 X9-6 X9-7 X9-8
GPIO_OUT_5_ISO/6.4D
GPIO_OUT_6_ISO/6.3D
5V0_GPIO_ISO
GND_ISO
X3-1 X3-2 X3-3 X3-4
PWM_DH_ISO/5.4C
PWM_DL_ISO/5.5C
PWM_EH_ISO/5.4D
PWM_EL_ISO/5.5D
X3-5 X3-6 X3-7 X3-8
PWM_FH_ISO/5.4D
PWM_FL_ISO/5.5D
5V0_PWM_ISO
GND_ISO
X10-1 X10-2 X10-3 X10-4
GPIO_IN_1_ISO/6.1B
GPIO_IN_2_ISO/6.2B
GPIO_IN_3_ISO/6.1B
GPIO_IN_4_ISO/6.2B
X10-5 X10-6 X10-7 X10-8
GPIO_IN_5_ISO/6.1B
GPIO_IN_6_ISO/6.2B
5V0_GPIO_ISO
GND_ISO
zedboard-interface
04.04.2016 17:35
Sheet: 3/7


1
2
3
4
5
6
7
8
A
B
C
D
E
5V0_MAIN_ISO
5V0_ADC1_ISO
C18
0.1u
C19
0.1u
U10
VDD1_2
VDD1
VDD2
VIN+
VIN-
MCLKIN
MDAT
NIC
NIC_5
NIC_4
NIC_3
NIC_2
GND2
GND2_2
GND1
GND1_2
AD7403BRIZ
IA_IN/3.1A
R106
22R
C16
47p
R107
22R
ADC_CLK_ISO/5.7A
ADC_IA_DATA_ISO/5.7C
GND_ISO
GND
5V0_MAIN_ISO
5V0_ADC2_ISO
C24
0.1u
C25
0.1u
U12
AD7403BRIZ
IB_IN/3.2A
R110
22R
C22
47p
R111
22R
ADC_CLK_ISO/5.7A
ADC_IB_DATA_ISO/5.8C
GND_ISO
GND
5V0_MAIN_ISO
5V0_ADC3_ISO
C20
0.1u
C21
0.1u
U11
AD7403BRIZ
IC_IN/3.1B
R108
22R
C17
47p
R109
22R
ADC_CLK_ISO/5.7A
ADC_IC_DATA_ISO/5.7C
GND_ISO
GND
5V0_MAIN_ISO
5V0_ADC4_ISO
C26
0.1u
C27
0.1u
U13
AD7403BRIZ
VD_IN/3.2B
R112
22R
C23
47p
R113
22R
ADC_CLK_ISO/5.7A
ADC_VD_DATA_ISO/5.8C
GND_ISO
GND
zedboard-interface
04.04.2016 17:35
Sheet: 4/7

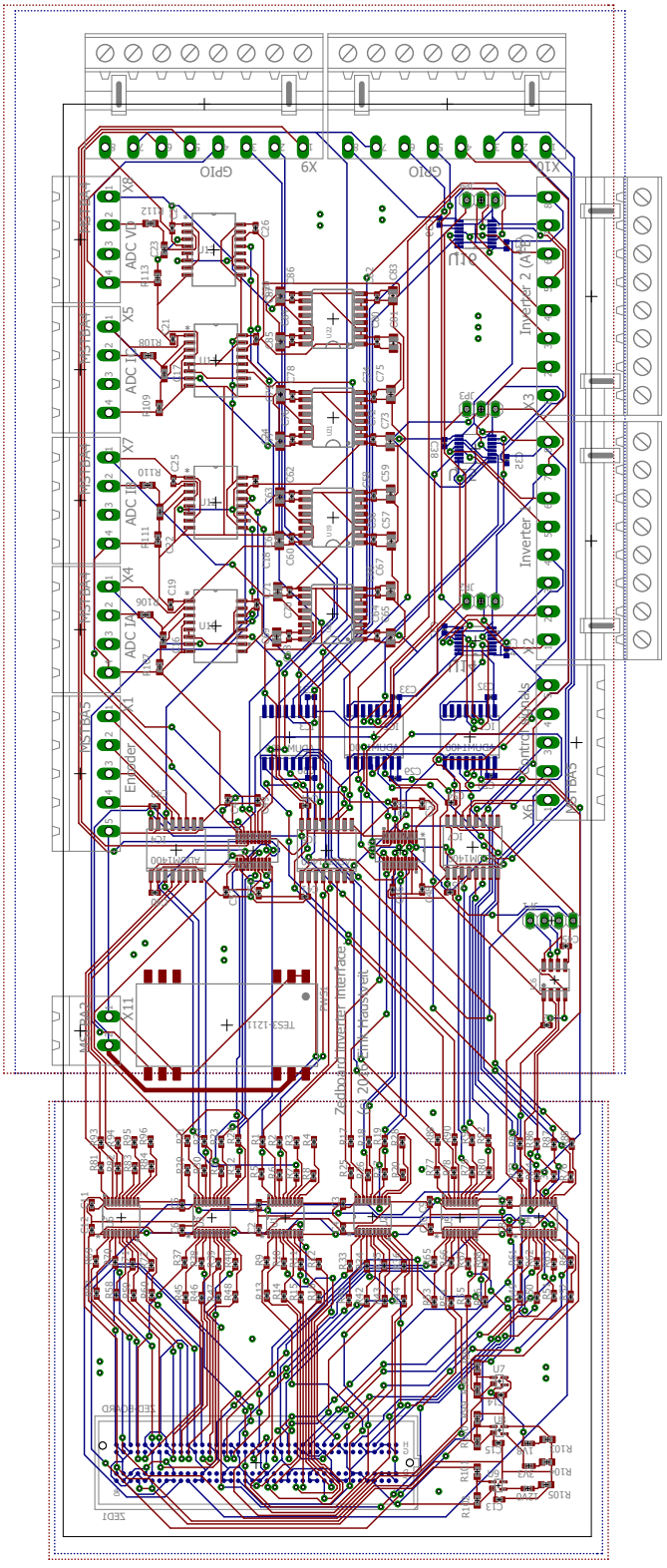

1
2
3
4
5
6
7
8
A
B
C
D
E
3V3_FMC
5V0_PWM_ISO
15V0_MAIN_ISO
5V0_PWM_ISO
JP2
C28
.1u
IC1
ADUM1400
C32
.1u
C34
.1u
C37
.1u
GND_ISO
GND_ISO
CD40109BPW
VCC
VDD
ENABLE_A
ENABLE_B
ENABLE_C
ENABLE_D
NC
VSS
GND_ISO
FMC_PWM_AL/2.7A
FMC_PWM_AH/2.8A
FMC_PWM_BH/2.8A
FMC_PWM_BL/2.7A
GND_FMC
GND_ISO
PWM_AH_ISO/3.1D
PWM_AL_ISO/3.1D
PWM_BH_ISO/3.1D
PWM_BL_ISO/3.1E
5V to 15V translation.
3V3_FMC
5V0_SENS_ISO
C40
.1u
IC4
ADUM1400
C43
.1u
FMC_ADC_CLK/1.2B
FMC_ENC_A/2.8D
FMC_ENC_B/2.7D
FMC_ENC_INDEX/2.8D
ADC_CLK_ISO/4.5C
ENC_A_ISO/3.1D
ENC_B_ISO/3.1D
ENC_INDEX_ISO/3.1C
GND_FMC
GND_ISO
Isolation for position measurement.
ADuM1400 supports bidirectional communication.
Hence we always keep the isolated side to the right of the chip.
3V3_FMC
5V0_PWM_ISO
15V0_MAIN_ISO
5V0_PWM_ISO
JP3
C29
.1u
IC2
ADUM1400
C33
.1u
C35
.1u
C38
.1u
GND_ISO
GND_ISO
CD40109BPW
VCC
VDD
ENABLE_A
ENABLE_B
ENABLE_C
ENABLE_D
NC
VSS
GND_ISO
FMC_PWM_CL/2.7B
FMC_PWM_CH/2.8B
FMC_PWM_DH/2.8B
FMC_PWM_DL/2.7B
GND_FMC
GND_ISO
PWM_CH_ISO/3.2D
PWM_CL_ISO/3.2D
PWM_DH_ISO/3.1E
PWM_DL_ISO/3.1E
5V to 15V translation.
3V3_FMC
5V0_SENS_ISO
C41
.1u
IC5
ADUM1400
C44
.1u
FMC_ADC_IB_DATA/1.2B
FMC_ADC_VD_DATA/1.2B
FMC_ADC_IA_DATA/1.2B
FMC_ADC_IC_DATA/1.2B
ADC_IA_DATA_ISO/4.1B
ADC_IB_DATA_ISO/4.5B
ADC_IC_DATA_ISO/4.1C
ADC_VD_DATA_ISO/4.5C
GND_FMC
GND_ISO
3V3_FMC
5V0_PWM_ISO
15V0_MAIN_ISO
5V0_PWM_ISO
JP4
C30
.1u
IC3
ADUM1400
C31
.1u
C36
.1u
C39
.1u
GND_ISO
GND_ISO
CD40109BPW
VCC
VDD
ENABLE_A
ENABLE_B
ENABLE_C
ENABLE_D
NC
VSS
GND_ISO
FMC_PWM_EL/2.7C
FMC_PWM_EH/2.8C
FMC_PWM_FH/2.8C
FMC_PWM_FL/2.7C
GND_FMC
GND_ISO
PWM_EH_ISO/3.1E
PWM_EL_ISO/3.1E
PWM_FH_ISO/3.2E
PWM_FL_ISO/3.2E
5V to 15V translation.
Isolation for PWM outputs.
3V3_FMC
5V0_SENS_ISO
C42
.1u
C45
.1u
IC6
SCL_FMC/1.6C
SDA_FMC/1.5C
SDA_ISO/3.5D
SCL_ISO/3.4D
GND_FMC
GND_ISO
Isolation for I2C bus.
zedboard-interface
04.04.2016 17:35
Sheet: 5/7


3V3_FMC
5V0_GPIO_ISO
U17
ADUM7640ARQZ
C46 0.1u
C48 0.1u
C50 0.1u
C52 0.1u
GND_FMC
GND_ISO
GPIO_IN_1_ISO/3.4E
GPIO_IN_2_ISO/3.4E
GPIO_IN_3_ISO/3.4E
GPIO_IN_4_ISO/3.4E
GPIO_IN_5_ISO/3.5E
GPIO_IN_6_ISO/3.5E
GPIO_IN_1/2.1C
GPIO_IN_2/2.1C
GPIO_IN_3/2.1C
GPIO_IN_4/2.1C
GPIO_IN_5/2.1C
GPIO_IN_6/2.1C
5V0_GPIO_ISO
3V3_FMC
U18
ADUM7640ARQZ
C47 0.1u
C49 0.1u
C51 0.1u
C53 0.1u
GND_ISO
GND_FMC
GPIO_OUT_1/2.1D
GPIO_OUT_2/2.1D
GPIO_OUT_3/2.1D
GPIO_OUT_4/2.1D
GPIO_OUT_5/2.1D
GPIO_OUT_6/2.1D
GPIO_OUT_1_ISO/3.4D
GPIO_OUT_2_ISO/3.4D
GPIO_OUT_3_ISO/3.4D
GPIO_OUT_4_ISO/3.4E
GPIO_OUT_5_ISO/3.5D
GPIO_OUT_6_ISO/3.5D
General purpouse input and output.
If you intend to modify this circuit, be carefult to connect the correct power supply depending on wich side is isolated.
3V3_FMC
5V0_SENS2_ISO
C54 .1u
C55 .1u
IC7
ADUM1400
FMC_GDRV1_EN/2.8C
FMC_GDRV2_EN/2.7C
FMC_GDRV1_ERROR/2.8C
FMC_GDRV2_ERROR/2.7C
GDRV1_EN_ISO/3.2C
GDRV2_EN_ISO/3.2D
GDRV1_ERROR_ISO/3.2D
GDRV2_ERROR_ISO/3.2D
GND_FMC
GND_ISO
Enable and error signals for gate drivers.
zedboard-interface
04.04.2016 17:35
Sheet: 6/7



20.04.2016 19:52 f=0.70 C:\user\development\eaglecad\zedboard-drive\zedboard-interface.sch (Sheet: 7/7)

A.2 Circuit board layout

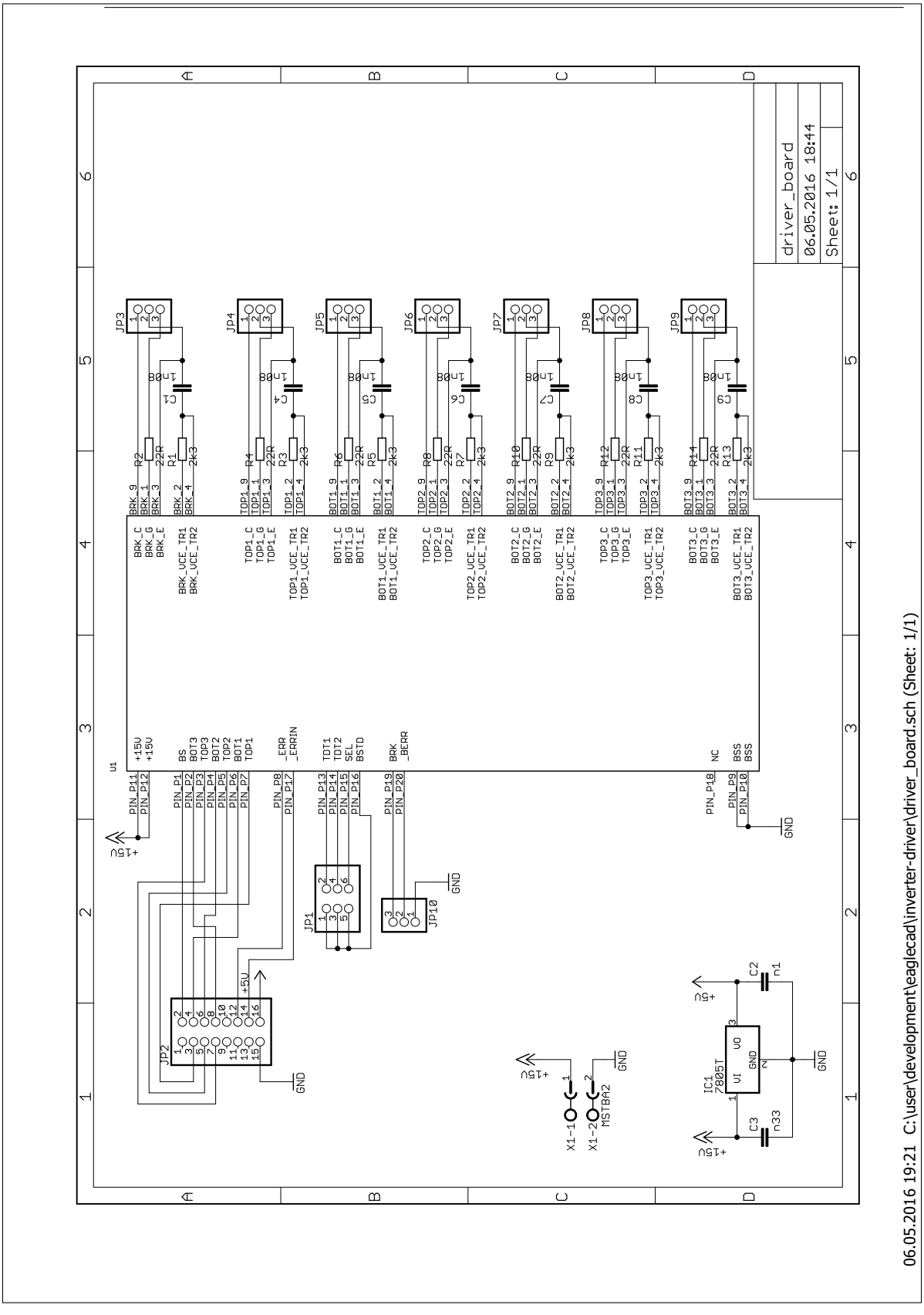


Appendix B

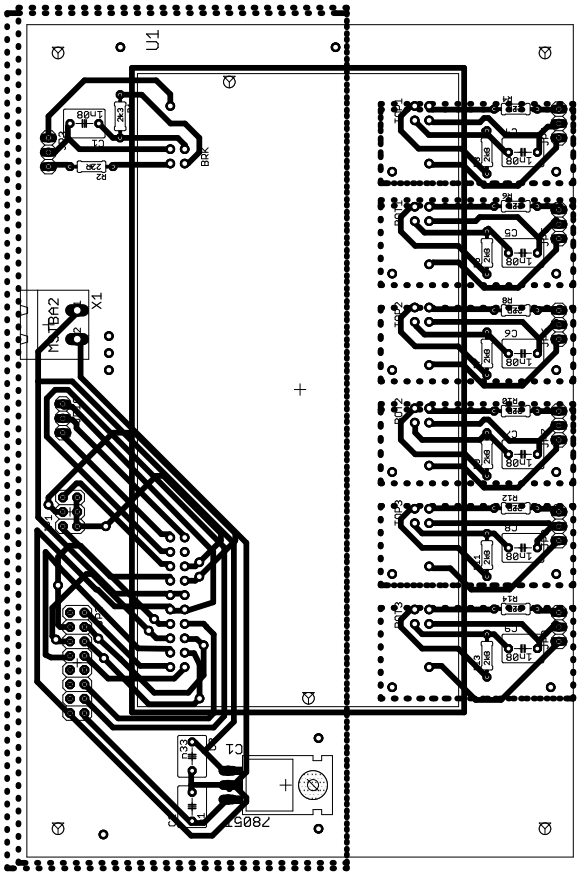
Gate driver circuit board

The gate driver circuit board is designed to support the SKHI 71 R gate driver from Semikron. In addition to the required components, a *7805* linear voltage regulator is added in order to supply 5 V at the controller interface. This is done in order to comply with a standard interface that has been used previously at Norges teknisk-naturvitenskapelige universitet (NTNU).

B.1 Gate driver board schematic



B.2 Board layout



06.05.2016 19:23 C:\user\development\eaglecad\inverter-driver\driver_board.brd

Appendix C

Thermal calculations

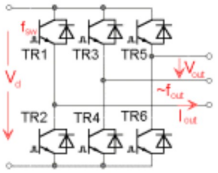
C.1 Thermal calculations for a 15kW motor.

SEMISEL™

SEMIKRON innovation+service

Project:

Topology	DC/AC
Circuit	Inverter 3 Phases



Circuit:

V_d	540 V
V_{out}	360 V
I_{out}	28 A
P_{out}	15 kW
f_{out}	100 Hz
cos(ϕ)	0.85
f_{sw}	10 kHz
$V_{min\ out}$	42 V
$f_{min\ out}$	2 Hz
Overload factor	2
Overload duration	10 sec

Overload characteristic

Device :

Product line	**MiniSKiiP**
Device	**SKiiP35NAB126V1***
Use maximum values	Yes
Max. junction temparature	150°C

Transistor

E_{tr}	= 12.6 mJ (@600V)
$V_{CE0.125}$	= 1.1 V
$r_{C.125}$	= 26 mOhm
$V_{CE.sat}$	= 2.40 V
I_c	= 50.00 A
$R_{th(j-c)}$	= 0.55 K/W
$R_{th(c-s)}$	= 0 K/W

Diode

E_d	= 4.7 mJ
$V_{T0.125}$	= 0.9 V
$r_{T.125}$	= 18 mOhm
V_f	= 1.80 V
I_f	= 50.00 A
$R_{th(j-c)}$	= 1 K/W

* - Not for new design
Data set from 2008/09/24

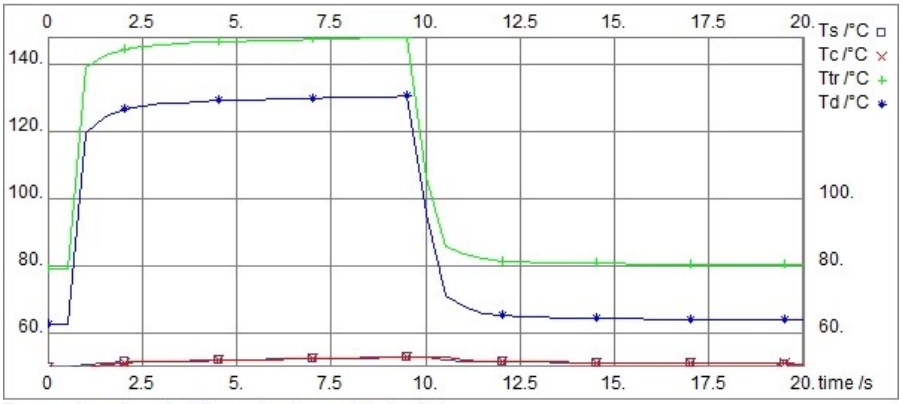
Cooling:

Ambient temperature	40°C
Number of switches per heat sink	6
Number of parallel devices on the same heat sink	1
Additional power source at this heat sink	0 W
Predefined SK-Heat Sink	P16_300_16B
Correction factor	1
Forced Air Cooling, Flow Rate:	295 m^3/h
$R_{th(s-a)}$	0.031 K/W

Calculated losses and temperatures with rated current, at overload and at $f_{min\ out}$:

	Rated current	Overload	f_{min} and Overload
$P_{cond\ tr}$	20 W	59 W	36 W
$P_{sw\ tr}$	23 W	53 W	51 W
P_{tr}	43 W	113 W	87 W
$P_{cond\ d}$	2.42 W	6.14 W	23 W
$P_{sw\ d}$	8.16 W	14 W	17 W
P_d	11 W	20 W	40 W
P_{tot}	324 W	798 W	760 W
	Average Values	Average Values	Maximum Values
T_s	50 °C	54 °C	53 °C

T_c	50 °C	54 °C	53 °C
T_{tr}	74 °C	115 °C	148 °C
T_d	61 °C	74 °C	131 °C



Temperature characteristic overload current during fmin

Evaluation:

Recommendation by SEMIKRON: Do not use SEMIKRON devices over 125 °C

Device driver suggestion

Name	Iout(av) /mA	Îout /A	Visol /kV	Vce max /V	Rgmin / Ohm	Channels
3x SKHI23/12 R	50	8	2.5	1200	2.7	2
1x SKHI61 R	20	2	2.5	900	10.0	6
3x SKYPER 32 R or SKYPER 32PRO R	50	15	4.0	1200	1.5	2

Additional Characteristics at given nominal operation conditions with one free parameter - X:

None selected

Appendix D

Datasheets

The following appendix contains the most relevant parts of the datasheets used when designing the laboratory platform. Please refer to the respective manufacturers for more detailed information.

D.1 Current transducer

LEM

Current Transducer LTS 15-NP

For the electronic measurement of currents: DC, AC, pulsed, mixed with galvanic isolation between the primary circuit (high power) and the secondary circuit (electronic circuit).

16058

I_{PN} =± 15 At

Features

- Closed loop (compensated) multi-range current transducer using the Hall effect
- Unipolar voltage supply
- Isolated plastic case recognized according to UL 94-V0
- Compact design for PCB mounting
- Incorporated measuring resistance
- Extended measuring range.

Advantages

- Excellent accuracy
- Very good linearity
- Very low temperature drift
- Optimized response time
- Wide frequency bandwidth
- No insertion losses
- High immunity to external interference
- Current overload capability.

Applications

- AC variable speed drives and servo motor drives
- Static converters for DC motor drives
- Battery supplied applications
- Uninterruptible Power Supplies (UPS)
- Switched Mode Power Supplies (SMPS)
- Power supplies for welding applications.

Application domain

- Industrial.

Electrical data

I_{PN}	Primary nominal current rms		15	At
I_{PM}	Primary current, measuring range		0 .. ± 48	At
$\hat{I}_P$	Overload capability		250	At
V_{OUT}	Output voltage (Analog) @ I_P		2.5±(0.625·I_P/I_{PN})	V
	I_P = 0		2.5 [1]	V
G	Sensitivity		41.6	mV/A
N_S	Number of secondary turns (± 0.1 %)		2000	
R_L	Load resistance		≥ 2	kΩ
R_{IM}	Internal measuring resistance (± 0.5 %)		83.33	Ω
TCR_{IM}	Temperature coefficient of R_{IM}		< 50	ppm/K
V_C	Supply voltage (± 5 %)		5	V
I_C	Current consumption @ V_C = 5 V	Typ	28+I_S [2]+(V_{OUT}/R_L)	mA

Accuracy - Dynamic performance data

X	Accuracy @ I_{PN}, T_A = 25°C		± 0.2		%
	Accuracy with R_{IM} @ I_{PN}, T_A = 25°C		± 0.7		%
ε_L	Linearity error		< 0.1		%
			Typ	Max	
TCV_{OUT}	Temperature coefficient of V_{OUT} @ I_P= 0	- 10°C .. + 85°C	65	120	ppm/K
		- 40°C .. - 10°C	65	170	ppm/K
TCG	Temperature coefficient of G	- 40°C .. + 85°C		50 [3]	ppm/K
V_{OM}	Magnetic offset voltage @ I_P = 0,				
	after an overload of 3 x I_{PN}			± 0.5	mV
	5 x I_{PN}			± 2.0	mV
	10 x I_{PN}			± 2.0	mV
t_{ra}	Reaction time @ 10 % of I_{PN}		< 100		ns
t_r	Response time to 90 % of I_{PN} step		< 400		ns
di/dt	di/dt accurately followed		> 35		A/µs
BW	Frequency bandwidth (0 .. - 0.5 dB)		DC .. 100		kHz
	(- 0.5.. 1 dB)		DC .. 200		kHz

General data

T_A	Ambient operating temperature	- 40 .. + 85	°C
T_S	Ambient storage temperature	- 40 .. + 100	°C
m	Mass	10	g
	Standards	EN 50178: 1997	
		IEC 60950-1: 2001	

Notes: [1] Absolute value @ T_A = 25°C, 2.475 < V_{OUT} < 2.525

[2] I_S = I_P/N_S

[3] Only due to TCR_{IM}.

Current Transducer LTS 15-NP

Isolation characteristics

V_d	Rms voltage for AC isolation test, 50 Hz, 1 min	3	kV
$\hat{V}_w$	Impulse withstand voltage 1.2/50 µs	> 8 Min	kV
V_e	Rms voltage for partial discharge extinction @ 10pC	> 1.5 Min	kV
dCp	Creepage distance [1]	15.5	mm
dCl	Clearance distance [2]	6.35	mm
CTI	Comparative Tracking Index (group IIIa)	175	

Notes: [1] On housing
[2] On PCB with soldering pattern UTEC93-703.

Applications examples

According to EN 50178 and IEC 61010-1 standards and following conditions:

- Over voltage category OV 3
- Pollution degree PD2
- Non-uniform field

	EN 50178	IEC 61010-1
dCp, **dCl**, $\hat{V}_w$	Rated insulation voltage	Nominal voltage
Single insulation	600 V	600 V
Reinforced insulation	300 V	300 V

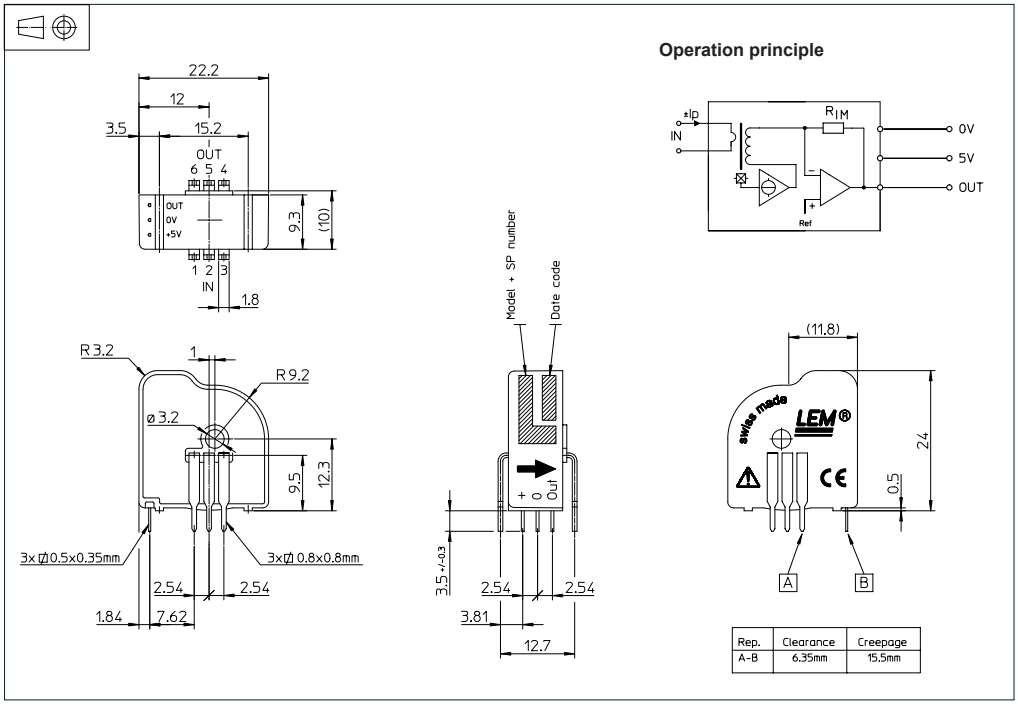
Safety

This transducer must be used in electric/electronic equipment with respect to applicable standards and safety requirements in accordance with the manufacturer's operating instructions.

Caution, risk of electrical shock

When operating the transducer, certain parts of the module can carry hazardous voltage (eg. primary busbar, power supply).
Ignoring this warning can lead to injury and/or cause serious damage.
This transducer is a build-in device, whose conducting parts must be inaccessible after installation.
A protective housing or additional shield could be used.
Main supply must be able to be disconnected.

Dimensions LTS 15-NP (in mm)



Number of primary turns	Primary nominal current rms I_{PN} [A]	Nominal output voltage V_{OUT} [V]	Primary resistance R_p [mΩ]	Primary insertion inductance L_p [µH]	Recommended connections
1	± 15	2.5 ± 0.625	0.18	0.013	OUT 6 5 4 / IN 1 2 3
2	± 7.5	2.5 ± 0.625	0.81	0.05	OUT 6 5 4 / IN 1 2 3
3	± 5	2.5 ± 0.625	1.62	0.12	OUT 6 5 4 / IN 1 2 3

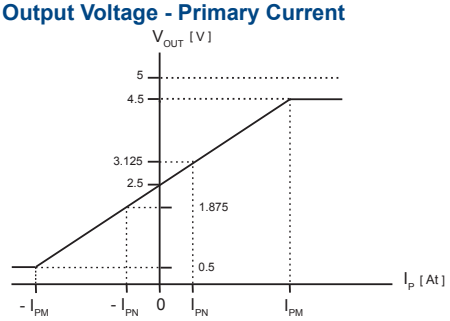
Mechanical characteristics

- General tolerance ± 0.2 mm
- Fastening & connection of primary 6 pins 0.8 x 0.8 mm
  Recommended PCB hole 1.3 mm
- Fastening & connection of secondary 3 pins 0.5 x 0.35 mm
  Recommended PCB hole 0.8 mm
- Additional primary through-hole Ø 3.2 mm

Remarks

- V_{OUT} swings above 2.5 V when I_p flows from terminals 1, 2, 3 to terminals 6, 5, 4 (with the arrow).
- Temperature of the primary jumper should not exceed 100°C.

Output Voltage - Primary Current



D.2 Delta-sigma modulator

ANALOG DEVICES

16-Bit, Isolated Sigma-Delta Modulator

Data Sheet **AD7403**

FEATURES

- 5 MHz to 20 MHz external clock input rate
- 16 bits, no missing codes
- Signal-to-noise ratio (SNR): 88 dB typical
- Effective number of bits (ENOB): 14.2 bits typical
- Offset drift vs. temperature
  - AD7403: 1.6 µV/°C typical
  - AD7403-8: 2 µV/°C typical
- On-board digital isolator
- On-board reference
- Full-scale analog input range: ±320 mV
- Operating range
  - AD7403: −40°C to + 125°C
  - AD7403-8: −40°C to + 105°C
- High common-mode transient immunity: >25 kV/µs
- Wide-body SOIC with increased creepage package
- Slew rate limited output for low EMI
- Safety and regulatory approvals
  - UL recognition
    - 5000 V rms for 1 minute per UL 1577
  - CSA Component Acceptance Notice 5A
  - VDE Certificate of Conformity
    - DIN V VDE V 0884-10 (VDE V 0884-10):2006-12
    - V_{IORM} = 1250 V_{PEAK}

APPLICATIONS

- Shunt current monitoring
- AC motor controls
- Power and solar inverters
- Wind turbine inverters
- Data acquisition systems
- Analog-to-digital and optoisolator replacements

FUNCTIONAL BLOCK DIAGRAM

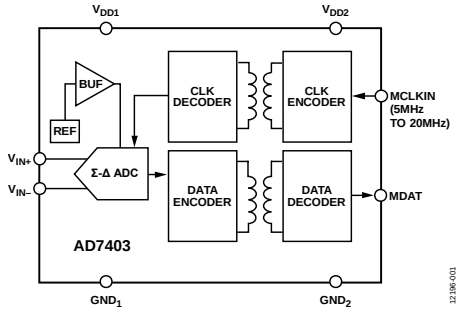


Figure 1.

GENERAL DESCRIPTION

The AD7403[1] is a high performance, second-order, Σ-Δ modulator that converts an analog input signal into a high speed, single-bit data stream, with on-chip digital isolation based on Analog Devices, Inc., *i*Coupler® technology. The device operates from a 5 V (V_{DD1}) power supply and accepts a differential input signal of ±250 mV (±320 mV full-scale). The differential input is ideally suited to shunt voltage monitoring in high voltage applications where galvanic isolation is required.

The analog input is continuously sampled by a high performance analog modulator, and converted to a ones density digital output stream with a data rate of up to 20 MHz. The original information can be reconstructed with an appropriate digital filter to achieve 88 dB signal to noise ratio (SNR) at 78.1 kSPS. The serial input/output can use a 5 V or a 3 V supply (V_{DD2}).

The serial interface is digitally isolated. High speed complementary metal oxide semiconductor (CMOS) technology, combined with monolithic transformer technology, means the on-chip isolation provides outstanding performance characteristics, superior to alternatives such as optocoupler devices. The AD7403 device is offered in a 16-lead, wide-body SOIC package and has an operating temperature range of −40°C to +125°C. The AD7403-8 device is offered in an 8-lead, wide-body SOIC package and has an operating temperature range of −40°C to +105°C.

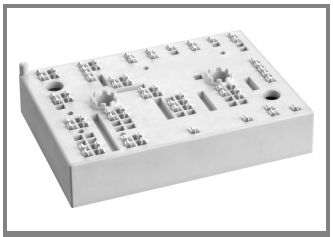
[1] Protected by U.S. Patents 5,952,849; 6,873,065; and 7,075,329.

Rev. B Document Feedback

One Technology Way, P.O. Box 9106, Norwood, MA 02062-9106, U.S.A.
Tel: 781.329.4700
Technical Support www.analog.com

D.3 IGBT module 35NAB126V10

SKiiP 35NAB126V10



MiniSKiiP® 3

3-phase bridge rectifier + brake chopper + 3-phase bridge inverter

SKiiP 35NAB126V10

Features

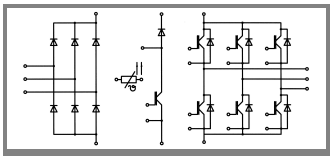
- Fast Trench IGBTs
- Robust and soft freewheeling diodes in CAL technology
- Highly reliable spring contacts for electrical connections
- UL recognised file no. E63532

Typical Applications*

- Inverter up to 28 kVA
- Typical motor power 15 kW

Remarks

- V_{CEsat} , V_F= chip level value



NAB

Absolute Maximum Ratings		T_s = 25 °C, unless otherwise specified	
Symbol	**Conditions**	**Values**	**Units**
IGBT - Inverter, Chopper			
V_{CES}		1200	V
I_C	T_s = 25 (70) °C	73 (55)	A
I_{CRM}		100	A
V_{GES}		± 20	V
T_j		- 40 ... + 150	°C
Diode - Inverter, Chopper			
I_F	T_s = 25 (70) °C	62 (46)	A
I_{FRM}		100	A
T_j		- 40 ... + 150	°C
Diode - Rectifier			
V_{RRM}		1600	V
I_F	T_s = 70 °C	67	A
I_{FSM}	t_p = 10 ms, sin 180 °, T_j = 25 °C	850	A
i^2t	t_p = 10 ms, sin 180 °, T_j = 25 °C	3600	A²s
T_j		- 40 ... + 150	°C
Module			
I_{tRMS}	per power terminal (20 A / spring)	80	A
T_{stg}		- 40 ... + 125	°C
V_{isol}	AC, 1 min.	2500	V

Characteristics		T_s = 25 °C, unless otherwise specified			
Symbol	**Conditions**	**min.**	**typ.**	**max.**	**Units**
IGBT - Inverter, Chopper					
V_{CEsat}	I_{Cnom} = 50 A, T_j = 25 (125) °C		1,7 (2)	2,1 (2,4)	V
$V_{GE(th)}$	V_{GE} = V_{CE}, I_C = 2 mA	5	5,8	6,5	V
$V_{CE(TO)}$	T_j = 25 (125) °C		1 (0,9)	1,2 (1,1)	V
r_T	T_j = 25 (125) °C		14 (22)	18 (26)	mΩ
C_{ies}	V_{CE} = 25 V, V_{GE} = 0 V, f = 1 MHz		3,7		nF
C_{oes}	V_{CE} = 25 V, V_{GE} = 0 V, f = 1 MHz		0,8		nF
C_{res}	V_{CE} = 25 V, V_{GE} = 0 V, f = 1 MHz		0,7		nF
$R_{th(j-s)}$	per IGBT		0,55		K/W
$t_{d(on)}$	under following conditions		85		ns
t_r	V_{CC} = 600 V, V_{GE} = ± 15 V		30		ns
$t_{d(off)}$	I_{Cnom} = 50 A, T_j = 125°C		430		ns
t_f	R_{Gon} = R_{Goff} = 12 Ω		90		ns
E_{on}	inductive load		6,5		mJ
E_{off}			6,1		mJ
Diode - Inverter, Chopper					
V_F = V_{EC}	I_{Fnom} = 50 A, T_j = 25 (125) °C		1,6 (1,6)	1,8 (1,8)	V
$V_{(TO)}$	T_j = 25 (125) °C		1 (0,8)	1,1 (0,9)	V
r_T	T_j = 25 (125) °C		12 (16)	14 (18)	mΩ
$R_{th(j-s)}$	per diode		1		K/W
I_{RRM}	under following conditions		71		A
Q_{rr}	I_{Fnom} = 50 A, V_R = 600 V		11,5		µC
E_{rr}	V_{GE} = 0 V, T_j = 125 °C		4,7		mJ
	di_F/dt = 1900 A/µs				
Diode - Rectifier					
V_F	I_{Fnom} = 40 A, T_j = 25 °C		1,1		V
$V_{(TO)}$	T_j = 125 °C		0,8		V
r_T	T_j = 125 °C		9		mΩ
$R_{th(j-s)}$	per diode		0,85		K/W
Temperature Sensor					
R_{ts}	3 %, T_r = 25 (100) °C		1000(1670)		Ω
Mechanical Data					
w			95		g
M_s	Mounting torque	2		2,5	Nm

D.4 YASKAWA emr1-43p7a-eu13 permanent magnet motor

Table D.1: YASKAWA emr1-43p7a-eu13 nameplate

Name	Data
Manufacturer	YASKAWA
Product	PMSM
Type	emr1-43p7a-eu13
Pole number	10
Rated power	3,7kW
Rated voltage	289V
Rated current	8,4A
Efficiency	0,91
Power factor	0,97
Rated speed	1750
Rated frequency	146

Table D.2: YASKAWA emr1-43p7a-eu13 datasheet

Description	Unit	Value
Rated power	kW	3,7
Rated frequency	Hz	145,8
Lowest frequency	Hz	6,0
Pole number	-	10
Rated voltage	V	360
Nominal current	A	8,3
Winding resistance	Ω	0,841
D-axis inductance	mH	10,40
Q-axis inductance	mH	10,40
Voltage constant	$\frac{mV \cdot s}{rpm}$	165,41

Appendix E

Source code listings

Listing E.1: DRP interface controller

```
function [den_o, daddr_o, val_o] = xadc_ctrl(drdy_i, do_i)
%#codegen

%% Constants and defines
% Fixed point data types
FM = hdlfimath;
NT_ADDR = numerictype(false,7,0); % Address
NT_VAL = numerictype(false,16,0); % Value
NT_STATE = numerictype(false,2,0); % State

% The state machine will loop over the defined addresses
% reading out each value
% See Xilinx ug480 for more information
ADDR_SEQ = fi([3 16 24], ... % Value addresses
NT_ADDR,FM);

NUM_VALS = numel(ADDR_SEQ); % Number of values to read

% Define the index data type based on the number of values
NT_ADDR_IDX = ...
    numerictype(false,ceil(log2(NUM_VALS))+1,0); % index

% State machine defines
STATE_IDLE = fi(0,NT_STATE,FM);
STATE_ADDR = fi(1,NT_STATE,FM);
STATE_EN = fi(2,NT_STATE,FM);
STATE_WAIT = fi(3,NT_STATE,FM);

% Declare registers
```

```
29      persistent state val addr den addr_idx;
30
31      % Set reset values
32      if isempty(state)
33      state = STATE_IDLE;
34      val = fi(zeros(NUM_VALS,1),NT_VAL,FM);
35      addr = fi(0,NT_ADDR,FM);
36      den = false;
37      addr_idx = fi(0,NT_ADDR_IDX,FM);
38      end
39
40      %% Drive the outputs
41      daddr_o = addr;
42      den_o = den;
43      val_o = val;
44
45      %% Control the state
46      switch (state)
47      case STATE_IDLE
48      % IDLE state, increment the index
49      if (addr_idx == NUM_VALS)
50      addr_idx(:) = 1;
51      else
52      addr_idx(:) = addr_idx + 1;
53      end
54      state(:) = STATE_ADDR;
55      case STATE_ADDR
56      % ADDR state, write out the address
57      addr(:) = ADDR_SEQ(addr_idx);
58      den(:) = false;
59      state(:) = STATE_EN;
60      case STATE_EN
61      % EN state, drive the enable high
62      state(:) = STATE_WAIT;
63      den(:) = true;
64      case STATE_WAIT
65      % WAIT state, wait for drdy
66      if (drdy_i)
67      val(addr_idx) = do_i;
68      state(:) = STATE_IDLE;
69      end
70      den(:) = false;
71      end
```

Listing E.2: Initialization of model predictive current controller for PMSM

```
1  \begin{lstlisting}
2  % Initialization file for model predictive control
```

```
% of a permanent magnet synchronus machine
%
% The following implementation utilizes alpha beta
% frame in order to simplify the implementation on FPGA.
%
% (c) Eirik Haustveit, 2016.

% Variables required by the control algorithm
global Ts Rs Ls p sw_states v_vect_alpha v_vect_beta
% Sampling time of the predictive algorithm [s]
Ts = 4e-5;

% Machine parameters
J = 0.0027; % Moment of inertia [kg m^2]
p = 1; % Pole pairs
Ls = 0.000395; % Stator inductance [H]
Rs = 0.0485; % Stator resistance [Ohm]
psi_m = 0.1194; % Permanent magnet flux [Wb]
%T_nom = 20; % Nominal torque [Nm]
i_max = 50; % Rated current [A]

% DC-link voltage [V]
% This should normally be measured.
Vdc = 540;

% Define the possible voltage output vectors
% from the 2 level, three phase inverter.
v0_alpha = 0;
v1_alpha = 2/3*Vdc;
v2_alpha = 1/3*Vdc;
v3_alpha = -1/3*Vdc;
v4_alpha = -2/3*Vdc;
v5_alpha = -1/3*Vdc;
v6_alpha = 1/3*Vdc;
v7_alpha = 0;

v0_beta = 0;
v1_beta = 0;
v2_beta = sqrt(3)/3*Vdc;
v3_beta = sqrt(3)/3*Vdc;
v4_beta = 0;
v5_beta = sqrt(3)/3*Vdc;
v6_beta = sqrt(3)/3*Vdc;
v7_beta = 0;

v_vect_alpha = [v0_alpha, v1_alpha, v2_alpha, v3_alpha, ...
    v4_alpha, v5_alpha, v6_alpha, v7_alpha];
v_vect_beta = [v0_beta, v1_beta, v2_beta, v3_beta, ...
    v4_beta, v5_beta, v6_beta, v7_beta];
```

```
50
51 % Define the possible switching states for the inverter.
52 % Each swiching state corresponds to a voltage vector
53 % as defined previously.
54 sw_states = [0 0 0;1 0 0;1 1 0;0 1 0;0 1 1;0 0 1;1 0 1;1 1 1];
```

Listing E.3: Model predictive current controller for PMSM

```
1  % Model predictive control of permanent magnet synchronus ...
       machine.
2
3  function [ga, gb, gc, id_pred_opt, iq_pred_opt, v_out_d, ...
       v_out_q] = fcn(id, iq, id_ref, iq_ref, omega_r, theta_r)
4
5  % Variables defined in the parameters file
6  global Rs Ls Ts psi_m sw_states v_vect_alpha v_vect_beta i_max
7  % Persistent variables
8  persistent x_opt
9
10 % Make sure x_opt always has a value
11 if isempty(x_opt), x_opt = 1; end
12
13 % Assign a large initial value to the optimum
14 % cost, in order for the initial value to not be optimal.
15 g_opt = inf;
16
17 % Compute the sine and cosine of the rotor angle.
18 p_sine = sin(theta_r);
19 p_cosine = cos(theta_r);
20
21 id_pred_opt = 0;
22 iq_pred_opt = 0;
23
24 % Iterate over the possible output voltage vectors
25 for i = 1:8
26 % Rotate the voltage vectors to dq-frame using the current
27 % rotor angle theta_r
28 v_out_q = (p_cosine * v_vect_alpha(i)) +  (p_sine * ...
       v_vect_beta(i));
29 v_out_d = (p_cosine * v_vect_beta(i)) -  (p_sine * ...
       v_vect_alpha(i));
30
31 % Predict the d-axis current at the next iteration
32 id_pred = (1 - (Rs*Ts/Ls))*id + (Ts*omega_r*iq) + ...
       (Ts/Ls)*v_out_d;
33 % Predict the q-axis current at the next iteration
34 iq_pred = (1 - (Rs*Ts/Ls))*iq + (Ts*omega_r*id) + ...
       (Ts/Ls)*v_out_q - (psi_m*omega_r*Ts);
```

```
35
36 if(id_pred > i_max || iq_pred > i_max)
37 g = inf;
38 else
39 g = abs(id_ref - id_pred) + abs(iq_ref - iq_pred);
40 %g = (id_ref - id_pred)^2 + (iq_ref - iq_pred)^2;
41 end
42
43 % If the cost of the vector at the current iteration is
44 % less than the cost at the previous iteration, a more
45 % optimum vector has been found.
46 if (g < g_opt)
47 g_opt = g;
48 id_pred_opt = id_pred;
49 iq_pred_opt = iq_pred;
50 x_opt = i;
51 end
52 end
53
54 % Output switching states
55 ga = sw_states(x_opt,1);
56 gb = sw_states(x_opt,2);
57 gc = sw_states(x_opt,3);
```

Listing E.4: Determine compare values for SV-PWM

```
1  function [Ta, Tb, Tc] = fcn(T1, T2, T0, Sector)
2  %#codegen
3  switch Sector
4  case 1
5  Ta = T1 + T2 + T0/2;
6  Tb = T2 + T0/2;
7  Tc = T0/2;
8  case 2
9  Ta = T1 + T0/2;
10 Tb = T1 + T2 + T0/2;
11 Tc = T0/2;
12 case 3
13 Ta = T0/2;
14 Tb = T1 + T2 + T0/2;
15 Tc = T2 + T0/2;
16 case 4
17 Ta = T0/2;
18 Tb = T1 + T0/2;
19 Tc = T1 + T2 + T0/2;
20 case 5
21 Ta = T2 + T0/2;
22 Tb = T0/2;
```

```
23 Tc = T1 + T2 + T0/2;
24 case 6
25 Ta = T1 + T2 + T0/2;
26 Tb = T0/2;
27 Tc = T1 + T0/2;
28 otherwise
29 Ta = 0;
30 Tb = 0;
31 Tc = 0;
32 end
```

Appendix F

Tools and equipment

Table F.1: List of software tools

Manufacturer	Name	Version no.	Description
MathWorks	Matlab	R2015b	Simulation and development
Xilinx	Vivado	2014.4 Webpack	FPGA development
National instruments	Multisim	14.0	Circuit simulation
Cadsoft	Eagle	7.5.0	Circuit board layout
Microchip	Filterlab	2.0	Active filter design

Table F.2: List of laboratory equipment

Equipment	Model	Serial no.	NTNU serial no.
Oscilloscope	Tektronix TDS2014	C013005	G04-0258
Oscilloscope	Tektronix TDS2014B	C103265	G04-0350
Differential probe	Tektronix P5200A	C020619	I06-0514
Differential probe	Tektronix P5200A	C020622	I06-0519
Current probe	Fluke 80i-110S		I04-0521
Function generator	TTi TG320	344011	C02-0622
Multimeter	Fluke 112		S03-0354
Power supply	Instek GPC-3030DQ		B02-0458

Appendix G

Operating system

G.1 Loading the bitstream to PL from PS

During development the bistream is typically transfered using JTAG. However as the FPGA configuration is lost during reboot, this is not practical for a production design.

If the Zynq SoC is running Linux, it is possible to have the operating system load the bitstream automatically.

The Linux kernel typically initializes a device file at */dev/xdevcfg* that may be used to program the PL. The bitstream file may be copied to the SD-card and it is the task of the operating system to transfer this file during the final stage of the boot process.

The following section covers how to load the bitstream manually by executing commands on the Linux terminal.

If the device file does not exist, it may be created with the *mknod* command.

```
1 mknod /dev/xdevcfg c 259 0 > /dev/null
```

This assumes that a appropriate driver is loaded at the correct device number. c tells mknod to create a char device, 259 and 0 are the major and minor device numbers respectively.

Preparing the bitstream

Before the bitsream is transfered it must be bitswapped. This may be achieved by the *bootgen* utillity supplied by Xilinx.

The *bootgen* utility utilizes a so called BIF file to define how it operates. The following example may be used if the goal is to simply generate a bitswapped bitstream, where *system_top_wrapper.bit* is the generated bitsream output from Vivado.

```
1 all:
2 {
3 system_top_wrapper.bit
4 }
```

If the file is named *all.bif*, the following command converts the bitstream.

```
1 bootgen -image all.bif -w -process_bitstream bin
```

The generated *system_top_wrapper.bit.bin* should be copied to an appropriate location on the SD-card.

Transferring

Once the operating system is booted, the bitstream may be transferred using the following Linux command. This should probably be executed automatically in a production design.

```
1 cat system_top_wrapper.bit.bin > /dev/xdevcfg
```

The following command should return 1 to indicate that programming was successful.

```
1 cat /sys/devices/amba.0/f8007000.devcfg/prog_done
```

Bibliography

[1] 754. "IEEE Standard for Floating-Point Arithmetic". In: *IEEE Std 754-2008* (Aug. 2008), pp. 1–70. DOI: 10.1109/IEEESTD.2008.4610935.

[2] Judit Baños García and Sonny Quillo. *Predictive Controller for PMSM Drive*. D EPARTMENT OF E NERGY T ECHNOLOGY A ALBORG U NIVERSITY, 2013.

[3] Felix Blaschke. "The method of field orientation for control of the rotating field machine." Germany: Techn. Univ. Braunschweig, 1984.

[4] Felix Blaschke. "The principle of field-orientation as applied to the transvector closed-loop control system for rotating-field machines". In: *Siemens* (1972).

[5] S. Bolognani, R. Oboe, and M. Zigliotto. "Sensorless full-digital PMSM drive with EKF estimation of speed and rotor position". In: *IEEE Transactions on Industrial Electronics* 46.1 (Feb. 1999), pp. 184–191. ISSN: 0278-0046. DOI: 10.1109/41.744410.

[6] G. S. Buja and M. P. Kazmierkowski. "Direct torque control of PWM inverter-fed AC motors - a survey". In: *IEEE Transactions on Industrial Electronics* 51.4 (Aug. 2004), pp. 744–757. ISSN: 0278-0046. DOI: 10.1109/TIE.2004.831717.

[7] Bruce Carter and Thomas R. Brown. *HANDBOOK OF OPERATIONAL AMPLIFIER APPLICATIONS*. 2001.

[8] Khalid Chikh, Mohamed Khafallah, and Abdallah Sa. "Improved DTC Algorithms for Reducing Torque and Flux Ripples of PMSM Based on Fuzzy Logic and PWM Techniques". In: *MATLAB - A Fundamental Tool for Scientific Computing and Engineering Applications - Volume 1*. Ed. by Vasilios Katsikis. InTech, Sept. 26, 2012. ISBN: 978-953-51-0750-7. URL: http://www.intechopen.com/books/matlab-a-fundamental-tool-for-scientific-computing-and-engineering-applications-volume-1/improved-dtc-algorithms-for-reducing-torque-and-flux-ripples-of-pmsm-based-on-fuzzy-logic-and-pwm-te (visited on 05/16/2016).

[9] M. Depenbrock. “Direct self-control (DSC) of inverter-fed induction machine”. In: *IEEE Transactions on Power Electronics* 3.4 (Oct. 1988), pp. 420–429. ISSN: 0885-8993. DOI: 10.1109/63.17963.

[10] Analog devices. *AD7403 16-Bit, Isolated Sigma-Delta Modulator datasheet*. 2015.

[11] Analog devices. *ADuM 5000 - Isolated DC-to-DC Converter*. 2012.

[12] Analog devices. *ADuM1400 Quad-Channel Digital Isolators datasheet*. 2015.

[13] Analog devices. *ADuM7640 Six-Channel Digital Isolators datasheet*. 2012.

[14] W. C. Duesterhoeft, M. W. Schulz, and E. Clarke. “Determination of Instantaneous Currents and Voltages by Means of Alpha, Beta, and Zero Components”. In: *Transactions of the American Institute of Electrical Engineers* 70.2 (July 1951), pp. 1248–1255. ISSN: 0096-3860. DOI: 10.1109/T-AIEE.1951.5060554.

[15] M. Elbuluk and Changsheng Li. “Sliding mode observer for wide-speed sensorless control of PMSM drives”. In: *Industry Applications Conference, 2003. 38th IAS Annual Meeting. Conference Record of the*. Industry Applications Conference, 2003. 38th IAS Annual Meeting. Conference Record of the. Vol. 1. Oct. 2003, 480–485 vol.1. DOI: 10.1109/IAS.2003.1257543.

[16] M. Sami Fadali and Antonio Visioli. *Digital Control Engineering Analysis and Design*. Second Edition. Elsevier, 2013. ISBN: 978-0-12-394391-0.

[17] C. French and P. Acarnley. “Direct torque control of permanent magnet drives”. In: *IEEE Transactions on Industry Applications* 32.5 (Sept. 1996), pp. 1080–1088. ISSN: 0093-9994. DOI: 10.1109/28.536869.

[18] Xavier del Toro Garcia et al. “Comparison between FOC and DTC Strategies for Permanent Magnet Synchronous Motors”. In: *Advances in Electrical and Electronic Engineering* 5.1 (June 21, 2011), pp. 76–81. ISSN: 1804-3119. URL: http://advances.uniza.sk/index.php/AEEE/article/view/179 (visited on 05/10/2016).

[19] Duan Graovac , and Marco Pürschel. “IGBT Power Losses Calculation Using the Data-Sheet Parameters”. In: (Jan. 29, 2009).

[20] M. E. Haque, Limin Zhong, and M. F. Rahman. “A sensorless initial rotor position estimation scheme for a direct torque controlled interior permanent magnet synchronous motor drive”. In: *IEEE Transactions on Power Electronics* 18.6 (Nov. 2003), pp. 1376–1383. ISSN: 0885-8993. DOI: 10.1109/TPEL.2003.818869.

[21] Eirik Haustveit. *Laboratory platform for PM machine drive study - Adaptive Torque Estimation*. scientific. Trondheim: NTNU, Dec. 16, 2015, p. 103.

[22] Markus Hermwille. *Gate Resistor Principles and Applications*. Semikron, Nov. 12, 2007.

[23] Niklas Hofstötter. *Limits and hints how to turn off IGBTs with unipolar supply*. June 3, 2015.

[24] J. Holtz. “Pulsewidth modulation-a survey”. In: *IEEE Transactions on Industrial Electronics* 39.5 (Oct. 1992), pp. 410–420. ISSN: 0278-0046. DOI: 10.1109/41.161472.

[25] Jun Hu and Bin Wu. “New integration algorithms for estimating motor flux over a wide speed range”. In: *IEEE Transactions on Power Electronics* 13.5 (Sept. 1998), pp. 969–977. ISSN: 0885-8993. DOI: 10.1109/63.712323.

[26] Zulkifilie Bin Ibrahim et al. “Simulation Investigation of SPWM, THIPWM and SVPWM Techniques for Three Phase Voltage Source Inverter”. In: *International Journal of Power Electronics and Drive Systems (IJPEDS)* 4.2 (June 1, 2014), pp. 223–240. ISSN: 2088-8694. URL: http://iaesjournal.com/online/index.php/IJPEDS/article/view/5833 (visited on 05/24/2016).

[27] Texas Instruments. *AM26C32 Quadruple Differential Line Receiver*. 2015.

[28] Texas Instruments. *CD40109 - CMOS Quad Low-to-High Voltage level shifter datasheet*. 2003.

[29] Texas Instruments. *Field Orientated Control of 3-Phase AC-Motors*. Application Note BPRA073. Texas Instruments Europe, Feb. 1998.

[30] Texas Instruments. *LF353-N Wide Bandwidth Dual JFET Input Operational Amplifier*. 2013.

[31] Texas Instruments. *LM311 Quad Differential Comparators datasheet*. 2015.

[32] T. M. Jahns, G. B. Kliman, and T. W. Neumann. “Interior Permanent-Magnet Synchronous Motors for Adjustable-Speed Drives”. In: *IEEE Transactions on Industry Applications* IA-22.4 (July 1986), pp. 738–747. ISSN: 0093-9994. DOI: 10.1109/TIA.1986.4504786.

[33] T. M. Jahns and W. L. Soong. “Pulsating torque minimization techniques for permanent magnet AC motor drives-a review”. In: *IEEE Transactions on Industrial Electronics* 43.2 (Apr. 1996), pp. 321–330. ISSN: 0278-0046. DOI: 10.1109/41.491356.

[34] J. G. Kassakian and T. M. Jahns. “Evolving and Emerging Applications of Power Electronics in Systems”. In: *IEEE Journal of Emerging and Selected Topics in Power Electronics* 1.2 (June 2013), pp. 47–58. ISSN: 2168-6777. DOI: 10.1109/JESTPE.2013.2271111.

[35] Hongryel Kim, Jubum Son, and Jangmyung Lee. “A High-Speed Sliding-Mode Observer for the Sensorless Speed Control of a PMSM”. In: *IEEE Transactions on Industrial Electronics* 58.9 (Sept. 2011), pp. 4069–4077. ISSN: 0278-0046. DOI: 10.1109/TIE.2010.2098357.

[36] C. Lascu, I. Boldea, and F. Blaabjerg. “A modified direct torque control (DTC) for induction motor sensorless drive”. In: *The 1998 IEEE Industry Applications Conference, 1998. Thirty-Third IAS Annual Meeting*. The 1998 IEEE Industry Applications Conference, 1998. Thirty-Third IAS Annual

Meeting. Vol. 1. Oct. 1998, 415–422 vol.1. DOI: 10.1109/IAS.1998.732336.

[37] LEM. *Voltage Transducer LV 25-1000 - datasheet*. 2006.

[38] Changsheng Li and M. Elbuluk. "A sliding mode observer for sensorless control of permanent magnet synchronous motors". In: *Conference Record of the 2001 IEEE Industry Applications Conference, 2001. Thirty-Sixth IAS Annual Meeting*. Conference Record of the 2001 IEEE Industry Applications Conference, 2001. Thirty-Sixth IAS Annual Meeting. Vol. 2. Sept. 2001, 1273–1278 vol.2. DOI: 10.1109/IAS.2001.955672.

[39] Julius Luukko. "Direct torque control of permanent magnet synchronous machines - Analysis and implementation". Doctor of Science. Lappeenranta University of Technology, 2000.

[40] Clive Maxfield. *The Design Warrior's Guide to FPGAs*. 2004. 560 pp.

[41] Ned Mohan. *Advanced electric drives*. 2014. 168 pp. ISBN: 978-1-118-48548-4.

[42] Roy Nilsen. *TET4120 - ELECTRIC DRIVES*. NORWEGIAN UNIVERSITY OF SCIENCE AND TECHNOLOGY, NTNU Department of Electric Power Engineering, Spring 2016.

[43] Ogasawara, Akagi, and Nabae , Nagaoka. "A novel PWM scheme of voltage source inverters based on space vector theory". In: *EPE Euro- pean Conf. Power Electron* (1990).

[44] R.H. Park. "Two-reaction theory of synchronous machines generalized method of analysis-part I". In: *American Institute of Electrical Engineers, Transactions of the* 48.3 (July 1929), pp. 716–727. ISSN: 0096-3860. DOI: 10.1109/T-AIEE.1929.5055275.

[45] D. J. Patterson et al. "The "goodness" of small contemporary permanent magnet electric machines". In: *Electric Machines and Drives Conference, 2003. IEMDC'03. IEEE International*. Electric Machines and Drives Conference, 2003. IEMDC'03. IEEE International. Vol. 2. June 2003, 1195–1200 vol.2. DOI: 10.1109/IEMDC.2003.1210392.

[46] P. D. C. Perera et al. "A sensorless, stable V/f control method for permanent-magnet synchronous motor drives". In: *IEEE Transactions on Industry Applications* 39.3 (May 2003), pp. 783–791. ISSN: 0093-9994. DOI: 10.1109/TIA.2003.810624.

[47] M. Preindl and S. Bolognani. "Model Predictive Direct Torque Control With Finite Control Set for PMSM Drive Systems, Part 1: Maximum Torque Per Ampere Operation". In: *IEEE Transactions on Industrial Informatics* 9.4 (Nov. 2013), pp. 1912–1921. ISSN: 1551-3203. DOI: 10.1109/TII.2012.2227265.

[48] M. Preindl and S. Bolognani. "Model Predictive Direct Torque Control With Finite Control Set for PMSM Drive Systems, Part 2: Field Weakening Op-

eration". In: *IEEE Transactions on Industrial Informatics* 9.2 (May 2013), pp. 648–657. ISSN: 1551-3203. DOI: 10.1109/TII.2012.2220353.

[49] Christophe Preve. "Calculation of Inrush Current During Capacitor Bank Energization". In: (2006).

[50] Jose Rodriguez and Patricio Cortes. *PREDICTIVE CONTROL OF POWER CONVERTERS AND ELECTRICAL DRIVES*. Universidad Tecnica Federico Santa Maria, Valparaiso, Chile: IEEE, 2012.

[51] Seimikron. *Semidriver SKHI 71 datasheet*. May 25, 2007.

[52] Seimikron. *SKiiP 35NAB126V10 datasheet*. Jan. 27, 2010.

[53] C.E. Shannon. "Communication in the Presence of Noise". In: *Proceedings of the IRE* 37.1 (Jan. 1949), pp. 10–21. ISSN: 0096-8390. DOI: 10.1109/JRPROC.1949.232969.

[54] A. Shinohara et al. "Comparison of stator flux linkage estimators for PWM-based direct torque controlled PMSM drives". In: *2015 IEEE 11th International Conference on Power Electronics and Drive Systems (PEDS)*. 2015 IEEE 11th International Conference on Power Electronics and Drive Systems (PEDS). June 2015, pp. 1035–1040. DOI: 10.1109/PEDS.2015.7203461.

[55] T. Sutikno et al. "An Improved FPGA Implementation of Direct Torque Control for Induction Machines". In: *IEEE Transactions on Industrial Informatics* 9.3 (Aug. 2013), pp. 1280–1290. ISSN: 1551-3203. DOI: 10.1109/TII.2012.2222420.

[56] I. Takahashi and T. Noguchi. "A New Quick-Response and High-Efficiency Control Strategy of an Induction Motor". In: *IEEE Transactions on Industry Applications* IA-22.5 (Sept. 1986), pp. 820–827. ISSN: 0093-9994. DOI: 10.1109/TIA.1986.4504799.

[57] Lixin Tang et al. "A novel direct torque control for interior permanent-magnet synchronous machine drive with low ripple in torque and flux-a speed-sensorless approach". In: *IEEE Transactions on Industry Applications* 39.6 (Nov. 2003), pp. 1748–1756. ISSN: 0093-9994. DOI: 10.1109/TIA.2003.818981.

[58] Lixin Tang et al. "A novel direct torque controlled interior permanent magnet synchronous machine drive with low ripple in flux and torque and fixed switching frequency". In: *IEEE Transactions on Power Electronics* 19.2 (Mar. 2004), pp. 346–354. ISSN: 0885-8993. DOI: 10.1109/TPEL.2003.823170.

[59] P. Vaclavek, P. Blaha, and I. Herman. "AC Drive Observability Analysis". In: *IEEE Transactions on Industrial Electronics* 60.8 (Aug. 2013), pp. 3047–3059. ISSN: 0278-0046. DOI: 10.1109/TIE.2012.2203775.

[60] Vacon. *Brake resistors user manual*. Apr. 24, 2015.

[61] T.J. Vyncke, R.K. Boel, and J.A.A. Melkebeek. “A comparison of stator flux linkage estimators for a direct torque controlled PMSM drive”. In: *35th Annual Conference of IEEE Industrial Electronics, 2009. IECON ’09*. 35th Annual Conference of IEEE Industrial Electronics, 2009. IECON ’09. Nov. 2009, pp. 971–978. DOI: 10.1109/IECON.2009.5414686.

[62] Zhang Xi. *How to calculate and minimize the dead time requirement for IGBTs properly*. Infineon Technologies, 2007.

[63] Ying-Shieh Kung et al. “FPGA Realization of Sensorless PMSM Speed Controller Based on Extended Kalman Filter”. In: (Oct. 22, 2013).

[64] Jorge Zambada and Debraj Deb. “Sensorless Field Oriented Control of a PMSM”. In: *Microchip Technology Inc.* (2010).

[65] L. Zhong et al. “A direct torque controller for permanent magnet synchronous motor drives”. In: *IEEE Transactions on Energy Conversion* 14.3 (Sept. 1999), pp. 637–642. ISSN: 0885-8969. DOI: 10.1109/60.790928.

[66] L. Zhong et al. “Analysis of direct torque control in permanent magnet synchronous motor drives”. In: *IEEE Transactions on Power Electronics* 12.3 (May 1997), pp. 528–536. ISSN: 0885-8993. DOI: 10.1109/63.575680.

[67] Musamettin Zurnaci. *MiniSKiiP ő Generation II Technical Explanations*. 2014.

Index

Glossary

covariance The covariance is a statistical parameter describing the strength of the correlation between two variables.. 39

perturbation A perturbation is an approximate solution to a mathematical problem, based upon the solution of a related, but simpler problem.. 39

Acronyms

ADC Analog to digital converter. 28, 116, 117

ASIC Application-specific integrated circuit. 28

AXI4 Advanced eXtensible Interface 4. 46, 55

BLDC Brushless DC. 3

CPU Central processing unit. 45, 53

DRP Dynamic reconfiguration port. 55

DTC Direct torque control. xv, xvii, 7, 17–19, 27, 28, 62, 87, 88

EKF Extended Kalman filter. 39

EMF electromotive force. 4, 6, 9, 28, 29, 31–33, 35–37, 97, 100, 104

FDTC Fuzzy Direct Torque Control. 17

FMC FPGA Mezzanine Card. 113

FOC Field-oriented control. xv, 7, 8, 22, 72, 79, 83, 97

FPGA Field-programmable gate array. 45, 47, 49, 51, 52, 59, 60, 64, 73, 145, XXXV

FPU Floating-point unit. 53

HDL Hardware description language. xvi, 46, 47, 51, 73, 83

IGBT Insulated-gate bipolar transistor. xviii, 112, 121–123, 126, 127, 131, 138, 145

IP-core Intellectual property core. 46, 57, 62

IPMSM Interior permanent magnet synchronus machine. xvi, 3–5, 9, 17–19, 80, 81, 111

JTAG Joint Test Action Group. XXXV

LPF Low-pass filter. 29

LUT Lookup table. 19

MPC Model predictive control. xv, xvii, 7, 22–24, 62, 93–95

NTNU Norges teknisk-naturvitenskapelige universitet. XI

PI Proportional integral. 8, 9, 18

PL Programmable logic. 53, 55

PMSM Permanent magnet synchronus machine. xv, 3, 4, 7, 8, 11, 17, 22, 23, 27–30, 33, 35, 40, 71, 72, 75, 97, 145

PS Processing system. 55, 68

p.u. Per unit. 53, 54

PWM Pulse width modulation. 28, 61, 62, 141

RTL Register-transfer level. 57

SiC Silicon Carbide. 112

SMO Sliding mode observer. 31, 35, 36, 100, 104, 146

SoC System on Chip. 45, 112

SPMSM Surface permanent magnet synchronus machine. 3, 4, 9, 18, 111, 141

SPWM Sinusoidal pulse width modulation. 12, 58

SV-PWM Space vector pulse width modulation. xv, xvi, 9, 13, 15, 16, 22, 58, 83

THIPWM Third Harmonic Injection Pulse Width Modulation. xv, 13, 15, 16

VHDL VHSIC Hardware Description Language. 47

VHSIC Very High Speed Integrated Circuit. 47

VSI voltage source inverter. 121

XADC Xilinx Analog Digital Converter. 55, 141

List of Symbols

e_s^p predicted stator back-EMF. 35

i_{sd}^p d-axis predicted current. 23

i_{sd}^* d-axis reference current. 23, 24

r_c IGBT on state resistance. 123

V_{ce0} IGBT on state zero current voltage drop. 123

i_s^p stator predicted current. 35

i_{sq}^p q-axis predicted current. 23

i_{sq}^* q-axis reference current. 23, 24

J mechanical inertia. 133

L_m stator magnetizing inductance. 35

$\mu(t)$ Measurement noise. 39

$R(t)$ Covariance of the measurement noise. 39

ω_m rotor mechanical angular velocity. 133

ω_m^p predicted rotor angular velocity. 25

ω_m^* rotor angular velocity reference. 25

ω_0 filter cutoff frequency. 36

R_s stator winding resistance. 35

$u(t)$ System input matrix. 39

$\sigma(t)$ System noise. 39

$Q(t)$ Covariance of the system noise. 39

$y(t)$ System output matrix. 39

$x(t)$ System state matrix. 39

v_s^* stator voltage reference. 35