Predictive Control of an Axial Flux Permanent Magnet Machine

Lynn Verkroost

Supervisors: Prof. dr. ir. Peter Sergeant, Dr. ir. Hendrik Vansompel
Counsellor: Joachim Druant

Master's dissertation submitted in order to obtain the academic degree of
Master of Science in Electromechanical Engineering

Department of Electrical Energy, Metals, Mechanical Constructions & Systems
Chair: Prof. dr. ir. Luc Dupré
Faculty of Engineering and Architecture
Academic year 2016-2017

GHENT
UNIVERSITY

Predictive Control of an Axial Flux Permanent Magnet Machine

Lynn Verkroost

Supervisors: Prof. dr. ir. Peter Sergeant, Dr. ir. Hendrik Vansompel
Counsellor: Joachim Druant

Master's dissertation submitted in order to obtain the academic degree of
Master of Science in Electromechanical Engineering

Department of Electrical Energy, Metals, Mechanical Constructions & Systems
Chair: Prof. dr. ir. Luc Dupré
Faculty of Engineering and Architecture
Academic year 2016-2017

GHENT
UNIVERSITY

Permission of Usage

Lynn Verkroost, June 2, 2017

Preface

Almost exactly one year ago, I had to choose the topic of my master's dissertation. Since I have a relatively wide field of interest, this choice was not immediately obvious for me. Eventually my passion for control engineering, combined with my interest for electric machines, the combination of both theoretical and somewhat more practical work, and the enthusiasm of my supervisor Hendrik Vansompel, convinced me to opt for the topic of predictive control of an axial flux permanent magnet machine. One year later, I do not regret this decision. It has been a busy, but at the same time very instructive year. After spending the first semester examining the different types of predictive controllers, studying their working principles and understanding their behavior by means of simulations, the second semester was dedicated to the validation of the control algorithms on the real machine. Soon it became clear that simulating a control algorithm is one thing, but making it work on a test set-up is another matter. Getting the different IO-signals to be read and sent properly, and learning how to program an FPGA were less evident than initially thought. Hence seeing the controllers work properly on the test set-up was extra satisfactory.

However, all this would not have been possible without the help of others. First of all, I would like to express my gratitude to my counselor and supervisors, ir. Joachim Druant, dr. ir. Hendrik Vansompel, and prof. dr. ir. Peter Sergeant. Not only did they provide me with useful feedback concerning my research and how to communicate this research clearly in this text, they also spent quite some hours in the lab, trying to get the machine running like it should. I would like to extend my appreciation to Florian Verbelen as well, for his help with the programming of the FPGA.

In addition, I would like to thank the technicians and other students in the lab for the very pleasant working atmosphere. My breaks would have been far less fun without them.

Furthermore, I wish to thank my family and friends, and in particular my mum, dad en sister Sarah. For inspiring me to take up this study, for keeping believing in me, and for supporting me when I needed it the most.

Last but not least, I want to thank my boyfriend Nick, for putting up with my late work, and for sharing the past five wonderful years together.

Lynn Verkroost, June 2, 2017

Predictive Control of an Axial Flux Permanent Magnet Machine

by

Lynn Verkroost

Master's dissertation submitted in order to obtain the academic degree of Master of Science in Electromechanical Engineering

Academic year 2016-2017

Supervisors: Prof. dr. ir. Peter Sergeant, Dr. ir. Hendrik Vansompel
Counsellor: Ir. Joachim Druant
Faculty of Engineering and Architecture
Ghent University

Department of Electrical Energy, Metals, Mechanical Constructions & Systems
Chair: Prof. dr. ir. Luc Dupré

Abstract
Electrical machines are mostly fed using a voltage source inverter. In order to achieve a given torque demand, a controller is needed to select an optimal sequence of switch states for this inverter. A large variety of control algorithms exists for this purpose, among them the relatively new family of predictive controllers. As the name states, these types of controllers have in common that they use a model of the machine and the inverter to predict the future system behavior. The aim of this master's dissertation is to control the torque on the rotor of an axial flux permanent magnet synchronous machine (AFPMSM) using predictive control. In Chapter 2 a short introduction is given on this novel machine topology. The operation principles of the two main categories of predictive controllers that will be discussed in this text, more specifically Finite-Set Model Based Predictive Control (FS-MBPC) and deadbeat (DB) control, are elucidated in Chapter 3. Since the quality of the system model is of extreme importance to enable predictive control, Chapter 4 is entirely devoted to this topic. In Chapter 5, the system model is used to directly regulate the torque and flux of the AFPMSM by means of FS-MBPC. Since these two control parameters are not independent, Chapter 6 examines how the torque can be controlled by means of two orthogonal stator current components, the so-called Field Oriented Control (FOC). This strategy is implemented not only by means of both FS-MBPC and DB control, but also by means of a hybrid form of these two types of controllers. Simulation results for the various control strategies are subjected to an extensive comparison in Chapter 7. Finally, the controllers are implemented on a Field Programmable Gate Array (FPGA) and applied on a real test set-up. The concluding results are presented in Chapter 8.

Keywords
Finite-Set Model Based Predictive Control, Deadbeat Control, Axial Flux Permanent Magnet Synchronous Machine, FPGA

Predictive Control of an Axial Flux Permanent Magnet Machine

Lynn Verkroost

Supervisors: ir. Joachim Druant, dr. ir. Hendrik Vansompel, and prof. dr. ir. Peter Sergeant

***Abstract*—This article presents a comparative study of three Predictive Current Control (PCC) schemes for an Axial Flux Permanent Magnet Synchronous Machine (AFPMSM). The first control scheme predicts the future behavior for each of the eight possible switch states of the two-level voltage source inverter (2L-VSI), and selects the most optimal one based on the evaluation of a cost function. This switch state is then applied during the entire next update period. The second control scheme uses the system model to calculate which input voltage is required to bring the AFPMSM to the requested state. This reference value for the voltage is synthesized afterwards by means of a separate pulse width modulation (PWM) algorithm. Since PWM enables to vary the duty cycle of each inverter leg separately, both the magnitude and the direction of the on average applied voltage vector can be varied continuously. The third control scheme combines the features of the previous schemes: for each of the six active voltage vectors of the 2L-VSI, the optimal duty cycle is determined. A cost function is used to select the most optimal pair of voltage vector and duty cycle. Simulation results are used to compare the performance of the three controllers. Implementation of the algorithms on a Field Programmable Gate Array (FPGA) enables an experimental verification of the conclusions on a test set-up.**

***Keywords*— Finite-Set Model Based Predictive Control, Deadbeat Control, Axial Flux Permanent Magnet Synchronous Machines, FPGA**

I. Introduction

The emergence of new and faster control hardware platforms like FPGAs made predictive control a feasible candidate for control of electric drives with time constants in the millisecond range [1]. Hence, predictive control can be employed to control the torque on the rotor of an AFPMSM. This torque control can be accomplished by direct regulation of both torque and stator flux, the so-called Predictive Torque Control (PTC). However, the direct link between torque and flux makes this control strategy relatively complicated and yields quite high Joule losses. Field oriented control (FOC), also known as vector control, offers a full-fledged alternative to regulate the torque by means of PCC. To apply FOC, the AFPMSM is described in the qd-reference frame, rotating in synchronism with the rotor flux provided by the permanent magnets (PMs). The stator current is split into two orthogonal, separately controlled components: i_q is responsible for the torque control, while i_d defines the flux. In FOC for PMSMs, the rotor flux can be chosen to be completely defined by the PMs, hence i_d is controlled to be zero, resulting in a decline in the Joule losses compared to PTC.

A predictive control strategy that is frequently mentioned in the context of inverter fed AC machines - and is the first predictive control scheme that will be implemented in this article - is Finite-Set Model Based Predictive Control (FS-MBPC) [1,2,3]. This technique is very well-suited to control systems with inherently discrete control signals. One of its major drawbacks, however, is the large current ripple resulting from the fact that only the voltage vectors directly available at the output of the inverter can be applied to the machine during one update period T_u of the controller. A second type of predictive controller, called deadbeat control, solves this issue by making use of PWM, hence enabling a continuous variation of both magnitude and direction of the on average applied voltage vector during T_u, and thus reducing the current ripple [3,4]. The third and last control algorithm that will be discussed and compared in this article is a hybrid version of the two previous controllers, enabling a continuous variation of the amplitude of the voltage vectors [3,5].

II. Machine Model Description

In order to obtain a proper predictive controller, an adequate system model must be available, describing the behavior of both the AFPMSM and the inverter.

A. AFPMSM Model

Since an AFPMSM possesses no rotor windings, the dynamic electrical equations only need to describe the dynamic operation of the stator windings. The state-space model of the machine in the synchronous qd-reference frame is expressed as:

$$\begin{bmatrix} i_q^{k+1} \\ i_d^{k+1} \end{bmatrix} = \begin{bmatrix} 1-\frac{R_\mathrm{s}T_\mathrm{u}}{L_q} & \omega^k T_\mathrm{u} \\ -\omega^k T_\mathrm{u} & 1-\frac{R_\mathrm{s}T_\mathrm{u}}{L_d} \end{bmatrix} \begin{bmatrix} i_q^k \\ i_d^k \end{bmatrix} + \begin{bmatrix} \frac{T_\mathrm{u}}{L_q} & 0 \\ 0 & \frac{T_\mathrm{u}}{L_d} \end{bmatrix} \begin{bmatrix} v_q^k \\ v_d^k \end{bmatrix} + \begin{bmatrix} \omega^k T_\mathrm{u} i_\mathrm{mag} \\ 0 \end{bmatrix} \quad (1)$$

The involved machine parameters are tabulated in Table 1. T_u represents the update period of the controller and ω^k the electrical speed at discrete time instant k. i_mag is a constant equivalent current in the d-axis, representing the PMs and resulting in the same flux level as generated by the PMs. The current components i_q, i_d are considered as the state variables, and the voltage components v_q, v_d as the inputs.

The mechanical dynamics are described by the equation of motion:

$$\Omega^{k+1} = \Omega^k + \frac{T_\mathrm{u}}{J}\left(T_\mathrm{em}^k - T_\mathrm{l}^k\right) \quad (2)$$

L. Verkroost, MSc student in Electromechanical Engineering, Ghent University (UGent), Gent, Belgium. E-mail: Lynn.Verkroost@UGent.be

where Ω represents the mechanical speed and T_l the load torque. The developed electromagnetic torque T_em is directly related to i_q:

$$\mathrm{T}_\mathrm{em}^k = -\frac{3}{2} N_p L_d i_\mathrm{mag} i_q^k \qquad (3)$$

Table 1: Machine parameters

Parameter	Symbol	Value
Number of pole pairs	N_p	8
Rated power [kW]	P_n	4
Rated speed [rpm]	N_n	2500
Rated torque [Nm]	T_n	15
Rated voltage [V]	V_n	152
Stator inductance [mH]	$L_q = L_d$	2.54
Stator resistance [mΩ]	R_s	325
Mechanical inertia [kg·m²]	J	0.0024
Equivalent PM current [A]	i_mag	-43.2

B. *Inverter Model*

The AFPMSM is fed by a 2L-VSI, shown in Figure 1. The inverter model can be used to express the phase voltages in function of the inverter's switch state $[S_a, S_b, S_c]$:

$$v_{x\mathrm{n}} = \left(S_x - \frac{1}{2}\right) V_\mathrm{dc} \qquad (4)$$

where V_dc represents the DC bus voltage and x the phase: $x \in \{a, b, c\}$. The voltage over the load is thus:

$$v_{x\mathrm{o}} = v_{x\mathrm{n}} - v_\mathrm{on} \qquad (5)$$

In case of a symmetrical load, the voltage of the star point o of the load with respect to the center tap n of the DC bus equals the average of the phase voltages:

$$v_\mathrm{on} = \frac{v_{a\mathrm{n}} + v_{b\mathrm{n}} + v_{c\mathrm{n}}}{3} \qquad (6)$$

The voltage components v_q, v_d can then be retrieved by applying the Parke transformation.

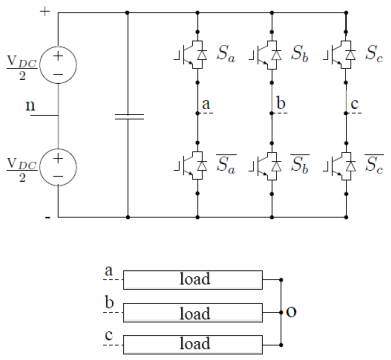


Figure 1: Topology of a 2L-VSI [2]

III. Control Schemes

A. *Finite-Set Model Based Predictive Control*

Model based predictive control (MBPC) is a collective term to denote all control strategies making use of a process model to forecast the output of a system given a certain input. To determine the optimal control action, minimization of a cost function is required. MBPC control allows fast and accurate control of multiple controlled variables, taking into account various constraints in a natural way by means of the adaptable cost function.

MBPC is very well-suited to control systems with inherently discrete control signals. For the 2L-VSI for instance, the number of switch states is limited to a finite set of eight. As a result, each possible state can be evaluated, and minimization of the cost function simplifies to selecting the state yielding the minimum cost. In this way, complicated and time-consuming minimization algorithms can be avoided. To refer to this kind of MBPC, the term finite-set MBPC (FS-MBPC) is used.

The working principle of FS-MBPC is illustrated in Figure 2. Three important steps can be defined.

1) Estimation

Based on measurements of the stator current and the rotor position at update instant k, and knowledge of the optimal control action S_k (determined during the previous time interval $k-1 \rightarrow k$), the stator current components i_q^{k+1} and i_d^{k+1} at time instant $k+1$ can be estimated by means of the machine and inverter model provided in Section II.

2) Prediction

For each of the eight possible switch states S_{k+1}, the controlled variables i_q^{k+2} and i_d^{k+2} at time instant $k+2$ are predicted by means of the system model, starting from the estimates at instant $k+1$.

3) Optimization

Based on the evaluation of a cost function, the most appropriate switch state is selected, bringing the system state closest to its desired value $[i_q^*, i_d^*]$. This optimal switch state S_{k+1} is applied to the system during the entire update period T_u, after which the algorithm is restarted. This is the so-called receding horizon principle. It is noteworthy that only the eight voltage vectors according to the eight different switch states of the 2L-VSI can be applied to the AFPMSM during T_u, yielding a high current ripple.

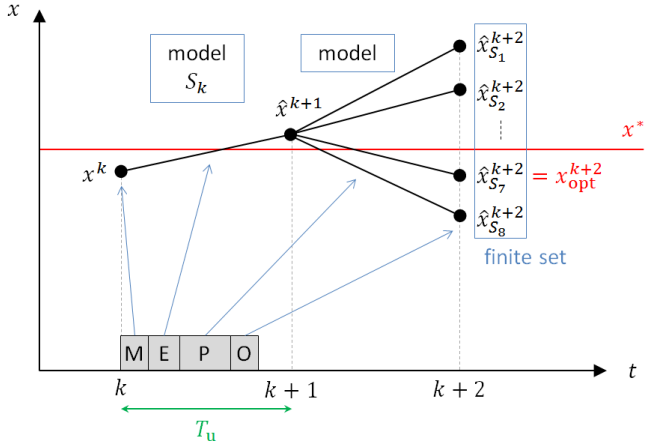


Figure 2: Working principle of FS-MBPC

The estimation step is included, since knowledge of the state variables at instant $k+1$ is required to find the optimal input at instant $k+1$. Measuring this state is not an option, as this would mean that the controller should be able to calculate S_{k+1} in an infinitely small period of time.

The cost function can be defined in various ways. A common choice is to express it as the sum of the squares of the deviations of the controlled variables from their set-points:

$$J^{k+1} = \left(\hat{i}_q^{k+2} - i_q^*\right)^2 + W_I \left(\hat{i}_d^{k+2} - i_d^*\right)^2 \qquad (7)$$

where W_I is a dimensionless weighting factor representing the relative strictness of the control of the current components.

This factor needs to be fine-tuned. Secondary control goals, such as reduction of the switching losses, can be easily expressed by adding extra cost terms.

B. Deadbeat Control

According to [2], deadbeat (DB) control is one of the most uncomplicated predictive controllers. The basic principle comprises the calculation of the required input voltage V^*_{k+1} of the AFPMSM to reduce the error on the controlled variables to zero in a finite amount of steps (generally one). Although DB control makes use of a system model, it is not a model based predictive control, since no cost function is used. In contrast to FS-MBPC, DB control needs a separate PWM process, since the control algorithm does not directly determine the switch state S_{k+1} of the inverter. Its output is a reference value for the required input voltage V^*_{k+1} of the AFPMSM, which can take on any real value. The working principle of DB control is illustrated in Figure 3. Contrary to FS-MBPC, DB control comprises only two major steps.

1) Estimation

The input voltage V_k, applied to the machine at instant k, together with the measured i_q^k, i_d^k and ω^k are used to calculate the system state at $k+1$ by means of the system model provided in Section II.

2) Deadbeat

To obtain the voltage V^*_{k+1} required to bring $\hat{\imath}_q^{k+2}$ and $\hat{\imath}_d^{k+2}$ to their respective reference values i_q^* and i_d^* in one time step T_u, it is assumed that the stator current equals its set-point at instant $k+2$: $\hat{\imath}_q^{k+2} = i_q^*$ and $\hat{\imath}_d^{k+2} = i_d^*$. A model inverse solution is then used to compute the voltage components to be applied at $k+1$:

$$\begin{aligned} v_q^{k+1} &= R_\mathrm{s}\hat{\imath}_q^{k+1} + \frac{L_q}{T_\mathrm{u}}\left(i_q^* - \hat{\imath}_q^{k+1}\right) - \omega^k L_d\left(\hat{\imath}_d^{k+1} + i_\mathrm{mag}\right) \\ v_d^{k+1} &= R_\mathrm{s}\hat{\imath}_d^{k+1} + \frac{L_d}{T_\mathrm{u}}\left(i_d^* - \hat{\imath}_d^{k+1}\right) + \omega^k L_q \hat{\imath}_q^{k+1} \end{aligned} \tag{8}$$

The reference voltages v_{xo} $(x \in \{a,b,c\})$ are obtained by applying the inverse Parke transformation. Eventually, a PWM algorithm determines the sequence of switch states that needs to be applied during the update period T_u. Since the PWM enables to vary the duty cycle of each inverter leg separately, both the magnitude as well as the direction of the on average applied voltage vector become variable.

The DB control algorithm does not involve a cost function. The tedious activity of tuning this cost function is hence avoided. The other side of the coin is that the option to impose secondary control goals disappears as well.

C. FS-MBPC with Duty Cycle Calculation

Although the DB controller enables to change the magnitude and the direction of the on average applied voltage vector during T_u, it also disables the opportunity to impose secondary control goals by means of a cost function. Therefore, a controller combining these two features is proposed here. The structure of the algorithm is similar to that of FS-MBPC, with the only difference that an optimal duty cycle (DC) for the six feasible active voltage vectors is determined as well. In [3], this technique is called Two-Configuration Predictive Control. A similar algorithm is applied in [5] to control the torque of an induction motor drive.

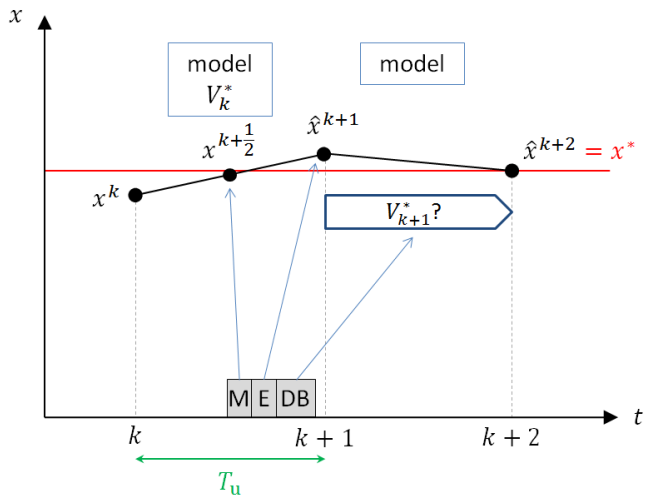


Figure 3: Working principle of DB control

Similar to FS-MBPC, three important steps can be defined in the working principle.

1) Estimation

This step is identical to the estimation step of FS-MBPC.

2) Prediction

Contrary to FS-MBPC, only the six active voltage vectors are evaluated separately. Just a fraction of the control period is allocated to this active voltage vector, the rest of the time a null voltage is applied. Therefore, two slopes must be calculated in each of the six steps: the current slope s_i $(i = 1 \ldots 6)$ if the active vector is applied, and the current slope s_0 if a zero vector is applied. Those slopes can be calculated by means of the dynamical equations (1).

$$\begin{cases} s_{q,0}^{k+1} = \dfrac{1}{L_q}\left[-R_\mathrm{s}\hat{\imath}_q^{k+1} + \omega^k L_d\left(\hat{\imath}_d^{k+1} + i_\mathrm{mag}\right)\right] \\ s_{q,i}^{k+1} = \dfrac{1}{L_q}\left[v_{q,i}^{k+1} - R_\mathrm{s}\hat{\imath}_q^{k+1} + \omega^k L_d\left(\hat{\imath}_d^{k+1} + i_\mathrm{mag}\right)\right] \end{cases} \tag{9}$$

$$\begin{cases} s_{d,0}^{k+1} = \dfrac{1}{L_d}\left[-R_\mathrm{s}\hat{\imath}_d^{k+1} - \omega^k L_q \hat{\imath}_q^{k+1}\right] \\ s_{d,i}^{k+1} = \dfrac{1}{L_d}\left[v_{d,i}^{k+1} - R_\mathrm{s}\hat{\imath}_d^{k+1} - \omega^k L_q \hat{\imath}_q^{k+1}\right] \end{cases} \tag{10}$$

The optimal duration $t_{\mathrm{opt},i}^{k+1}$ for the active voltage vector under consideration is determined according to the deadbeat principle:

$$i_q^* = \hat{\imath}_q^{k+2} = \hat{\imath}_q^{k+1} + s_{q,0}^{k+1}\left(T_\mathrm{u} - t_{\mathrm{opt},i}^*\right) + s_{q,i}^{k+1} t_{\mathrm{opt},i}^* \tag{11}$$

and consequently:

$$t_{\mathrm{opt},i}^* = \frac{i_q^* - \hat{\imath}_q^{k+1} - s_{q,0}^{k+1} T_\mathrm{u}}{s_{q,i}^{k+1} - s_{q,0}^{k+1}} \tag{12}$$

$$\begin{cases} t_{\mathrm{opt},i}^{k+1} = 0, & \text{if } t_{\mathrm{opt},i}^* < 0 \\ t_{\mathrm{opt},i}^{k+1} = t_{\mathrm{opt},i}^*, & \text{if } 0 \le t_{\mathrm{opt},i}^* \le T_\mathrm{u} \\ t_{\mathrm{opt},i}^{k+1} = T_\mathrm{u}, & \text{if } T_\mathrm{u} < t_{\mathrm{opt},i}^* \end{cases} \tag{13}$$

Eventually, the current components at instant $k+2$ can be predicted based on s_0^{k+1}, s_i^{k+1} and $t_{\mathrm{opt},i}^{k+1}$.

$$\begin{aligned} \hat{\imath}_q^{k+2} &= \hat{\imath}_q^{k+1} + s_{q,0}^{k+1}\left(T_\mathrm{u} - t_{\mathrm{opt},i}^{k+1}\right) + s_{q,i}^{k+1} t_{\mathrm{opt},i}^{k+1} \\ \hat{\imath}_d^{k+2} &= \hat{\imath}_d^{k+1} + s_{d,0}^{k+1}\left(T_\mathrm{u} - t_{\mathrm{opt},i}^{k+1}\right) + s_{d,i}^{k+1} t_{\mathrm{opt},i}^{k+1} \end{aligned} \tag{14}$$

3) Optimization

This step is identical to the optimization step of FS-MBPC. The selected null vector ([0 0 0] or [1 1 1]) is the one that requires the lowest number of switches to change their state.

Contrary to DB control followed by PWM, the optimization of the duty cycle only allows to change the magnitude of the applied voltage vector, and not its direction.

The difference in switching between the three proposed control strategies is illustrated in Figure 4 for further clarification.

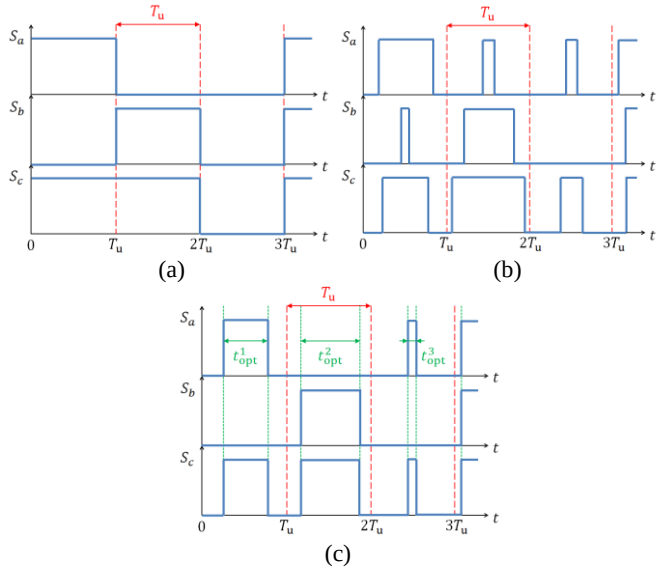


Figure 4: Difference in switching between (a) FS-MBPC, (b) DB control, (c) FS-MBPC with DC calculation

IV. Simulation Results

To compare the performance of the predictive controllers, they are simulated in a MATLAB®&Simulink® environment. A DC bus voltage V_{dc} of 250V is applied, and the mechanical speed of the machine is maintained at N = 1000 rpm. The chosen update frequency f_u amounts to 10 kHz.

The simulation results in Figure 5 show that all controllers are able to track their references well. The difference in current ripple - and thus torque ripple - strikes immediately. The finite set of applicable voltage vectors for FS-MBPC clearly causes a higher current ripple than the large variety of voltage vectors that can be selected as input for DB control. Adding the calculation of t_{opt} to the FS-MBPC algorithm significantly improves the ripple. Due to its high current ripple, the Joule losses for standard FS-MBPC are slightly higher than for the other two controllers. The high control quality obtained by DB control results in low harmonic distortion of the stator currents. It should be noted, however, that the DB controller displays a small systematic error – called bias – between its reference and simulated current, especially for i_d.

Standard FS-MBPC excels when it comes to switching losses. Since standard FS-MBPC does not allow the switch state to be changed during the entire update period, while this is possible for the other two algorithms, the switching frequency of FS-MBPC will be relatively low. However, the switch state changes that do take place are very likely to violate the pulse polarity consistency rule (PPCR), hence heavily burdening the machine isolation. The higher switching frequency when DC calculation is added, and especially when DB control is applied, causes the PPCR to be fulfilled more often.

As to dynamic behavior, only the DB controller shows distinct overshoot peaks in i_q when a step is applied in its reference value. The rise time required to bring i_q to its reference value is smallest for the standard FS-MBPC.

To get an idea of the parameter sensitivity of the control algorithms, their performance is tested when inaccurate parameters are passed on to the controllers. It appears that the DB controller is most prone to instability in case the system model is not accurate, since the exact motor parameters are required to compute the optimal voltage vector. FS-MBPC on the contrary only makes use of the eight input vectors directly available at the output of the 2L-VSI, rendering this control strategy more robust. After all, a small deviation in the predicted state due to parameter mismatch will only occasionally result in the selection of a different voltage vector. Since FS-MBPC with DC calculation combines the FS-MBPC and DB strategies, its stability features lie somewhere in between these two strategies.

An overview of the relative ratings of the different control algorithms based on their simulation results is given in Table 2.

Table 2: Strengths and weaknesses of the controllers, based on simulations

	FS-MBPC	DB	FS-MBPC with DC
ripple	-	+	0
bias	0	-	+
Joule losses	-	0	+
switching losses	+	-	0
PPCR	-	+	0
THD	-	+	0
overshoot	+	-	+
rise time	+	0	-
robustness	+	-	0

V. Experimental Verification

To validate the real-life performance of the controllers, the same tests as in Section IV are performed on a real AFPMSM, of which the specifications are given in Table 1. The control algorithms are implemented on a Xilinx® Kintex®-7 XC7K325T FPGA embedded in a dSPACE MicroLabBox. The FPGA is programmed using the Xilinx System Generator blockset in a MATLAB®&Simulink® environment. The AFPMSM is connected to an induction machine, ensuring the constant speed of the AFPMSM.

The experimental results are shown in Figure 6. All controllers are able to follow the i_q reference. i_d however, deviates from zero. A possible explanation might be that the parameter L_q is slightly underestimated. The combination of a large ω and i_q in the term representing the back EMF of the machine in the electrical equations (1) renders i_d indeed negative in this case.

Although the current ripple has generally increased due to measurement noise, the difference between FS-MBPC and DB control still catches the eye immediately. However, the results of FS-MBPC with DC calculation are slightly disappointing: the current ripple did reduce on average, but the maximal ripple is still high. Furthermore, i_q shows more bias due to the optimization of the duty cycle. The bias of the DB controller is also more pronounced in the experimental result than in

simulation. This decline in control quality compared to the quality in simulation translates into a higher harmonic distortion of the stator currents as well.

The conclusions concerning the switching losses and violations of the PPCR drawn from the simulation results are endorsed by the experimental results.

A study of the transient performance, however, leads to the conclusion that the DB controller shows less overshoot than in simulation, but it requires the longest rise time. FS-MBPC with DC calculation appears to be the fastest controller in real-life. However, the differences in rise time are rather small.

A very surprising result is that the DB control exhibits the highest level of robustness against parameter mismatch in reality. FS-MBPC with DC calculation appears to be the most parameter sensitive.

The relative rating of the controllers based on the measurements is summarized in Table 3. In general, the features showing up in the simulations are observed in the experiments as well. The ratings that did change, however, compared to Table 2, received a green color when they were upgraded and a red color when they were downgraded.

VI. Conclusion

Three different predictive current controllers for an AFPMSM were presented and compared in this paper. The first one, FS-MBPC, resulted in large torque ripple, due to its finite set of feasible voltage vectors that can be applied during the update period of the controller. The second type, the deadbeat controller, solved this issue by synthesizing the optimal required voltage vector by means of PWM, but at the expense of higher switching losses and less flexibility in the expression of control goals. The third controller, FS-MBPC with DC calculation, combined the advantages of the previous control algorithms. The performance of the three controllers was validated by means of simulation in MATLAB®&Simulink® and verification on a test set-up by implementing the control algorithms on an FPGA.

Table 3: Strengths and weaknesses of the controllers, based on measurements

	FS-MBPC	DB	FS-MBPC with DC
ripple	-	+	0
bias	+	-	0
Joule losses	-	+	+
switching losses	+	-	0
PPCR	-	+	0
THD	0	+	-
overshoot	+	-	+
rise time	0	-	+
robustness	0	+	-

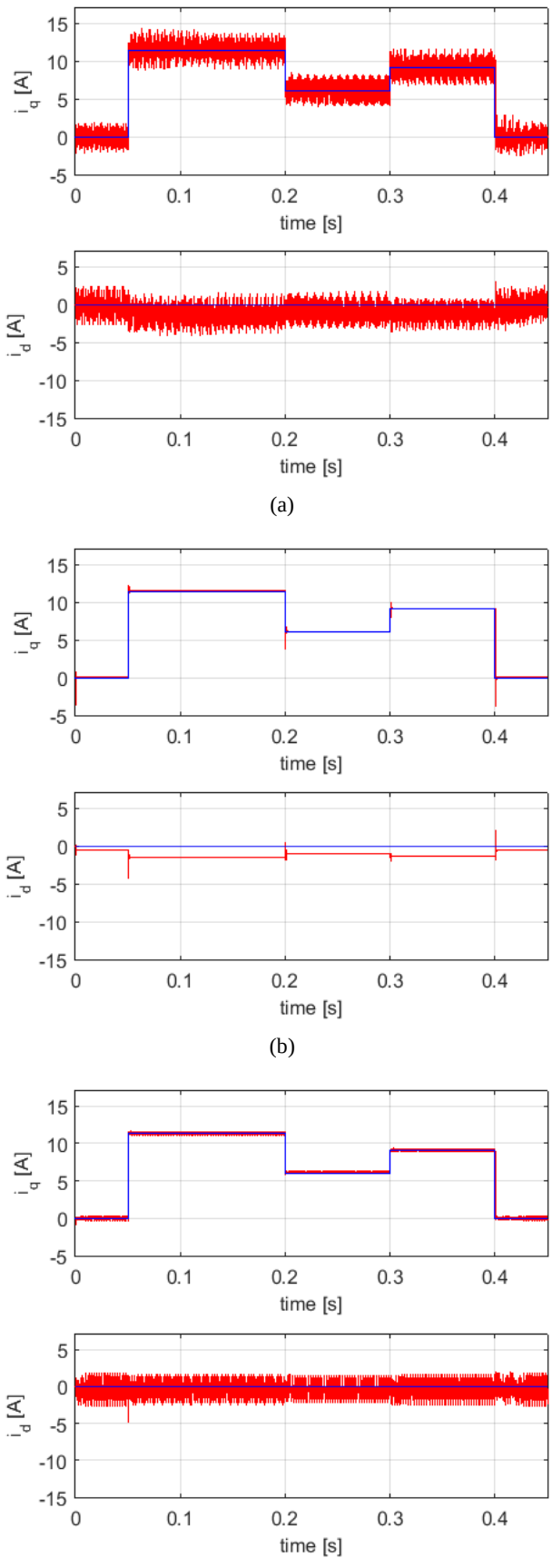


Figure 5: Simulation results for the stator current components for (a) FS-MBPC, (b) DB control, (c) FS-MBPC with DC calculation (red: simulation, blue: reference)

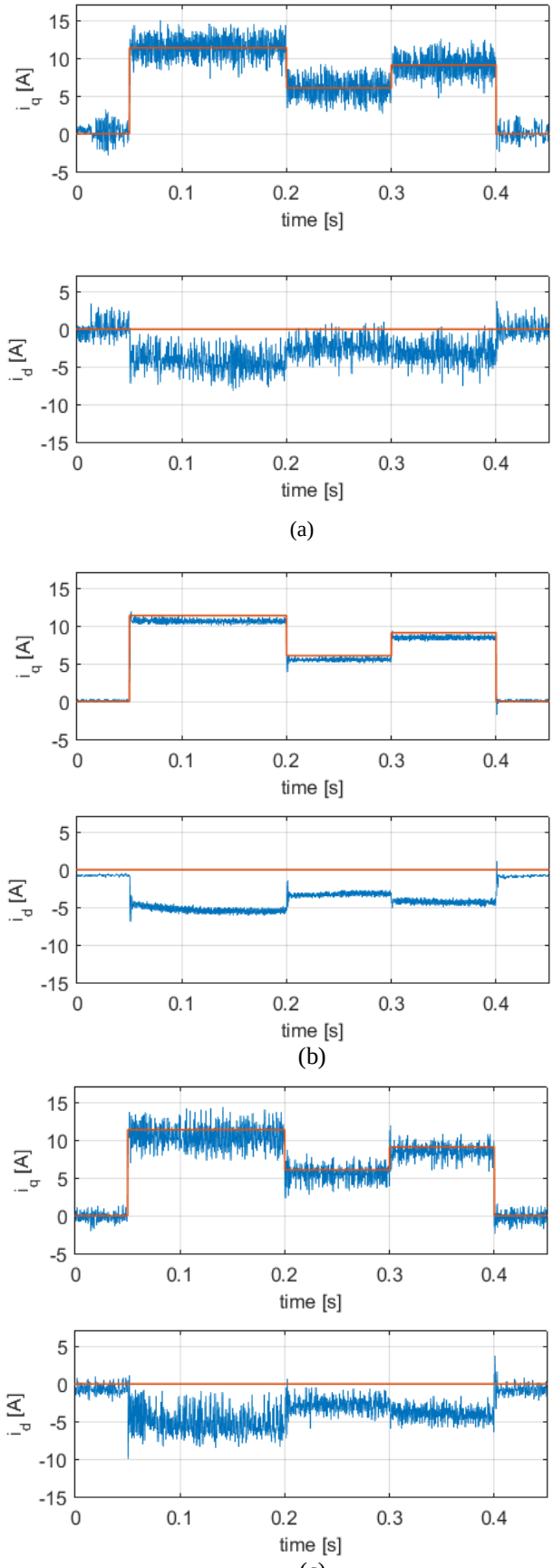


Figure 6: Experimental results for the stator current components for (a) FS-MBPC, (b) DB control, (c) FS-MBPC with DC calculation (blue: measurement, red: reference)

REFERENCES

[1] T. J. Vyncke, S. Thielemans, and J. A. Melkebeek, "Finite-set model-based predictive control for flying-capacitor converters: Cost function design and efficient FPGA implementation," *IEEE Transactions on Industrial Informatics*, vol. 9, pp. 1113-1121, May 2013.

[2] T. Vyncke, *Voorspellende regelaars voor directe koppelcontrole van draaiveldmachines*. PhD thesis, Ghent University, 2012.

[3] F. Morel, X. Lin-Shi, J.-M. Rétif, B. Allard, and C. Buttay, "A comparative study of predictive current control schemes for a permanent -magnet synchronous machine drive," *IEEE Transactions on Industrial Electronics*, vol. 56, no. 7, pp. 2715-2728, 2009.

[4] A. D. Alexandrou, N. K. Adamopoulos, and A. G. Kladas, "Development of a constant switching frequency deadbeat predictive control technique for field-oriented synchronous permanent-magnet motor drive," *IEEE Transactions on Industrial Electronics*, vol. 63, pp. 5167-5175, Aug. 2016.

[5] Y. Zhang and H. Yang, "Model predictive torque control of induction motor drives with optimal duty cycle control," *IEEE Transactions on Power Electronics*, vol. 29, no. 12, pp. 6593-6603, 2014.

Contents

List of Figures

List of Tables

List of Abbreviations & Symbols

Abbreviations

2L-VSI	two-level voltage source inverter
(AF)PMSM	(axial flux) permanent magnet synchronous machine
DB	deadbeat
DC	1. direct current; 2. duty cycle
DTC	direct torque control
EMF	electromotive force
FOC	field oriented control
FS-MBPC	finite-set model based predictive control
IPMSM	interior permanent magnet synchronous machine
KPI	key performance indicator
MAE	mean absolute error
MAR	mean absolute ripple
MIMO	multiple-input multiple-output
MSE	mean square error
MSR	mean square ripple
PCC	predictive current control
PM	permanent magnet
PPCR	pulse polarity consistency rule
PTC	predictive torque control
SPMSM	surface permanent magnet synchronous machine
SVM	space vector modulation
(SV)PWM	(space vector) pulse width modulation
THD	total harmonic distortion
YASA	yokeless and segmented armature

Only the abbreviations that are used repeatedly are included in this list.

Symbols

prefix:

Δ	dimensionless deviation from the reference value

quantities:

f_s	[kHz]	sampling frequency
f_switch	[kHz]	switching frequency of the inverter
f_u	[kHz]	update frequency of the controller
S	[-]	switch state of the inverter
V^*	[V]	reference value for the input voltage of the AFPMSM

subscripts:

k	discrete time index
l	boundary of the lower tolerance band
q, d	quadrature and direct axis components
ref	value used to transform cost terms and KPIs into p.u. values
s	stator
u	boundary of the upper tolerance band

superscripts:

$*$	reference value for a controlled variable
k	discrete time index

Matrices and vectors are indicated by boldface symbols. If the space vector characteristic of a quantity needs to be emphasized, the concerning symbol is underlined. Estimates are indicated by a hat above their symbol. Only the symbols that are used repeatedly and may cause confusion when misinterpreted are included in this list, others are all declared in the text.

Chapter 1

Introduction

Our society consumes too much energy. If our present consumption rate is maintained, running out of fossil fuel sources is just a matter of time. On top of that, burning of fossil fuels causes CO_2 emissions to go through the roof. Increasing evidence of the link between emissions and climate change has stimulated the search for alternative energy sources. Renewable energy sources (such as wind, solar and hydraulic energy) are gaining increasing attention nowadays. Electricity is often employed as energy carrier in this context. Just think of sustainable energy applications, like windmills and electric cars, requiring highly reliable electrical machines.

As part of the same mindset, the current development in rotating electrical machine technology is dedicated to the three E´s: *ecodesign*, *energy* and *economy* [1]. As to *ecodesign*, the development of new electric motors and generators has a significant role to play, tackling the new requirements imposed by their new applications. The axial flux permanent magnet synchronous machine (AFPMSM) is a typical example. Its high power density and efficiency makes this technology very suitable for integration in electrical drive trains for industrial applications, transport applications and sustainable energy conversion.

The second E, *energy* efficiency, requires efficient exploitation of electrical machines. This is the point where new types of controllers come into play. Predictive control for instance - making use of a system model to predict the future system behavior - is one of these relatively new techniques to control electrical machines. It has the advantage of very fast dynamics, while simultaneously taking multiple constraints into account, such as the reduction of Joule and switching losses.

Whereas ecodesign and energy efficiency tend to make electrical machines and their drives more expensive, an acceptable *economic* cost is still indispensable. The commercialization of new and faster control hardware platforms, for instance, enabled the deployment of computationally demanding control schemes for electrical drives with time constants in the millisecond range. Hence, predictive control became a realistic candidate for application to an AFPMSM, for instance. A Field Programmable Gate Array (FPGA) - a very powerful micro-controller able to process a large amount of data in a small time span - is an example of such a new hardware platform.

The main objective of this master's dissertation is to examine the possibilities that predictive control offers to the rotor torque regulation of an AFPMSM fed by a two-level voltage source inverter. This converter consists of a series of switches (transistors) that can connect the stator phases of the AFPMSM to the positive or negative voltage of a DC bus. The current through the windings, and thus the torque on the rotor, is determined by the sequence in which the transistors are switched. The main objective of the controller is to select an optimal switching sequence in order to achieve a given torque demand. The faster the machine can produce the required torque, the better. However, the controller must take into account limitations such as

maximum voltage or current, and the feasible inverter states as well.

In a first step, the examined control algorithms will be tested and fine-tuned in simulation in a MATLAB®&Simulink® environment. Finally they will be implemented on-line by means of an FPGA. Hence the performance of the control strategies can be compared by applying them on the test set-up already present in the lab.

Chapter 2

Axial Flux Permanent Magnet Synchronous Machines

2.1 Introduction

Climate change continues to become a driving force on different aspects of social living. Reducing greenhouse gas emissions is one of the main objectives to create a more sustainable society. Since the transport and energy sector are major emitters of CO_2, research towards the development and integration of electric motors in vehicles and renewable energy conversion has bloomed. The development of new electric motors and generators plays an important role in this focus on ecology. Studies have shown that new adapted machine topologies are better qualified to fulfill the present-day requirements than the classical ones. An example of such a new machine topology is the axial flux permanent magnet synchronous machine (AFPMSM), which is particularly suitable for electric vehicles and small to medium power range wind energy systems.

AFPMSMs exist in different topologies. In this master's dissertation an AFPMSM with a yokeless and segmented armature (YASA) topology is used, with surface mounted permanent magnets. Therefore this chapter will focus on this particular topology. More exhaustive information about the design and the different topologies of AFPMSMs can be found in [1].

2.2 Working Principle

The operation principle of an AFPMSM is similar to that of a radial flux permanent magnet machine, except that the magnets are now magnetized in the axial instead of the radial direction, leading to an axial flux instead of a radial one.

In generation mode, an external force rotates the rotor disc with respect to the stator. In this way, a time varying magnetic flux is generated in the stator teeth, inducing a voltage - the back electromotive force (EMF) - in the stator windings. Power is transferred from the external force imposing the rotation to electric power dissipated in the load connected to the stator windings.

In motor working, however, an externally imposed stator current results in a magnetic field that interacts with the magnetic field originating from the permanent magnets. The result is a rotation of the rotor disc. Electric power is thus transferred into mechanical power, available at the shaft of the machine.

2.3 Yokeless And Segmented Armature Topology

The AFPMSM used throughout this master's dissertation has a yokeless and segmented armature (YASA) topology, which is illustrated in Figure 2.1.

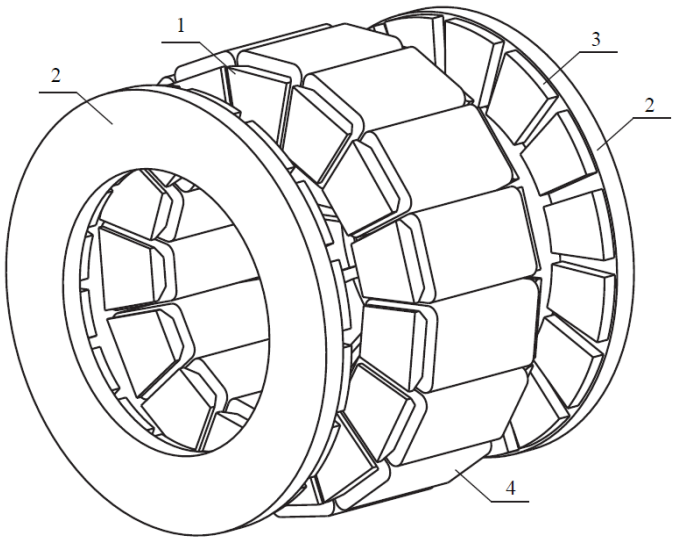


Figure 2.1: The YASA topology: (1) stator consisting of multiple stator core elements; (2) rotor; (3) permanent magnet; (4) tooth coil winding [1]

The machine consists of one stator and two rotors. The permanent magnets on both rotors are magnetized in equal direction, in the same way as for a so-called north south (NS) topology. Figure 2.2 gives a planar representation of this topology, showing the axial antisymmetry in the machine. Some magnetic flux paths are indicated as well. Since the magnetic symmetry in the machine does not require a closing path for the magnetic flux through the stator yoke, this yoke is removed entirely in the YASA axial flux machine, resulting in the existence of individually segmented armature elements.

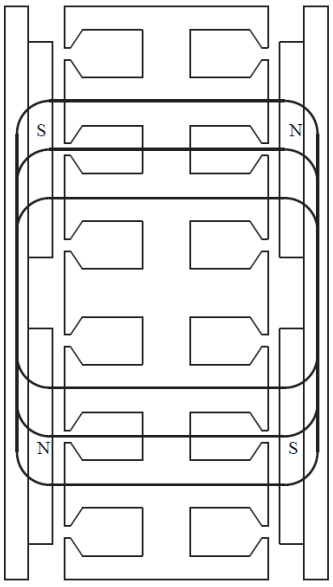


Figure 2.2: Planar version of the NS torus machine topology, with indicated magnetic flux paths [1]

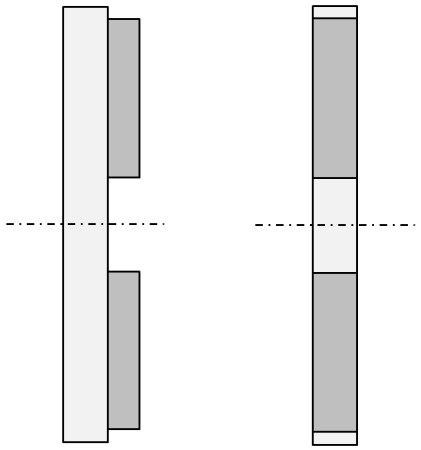
Around each individually segmented armature element a double layer concentrated fractional

pitch winding is wound, which is very often called a tooth coil winding. Compared to the original NS torus topology with a short stator yoke, the winding arrangement complexity is significantly reduced.

The absence of a stator yoke has an advantageous influence on stator core loss and power density as well. The easy winding arrangement enables short end windings and a good filling factor of the conductor in the slots when using wires with a rectangular cross section. Hence the YASA topology exhibits a superior power density and an excellent energy efficiency compared to other axial flux machine topologies.

2.4 Mounting of the Permanent Magnets

There are two typical possibilities to mount the magnets to the rotor. The first possibility is to attach them to the surface of the rotor. This type of PMSM (permanent magnet synchronous machine) is usually denoted as SPMSM (surface permanent magnet synchronous machine) and is shown in Figure 2.3a. Another possibility is to sink the magnets into the rotor, resulting in an IPMSM (interior permanent magnet synchronous machine) illustrated in Figure 2.3b. Since the magnetic permeance of NdFeB magnets is comparable to the one of air, a SPMSM has a smooth air gap from magnetic point of view, without a variance in inductance. An IPMSM on the contrary exhibits a significant variance in inductance, resulting in a reluctance torque. Throughout this master's dissertation, a SPMSM is used.



(a) SPMSM **(b)** IPMSM

Figure 2.3: Mounting of the permanent magnets

2.5 Main Advantages of AFPMSMs

One of the most important advantages of PMSMs in general is the fact that they have a good energy efficiency. Usually permanent magnets with a high energy density, such as neodymium (NdFeB) magnets, are used to generate the magnetic field. As a result, no field excitation current is necessary and the corresponding copper losses are absent. Large torque-ampere ratio's are obtained.

Besides, PM motor drives are relatively easy to control compared to, for instance, induction motor drives. Therefore, PMSMs are often used when fast torque response and high-performance

operation are required.

A major advantage of AFPMSMs is their compactness. Their axial length is small in comparison with that of a radial machine. Due to their light weight, AFPMSMs have a relatively high power and torque density. Especially in electric vehicle applications, compactness and high power density are crucial properties.

2.6 Conclusion

The electric machine used throughout this master's dissertation is an AFPMSM with a YASA topology and surface mounted permanent magnets. A YASA machine consists of two rotors with permanent magnets and one stator composed of multiple stator core elements, wound with tooth coil windings. Compared to other topologies, YASA machines exhibit an overall better performance: they have the advantage of a good energy efficiency, compactness, and a relatively high power and torque density.

Chapter 3

Predictive Control

3.1 Introduction

The emergence and continuous evolution of digital signal processing in the last half-century offered immense potential for converter control techniques. Besides the already existing analog controllers - such as linear controllers with modulation schemes and non-linear controllers based on hysteresis comparators - new, more advanced control techniques could be developed, in order to improve the transient performance and efficiency of electric drives. Examples are fuzzy, adaptive, sliding mode and predictive controls. The rise of the latter could not have been made possible without the availability of high processing power. After all, predictive controllers make use of a system model to predict the system behavior, involving a large computational burden.

After decades of research, the domain of predictive control comprises a large variety of algorithms. They appear to be very appropriate to handle constraints and non-linear processes. In this chapter, the basic concepts of two of them will be elucidated: model based predictive control and deadbeat control.

3.2 Model Based Predictive Control

Model based predictive control (MBPC) is a collective term to denote all control strategies making use of a process model to forecast the output of a system given a certain input. To determine the optimal control action, minimization of a cost function is required. MBPC allows fast and accurate control of multiple controlled variables.

The MBPC-principle is summarized as follows in [3]:

1. Use the process model to predict the evolution of the process output as a function of the intended control actions.
2. Minimize a specific cost function over these control actions. The cost function represents the requirements for a good control and includes the difference between the desired and predicted process outputs. The control action resulting in the lowest cost will be applied to the actual system.

The various algorithms differ mainly in the type of model used to represent the process and its disturbances (e.g. linear or non-linear), and the cost function to be minimized (e.g. with or without the required control effort included). This flexibility of the model and the cost function is a great advantage, since it enables to include multiple variables, system constraints, additional

control objectives, and basically every characteristic that can be measured and mathematically modeled. In [4], the difference between using a classical, parametric machine model and a back propagation artificial neural network is studied. A comparison between different types of cost functions is made as well.

MBPC is very well-suited to control systems with inherently discrete control signals. For power electronic converters for instance, the number of switch states is limited to a finite set. As a result, each possible state can be evaluated and minimization of the cost function simplifies to selecting the state leading to the minimum cost. In this way complicated and time-consuming minimization algorithms can be avoided, and the algorithm design is very intuitive. To refer to this kind of MBPC the term finite-set MBPC (FS-MBPC) is used. The authors of [5] use the name hybrid torque control, since the electric machine constitutes a continuous process, while the state variables of the converter are discrete. In [6] the algorithm is denoted as finite alphabet model predictive control (FAMPC).

3.2.1 Working Principle

The working principle of FS-MBPC is illustrated in Figure 3.1. As only one measurement will be taken during an update period T_u of the controller, the sampling period T_s is equal to T_u.

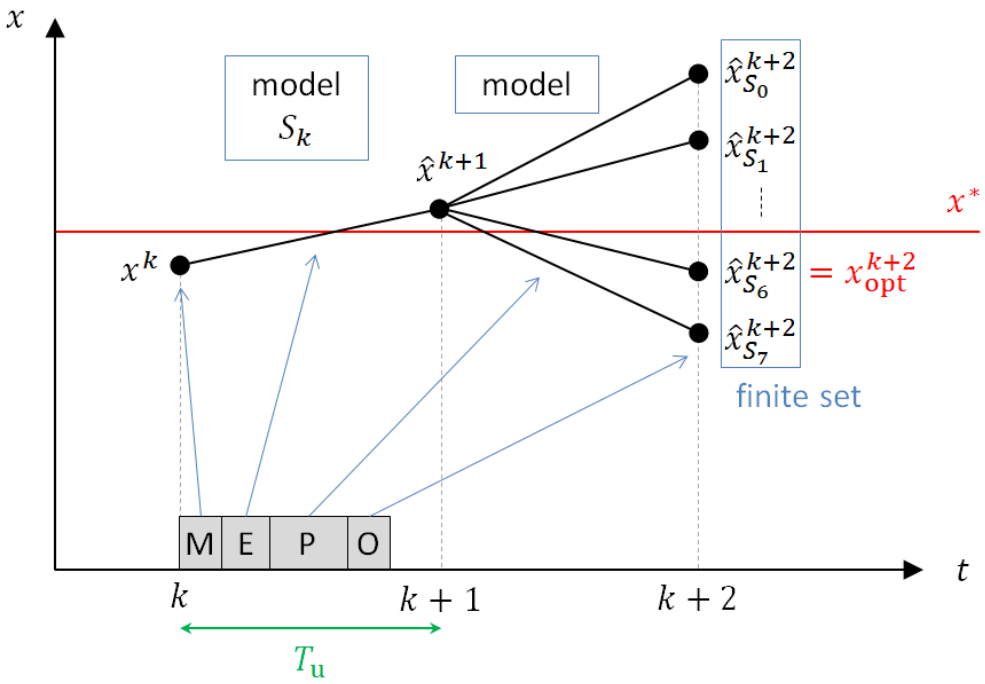


Figure 3.1: Working principle of FS-MBPC, visualizing the basic steps: measurement (M), estimation (E), prediction (P) and optimization (O) (Adapted from [7])

Three steps can be defined: estimation, prediction and optimization.

Estimation

At update instant k, the optimal control action S_k (determined during the previous time interval $k-1 \rightarrow k$) is applied, and the state x_k is measured, providing feedback to the controller. S_k will be kept constant throughout the period $k \rightarrow k+1$, giving the controller one update period

T_u to find the new optimal control action at time instant $k+1$. In order to do so, it requires knowledge of the state x_{k+1} as well. Measuring the state x_{k+1} at time $k+1$ is not an option, as this would mean that the controller should be able to calculate S_{k+1} in an infinitely small period of time. Therefore x_{k+1} is estimated using the system model, starting from x_k and S_k. In [8] this step is called initial state position, while in [9] the term two-step-ahead prediction is used for an estimation step followed by a single prediction step.

Prediction

For every possible control action S_{k+1}, the future state $\hat{x}_{k+2}$ is calculated starting from the estimated state $\hat{x}_{k+1}$. Since there is only a finite set of possible switch states, each of them can be evaluated. The computational burden and control quality will directly depend on the complexity and the accuracy of the model.

Optimization

Based on the evaluation of a cost function, the most appropriate control action is selected, bringing the system state closest to its desired value x^*. As T_u is sufficiently small compared with the dynamical behavior of the system, x^*_{k+2} can be assumed to be equal to x^*_k. Since there is only a finite set of control actions, no complex and time-consuming minimization algorithm is needed: calculating the cost for every single switch state and picking the one leading to the smallest cost suffices. This optimal control action S_{k+1} will then be applied to the system, and the algorithm is restarted. This is the so-called receding horizon principle. It is noteworthy that this algorithm directly selects one - and only one - switch state of the inverter per update period, and this switch state is maintained during the entire update period. In other words: the inverter does not switch during T_u.

An important drawback of FS-MBPC is that the switching frequency may vary - as the switch state will not change at every update instant - making it harder to design a proper electromagnetic interference filter. Moreover, a variable switching frequency can lead to acoustic noise.

The specific application of FS-MBPC for torque and current control will be discussed more extensively in Chapters 5 and 6.

3.2.2 Timing

In Figure 3.1 the measurement of x_k is taken immediately after the control action S_k is applied. In reality, this is not good practice. When a new control action is applied, this introduces transient behavior in the system. Measuring the state during these transients gives useless information to the controller, deteriorating the control quality.

A solution is presented in Figure 3.2. When the control action S_k is applied at instant k, the state $x_{k+1/2}$ is measured half an update period $T_u/2$ later. This means that the controller needs to execute the control algorithm during the remaining half of the update period. Its computation time is halved and in fact only half its capacity is used, which is not very efficient. On the other hand, postponing the state measurement decreases the time step used for the linearization in the estimation step, leading to a more accurate linearized model. This will be commented in greater detail in Section 4.2.3 of the next chapter.

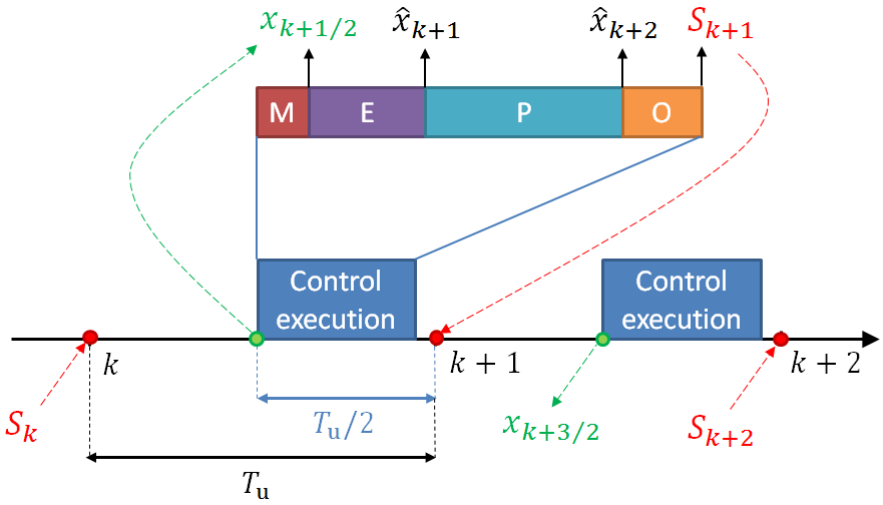


Figure 3.2: Timing of the basic steps in MBPC:
measurement (M), estimation (E), prediction (P) and optimization (O)

Specifically for this master's dissertation, the proposed timing schedule will only have advantages, since it is not the controller but the inverter that imposes limitations on the update frequency. The controller is powerful enough to execute the computations in half an update period, leading to representative measurements and a more accurate linearized model in the estimation step.

3.3 Deadbeat Control

Deadbeat (DB) control is one of the most uncomplicated predictive controllers, according to [2]. The basic principle comprises the calculation of the required control signals to reduce the error on the controlled variables to zero in a finite amount of steps (generally one). Although deadbeat control makes use of a system model, it is not a model based predictive control, since no cost function is used. In contrast to FS-MBPC, DB control needs a separate pulse modulation process, since the control algorithm does not directly determine the switch state S_{k+1} of the inverter. Its output is a reference value for the required input voltage V^*_{k+1} of the machine, which can take on any real value.

3.3.1 Working Principle

The general working principle of a DB controller is illustrated in Figure 3.3. Just like FS-MBPC, a DB controller comprises a measurement and an estimation. But instead of evaluating all possible inputs in a prediction step and selecting the most optimal one in an optimization step, the deadbeat controller computes which input is required to reach the reference in a finite amount of update periods (one period in Figure 3.3). Therefore, DB is less computationally complex than FS-MBPC. The timing explained in Section 3.2.2 also applies to the DB controller and is already taken into account in Figure 3.3.

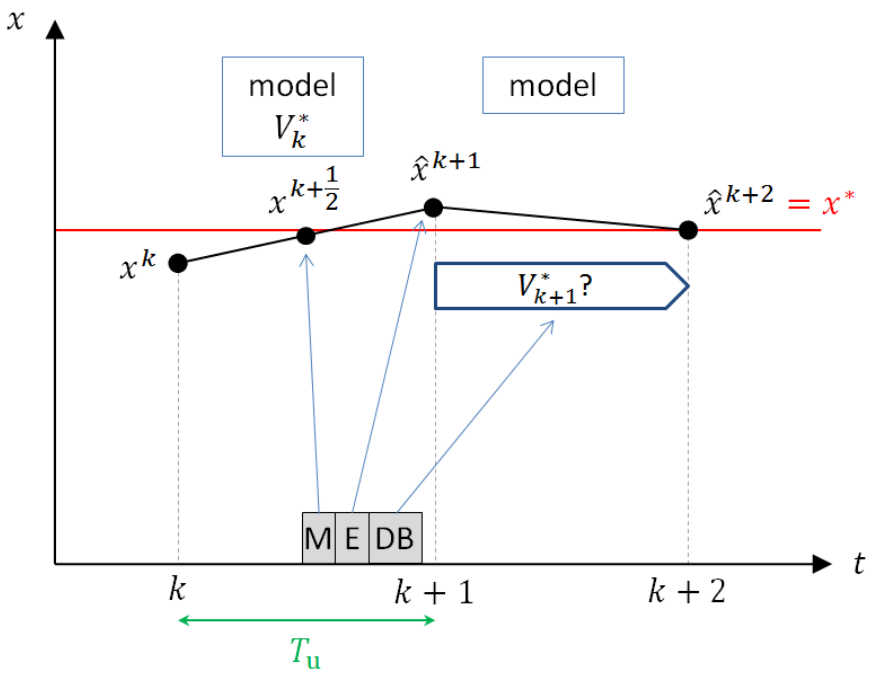


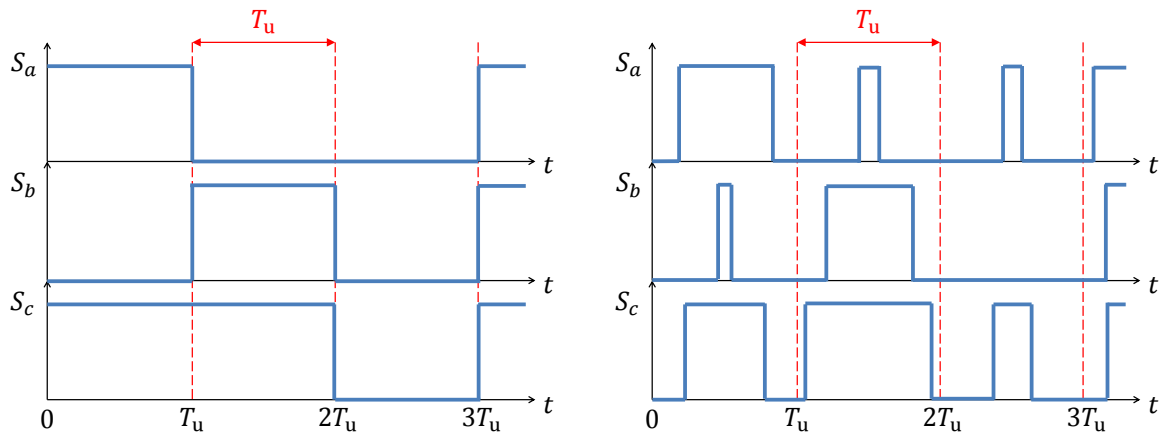
Figure 3.3: Working principle of DB control, visualizing the basic steps: measurement (M), estimation (E) and deadbeat (DB)

Estimation

The first step in the computation of the input voltage V_{k+1}^* required to bring the output to its reference value is very similar to the estimation step in FS-MBPC. The input voltage V_k applied to the machine at instant k and the system outputs measured half an update period later at $k + 1/2$ are used to calculate the output at $k + 1$ by means of the system model.

Deadbeat

The second step in the DB algorithm, however, distinguishes DB control from FS-MBPC. To obtain the required input at $k + 1$, one assumes that the output x_{k+2} equals the set-point x^*. The system model can then be used to compute the required input at $k + 1$. Contrary to FS-MBPC, the DB controller does not determine the optimal switch state of the inverter directly. It determines the reference signal V_{k+1}^* for the input that is required to bring the output to its set-point instead. Because the inverter supply of the PMSM is restricted to only a limited set of switch states, the control action imposed by the deadbeat controller can only very rarely be applied to the PMSM immediately. Therefore a modulator is used in combination with the controller, for instance a classical space vector pulse width modulator (SVPWM), which will be explained in Section 4.3.2. Consequently, the switch state of the inverter will not be constant throughout an update period T_u: the modulator divides T_u in smaller pieces in which the switch state can change. Consequently, for the same T_u, the switching frequency f_{switch} will be much higher for DB control than it will be for FS-MBPC: for DB control $f_{switch} \approx 2 \cdot f_u$ for each phase, while for FS-MBPC $f_{switch} \leq f_u$, as is illustrated in Figure 3.4.



(a) FS-MBPC **(b)** DB control

Figure 3.4: Difference in switchings between FS-MBPC and DB control

Using the DB principle, no cost function is needed to decide which control action needs to be applied. The fact that the burdensome process of the tuning of the shape and the parameters of the cost function is avoided in this way, can be seen as an advantage. However, the freedom to include extra requirements on the control actions (for instance to reduce switching losses) is lost as well.

In Chapter 6 the application of the deadbeat algorithm for current control will be elaborated.

Another topic that needs to be studied is the fact that deadbeat controllers are very sensitive to the model accuracy, which makes them susceptible to stability issues. Section 7.2.4 will delve deeper into this subject.

3.4 Conclusion

This chapter introduced the basic working principles of two predictive controllers: FS-MBPC and DB control.

The three basic steps of FS-MBPC were explained: estimation, prediction and optimization. The output of the algorithm is the required switch state of the inverter.

Contrary to FS-MBPC, a DB controller executes two steps: estimation and deadbeat. The algorithm outputs a reference value for the input voltage of the machine, which is translated by a modulator into the necessary switch states for the inverter.

Some attention was devoted to the timing aspect as well. To avoid measuring transients and to obtain a more accurate linearized model, the state will be measured half an update period after a new control action is applied. A disadvantage of this timing is that only half the capacity of the controller is used.

Chapter 4

Modeling

4.1 Introduction

Predictive torque controllers make use of a system model to forecast the output of a system given a certain input. The accuracy of the used model is not only tightly connected to the performance of the controller, but also to its computational burden. The key is to derive an adequate model, trading off both conflicting requirements. This chapter is devoted to the modeling of the machine and the inverter.

4.2 Machine Model

4.2.1 Assumptions

To obtain a practicable model of an AFPMSM, a number of assumptions is made in [2, 10] to simplify the machine equations considerably:

- The machine is three phase symmetrical.
- The stator windings are sinusoidally distributed in space for the fundamental component.
- There are no skin-effects.
- There is no saturation.
- There are no stator slot-effects, nor stator saliency.

On the contrary, the rotor may contain saliency effects. However, since in this master's dissertation a SPMSM is considered, from magnetic point of view the air gap is smooth, showing no variation in inductance. This is due to the fact that the magnetic permeance of NdFeB magnets is similar to the one of air.

An additional assumption concerns the modeling of these permanent magnets:

- The permanent magnets are replaced by an equivalent two phase rotor winding in the qd-reference frame, attached to the rotor, with an excitation in the d-axis resulting in the same magnetization as caused by the PMs.

It has to be emphasized that none of the assumptions requires a symmetrical sinusoidal three-phase supply. Consequently, the model will be valid for inverter supply as well.

4.2.2 Derivation of the Dynamic Model

In [10] is explained in detail how the dynamic model of the machine is derived. Since there are no rotor windings, the derivation starts from the dynamical electrical equations at time instant i for the three stator phases only:

$$\mathbf{V}^i = \mathbf{R}^i\mathbf{I}^i + \frac{\mathrm{d}\mathbf{L}^i(\theta)\mathbf{I}^i}{\mathrm{d}t} \tag{4.1}$$

The vector $\mathbf{V}^i$ contains the instantaneous values of the three stator voltages, while the vector $\mathbf{I}^i$ describes the phase currents:

$$\mathbf{V}^i = \begin{bmatrix} v_{\mathrm{s}a} & v_{\mathrm{s}b} & v_{\mathrm{s}c} \end{bmatrix}^T \tag{4.2}$$

$$\mathbf{I}^i = \begin{bmatrix} i_{\mathrm{s}a} & i_{\mathrm{s}b} & i_{\mathrm{s}c} \end{bmatrix}^T \tag{4.3}$$

$\mathbf{R}^i$ represents the instantaneous resistance matrix and $\mathbf{L}^i(\theta)$ the instantaneous inductance matrix, depending on the relative angle θ between stator and rotor.

However, in most cases it is not necessary to consider all three stator equations, since there cannot be any zero-sequence power. Indeed: for a delta connected machine the zero-sequence voltages are inherently zero, while for a wye connected machine the neutral is almost never connected, meaning that zero-sequence currents cannot flow. Therefore the zero-sequence equations will be omitted. For this purpose, the non power-invariant Clarke transformation is used [1]:

$$\mathbf{T}_{\mathrm{c}}{}^{-1} = \frac{2}{3}\begin{bmatrix} 1 & -1/2 & -1/2 \\ 0 & \sqrt{3}/2 & -\sqrt{3}/2 \end{bmatrix} \tag{4.4}$$

$$\begin{bmatrix} v_\alpha \\ v_\beta \end{bmatrix} = \mathbf{T}_{\mathrm{c}}{}^{-1}\begin{bmatrix} v_{\mathrm{s}a} \\ v_{\mathrm{s}b} \\ v_{\mathrm{s}c} \end{bmatrix} \tag{4.5}$$

where v_α and v_β are the voltages expressed in the equivalent two-phase stator reference frame. In this way, the number of electrical equations reduces to two, simplifying the model of the machine. However, the equations in the stator reference frame are time dependent, as the angle θ between rotor and stator changes in time. The cause is that the stator reference frame is at standstill, while the rotor is rotating with rotor speed. Transforming the equations to a common reference frame with electrical rotor speed $\omega = d\theta/dt$ using the transformation

$$\mathbf{T}_{\mathrm{r}}{}^{-1} = \begin{bmatrix} \cos\theta & \sin\theta \\ -\sin\theta & \cos\theta \end{bmatrix} \tag{4.6}$$

will transform the $\alpha\beta$-stator variables to qd-variables fixed to the rotor, and thus make the equations stationary. Since a PMSM is considered, the magnetic flux originatig from the permanent magnets rotates at the same speed as the rotor.

Eventually, two non-linear, stationary electrical equations are obtained in the rotor reference frame:

[1] The non power-invariant Clarke transformation preserves the amplitude of currents and voltages, but not the power. The power-invariant Clarke transformation, however, (with coefficient $\sqrt{\frac{2}{3}}$ instead of $\frac{2}{3}$) preserves the power, but not the amplitude of currents and voltages.

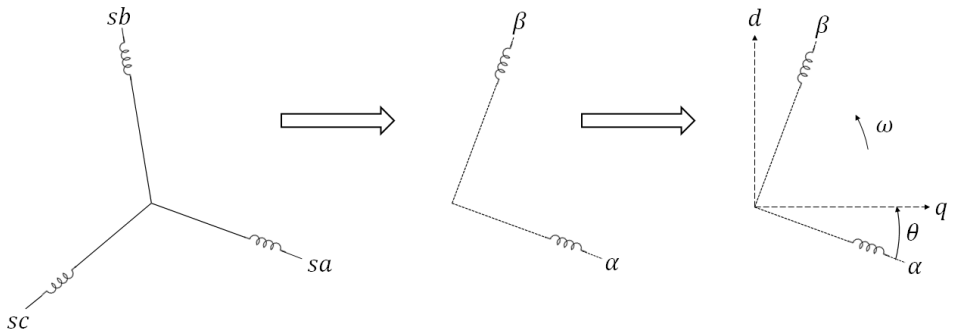


Figure 4.1: Transformation to the *qd*-reference frame

$$\begin{bmatrix} v_q \\ v_d \end{bmatrix} = \mathbf{T}_\mathrm{r}{}^{-1}\mathbf{T}_\mathrm{c}{}^{-1} \begin{bmatrix} v_{sa} \\ v_{sb} \\ v_{sc} \end{bmatrix} \tag{4.7}$$

$$= \begin{bmatrix} R_\mathrm{s} i_q + L_q \frac{\mathrm{d}i_q}{\mathrm{d}t} - \omega L_d (i_d + i_\mathrm{mag}) \\ R_\mathrm{s} i_d + L_d \frac{\mathrm{d}i_d}{\mathrm{d}t} + \omega L_q i_q \end{bmatrix} \tag{4.8}$$

where

- R_s is the stator resistance
- L_q and L_d are the q- and d-axis inductances respectively
- i_q and i_d are the currents in the qd-reference frame
- i_mag is a constant equivalent current in the d-axis, representing the permanent magnets and resulting in the same flux level as caused by the permanent magnets

Figure 4.1 visualizes the transformation to the qd-reference frame.

The machine parameters R_s, L_q and L_d are assumed constant, meaning that the effect of saturation is not included in this model. L_q and L_d are equal since a SPMSM is considered.

The last equation that needs to be included in the dynamic machine model is the equation of motion:

$$J\frac{\mathrm{d}\Omega}{\mathrm{d}t} = T_\mathrm{em} - T_\mathrm{l} \tag{4.9}$$

where

- J is the total inertia of machine and load
- Ω is the mechanical speed, defined as ω/N_p: the electrical speed divided by the number of pole pairs
- T_em is the electromagnetic torque
- T_l is the load torque

The electromagnetic torque T_em is calculated as:

$$T_\mathrm{em} = \frac{3}{2} N_p \Im(\underline{\Psi}_\mathrm{s}^* \cdot \underline{I}_\mathrm{s}) \tag{4.10}$$

$$= -\frac{3}{2} N_p L_d i_\mathrm{mag} i_q \tag{4.11}$$

where $\underline{\Psi}_s$ and $\underline{I}_s$ are the stator flux and current vector respectively. An equivalent scheme and a vector diagram of the AFPMSM are presented in Figures 4.2 and 4.3 respectively.

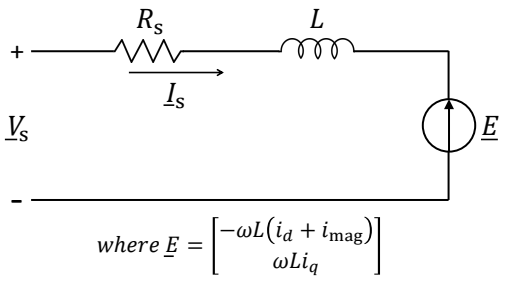


Figure 4.2: Equivalent scheme

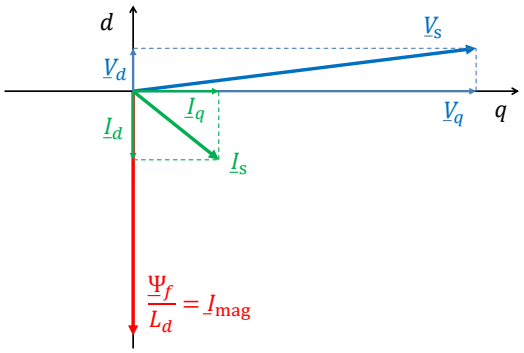


Figure 4.3: Vector diagram

After solving the set of electrical equations (Eq. (4.8)) and the equation of motion (Eq. (4.9)), the qd-variables can be transformed back to the phase variables by applying the inverse transformation:

$$\mathbf{T}_c = \begin{bmatrix} 1 & 0 \\ -1/2 & \sqrt{3}/2 \\ -1/2 & -\sqrt{3}/2 \end{bmatrix} \tag{4.12}$$

$$\mathbf{T}_r = \begin{bmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{bmatrix} \tag{4.13}$$

$$\begin{bmatrix} v_{sa} \\ v_{sb} \\ v_{sc} \end{bmatrix} = \mathbf{T}_c \mathbf{T}_r \begin{bmatrix} v_q \\ v_d \end{bmatrix} \tag{4.14}$$

4.2.3 Discrete Time Model

Since nowadays most controllers are digitally implemented, the machine equations need to be rewritten in a discrete time representation. The time step Δt is chosen to be equal to (half of) the update period T_u of the controllers in this master's dissertation. A smaller T_u will result in a more accurate linearization, but less computation time for the controller. Since the optimization of the speed of the controller is outside the scope of this thesis, f_u will be chosen close to the half of the maximum switching frequency of the inverter, giving the controller sufficient computation time without overloading the inverter when using DB control.

The model will be linearized using the first order Euler approximation for the derivatives:

$$f(x_i, u_i, i) = \frac{x_{i+1} - x_i}{\Delta t} \tag{4.15}$$

where $x_i = x(t)$, $x_{i+1} = x(t + \Delta t)$ and $f(x_i, u_i, i)$ represents the derivative of the quantity x evaluated at time t (or discrete time instant i) using the input u_i and state x_i at that same time instant. The required $f(x_i, u_i, i)$ are calculated by means of the electrical model given by Eq. (4.8). In this way, the value of x at an instant Δt later than the considered moment can be obtained, which is useful in the control algorithms introduced in Chapter 3:

$$x_{i+1} = x_i + \Delta t \cdot f(x_i, u_i, i) \tag{4.16}$$

Hence, the discrete time version of the dynamic electrical equations is:

$$v_q^k = R_\mathrm{s} i_q^k + \frac{L_q}{T_\mathrm{u}} \left(i_q^{k+1} - i_q^k \right) - \omega^k L_d (i_d^k + i_\mathrm{mag}) \tag{4.17}$$

$$v_d^k = R_\mathrm{s} i_d^k + \frac{L_d}{T_\mathrm{u}} \left(i_d^{k+1} - i_d^k \right) + \omega^k L_q i_q^k \tag{4.18}$$

In [11] can be found that the local truncation error (the error made when going from x_i to x_{i+1} due to the used methodology) when solving this differential equation will be of the order $\mathcal{O}(\Delta t^2)$.

In Section 6.4.3, the effect of switching to the modified Euler approximation will be studied.

4.2.4 Machine Parameters

The machine used throughout this master's dissertation is a sixteen-pole machine ($N_p = 8$). The system has a total inertia of $J = 0.0024$ kgm^2. The machine parameters represented in Table 4.1 are deduced from a finite element model for a flux level leading to a flux density of 1.6T in the stator teeth. For this flux level, a good balance is reached between converting an acceptable amount of power given the size of the machine, and avoiding too much losses. Moreover, in this way the machine is not saturated and the linearity assumption will hold.

Table 4.1: Machine parameters

Parameter	Symbol	Value
Number of pole pairs	N_p	8
Rated power [kW]	P_n	4
Rated speed [rpm]	N_n	2500
Rated torque [Nm]	T_n	15
Rated voltage [V]	V_n	152
Stator inductance [mH]	$L_q = L_d$	2.54
Stator resistance [mΩ]	R_s	325
Mechanical inertia [kg · m^2]	J	0.0024
Equivalent PM current [A]	i_mag	-43.2

4.3 Inverter Model

4.3.1 Topology

To supply voltage to the stator windings, a two-level voltage source inverter (2L-VSI) is used. The topology of this type of inverter is presented in Figure 4.4. A simplified model is used, considering perfect switches and no dead time.

In [12] the working principle of a 2L-VSI is explained. The inverter consists of six power electronic switches, which are steered two by two in a complementary way. Each pair of complementary switches constitutes one phase leg, determining whether the load in that phase is connected to the positive or the negative DC bus. When, for instance, in phase a $S_a = 1$, this means that switch S_a conducts current and hence phase a of the load is connected to the positive DC bus. Due to complementarity, $\overline{S}_a = 0$ in this case, so switch $\overline{S}_a$ is not conducting. However, when $S_a = 0$, switch S_a is not conducting, while $\overline{S}_a$ is, resulting in the connection of the load in

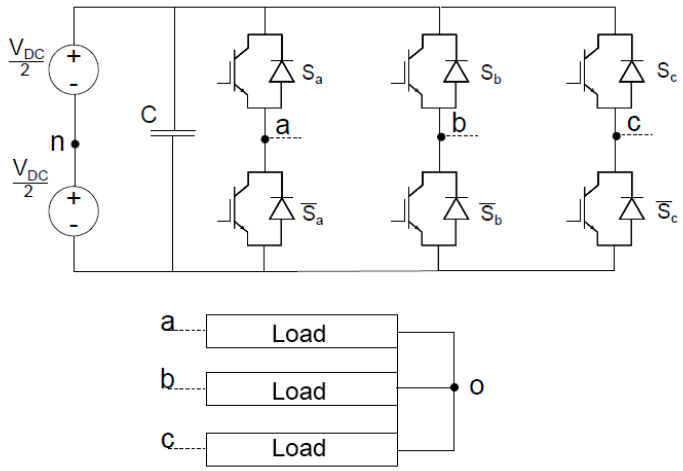


Figure 4.4: Topology of a 2L-VSI [7]

phase a to the negative DC bus. In a similar way, the other two phases have two switching possibilities as well, resulting in $2^3 = 8$ possible output states for the three phase voltage at the output of the inverter. These switching possibilities can be expressed by a vector $[S_a, S_b, S_c]$. In reality, some dead time is included between the switching of complementary switches, to assure that the previously conducting switch is extinguished before its complementary switch is fired. Without this precautionary measure, both switches would be conducting concurrently, resulting in a short-circuit of the DC bus. This so-called shoot-through would lead to unacceptably high currents, and may result in damage to the voltage source and the switches of the inverter.

For a known switch state vector, the phase voltages can be calculated as:

$$v_{x\mathrm{n}} = \left(S_x - \frac{1}{2}\right) V_{\mathrm{dc}} \tag{4.19}$$

where x represents the phase: $x \in \{a, b, c\}$. The eight possible phase voltages are represented in the $\alpha\beta$-reference frame in Figure 4.5. As can be seen, two of these voltages coincide with the origin, as a result of connecting all phases to the positive or the negative DC bus. These vectors are called null vectors, while the remaining six vectors are called active vectors.

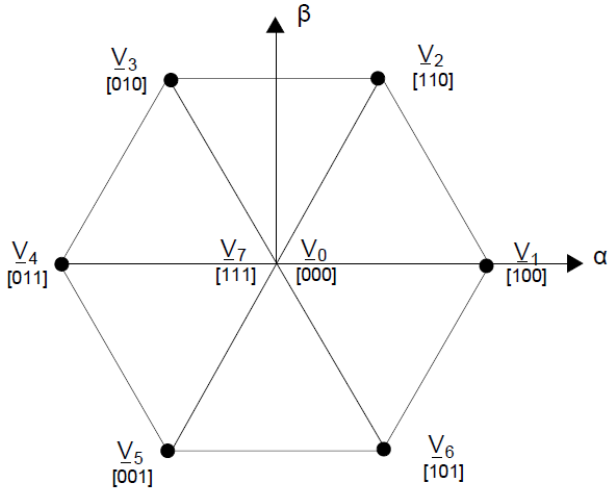


Figure 4.5: Phase voltage vectors of a 2L-VSI [7]

For the load, the voltage v_{xo} (where o represents the star point of the load) is of more importance than the phase voltage v_{xn} (where n refers to the center tap of the DC bus). The voltage over the load can be calculated as follows:

$$v_{xo} = v_{xn} - v_{on} \tag{4.20}$$

where the voltage of the star point o of the load with respect to the center tap n of the DC bus equals the average of the phase voltages if the load is symmetrical:

$$v_{on} = \frac{v_{an} + v_{bn} + v_{cn}}{3} \tag{4.21}$$

4.3.2 Modulation

When an inverter is used to convert a DC voltage to an AC voltage, the switch states must be alternated in a suitable way. Determining the sequence of the switch states defines the essence of inverter modulation. Three possible techniques will be presented: pulse width modulation (PWM), space vector modulation (SVM) and direct control, which are discussed in detail in [2] and [13].

Pulse Width Modulation

Pulse width modulation offers a technique to adjust both the frequency and the amplitude of the output voltage to a certain extent. The working principle is sketched in Figure 4.6. By using the intersections of an (in most cases) triangular carrier wave and a reference wave with the desired amplitude and frequency, the required pulse widths are obtained.

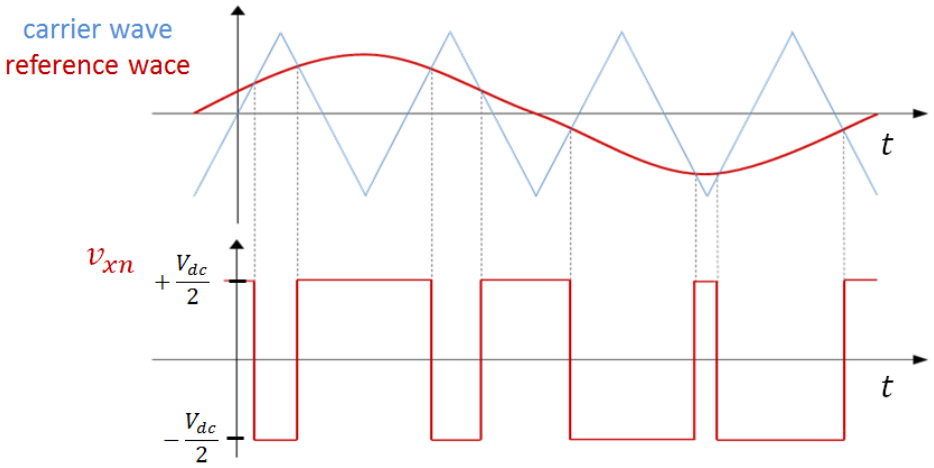


Figure 4.6: Working principle of PWM

Space Vector Modulation

The principle of space vector modulation can be explained based on the space vector representation of the output voltage. Therefore the $\alpha\beta$-plane needs to be subdivided in six sectors, as shown in Figure 4.7. An arbitrary voltage vector $\underline{v}_s$ is approximated by an appropriate sequence

of two null vectors and the two active vectors confining the sector where $\underline{v}_s$ is situated, in particular $\underline{V}_1$ and $\underline{V}_2$ in Figure 4.7. To obtain a voltage vector which equals on average the desired voltage vector $\underline{v}_s$ over a period T_c, $\underline{V}_1$ is applied over a period t_1 and $\underline{V}_2$ over a period t_2. During the rest of the period T_c - more specifically the period t_0 - a null vector will be applied, such that:

$$T_c \underline{v}_s = t_1 \underline{V}_1 + t_2 \underline{V}_2 \tag{4.22}$$

$$T_c = t_0 + t_1 + t_2 \tag{4.23}$$

In this way, the on- and off-times of the switches can be computed. By the choice of the switching pattern and the division of t_0 over the two null vectors, the modulation is fully defined. According to [2] for each type of SVM an equivalent PWM scheme can be found.

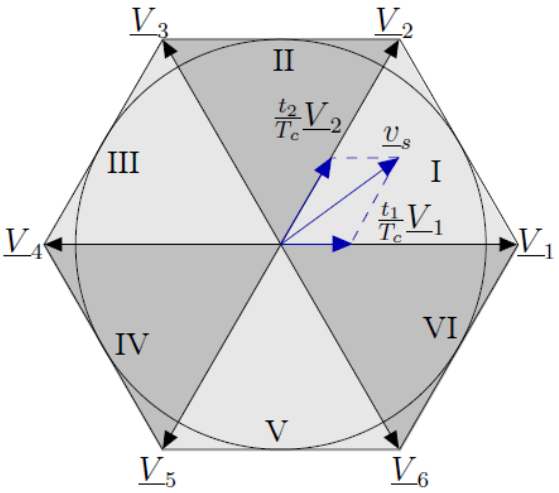


Figure 4.7: Working principle of SVM [2]

Direct Control

From the previous discussion about PWM and SVM it is clear that these modulation techniques are used to follow a voltage reference with a certain frequency in open-loop control: they make use of an a priori mapped out switching scheme. For MBPC, however, which was introduced in Section 3.2, this is not the case. In MBPC the feedback of one or more physical signals is processed by a controller, and based on the instantaneous state the optimal switching possibility is selected on-line.

4.4 Conclusion

In this chapter, the dynamical model of the AFPMSM was introduced. In order to obtain time-invariant stator equations the equivalent two-phase symmetrical model and the rotation to the qd-reference frame were discussed.

The model of the 2L-VSI was presented as well, together with three possible modulation techniques: PWM, SVM and direct control using MBPC. It was concluded that PWM and SVM make use of an a priori defined switching scheme, while MBPC uses feedback for on-line selection of the optimal voltage vector.

Chapter 5

Predictive Torque Control

5.1 Introduction

Predictive torque controllers (PTC) are part of the family of direct torque controllers (DTC), which regulate the torque and stator flux of electric machines in a direct way. The DTC family comprises various control strategies, going from the classical DTC schemes - introduced by Takahashi in the mid-1980's using look-up tables and hysteresis comparators - to control strategies making use of MBPC. In this master's dissertation only the PTC strategy based on FS-MBPC will be elaborated. In [2, 14, 15] a more complete overview of the DTC family is given.

This chapter will illustrate the three basic steps of FS-MBPC for PTC, i.e. estimation, prediction and optimization. The performance of the controller will be evaluated by simulations. An experimental verification can be found in Chapter 8.

5.2 Working Principle

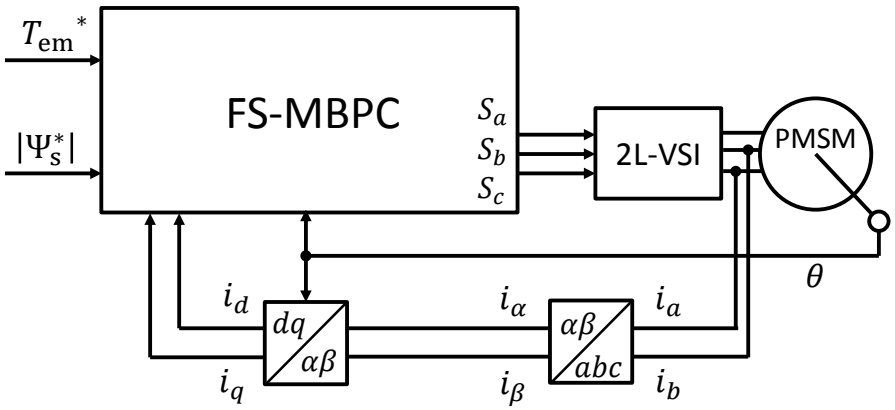


Figure 5.1: Principle of PTC

A predictive torque controller directly regulates both the electromagnetic torque and the modulus of the stator flux. The general structure is visualized in Figure 5.1. The algorithm uses as inputs the measured stator current and motor position, and the set-points for the controlled

variables, more specifically the electromagnetic torque T_{em} and the modulus of the stator flux $|\Psi_s|$. Those inputs are processed to determine the optimal switch state for the inverter by means of FS-MBPC. The following sections will delve deeper into the three basic steps of FS-MBPC, which were introduced in Section 3.2.

5.2.1 Estimation

At update instant k a new switch state S_k is applied to the inverter, resulting in a change in the stator voltage and thus the stator current of the AFPMSM. The new value of the current is measured half an update period after the control action - thus at $k + 1/2$ - in order to avoid measuring a transient. Of course the currents keep on evolving during the remaining part of the update period as well. Because a finite time is required to compute the optimal S_{k+1}, and the predictions in the following step are based on the currents and voltages at time instant $k + 1$, these values need to be estimated. This is done by means of the current switch state S_k of the inverter, the measured currents and the discrete time system model.

Stator voltage

In a first step, the phase voltages $v_{x\text{o}}^k$ over the AFPMSM ($x \in \{a, b, c\}$) are computed by means of the inverter model discussed in Section 4.3.1. The Clarke transformation is applied, followed by a rotation to bring the stator voltages to the synchronous qd-reference frame, as explained in Section 4.2.2. Since the switch state S_k is only changed again at instant $k + 1$, the voltages are constant during the interval $k \to k + 1$.

Stator current

In a second step, the stator current components at time instant $k + 1$ are estimated using the current measurement at instant $k + 1/2$ and the previously computed stator phase voltages. The current measurements are first transformed to the qd-reference frame, and then the discrete time version of the electrical equations derived in Section 4.2.2 is used to estimate the current components at instant $k + 1$:

$$\hat{i}_q^{k+1} = i_q^{k+1/2} + \frac{T_{\text{u}}}{2L_q}\left(v_q^k - R_{\text{s}} i_q^{k+1/2} + \omega^{k+1/2} L_d (i_d^{k+1/2} + i_{\text{mag}})\right) \tag{5.1}$$

$$\hat{i}_d^{k+1} = i_d^{k+1/2} + \frac{T_{\text{u}}}{2L_d}\left(v_d^k - R_{\text{s}} i_d^{k+1/2} - \omega^{k+1/2} L_q i_q^{k+1/2}\right) \tag{5.2}$$

The electrical speed $\omega^{k+1/2}$ equals the measured mechanical speed $\Omega^{k+1/2}$ multiplied with the number of pole pairs N_p, and is assumed to be constant during the time interval $k+1/2 \to k+2$. This assumption can be justified since T_{u} is small compared to the speed. Besides, the mechanical time constants of machines are typically larger than the update period. For the same reasons, the rotor angle θ is considered constant during T_{u}.

Stator flux

Contrary to the stator currents, the stator flux cannot be measured directly. It needs to be estimated based on other measured variables. An overview of different techniques that can be used for this purpose is given in [2]. Since the permanent magnets determine the position of

the rotor flux for a PMSM, it is appropriate to calculate the stator flux by means of the current model in the qd-reference frame:

$$\left.\begin{aligned}\hat{\Psi}_q^{k+1} &= L_q \hat{i}_q^{k+1} \\ \hat{\Psi}_d^{k+1} &= L_d(\hat{i}_d^{k+1} + i_{\text{mag}})\end{aligned}\right\} \Rightarrow |\hat{\Psi}_s^{k+1}| = \sqrt{\left(\hat{\Psi}_q^{k+1}\right)^2 + \left(\hat{\Psi}_d^{k+1}\right)^2} \tag{5.3}$$

A disadvantage of this technique is that it strongly depends on the stator inductances, which are susceptible to saturation of the machine. These deviations in inductance will influence the flux estimation as well.

However, an advantage of this method is that is does not make use of the stator resistance, significantly varying due to skin effect and temperature changes. Besides, it ensures an accurate estimation over the full speed range. This would not be the case if the flux were estimated by integration of the back EMF using a low pass filter. For an induction machine, the latter technique benefits from the fact that the flux and the torque can be estimated in the stator reference frame, without knowledge of θ. Nevertheless, in a PMSM the flux originating from the permanent magnets is defined in the qd-reference frame anyway, undermining this advantage.

Electromagnetic torque

The stator current components can be used to estimate the torque as well:

$$\hat{T}_{\text{em}}^{k+1} = -\frac{3}{2} N_p L_d i_{\text{mag}} \hat{i}_q^{k+1} \tag{5.4}$$

5.2.2 Prediction

For the prediction of the outputs for all eight switching possibilities S_{k+1}, the same computations are executed as in the estimation step. The only difference is that the time step used for the prediction of the current components equals T_{u} now instead of $T_{\text{u}}/2$.

5.2.3 Optimization

To decide which switching possibility results in the best control of both electromagnetic torque and stator flux, a cost function is used. For each controlled variable a cost is determined, representing how well the value approximates its corresponding reference. The cost can be defined in various ways. A common choice is to express it as the square of the deviation of the variable from its set-point. In this way, large deviations are punished more severely than small deviations. To facilitate putting variables of different orders of magnitude in one function, the use of p.u. values is recommended. The total cost J^{k+1} corresponding to the application of a certain switch state S_{k+1} is then presented by:

$$J^{k+1} = \frac{(\hat{T}_{\text{em}}^{k+2} - T_{\text{em}}^*)^2}{T_{\text{ref}}^2} + W_\Psi \frac{(|\hat{\Psi}_{\text{s}}^{k+2}| - |\Psi_{\text{s}}^*|)^2}{\Psi_{\text{ref}}^2} \tag{5.5}$$

where W_Ψ is a dimensionless weighting factor representing the relative strictness of the flux control compared to the torque control, and T_{ref} and Ψ_{ref} are used to transform the two cost terms into p.u. values. T_{em}^* and $|\Psi_{\text{s}}^*|$ represent the required torque and flux respectively. In this text, T_{ref} usually equals the required torque T_{em}^*, unless zero torque is required from the machine. In that case, $T_{\text{em}}^* = 0$ Nm, while T_{ref} is chosen to be 1 Nm. The same holds for $|\Psi_{\text{s}}^*|$ and Ψ_{ref}. In this way, a deviation of the same size will lead to a higher cost for small reference values than for large reference values.

Another possibility is to use the absolute value of the errors in the cost function instead of quadratic terms, as is done in [9] and [16]. However, in [4] is demonstrated that this leads to a comparable control quality.

The use of weighting factors is avoided in [17], since there is no systematic way to define them, and their appropriateness depends on the system parameters and operating point. The authors propose an alternative strategy in which the input voltage vectors are ranked for each separate variable, according to the cost for that variable. Voltage vectors resulting in a lower error are assigned a lower ranking, while voltage vectors leading to higher errors receive a higher ranking. The vector that has the lowest average value of its ranking is selected as the best overall voltage and applied to the system. However, since this strategy requires more computational effort and its advantages are only pronounced for controllers dealing with plenty of variables, in this master's dissertation weighting factors will be used. In Section 5.3, more attention will be devoted to the tuning of the weighting factors.

The reference value T_{em}^* for the torque is prescribed by the application. The flux, however, is controlled to respect the linearity assumption of the used model and to constrain the Joule losses. When the stator flux rises above a certain level (as a rule of thumb, this level equals 1.6T), saturation of the machine occurs. As a result, the accuracy of the linear model decreases, which might jeopardize the torque control. Reduction of the Joule losses is another motivation to regulate the flux. Since for PMSMs the rated flux is provided by the permanent magnets, the effect of the stator current on the flux needs to be limited (except when field weakening is required). For SPMSM the torque by ampere ratio is maximized by fixing the current i_d to zero. This is called the principle of maximum torque per ampere (MTPA) in [18][1]. Reduction of the stator current amplitude and control of the flux level can be obtained by defining the reference value for the modulus of the flux as:

$$|\Psi_{\text{s}}^*| = \sqrt{(L_q i_q^*)^2 + (L_d i_{\text{mag}})^2} \tag{5.6}$$

where the reference value i_q^* is implicitly imposed by the reference value for the torque:

$$i_q^* = \frac{T_{\text{em}}^*}{-\frac{3}{2} N_p L_d i_{\text{mag}}} \tag{5.7}$$

Stator flux and torque are clearly correlated by the stator current component i_q, making it difficult to understand the actions of the controller. The fact that the control of the two regulated variables cannot be decoupled, is a disadvantage of PTC. Furthermore, Eq. (5.6) is actually a very circuitous and artificial way to express that $i_d^* = 0$. This topic will feature in Section 6.2.

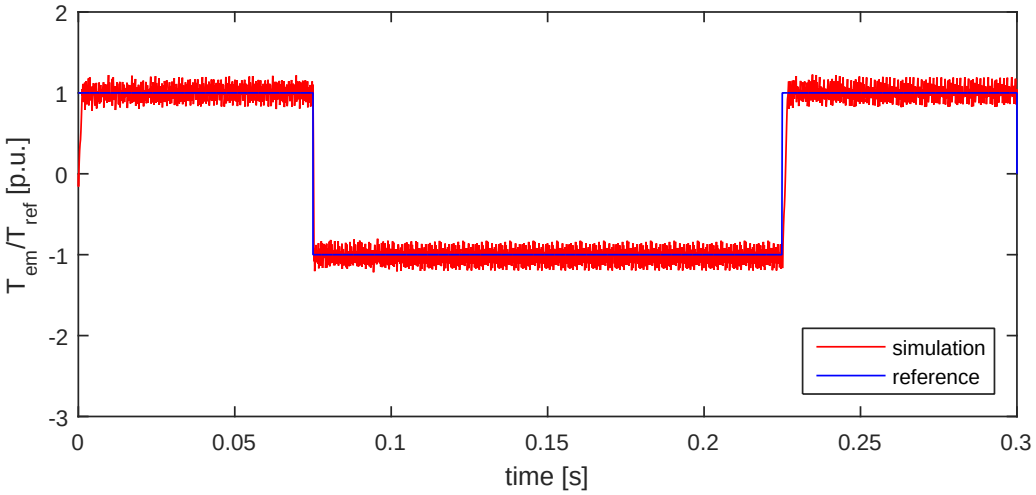
5.3 Simulations

The simulations in this master's dissertation are performed in a MATLAB®&Simulink® environment. Figure 5.2 shows the simulation results for the quadratic cost function of Eq. (5.5) with a weighting factor $W_\Psi = 3$. The values for the other simulation parameters are listed in Table 5.1. Those values will be used for all the simulations in this and the next chapter, unless stated otherwise.

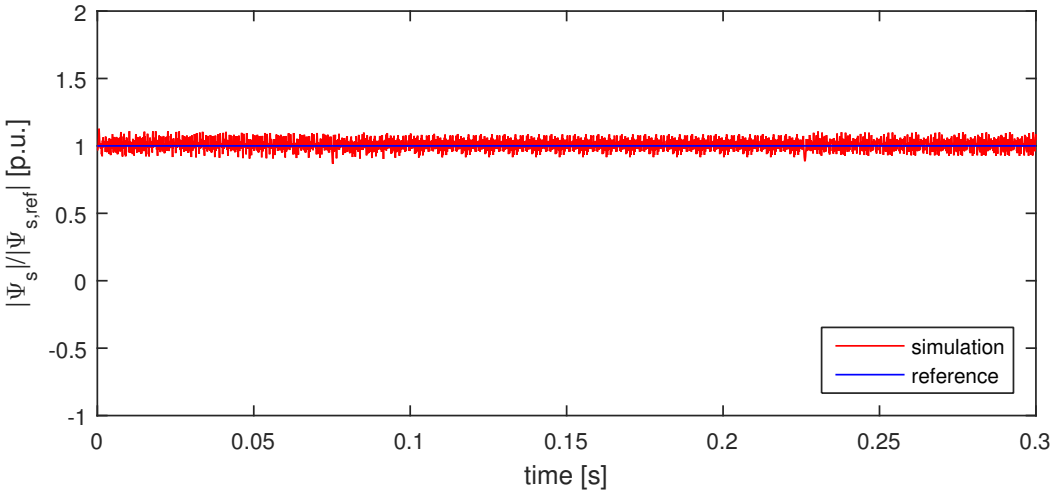
[1] The MTPA principle implies only for SPMSMs with $L_q = L_d$ that i_d needs to be zero. For machines where $L_q \neq L_d$, the MTPA principle requires a non-zero i_d.

Table 5.1: Standard simulation parameters

Parameter	Value
T_{ref} [Nm]	15
$I_{q,\text{ref}} = T_{\text{ref}}/[-(3/2)N_p L_d i_{\text{mag}}]$ [A]	11.39
$\Psi_{\text{ref}} = \sqrt{(L_d i_{\text{mag}})^2 + (L_q I_{q,\text{ref}})^2}$ [Wb]	0.11
V_{dc} [V]	200
$f_{\text{u}} = f_s$ [kHz]	10
N [rpm]	1000



(a) Electromagnetic torque



(b) Stator flux

Figure 5.2: PTC with quadratic cost function ($W_\Psi = 3$)

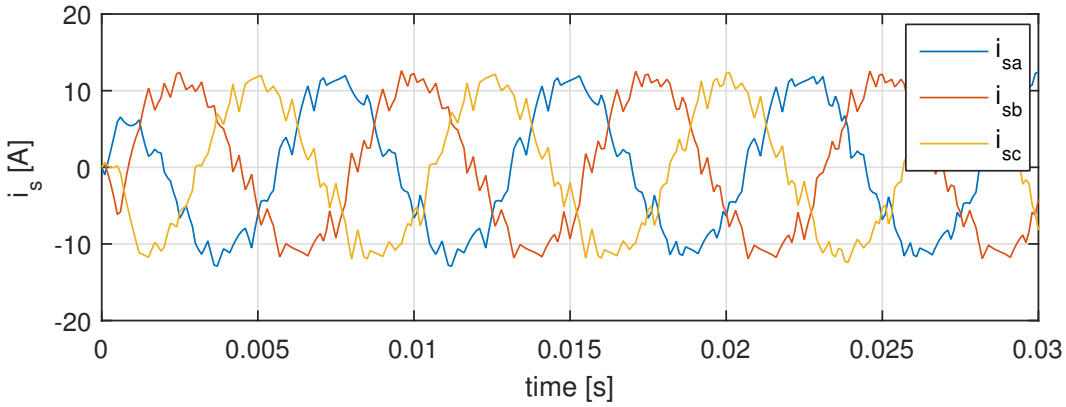


Figure 5.3: Stator currents for PTC with quadratic cost function ($W_\Psi = 3$)

It is clear that the torque and flux follow their reference values well. Although no current control is included in the control algorithm, the stator currents plotted in Figure 5.3 appear to be highly sinusoidal. The ripple due to the inverter voltage supply is however clearly observable.

Nevertheless, the quadratic cost function given by Equation (5.5) results in a typical tracking control, in which the controller tries to bring both torque and flux exactly to their reference values. But a voltage vector realizing the set-points for both controlled values will exist only very rarely. As a result, the controller will continuously adapt the input voltage in vain, trying to bring the state to its reference value. Actually, this is not the behavior that is wished for. A direct control aims at reaching an acceptable instantaneous deviation instead of trying to reach a minimal steady-state deviation. Therefore the performance of a different type of cost function will be tested as well, a cost function that does not work with tracking but with hard and soft limits. Hard limits cannot be exceeded (which can prove useful to determine safety limits, for instance for current limitation), soft limits can, but at a high cost.

An adaptation of the cost function that significantly influences the normal working of the machine when using PTC, and represents the hard and soft limits, is the definition of tolerance bands for the torque and/or the modulus of the stator flux. Deviations from the set-point that fall within these tolerance bands are not - or less severely - punished by a cost. The form of the new cost terms is sketched in Figure 5.4.

$$J_T^{k+1} = \begin{cases} 0, & 0 \leq |\Delta T_{\text{em}}| \leq T_{\text{em,l}} \\ (\Delta T_{\text{em}})^2, & T_{\text{em,l}} \leq |\Delta T_{\text{em}}| \leq T_{\text{em,u}} \\ W_{T\infty}(\Delta T_{\text{em}})^2, & |\Delta T_{\text{em}}| > T_{\text{em,u}} \end{cases} \tag{5.8}$$

$$J_\Psi^{k+1} = \begin{cases} 0, & 0 \leq |\Delta|\Psi_{\text{s}}|| \leq |\Psi_{\text{s}}|_{\text{l}} \\ W_\Psi(\Delta|\Psi_{\text{s}}|)^2, & |\Psi_{\text{s}}|_{\text{l}} \leq |\Delta|\Psi_{\text{s}}|| \leq |\Psi_{\text{s}}|_{\text{u}} \\ W_{\Psi\infty}(\Delta|\Psi_{\text{s}}|)^2, & |\Delta|\Psi_{\text{s}}|| > |\Psi_{\text{s}}|_{\text{u}} \end{cases} \tag{5.9}$$

$$J^{k+1} = J_T^{k+1} + J_\Psi^{k+1} \tag{5.10}$$

where Δ represents the dimensionless deviation from the reference value, W_Ψ is again the weighting factor representing the relative strictness of torque and flux control, X_{l} and X_{u} are respectively the boundaries of the lower and upper tolerance band, and $W_{X\infty}$ is a relatively large number representing the fact that the region $|\Delta X| > X_{\text{u}}$ must be avoided whenever possible. Choosing $W_{X\infty} = \infty$ is the same as expressing that X_{u} is a hard limit. In this text, $W_{T\infty}$ and $W_{\Psi\infty}$ are chosen to be finite numbers in order to avoid infeasibilities, situations in which the controller cannot find an appropriate switch state since all voltage vectors from the finite set lead to an infinite cost.

The tolerance bands can be compared with the flux and torque hysteresis bands for DTC. Their magnitude will greatly influence the inverter's switching frequency, as has been investigated for DTC in [19]. The authors of [20] have even gone one step further in the reduction of the switching frequency than will be done in this text: for each feasible switch state they extrapolate its output trajectory to derive the number of time steps after which at least one of the controlled variables leaves its tolerance band. In this way, priority can be given to the switch state that yields the highest number of time steps that can be applied before switching is required again.

An extensive study about the design of tolerance bands in a cost function is offered in [21]. One of the conclusions is that relaxing the flux control (which is the same as using a wider flux tolerance band) results in better control quality of the torque. However, $|\Psi_\mathrm{s}|_\mathrm{l}$ cannot be taken infinitely large, since the stator current is not minimized in this way: Joule losses would increase due to a stator current component that does not contribute to the torque, nor is useful for the flux. Thus, a compromise has to be made between the switching losses and the torque control quality on the one hand, and the Joule losses and the flux control quality on the other hand.

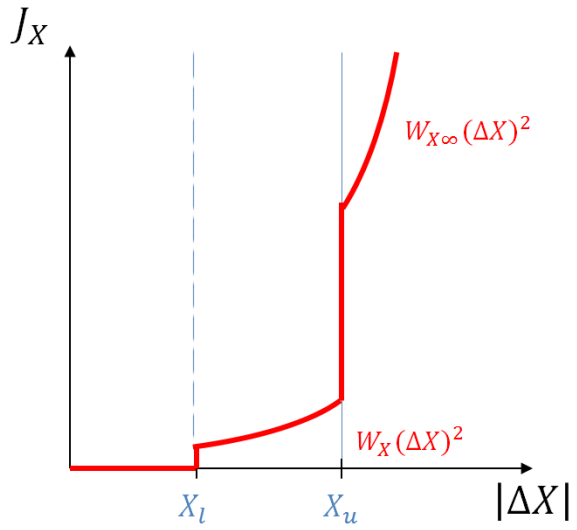
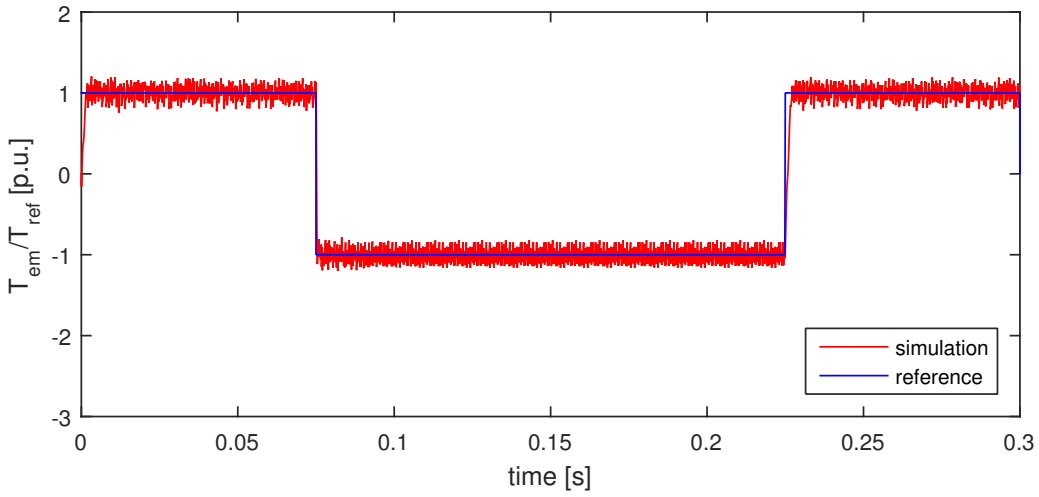


Figure 5.4: Cost function with tolerance bands

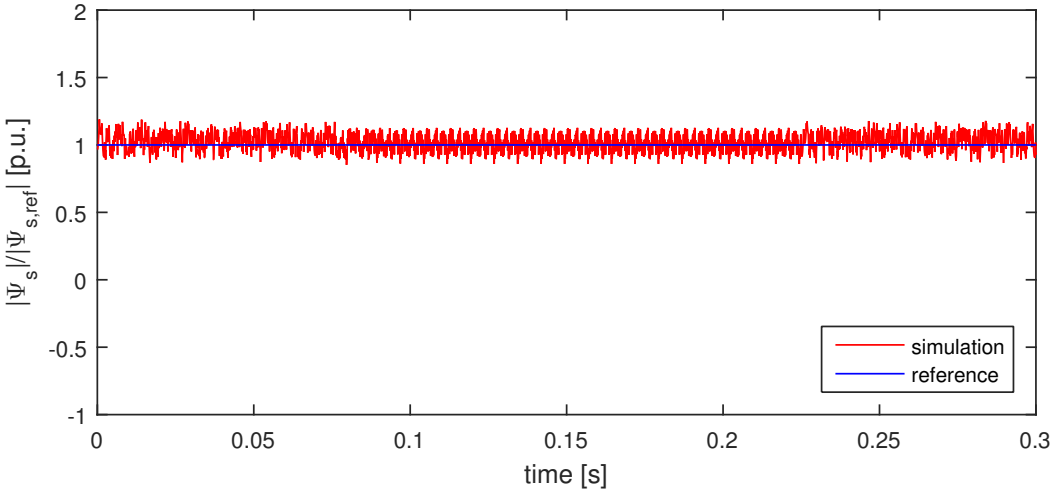
Figure 5.5 plots the simulation results for the cost function given by Eq. (5.10) with $T_{\mathrm{em,l}} = 0.15$ p.u., $T_{\mathrm{em,u}} = 0.25$ p.u., $W_{T\infty} = 10$, $W_\Psi = 1$, $W_{\Psi\infty} = 100$, $|\Psi_\mathrm{s}|_\mathrm{l} = 0.12$ p.u. and $|\Psi_\mathrm{s}|_\mathrm{u} = 0.2$ p.u. How these parameters are derived, is explained in Section 5.5.

At first sight, the introduction of the tolerance bands increased the ripple on the stator flux remarkably, while the torque ripple remained more or less the same. The detail views of Figures 5.6 and 5.7 confirm these assumptions. The switching frequency seems to have lowered by the tolerance band. In Figures 5.6b and 5.7b can be noticed that torque and stator flux remain within their inner tolerance band for most of the time, while they never leave their outer tolerance band. This is exactly the kind of behavior cost function (5.10) aims for.

However, comparing two controllers based on a visual inspection of simulation results only, does not result in well-founded conclusions. In order to assess simulation results of different controllers in a proper way, an objective quantification of the control quality is urgently needed. The following section addresses this topic.

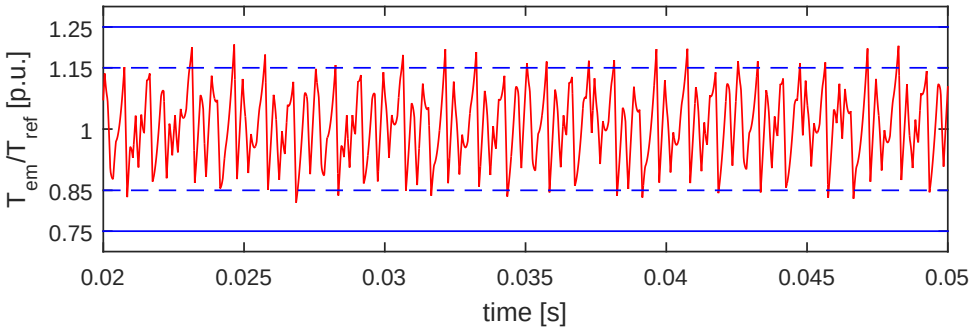


(a) Electromagnetic torque

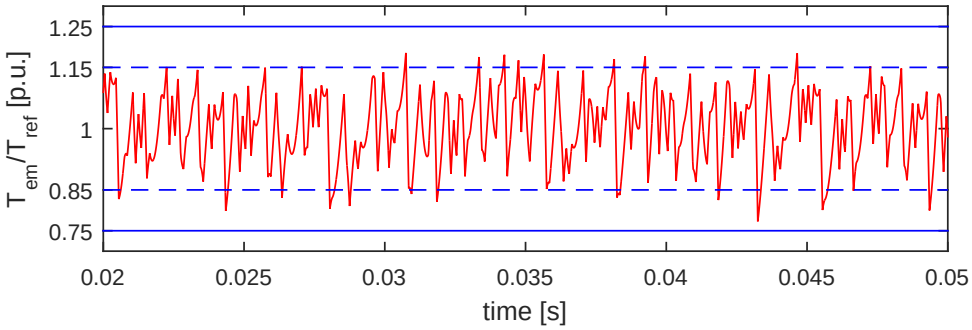


(b) Stator flux

Figure 5.5: PTC with cost function with tolerance bands for both torque and flux ($W_\Psi = 1$)

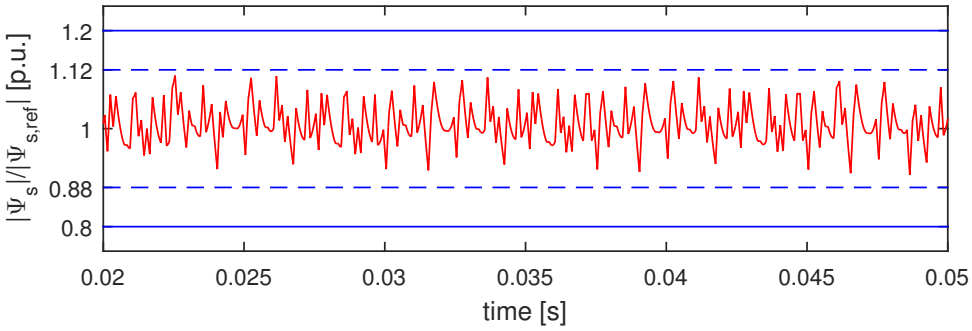


(a) Quadratic cost function

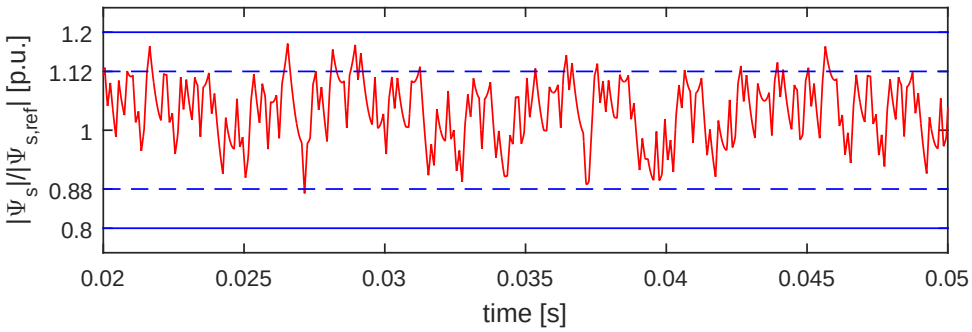


(b) Cost function with tolerance bands

Figure 5.6: Detail views showing the effect of tolerance bands on the electromagnetic torque



(a) Quadratic cost function



(b) Cost function with tolerance bands

Figure 5.7: Detail views showing the effect of tolerance bands on the stator flux

5.4 Objective Quantification of Quality by Means of Key Performance Indicators

5.4.1 Control Quality

To compare different controllers in an objective way, [2] suggests to use the mean square error (MSE) of the controlled variables. The equation for the MSE of the variable x compared to its reference value x^* is given in Equation 5.11. m is the amount of samples used in the calculation.

$$\mathrm{MSE}(x) = \frac{\sum_{k=1}^{m} (x_k^* - x_k)^2}{m} \tag{5.11}$$

To express the global control quality, the (weighted) sum of the MSE-values for all controlled variables is frequently used. Therefore p.u. values are recommended in the computation of the total MSE. The used reference values are given in Table 5.1. Actually, the MSE can be compared to a quadratic cost function: the smaller the MSE, the better the control quality.

An alternative for the MSE is the mean absolute error (MAE):

$$\mathrm{MAE}(x) = \frac{\sum_{k=1}^{m} |x_k^* - x_k|}{m} \tag{5.12}$$

The main difference between the MSE and MAE method is that MSE values show larger variation than MAE values for the same deviations: in the MSE method large deviations from the reference value have more weight than small deviations. This is not the case for MAE values. As a result, a logarithmic scale is more appropriate for MSEs, while a linear scale is more appropriate when using MAEs. An argument in the advantage of the MAE is that those values can be interpreted more intuitively, especially when they are expressed in %. However, since larger deviations should be penalized more in a good control, it is opted to mainly use the MSE in the continuation of this text.

5.4.2 Average Switching Frequency and Voltage Quality

In the assessment of a certain control strategy, not only the quality of the control itself should be taken into account. The switching losses and the voltage quality are important aspects of the global quality as well.

Switching is inherently coupled with switching losses. Therefore the switching frequency is aimed to be as low as possible. In this text the switching frequency f_{switch} is defined as the number of times a switch changes its state from on to off or vice-versa, averaged over a certain period of time and the three pairs of complementary switches.

Another important quality aspect is the voltage quality. In [2] the importance of the voltage quality is emphasized. Four aspects are discussed:

- *copper losses:* Due to the current ripple caused by a low-quality voltage waveform the RMS-value of the current, and consequently the copper losses in the stator windings, increase.
- *iron losses:* Due to voltage pulses, additional eddy currents originate in the iron core, resulting in extra iron losses.
- *motor isolation:* An electrical machine is fed by an inverter through a cable. When this cable is long, transmission line effects become important and may cause reflections at the

machine plugs, resulting in overvoltage. The magnitude of the overvoltage depends on the length of the cable and the voltage transients: the larger both, the larger the overvoltage.

- *axis and bearing currents:* Due to the effect of fast switching and parasitic capacitive coupling, high-frequent bearing current can arise, lowering the service life of the bearings.

In general, voltage quality assessment compares a voltage waveform with its reference voltage waveform. However, since closed-loop torque control does not provide such a reference voltage waveform, these techniques are not possible. In [2, 22] the extended pulse polarity consistency rule (PPCR) is recommended as an alternative for this purpose, stating that a pulse in the coupled voltage can not be followed by a pulse of opposite polarity. After all, polarity reversal heavily burdens the machine isolation. Concretely this means that for a 2L-VSI the only allowed switch state transitions are the ones between neighboring voltage vectors in three-phase space vector representation (also called the $\alpha\beta$-representation). Figure 5.8 illustrates this requirement for a 2L-VSI that needs to switch from state [1 1 0]. In this situation only four voltage vectors can be chosen, under which one redundant null vector. For the null vector both [1 1 1] as [0 0 0] can be chosen, but to reduce the switching losses [1 1 1] is the best option.

Although PPCR is a qualitative benchmark, it can be used quantitatively by counting how many times the PPCR is not fulfilled for the switchings during a certain period of time.

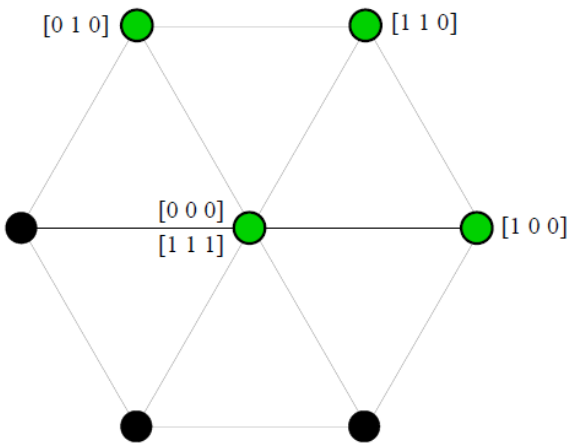


Figure 5.8: The four allowable voltage vectors for a three-phase 2L-VSI according to the PPCR when the current switch state is [1 1 0] [2]

5.5 Cost Function Design

When using FS-MBPC, one of the hardest tasks is the definition of an appropriate cost function. The importance of the cost function cannot be underestimated: when it does not reflect the desired behavior well, the controller will never be able to reach its expected performance. The key performance indicators (KPIs) introduced in the previous section prove very useful in this context. Not only can they be used to evaluate different types of cost functions, they can be of great help in choosing well-suited parameters (like for instance the weighting factors and tolerance bands in Equations (5.8) and (5.9)) as well.

To compare the simulation results of Section 5.3 for example, their KPIs are summarized in Table 5.2. To compute MSE(i_d) and MAE(i_d), i_d^* has been assumed zero, since this leads to the lowest Joule losses. Due to the fact that i_q is just a rescaling of T_{em} (according to Eq. (5.4)), their MSE and MAE values are the same in p.u. As inferred by the simulation plots,

the introduction of the tolerance bands increased the deviation of the stator flux significantly, but within acceptable limits. The new cost function achieved its purpose: for a similar torque ripple, the switching frequency and the percentage of switchings not fulfilling the PPCR have been reduced, at the expense of an increased, but still acceptable Joule loss.

Table 5.2: Comparison in KPIs of a quadratic cost function and a cost function with tolerance bands

KPI	quadratic	tolerance band
MSE(T_{em}) = MSE(i_q) [p.u.]	0.0066	0.0064
MSE($\lvert\Psi_{\mathrm{s}}\rvert$) [p.u.]	0.0013	0.0050
MSE(i_d) [p.u.]	0.0232	0.0837
MAE(T_{em}) = MAE(i_q) [p.u.]	0.0668	0.0667
MAE($\lvert\Psi_{\mathrm{s}}\rvert$) [p.u.]	0.0285	0.0599
MAE(i_d) [p.u.]	0.1206	0.2451
Joule loss [W]	43.90	46.00
f_{switch} [kHz]	3.30	2.81
non-PPCR [%]	32.44	20.55

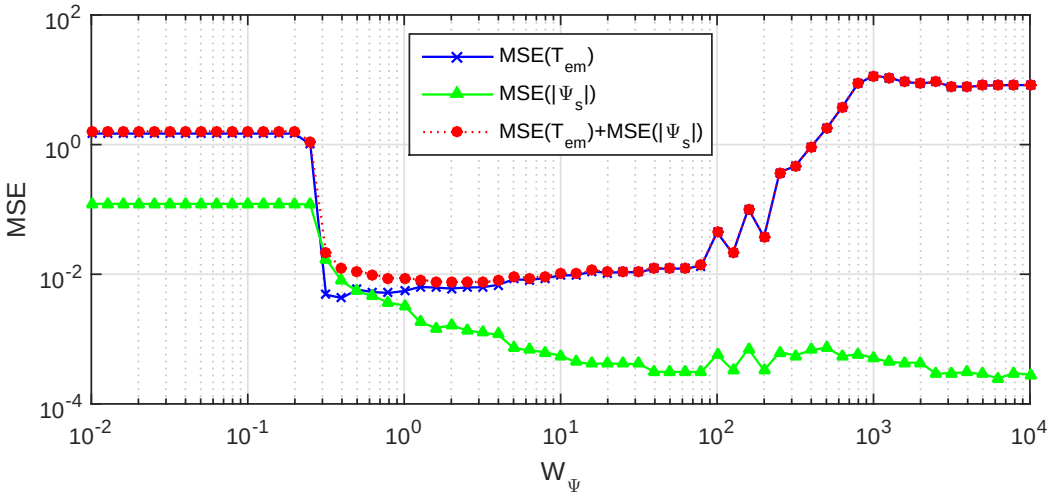


Figure 5.9: MSE for varying weighting factor W_Ψ in a quadratic cost function

The MSE values can be used to find a proper value for the weighting factor W_Ψ for the quadratic cost function as well. The plot in Figure 5.9 confirms that $W_\Psi = 3$ leads to good results. Remark that the minimum of MSE(T_{em}) + MSE($|\Psi_{\mathrm{s}}|$) is flat, so small deviations of the optimum do not strongly affect the MSE. Since the real system is never perfectly represented by the system parameters of the model, it is preferable to have a range of weighting factors leading to good results. In [23] is concluded that a similar range of appropriate weighting factors can be obtained by experiments.

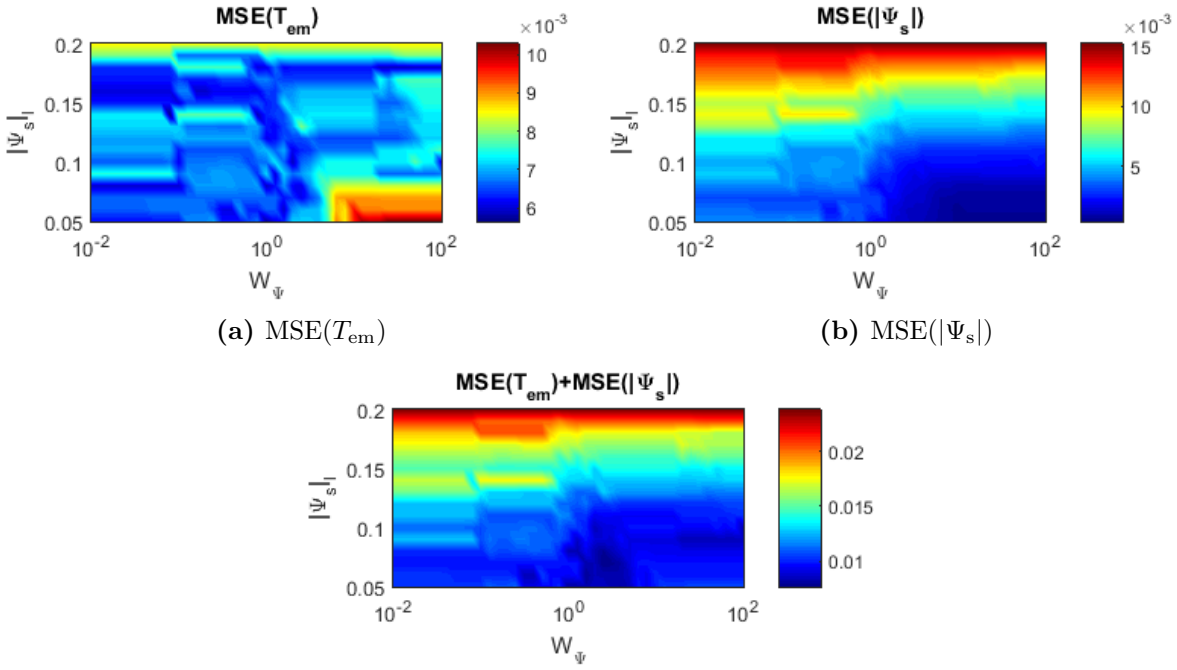
In general, MSE(T_{em}) decreases for smaller W_Ψ, since reducing W_Ψ means putting more emphasis on the torque control. However, when W_Ψ is very small, the controller loses track of the torque reference value. The cause of this phenomenon can be found in the electrical model of the machine. The derivatives of the current components are:

$$\frac{\mathrm{d}i_q}{\mathrm{d}t} = \frac{1}{L_q}\left[v_q - R_{\mathrm{s}} i_q + \omega L_d (i_d + i_{\mathrm{mag}})\right] \tag{5.13}$$

$$\frac{\mathrm{d}i_d}{\mathrm{d}t} = \frac{1}{L_d}\left[v_d - R_{\mathrm{s}} i_d - \omega L_q i_q\right] \tag{5.14}$$

If the extreme case $W_\Psi = 0$ is considered, the only requirement the control imposes on the system at start-up is that T_{em}, and thus i_q, must rise. Since for $N = 1000$ rpm the factor ωL_d of the cross-coupling term exceeds R_s by more than a factor 6, the increase of i_q is strongest when i_d is large and positive. However, when i_q rises, $\text{d}i_d/\text{d}t$ decreases until it becomes negative, causing i_d to reduce until this current component falls far below zero. Since v_q is limited by the DC bus voltage, eventually i_q will start to decline as well. The required torque can never be reached, which is reflected in a high MSE. The cross-coupling terms thus lead to a counterintuitive result: focusing too much on the control of the torque deteriorates the controller's performance instead of ameliorating it. The physical cause of this paradox is the back EMF of the AFPMSM. According to Faraday's law, a change in magnetic flux (due to a change in the stator current in this case) induces an electromotive force. Lenz' law states that this induced back EMF opposes the change that produced it, which is exactly the observed phenomenon.

When the speed N is lowered, the effect of the cross-coupling is weakened, enabling a controller with low W_Ψ to attain the required torque level as well. Another solution is to increase the DC bus voltage. This will cause i_q and i_d to have faster dynamics, but at the expense of a higher torque and flux ripple, and more losses in the inverter switches.



(a) MSE(T_{em})

(b) MSE($|\Psi_\text{s}|$)

(c) MSE(T_{em})+MSE($|\Psi_\text{s}|$)

Figure 5.10: Effect of the tolerance band and weighting factor on the MSE

Finding an appropriate W_Ψ for a controller with a cost function with tolerance bands is somewhat more complex, since the width of the tolerance bands comes into play as well. When $T_{\text{em,l}}$, $T_{\text{em,u}}$ and $|\Psi_\text{s}|_\text{u}$ are, for instance, imposed by control quality requirements, W_Ψ and $|\Psi_\text{s}|_\text{l}$ can be tuned in order to reduce the MSE values. Figure 5.10 shows how the MSE values vary in function of both parameters. The simulations were performed for $T_{\text{em,l}} = 0.15$ p.u., $T_{\text{em,u}} = 0.25$ p.u., $W_{T\infty} = 10$, $W_{\Psi\infty} = 100$ and $|\Psi_\text{s}|_\text{u} = 0.2$ p.u. to avoid too large deviations from the rated stator flux. In general an increase in W_Ψ and/or a decrease in $|\Psi_\text{s}|_\text{l}$ results in a lower MSE($|\Psi_\text{s}|$), thus a better control of the stator flux. MSE(T_{em}) on the other hand only increases significantly when at the same time W_Ψ is high and $|\Psi_\text{s}|_\text{l}$ is low, since the controller gives most of its attention to the regulation of $|\Psi_\text{s}|_\text{l}$ in this case. When W_Ψ is low or $|\Psi_\text{s}|_\text{l}$ is high, more importance can be attached to the control of T_{em}.

In contrast to the quadratic cost function, the controller with tolerance bands never loses his grip on the torque due to the fact that $W_{\Psi\infty} = 100$ for all values of W_Ψ. The outer tolerance band prevents that i_d falls too far below zero, keeping $\mathrm{d}i_q/\mathrm{d}t$ positive until T_{em}^* is reached. Nevertheless, in Figure 5.10a can be seen that the quality of the regulation of T_{em} has declined for $|\Psi_\mathrm{s}|_\mathrm{l} = 0.2$, as a result of a less strict control of the stator flux.

Figure 5.10a shows that for $W_\Psi = 1$ the torque is properly controlled for $|\Psi_\mathrm{s}|_\mathrm{l} \in [0.05, 0.2]$. In order to tune $|\Psi_\mathrm{s}|_\mathrm{l}$, the effect of this parameter on the other KPIs must be taken into account as well. Figure 5.11 shows the variation of the switching frequency in function of $|\Psi_\mathrm{s}|_\mathrm{l}$. It can be concluded that in general a larger bandwidth results in a lower switching frequency and thus lower switching losses. The effect of $|\Psi_\mathrm{s}|_\mathrm{l}$ on the percentage of switchings that is not conform the PPCR is illustrated in Figure 5.12. The general conclusion is again that the voltage quality improves for larger bandwidths.

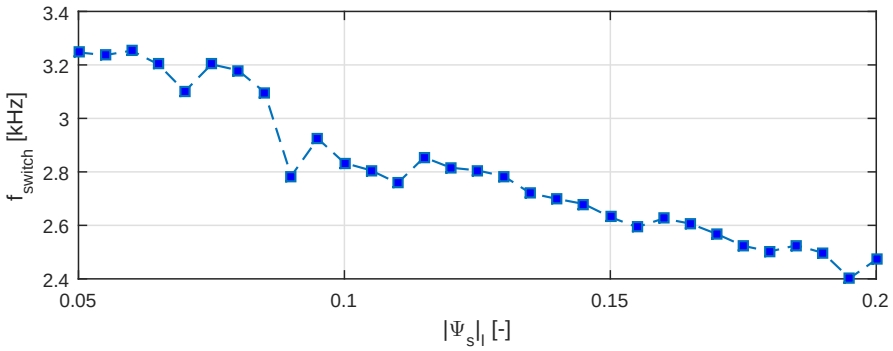


Figure 5.11: Effect of the tolerance band for the flux on the average switching frequency

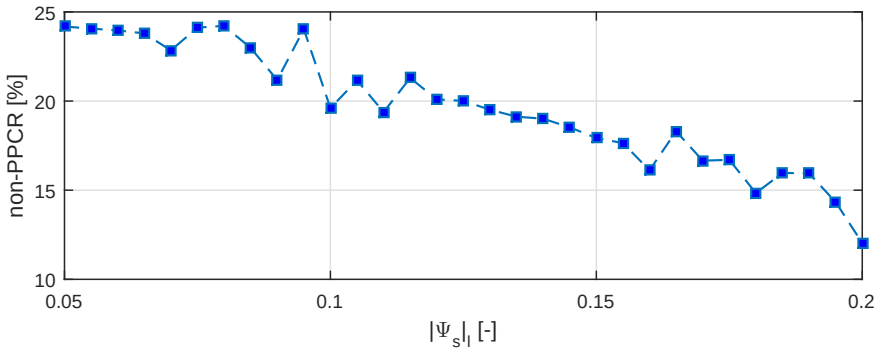


Figure 5.12: Effect of the tolerance band for the flux on the percentage of switchings violating the PPCR

To get an idea of the effect of the width of the tolerance band on the average Joule losses, they are plotted in function of $|\Psi_\mathrm{s}|_\mathrm{l}$ in Figure 5.13. The Joule losses increase with $|\Psi_\mathrm{s}|_\mathrm{l}$, because a large $|\Psi_\mathrm{s}|_\mathrm{l}$ means that the stator current is not minimized. Nevertheless, since R_s is rather small, the losses are still acceptable for $|\Psi_\mathrm{s}|_\mathrm{l} = 0.2$ as well. $|\Psi_\mathrm{s}|_\mathrm{l} = 0.12$ thus leads to a good compromise and the choice $|\Psi_\mathrm{s}|_\mathrm{u} = 0.2$ is also justified.

In [19] an important remark is made on the width of the tolerance bands. Since the current ripple varies with the operating point, the tolerance bands cannot be optimal for each of them. A lower speed, for instance, requires a lower DC bus voltage and consequently results in lower ripple. The tolerance band, however, is set to cope with the worst case scenario, so in the low speed region the system performance will be degraded. This disadvantage is not present when using a quadratic cost function.

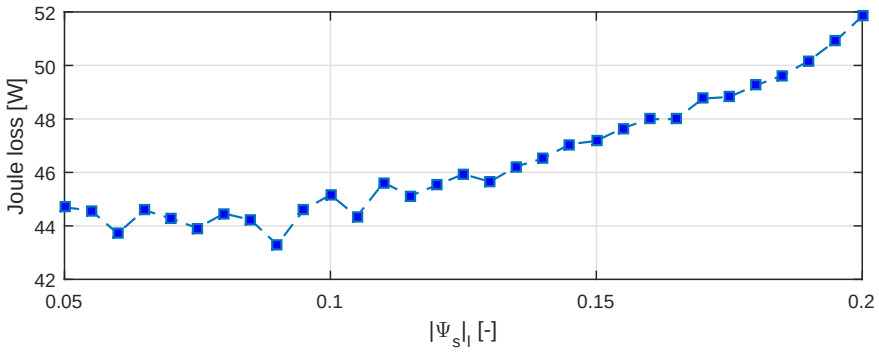


Figure 5.13: Effect of the tolerance band for the flux on the average Joule loss

5.6 Conclusion

In this chapter, a comprehensive discussion of PTC was given. PTC uses FS-MBPC to regulate torque and stator flux directly. The three basic steps of FS-MBPC, i.e. estimation, prediction and optimization, were elaborated in the specific context of PTC for an AFPMSM. An overview is given by the flowchart in Figure 5.14, where the state x represents the torque and stator flux.

Two different types of cost function were proposed for the optimization step: a quadratic cost function, resulting in a typical tracking control, and a cost function with tolerance bands, allowing the controlled variables to deviate a little from their reference values. To allow a fair comparison between those two types, some KPIs to assess the control quality were introduced:

- MSE and MAE, to evaluate how much the controlled variables deviate from their reference values
- f_{switch}, to give an indication of the switching losses
- the PPCR, to asses the voltage quality
- Joule losses

It could be concluded that a quadratic cost function leads to low MSE and MAE values, and low Joule losses, but at the expense of a high f_{switch} and violations of the PPCR. A tolerance band on the other hand increased the MSE and MAE for the flux, in favor of reduced switching losses and a higher voltage quality.

The correlation between stator flux and torque was mentioned as well. The fact that the two controlled variables are linked by the stator current component i_q makes it more difficult to predict the actions of the controller, which is a disadvantage of PTC.

Eventually, the effectiveness of the control strategy was validated by simulations.

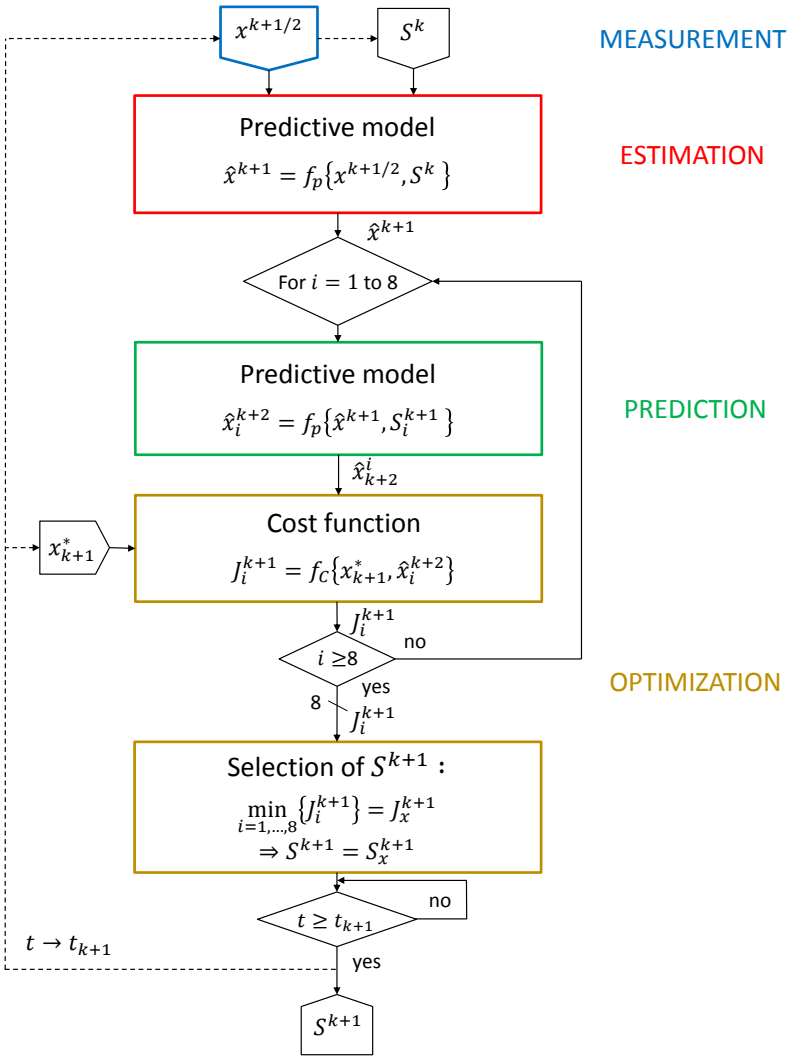


Figure 5.14: PTC algorithm (Adapted from [9])

Chapter 6

Field Oriented Control

6.1 Introduction

The DC commutator machine used to be the preferred machine for applications that required four-quadrant operation and a fast response. For constant excitation and ideal compensation, the torque of a DC machine is proportional to its current. By controlling the current, for instance with a DC chopper or controlled rectifier, the torque is controllable over the entire speed range.

Advances in power electronics have made it possible to obtain controllable AC current supply for AC rotating field machines as well. The combination with a control strategy called field orientation allows to obtain a similar ideal dynamic behavior as for the DC machine. Field oriented control (FOC), also known as vector control, has been proposed by Hasse and Blaschke. In [2,10,13] a comprehensive explanation of this control strategy is given. Its greatest advantage compared to PTC is that electromagnetic torque and rotor flux are decoupled, and Joule losses are reduced.

In this chapter, four different implementations of field oriented control will be discussed and simulated: PI control, FS-MBPC, deadbeat control and FS-MBPC with optimal duty cycle calculation. An experimental verification can be found in Chapter 8.

6.2 Working Principle

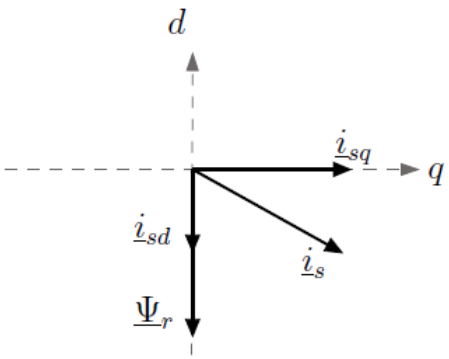


Figure 6.1: qd-reference frame attached to the rotor and with the negative d-axis along the rotor flux [2]

To apply field oriented control, the AFPMSM is described in the qd-reference frame, rotating in synchronism with the rotor flux vector $\underline{\Psi}_r$. The stator current is split into two orthogonal current

components with amplitude i_q and i_d, oriented along the quadrature and direct axis respectively, as illustrated in Figure 6.1. The new coordinates are called field coordinates. In Section 4.2.2, a model is derived using these orthogonal components. The component i_q is responsible for the torque, while i_d defines the flux. Both components are controlled separately.

In FOC for PMSMs, the rotor flux can be chosen to be completely defined by the permanent magnet flux $\underline{\Psi}_f$ (unless field weakening is required). This means that the current component in the direct axis is controlled to be zero: $i_d^* = 0$. The stator current vector is perpendicular to the flux of the permanent magnets in this way. Thus, rotor flux and stator current are decoupled: the torque will be proportional to i_q, while the useful flux component is not influenced by the stator current anymore. If T_{em}^* is the desired value for the torque, the reference values for the current components can be computed using the equation for the electromagnetic torque (Eq. (4.11)):

$$i_q^* = \frac{T_{\text{em}}^*}{-\frac{3}{2}N_p L_d i_{\text{mag}}} = \frac{T_{\text{em}}^*}{K} \tag{6.1}$$

$$i_d^* = 0 \tag{6.2}$$

For a PMSM, PTC and FOC are thus rather similar. The only difference is that Eq. (6.2) replaces Eq. (5.6) of the predictive torque controller. While Eq. (5.6) implicitly imposes that i_d^* should be zero to reduce the Joule losses, this is done explicitly for FOC. In [21], it is concluded that as a result of this explicit requirement the Joule losses for FOC are reduced in comparison with PTC. The loophole of PTC is that, when i_q does not equal its reference value i_q^*, Eq. (5.6) implies that i_d should not equal zero, which causes indeed higher Joule losses. Only when $i_d = 0$ the current layer and the magnetic flux density by the permanent magnets reach their maximum values simultaneously, which results in the maximal producible electromagnetic torque per amp. The MSE(i_d) and MAE(i_d) values for PTC of Table 5.2 confirm that i_d deviates from zero. This extra freedom for PTC makes it more difficult to understand the actions of the controller as well, since a change in i_q affects both controlled variables.

As is mentioned in [14], FOC is also known under the name vector control, since magnitude, frequency and instantaneous positions of voltage, current and flux linkage space vectors are regulated. Vector control is a general control philosophy that acts on the positions of the space vectors and provides their correct orientation both in steady-state and during transients. In scalar control on the contrary, which is based on relationships valid in steady-state, only the magnitude and frequency of the space vectors are regulated.

In the following sections, some attention is devoted to the implementation of FOC using different types of controllers. Since plenty of implementations exist, this overview does not claim to be complete, but is meant to give an idea of the possibilities.

6.3 PI Controller

One way to implement the FOC strategy, is by using a PI controller. Figure 6.2 illustrates the main principles.

In a first step, the reference value for the torque T_{em}^* is converted to a reference value for i_q^*. The desired value for i_d is zero, according to the FOC principle. Then both current components are regulated using a PI controller. Such a PI controller is defined by the parameters K_p and K_i, representing respectively the proportional and integrating action. In the Laplace domain, the PI controller can be presented as:

$$PI(s) = \frac{sK_p + K_i}{s} \tag{6.3}$$

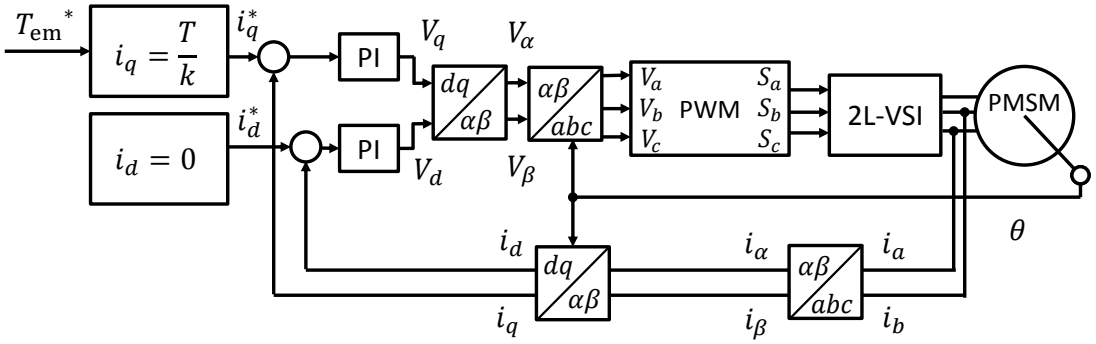


Figure 6.2: Principle of FOC with PI controller and PWM (Adapted from [2])

To tune the parameters, the electrical equations of the AFPMSM (Eq. (4.8)) are transformed to the Laplace domain as well:

$$V_q = R_s I_q + sL_q I_q - \omega L_d (I_d + I_{\text{mag}}) \tag{6.4}$$

$$V_d = R_s I_d + sL_d I_d + \omega L_q I_q \tag{6.5}$$

Due to the last terms of these two equations, representing the back EMF of the PMSM, the current control loops will not be independent. To be able to tune the PI parameters using a SISO system method, the loops need to be decoupled, for instance by assuming the speed ω to be zero[1]. In this way, the control loop for both the q- and d-axis is as given in Figure 6.3. The consequence of setting ω equal to zero during the design stage is that the PI controller will have less performance when applied to the real, coupled system than when applied to the decoupled system.

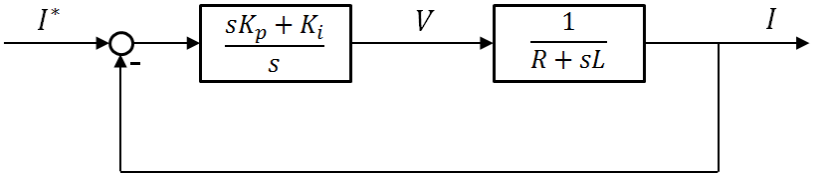


Figure 6.3: Decoupled current control loop

The closed loop transfer function I/I^* can be deduced from Figure 6.3:

$$\frac{I}{I^*}(s) = \frac{s(K_p/L) + K_i/L}{s^2 + s(R + K_p)/L + K_i/L} \tag{6.6}$$

The denominator can be rewritten in the standard form of a second order transfer function:

$$s^2 + \frac{R + K_p}{L}s + \frac{K_i}{L} = s^2 + 2\zeta\omega_n s + \omega_n^2 \tag{6.7}$$

with damping factor ζ and natural frequency ω_n.

Imposing requirements for the settling time $T_{settling}$ [2] of the system and the maximum percentage overshoot $\%OS$ (shown in Figure 6.4), ζ and ω_n can be calculated according to the principles

[1] The ideal way is to use feedforward to decouple the current control loops.

[2] In order to avoid confusion with the sampling period T_s, the settling time will be denoted by $T_{settling}$ in this text, instead of the more common notation T_s.

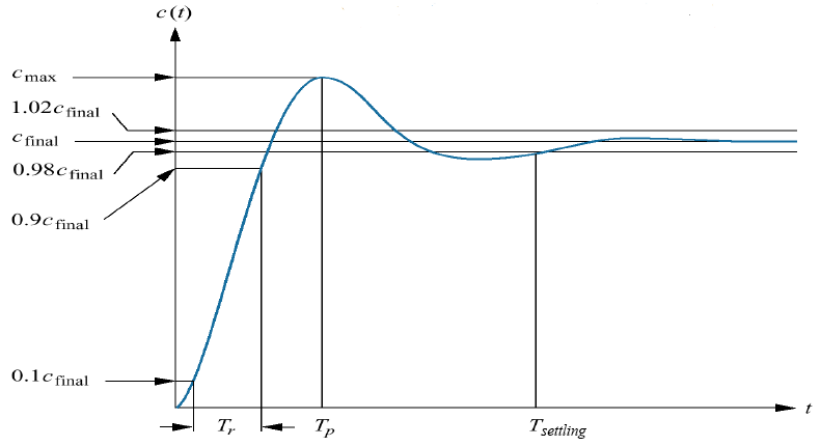


Figure 6.4: Control requirements for a second order transfer function: peak time T_p, overshoot $\%OS = 100(c_{max} - c_{final})/c_{final}$, rise time T_r and settling time $T_{settling}$ [24]

explained in [24]:

$$\zeta = \frac{-\ln(\%OS/100)}{\sqrt{\pi^2 + \ln^2(\%OS/100)}} \tag{6.8}$$

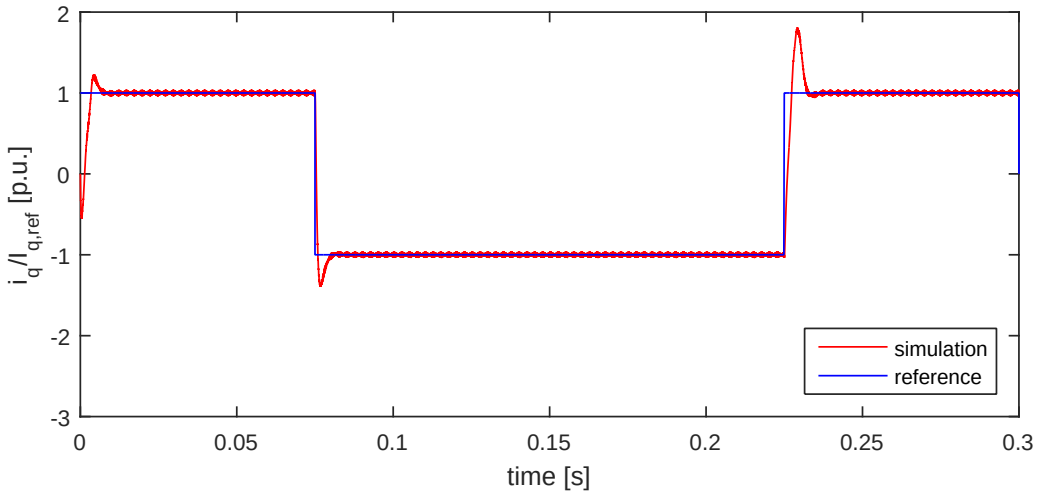
$$\omega_n T_{settling} = \frac{-\ln(0.02\sqrt{1-\zeta^2})}{\zeta} \tag{6.9}$$

Eventually K_p and K_i can be computed by comparing the coefficients of Eq. (6.7). Since for a SPMSM L_q equals L_d, the same controller can be used for both components.

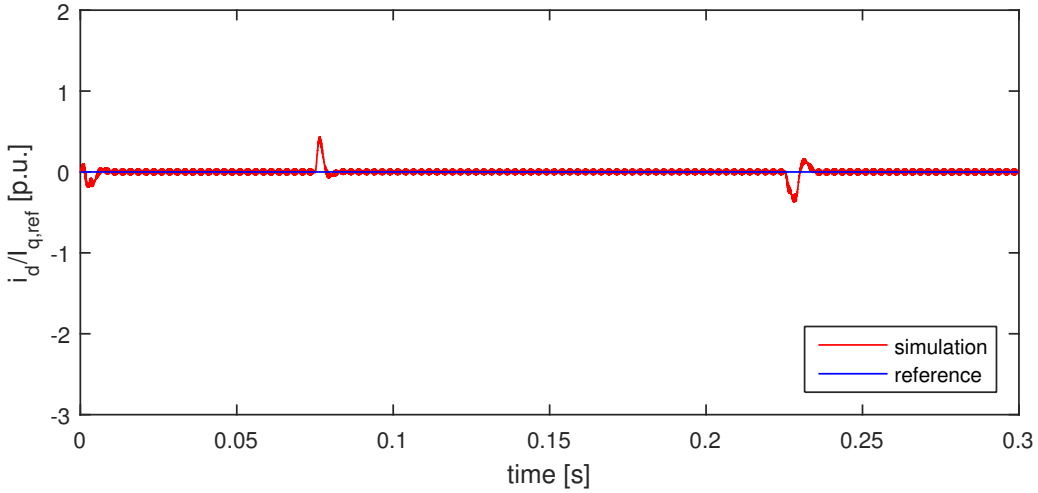
Figure 6.2 shows that the output of the PI controller provides the reference value for the stator voltages V in the qd-reference frame. After a rotation to the $\alpha\beta$-reference frame, PWM can be used to determine the switching sequence of the 2L-VSI. SVM is suited for this purpose as well. Since the 2L-VSI has only a finite DC bus voltage, the stator voltage will be limited to the radius of the circle inscribed into the hexagonal formed by the six VSI vectors (thus to $V_{\mathrm{dc}}/\sqrt{3}$) by means of an anti-reset windup. Eventually the stator currents and the position of the rotor are fed back from the AFPMSM to the control loop.

6.3.1 Simulation

Figure 6.5 shows the simulation results when the requirements are $\%OS = 2\%$ and $T_{settling} = 5$ ms, leading to $\zeta = 0.78$ and $\omega_n = 1123.6$ rad/s, or $K_p = 4.13$ and $K_i = 3206.4$. PWM is used for this simulation, with a carrier frequency of 10 kHz. The other simulation parameters are summarized in Table 5.1. An overview of the KPIs is presented in Table 6.1.



(a) q-component of the stator current



(b) d-component of the stator current

Figure 6.5: PI torque controller using the FOC principle and PWM

Table 6.1: KPIs of the PI controller

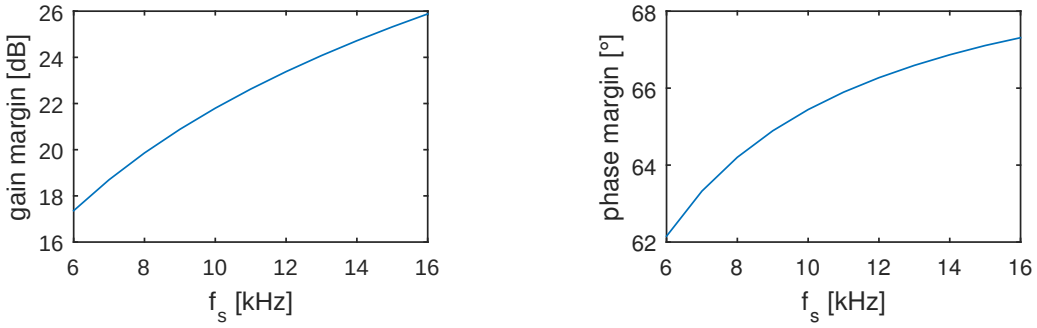
KPI	Value
MSE(T_{em}) = MSE(i_q) [p.u.]	$2.98 \cdot 10^{-4}$
MSE($\lvert\Psi_{\text{s}}\rvert$) [p.u.]	$5.74 \cdot 10^{-6}$
MSE(i_d) [p.u.]	$2.96 \cdot 10^{-4}$
MAE(T_{em}) = MAE(i_q) [p.u.]	0.0138
MAE($\lvert\Psi_{\text{s}}\rvert$) [p.u.]	0.0017
MAE(i_d) [p.u.]	0.0123
Joule loss [W]	12.00
f_{switch} [kHz]	20.00
non-PPCR [%]	0

It is clear that both current components follow their reference values well. However, the overshoot exceeds 2% and $T_{settling} \approx 10$ ms. The control is indeed slower for the real, coupled system than for the uncoupled system. The zero in the closed loop transfer function might

have increased the overshoot as well. Besides, the PI controller is implemented digitally: the performance and stability of the controller depends on the sampling frequency f_s. The higher f_s, the more the discrete time system approaches the continuous Laplace equations, and thus the more appropriate the calculated K_p and K_i values are. For low f_s, however, those control parameters need to be tuned again. In Figure 6.6 the gain and phase margins of the system for f_s ranging from 6 to 16 kHz are shown, indicating that the closed loop is stable in this frequency range.

The fact that the dynamics of i_q and i_d are coupled is visible when a step is applied in i_q^*. The cross-coupling terms representing the back EMF cause i_d to increase when i_q is decreased and vice versa. However, the controller is able to bring i_d back to zero in 10 ms.

Compared to the PTC controller, the MSE and MAE values are very low and the ripple on the controlled variables has decreased. However, this is actually not a fair comparison. The decrease is also caused by the fact that the PI controller does not keep the switch state constant during the whole update period T_u, while this is the case for PTC. Compared to PTC, the switching frequency f_{switch} has increased fivefold. Thus the switching losses have increased as well. On the other hand, the fast switching makes it possible to fulfill the PPCR at any time, improving the voltage quality. To allow a fair comparison, FOC will be implemented by means of FS-MBPC in the following section.



(a) Gain margin **(b)** Phase margin

Figure 6.6: Stability of the current control loop in function of f_s

6.4 Finite-Set Model Based Predictive Control

A second way to implement FOC is by means of FS-MBPC. This strategy is called predictive current control (PCC). The control scheme is given in Figure 6.7. According to Section 3.2 the FS-MBPC algorithm selects the most appropriate switch state by means of a cost function for every update period T_u, after which this voltage vector is applied during the entire following update period. The inputs of the controller are the stator current reference values computed according to the FOC principle (Eq. (6.1) and (6.2)) and the measurements of the stator current components, as well as the actual switch state of the inverter and the position of the motor.

This section will give an in-depth examination of the estimation, prediction and optimization step introduced in Section 3.2, specifically applied to FOC. The timing scheme of Figure 3.2 will be used. The main goal of the control is to follow the set-points of the stator current components accurately. Secondary control goals, like for instance reducing the switching losses, can be formulated as well by means of the cost function.

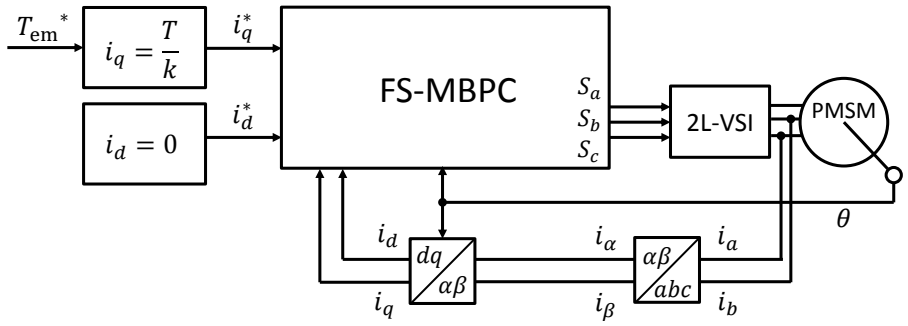


Figure 6.7: Principle of PCC (Adapted from [2])

Estimation

As was already explained in Section 3.2, the first step of the algorithm serves to compensate for the finite calculation time of the controller itself: the controller needs a finite time to select the subsequent optimal switch state. At update instant k a new switch state S_k is applied to the inverter, giving rise to a stator voltage change and consequently a change in the stator current. The new motor position and state of the current are measured at $k + 1/2$ to avoid measuring transient phenomena. Together with the inverter model (Section 4.3.1) and the discrete electrical equations (Eq. (5.1) and (5.2)) they enable to estimate the stator current components $\hat{i}_q^{k+1}$ and $\hat{i}_d^{k+1}$. Contrary to PTC, stator flux vector and torque estimation are not required.

Prediction

Since there are only eight different S_{k+1}, the easiest way to choose the optimal switching state is by predicting the output for all eight possibilities and afterwards picking the best one among them. The same equations that are used in the estimation step are applied to predict these outputs. Only now a time step T_u needs to be used, instead of a time step $T_\mathrm{u}/2$.

Optimization

In this last step a certain cost will be associated to each of the eight possible S_{k+1}. Depending on the imposed requirements on the control quality, various forms of cost functions exist for this purpose. The voltage vector resulting in the lowest cost is applied to the system.

6.4.1 Simulations

Just like for PTC, controllers with two different types of cost function will be simulated: a quadratic cost function and a cost function with tolerance bands.

An appropriate weighting factor W_I for the quadratic cost function

$$J^{k+1} = \frac{(\hat{i}_q^{k+2} - i_q^*)^2}{I_{q,\mathrm{ref}}^2} + W_I \frac{(\hat{i}_d^{k+2} - i_d^*)^2}{I_{q,\mathrm{ref}}^2} \tag{6.10}$$

can be found by plotting the MSE values of i_q and i_d in function of W_I. The result is presented in Figure 6.8. In general, MSE(i_q) increases with increasing W_I, while MSE(i_d) decreases. After all, a larger W_I means putting more emphasis on the control of i_d. For very small W_I, however,

the back EMF jeopardizes a proper control of i_q. Neglecting the regulation of i_d leads to a situation in which the stator voltage is not able to bring i_q to its reference value anymore, as was explained in detail in Section 5.5. $W_I = 1$ leads to an accurate regulation of both i_q and i_d. The simulation results for this weighting factor are presented in Figure 6.9. Its KPIs are summarized in Table 6.2. Since MSE(i_q)+MSE(i_d) has a flat optimum, small deviations of the weighting factor will have limited influence.

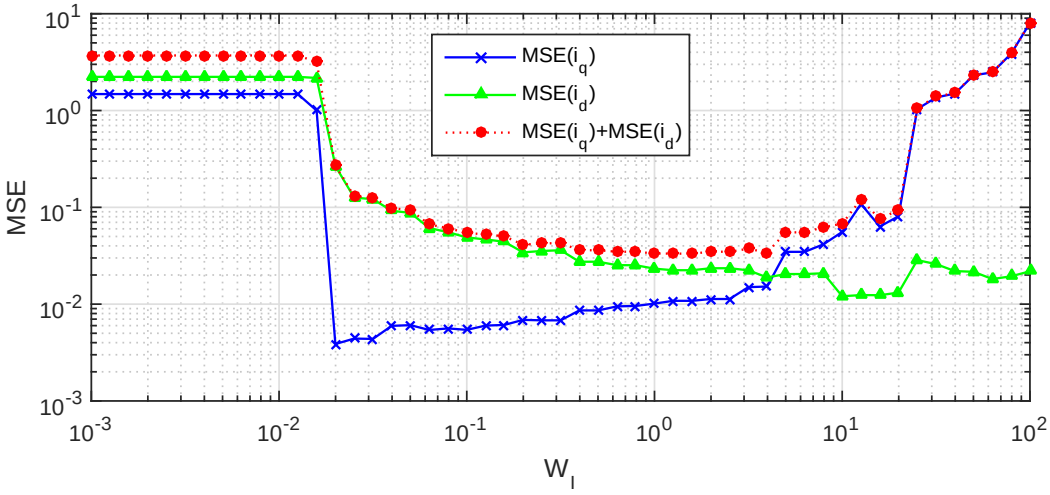


Figure 6.8: MSE for varying weighting factor W_I in a quadratic cost function

Table 6.2: Comparison in KPIs of a quadratic cost function and a cost function with tolerance bands

KPI	quadratic	tolerance band
MSE(T_{em}) = MSE(i_q) [p.u.]	0.0101	0.0104
MSE($\lvert\Psi_{\text{s}}\rvert$) [p.u.]	0.0014	0.0004
MSE(i_d) [p.u.]	0.0234	0.0062
MAE(T_{em}) = MAE(i_q) [p.u.]	0.0849	0.0864
MAE($\lvert\Psi_{\text{s}}\rvert$) [p.u.]	0.0320	0.0165
MAE(i_d) [p.u.]	0.1291	0.0652
Joule loss [W]	42.87	41.41
f_{switch} [kHz]	3.11	3.20
non-PPCR [%]	24.90	30.84

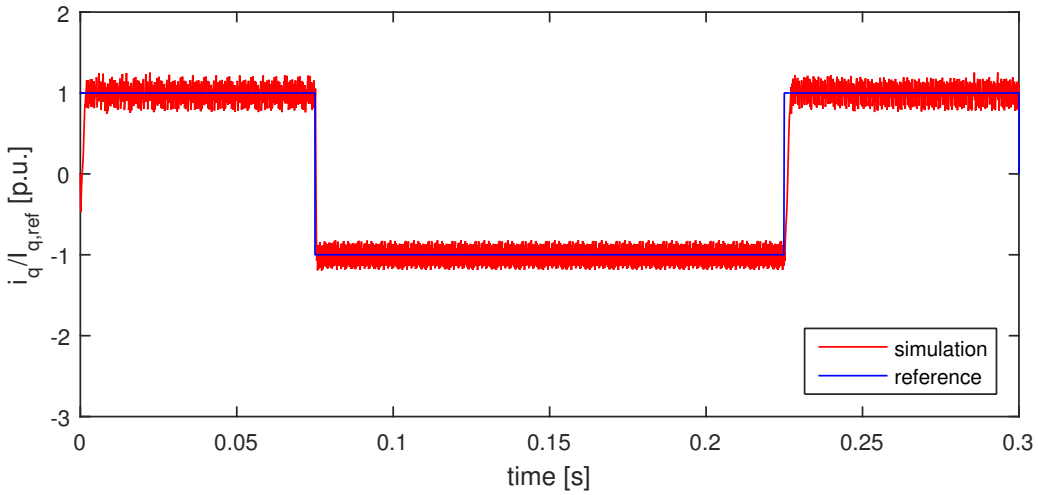
For a second simulation is opted for a cost function with tolerance bands:

$$J_{I_q}^{k+1} = \begin{cases} 0, & 0 \leq |\Delta I_q| \leq I_{q,\text{l}} \\ (\Delta I_q)^2, & I_{q,\text{l}} \leq |\Delta I_q| \leq I_{q,\text{u}} \\ W_{I_q\infty}(\Delta I_q)^2, & |\Delta I_q| > I_{q,\text{u}} \end{cases} \tag{6.11}$$

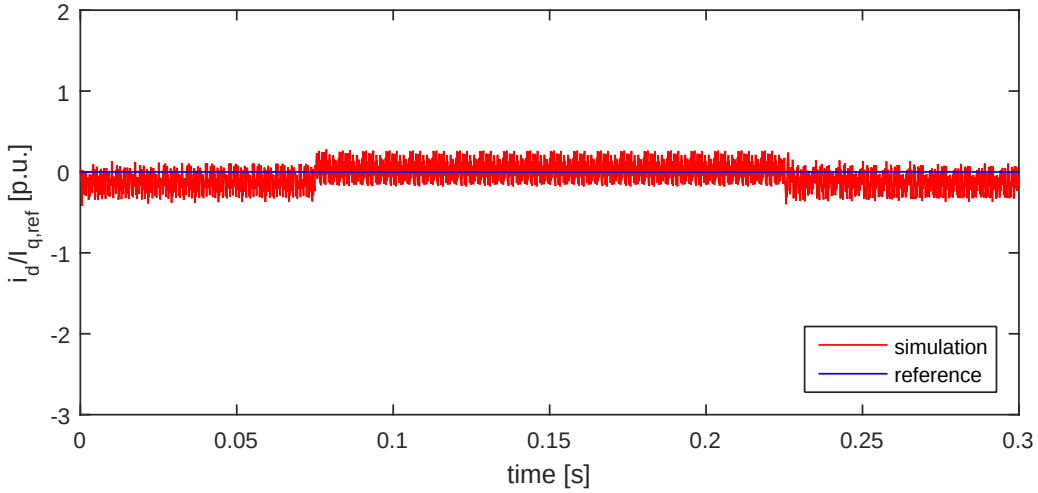
$$J_{I_d}^{k+1} = \begin{cases} 0, & 0 \leq |\Delta I_d| \leq I_{d,\text{l}} \\ W_I(\Delta I_d)^2, & I_{d,\text{l}} \leq |\Delta I_d| \leq I_{d,\text{u}} \\ W_{I_d\infty}(\Delta I_d)^2, & |\Delta I_d| > I_{d,\text{u}} \end{cases} \tag{6.12}$$

$$J^{k+1} = J_{I_q}^{k+1} + J_{I_d}^{k+1} \tag{6.13}$$

When $I_{q,\text{l}}$, $I_{q,\text{u}}$ and $I_{d,\text{u}}$ are imposed by the application, appropriate values for W_I and $I_{d,\text{l}}$ can be found by studying their influence on the KPIs, as described in Section 5.5 for PTC. Figure



(a) q-component of the stator current



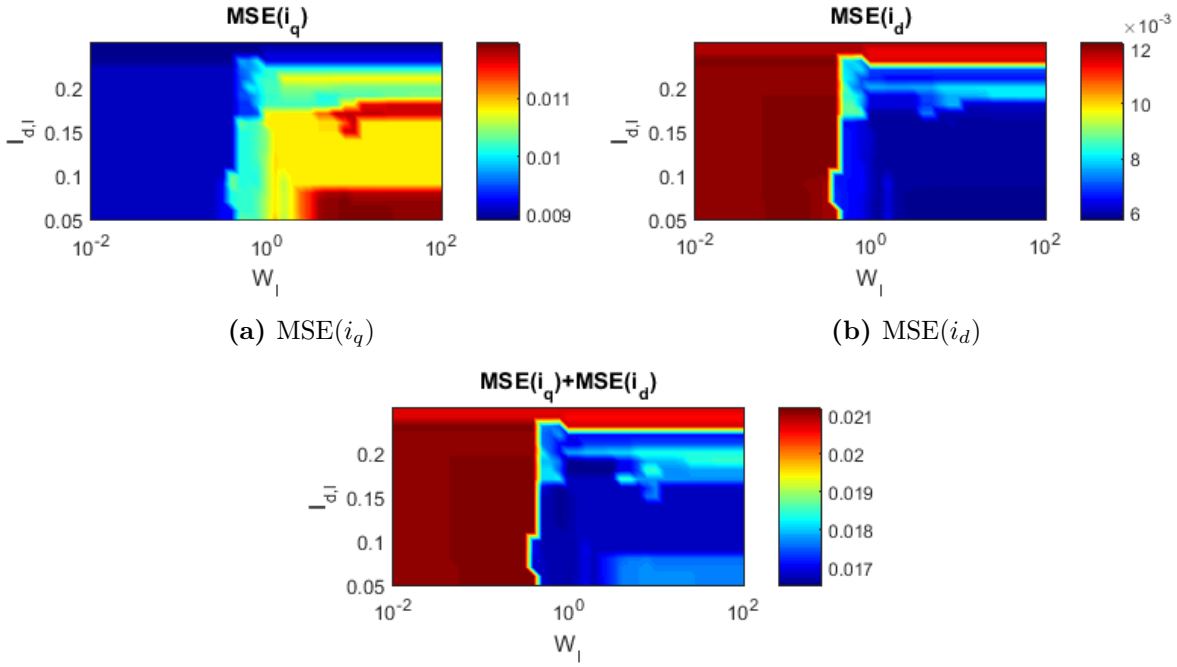
(b) d-component of the stator current

Figure 6.9: PCC with quadratic cost function ($W_I = 1$)

6.10 illustrates how the MSEs vary in function of W_I and $I_{d,\mathrm{l}}$. For the simulations is opted for $I_{q,\mathrm{l}} = 0.15$ p.u., $I_{q,\mathrm{u}} = 0.25$ p.u., $W_{I_q\infty} = 100$, $W_{I_d\infty} = 100$ and $I_{d,\mathrm{u}} = 0.25$ p.u. The main difference with PTC is that MSE(i_d) does not vary a great deal with $I_{d,\mathrm{l}}$ for low W_I, while the effect of $|\Psi_\mathrm{s}|_\mathrm{l}$ on MSE($|\Psi_\mathrm{s}|$) for low W_Ψ is more pronounced, as can be seen in Figure 5.10. When W_I is low, little emphasis is put on the regulation of i_d, irrespective of the width of the inner tolerance band of i_d. The situation involving the stator flux is more complicated, since i_q is proportional to T_em, and has a direct influence on $|\Psi_\mathrm{s}|$, as is clear from Eq. (5.3). Therefore a low W_Ψ does not necessarily result in a high MSE($|\Psi_\mathrm{s}|$): this value is kept low by the low error on T_em, and thus on i_q.

A similar line of thought can be followed to explain the abrupt changes in MSE(i_q) and MSE(i_d), while MSE(T_em) and MSE($|\Psi_\mathrm{s}|$) vary more smoothly. For $W_I = 0.5$ the cost term related to i_d in cost function (6.13) surpasses the term related to i_q, causing the priority of the controller to switch from keeping i_q in its inner tolerance band to keeping i_d in its inner tolerance band. Thus, when the controller focuses on keeping J_{I_q} low, J_{I_d} increases and vice versa. The situation is different for PTC. Since $|\Psi_\mathrm{s}|$ depends on both i_q and i_d, J_T and J_Ψ can both be decreased by improving the control of i_q. However, when the deviation of i_d from zero grows, this will only

increase J_Ψ and not J_T. J_Ψ can also be lowered while J_T rises: this situation occurs when the error in i_d decreases at a higher rate than the error in i_q increases. Briefly worded: focusing on keeping J_T low does not necessarily result in an increase in J_Ψ and vice versa.



(a) MSE(i_q) **(b)** MSE(i_d)

(c) MSE(i_q)+MSE(i_d)

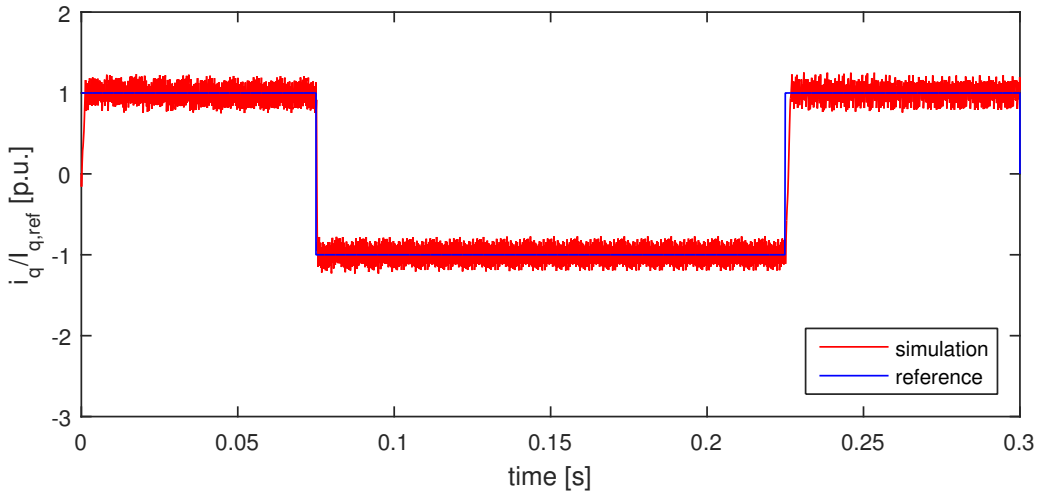
Figure 6.10: Effect of tolerance band and weighting factor on MSE

Bearing in mind the effect of $|\Psi_s|_l$, $I_{q,l}$ also influences f_{switch}, the percentage of switchings violating the PPCR and the Joule losses. A good compromise between the quality of the voltage and the control of i_q on the one hand, and the Joule losses and the quality of the control of i_d on the other hand, can be found for $W_I = 1$ and $I_{d,l} = 0.15$. The simulation results for these parameters are plotted in Figure 6.11, while its KPIs are listed in Table 6.2. From the results in this table can be concluded that a cost function with tolerance bands can also improve the regulation of the controlled variables, by means of narrow tolerance bands. However, the advantageous effects on f_{switch} and the percentage of switchings not fulfilling the PPCR vanish in this case.

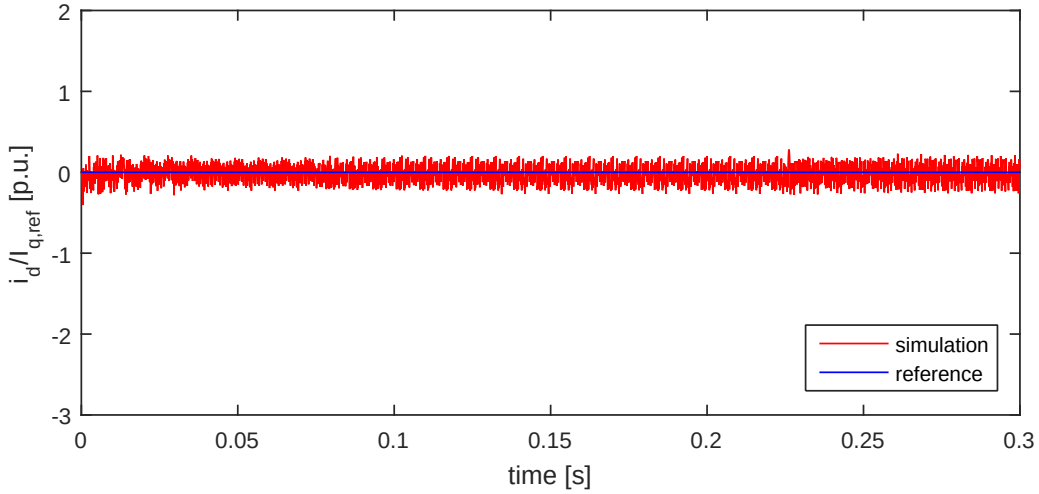
6.4.2 Effect of a Larger Prediction Horizon

In Section 3.2, the basic working principle of MBPC is explained. In this explanation of the basic steps of FS-MBPC is assumed that the effect of the control action applied at instant k already affects the state at instant $k+1$. This is not always the case: in general a system can exhibit a dead time N_d larger than one. The states at instants $k+i$ ($i \in \{1, ..., N_d\}$) need to be estimated in the estimation step, based on the previous control inputs. In Figure 6.12 these control inputs are denoted as $u(t)$, the states as $x(t)$.

Moreover, the prediction phase does not need to be limited to one step in the future. A prediction step with a prediction horizon of N time steps is possible as well. The control horizon N_u, in which a change of the control signals is allowed, can be adapted as well. In the general case, the evolution of the controlled variables is predicted over an interval $[k+N_d+1, k+N_d+N]$ for all the possible control sequences in the interval $[k+1, k+N_u]$. N_u does not need to equal N. When $N_u < N$ the control signals can be varied over the first N_u update periods, after which



(a) q-component of the stator current



(b) d-component of the stator current

Figure 6.11: PCC with cost function with tolerance bands ($W_I = 1$, $I_{d,1} = 0.15$)

the input remains constant.

In the optimization step, the optimal control action will be selected by means of a cost function. Only the first control signal of the sequence will be effectively applied to the system, since at the following update instant the estimation, prediction and optimization step are repeated. This is called the receding horizon principle.

The amount of required calculations increases with the power of N_u. When $N = N_u = 1$ (as is the case in all the previous simulations), only $2^3 = 8$ switching possibilities need to be evaluated. For a larger control horizon, this number increases to 8^{N_u}. Thus for $N_u = 2$ already 64 states need to be calculated in the prediction step, and for $N_u = 3$ even 512. This larger required computation capacity can only be justified if the control quality is significantly improved.

A simulation performed for $N_d = 1$ and $N = N_u = 3$, with cost function

$$J^{k+1} = \sum_{n=1}^{N} \left[\frac{(i_q^{k+1+n} - i_q^*)^2}{I_{q,\text{ref}}^2} + W_I \frac{(i_d^{k+1+n} - i_d^*)^2}{I_{q,\text{ref}}^2} \right] \tag{6.14}$$

for $W_I = 1$, demonstrates however that an extended prediction horizon even results in a degra-

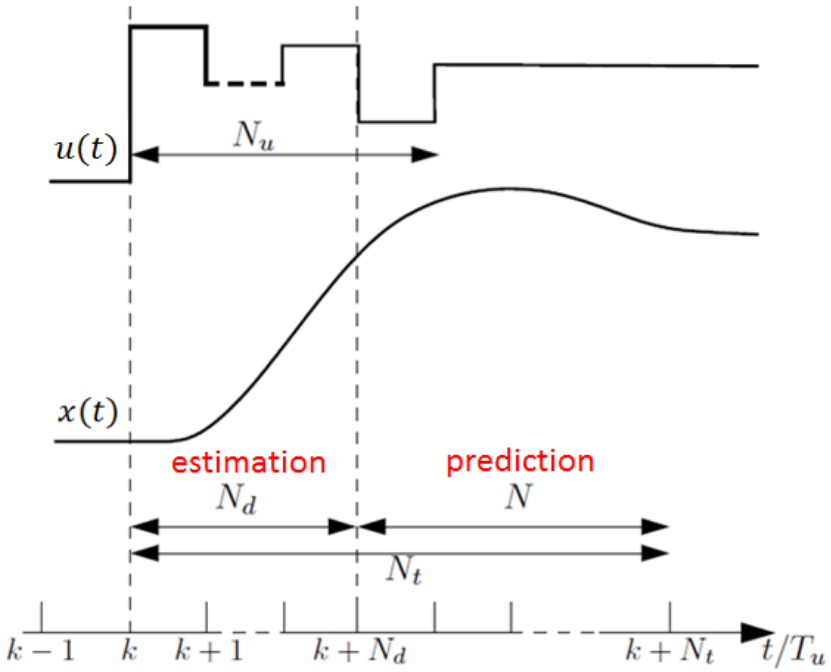


Figure 6.12: MBPC principle with general dead time N_d, prediction horizon N and control horizon N_u (Adapted from [2])

dation of the control quality, as testified by the KPIs summarized in the second column of Table 6.3. The cause of this poor result lies in the discretization of the machine equations. Irrespective of the local truncation error of the first order Euler approximation, the assumption that θ is constant during the entire prediction phase leads to an accumulation of errors. If θ is adapted by means of the linear model

$$\theta_{i+1} = \theta_i + \omega_i \cdot \Delta t \tag{6.15}$$

during the prediction step, a slight improvement of the KPIs is achieved by an expansion of the prediction horizon. Nevertheless, this modest improvement is not worth the extra computational burden. In fact, this poor result could have been predicted a priori, bearing in mind the perfect prediction model used in simulation (solely deviating from the implemented machine by linearization), and the fact that only a limited amount of input voltages is enabled by the 2L-VSI. Under these circumstances, the optimal switch state in the next update period will be almost always the same, regardless of how many update periods are considered. In [2, 25] is stated that for real-life implementations a larger prediction horizon even can lead to an inferior control quality, since the deviations between the (simplified) prediction model and the real machine will make predictions further in the future less accurate. In [26] is mentioned that the large computational burden for large N requires low update frequencies, which will increase the torque ripple and reduce the quality of the stator currents.

In [22,27] is stated that increasing N does not increase the control quality if no additional control terms (such as number of switch events) are added to the cost function. The situation changes however if it is a secondary goal to improve the voltage quality. Increasing the prediction horizon may be useful in this context. This secondary goal requires an adaptation of the cost function:

$$J^{k+1} = \sum_{n=1}^{N} \left\{ \frac{(i_q^{k+1+n} - i_q^*)^2}{I_{q,\mathrm{ref}}^2} + W_I \frac{(i_d^{k+1+n} - i_d^*)^2}{I_{q,\mathrm{ref}}^2} + W_S \left[\sum_{j=1}^{3} \left(S_j^{k+n} - S_j^{k+n-1} \right) \right] \right\} \tag{6.16}$$

where W_S is a weighting factor representing the importance of the improvement of the voltage quality relative to the importance of the current control , and $j \in \{1, 2, 3\}$ represents the three

Table 6.3: Effect of the prediction horizon on the KPIs for cost function (6.14)

	without θ correction		with θ correction
KPI	$N = 1$	$N = 3$	$N = 3$
MSE(T_{em}) = MSE(i_q) [p.u.]	0.0101	0.0108	0.0099
MSE($\lvert\Psi_{\text{s}}\rvert$) [p.u.]	0.0014	0.0016	0.0014
MSE(i_d) [p.u.]	0.0234	0.0269	0.0216
MAE(T_{em}) = MAE(i_q) [p.u.]	0.0849	0.0880	0.0842
MAE($\lvert\Psi_{\text{s}}\rvert$) [p.u.]	0.0320	0.0342	0.0317
MAE(i_d) [p.u.]	0.1291	0.1388	0.1238
Joule loss [W]	42.87	42.93	44.12
f_{switch} [kHz]	3.11	3.20	3.15
non-PPCR [%]	24.90	21.75	23.31

phases $\{a, b, c\}$. The extra term in the cost function discourages any change in the switch state that is not absolutely necessary for a good current control, and especially changes that violate the PPCR. Small errors in i_q and i_d will thus be tolerated, if this permits to make less changes in the switch states. The KPIs in Table 6.4 for $W_I = 1$ and $W_S = 0.1$ prove that f_{switch} and the percentage non-PPCR indeed have lowered by means of cost function (6.16), at the expense of the other control quality parameters. Extending the prediction horizon to $N = 3$ causes the percentage non-PPCR to reduce even further. In principle, the controller is able to evolve towards every possible switch state in this period without any violation of the PPCR, independent of the initial state of the inverter.

Table 6.4: Effect of the prediction horizon on the KPIs for cost function (6.16) with extra cost term

KPI	$N = 1$	$N = 3$
MSE(T_{em}) = MSE(i_q) [p.u.]	0.0165	0.0218
MSE($\lvert\Psi_{\text{s}}\rvert$) [p.u.]	0.0019	0.0018
MSE(i_d) [p.u.]	0.0265	0.0225
MAE(T_{em}) = MAE(i_q) [p.u.]	0.1063	0.1225
MAE($\lvert\Psi_{\text{s}}\rvert$) [p.u.]	0.0360	0.0348
MAE(i_d) [p.u.]	0.1332	0.1192
Joule loss [W]	48.92	52.88
f_{switch} [kHz]	2.06	2.33
non-PPCR [%]	9.71	2.38

In [28] a control scheme called Long Horizon Few Switches (LHFS) is proposed using a longer prediction horizon, without applying the receding horizon principle. The investigated strategy searches not only for the optimal voltage vectors, but also for the best switching instants, at which the next iteration of the FS-MBPC scheme needs to be performed. This new method reduces the switching frequency efficiently, without compromising the current tracking too much.

6.4.3 Effect of the Discretization

In Section 4.2.3 is explained how the machine model is linearized and discretized by means of the first order Euler approximation. In [11] can be found that the local truncation error will be of order $\mathcal{O}(\Delta t^2)$ in this case. In order to obtain a more accurate discretization when the quantity x has a changing slope in the time interval Δt, the modified Euler approximation can

be used as well:

$$x_{i+1}^{EU} = x_i + \Delta t \cdot f(x_i, u_i, i) \tag{6.17}$$

$$x_{i+1} = x_i + \frac{\Delta t}{2} \cdot \left(f(x_i, u_i, i) + f(x_{i+1}^{EU}, u_{i+1}, i+1 \right) \tag{6.18}$$

where $x_i = x(t)$, $x_{i+1} = x(t + \Delta t)$ and $f(x_i, u_i, i)$ represents the derivative of the quantity x evaluated at time t using the input u_i and state x_i at that same time instant. The local truncation error is reduced to order $\mathcal{O}(\Delta t^3)$.

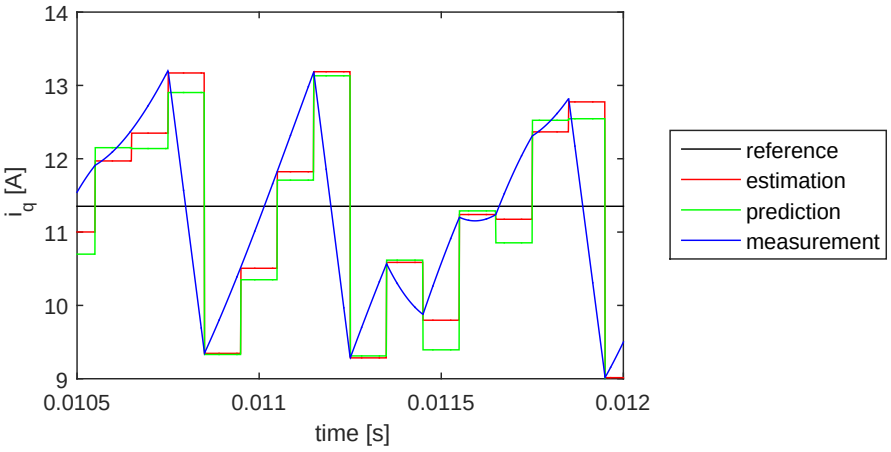
In Figure 6.13 can be seen that the difference between the estimated and measured current i_q has not visibly improved by using the modified Euler method, since the time step for the linearization is only $T_\mathrm{u}/2$ in this case. The error between the measured current and the predicted current, however, has improved a little. This is due to the fact that the error from the estimation has accumulated with the error from the linearization with $\Delta t = T_\mathrm{u}$ in the prediction step. However, the effect of the linearization method on the control quality is negligible, as can be concluded from the KPIs given in Table 6.5. For the considered operating point, MSE(i_d) even increases a little when the modified Euler approximation is used, while it decreases for other operating points. Even for a larger prediction horizon ($N = 3$), where the linearization errors are accumulated, the influence of changing to the modified Euler approximation on the KPIs is negligible. Since there are only eight possible switch states, a small change in the predicted values will only very rarely influence the choice of S_{k+1} in the optimization step. Therefore the first order Euler approximation will be used throughout this master's dissertation.

Table 6.5: Effect of the linearization method on the control quality

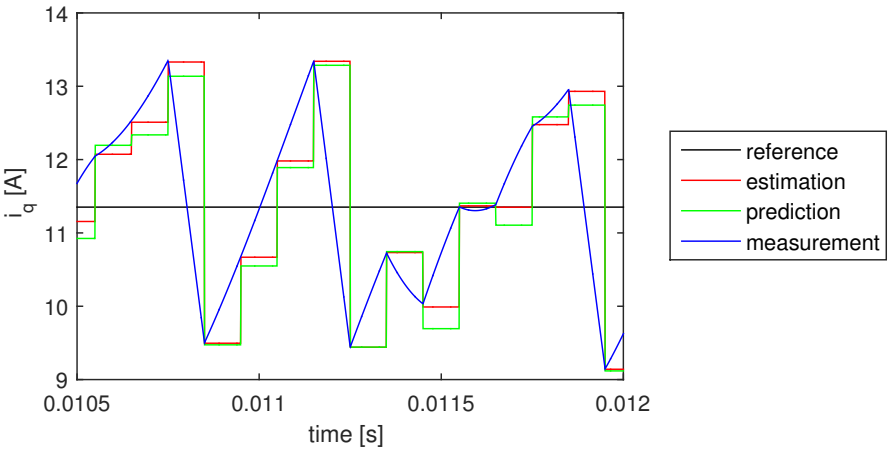
	$N = 1$		$N = 3$	
KPI	first order	modified	first order	modified
MSE(T_em) = MSE(i_q) [p.u.]	0.0101	0.0101	0.0099	0.0099
MSE($\lvert\Psi_\mathrm{s}\rvert$) [p.u.]	0.0014	0.0016	0.0014	0.0013
MSE(i_d) [p.u.]	0.0234	0.0259	0.0216	0.0201
MAE(T_em) = MAE(i_q) [p.u.]	0.0849	0.0842	0.0842	0.0845
MAE($\lvert\Psi_\mathrm{s}\rvert$) [p.u.]	0.0320	0.0341	0.0317	0.0310
MAE(i_d) [p.u.]	0.1291	0.1367	0.1238	0.1185
Joule loss [W]	42.87	42.86	44.12	44.36
f_switch [kHz]	3.11	3.11	3.15	3.11
non-PPCR [%]	24.90	20.47	23.31	22.67

6.4.4 Torque and Current Ripple

It can be noticed that the current ripple of FS-MBPC is large in comparison with the simulation results for PI control. In many high performance applications, low acoustic noise and high efficiency are highly demanded, as well as a long lifetime. Torque ripple is a critical concern in this context. Since the existence of torque ripple degrades the control accuracy and makes the motor less stable, numerous research activities have been conducted to solve this problem, for instance in [18, 29, 30]. Because the magnitude of the torque ripple is influenced by the voltage magnitude of the DC bus and the update frequency of the controller, it might be advantageous to adapt these two parameters. However, if the magnitude of the DC bus voltage is too small, a quick torque response cannot be assured anymore. Reducing T_u lowers the oscillation amplitude as well, but is not a suitable solution either, since this results in an increase in the switching loss and requires a larger computational effort.



(a) First order Euler approximation



(b) Modified Euler approximation

Figure 6.13: Effect of the discretization method. The estimated and predicted current have been shifted back in time over $T_u/2$.

The main issue of the FS-MBPC algorithm is that the number of voltage vectors that can be applied during an update period T_u is limited. One solution to overcome this problem is the use of multilevel inverters, providing more voltage vectors in modulation and thus lower torque ripple and lower harmonic content. Nonetheless, an increasing amount of power switches increases the complexity of the system and the hardware cost as well. Moreover, the number of voltage vectors remains limited in this case. A more effective solution is to allocate a fraction of T_u to an appropriate null vector. This is achieved by using a space vector pulse width modulation (SVPWM)-based control scheme, making use of the principle explained in Section 4.3.2. Two main advantageous features can be realized by means of this technique: inputs with a varying average voltage amplitude, and a fixed switching frequency, just like for the PI controller. In the following sections, two SVPWM-based control schemes will be presented, making use of FOC. Similar practices can be adopted for torque control as well. In [30], for instance, a DTC algorithm using SVM for IPMSMs is investigated, in order to attain low ripple in flux and torque. The authors of [18], [29], and [31] propose a PTC strategy in which an optimal reference voltage vector is calculated based on a mathematical model for the torque, and applied to the machine using SVPWM.

6.5 Deadbeat Control

A deadbeat controller combines the advantages of both FS-MBPC and PI control: on the one hand, it can benefit from its knowledge of the machine model; on the other hand, it is used in combination with a PWM, reducing the current and torque ripple. Moreover, the substantial calculation effort of testing all feasible voltage vectors is avoided by using a DB solution. Hence, the process of selecting the best switch state is optimized.

In Section 3.3 is explained how the deadbeat algorithm makes use of a model inverse solution to compute the reference value for the manipulated input, required to bring the output to its reference value in one time step T_u. In this section, the use of the DB algorithm for FOC will be elaborated. The control scheme is given in Figure 6.14. The inputs for this controller are the stator current reference values computed according to Eq. (6.1) and (6.2), the measured current components, the previously computed stator voltage reference value, and the motor position.

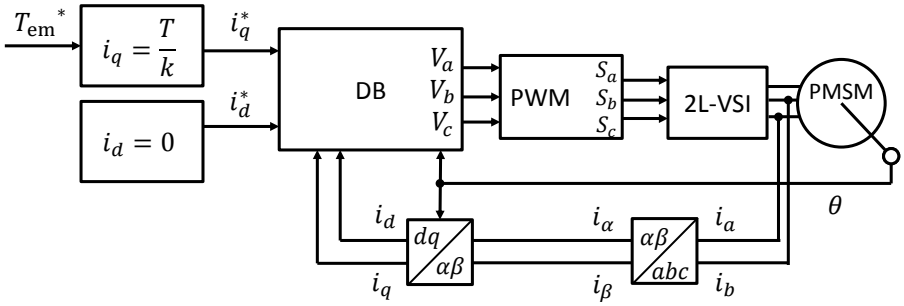


Figure 6.14: Principle of FOC with DB and PWM (Adapted from [2])

Estimation

The stator voltage components computed in the previous update period and currently applied to the PMSM (v_q^k and v_d^k), and the stator current components measured at half the update period ($i_q^{k+1/2}$ and $i_d^{k+1/2}$) are used to calculate the stator currents at instant $k+1$ by means of the discrete time electrical model given by Equations (5.1) and (5.2). This estimation step is identical to the estimation step of FS-MBPC.

Deadbeat

The aim of the controller is to bring i_q and i_d to their reference values in one time step T_u. To obtain the voltage V_{k+1}^* required to achieve this goal, it is assumed that the stator current equals its set-point at instant $k+2$: $\hat{i}_q^{k+2} = i_q^*$ and $\hat{i}_d^{k+2} = i_d^*$. In this way, the voltage to be applied at $k+1$ can be computed as:

$$v_q^{k+1} = R_\mathrm{s}\hat{i}_q^{k+1} + \frac{L_q}{T_\mathrm{u}}\left(i_q^* - \hat{i}_q^{k+1}\right) - \omega^{k+1/2}L_d\left(\hat{i}_d^{k+1} + i_\mathrm{mag}\right) \tag{6.19}$$

$$v_d^{k+1} = R_\mathrm{s}\hat{i}_d^{k+1} + \frac{L_d}{T_\mathrm{u}}\left(i_d^* - \hat{i}_d^{k+1}\right) + \omega^{k+1/2}L_q\hat{i}_q^{k+1} \tag{6.20}$$

The reference waves $v_{x\mathrm{o}}$ ($x \in \{a, b, c\}$) for the voltages over the load are obtained by applying the rotation (4.13) to the $\alpha\beta$-reference frame, followed by the inverse Clarke transformation

(4.12). Eventually, a PWM algorithm determines the switch states during the update period, as explained in Section 4.3.2. Since the PWM enables to vary the duty cycle of each inverter leg separately, not only the magnitude of the voltage vector can be changed, but also its direction.

In case the amplitude of the control action computed by the DB controller is larger than the maximum feasible input vector amplitude of the inverter, the vector amplitude will be saturated to the maximum value that fits in the hexagon of Figure 4.7. This so-called clipping strategy is far from optimal, but more sophisticated limitation methods are out of the scope of this master's dissertation.

The deadbeat control algorithm does not involve a cost function. The tedious activity of tuning this cost function is hence avoided. The other side of the coin is that the option to impose secondary control goals disappears as well.

Since the DB controller determines more than one configuration of the inverter during one update period, T_u can be increased to obtain the same switching frequency as for FS-MBPC. In [32] is pointed out that the real-time constraint of the DB controller is then less than for FS-MBPC, so cheaper controllers can be used. Moreover, for FS-MBPC the mean switching frequency depends on the operating point, while DB allows to have a more or less constant and known switching frequency, equal to two times the update frequency.

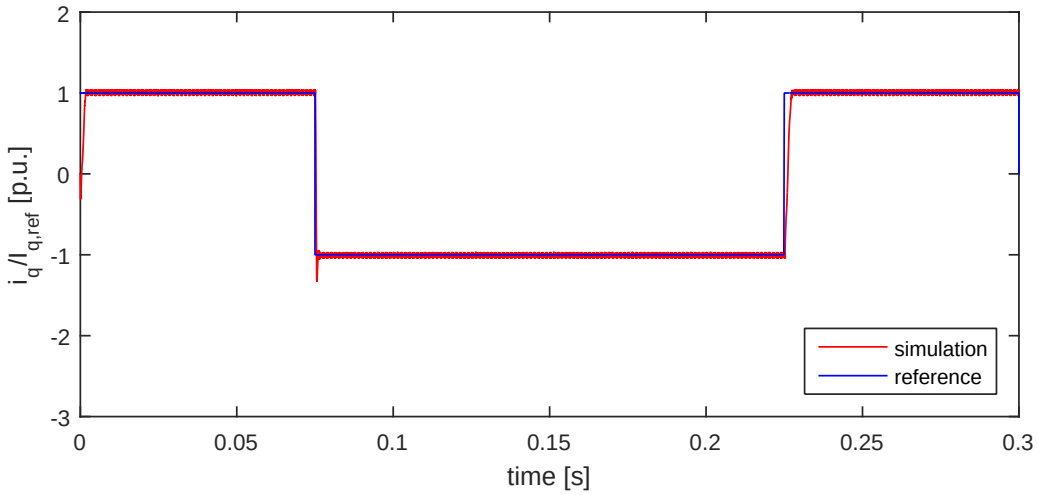
In [26], a DB control idea is used to reduce the calculation effort of the classic PTC. The main idea is to estimate the voltage vector that is required to bring the controlled variables to their reference value by means of the DB principle. Only the three voltage vectors near this optimal input vector are necessary to be tested by the cost function. Thus, the exhaustive evaluation of all the feasible voltage vectors of the inverter is avoided.

6.5.1 Simulation

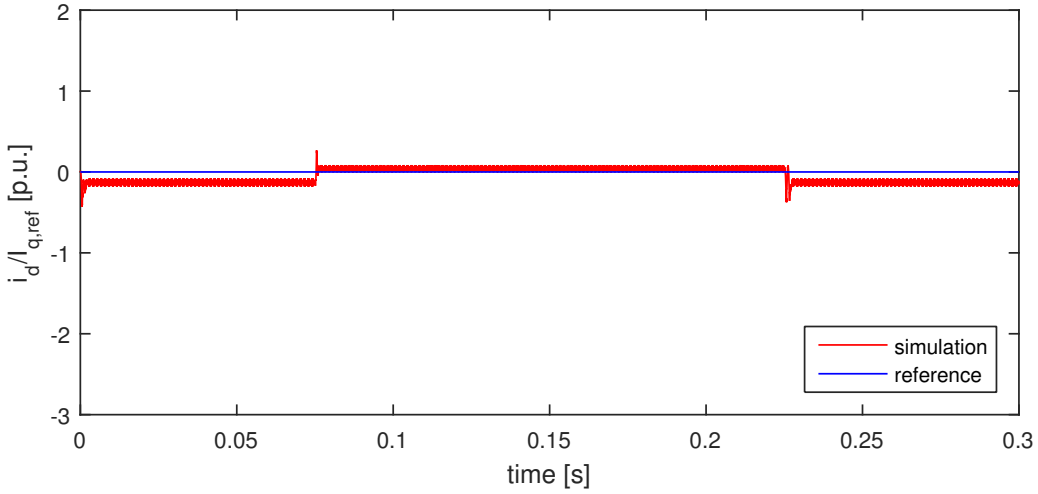
The major advantage of DB control compared to PI control is that the former can use the information provided by the machine model. Compared to FS-MBPC, the current ripple (and thus the torque ripple as well) is reduced by using DB control, which is clearly visible in the simulation results of Figure 6.15. Notice that the current peaks at the steps in the i_q set-point are strongly reduced compared to the PI control shown in Figure 6.5. However, it must be admitted that the other KPIs of the PI controller still outshine the DB controller. The major disadvantage of DB controllers mentioned in literature is that they are prone to instability in case of parameter mismatch. After all, the DB solution relies on precise machine parameters to calculate the appropriate input voltage. In practice, however, it is difficult to implement accurate parameters, due to variations injected by noise or temperature. Therefore the robustness of the controllers will be compared in Section 7.2.4.

Table 6.6: KPIs of the DB controller

KPI	Value
MSE(T_em) = MSE(i_q) [p.u.]	$4.34 \cdot 10^{-4}$
MSE($\lvert\Psi_\mathrm{s}\rvert$) [p.u.]	0.0011
MSE(i_d) [p.u.]	0.0176
MAE(T_em) = MAE(i_q) [p.u.]	0.0172
MAE($\lvert\Psi_\mathrm{s}\rvert$) [p.u.]	0.0330
MAE(i_d) [p.u.]	0.1313
Joule loss [W]	43.68
f_switch [kHz]	17.33
non-PPCR [%]	0.02



(a) q-component of the stator current



(b) d-component of the stator current

Figure 6.15: DB torque controller using the FOC principle and PWM

6.6 Model Based Predictive Control with Duty Cycle Calculation

Although the DB controller provides a solution to decrease the torque ripple, it also disables the opportunity to impose secondary control goals by means of the cost function. Therefore, a controller combining these two features will be proposed in this section. The structure of the algorithm is similar to that of FS-MBPC, with the only difference that an optimal duty cycle for the applied voltage vectors is determined as well, responding to the need for inputs with a varying voltage amplitude. In [32], this technique is called Two-Configuration Predictive Control (2PC). The authors of [16, 33, 34] implemented a similar controller for PTC. The algorithm elaborated in this section is derived from the PTC with optimal duty cycle control for induction motor drives presented in [34].

Estimation

This step is identical to the estimation step of FS-MBPC, elaborated in Section 6.4.

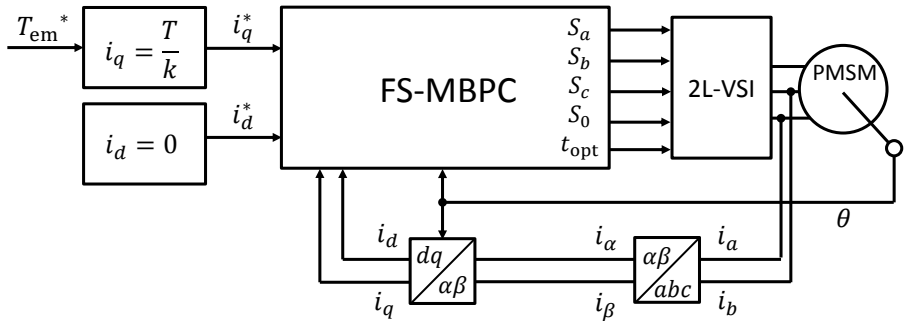


Figure 6.16: Principle of FS-MBPC with duty cycle calculation (Adapted from [2])

Prediction

Contrary to the classic FS-MBPC scheme, in which all eight switch states of the inverter are evaluated successively, and eventually only one voltage vector is selected and applied during the whole update period T_u, the new proposed algorithm evaluates only the six active vectors separately. Just a fraction of the control period is allocated to the optimal active vector. The rest of the time a null voltage is applied. Therefore two slopes must be calculated in each of the six steps: the current slope s_i $(i = 1, ..., 6)$ if the active vector is applied, and the current slope s_0 if a zero vector is applied. Those slopes can be calculated by means of the dynamic model derived in Section 4.2.2.

$$\begin{cases} s_{q,0}^{k+1} = \frac{1}{L_q} \cdot \left[-R_\mathrm{s} \hat{i}_q^{k+1} + \omega^{k+1/2} L_d \left(\hat{i}_d^{k+1} + i_\mathrm{mag} \right) \right] \\ s_{q,i}^{k+1} = \frac{1}{L_q} \cdot \left[v_{q,i}^{k+1} - R_\mathrm{s} \hat{i}_q^{k+1} + \omega^{k+1/2} L_d \left(\hat{i}_d^{k+1} + i_\mathrm{mag} \right) \right] \end{cases} \tag{6.21}$$

$$\begin{cases} s_{d,0}^{k+1} = \frac{1}{L_d} \cdot \left[-R_\mathrm{s} \hat{i}_d^{k+1} - \omega^{k+1/2} L_q \hat{i}_q^{k+1} \right] \\ s_{d,i}^{k+1} = \frac{1}{L_d} \cdot \left[v_{d,i}^{k+1} - R_\mathrm{s} \hat{i}_d^{k+1} - \omega^{k+1/2} L_q \hat{i}_q^{k+1} \right] \end{cases} \tag{6.22}$$

The optimal duration $t_{\mathrm{opt},i}^{k+1}$ for the active voltage vector under consideration is determined according to the deadbeat principle, by assuming that $\hat{i}_q^{k+2}$ equals its reference value:

$$i_q^* = \hat{i}_q^{k+2} = \hat{i}_q^{k+1} + s_{q,0}^{k+1} \cdot \left(T_\mathrm{u} - t_{\mathrm{opt},i}^* \right) + s_{q,i}^{k+1} \cdot t_{\mathrm{opt},i}^* \tag{6.23}$$

and thus

$$t_{\mathrm{opt},i}^* = \frac{i_q^* - \hat{i}_q^{k+1} - s_{q,0}^{k+1} \cdot T_\mathrm{u}}{s_{q,i}^{k+1} - s_{q,0}^{k+1}} \tag{6.24}$$

$$\begin{cases} t_{\mathrm{opt},i}^{k+1} = 0, & \text{if } t_{\mathrm{opt},i}^* < 0 \\ t_{\mathrm{opt},i}^{k+1} = t_{\mathrm{opt},i}^*, & \text{if } 0 \leq t_{\mathrm{opt},i}^* \leq T_\mathrm{u} \\ t_{\mathrm{opt},i}^{k+1} = T_\mathrm{u}, & \text{if } T_\mathrm{u} < t_{\mathrm{opt},i}^* \end{cases} \tag{6.25}$$

Eventually, the current components at instant $k+2$ can be predicted based on s_0^{k+1}, s_i^{k+1} and $t_{\mathrm{opt},i}^{k+1}$.

$$\hat{i}_q^{k+2} = \hat{i}_q^{k+1} + s_{q,0}^{k+1} \cdot \left(T_\mathrm{u} - t_{\mathrm{opt},i}^{k+1} \right) + s_{q,i}^{k+1} \cdot t_{\mathrm{opt},i}^{k+1} \tag{6.26}$$

$$\hat{i}_d^{k+2} = \hat{i}_d^{k+1} + s_{d,0}^{k+1} \cdot \left(T_\mathrm{u} - t_{\mathrm{opt},i}^{k+1} \right) + s_{d,i}^{k+1} \cdot t_{\mathrm{opt},i}^{k+1} \tag{6.27}$$

Optimization

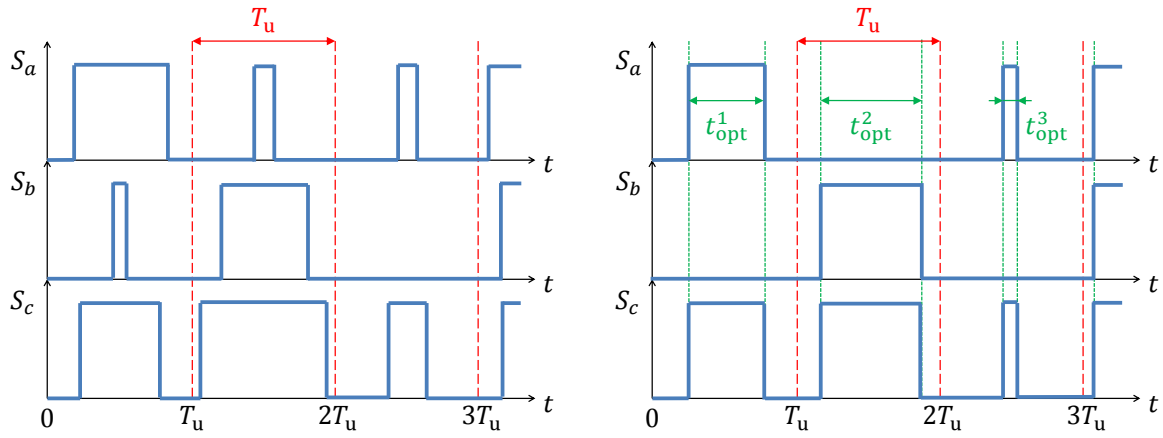
To compare the appropriateness of the six active voltage vectors and their accompanying duty cycle, a cost is assigned to each of them:

$$J^{k+1} = \frac{\left(\hat{i}_q^{k+2} - i_q^*\right)^2}{I_{q,\text{ref}}^2} + W_I \frac{\left(\hat{i}_d^{k+2} - i_d^*\right)^2}{I_{q,\text{ref}}^2} \tag{6.28}$$

To allow an easy comparison with the previous controllers, it is opted to use a quadratic cost function. Nonetheless, other types of cost functions are also feasible.

The selected null vector is the one that requires the lowest number of switches to change their state.

Contrary to DB control followed by a modulation algorithm, the optimization of the duty cycle only allows to change the magnitude of the applied voltage vector, and not its direction. The difference in switchings is illustrated in Figure 6.17.



(a) DB control

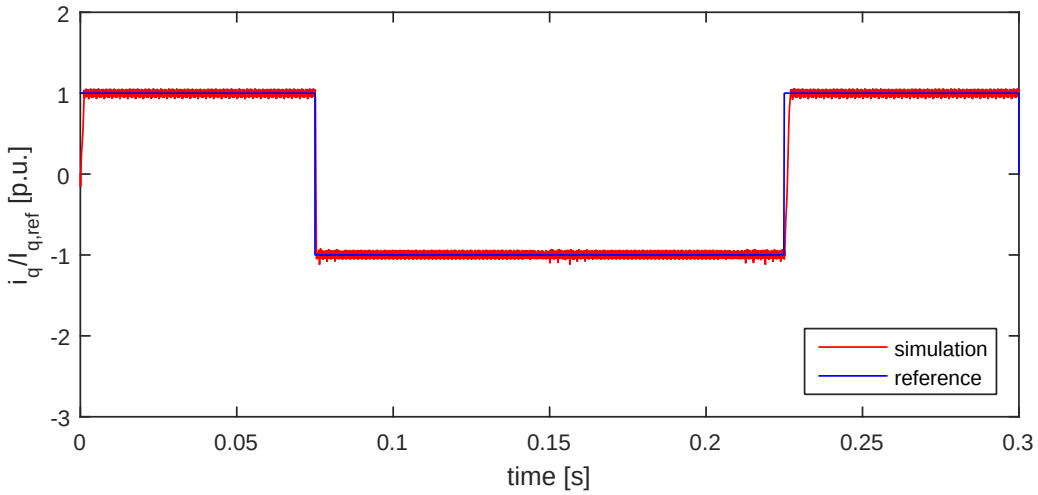
(b) PCC with DC calculation

Figure 6.17: Difference in modulation between DB control and PCC with DC calculation

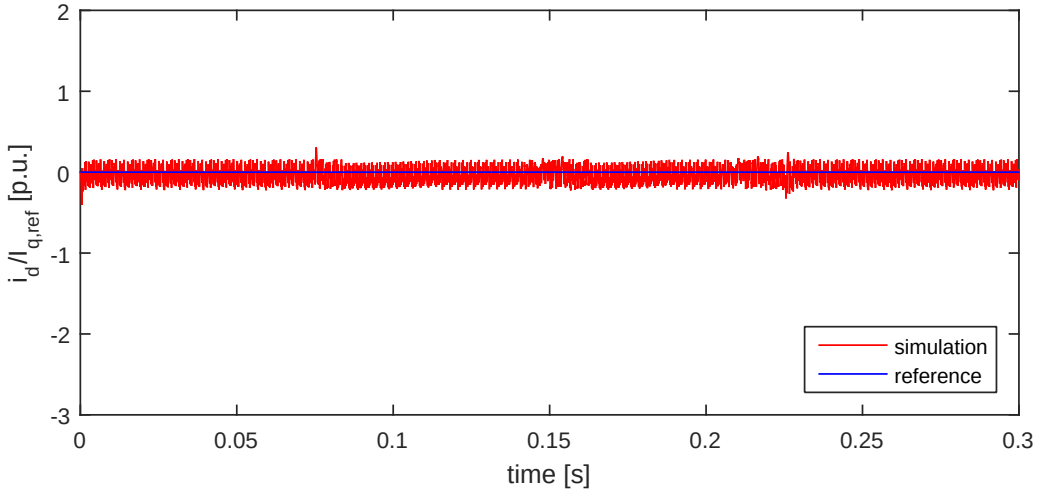
In [16] a similar control algorithm is used to reduce the torque ripple of induction motor drives controlled by means of PTC. However, instead of taking the duty cycle calculation into account in the optimization process, t_{opt} is determined after the selection of the active vector in this article. Although the torque ripple is effectively reduced in this way, in [34] is justly remarked that the active voltage vector selected by means of conventional FS-MBPC may no longer be the optimal one when a zero vector is inserted along with it. Therefore, the selection of the vector and the computation of its duration are optimized simultaneously in this master's dissertation.

6.6.1 Simulation

To allow a fair comparison with PCC without the calculation of t_{opt}, $W_I = 1$ is used for the simulation of the new controller as well. In Figure 6.18 can be seen that the current ripple in the q-component of the stator current has reduced. This same effect is reflected in the KPIs: the MSE and MAE values are low, but at the expense of a worse voltage quality.



(a) q-component of the stator current



(b) d-component of the stator current

Figure 6.18: PCC with duty cycle calculation

Table 6.7: KPIs of the PCC with duty cycle calculation

KPI	PCC with DC
MSE(T_{em}) = MSE(i_q) [p.u.]	$9.2797 \cdot 10^{-4}$
MSE($\lvert\Psi_{\text{s}}\rvert$) [p.u.]	$2.9616 \cdot 10^{-4}$
MSE(i_d) [p.u.]	0.0050
MAE(T_{em}) = MAE(i_q) [p.u.]	0.0256
MAE($\lvert\Psi_{\text{s}}\rvert$) [p.u.]	0.0128
MAE(i_d) [p.u.]	0.0532
Joule loss [W]	41.6635
f_{switch} [kHz]	15.19
non-PPCR [%]	78.13

6.7 Conclusion

This chapter started with an explanation of the FOC principle. According to FOC, the stator current can be thought of as consisting of two orthogonal components. One component (i_q) is responsible for the torque of the PMSM, the other component (i_d) for the flux. Since it is assumed that the permanent magnets completely define the rotor flux, the required i_d^* can be set equal to zero.

FOC can be implemented in different ways. Four implementations were discussed:

- PI control
- FS-MBPC
- DB control
- FS-MBPC with duty cycle calculation

The basic working principles of FS-MBPC and DB control introduced in Chapter 3 were elaborated for the specific case of FOC, and combined into the FS-MBPC with DC calculation strategy. A summary of these control algorithms can be found in the flow charts presented in Figures 5.14, 6.19 and 6.20 respectively. The state x represents the stator current components i_q and i_d.

Simulation results were given for each of the four control options. It was noticed that the PI controller led to smoother results than FS-MBPC, due to its use of SVPWM. SVPWM permits to change the amplitude and direction of the on average applied voltage vector during T_u, hence reduces the ripple in torque and current at the expense of higher switching losses. However, FS-MBPC has the advantage that it can make use of the information provided by the model of the AFPMSM. DB control combines these two advantages, but compared to FS-MBPC it allows less freedom concerning the control optimization. After all, the DB algorithm contains no cost function in which secondary control goals can be expressed. Therefore, the deadbeat and FS-MBPC principle were combined in a FS-MBPC algorithm with duty cycle calculation. The use of a cost function is preserved in this method, as well as the possibility to change the average amplitude of the voltage vector during an update period. Consequently, the torque ripple is reduced, while secondary control goals can still be included.

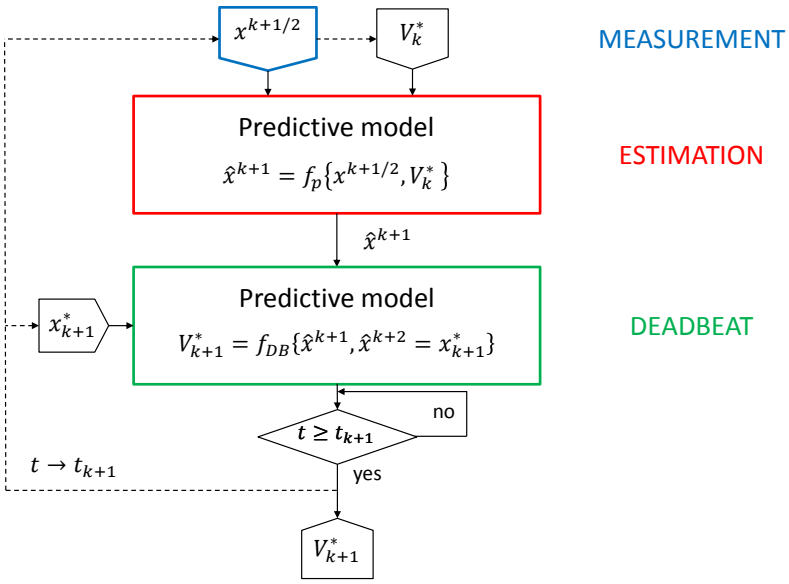


Figure 6.19: DB algorithm

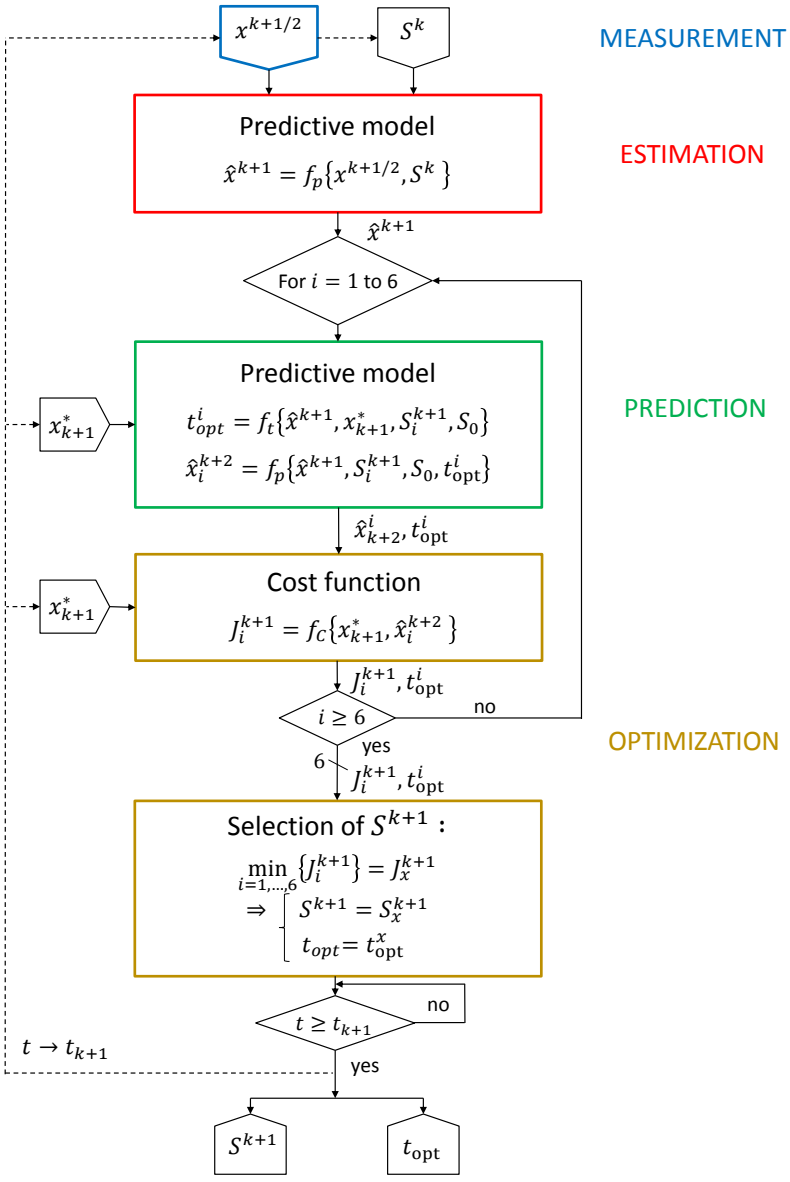


Figure 6.20: FS-MBPC with DC calculation algorithm

Chapter 7

Comparison of the Controllers

7.1 Introduction

In Chapter 5 and 6, different types of predictive controllers were studied. An overview of the most prominent ones is given in Figure 7.1. In order to be able to compare the performance of the predictive controllers with a more classical control strategy, the FOC principle was also implemented by means of a simple PI controller.

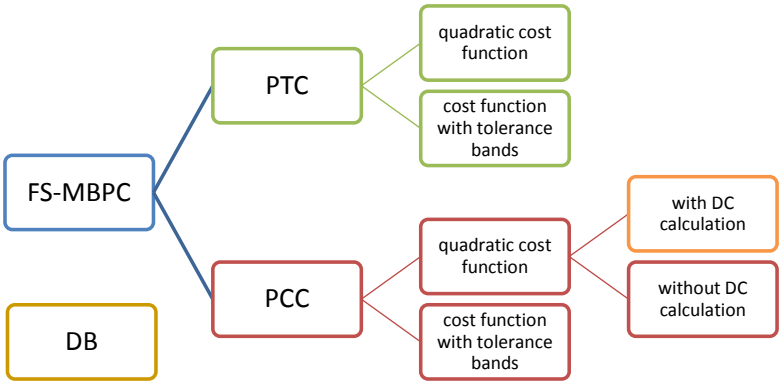


Figure 7.1: Overview of the most prominent predictive controllers of Chapter 5 and 6

In this chapter, a more in-depth evaluation and comparison of the control algorithms will be given. Apart from the KPIs proposed in Chapter 5, some new parameters will be introduced for this purpose as well.

In Chapter 8, the same tests will be conducted on a real AFPMSM, in order to validate the conclusions derived from the simulation results.

7.2 Key Performance Indicators

In Section 5.4, MSE, MAE, Joule losses, f_{switch}, and the percentage of switchings violating the PPCR were put forward to give an objective quantification of the global control quality. Although these parameters provide a well-founded basis to study the differences in behavior between the controllers, they do not cover all the nuances of a proper control, as will become clear in the next sections.

7.2.1 Control Quality

The main goal of the controllers is to track the reference values precisely. Torque and current ripple must be minimized, and their average values must approximate their reference as close as possible. In Section 5.4, the MSE and MAE values were used to evaluate how well a controller meets both requirements. However, neither of these two parameters can distinguish between the ripple on the one hand, and the average deviation of the controlled variables on the other hand. Since the machine model cannot be perfectly estimated, biased machine parameters will give biased currents and torques when the controllers are tested on the real machine. In [28] is justly remarked that this systematic error due to slightly inaccurate machine parameters should not be taken into account as ripple. To obtain a more nuanced assessment of the ripple and the systematic error, three extra KPIs are proposed in this master's dissertation: the mean absolute ripple (MAR), the mean square ripple (MSR) and the average deviation of the reference value x^* (bias):

$$\text{MAR}(x) = \frac{\sum_{k=1}^{m} |\bar{x} - x_k|}{m} \tag{7.1}$$

$$\text{MSR}(x) = \frac{\sum_{k=1}^{m} (\bar{x} - x_k)^2}{m} \tag{7.2}$$

$$\text{bias}(x) = \frac{\sum_{k=1}^{m} x_k - x_k^*}{m} \tag{7.3}$$

m is the amount of samples used in the calculation and $\bar{x}$ is the average value of the controlled variable x over these m samples. The difference between MSR and MAR is similar to the difference between MSE and MAE: MSR punishes large deviations more severely, while MAR is more convenient to interpret. To simultaneously express the quality of multiple control parameters, the weighted sum of multiple MSR, MAR and bias values is frequently used. Utilizing p.u. values is therefore strongly recommended.

In case of excessive bias, [35] proposes to add an integrating action, which sums the deviations of the controlled variables from their reference values over time. By adding this so-called I-action to the cost function, the bias can be brought close to zero. However, this extra cost term requires the tuning of an extra weighting factor. Moreover, the I-action can cause stability issues if not implemented properly.

7.2.2 Harmonic Distortion

In [8, 36] is recommended to study the total harmonic distortion (THD) that is present in the stator current, since it represents a rough measure of the motor losses and indicates the different harmonics introduced by the inverter switching.

Different expressions exist for the THD, but they all have in common that they make use of the root-mean-square (RMS) value of the fundamental current component I_1 and the DC component I_0. When the stator current $i_s(t)$ is decomposed by means of the Fourier transform

$$i_s(t) = I_0 + \sqrt{2} I_1 \cos(\omega t + \phi_1) + \sqrt{2} I_2 \cos(2\omega t + \phi_2) + ... \tag{7.4}$$

with fundamental frequency ω, the RMS-current can be expressed as:

$$I_{\text{RMS}} = \sqrt{I_0^2 + \sum_{n=1}^{\infty} I_n^2} \tag{7.5}$$

Since only the fundamental component is desired, the THD is defined as:

$$\text{THD} = \sqrt{\left(\frac{I_0}{I_1}\right)^2 + \sum_{n=2}^{\infty}\left(\frac{I_n}{I_1}\right)^2} \tag{7.6}$$

To apply this definition, the sum can be broken off after a sufficiently high number of harmonics, since the high frequency components are mostly less important.

7.2.3 Transient Behavior

The studied controllers not only differ in static performance, their dynamic behavior needs to be examined as well. In Figure 7.2, two parameters indicating the transient performance are introduced: the rise time T_{rise} and the overshoot OS. It is advised to express the overshoot as a percentage of the reference value.

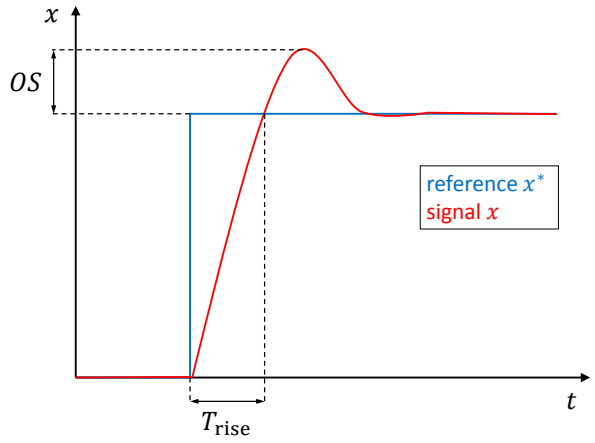


Figure 7.2: Definition of the dynamic behavior KPIs

7.2.4 Robustness

In [26,32,36–38] is mentioned that the performance of predictive controllers - and especially the DB controller - is often sensitive to model parameter mismatching errors, system delay, and the inverter's non-linearity. The feature that predictive controllers can use the information provided by the machine model, can thus be a disadvantage as well. Knowledge of system parameters such as R_s, L_q, L_d and back EMF is crucial for a good control quality, but their values cannot be obtained precisely, due to for instance disturbances, temperature variations, saturation and time delay. The flux of a permanent magnet machine is for example higher for a brand new machine than for a heavily used one, and the stator resistance increases with increasing temperature. Passing on inappropriate parameters might lead to unsatisfactory control.

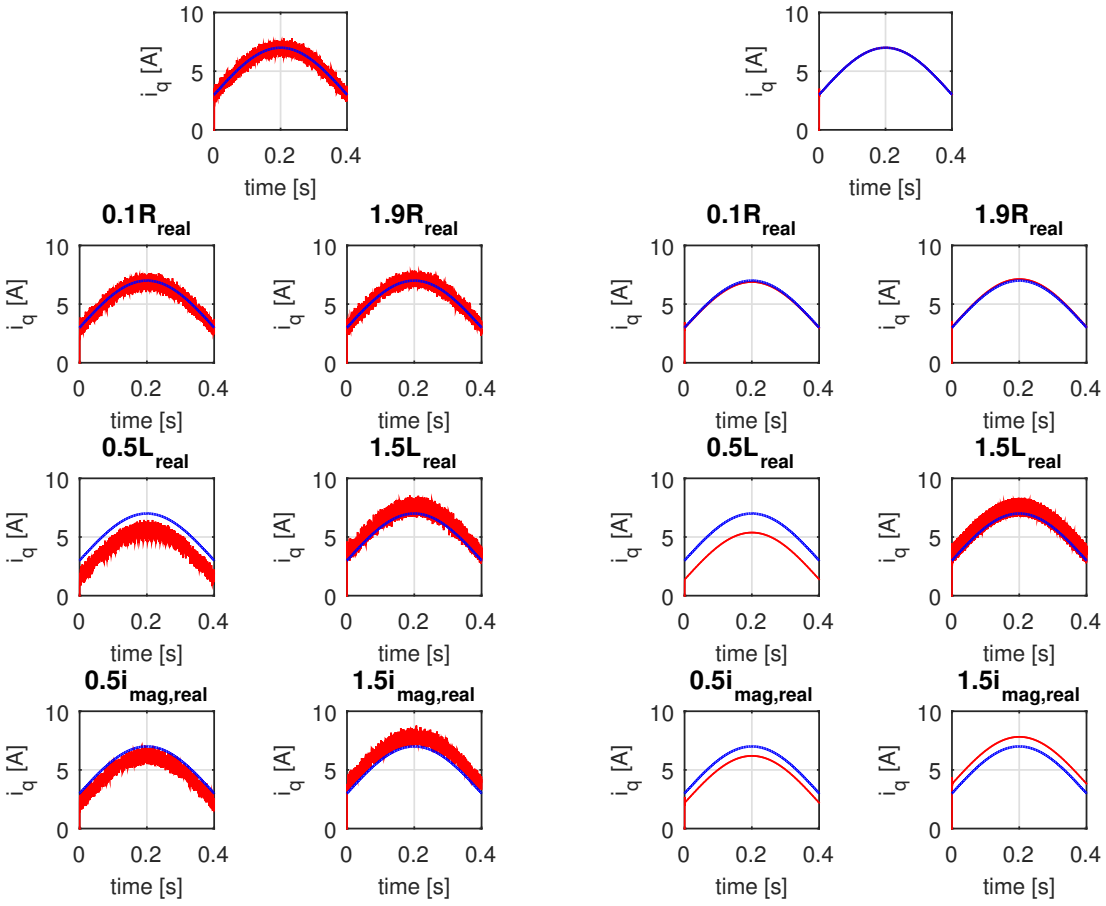
In [38] the issue of parameter sensitivity is solved by developing a model-free approach that alleviates the need for excessive prior knowledge about the system. Only the stator currents as well as the current difference corresponding to different switch states of the inverter are utilized. No additional information on resistance, inductance and back EMF is required. As a result, the new approach can be applied to non-PMSM AC motors as well, without adaptation. It is noteworthy that the experimental results provided in the mentioned article are satisfactory, both in transient as in steady-state behavior. This Model Free Predictive Control (MFPC) is

also simulated in [7]. It is concluded that MFPC leads to reliable results, but does not match the level of performance of MBPC.

According to [16], the estimation of the state variables of induction motor drives can be made more robust against motor parameter variations by introducing error feedback of the stator current. The authors also propose an alternative method to calculate t_{opt}, based on the difference between the measurements and the reference values and thus eliminating the machine parameter dependence.

In [37], the root locus technique is applied to analyze the stability of a deadbeat current controller for field oriented induction motor drives. In this text, a similar working method as in [32] is employed: a comparative study is performed to examine the difference in parameter sensitivity between various controllers.

Figure 7.3 shows simulation results for both the PCC with cost function (6.10) ($W_I = 1$) and the DB controller. The parameters in the 'real' machine in the loop are kept equal to the machine parameters summarized in Table 4.1, while the machine parameters in the control algorithm are adapted, in order to see the effect of inaccurate machine parameters on the control quality. The reference value T^*_{em} for the torque is adapted in such a way as to keep i^*_q the same, regardless of the parameter variations. The speed of the AFPMSM is set to 300 rpm and the DC bus voltage to 100V. The update frequency f_{u} amounts to 10 kHz, unless stated otherwise.



(a) PCC with quadratic cost function ($W_I = 1$) **(b)** DB controller

Figure 7.3: Parameter sensitivity (blue: reference i^*_q; red: simulated i_q)

From Figure 7.3 can be deduced that variations in R_{s} have little effect on the torque control

quality, while variations in $L = L_q = L_d$ and i_{mag} cause a systematic deviation between i_q and i_q^*. Moreover, an overestimation of L clearly causes the current ripple to increase for the DB controller. Similar figures for the other controllers, given in Section A.1 of the appendices, lead to similar conclusions. Therefore, it seems appropriate to examine the effect of parameter inaccuracy by means of the resulting change in bias and MAR:

$$\Delta\text{bias} = \frac{|\text{bias}| - |\text{bias}_{\text{ref}}|}{|\text{bias}_{\text{ref}}|} \tag{7.7}$$

$$\Delta\text{MAR} = \frac{\text{MAR} - \text{MAR}_{\text{ref}}}{\text{MAR}_{\text{ref}}} \tag{7.8}$$

where bias_{ref} and MAR_{ref} are respectively the bias and MAR values in case the parameters of Table 4.1 are used in both the 'real' machine and the control algorithm. Tables 7.1 and 7.2 give an overview of those parameters for the control of i_q by means of PCC and DB. Similar tables for the other controllers can be found in Appendix A.

Table 7.1: Robustness of PCC

KPI	$0.1 \cdot R_{\text{real}}$	$1.9 \cdot R_{\text{real}}$	$0.5 \cdot L_{\text{real}}$	$1.5 \cdot L_{\text{real}}$	$0.5 \cdot i_{\text{mag,real}}$	$1.5 \cdot i_{\text{mag,real}}$
Δbias [-]	13.1493	12.3433	234.4478	79.2537	120.3582	119.5224
ΔMAR [-]	0	0	0.0952	0.1238	-0.0095	0

Table 7.2: Robustness of DB

KPI	$0.1 \cdot R_{\text{real}}$	$1.9 \cdot R_{\text{real}}$	$0.5 \cdot L_{\text{real}}$	$1.5 \cdot L_{\text{real}}$	$0.5 \cdot i_{\text{mag,real}}$	$1.5 \cdot i_{\text{mag,real}}$
Δbias [-]	47.5790	50.8947	853.7368	282.4211	424.8947	426.7368
ΔMAR [-]	1.3333	1.6667	0	53.3333	0	0

The presumptions arisen from Figures 7.3a and 7.3b are confirmed by the results in Tables 7.1 and 7.2. The DB controller is indeed more sensitive to parameter mismatch than the PCC. The bias of the controllers is affected the most. Therefore Δbias will be used as the KPI to represent the robustness of the control strategies.

The results in Tables 7.1 and 7.2 also endorse the fact that an incorrect estimation of L or i_{mag} affects the controller's performance most. This can be understood by taking a closer look at the dynamic model equation (4.8). The term containing R_{s} is small compared to the current derivatives and the back EMF (since the simulations are at high speed), and thus the influence of R_{s} will be limited. An underestimation of the stator resistance causes i_q to stay just below its reference value on average, since the reference value for the voltage will be slightly underestimated in this case. An overestimation of R_{s} has the inverse effect.

When the absolute value of i_{mag} is underestimated, the back EMF term $(-\omega L_d(i_d + i_{\text{mag}}))$ is underestimated, leading to the application of a lower voltage - and thus a lower torque - than actually required. A similar way of thinking leads to a similar conclusion for the bias caused by an inaccurate stator inductance.

The effect of L on the torque ripple can be explained by means of Eq. (5.13), providing an expression for the derivative of i_q. Since L can be found in the denominator of this equation, an overestimation of L causes the controller to underestimate the torque ripple. Consequently, the torque ripple will be larger in the real machine.

7.3 Simulation Results

In this section, the performance of various controllers with different design choices (such as the type of cost function, or the update frequency) are compared. The same torque reference signal is applied to the following list of controllers:

(**a**) **PTC$_{\mathbf{quadr}}$**: PTC with quadratic cost function (5.5) with $W_\Psi = 3$

(**b**) **PTC$_{\mathbf{tol}}$**: PTC with cost function (5.10) with tolerance bands with $T_{\mathrm{em,l}} = 0.15$ p.u., $T_{\mathrm{em,u}} = 0.25$ p.u., $W_{T\infty} = 10$, $W_\Psi = 1$, $W_{\Psi\infty} = 100$, $|\Psi_\mathrm{s}|_\mathrm{l} = 0.12$ p.u. and $|\Psi_\mathrm{s}|_\mathrm{u} = 0.2$ p.u.

(**c**) **PCC$_{\mathbf{quadr}}$**: PCC with quadratic cost function (6.10) with $W_I = 1$

(**d**) **PCC$_{\mathbf{tol}}$**: PCC with cost function (6.13) with tolerance bands with $I_{q,\mathrm{l}} = 0.15$ p.u., $I_{q,\mathrm{u}} = 0.25$ p.u., $W_{I_q\infty} = 100$, $W_I = 1$, $W_{I_d\infty} = 100$, $I_{d,\mathrm{l}} = 0.15$ p.u. and $I_{d,\mathrm{u}} = 0.25$ p.u.

(**e**) **PCC$_{\mathbf{2f}}$**: PCC with quadratic cost function (6.10) and double f_u with $W_I = 1$

(**f**) **PCC$_{\mathbf{DC}}$**: PCC with DC calculation and quadratic cost function (6.10) with $W_I = 1$

(**g**) **PI** with $K_p = 4.13$ and $K_i = 3206.4$

(**h**) **DB**

A DC bus voltage of 250V is applied, to be able to control the machine rotating at 1000 rpm. The standard update frequency amounts to 10 kHz. Only for controller (**e**), f_u is doubled to 20 kHz.

The reference values to transform the cost terms into p.u. values are adapted to T^*_em according to Table 7.3, since a similar absolute deviation has relatively more importance for small T^*_em than for large T^*_em. Nevertheless, it must be kept in mind that the weighting factors in Chapter 5 were tuned for the T_ref and Ψ_ref values corresponding to $T^*_\mathrm{em} = 15$ Nm. Changing those reference values according to Table 7.3 actually increases the relative weight of the flux regulation. However, from Figures 5.9 and 5.10 can be concluded that putting more emphasis on the flux regulation does not jeopardize the control.

Table 7.3: Reference values to transform the cost terms into p.u. values

	$T^*_\mathrm{em} = 0$Nm	$T^*_\mathrm{em} = 8$Nm	$T^*_\mathrm{em} = 12$Nm	$T^*_\mathrm{em} = 15$Nm
T_ref [Nm]	15	8	12	15
$I_{q,\mathrm{ref}}$ [A]	1	6.0772	9.1158	11.3947
Ψ_ref [Wb]	0.1097	0.1108	0.1121	0.1135

The simulation results for all these controllers are assembled in Figures 7.5 and 7.6. Figure 7.5 shows the simulation plots for i_q, while Figure 7.6 contains the results for i_d. The difference in current ripple between the controllers strikes immediately.

The effect of changing T_ref and Ψ_ref with changing T^*_em is clearly visible for PTC in Figure 7.6a. For larger T^*_em, T_ref increases more than Ψ_ref, resulting in more strict control of $|\Psi_\mathrm{s}|$ and less strict control of T_em, and thus i_q. As a result, i_d will be controlled better when T^*_em is high, and

worse when T^*_{em} is low. This is not the case for PCC, since both cost terms are scaled by means of $I_{q,\text{ref}}$.

Taking a closer look at the current components of PCC_{DC} gives some deeper insight in its working principle. In Figure 7.4, a continuously repeating pattern can be noticed. Since the calculation of t_{opt} focuses on i_q only, i_d always starts to deviate more and more from zero, while i_q is kept close to i^*_q. At a certain time instant, however, the deviation of i_d is too flagrant, causing the cost function to select the voltage vector bringing i_d back to zero, although this results in a temporary larger deviation from i^*_q.

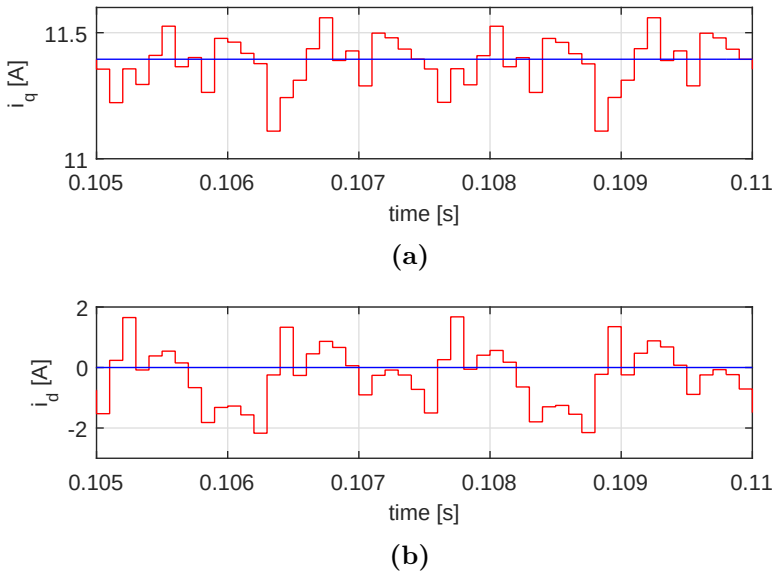
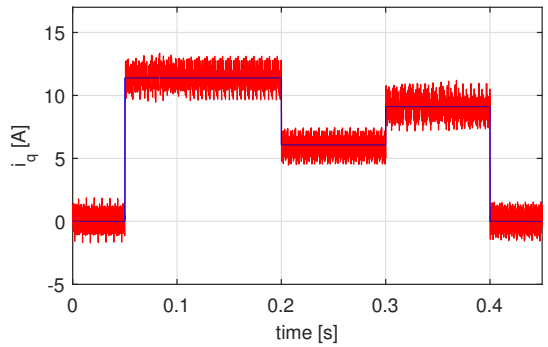


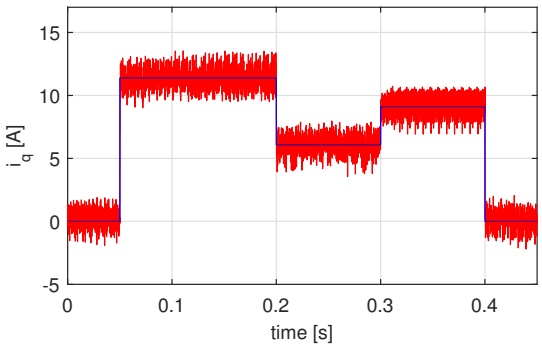
Figure 7.4: Detail views clarifying the control principle of PCC_{DC} (blue: reference; red: simulation)

The i_q-waveforms of both PI and DB control, shown in Figures 7.5g and 7.5h respectively, are the only two that show some distinct overshoot peaks. For DB control, these peaks are due to the fact that the first order Euler approximation is not that accurate when i_q has a changing slope. The fact that i_q and i_d are coupled by the back EMF, is visible in the peaks in i_d in Figure 7.6g: a change in i_q also affects i_d. Actually this link can be perceived in all i_d-plots for PCC in Figure 7.6, since i_d deviates more from zero on average if i^*_q is large. In Eq. (5.2) can be seen that the combination of a large ω and a large i_q are indeed very likely to decrease i_d. Therefore, prediction errors due to the discretization of the machine model can cause i_d to slightly deviate from zero, since in Figure 6.13 can be noticed that i_q is more frequently underestimated than overestimated in the prediction step.

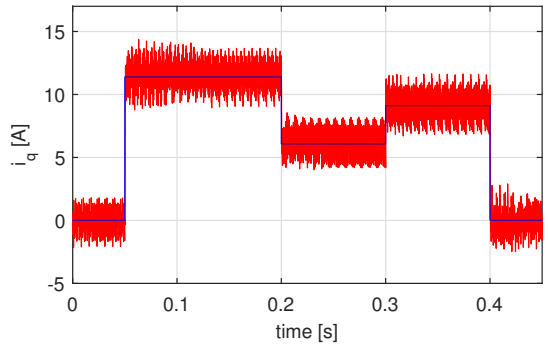
An objective comparison of the different controllers can be obtained by comparing their KPIs, which is the intention of the next section.



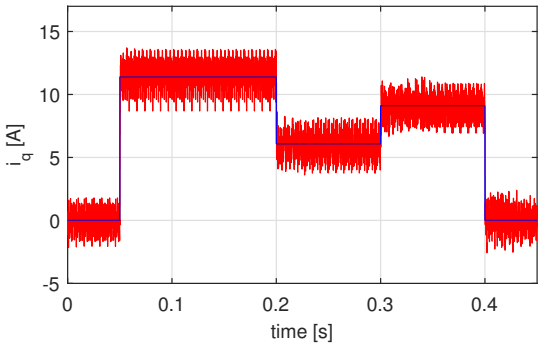
(a) PTC with quadratic cost function



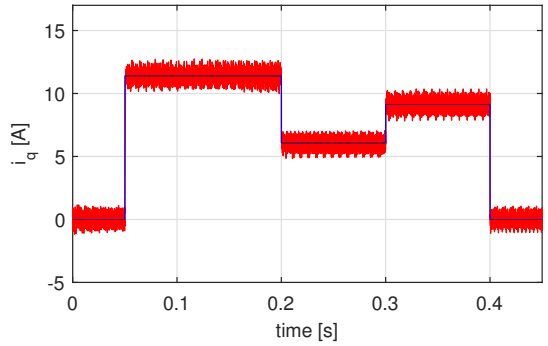
(b) PTC with cost function with tolerance bands



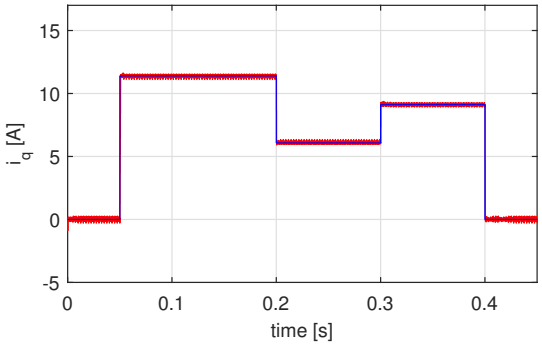
(c) PCC with quadratic cost function



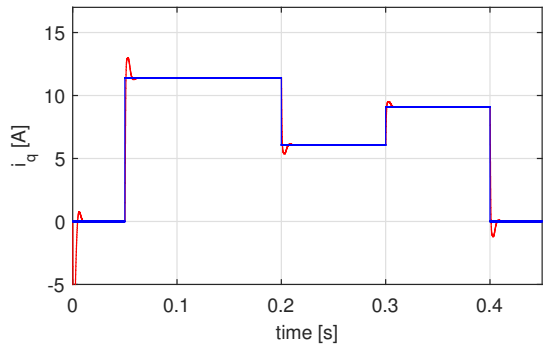
(d) PCC with cost function with tolerance bands



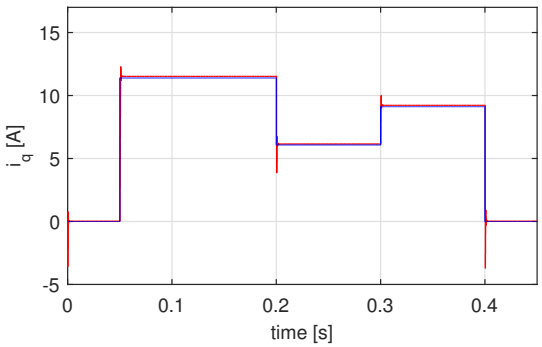
(e) PCC with double f_{u}



(f) PCC with DC calculation

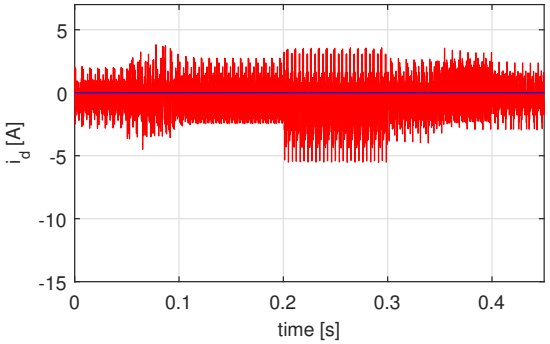


(g) PI

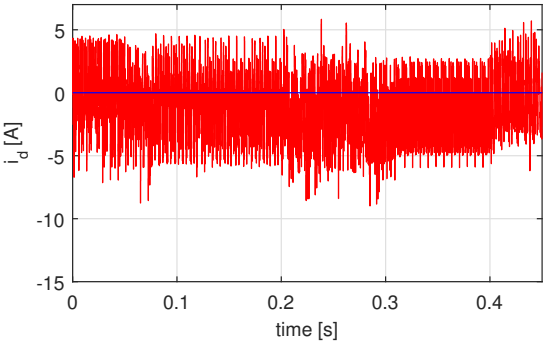


(h) DB

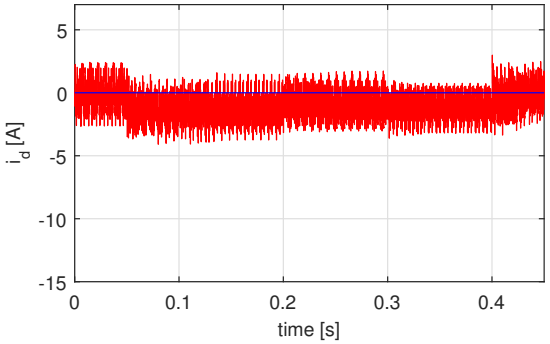
Figure 7.5: Simulation results for the control of i_q (blue: reference i_q^*; red: simulated i_q)



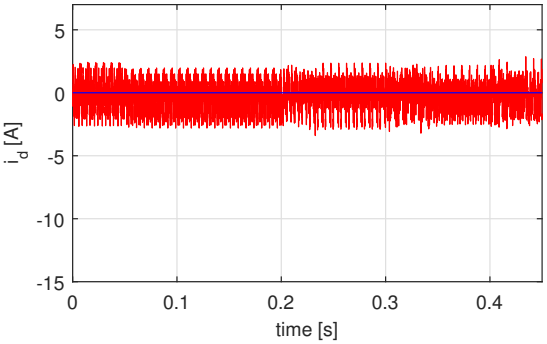
(a) PTC with quadratic cost function



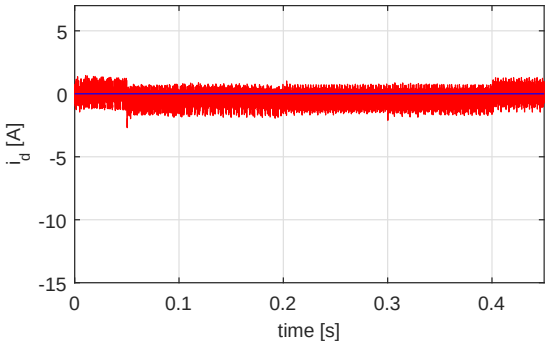
(b) PTC with cost function with tolerance bands



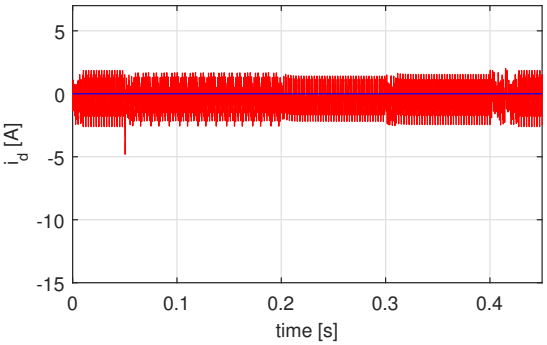
(c) PCC with quadratic cost function



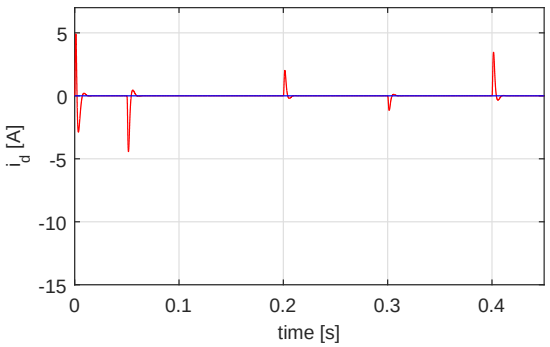
(d) PCC with cost function with tolerance bands



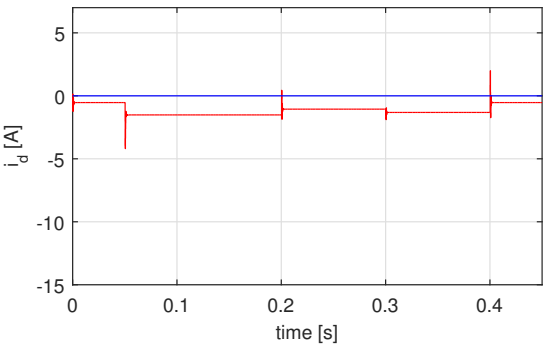
(e) PCC with double f_{u}



(f) PCC with DC calculation

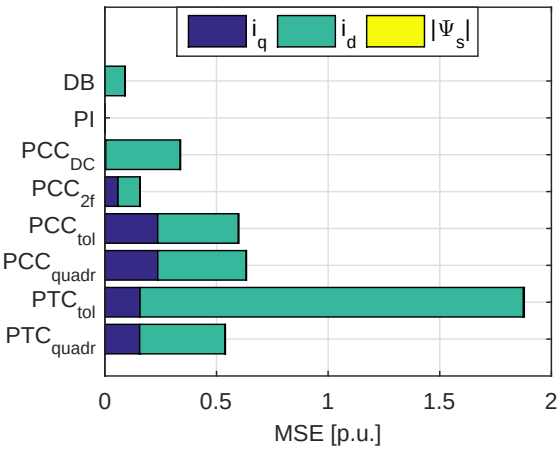


(g) PI

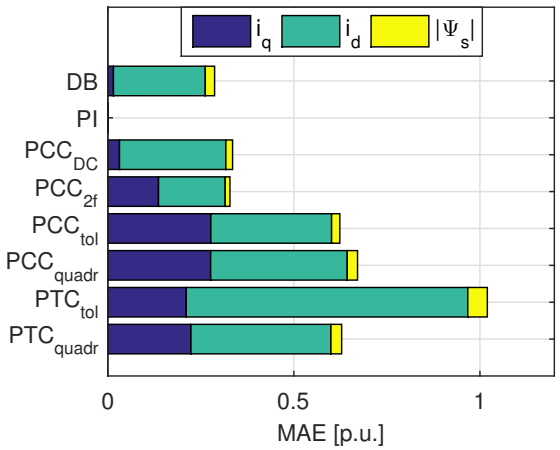


(h) DB

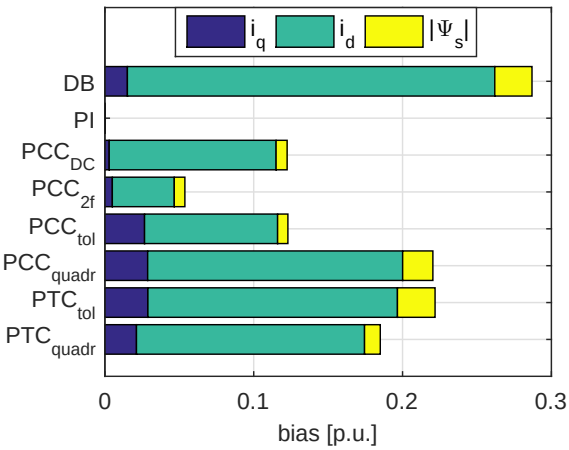
Figure 7.6: Simulation results for the control of i_d (blue: reference i_d^*; red: simulated i_d)


i_q
i_d
|Ψ_s|
DB
PI
PCC_DC
PCC_2f
PCC_tol
PCC_quadr
PTC_tol
PTC_quadr
0
0.5
1
1.5
2
MSE [p.u.]

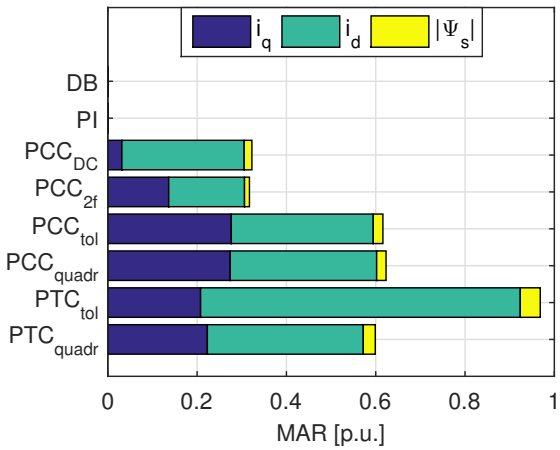
(a)


i_q
i_d
|Ψ_s|
DB
PI
PCC_DC
PCC_2f
PCC_tol
PCC_quadr
PTC_tol
PTC_quadr
0
0.5
1
MAE [p.u.]

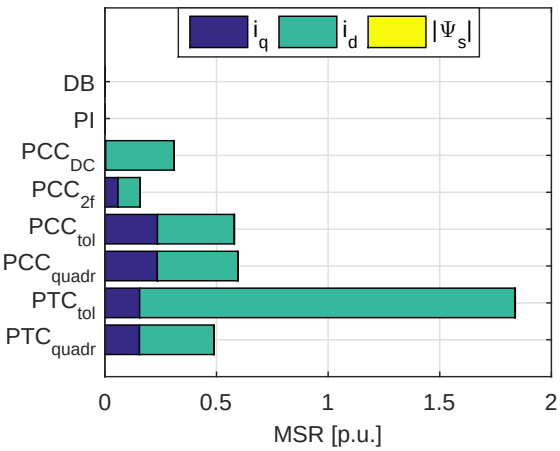
(b)


i_q
i_d
|Ψ_s|
DB
PI
PCC_DC
PCC_2f
PCC_tol
PCC_quadr
PTC_tol
PTC_quadr
0
0.1
0.2
0.3
bias [p.u.]

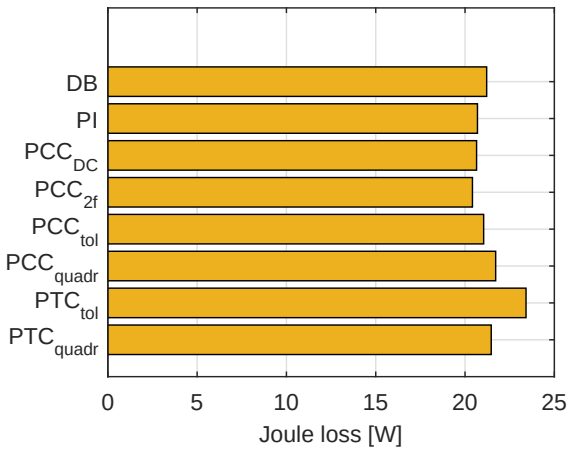
(c)


i_q
i_d
|Ψ_s|
DB
PI
PCC_DC
PCC_2f
PCC_tol
PCC_quadr
PTC_tol
PTC_quadr
0
0.2
0.4
0.6
0.8
1
MAR [p.u.]

(d)


i_q
i_d
|Ψ_s|
DB
PI
PCC_DC
PCC_2f
PCC_tol
PCC_quadr
PTC_tol
PTC_quadr
0
0.5
1
1.5
2
MSR [p.u.]

(e)


DB
PI
PCC_DC
PCC_2f
PCC_tol
PCC_quadr
PTC_tol
PTC_quadr
0
5
10
15
20
25
Joule loss [W]

(f)

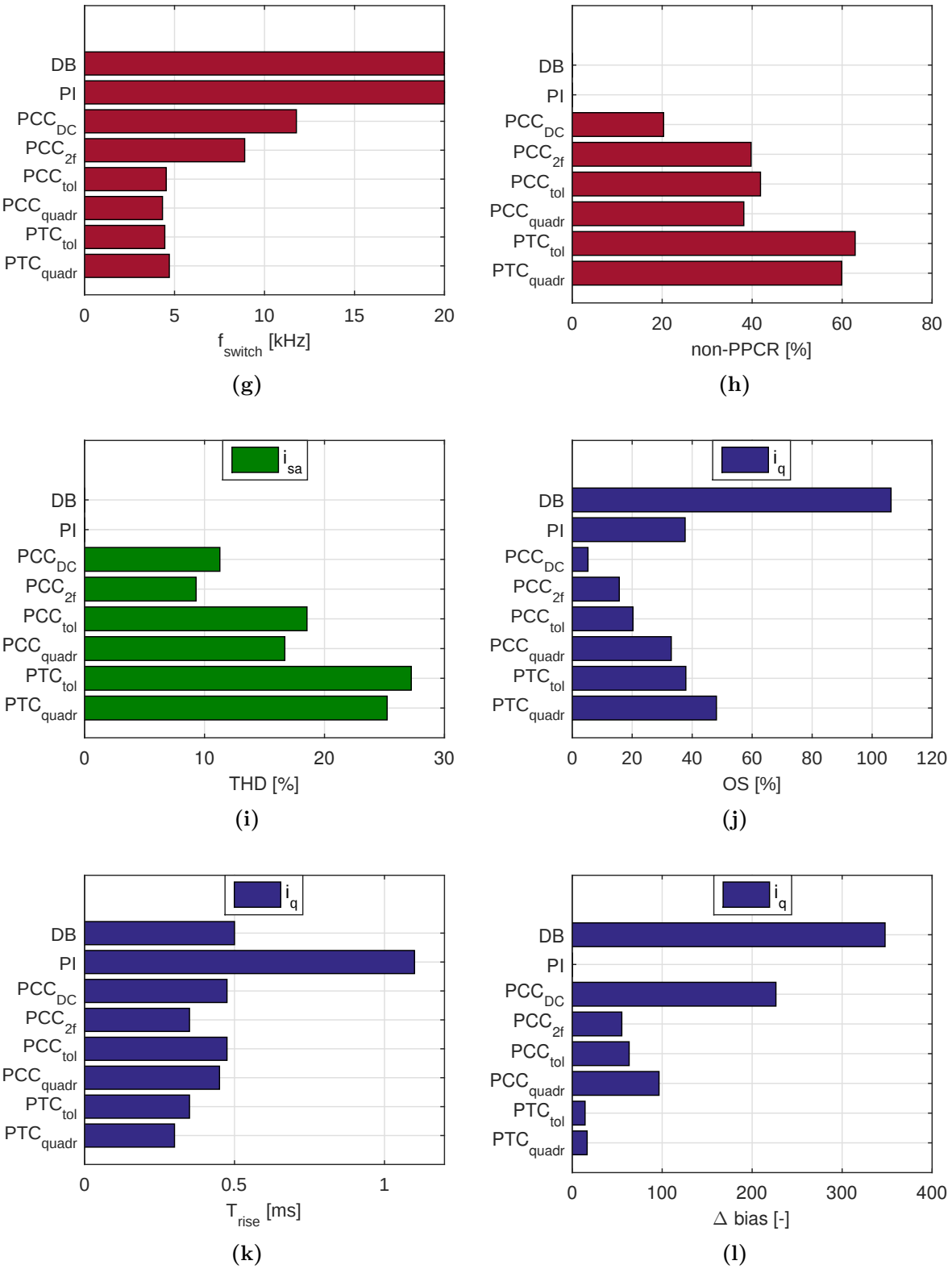


Figure 7.7: Overview of the KPIs of the simulations

7.4 Discussion of the Simulated KPIs

To facilitate the comparison of the controllers, their KPIs are summarized in Figure 7.7. The charts display values averaged over the different reference values for the torque; a more detailed overview of the KPIs can be found in the tables in Appendix A. The following sections are dedicated to a discussion concerning the differences in KPIs between the different controllers.

7.4.1 Control Quality

Figures 7.7a, 7.7b, 7.7c, 7.7d and 7.7e compare the KPIs assessing the control quality. A first observation is that the error on $|\Psi_s|$ is rather small compared to the error on i_q and especially i_d. The reason can be found in Eq. (5.3): a substantial part of $|\Psi_s|$ is determined by the fixed flux of the permanent magnets, keeping the instantaneous flux always relatively close to its reference value.

In general can be concluded that PTC_{tol} leads to the lowest control quality. This result is mainly caused by the not so strict control of i_d. Controlling the flux already permits some degree of freedom for i_d, a feature that becomes even more apparent when a small deviation of the flux itself is allowed as well. As to the control of i_q on the contrary, PTC actually outperforms PCC. The PI controller however outshines every predictive controller. The DB controller and PCC_{DC} are the best performing predictive controllers. The weakest point of the DB controller is its offset in i_q and especially in i_d. Concerning current ripple, it performs similar to PI control.

Another feature that can be noticed, is the fact that the type of cost function does not strongly affect the control quality of i_q. Its influence on i_d is more pronounced. For PTC, the tolerance bands worsen the quality of the i_d control, while for PCC the bands are narrow enough to improve the control of i_d. However, those KPIs are strongly related to the chosen design parameters for the cost function. A smaller $|\Psi_s|_l$, for instance, might have led to similar conclusions for both PTC and PCC.

Doubling the update frequency of PCC leads to significantly improved results concerning the control quality.

The Joule losses in Figure 7.7f are tightly connected to the current control. The large ripple in PTC_{tol} leads to the highest Joule losses, while the PI controller - with its small ripple and bias - results in the lowest Joule losses.

7.4.2 Voltage Quality

Figures 7.7g and 7.7h illustrate the voltage quality achieved by the controllers. The switching frequency of the DB and PI controller amounts to the twofold of the update frequency, since each switch changes its state from off to on, and then back to off during T_u. f_{switch} of PCC_{DC} is lower than 20 kHz, since not every switch needs to change its state during T_u. When, for instance, the active voltage vector corresponds to $S_k = [110]$ and the null vector to $S_{k,0} = [111]$, only the third switch needs to change its state from 1 to 0, and back to 1. The standard PTC and PCC lead to the lowest switching frequency, regardless of the type of cost function. Doubling f_u results in almost doubling f_{switch}.

On the other hand, the low f_{switch} of the standard PCC and PTC forces the switchings that do take place to violate the PPCR. Maintaining the same switch state during the whole update period leads to large deviations from the reference after all, causing the input voltage for the next T_u to be changed more abruptly. Halving T_u does not resolve this problem, since it does not

tackle the root cause of the issue, which is the fact that the amplitude of the voltage vector cannot be changed when applying standard PTC or PCC. Since PCC_{DC} does address this shortcoming, its percentage of switchings violating the PPCR is lower. Because the PI and DB controller allow to change not only the amplitude of the voltage vectors, but also their direction, almost all their switchings fulfill the PPCR.

7.4.3 Harmonic Distortion

In general, a higher f_{switch} reduces the harmonic distortion of the phase currents. This statement is endorsed by the decrease in THD of $\text{PCC}_{2\text{f}}$ compared to PCC with update frequency f_{u}, and the low THD of the PI and DB controller. Since PCC directly regulates both q- and d-current components, while for PTC the d-component is only controlled indirectly, PCC shows superior properties concerning harmonic distortion when compared to PTC.

7.4.4 Transient Behavior

An overview of the percentage overshoot and the rise time of i_q for the different controllers is presented in Figures 7.7j and 7.7k respectively. The first comment that must be made, is that the $\%OS$ measured for all the PTC and PCC controllers, is only due to the current ripple that is present in their static behavior as well. As could be seen in Figure 7.5, only the DB and the PI controller show some distinct overshoot peaks. From Figure 7.7j can be concluded that the DB controller displays a larger overshoot than the PI controller on average. However, the rise time of the PI controller lasts longer than the twofold of T_{rise} of the DB controller. Furthermore, it must be borne in mind that the PI controller can be tuned to have a lower T_{rise} as well, but at the expense of more overshoot and less stable behavior.

In general can be stated that the PTC and PCC controllers excel in both low $\%OS$ and a short T_{rise}.

7.4.5 Robustness

Especially the deadbeat controller is prone to instability in case the system model is not fully accurate, since the exact motor parameters are required to compute the optimal voltage vector. FS-MBPC on the contrary only makes use of the eight input vectors directly available at the output of the inverter, rendering this control strategy more robust. After all, a small deviation in the predicted state $\hat{x}_{k+2}$ due to parameter mismatch will only occasionally result in the selection of a different voltage vector. This effect is even reinforced by the use of tolerance bands in the cost function. Doubling f_{u} - and thus the sampling frequency - ameliorates the robustness a little, since the faster feedback allows the controller to intervene more quickly in case of errors. Since FS-MBPC with DC calculation combines the FS-MBPC and DB strategy, its stability features lie somewhere in between those two strategies. In [26] is pointed out that the calculated t_{opt} has a relative wide error range, since it is based on the DB principle. However, due to the fact that only the amplitude of the voltage vectors is varied by means of t_{opt}, but not their direction, this controller will be less sensitive to parameter mismatch than the DB controller. For the PI controller at last, the model parameters are only used to tune K_p and K_i, hence this strategy is assumed to be the most robust against parameter mismatch. The strategy of passing on wrong parameters to the controller in order to test its robustness cannot be applied however. In order to test its sensitivity to parameter inaccuracy, an alternative but similar working method is employed: the parameters in the simulation model of the 'real' machine are adapted. The average Δbias obtained in this way equals 0.33, which confirms that PI control

is indeed the least sensitive to parameter inaccuracy when compared to the Δbias values of the other types of controllers in Figure 7.7l.

7.4.6 PI Controller versus Predictive Controllers

In the previous discussion, it is very striking that the PI controller often outperforms the more advanced predictive controllers. Only the deadbeat controller can approach its quality of torque regulation. The other predictive controllers excel solely in terms of switching losses and dynamic performance. Consequently, the use of a predictive control algorithm to regulate an AFPMSM fed by a 2L-VSI seems a bit overkill. A similar conclusion was made in [7].

The major reason for this unexpected outcome, is the fact that the major strengths of the predictive controllers are not fully exploited. FS-MBPC, for instance, is particularly well-suited to deal with Multiple-Input Multiple-Output (MIMO) systems. In addition, this type of controller is well-known for its natural way of dealing with constraints. Thus, the situation might have been different if, for instance, flying-capacitor multi-level inverters would have been used, with two pairs of complementary switches in each inverter leg, and capacitors which can be put in series with the DC bus by means of the extra pair of switches. For this type of inverter, a larger variation of switch states is available than for the 2L-VSI, and regulation of the voltage over the capacitors is required as well. This situation would reflect the strengths of FS-MBPC much better. A similar conclusion holds for highly non-linear processes.

7.5 Conclusion

The different controllers introduced in the previous chapters were subjected to a profound comparison. For this purpose, some new KPIs were introduced, apart from the ones that were already in use in previous chapters:

- MAR and MSR, to give an indication of the torque and current ripple
- the bias, to quantify the systematic error between the controlled variables and their references
- THD, to indicate the harmonics in the stator current
- T_{rise} and $\%OS$, to examine the dynamic performance
- Δbias, to study the sensitivity to parameter inaccuracy

A comparison of all those KPIs led to the conclusion that the major disadvantages of the classical PTC and PCC controllers are their high torque and current ripple (especially for PTC with a cost function with tolerance bands), their high harmonic content in the stator current, and their poor performance concerning the fulfillment of the PPCR. However, these controllers excelled in their dynamic performance and low switching losses. The DB and PI controller on the contrary, showed outstanding simulation results for the same KPIs for which PTC and PCC performed not so well. However, those controllers led to high switching losses and exhibited distinct overshoot peaks. Moreover, it took the PI controller a relatively long time to bring the torque to its reference value. The major weaknesses of the DB controller were its systematic error in both current components and its parameter sensitivity. The other types of controllers showed mediocre performance. A more complete overview of the controller's ratings can be found in Table 7.4.

Table 7.4: Strengths and weaknesses of the controllers, based on simulations

	PTC	PCC	PCC_{2f}	PCC_{DC}	DB	PI
ripple	- -	-	+	+	++	++
bias	-	-	+	-	- -	++
Joule losses	-	+	+	+	+	+
switching losses	++	++	+	-	- -	- -
PPCR	- -	-	-	+	++	++
THD	- -	-	+	+	++	++
overshoot	++	++	++	++	- -	-
rise time	++	+	++	+	+	- -
robustness	+	+	+	-	- -	++

Finally it was concluded that the use of predictive control for an AFPMSM fed by a 2L-VSI is a slight overkill, since the major advantages of predictive controllers - like their appropriateness to deal with MIMO systems and systems constraints - are not fully exploited.

Chapter 8

Experimental Verification

8.1 Introduction

The major drawback of model predictive control is its high computational cost and its associated long calculation times. This used to make MBPC unattractive for processes with small time constants. The development of new and faster control hardware platforms, however, enabled the deployment of computationally demanding control schemes for fast real-time applications, even for processes with time constants in the millisecond range. Hence predictive control became a realistic candidate for application to electric drives.

An example of such a new hardware platform is a Field Programmable Gate Array (FPGA), an integrated circuit consisting of programmable logic components. In this chapter, the previously examined control algorithms will be implemented on-line by means of such an FPGA. After a short description of the test set-up, and an overview of the main characteristics of programming an FPGA, the features of the controllers will be verified by repeating the simulations of the previous chapter on the test set-up.

8.2 Test Set-up

To validate the performance of the controllers on the real AFPMSM, the set-up of Figure 8.1 is used. The variable DC source is fed from a 3-phase 400V grid. The inverter converts this rectified voltage in a sequence of voltage vectors, determined by the torque and current controllers. Those controllers are implemented on a Xilinx® Kintex®-7 XC7K325T FPGA, which has a clock period of 10 ns and is embedded in a MicroLabBox of dSPACE. The digital signal processor (DSP) of the latter is used to change the control parameters on-line. The MicroLabBox also provides the interface between the AFPMSM, its FPGA, its 2GHz dual-core real-time Freescale QorlQ P5020 processor and the computer. The FPGA is programmed using the Xilinx System Generator blockset (part of the Vivado® System Edition Design Suite) in a MATLAB®&Simulink® environment. Signals are visualized by means of the dSPACE ControlDesk® software.

The employed AFPMSM is a prototype developed in the EELab research group of Ghent University. Its specifications are presented in Table 4.1. An induction machine (IM) is connected to the AFPMSM as load. It is the IM that ensures the constant speed of the AFPMSM. The speed control of the IM was already implemented at the start of this master's dissertation by means of a DS1104 R&D Controller Board of dSPACE.

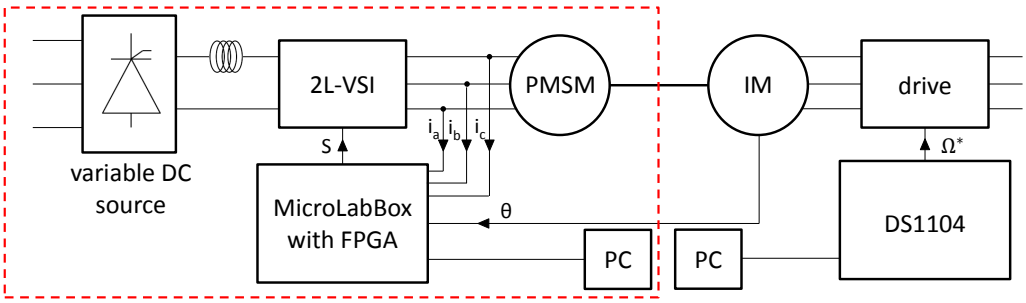


Figure 8.1: Experimental set-up

8.3 Field Programmable Gate Array

According to [21, 22, 27, 39], the main advantage FPGAs offer is the possibility of parallel processing. Since the algorithms are directly implemented into hardware (such as flip-flops), their parallelism can be maintained in their implementation, resulting in a short execution time. For FS-MBPC, for instance, this feature is used for parallel evaluation of the q- and d-variables, and for calculation of the different cost terms. The calculation time is reduced in this way. On the other hand, parallel processing requires special attention to the timing aspect. When the results of two parallel operations need to be processed together, for instance, it must be ensured that those two parallel operations last exactly the same amount of FPGA clock cycles. This condition can be fulfilled by adding delays to the fastest process, or by simultaneous enabling of the results of the two branches by means of triggers from a central counter. In case this requirement is not met, there is a real risk of processing the signal of the fastest parallel branch with a former result of the slowest branch, impeding a correct execution of the algorithm. Profound knowledge of the timing of signal flows in the model is thus a vital prerequisite for programming an FPGA.

Another key factor mentioned in [27] is the possibility of pipelining. Since paralleling the calculations for all switch states in the prediction step, for instance, is not possible due to the limited FPGA resources, and each switch state requires the evaluation of the same equations, pipelining is particularly well-suited for this case. At each clock cycle, a new switch state is fed into the pipeline. Since it takes 27 FPGA clock periods of 10 ns to go through all the calculations of the prediction step, the first prediction (according to the first switch state) will be available at the end of the pipeline after these 27 clock periods. The prediction for the second switch state will leave the pipeline one clock period later, and so on for the other six switch states, until the pipeline is empty. With this fully pipelined prediction block, all predictions are available after only $27 + 7 = 34$ clock periods. Without pipelining, the same calculations would require a computation time of $27 \cdot 8 = 216$ clock periods.

The major drawback of working with an FPGA is that it only allows to perform simple calculations. Addition and subtraction, multiplication of two signals or multiplication of one signal with a constant, and logical and bitwise operations pose no problems. Sines and cosines, division of two varying signals, and taking the square root of a signal belong to the possibilities as well, but require a great amount of resources and time steps compared to the other operations. Other functions need to be built by the programmers themselves by means of the operators that are available.

A fourth important aspect is the availability of a wide range of design tools. For this master's dissertation, for instance, the whole system is designed in a MATLAB®&Simulink® environment. The controllers can be graphically built up in this environment, and the underlying software generates the corresponding FPGA bit stream. Since the control algorithms to be implemented on the FPGA can be simulated beforehand in Simulink®, a first debugging of the controllers can

already be done off-line. In this way, the risk of applying an incorrect control to the real process is reduced. The only drawback is the fact that the controllers must be simulated with small time steps, determined by the clock period of the FPGA. Since this period equals 10 ns, those simulations are very costly in terms of processing time and required memory of the computer.

Moreover, the Simulink® environment automatically encourages a modular build up of the algorithms. For FS-MBPC, for instance, separate subsystems can be implemented for the measurements, the estimation, the prediction, the optimization and the generation of the inverter switch states. Signal latency is provided in each subsystem, and each subsystem is enabled separately by means of triggers from a central counter, greatly simplifying a correct timing of the operations. In this way, the subsystems are decoupled and reusable. Extending the prediction horizon, for instance, can easily be done by reusing the original prediction block.

Another difference between an FPGA implementation and a processor implementation, is that for each fixed-point arithmetic operation, the bit precision can be adapted manually for an FPGA. Depending on the magnitude of the signals the block works with and the required accuracy, its fixed point position and the number of available bits can be differed, enabling efficient use of the available hardware resources and short computation times. On the other hand, this characteristic requires knowledge of the order of magnitude of each considered signal, in order to avoid overflow errors. Round-off errors are another inevitably drawback of the limited number of available bits. Thorough research is done into these two type of errors in [40]. However, this analysis is out of the scope of this master's dissertation.

Due to all this properties, FPGAs are particularly well-suited to implement predictive controllers, which is confirmed by numerous articles, such as [39, 41].

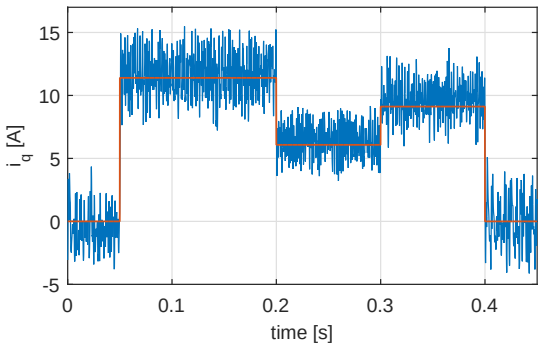
8.4 Measurements

Since measurement noise, the utilization of a simplified model, and parameter uncertainty (or variability under operation) can degrade the performance of the control systems and even affect their stability, the controllers are tested in real-life as well. The same tests as in Section 7.3 are carried out on the experimental set-up. Plots of the measurements for i_q and i_d are assembled in Figures 8.2 and 8.3 respectively. All controllers are able to track the reference value for i_q, and thus for the torque. The difference in current ripple between the various controllers and the fact that only the PI controller is able to keep i_d equal to zero properly, catch the eye as well.

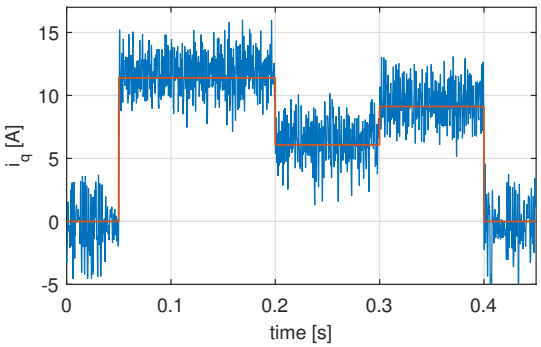
The fact that i_d is only properly controlled by the PI controller, can be explained by a similar argument as for the simulations: the large back EMF term in Eq. (5.2), involving the large factors ω and i_q, tends to pull i_d away from zero; the larger i_q, the larger this effect. Due to measurement noise and inaccuracies in the machine model, this effect is more pronounced in the measurements than in the simulations, and only noticed for the predictive controllers. An underestimation of L_q, for instance, causes i_d to be more negative, since the back EMF is underestimated in this case.

Concerning inaccuracies in the machine model, it can be remarked as well that the waveforms of the PI controller in Figures 8.2g and 8.3g are less damped than in simulation. Hence, it can be concluded that the actual machine exhibits less damping than inferred by the model. The larger bias in Figures 8.2f and 8.2h can be explained in a similar way.

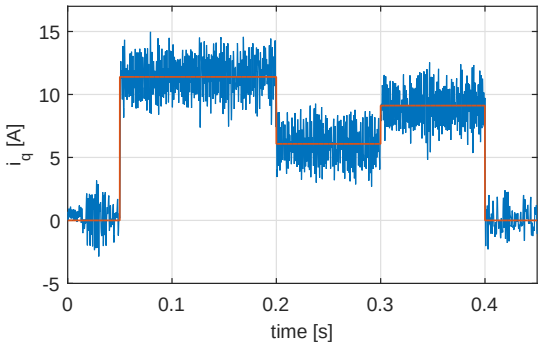
Compared to the simulation results in Figures 7.5 and 7.6, the measurements look less 'clean', which can also be ascribed to measurement noise and parasitic effects in the real machine that were not included in the machine model.



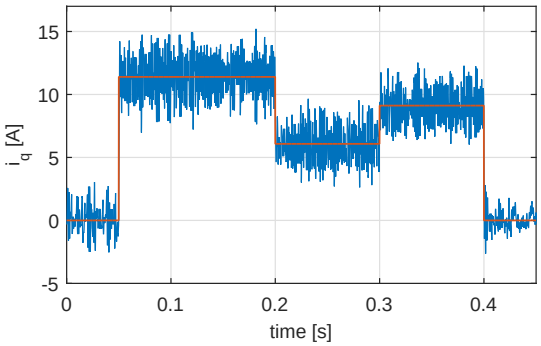
(a) PTC with quadratic cost function



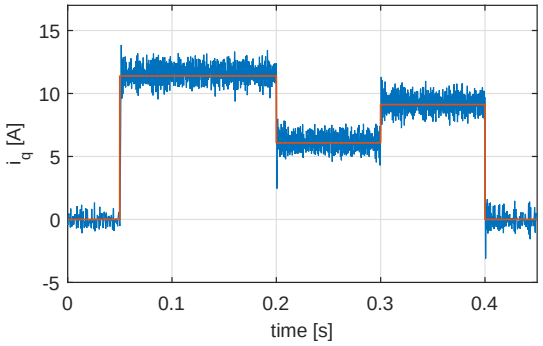
(b) PTC with cost function with tolerance bands



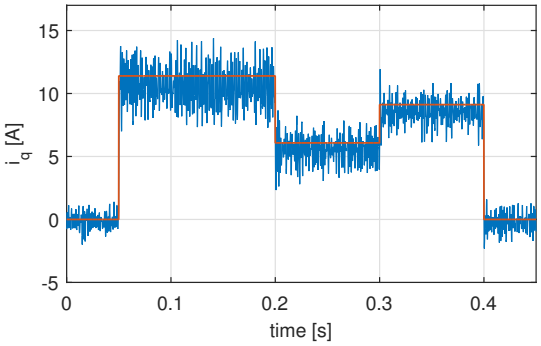
(c) PCC with quadratic cost function



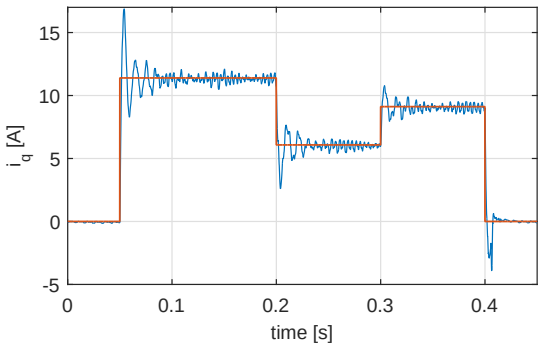
(d) PCC with cost function with tolerance bands



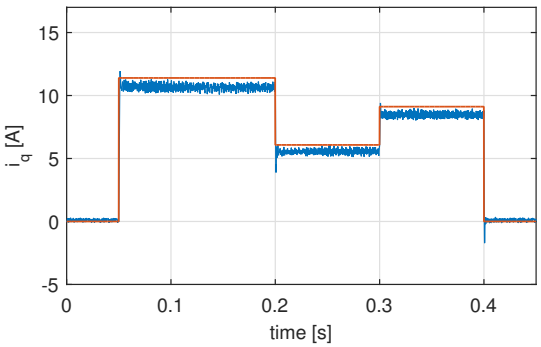
(e) PCC with double f_{u}



(f) PCC with DC calculation

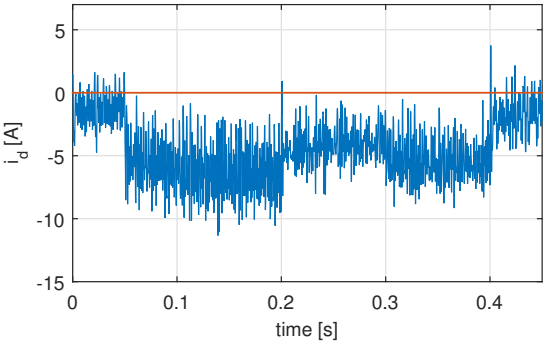


(g) PI

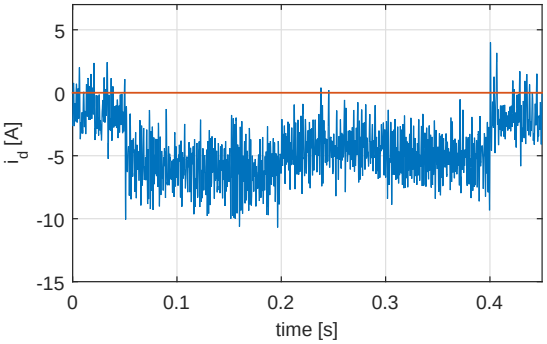


(h) DB

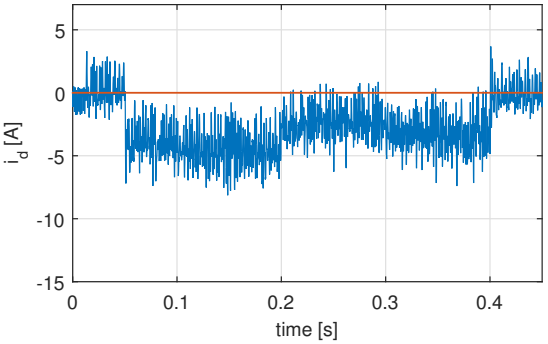
Figure 8.2: Experimental validation of the control of i_q (red: reference i_q^*; blue: measured i_q)



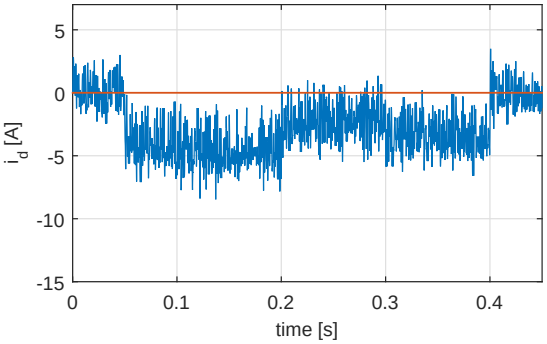
(a) PTC with quadratic cost function



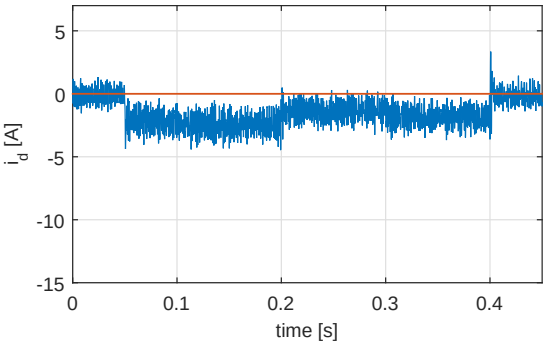
(b) PTC with cost function with tolerance bands



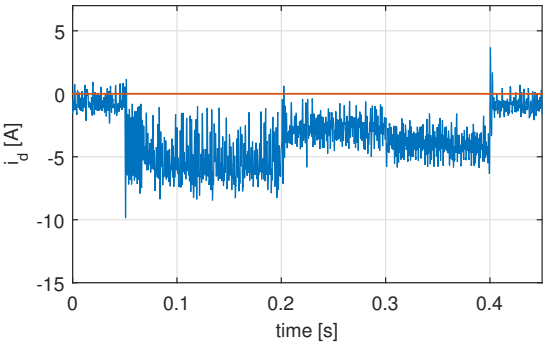
(c) PCC with quadratic cost function



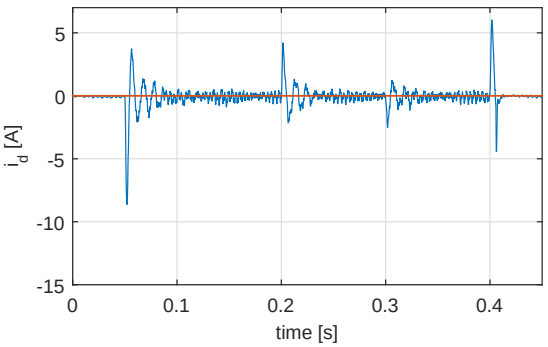
(d) PCC with cost function with tolerance bands



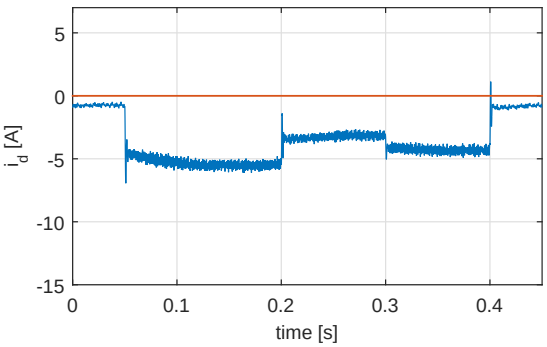
(e) PCC with double f_{u}



(f) PCC with DC calculation

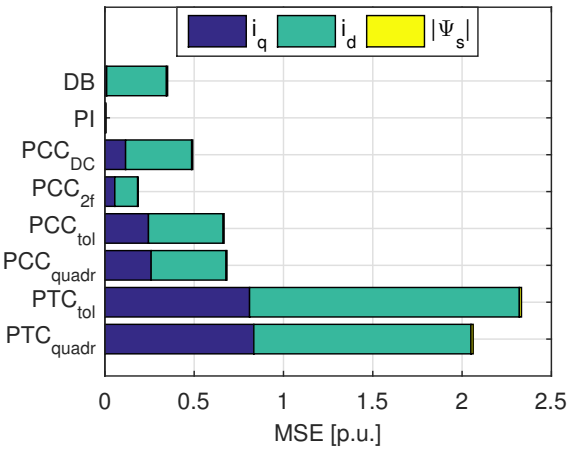


(g) PI

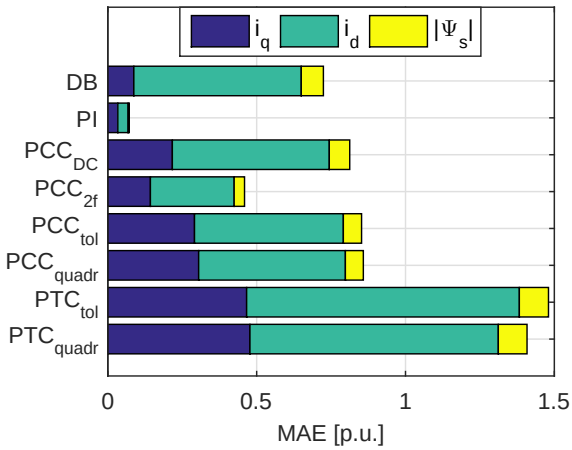


(h) DB

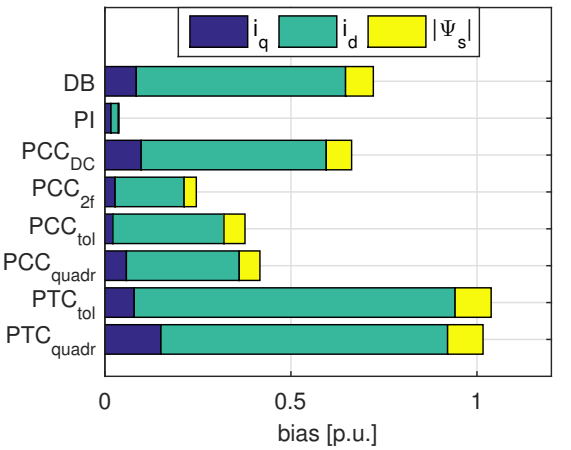
Figure 8.3: Experimental validation of the control of i_d (red: reference i_d^*; blue: measured i_d)


i_q
i_d
|Ψ_s|
DB
PI
PCC_DC
PCC_2f
PCC_tol
PCC_quadr
PTC_tol
PTC_quadr
0
0.5
1
1.5
2
2.5
MSE [p.u.]

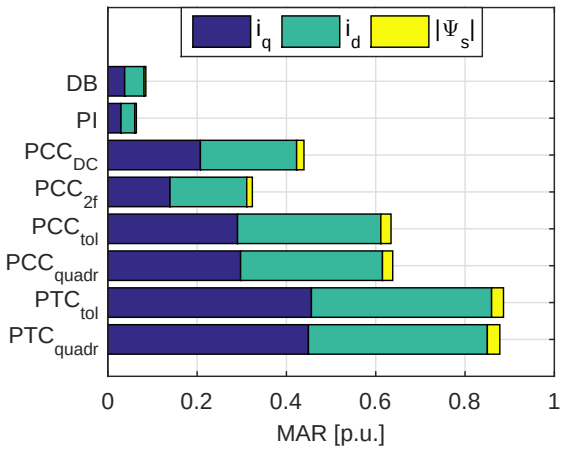
(a)


i_q
i_d
|Ψ_s|
DB
PI
PCC_DC
PCC_2f
PCC_tol
PCC_quadr
PTC_tol
PTC_quadr
0
0.5
1
1.5
MAE [p.u.]

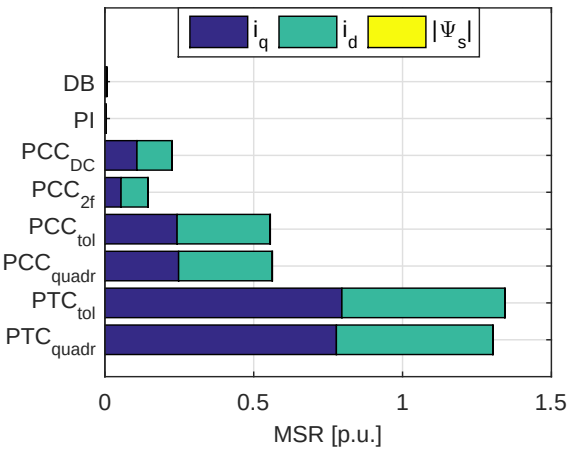
(b)


i_q
i_d
|Ψ_s|
DB
PI
PCC_DC
PCC_2f
PCC_tol
PCC_quadr
PTC_tol
PTC_quadr
0
0.5
1
bias [p.u.]

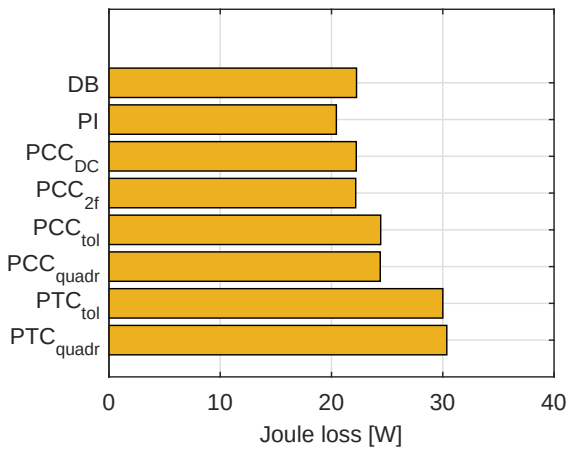
(c)


i_q
i_d
|Ψ_s|
DB
PI
PCC_DC
PCC_2f
PCC_tol
PCC_quadr
PTC_tol
PTC_quadr
0
0.2
0.4
0.6
0.8
1
MAR [p.u.]

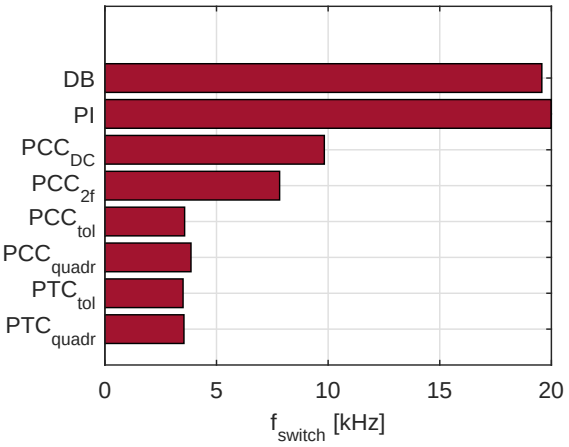
(d)


i_q
i_d
|Ψ_s|
DB
PI
PCC_DC
PCC_2f
PCC_tol
PCC_quadr
PTC_tol
PTC_quadr
0
0.5
1
1.5
MSR [p.u.]

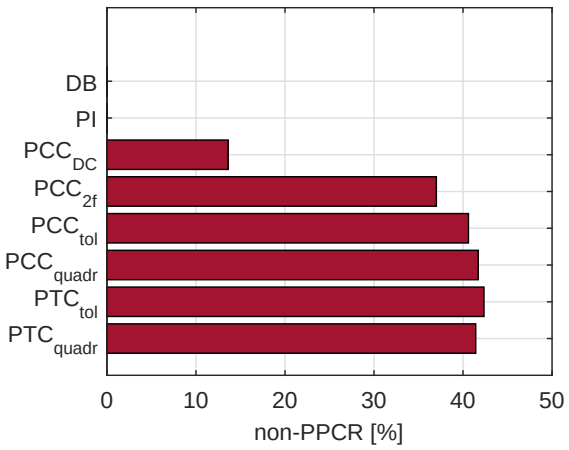
(e)


DB
PI
PCC_DC
PCC_2f
PCC_tol
PCC_quadr
PTC_tol
PTC_quadr
0
10
20
30
40
Joule loss [W]

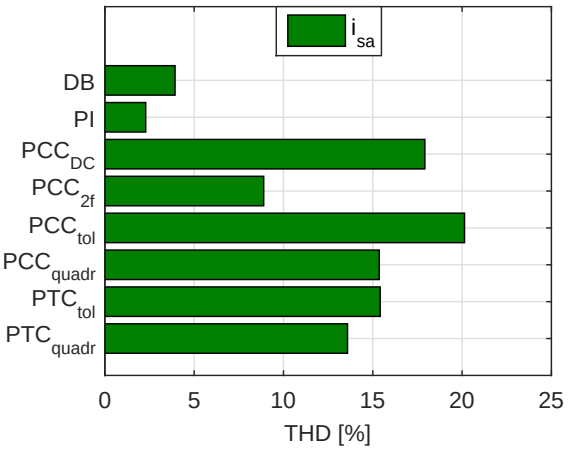
(f)



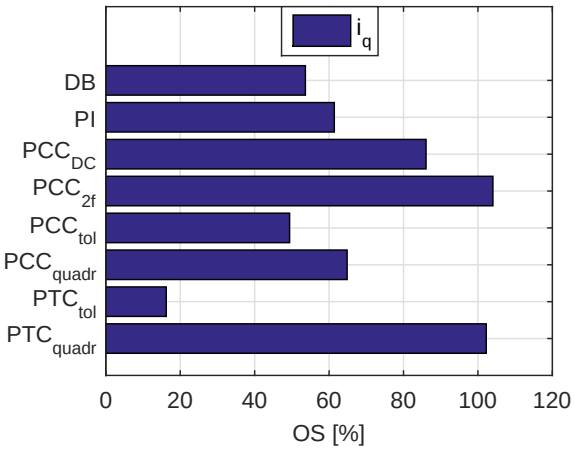
(g)



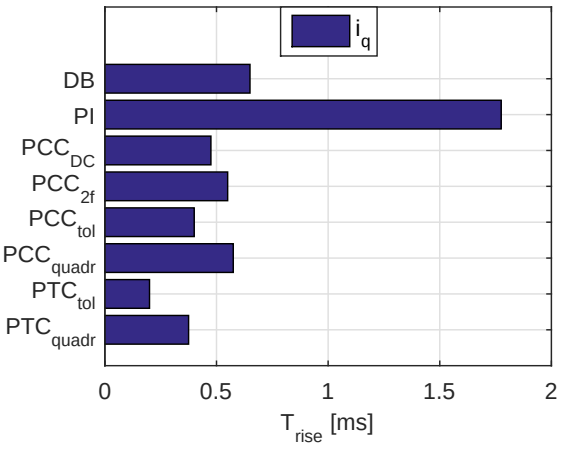
(h)



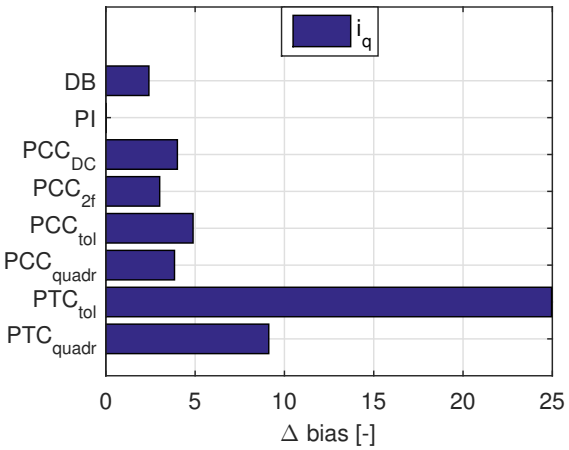
(i)



(j)



(k)



(l)

Figure 8.4: Overview of the KPIs of the measurements

8.5 Discussion of the Measured KPIs

To enable an easy comparison of the controllers, their KPIs are assembled in Figure 8.4. The values displayed in the charts are averaged over the different T_{em}^*; separate results can be found in the tables in Appendix B. In this discussion, the focus lies on the differences between the measurements on the one hand, and the simulation results of Section 7.4 on the other hand.

8.5.1 Control Quality

Compared to the simulation results of Section 7.4, the ripple on i_q has increased due to measurement noise, as have the Joule losses. For all controllers but the PI controller, the static error on i_d has seriously deteriorated. Especially $\text{PTC}_{\text{quadr}}$ performed worse than expected, on all fronts of the control quality. As to its bias, it even surpassed the DB controller and PCC_{DC}. Its high current ripple generated an unacceptable level of acoustic noise.

The performance of PCC_{DC} is slightly disappointing as well. The average ripple might be reduced compared to the standard PCC, but the maximal ripple amplitude is still high. Furthermore, i_q displays a large bias.

In general, PI control still shows superior control quality, followed closely by DB control.

8.5.2 Voltage Quality

The switching frequencies themselves have slightly changed compared to the simulation results, but the general conclusion concerning the switching losses is still valid: the DB and PI controller cause the highest switching losses, while the classic PTC and PCC perform best for this KPI. The cost function with tolerance bands slightly decreased f_{switch} for both PTC and PCC compared to the quadratic cost function, but the difference is hardly noticeable. Qua violations of the PPCR, PTC has achieved a better result than was anticipated: while the simulations predicted that this controller would have a higher percentage of switchings not fulfilling the PPCR, it actually performed similar to PCC.

8.5.3 Harmonic Distortion

Due to measurement noise and non-linearities in the behavior of the real inverter, the stator currents obtained by DB and PI control also contain some harmonics in reality. The harmonic distortion of PCC_{DC} is worse than in simulation, due to its deterioration in control quality. The statement that using tolerance bands in the cost function leads to an increase in harmonic distortion has been endorsed by the measurements. The fact that PTC shows less harmonics in its stator current than PCC is however very surprising.

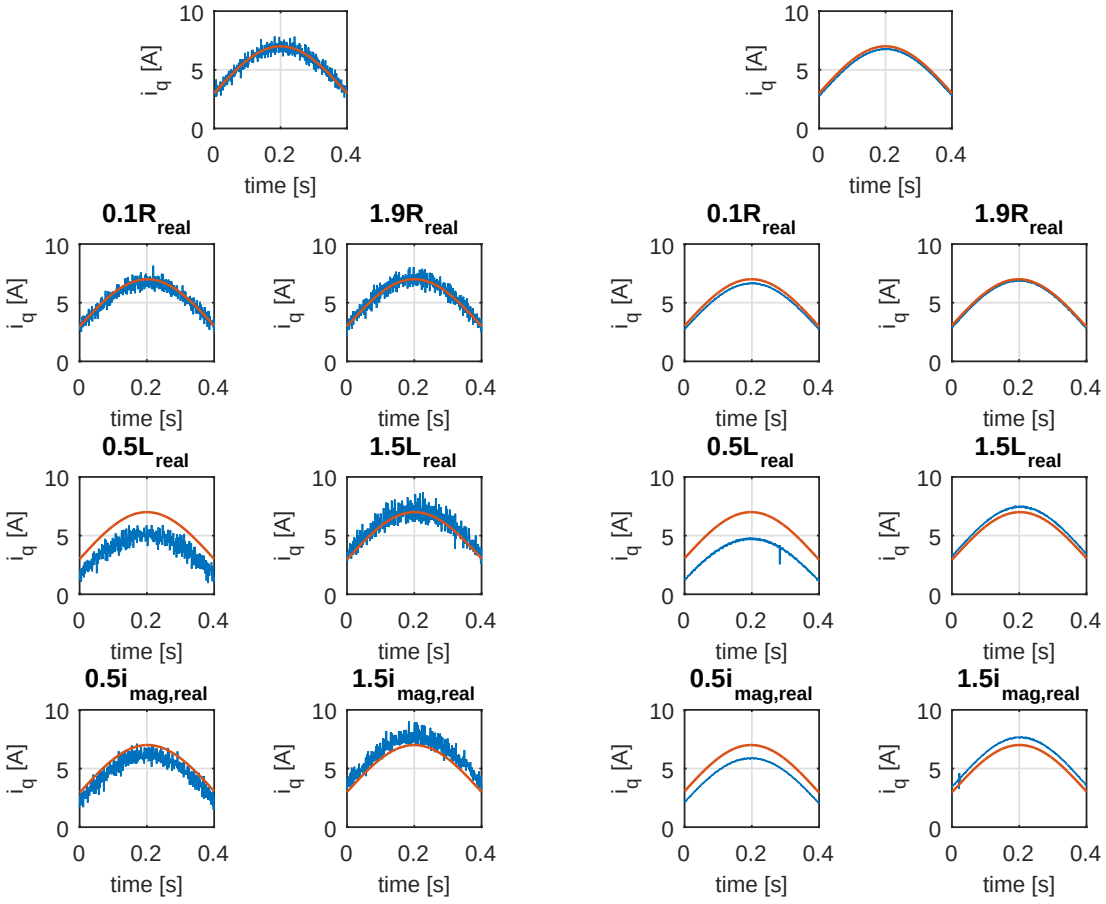
8.5.4 Transient Behavior

As opposed to what was predicted by the simulation results, the DB controller shows less overshoot in real-life than the PI controller. The so-called overshoot of the other controllers is actually only due to their current ripple. In Figure 8.2 can indeed be observed that only the DB and the PI controller show distinct overshoot peaks when applying a step in the reference value.

Concerning the rise time, the PI controller is a bit slower than anticipated. The measurements validate that the shortest rise time is achieved by using PTC.

8.5.5 Robustness

While PTC showed to be the most robust against parameter inaccuracy among the predictive controllers in simulation, and the DB controller the least, the measurement results claim exactly the opposite. However, these results need to be nuanced a little bit by bearing in mind that Δbias represents the relative change in bias due to passing on inaccurate model parameters to the controllers, and not the absolute change. The difference in robustness between, for instance, PTC_{tol} and DB is much harder to notice when comparing the plots in Figure 8.5. Similar plots for the other types of controllers can be found in Appendix B.



(a) PTC_{tol} **(b)** DB

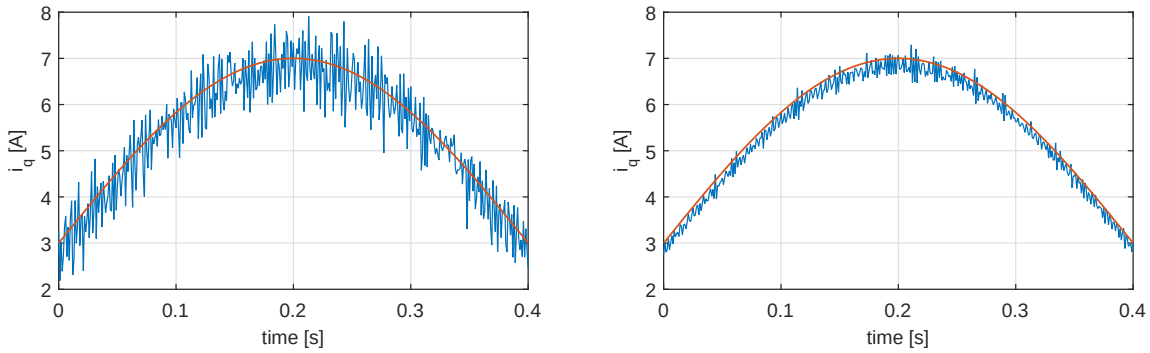
Figure 8.5: Parameter sensitivity (red: reference i_q^*; blue: measured i_q)

8.5.6 Measurements versus Simulations

In general, the main features of the controllers that showed up in the simulations are preserved in the experiments as well. The classical PTC and PCC controllers are still characterized by their high torque and current ripple, their high harmonic content in the stator current, and their high percentage of switchings violating the PPCR, while the PI and DB controller excel in those same properties. On the contrary, the low switching losses of PTC and PCC are in favor of these two control strategies, as well as their superior dynamic performance.

The most striking difference between the measurements and the simulations is that it is not DB control that performs worse concerning bias and parameter sensitivity, but PTC. Furthermore,

the real-life performance of PCC_{DC} qua decrease in current ripple is slightly disappointing under the tested circumstances ($V_{\text{dc}} = 250\text{V}$, $N = 1000$ rpm). However, under the operation circumstances of the robustness test ($V_{\text{dc}} = 100\text{V}$, $N = 300$ rpm), the improvement in the current ripple is much more pronounced, as is illustrated in Figure 8.6. Therefore the effect of the speed and the DC bus voltage should have been taken into account for a more complete comparison.



(a) $\text{PCC}_{\text{quadr}}$ **(b)** PCC_{DC}

Figure 8.6: Difference in torque ripple for $V_{\text{dc}} = 100\text{V}$ and $N = 300\text{rpm}$ (red: reference i_q^*; blue: measured i_q)

However, based on the measurements that were performed at 250V and 1000 rpm, a table similar to Table 7.4 can be constructed, summarizing the relative ratings for the controllers for different features. The ratings that were changed compared to Table 7.4 received a green background when they were upgraded, and a red background when they were downgraded.

Table 8.1: Strengths and weaknesses of the controllers, based on experiments

	PTC	PCC	PCC_{2f}	PCC_{DC}	DB	PI
ripple	- -	-	+	-	++	++
bias	- -	+	+	-	-	++
Joule losses	- -	-	+	+	+	+
switching losses	++	++	+	-	- -	- -
PPCR	- -	- -	- -	+	++	++
THD	-	- -	+	- -	++	++
overshoot	++	++	++	++	-	- -
rise time	++	+	+	+	+	- -
robustness	- -	+	+	+	+	++

8.6 Conclusion

The simulations performed in Chapter 7 were repeated on a test set-up, in order to test the performance of the controllers when exposed to measurement noise and the actual AFPMSM instead of its simplified model. The control algorithms were implemented by means of an FPGA. Short calculation times were obtained by exploiting the major strengths of the FPGA, more specifically paralleling and pipelining. Experimental results confirmed that all controllers were able to track their reference value. In general, they exhibited similar features as in simulation.

Chapter 9

Conclusion and Future Work

9.1 General Conclusion

The recent focus on ecodesign led to the development of new electrical machine topologies. The AFPMSM with a YASA topology and surface mounted permanent magnets used throughout this thesis, is a typical example. In Chapter 2 a brief description of its main features was given. Compared to other electrical machines, the AFPMSM excels in its outstanding efficiency, power density and compactness.

The major goal of this master's dissertation was to regulate the rotor torque of an AFPMSM (fed by a 2L-VSI) by means of predictive control. The task of the controllers was to select the optimal sequence of switch states for the inverter, in order to achieve a given torque demand. Two prominent predictive control families were closely examined for this purpose: FS-MBPC and DB control. In Chapter 3, the three basic steps of FS-MBPC were explained: estimation, prediction and optimization. This control algorithm directly outputted the switch states to be sent to the inverter. DB control on the contrary, consisted of only two steps - estimation and deadbeat - and determined only a reference for the input voltage of the machine. A separate modulation algorithm was thus required to obtain the necessary switch states for the inverter.

Since an adequate system model is an absolute must for a proper predictive control, Chapter 4 was devoted to the modeling of the AFPMSM and the 2L-VSI. A compromise between high accuracy and low computational burden had to be made. Different types of modulation techniques were studied as well.

In Chapter 5, the electromagnetic torque and stator flux of the AFPMSM were controlled directly by implementing the FS-MBPC algorithm in a MATLAB®&Simulink® environment. Two different types of cost function were proposed for the optimization step: a quadratic cost function, resulting in a typical tracking control, and a cost function with tolerance bands, allowing the controlled variables to deviate from their reference values within acceptable limits. Some attention was devoted to the tuning of the width of these tolerance bands, as well as to the weighting factors expressing the relative importance of the control goals. A set of KPIs was introduced to compare not only the differences in control quality, but also the variation in switching losses, Joule losses and voltage quality.

It was concluded, however, that the direct link between torque and stator flux made the control more difficult. Therefore, Chapter 6 examined how the torque could be regulated by applying the FOC principle. Four different implementations were discussed: a simple PI control, FS-MBPC, DB control and a hybrid version of the two latter ones. Simulation results revealed that FS-MBPC led to quite high current ripple compared to PI and DB control. However,

FS-MBPC had the advantage of a large flexibility to express secondary control goals by means of an adaptation of its cost function. Combining these two advantages led to the emergence of a hybrid version of FS-MBPC and DB control: FS-MBPC with DC calculation.

Chapter 7 assembled a selection of the previously introduced controllers and subjected their simulation results to a profound comparison. New KPIs were determined for this purpose, assessing the harmonic distortion in the stator current, their dynamic performance, and their robustness against parameter mismatch as well. The controller's ratings were summarized in Table 7.4. It was concluded that the use of predictive control for an AFPMSM fed by a 2L-VSI was a slight overkill, since the major advantages of predictive controllers - like their appropriateness to deal with MIMO systems and system constraints - were not fully exploited. Out of all the evaluated predictive controllers, DB control performed best, in spite of its high switching losses compared to FS-MBPC.

Finally, the performance of the controllers was experimentally verified on a test set-up, by implementing the algorithms on an FPGA. The measurements provided in Chapter 8 confirmed that all controllers were able to track their reference value. Barring a few exceptions, the experimental results revealed similar features as the simulations.

9.2 Future Work

Although the major part of the controllers tested in simulation were applied on the real AFPMSM, not all their variations have been experimentally verified yet. The effect of extending the prediction horizon, for instance, or adding an extra term to the cost function in order to reduce the switching losses and to improve the voltage quality, still has to be validated on the test set-up.

The algorithms that were implemented on the FPGA still need some optimization as well. In [21, 22, 27] is mentioned, for instance, that the same calculation core can be used for both the estimation and the prediction step by means of multiplexing, as the same system model is used for both steps. This would considerably reduce the utilized number of FPGA resources.

In order to reduce the bias noticed in the experimental results, it might be advisable to test the effect of the I-action mentioned in [35] as well.

Finally, in Chapter 8 was remarked that the characteristics of the controllers could vary with the operation point. To examine this influence, extra measurements should be performed. Besides, a statistical analysis would be useful to demonstrate the significance of the perceived effects.

Bibliography

[1] H. Vansompel, *Design of an Energy Efficient Axial Flux Permanent Magnet Machine.* PhD thesis, Ghent University, 2013.

[2] T. Vyncke, *Voorspellende regelaars voor directe koppelcontrole van draaiveldmachines.* PhD thesis, Ghent University, 2012.

[3] R. De Keyser and C. Ionescu, "Computer control of industrial processes." Ghent: Course notes, Ghent University, SYSTeMS, 2016.

[4] T. J. Vyncke, S. Thielemans, T. Dierickx, R. Dewitte, M. Jacxsens, and J. A. Melkebeek, "Design choices for the prediction and optimization stage of finite-set model based predictive control," in *Proc. Workshop Predictive Control of Electrical Drives and Power Electronics*, pp. 47–54, Oct. 2011.

[5] F. Morel, J. M. Retif, X. Lin-Shi, and C. Valentin, "Permanent magnet synchronous machine hybrid torque control," *IEEE Transactions on Industrial Electronics*, vol. 55, pp. 501–511, Feb. 2008.

[6] E. J. Fuentes, J. Rodriguez, C. Silva, S. Diaz, and D. E. Quevedo, "Speed control of a permanent magnet synchronous motor using predictive current control," in *Power Electronics and Motion Control Conference, 2009. IPEMC'09. IEEE 6th International*, pp. 390–395, IEEE, 2009.

[7] J. Druant, "Modelgebaseerde voorspellende controle van multi-level invertoren met geintegreerde foutdetectie," Master's thesis, Ghent University, EESA, EELAB, June 2013.

[8] G. Papafotiou, J. Kley, K. G. Papadopoulos, P. Bohren, and M. Morari, "Model predictive direct torque control —part ii: Implementation and experimental evaluation," *IEEE Transactions on Industrial Electronics*, vol. 56, pp. 1906–1915, June 2009.

[9] S. Kouro, P. Cortés, R. Vargas, U. Ammann, and J. Rodríguez, "Model predictive control - a simple and powerful method to control power converters," *IEEE Transactions on Industrial Electronics*, vol. 56, no. 6, pp. 1826–1838, 2009.

[10] J. Melkebeek, "Dynamics of electrical machines and drives." Ghent: Course notes, Ghent University, EESA, EELAB, 2016.

[11] G. Crevecoeur, "Modelling and simulation of dynamical systems." Ghent: Course notes, Ghent University, 2016.

[12] J. Melkebeek, "Elektrische aandrijftechniek." Ghent: Course notes, Ghent University, EESA, EELAB, 2015.

[13] J. Melkebeek, "Controlled electrical drives." Ghent: Course notes, Ghent University, EESA, EELAB, 2015.

[14] G. S. Buja and M. P. Kazmierkowski, "Direct torque control of PWM inverter-fed AC motors - a survey," *IEEE Transactions on Industrial Electronics*, vol. 51, pp. 744–757, Aug. 2004.

[15] T. Vyncke, R. Boel, and J. Melkebeek, "Direct torque control of permanent magnet synchronous motors–an overview," in *Proceedings of the 3rd IEEE Benelux Young Researchers Symposium in Electrical Power Engineering*, UGent-EESA, 2006.

[16] Y. Zhang, H. Yang, and B. Xia, "Model predictive torque control of induction motor drives with reduced torque ripple," *IET Electric Power Applications*, vol. 9, no. 9, pp. 595–604, 2015.

[17] C. A. Rojas, J. Rodriguez, F. Villarroel, J. R. Espinoza, C. A. Silva, and M. Trincado, "Predictive torque and flux control without weighting factors," *IEEE Transactions on Industrial Electronics*, vol. 60, no. 2, pp. 681–690, 2013.

[18] Y. Cho, K.-B. Lee, J.-H. Song, and Y. I. Lee, "Torque-ripple minimization and fast dynamic scheme for torque predictive control of permanent-magnet synchronous motors," *IEEE Transactions on Power Electronics*, vol. 30, no. 4, pp. 2182–2190, 2015.

[19] J.-W. Kang and S.-K. Sul, "Analysis and prediction of inverter switching frequency in direct torque control of induction machine based on hysteresis bands and machine parameters," *IEEE Transactions on industrial electronics*, vol. 48, no. 3, pp. 545–553, 2001.

[20] T. Geyer, G. Papafotiou, and M. Morari, "Model predictive direct torque control —part i: Concept, algorithm, and analysis," *IEEE Transactions on Industrial Electronics*, vol. 56, no. 6, pp. 1894–1905, 2009.

[21] E. De Brabandere, "Model predictive control of an electric variable transmission for hybrid electric vehicles," Master's thesis, Ghent University, EESA, EELAB, June 2015.

[22] T. J. Vyncke, S. Thielemans, and J. A. Melkebeek, "Finite-set model-based predictive control for flying-capacitor converters: Cost function design and efficient FPGA implementation," *IEEE Transactions on Industrial Informatics*, vol. 9, pp. 1113–1121, May 2013.

[23] S. Thielemans, T. Vyncke, and J. Melkebeek, "Voltage quality analysis of a three-level flying capacitor inverter with model based predictive control," in *Power Electronics and ECCE Asia (ICPE & ECCE), 2011 IEEE 8th International Conference on*, pp. 124–131, IEEE, 2011.

[24] R. De Keyser, "Modelleren en regelen van dynamische systemen." Ghent: Course notes, Ghent University, SYSTeMS, 2015.

[25] T. Vyncke, S. Thielemans, M. Jacxsens, and J. Melkebeek, "Analysis of some design choices in model based predictive control of flying-capacitor inverters," *COMPEL-The international journal for computation and mathematics in electrical and electronic engineering*, vol. 31, no. 2, pp. 619–635, 2012.

[26] W. Xie, X. Wang, F. Wang, W. Xu, R. M. Kennel, D. Gerling, and R. D. Lorenz, "Finite-control-set model predictive torque control with a deadbeat solution for pmsm drives," *IEEE Transactions on Industrial Electronics*, vol. 62, no. 9, pp. 5402–5410, 2015.

[27] S. Thielemans, T. Vyncke, M. Jacxsens, and J. Melkebeek, "Fpga implementation of online finite-set model based predictive control for power electronics," in *Predictive Control of Electrical Drives and Power Electronics (PRECEDE), 2011 Workshop on*, pp. 63–69, IEEE, 2011.

[28] P. Goedertier, T. Vyncke, F. De Belie, and J. Melkebeek, "Predictive current control of asynchronous machines by optimizing the switching moments," in *Sensorless Control for Electrical Drives and Predictive Control of Electrical Drives and Power Electronics (SLED/PRECEDE), 2013 IEEE International Symposium on*, pp. 1–7, IEEE, 2013.

[29] H. Zhu, X. Xiao, and Y. Li, "Torque ripple reduction of the torque predictive control scheme for permanent-magnet synchronous motors," *IEEE Transactions on Industrial Electronics*, vol. 59, no. 2, pp. 871–877, 2012.

[30] L. Tang, L. Zhong, M. F. Rahman, and Y. Hu, "A novel direct torque controlled interior permanent magnet synchronous machine drive with low ripple in flux and torque and fixed switching frequency," *IEEE Transactions on power electronics*, vol. 19, no. 2, pp. 346–354, 2004.

[31] Y. Xu, Q. Zhou, and B. Zhang, "A model predictive torque control strategy of pmsm with torque deadbeat duty cycle control," in *Power Electronics and Motion Control Conference (IPEMC-ECCE Asia), 2016 IEEE 8th International*, pp. 782–785, IEEE, 2016.

[32] F. Morel, X. Lin-Shi, J.-M. Rétif, B. Allard, and C. Buttay, "A comparative study of predictive current control schemes for a permanent-magnet synchronous machine drive," *IEEE Transactions on Industrial Electronics*, vol. 56, no. 7, pp. 2715–2728, 2009.

[33] M. Siami and S. A. Gholamian, "Predictive torque control of three phase axial flux permanent magnet synchronous machines," *Majlesi Journal of Electrical Engineering*, vol. 6, no. 2, 2012.

[34] Y. Zhang and H. Yang, "Model predictive torque control of induction motor drives with optimal duty cycle control," *IEEE Transactions on Power Electronics*, vol. 29, no. 12, pp. 6593–6603, 2014.

[35] R. Dewitte, "Analyse van de kostenfunctie bij een voorspellende regeling van flying capacitor convertoren.," Master's thesis, Ghent University, EESA, EELAB, June 2011.

[36] A. D. Alexandrou, N. K. Adamopoulos, and A. G. Kladas, "Development of a constant switching frequency deadbeat predictive control technique for field-oriented synchronous permanent-magnet motor drive," *IEEE Transactions on Industrial Electronics*, vol. 63, pp. 5167–5175, Aug. 2016.

[37] S.-M. Yang and C.-H. Lee, "A deadbeat current controller for field oriented induction motor drives," *IEEE Transactions on Power Electronics*, vol. 17, no. 5, pp. 772–778, 2002.

[38] C.-K. Lin, T.-H. Liu, L.-C. Fu, C.-F. Hsiao, *et al.*, "Model-free predictive current control for interior permanent-magnet synchronous motor drives based on current difference detection technique," *IEEE transactions on industrial electronics*, vol. 61, no. 2, pp. 667–681, 2014.

[39] B. Alecsa, M. N. Cirstea, and A. Onea, "Simulink modeling and design of an efficient hardware-constrained fpga-based pmsm speed controller," *IEEE Transactions on Industrial Informatics*, vol. 8, no. 3, pp. 554–562, 2012.

[40] J. L. Jerez, P. J. Goulart, S. Richter, G. A. Constantinides, E. C. Kerrigan, and M. Morari, "Embedded online optimization for model predictive control at megahertz rates," *IEEE Transactions on Automatic Control*, vol. 59, no. 12, pp. 3238–3251, 2014.

[41] M. Leuer and J. Bocker, "Real-time implementation of an online model predictive control for ipmsm using parallel computing on fpga," in *Power Electronics Conference (IPEC-Hiroshima 2014-ECCE-ASIA), 2014 International*, pp. 346–350, IEEE, 2014.

Appendix A

Simulation Results

A.1 Robustness

For all plots in this section applies that the blue curve represents the reference signal i_q^* and the red curve the simulated signal i_q.

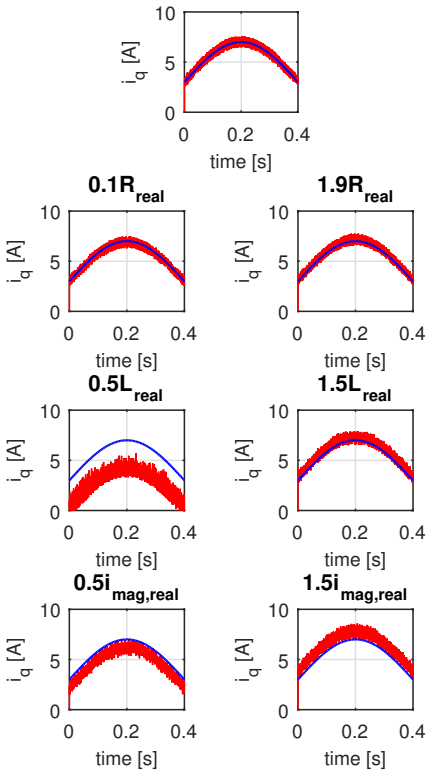


Figure A.1: Parameter sensitivity $\mathrm{PTC}_{\mathrm{quadr}}$

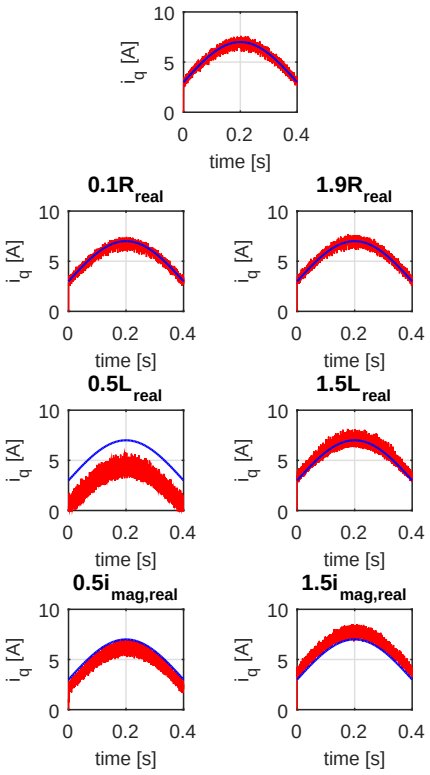


Figure A.2: Parameter sensitivity $\mathrm{PTC}_{\mathrm{tol}}$

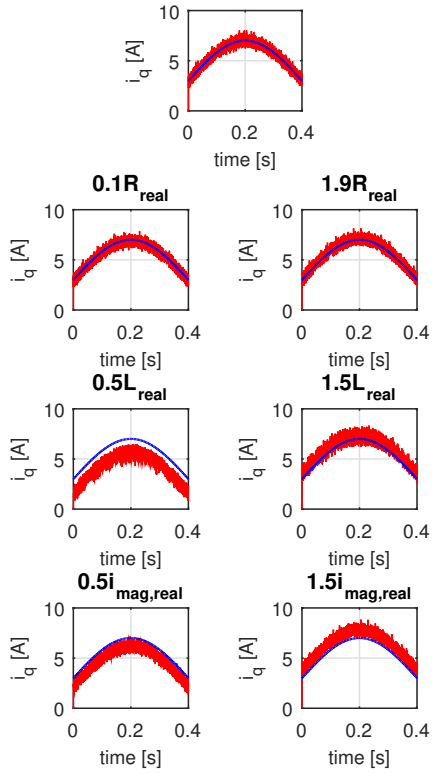


Figure A.3: Parameter sensitivity PCC_{tol}

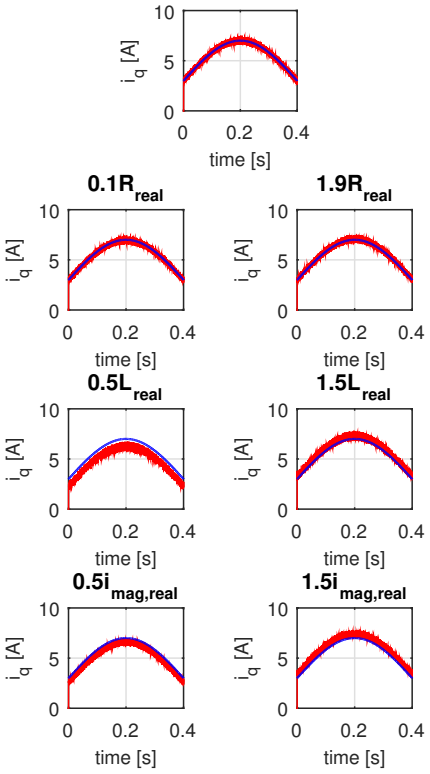


Figure A.4: Parameter sensitivity PCC_{2f}

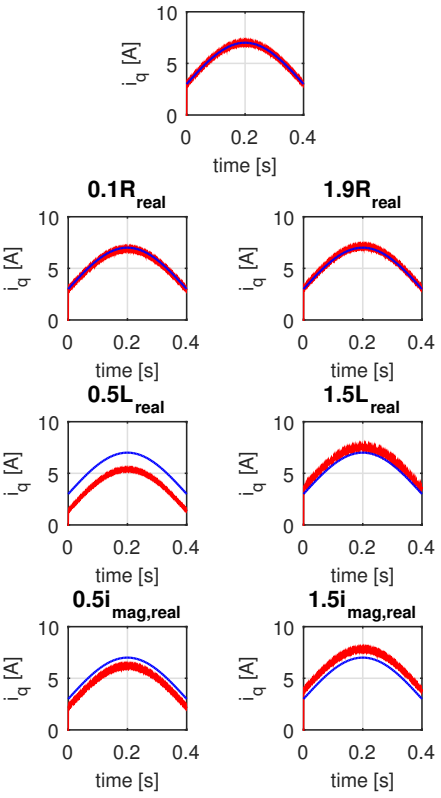


Figure A.5: Parameter sensitivity PCC_{DC}

Table A.1: MAR(i_q) [A]

	PTC_{quadr}	PTC_{tol}	PCC_{quadr}	PCC_{tol}	PCC_{2f}	PCC_{DC}	DB
ref	0.0075	0.0078	0.0105	0.0102	0.0054	0.0004	0.0003
$0.1R_{real}$	0.0075	0.0076	0.0105	0.0104	0.0052	0.0007	0.0007
$1.9R_{real}$	0.0073	0.0075	0.0105	0.0104	0.0054	0.0008	0.0008
$0.5L_{real}$	0.0142	0.0148	0.0115	0.0119	0.0058	0.0006	0.0003
$1.5L_{real}$	0.0083	0.0109	0.0118	0.0120	0.0057	0.0006	0.0163
$0.5i_{mag,real}$	0.0083	0.0098	0.0104	0.0105	0.0052	0.0004	0.0003
$1.5i_{mag,real}$	0.0081	0.0090	0.0105	0.0104	0.0052	0.0004	0.0003

Table A.2: bias(i_q) [A]

	PTC_{quadr}	PTC_{tol}	PCC_{quadr}	PCC_{tol}	PCC_{2f}	PCC_{DC}	DB
ref	0.0473	-0.0514	-0.0067	-0.0098	-0.0058	-0.0003	0.0019
$0.1R_{real}$	-0.0615	-0.1406	-0.0948	-0.0997	-0.0521	-0.0945	-0.0923
$1.9R_{real}$	0.1509	0.0456	0.0894	0.0871	0.0468	0.0964	0.0986
$0.5L_{real}$	-2.7852	-2.7850	-1.5775	-1.5606	-0.7865	-1.6137	-1.6240
$1.5L_{real}$	0.3663	0.1767	0.5377	0.4138	0.2649	0.5392	0.5385
$0.5i_{mag,real}$	-0.8256	-0.8540	-0.8131	-0.8050	-0.4003	-0.8111	-0.8092
$1.5i_{mag,real}$	0.8164	0.7113	0.8075	0.8156	0.4036	0.8105	0.8127

A.2 Detailed KPIs

Table A.3: $\text{PTC}_{\text{quadr}}$

KPI	0 Nm			8 Nm		
	i_q	i_d	$\lvert\Psi_\text{s}\rvert$	i_q	i_d	$\lvert\Psi_\text{s}\rvert$
MSE [p.u.]	0.5946	1.3515	0.0007	0.0149	0.1364	0.0026
MAE [p.u.]	0.6566	0.9721	0.0226	0.1014	0.2940	0.0404
average [A, A, Wb]	0.0582	-0.4256	0.1108	5.9913	-0.8894	0.1130
bias [p.u.]	0.0582	-0.4256	0.0100	-0.0141	-0.1464	0.0208
MAR [p.u.]	0.6537	0.8897	0.0204	0.1028	0.2722	0.0362
MSR [p.u.]	0.5912	1.1703	0.0006	0.0152	0.1183	0.0021
f_switch [kHz]		5.05			4.63	
non-PPCR [%]		55.29			70.27	
THD [%]		-			42.79	
Joule loss [W]		0.67			13.54	
OS [%]		-142.72			-20.04	
T_rise [ms]		0.2			0.2	
KPI	12 Nm			15 Nm		
	i_q	i_d	$\lvert\Psi_\text{s}\rvert$	i_q	i_d	$\lvert\Psi_\text{s}\rvert$
MSE [p.u.]	0.0086	0.0274	0.0011	0.0054	0.0157	0.0009
MAE [p.u.]	0.0758	0.1367	0.0273	0.0600	0.1019	0.0249
average [A, A, Wb]	9.0824	-0.0924	0.1124	11.4964	-0.3514	0.1144
bias [p.u.]	-0.0037	-0.0101	0.0027	0.0089	-0.0308	0.0088
MAR [p.u.]	0.0759	0.1371	0.0272	0.0592	0.0979	0.0237
MSR [p.u.]	0.0087	0.0275	0.0011	0.0052	0.0145	0.0008
f_switch [kHz]		4.66			4.51	
non-PPCR [%]		61.39			52.71	
THD [%]		18.53			14.34	
Joule loss [W]		27.92			43.73	
OS [%]		18.25			11.37	
T_rise [ms]		0.2			0.6	

Table A.4: $\mathrm{PTC_{tol}}$

KPI	0 Nm			8 Nm		
	i_q	i_d	$\lvert\Psi_\mathrm{s}\rvert$	i_q	i_d	$\lvert\Psi_\mathrm{s}\rvert$
MSE [p.u.]	0.6021	6.4267	0.0035	0.0160	0.2933	0.0056
MAE [p.u.]	0.6130	2.1277	0.0493	0.1023	0.4489	0.0616
average [A, A, Wb]	0.0844	-0.1188	0.1100	6.1462	-1.8365	0.1155
bias [p.u.]	0.0844	-0.1188	0.0027	0.0114	-0.3022	0.0424
MAR [p.u.]	0.6048	2.1203	0.0489	0.1009	0.3610	0.0478
MSR [p.u.]	0.5949	6.4126	0.0034	0.0156	0.1974	0.0035
f_switch [kHz]		4.71			4.46	
non-PPCR [%]		62.23			69.20	
THD [%]		-			33.15	
Joule loss [W]		2.29			15.90	
OS [%]		-127.69			-10.05	
T_rise [ms]		0.4			0.2	
KPI	12 Nm			15 Nm		
	i_q	i_d	$\lvert\Psi_\mathrm{s}\rvert$	i_q	i_d	$\lvert\Psi_\mathrm{s}\rvert$
MSE [p.u.]	0.0075	0.0911	0.0037	0.0052	0.0569	0.0035
MAE [p.u.]	0.0696	0.2589	0.0523	0.0584	0.1909	0.0469
average [A, A, Wb]	9.2056	-1.5745	0.1161	11.5093	-0.8762	0.1157
bias [p.u.]	0.0099	-0.1727	0.0357	0.0101	-0.0769	0.0203
MAR [p.u.]	0.0683	0.2016	0.0387	0.0570	0.1810	0.0436
MSR [p.u.]	0.0073	0.0601	0.0022	0.0050	0.0499	0.0029
f_switch [kHz]		4.36			4.31	
non-PPCR [%]		65.36			54.79	
THD [%]		26.36			22.18	
Joule loss [W]		30.13			45.34	
OS [%]		4.04			9.80	
T_rise [ms]		0.2			0.6	

Table A.5: $\text{PCC}_{\text{quadr}}$

KPI	0 Nm			8 Nm		
	i_q	i_d	$\lvert\Psi_\text{s}\rvert$	i_q	i_d	$\lvert\Psi_\text{s}\rvert$
MSE [p.u.]	0.8956	1.4771	0.0008	0.0288	0.0618	0.0012
MAE [p.u.]	0.7906	1.0110	0.0234	0.1400	0.2071	0.0294
average [A, A, Wb]	-0.0726	-0.3092	0.1105	6.2178	-0.9458	0.1132
bias [p.u.]	-0.0726	-0.3092	0.0073	0.0232	-0.1556	0.0226
MAR [p.u.]	0.7880	0.9762	0.0224	0.1350	0.1590	0.0222
MSR [p.u.]	0.8903	1.3815	0.0007	0.0270	0.0359	0.0007
f_{switch} [kHz]		4.66			4.37	
non-PPCR [%]		41.87			37.07	
THD [%]		-			23.46	
Joule loss [W]		0.90			13.81	
OS [%]		-87.65			-13.00	
T_rise [ms]		0.5			0.5	
KPI	12 Nm			15 Nm		
	i_q	i_d	$\lvert\Psi_\text{s}\rvert$	i_q	i_d	$\lvert\Psi_\text{s}\rvert$
MSE [p.u.]	0.0142	0.0239	0.0010	0.0086	0.0232	0.0015
MAE [p.u.]	0.0984	0.1218	0.0250	0.0764	0.1283	0.0330
average [A, A, Wb]	9.0548	-0.9532	0.1145	11.5381	-1.3201	0.1168
bias [p.u.]	-0.0067	-0.1046	0.0214	0.0126	-0.1159	0.0300
MAR [p.u.]	0.0989	0.0947	0.0190	0.0746	0.0818	0.0206
MSR [p.u.]	0.0143	0.0132	0.0005	0.0082	0.0095	0.0006
f_switch [kHz]		4.19			4.11	
non-PPCR [%]		40.18			33.48	
THD [%]		15.11			11.51	
Joule loss [W]		27.69			44.46	
OS [%]		23.39			8.09	
T_rise [ms]		0.2			0.6	

Table A.6: $\mathrm{PCC_{tol}}$

KPI	0 Nm			8 Nm		
	i_q	i_d	$\lvert\Psi_\mathrm{s}\rvert$	i_q	i_d	$\lvert\Psi_\mathrm{s}\rvert$
MSE [p.u.]	0.8966	1.3844	0.0007	0.0294	0.0336	0.0007
MAE [p.u.]	0.7958	0.9591	0.0222	0.1418	0.1428	0.0208
average [A, A, Wb]	-0.0677	-0.2604	0.1104	5.9728	-0.0874	0.1110
bias [p.u.]	-0.0677	-0.2604	0.0064	-0.0172	-0.0144	0.0018
MAR [p.u.]	0.7944	0.9372	0.0215	0.1428	0.1458	0.0208
MSR [p.u.]	0.8920	1.3166	0.0007	0.0301	0.0346	0.0007
f_switch [kHz]		4.63			4.75	
non-PPCR [%]		41.91			47.92	
THD [%]		-			25.4	
Joule loss [W]		0.83			12.41	
OS [%]		-34.77			-20.56	
T_rise [ms]		0.3			0.4	
KPI	12 Nm			15 Nm		
	i_q	i_d	$\lvert\Psi_\mathrm{s}\rvert$	i_q	i_d	$\lvert\Psi_\mathrm{s}\rvert$
MSE [p.u.]	0.0120	0.0153	0.0007	0.0086	0.0135	0.0009
MAE [p.u.]	0.0915	0.1014	0.0215	0.0781	0.0951	0.0239
average [A, A, Wb]	9.0126	-0.3660	0.1130	11.5136	-0.4859	0.1148
bias [p.u.]	-0.0113	-0.0402	0.0080	0.0104	-0.0426	0.0115
MAR [p.u.]	0.0921	0.0970	0.0204	0.0769	0.0917	0.0236
MSR [p.u.]	0.0122	0.0140	0.0006	0.0083	0.0115	0.0007
f_switch [kHz]		4.42			4.35	
non-PPCR [%]		39.52			38.08	
THD [%]		15.29			14.93	
Joule loss [W]		27.02			43.91	
OS [%]		17.6			8.11	
T_rise [ms]		0.6			0.6	

Table A.7: PCC_{2f}

KPI	0 Nm			8 Nm		
	i_q	i_d	$\lvert\Psi_s\rvert$	i_q	i_d	$\lvert\Psi_s\rvert$
MSE [p.u.]	0.2242	0.3741	0.0002	0.0062	0.0119	0.0002
MAE [p.u.]	0.3951	0.5065	0.0117	0.0652	0.0891	0.0124
average [A, A, Wb]	0.0081	-0.0164	0.1097	6.0185	-0.3528	0.1117
bias [p.u.]	0.0081	-0.0164	0.0000	-0.0097	-0.0581	0.0081
MAR [p.u.]	0.3950	0.5064	0.0117	0.0655	0.0770	0.0106
MSR [p.u.]	0.2241	0.3738	0.0002	0.0063	0.0087	0.0002
f_{switch} [kHz]		9.13			9.01	
non-PPCR [%]		41.30			40.92	
THD [%]		-			12.19	
Joule loss [W]		0.23			12.05	
OS [%]		-48.52			-7.40	
T_{rise} [ms]		0.25			0.20	
KPI	12 Nm			15 Nm		
	i_q	i_d	$\lvert\Psi_s\rvert$	i_q	i_d	$\lvert\Psi_s\rvert$
MSE [p.u.]	0.0032	0.0066	0.0003	0.0021	0.0045	0.0003
MAE [p.u.]	0.0472	0.0663	0.0135	0.0375	0.0547	0.0137
average [A, A, Wb]	9.1136	-0.4759	0.1133	11.3714	-0.4536	0.1146
bias [p.u.]	-0.0002	-0.0522	0.0107	-0.0020	-0.0398	0.0097
MAR [p.u.]	0.0472	0.0509	0.0105	0.0375	0.0450	0.0115
MSR [p.u.]	0.0032	0.0039	0.0002	0.0021	0.0030	0.0002
f_{switch} [kHz]		8.84			8.62	
non-PPCR [%]		37.88			38.96	
THD [%]		8.71			6.99	
Joule loss [W]		27.24			42.14	
OS [%]		5.38			1.48	
T_{rise} [ms]		0.30			0.65	

Table A.8: $\mathrm{PCC_{DC}}$

KPI	0 Nm			8 Nm		
	i_q	i_d	$\lvert\Psi_\mathrm{s}\rvert$	i_q	i_d	$\lvert\Psi_\mathrm{s}\rvert$
MSE [p.u.]	0.0153	1.2909	0.0007	0.0003	0.0242	0.0005
MAE [p.u.]	0.0978	0.8586	0.0199	0.0134	0.1252	0.0172
average [A, A, Wb]	0.0051	-0.3313	0.1105	6.1084	-0.2952	0.1115
bias [p.u.]	0.0051	-0.3313	0.0008	0.0312	-0.2952	0.0008
MAR [p.u.]	0.0984	0.8261	0.0190	0.0128	0.1154	0.0158
MSR [p.u.]	0.0152	1.1811	0.0006	0.0002	0.0216	0.0004
f_switch [kHz]		11.96			10.70	
non-PPCR [%]		20.89			16.74	
THD [%]		-			14.77	
Joule loss [W]		0.47			12.49	
OS [%]		-15.35			-2.84	
T_rise [ms]		0.4			0.3	
KPI	12 Nm			15 Nm		
	i_q	i_d	$\lvert\Psi_\mathrm{s}\rvert$	i_q	i_d	$\lvert\Psi_\mathrm{s}\rvert$
MSE [p.u.]	0.0001	0.0133	0.0005	0.0001	0.0080	0.0005
MAE [p.u.]	0.0076	0.0895	0.0179	0.0068	0.0701	0.0172
average [A, A, Wb]	9.1117	-0.3578	0.1130	11.3838	-0.3352	0.1143
bias [p.u.]	-0.0041	-0.3578	0.0009	-0.0109	-0.3352	0.0008
MAR [p.u.]	0.0076	0.0857	0.0171	0.0069	0.0689	0.0168
MSR [p.u.]	0.0001	0.0118	0.0005	0.0001	0.0072	0.0004
f_switch [kHz]		12.20			12.23	
non-PPCR [%]		21.63			21.75	
THD [%]		10.90			8.17	
Joule loss [W]		27.32			42.31	
OS [%]		1.86			1.02	
T_rise [ms]		0.4			0.8	

Table A.9: PI

KPI	0 Nm			8 Nm		
	i_q	i_d	$\lvert\Psi_\mathrm{s}\rvert$	i_q	i_d	$\lvert\Psi_\mathrm{s}\rvert$
MSE [p.u.]	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
MAE [p.u.]	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
average [A, A, Wb]	0.0000	0.0000	0.1097	6.0772	0.0000	0.1108
bias [p.u.]	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
MAR [p.u.]	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
MSR [p.u.]	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
f_switch [kHz]		20.00			20.00	
non-PPCR [%]		0.07			0.00	
THD [%]		-			0.00	
Joule loss [W]		1.34			12.08	
OS [%]		-120.67			-11.58	
T_rise [ms]		1.1			1.1	
KPI	12 Nm			15 Nm		
	i_q	i_d	$\lvert\Psi_\mathrm{s}\rvert$	i_q	i_d	$\lvert\Psi_\mathrm{s}\rvert$
MSE [p.u.]	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
MAE [p.u.]	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
average [A, A, Wb]	9.1158	0.0000	0.1121	11.3947	0.0000	0.1135
bias [p.u.]	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
MAR [p.u.]	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
MSR [p.u.]	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
f_switch [kHz]		20.00			20.00	
non-PPCR [%]		0.00			0.00	
THD [%]		0.00			0.00	
Joule loss [W]		27.01			42.35	
OS [%]		4.41			13.98	
T_rise [ms]		1.1			1.1	

Table A.10: DB

KPI	0 Nm			8 Nm		
	i_q	i_d	$\lvert\Psi_\mathrm{s}\rvert$	i_q	i_d	$\lvert\Psi_\mathrm{s}\rvert$
MSE [p.u.]	0.0006	0.2867	0.0002	0.0002	0.0303	0.0006
MAE [p.u.]	0.0250	0.5354	0.0124	0.0129	0.1741	0.0243
average [A, A, Wb]	0.0250	-0.5354	0.1111	6.1558	-1.0582	0.1135
bias [p.u.]	0.0250	-0.5354	0.0128	0.0129	-0.1741	0.0244
MAR [p.u.]	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
MSR [p.u.]	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
f_switch [kHz]		20.00			20.00	
non-PPCR [%]		0.03			0.00	
THD [%]		-			0.00	
Joule loss [W]		0.19			12.77	
OS [%]		-371.20			-36.38	
T_rise [ms]		0.4			0.4	
KPI	12 Nm			15 Nm		
	i_q	i_d	$\lvert\Psi_\mathrm{s}\rvert$	i_q	i_d	$\lvert\Psi_\mathrm{s}\rvert$
MSE [p.u.]	0.0001	0.0210	0.0009	0.0001	0.0177	0.0011
MAE [p.u.]	0.0116	0.1448	0.0298	0.0110	0.1330	0.0335
average [A, A, Wb]	9.2212	-1.3197	0.1155	11.5202	-1.5157	0.1173
bias [p.u.]	0.0116	-0.1448	0.0294	0.0110	-0.1330	0.0335
MAR [p.u.]	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
MSR [p.u.]	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
f_switch [kHz]		20.00			19.96	
non-PPCR [%]		0.00			0.00	
THD [%]		0.00			0.00	
Joule loss [W]		28.16			43.72	
OS [%]		9.78			7.89	
T_rise [ms]		0.4			0.8	

Appendix B

Experimental Results

B.1 Robustness

For all plots in this section applies that the red curve represents the reference signal i_q^* and the blue curve the simulated signal i_q.

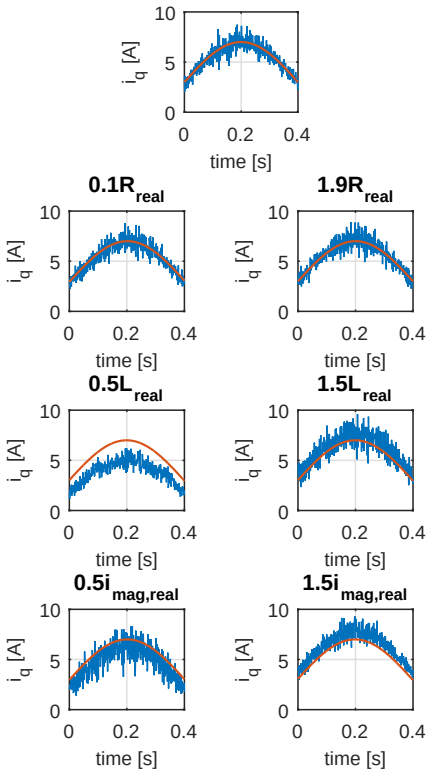


Figure B.1: Parameter sensitivity $\text{PTC}_{\text{quadr}}$

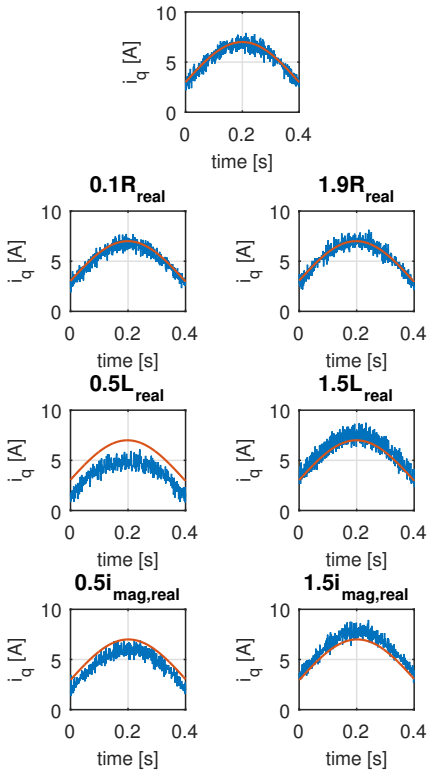


Figure B.2: Parameter sensitivity $\text{PCC}_{\text{quadr}}$

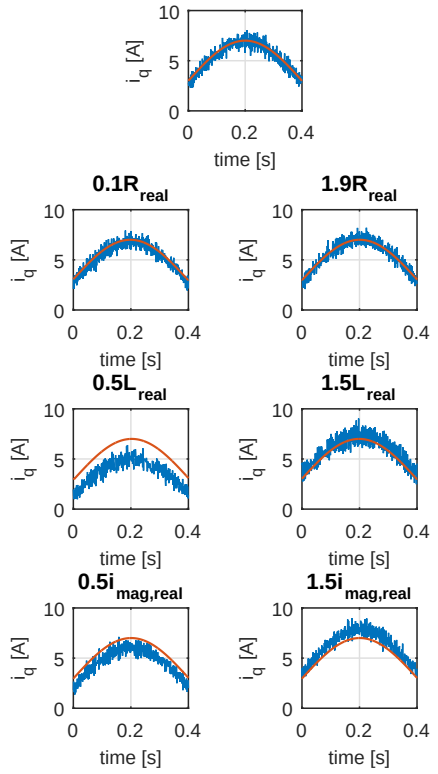


Figure B.3: Parameter sensitivity $\mathrm{PCC_{tol}}$

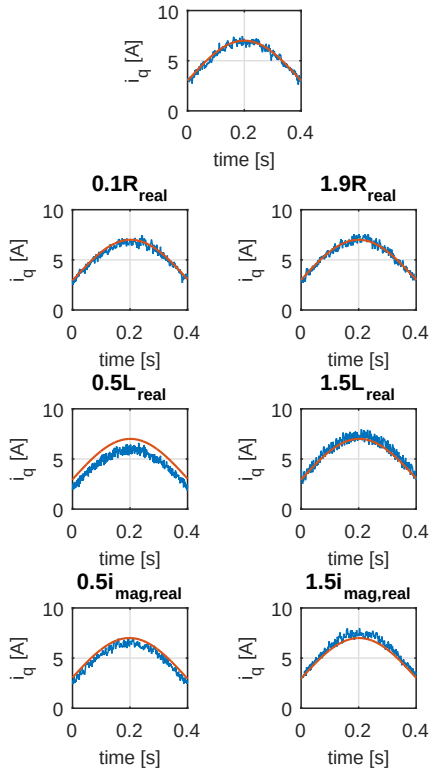


Figure B.4: Parameter sensitivity $\mathrm{PCC_{2f}}$

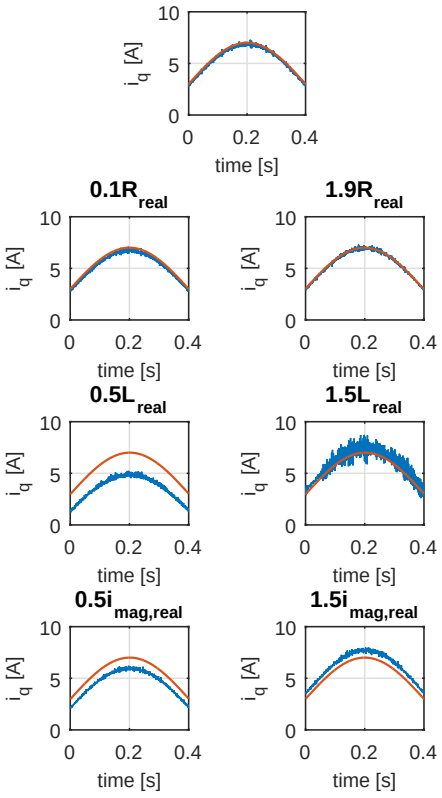


Figure B.5: Parameter sensitivity $\mathrm{PCC_{DC}}$

Table B.1: MAR(i_q) [A]

	PTC_{quadr}	PTC_{tol}	PCC_{quadr}	PCC_{tol}	PCC_{2f}	PCC_{DC}	DB
ref	0.0116	0.0050	0.0084	0.0092	0.0027	0.0014	0.0018
$0.1R_{real}$	0.0117	0.0062	0.0091	0.0092	0.0032	0.0026	0.0035
$1.9R_{real}$	0.0129	0.0069	0.0077	0.0083	0.0026	0.0007	0.0007
$0.5L_{real}$	0.1115	0.1050	0.1201	0.1222	0.0352	0.1231	0.1444
$1.5L_{real}$	0.0304	0.0148	0.0218	0.0198	0.0072	0.0185	0.0052
$0.5i_{mag,real}$	0.0495	0.0264	0.0367	0.0361	0.0099	0.0306	0.0356
$1.5i_{mag,real}$	0.0304	0.0246	0.0225	0.0250	0.0066	0.0151	0.0112

Table B.2: bias(i_q) [A]

	PTC_{quadr}	PTC_{tol}	PCC_{quadr}	PCC_{tol}	PCC_{2f}	PCC_{DC}	DB
ref	-0.0675	-0.0236	-0.1520	-0.1225	-0.1008	-0.1478	-0.2309
$0.1R_{real}$	-0.1229	-0.1264	-0.2619	-0.2229	-0.1551	-0.2442	-0.3202
$1.9R_{real}$	0.0327	0.0738	-0.0776	-0.0231	-0.0144	-0.0571	-0.1382
$0.5L_{real}$	-1.7431	-1.6945	-1.8224	-1.8455	-0.9870	-1.9173	-2.0827
$1.5L_{real}$	0.5199	0.2849	0.4953	0.3918	0.2562	0.4322	0.3916
$0.5i_{mag,real}$	-0.8403	-0.7727	-0.9457	-0.9292	-0.4862	-0.9520	-1.0353
$1.5i_{mag,real}$	0.7713	0.7232	0.6677	0.7138	0.3573	0.6561	0.5745

B.2 Detailed KPIs

Table B.3: $\text{PTC}_{\text{quadr}}$

KPI	0 Nm			8 Nm		
	i_q	i_d	$\lvert\Psi_\text{s}\rvert$	i_q	i_d	$\lvert\Psi_\text{s}\rvert$
MSE [p.u.]	3.2394	3.6843	0.0020	0.0482	0.4896	0.0097
MAE [p.u.]	1.4754	1.5528	0.0366	0.1826	0.6643	0.0935
average [A, A, Wb]	-0.4556	-1.2970	0.1131	6.4616	-4.0372	0.1211
bias [p.u.]	-0.4556	-1.2970	0.0310	0.0633	-0.6643	0.0939
MAR [p.u.]	1.4022	1.1619	0.0261	0.1646	0.1639	0.0227
MSR [p.u.]	3.0319	2.0020	0.0010	0.0391	0.0427	0.0008
f_switch [kHz]		3.67			3.85	
non-PPCR [%]		45.95			37.38	
THD [%]		-			15.71	
Joule loss [W]		2.44			20.16	
OS [%]		-330.90			-40.52	
T_rise [ms]		0.3			0.3	
KPI	12 Nm			15 Nm		
	i_q	i_d	$\lvert\Psi_\text{s}\rvert$	i_q	i_d	$\lvert\Psi_\text{s}\rvert$
MSE [p.u.]	0.0267	0.3738	0.0163	0.0211	0.3163	0.0204
MAE [p.u.]	0.1325	0.5870	0.1219	0.1193	0.5332	0.1340
average [A, A, Wb]	9.6790	-5.3513	0.1258	11.6466	-6.0753	0.1287
bias [p.u.]	0.0618	-0.5870	0.1222	0.0221	-0.5332	0.1340
MAR [p.u.]	0.1152	0.1308	0.0269	0.1153	0.1461	0.0359
MSR [p.u.]	0.0203	0.0259	0.0011	0.0197	0.0307	0.0019
f_switch [kHz]		3.68			2.97	
non-PPCR [%]		42.88			39.50	
THD [%]		14.90			10.15	
Joule loss [W]		41.02			57.83	
OS [%]		6.53			30.96	
T_rise [ms]		0.2			0.7	

Table B.4: $\mathrm{PTC_{tol}}$

KPI	0 Nm			8 Nm		
	i_q	i_d	$\lvert\Psi_\mathrm{s}\rvert$	i_q	i_d	$\lvert\Psi_\mathrm{s}\rvert$
MSE [p.u.]	3.1169	4.8530	0.0027	0.0763	0.5410	0.0108
MAE [p.u.]	1.3887	1.8780	0.0442	0.2250	0.6921	0.0974
average [A, A, Wb]	-0.2029	-1.6635	0.1140	6.3821	-4.2027	0.1216
bias [p.u.]	-0.2029	-1.6635	0.0392	0.0502	-0.6916	0.0975
MAR [p.u.]	1.3775	1.1784	0.0261	0.2080	0.1898	0.0264
MSR [p.u.]	3.0757	2.0857	0.0010	0.0669	0.0569	0.0011
f_switch [kHz]		3.78			3.68	
non-PPCR [%]		46.68			48.05	
THD [%]		-			19.46	
Joule loss [W]		2.67			20.71	
OS [%]		-2.56			-11.62	
T_rise [ms]		0.2			0.1	
KPI	12 Nm			15 Nm		
	i_q	i_d	$\lvert\Psi_\mathrm{s}\rvert$	i_q	i_d	$\lvert\Psi_\mathrm{s}\rvert$
MSE [p.u.]	0.0285	0.3412	0.0147	0.0187	0.3069	0.0197
MAE [p.u.]	0.1401	0.5602	0.1155	0.1114	0.5342	0.1345
average [A, A, Wb]	9.4406	-5.1064	0.1251	11.7014	-6.0865	0.1287
bias [p.u.]	0.0356	-0.5602	0.1160	0.0269	-0.5342	0.1349
MAR [p.u.]	0.1326	0.1302	0.0261	0.1058	0.1156	0.0281
MSR [p.u.]	0.0254	0.0256	0.0010	0.0170	0.0205	0.0012
f_switch [kHz]		3.49			3.06	
non-PPCR [%]		42.56			32.12	
THD [%]		17.90			8.88	
Joule loss [W]		39.09			57.55	
OS [%]		17.23			33.55	
T_rise [ms]		0.0			0.5	

Table B.5: $\mathrm{PCC_{quadr}}$

KPI	0 Nm			8 Nm		
	i_q	i_d	$\lvert\Psi_\mathrm{s}\rvert$	i_q	i_d	$\lvert\Psi_\mathrm{s}\rvert$
MSE [p.u.]	0.9361	1.1826	0.0006	0.0563	0.1733	0.0034
MAE [p.u.]	0.7697	0.8602	0.0199	0.2024	0.3661	0.0511
average [A, A, Wb]	0.2130	0.1086	0.1095	6.0596	-2.1872	0.1163
bias [p.u.]	0.2130	0.1086	-0.0018	-0.0029	-0.3599	0.0506
MAR [p.u.]	0.7406	0.8690	0.0201	0.2029	0.1672	0.0223
MSR [p.u.]	0.8907	1.1708	0.0006	0.0566	0.0440	0.0008
f_switch [kHz]		4.58			3.93	
non-PPCR [%]		47.79			37.94	
THD [%]		-			21.22	
Joule loss [W]		0.76			14.94	
OS [%]		-203.88			-43.34	
T_rise [ms]		0.5			0.4	
KPI	12 Nm			15 Nm		
	i_q	i_d	$\lvert\Psi_\mathrm{s}\rvert$	i_q	i_d	$\lvert\Psi_\mathrm{s}\rvert$
MSE [p.u.]	0.0276	0.1547	0.0065	0.0168	0.1662	0.0104
MAE [p.u.]	0.1400	0.3580	0.0734	0.1085	0.3872	0.0969
average [A, A, Wb]	9.1629	-3.2566	0.1203	11.5122	-4.4123	0.1244
bias [p.u.]	0.0052	-0.3572	0.0731	0.0103	-0.3872	0.0970
MAR [p.u.]	0.1391	0.1315	0.0251	0.1068	0.1033	0.0243
MSR [p.u.]	0.0273	0.0268	0.0010	0.0163	0.0159	0.0009
f_switch [kHz]		3.63			3.29	
non-PPCR [%]		42.78			38.32	
THD [%]		13.37			11.50	
Joule loss [W]		31.98			49.84	
OS [%]		1.30			10.90	
T_rise [ms]		0.7			0.7	

Table B.6: $\mathrm{PCC_{tol}}$

KPI	0 Nm			8 Nm		
	i_q	i_d	$\lvert\Psi_\mathrm{s}\rvert$	i_q	i_d	$\lvert\Psi_\mathrm{s}\rvert$
MSE [p.u.]	0.8788	1.1700	0.0006	0.0566	0.1672	0.0033
MAE [p.u.]	0.7221	0.8770	0.0202	0.2002	0.3497	0.0488
average [A, A, Wb]	0.0679	0.0805	0.1095	6.0950	-2.0677	0.1160
bias [p.u.]	0.0679	0.0805	-0.0018	0.0029	-0.3402	0.0478
MAR [p.u.]	0.7225	0.8799	0.0203	0.1996	0.1880	0.0252
MSR [p.u.]	0.8742	1.1635	0.0006	0.0563	0.0511	0.0009
f_{switch} [kHz]		4.16			3.61	
non-PPCR [%]		43.83			40.39	
THD [%]		-			21.63	
Joule loss [W]		0.70			14.92	
OS [%]		-145.87			-23.66	
T_rise [ms]		0.3			0.5	
KPI	12 Nm			15 Nm		
	i_q	i_d	$\lvert\Psi_\mathrm{s}\rvert$	i_q	i_d	$\lvert\Psi_\mathrm{s}\rvert$
MSE [p.u.]	0.0223	0.1624	0.0068	0.0187	0.1726	0.0108
MAE [p.u.]	0.1276	0.3771	0.0772	0.1148	0.3962	0.0988
average [A, A, Wb]	9.2177	-3.4376	0.1208	11.4431	-4.5144	0.1247
bias [p.u.]	0.0112	-0.3771	0.0776	0.0042	-0.3962	0.0987
MAR [p.u.]	0.1257	0.1157	0.0228	0.1142	0.0999	0.0232
MSR [p.u.]	0.0217	0.0198	0.0008	0.0185	0.0155	0.0008
f_{switch} [kHz]		3.50			3.02	
non-PPCR [%]		44.29			33.93	
THD [%]		23.83			14.95	
Joule loss [W]		32.11			49.96	
OS [%]		3.22			24.82	
T_rise [ms]		0.1			0.7	

Table B.7: PCC_{2f}

KPI	0 Nm			8 Nm		
	i_q	i_d	$\lvert\Psi_s\rvert$	i_q	i_d	$\lvert\Psi_s\rvert$
MSE [p.u.]	0.1998	0.3472	0.0002	0.0123	0.0656	0.0013
MAE [p.u.]	0.3559	0.4793	0.0111	0.0944	0.2290	0.0321
average [A, A, Wb]	-0.0565	-0.0965	0.1100	6.2128	-1.3858	0.1143
bias [p.u.]	-0.0565	-0.0965	0.0027	0.0223	-0.2280	0.0316
MAR [p.u.]	0.3501	0.4735	0.0109	0.0902	0.0944	0.0131
MSR [p.u.]	0.1966	0.3379	0.0002	0.0113	0.0130	0.0003
f_{switch} [kHz]		9.16			7.83	
non-PPCR [%]		46.79			37.19	
THD [%]		-			12.69	
Joule loss [W]		0.29			13.56	
OS [%]		-310.72			-59.72	
T_{rise} [ms]		0.4			0.4	
KPI	12 Nm			15 Nm		
	i_q	i_d	$\lvert\Psi_s\rvert$	i_q	i_d	$\lvert\Psi_s\rvert$
MSE [p.u.]	0.0062	0.0471	0.0020	0.0044	0.0507	0.0032
MAE [p.u.]	0.0667	0.2030	0.0417	0.0552	0.2143	0.0543
average [A, A, Wb]	9.2300	-1.8507	0.1168	11.6097	-2.4419	0.1196
bias [p.u.]	0.0125	-0.2030	0.0419	0.0189	-0.2143	0.0546
MAR [p.u.]	0.0647	0.0632	0.0127	0.0520	0.0563	0.0137
MSR [p.u.]	0.0059	0.0058	0.0002	0.0039	0.0046	0.0003
f_{switch} [kHz]		7.36			6.96	
non-PPCR [%]		34.68			29.36	
THD [%]		7.99			6.00	
Joule loss [W]		29.20			45.66	
OS [%]		24.07			21.60	
T_{rise} [ms]		0.3			1.1	

Table B.8: $\mathrm{PCC_{DC}}$

KPI	0 Nm			8 Nm		
	i_q	i_d	$\lvert\Psi_\mathrm{s}\rvert$	i_q	i_d	$\lvert\Psi_\mathrm{s}\rvert$
MSE [p.u.]	0.3930	0.8264	0.0004	0.0334	0.2171	0.0039
MAE [p.u.]	0.4993	0.7647	0.0178	0.1466	0.4429	0.0597
average [A, A, Wb]	-0.1529	-0.6452	0.1114	5.4806	-2.6914	0.1174
bias [p.u.]	-0.1529	-0.6452	0.0155	-0.0982	-0.4429	0.0596
MAR [p.u.]	0.4866	0.5228	0.0119	0.1381	0.1312	0.0150
MSR [p.u.]	0.3696	0.4101	0.0002	0.0292	0.0257	0.0003
f_switch [kHz]		9.83			10.09	
non-PPCR [%]		13.60			14.57	
THD [%]		-			20.94	
Joule loss [W]		0.47			12.84	
OS [%]		-231.03			-61.55	
T_rise [ms]		0.1			0.1	
KPI	12 Nm			15 Nm		
	i_q	i_d	$\lvert\Psi_\mathrm{s}\rvert$	i_q	i_d	$\lvert\Psi_\mathrm{s}\rvert$
MSE [p.u.]	0.0147	0.2063	0.0080	0.0230	0.2292	0.0131
MAE [p.u.]	0.0959	0.4428	0.0874	0.1252	0.4584	0.1099
average [A, A, Wb]	8.5189	-4.0366	0.1219	10.5517	-5.2237	0.1259
bias [p.u.]	-0.0655	-0.4428	0.0874	-0.0740	-0.4584	0.1102
MAR [p.u.]	0.0856	0.0871	0.0149	0.1191	0.1209	0.0238
MSR [p.u.]	0.0119	0.0116	0.0003	0.0205	0.0222	0.0009
f_switch [kHz]		9.59			9.81	
non-PPCR [%]		12.70			13.57	
THD [%]		13.01			19.80	
Joule loss [W]		29.26			46.35	
OS [%]		30.88			20.79	
T_rise [ms]		0.3			1.4	

Table B.9: PI

KPI	0 Nm			8 Nm		
	i_q	i_d	$\lvert\Psi_\mathrm{s}\rvert$	i_q	i_d	$\lvert\Psi_\mathrm{s}\rvert$
MSE [p.u.]	0.0043	0.0061	0.0000	0.0019	0.0017	0.0000
MAE [p.u.]	0.0539	0.0621	0.0014	0.0359	0.0335	0.0046
average [A, A, Wb]	-0.0413	-0.0538	0.1098	6.0243	-0.0551	0.1109
bias [p.u.]	-0.0413	-0.0538	0.0009	-0.0087	-0.0091	0.0009
MAR [p.u.]	0.0409	0.0474	0.0011	0.0350	0.0333	0.0045
MSR [p.u.]	0.0026	0.0032	0.0000	0.0019	0.0016	0.0000
f_switch [kHz]		20.01			20.00	
non-PPCR [%]		0.03			0.00	
THD [%]		-			3.09	
Joule loss [W]		0.53			12.01	
OS [%]		-121.87			-57.29	
T_rise [ms]		2.0			2.0	
KPI	12 Nm			15 Nm		
	i_q	i_d	$\lvert\Psi_\mathrm{s}\rvert$	i_q	i_d	$\lvert\Psi_\mathrm{s}\rvert$
MSE [p.u.]	0.0008	0.0008	0.0000	0.0007	0.0006	0.0000
MAE [p.u.]	0.0237	0.0228	0.0045	0.0208	0.0201	0.0048
average [A, A, Wb]	9.0376	-0.0801	0.1123	11.3115	-0.0878	0.1136
bias [p.u.]	-0.0086	-0.0088	0.0018	-0.0073	-0.0077	0.0018
MAR [p.u.]	0.0228	0.0224	0.0044	0.0196	0.0194	0.0047
MSR [p.u.]	0.0008	0.0007	0.0000	0.0006	0.0006	0.0000
f_switch [kHz]		20.00			19.93	
non-PPCR [%]		0.00			0.00	
THD [%]		1.64			2.13	
Joule loss [W]		26.75			42.46	
OS [%]		18.38			48.09	
T_rise [ms]		1.4			1.7	

Table B.10: DB

KPI	0 Nm			8 Nm		
	i_q	i_d	$\lvert\Psi_\mathrm{s}\rvert$	i_q	i_d	$\lvert\Psi_\mathrm{s}\rvert$
MSE [p.u.]	0.0211	0.6026	0.0003	0.0088	0.2772	0.0051
MAE [p.u.]	0.1229	0.7693	0.0178	0.0872	0.5249	0.0710
average [A, A, Wb]	0.1112	-0.7693	0.1117	5.5474	-3.1896	0.1186
bias [p.u.]	0.1112	-0.7693	0.0182	-0.0872	-0.5248	0.0713
MAR [p.u.]	0.0785	0.0832	0.0019	0.0320	0.0394	0.0045
MSR [p.u.]	0.0087	0.0108	0.0000	0.0014	0.0021	0.0000
f_switch [kHz]		20.03			19.99	
non-PPCR [%]		0.03			0.00	
THD [%]		-			5.20	
Joule loss [W]		0.28			13.51	
OS [%]		-170.56			-36.03	
T_rise [ms]		0.4			0.4	
KPI	12 Nm			15 Nm		
	i_q	i_d	$\lvert\Psi_\mathrm{s}\rvert$	i_q	i_d	$\lvert\Psi_\mathrm{s}\rvert$
MSE [p.u.]	0.0058	0.2266	0.0088	0.0049	0.2314	0.0133
MAE [p.u.]	0.0722	0.4751	0.0935	0.0672	0.4804	0.1152
average [A, A, Wb]	8.4580	-4.3305	0.1226	10.6289	-5.4736	0.1265
bias [p.u.]	-0.0722	-0.4751	0.0937	-0.0672	-0.4804	0.1155
MAR [p.u.]	0.0220	0.0280	0.0047	0.0184	0.0232	0.0048
MSR [p.u.]	0.0006	0.0010	0.0000	0.0005	0.0007	0.0000
f_switch [kHz]		20.00			18.27	
non-PPCR [%]		0.00			0.01	
THD [%]		3.81			2.78	
Joule loss [W]		29.23			45.98	
OS [%]		3.21			4.75	
T_rise [ms]		0.7			1.1	