Programming and Industrial Control

Model-Based Predictive Control of 3-Level Inverters

Vom Fachbereich

Elektrotechnik, Informatik und Medientechnik

der Bergischen Universität Wuppertal

zur Erlangung des akademischen Grades eines Doktor-Ingenieurs

genehmigte Dissertation

vorgelegt von

Diplom-Ingenieur Nael AL_Sheakh Ameen

aus Aleppo, Syrien

Referent: Prof. Dr.-Ing. Ralph M. Kennel

Korreferent: Prof. Dr.-Ing. Bernd Tibken

Tag der mündlichen Prüfung 03. Februar 2012

Diese Dissertation kann wie folgt zitiert werden:

urn:nbn:de:hbz:468-20120503-135944-0

[http://nbn-resolving.de/urn/resolver.pl?urn=urn:nbn:de:hbz:468-20120503-135944-0]

To my Family

DEUTSCHE ZUSAMMENFASSUNG DER DISSERTATION

Anwendung der expliziten Lösung bei modellbasierten Prädiktivregelungen in elektrischen Antrieben

Mehrpunktwechselrichter erzeugen Ausgangsspannungen, die der Sinusform weit ähnlicher sind als die Ausgangsspannungen von üblichen 6-Puls-Wechselrichtern. In Anwendungen, in denen die Sinusform von entscheidender Bedeutung ist (z. B. bei der Energieversorgung von ländlichen Siedlungen oder Gehöften oder bei elektrischen Antrieben mit der Forderung nach geringer Drehmomentwelligkeit), würde diese Art von Umrichtern den Aufwand an notwendigen Ausgangsfiltern deutlich reduzieren.

Der wesentliche Nachteil von Mehrpunktwechselrichtern – der der Anwendung dieses Konzepts in der Industrie bisher im Wege steht – ist die im Vergleich mit Standard-Umrichtern hohe Anzahl von Leistungshalbleitern und die Notwendigkeit einer komplexen Regelung. Da in jedem Umrichterzweig eines Mehrpunkt-Wechselrichters mehrere Leistungshalbleiter in Serie geschaltet sind, braucht die Sperrspannung der einzelnen Leistungshableiter allerdings nicht so hoch zu sein wie bei Standard-Pulsumrichtern.

Der Preis für mehrere Leistungshalbleiter niedriger Nennspannung ist derzeit zwar immer noch höher als für einen einzelnen Leistungshalbleiter hoher Nennspannung – diese Differenz nimmt allerdings ab. Auch wenn nicht davon ausgegangen werden kann, dass der Preisunterschied vollständig verschwinden wird, ist zu erwarten, dass die Möglichkeit und Wahrscheinlichkeit der Anwendung von Mehrpunktwechselrichtern in solchen Fällen zunimmt, in denen die oben erwähnten Vorteile wichtig sind.

Das Hauptziel der vorliegenden Arbeit ist die Kombination eines Mehrpunktwechselrichters mit einem prädiktiven Regelverfahren, um ein bezüglich Aufwand und Nutzen optimiertes System zur Erzeugung dreiphasiger sinusförmiger Ausgangsspannungen zu erhalten.

Ein Modellbasierter Prädiktivregler (MPC) ist eine sogenannte long-range Prädiktivregelungs-Strategie; der Prädiktionshorizont umfasst mehr als einen Abtastzyklus.

Der bekannteste Nachteil von MPC für mehr als einen Prädiktionsschritt in industriellen Anwendungen ist die Komplexität der Online-Umsetzung, vor allem bei Anwendungen mit schnellen Abtastraten wie in elektrischen Antriebssystemen. Die hierfür notwendige Prozessorleistung steht in industriellen Umrichtern nicht zur Verfügung. Um diesen Nachteil auszugleichen, haben Morari et. al das MPC-Optimierungsproblem zu einem multi-parametrischen quadratischen Program umgewandelt.

Durch die Lösung dieses mp-QP/-LP Problems erhält man die optimale Lösung als PWA[1]-Funktion von Systemzuständen. In jedem Polytop (Region) des gesamten zulässigen Zustandsraums existiert nur eine einzige Optimal-Lösung, die als PWA-Funktion berechnet wird.

Um die Regelung praktisch zu realisieren muss man zuerst bestimmen, in welchem der berechneten Polytope der momentane Systemzustand liegt, und dann das entsprechende affine (lineare) Regelgesetz anzuwenden. Dadurch verringert sich die Implementierung des Optimierungsproblems auf eine einfache Suche in Lookup-Tabellen. Tondel et. al haben eine Strategie (Binärer Suchbaum) vorgestellt, um die eigentlich notwendige vollständige „Enumeration“ und online-Auswertung aller vorhandenen und nicht sortierten Polytope sehr effizient zu verkürzen. Durch diese Strategie ist es möglich, die sich ergebende Lookup-Tabelle auf ein EPROM zu speichern; hierdurch vereinfacht sich die Implementierung des modellbasierten Prädiktivreglers für die schnellen Abtastraten-Anwendungen bei Antriebssystemen. Eine Realisierung mit den derzeit zur Verfügung stehenden Mitteln ist dann möglich.

[1] Piecewise affine (PWA) ist der mathematisch korrekte Begriff – Ingenieurwissenschaftler sprechen meist – etwas ungenau – von stückweise „linearen“ Funktionen.

TABLE OF CONTENTS

TABLE OF FIGURES

TABLE OF TABELS

1 GENERAL INTRODUCTION AND MOTIVATION

The increasing attention paid to the electrical machines, design and control points of views, opens a lot of fields of interest in research. Controlling electrical machines effectively is one of the main tasks encountered in industrial applications. The work presented here focuses on electric drives design and control with one of the most complicated control algorithms, namely predictive control. A lot of investigation has already been done in predictive control; a number of problems, however, are still to be solved. Therefore the research in predictive control for this kind of dynamic and non-linear industrial processes seems to be rather promising.

1.1 Model-Based Predictive Control

Model-based predictive controllers (MPC) with their objective functions and the previous knowledge of the controlled system try to emulate the human thinking in predicting the system behavior and solving the constrained problems optimally. Model-based Predictive Control is an optimal-control based method for constrained feedback control, where the optimization problem is solved at each time step starting from the current state and over a finite horizon called the prediction horizon. With respect to all constraints on states, outputs, and inputs only the first element of the resulting optimal control sequence is applied to the controlled system while the rest is discarded. At the following time step, this computation is repeated with a new state and over a shifted horizon. Therefore, the MPC problem can be addressed as a constrained optimization problem (with a linear, quadratic, or infinite norm) with receding horizon control (RHC) policy. MPC control is considered as the standard and the most popular control approach for the constrained multivariable control problems in process industry and with some limitations in electric drive applications. Predictive control for power electronics is researched since the beginning of the 1980's [62], [63]. Its main difference to the well-known feedback control schemes is the pre-calculation of the system behavior and consideration of this behavior in the control design before a difference between the real value and the reference value really occurs. On the other hand, the feedback control only reacts and tries to correct a control difference already existing. Furthermore, in closed-loop feedback control systems there must also be a decision, whether the system should be optimized to variations of the reference value or to disturbing influences of the system. Any solution based on linear control theory or on closed-loop controls can only be a compromise - it never fulfills optimally all demands at the same time.

With MPC, the pre-calculation and optimization of future control actions is performed using a cost function, which can be defined freely, i.e. optimal future control values can be determined according to minimize the current error. The optimal behavior of the controlled system is determined in advance - either by pre-calculation or by selection of an optimal trajectory. Even if the chosen behavior does not fulfill all demands of the application, the user can be sure that this is the best possible behavior within physical limits of the system. In case of transient events predictive control chooses the output characteristic closest to the reference curve - remaining differences are due to physical limitations (e.g. maximum currents or maximum voltages). In the case of steady state operation predictive control also chooses the output characteristic closest to the wished behavior - remaining differences only depend on the physical limitations of the system, which cannot be overcome. Furthermore, this control strategy fulfills all demands for industrial control, since it allows to model systems with continuous and discrete as well as Boolean variables. This facility makes it possible to control the machine and the voltage source inverter in common using one hybrid structured controller. Hence, controlling the AC machine and the voltage source inverter can be done either in a direct or in an indirect way. In the direct control scheme, the controller generates the inverter switching states directly without need for a modulation strategy. In the indirect control scheme, the generated control inputs are to be modulated before being applied to the inverter, by one of the modulation strategies [39]. The modulator stands in this case to generate the respective inverter switching states. However, indirect MPC control with solving the optimization problem for continuous input variables completely online is time consuming [76].

1.2 The Explicit Solution of Model-Based Predictive Control

The main limitation of MPC to be a standard in electric drive applications is the extensive on-line computation effort needed for the optimization problem, which is increased dramatically with number of inputs and the increased prediction horizon. In [65], the authors proposed multi-parametric programming (MPP) approach as a solution to move the time-consuming part of the model predictive control problem off-line. Solving the MPC problem using the multi-parametric approach, the state variables are treated as free parameters and then the optimization problem is solved for all possible values of these free parameters. Thus, the implicit solution of the MPC optimization problem is converted into an explicit solution, and the state space of the free parameter is divided into several polytopes [67] (regions) and inside each polytope, one single affine/ linear control law being valid. Depending on the Karush-Kuhn-Tucker (KKT) optimality conditions for stability of constrained linear system [66], the resulting control action will be a piecewise affine (PWA) function of the state variables. The explicit solution has the same characteristics concerning control performance and stability of the full on-line solution of the MPC problem. The resulting explicit solution contains all feasible regions in state space and the associated control laws in a look-up-table. To control the plant, there is no optimization problem to be solved online at each sampling time. Rather, the controller simply has to find out in real time, in which polytope the current state-vector is located, and to evaluate the appropriate affine control law. The explicit solution of model-based predictive control (Exp-MPC) for drive applications is introduced in the literature extensively, especially in last three years and for different kinds of machines and for different applications. Furthermore, similar ideas were repeated and applied in different applications for induction machines [100], [106], [107], synchronous machines [101], [103], [104], [99], with field-oriented control scheme (FOC) [76] or direct torque scheme (DTC) [103] with two or multilevel inverters in a direct [81], [100] or an indirect way [105].

One of the main contributions of the work presented in this thesis is reflected in the way of dealing with non-linear terms, appearing in AC machine models [112]. Consideration of these nonlinearities in formulating the model predictive control (MPC) problem with offline optimization were normally ignored [76], [24], [76], [100], considered as an additive disturbance [99], or linearized around some operating points [101], [102], [102]. In this work, nonlinear terms are considered without linearization and compensated either inside or outside the MPC controller structure in different configurations. The cross-coupling and nonlinearities between state variables in the fundamental model of AC machines arise in term of induced voltages. The effect of these voltages on the MPC current controller increases proportionally by accelerating the machine. Therefore, an unloaded induction machine is introduced in this work as a case study. Furthermore, a comparative analysis concerning the compensation procedure of these terms was introduced between the proposed methods for MPC control and other ones in conventional scalar and complex PI controllers. Introduction and implementation of the complex PI controller in this work aims to show the way of complete and active compensation of the motion- and rotor-induced voltages. Such compensation procedure inside the control structure is used with the MPC controller without any simplifications. The validity of the external and internal compensation is further verified and used for direct control of a three-level neutral-point diode-clamped inverter (NPC), where the external compensation with discrete inputs is not possible any more. Controlling the induction machine and the NPC inverter can be done either in a direct or in an indirect way. In the direct control scheme, the controller generates the inverter switching states directly without need for a modulation strategy. In the indirect control scheme, the controller generated control inputs are to be modulated before being applied to the inverter, by one of the modulation strategies. The modulator stands in this case to generate the respective inverter switching states. However, indirect MPC control with solving the optimization problem completely online is time consuming [76]. Therefore, the Exp-MPC is the right choice for an indirect control of AC machines. This approach is much more time efficient than a complete on-line optimization, e.g. generalized predictive control (GPC) [76].

1.3 Direct Model-Based Predictive Control

Some contributions were introduced in the literature for direct MPC control (DMPC) of multilevel inverters. The high number of possible switching states increases the online computation complexity. High time demand to evaluate the cost function for all of inverter switching possibilities limits the evaluation of the implicit DMPC to only one step prediction horizon [119]. In the finite-set control scheme [119], the DMPC controller has to evaluate all 27-switching possibilities of the NPC inverter and chooses the optimal switching state that minimizes a predefined cost function [120]. The linear cost function in the NPC topology aims at minimizing the state-reference tracking error in the next sampling step, switching losses, neutral point potential deviation, or all together. Following this finite-set control scheme to evaluate the cost function with a quadratic norm or with more than one prediction step for multilevel inverters is time critical. The pioneer work introduced in [74] makes the online evaluation of DMPC controller with long range prediction horizon realizable. This control scheme exclude from the cost function evaluation all the switching state sequences that make the predicted current trajectories violate the feasibility bounds. Finally, only the switching sequences that guarantee the problem feasibility for a short switching horizon are evaluated using the cost function. The cost function in [74], [114] intends to minimize the power switch commutations and the tracking error. More information about this control scheme can be found in the publications from T. Geyer [113], [114], [115], [116].

The explicit solution of DMPC control for multilevel inverter in stationary coordinates was not published yet, and it is first introduced in this work. The proposed DMPC model composed of both machine and inverter models with their nonlinearities combined together in one hybrid structured system model. The proposed model can be reformulated to work as well for a two-level inverter. However, the reported results are obtained for three-level NPC inverter only.

1.4 Polynomial Approximation of the Explicit Solution of Model-Based Predictive Control

In the last five years, a significant and an increasing attention has been paid to use the explicit solution of model-based predictive control (Exp-MPC) in a wide range of electric drive applications, where a lot of literatures was published to reduce the online computation complexity of the MPC to make it more feasible for this field of applications [74], [75], [76]; For example, using the binary search tree (BST) strategy [69] to reduce the evaluation time plays a significant role in that direction. Nevertheless, there are still some limitations like memory size and the computation capability of the control set-up, which make Exp-MPC until now not the right choice and the right alternative for different kinds of controllers for AC drive applications [82]. The explicit solution of model-based predictive control becomes more complex when detailed motor models are considered while building Exp-MPC controller. This complexity is reflected in the number of generated regions. Increase in the number of regions requires more memory space to store the explicit solution of MPC as well as the evaluation time of the proper affine control laws.

This work proposes the polynomial approximation of the Exp-MPC using curve fitting strategies as a new contribution to reduce the implementation complexity of Exp-MPC. The proposed method intends to replace the optimal linear/ affine control laws, defined over several regions of the feasible state space, with one polynomial. The polynomial will be multivariate, with higher order, and defined over a combination of feasible regions. Using the polynomial approximation, the memory space needed to store region coordinates and the coefficients of optimal control laws, is reduced significantly. Although the accuracy of the optimal controller is reduced, the polynomial approximation makes the implementation more realistic in cases where original Exp-MPC is impossible or very difficult to implement [82]; provided the stability and feasibility of the problem is still guaranteed.

Complexity reduction of the Exp-MPC using the polynomial approximation, without considering the cross-product terms (i.e.: polynomial terms with multiplications between the polynomial variables of different powers), was introduced in the literature in [119], [122], [123], [124]. The authors in these contributions use the sum-of-squares and the Polya's theorem to find an approximate polynomial inside a stabilization set defined as Polytope. Building the stabilization set depends on the principle of existence of more than one Lyapunov function assure the stability of the optimization problem. On the other hand and due to the expensive symbolic computation of the extended Polya's polynomial, this formulation is just suitable for small problems with some limitations. It works well for small problem size of linear or infinite norm and with guarantee for stability and feasibility of the problem.

The novelty of the work presented here is in introducing the multivariate polynomial approximation using the cross-product terms for drives applications. The simulation results carried on vector controlled motor drive prove the feasibility and the stability of the machine problem with the approximated control laws. This work reformulates the explicit solution of model-based predictive control using least-square curve fitting strategy.

2 ELECTRIC DRIVES APPLICATIONS

2.1 Introduction

The AC machines are the world's most efficient prime movers in the industrial applications. DC machines had occupied this position due to easy and direct control of their flux and torque, and therefore the angular velocity of the rotor shaft. The role and dominance of DC machines in the automated and traction applications started to decline by arising some drawbacks associated with the brushes, mechanical commutation, and the associated maintenance cost. In contrast, three-phase AC machines and especially induction machines (IM) with squirrel-cage rotor are considered as the workhorses of the today's industry; that's because of the low-cost construction and nearly maintenance-free operations. Induction machines are less expensive than DC machines and synchronous machines of the same size, and they work under the synchronous speed in motoring operation mode or above the synchronous speed when generating. On the other hand, the synchronous machine (SM) is also an AC rotating machine, whose speed under steady state condition is proportional to the frequency of excitation in its stator windings. Although the SM's construction is more expensive than the induction, it still preserves its place in the market for its high efficiency. Under the synchronous machine category, permanent magnet synchronous (PMSM) and synchronous reluctance (SyncRel) machines can be distinguished, which are quite popular at low power rating applications. The conventional wound-rotor synchronous machine must have a DC excitation on the rotor, which requires again using brushes and slip-rings. This means rotor losses, brushes maintenance and consequently downtime. Thus, the main reason to use the permanent magnet in the rotor instead of the rotor windings is to overcome these disadvantages of the SM. Therefore the mathematical model of the PMSM is almost the same as for the wound-rotor SM. The rotor windings of the AC machines are either replaced by short circuit in IM with squirrel-cage and in SM with slip-ring, or replaced by permanent magnets in the PMSM. Consequently, controlling the stator currents of these kinds is the only way to control the AC machine torque and flux. However, and due to the coupling effect between the machine currents and their embedded and non-linear relationship, controlling the AC machines' torque and flux cannot be achieved easily; like in the case of the DC machines. This problem is solved by using the field-oriented control strategy (FOC). Following these strategy, the machine phase-currents are to be represented in a virtual coordinate system rotating with one of the machine fluxes (stator, rotor or air gap flux), such that they have a direct effect and control of both the machine flux and torque, independently. Aligning the so-called real axis of the rotating coordinate system (*dq-reference frame*) with space vector of the rotor flux, the stator space vector $\boldsymbol{i}_s$ of the machine currents can be projected on the new coordinate system to get in stationary state two DC current components; the torque producing component i_{sq} and the flux producing component i_{sd}. Both components can be controlled in this control structure independently from each other in a similar way as for the DC machines. Since the rotor speed ω_m of the AC machines is proportional to the stator excitation frequency ω_s, (with some slip frequency in induction machines ω_r), the speed of the AC machine can be controlled by varying the stator frequency $\omega_s=2\pi f_s$. Then, delivering three-phase AC voltages with variable frequency for variable speed applications is possible by using voltage source inverters (VSI) as an intermediate "amplifier" between the supply and the motor, Figure 2.1. The figure shows three-phase AC machine connected to an inverter, which delivers three-phase voltages with an adjustable frequency. The constant-frequency supply from the mains is first rectified with a three-phase rectifier.

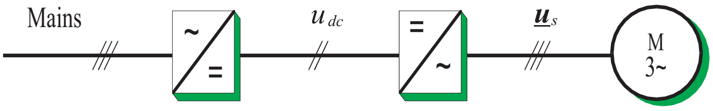


Figure 2.1: Power conversion in variable-speed drive of AC machines

In this chapter, space vector representation of the AC machines is introduced, followed by the time-normalized mathematical models of both IM and PMSM. With short overview of the modulation strategy used in this work, three-level voltage source inverter is finally discussed with more details.

2.2 Space Vector Representation of AC Machines

2.2.1 Overview

Ideal three-phase AC machines have three equally and spatially sinusoidal-distributed windings in the stator and rotor, with 120° intensity shift to each other. Each winding in the stator is a collection of a given number of coils and it is conducted with one of the machine phases. Whereas the rotor structure of AC machines could be either: 1- electrically conducting rotor windings, as in induction machines (with copper or aluminum squirrel cage embedded in iron laminations) 2- or wound-rotor with DC excitation as in SM. By supplying the stator windings of the AC machine with three-phase balanced AC voltages of frequency f_s, sinusoidal-distributed current densities and hence the associated magnetomotive force will be produced in the space airgap of the machine.

Figure 2.4 shows the total current density distribution and the associated current space vector of three-phase AC machine. As a result, a spatially distributed magnetic flux density will be generated in the air gap between the stator and the rotor, and rotates at the stator frequency $\omega_s=2\pi\ f_s$. For an IM, the air gap magnetic flux will cross the short-circuited rotor coils and induce the rotor currents, whenever the angular velocity of the rotor shaft ω_m is different from the angular velocity of the stator field ω_s. Observing these currents from the rotor, they rotate at slip frequency ω_r, where $\omega_r = \ \omega_s - \omega_m$. The induced currents through the rotor conductors will generate the rotor magnetic flux density. The interaction between both magnetic flux densities in the air gap and in the rotor windings generates tangential forces on the rotor surface. Consequently, the force density distribution on the rotor surface produces the electromagnetic torque m_e (under the condition that the rotor shaft and the stator currents velocities are rotating asynchronous to each other). The rotor magnetic flux in a synchronous machine results either from an auxiliary excited DC current in the rotor windings of SM, or from permanent magnets integrated in the rotor structure of PMSM.

2.2.2 AC Machines Modeling and Normalization[2]

The scalar currents i_{sa} , i_{sb} , i_{sc} are the winding currents, whose instantaneous values i_{sa} (t), i_{sb}(t), i_{sc} (t) can be measured at the machine terminals, which produce spatially sinusoidal-distributed current densities inside the machine. The sinusoidal-distributed current densities can be represented mathematically in a complex plane as space vectors[3] $\boldsymbol{A}_{sa}$, $\boldsymbol{A}_{sb}$, $\boldsymbol{A}_{sc}$. The associated current space vectors $\boldsymbol{i}_{sa}$, $\boldsymbol{i}_{sb}$, $\boldsymbol{i}_{sc}$ rotate at the frequency of the stator currents and coincide with the directions of the respective winding axes of those currents, being displaced by 120° from each other (*abc-reference frame*), Figure 2.2 and Figure 2.3.

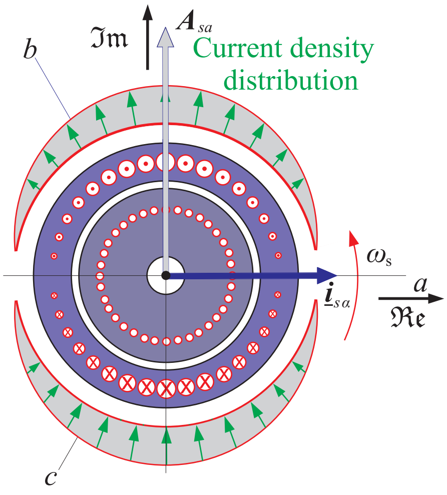


Figure 2.2: Cross-section of an AC machine, with representation of phase ‘*a*’ winding

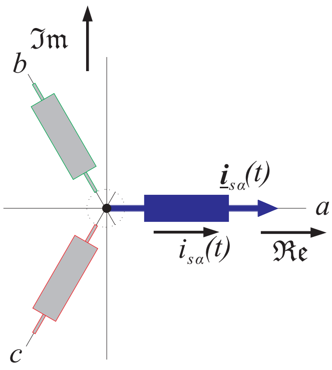


Figure 2.3: Phase ‘*a*’ winding, current, and space vector

In Figure 2.2, the space vector $\boldsymbol{A}_s$ is oriented towards the direction of the maximum current intensity of phase ‘*a*’ and rotates at the phase ‘*a*’ with current frequency ω_s. The half-moon shape in Figure 2.2 represents the increasing intensity of the phase current density in the airgap.

Similarly, the total current density $\boldsymbol{A}_s$ and therefore the associated current space vector $\boldsymbol{i}_s$, as shown in Figure 2.4 and Figure 2.5, is obtained by superimposing the current densities of the three phases, equation (2.1).

[2] All machine quantities introduced in this work are time-normalized quantities. The normalization procedure is explained in the technical report [110] and summarized in the Appendix-B.

[3] The spaces vectors in the figures are underlined extra for more clarity.

$$\boldsymbol{i}_s = \frac{2}{3}\left(\mathbf{1}.\,i_{sa} + \boldsymbol{a}.\,i_{sb} + \boldsymbol{a}^2.\,i_{sc}\right) \tag{2.1}$$

The operator $\boldsymbol{a} = e^{(j2\pi/3)}$ is the so-called the spatial operator, which reflects the spatial location of the windings. The coefficient (2/3) is a scaling factor associated with the coordinate transformation of the machine windings from the three-phase windings into only two orthogonal windings in the complex plane, such that both windings' representation deliver the same physical quantities along the airgap.

Equation (2.1) describes time-varying current vector of the stator windings in the complex plane, which can be graphically derived from the instantaneous values of the phase currents (as shown in Figure 2.5). Under steady state conditions, this vector describes a circular trajectory for increasing time in the stationary reference frame. The transient state is characterized by rapid change in both the magnitude and the angle of the space vector.

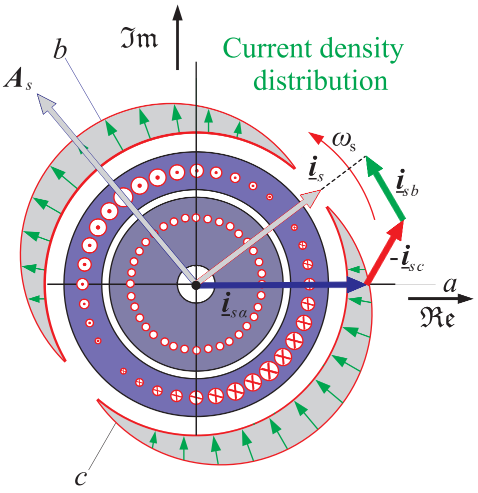


Figure 2.4: Cross-section of an AC machine

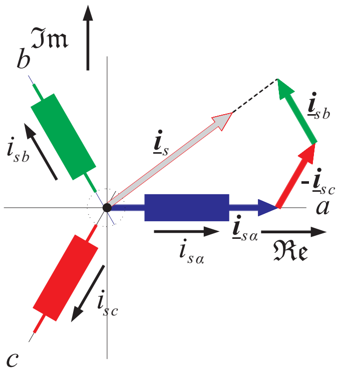


Figure 2.5: Space vector composition of stator currents

The current space vector in the complex plane can be, in general, projected into its orthogonal components, as shown in equation (2.2):

$$\begin{aligned} i_{s\alpha} &= \mathfrak{Re}\{\boldsymbol{i}_s\} = \frac{2}{3}\left(i_{\text{sa}} - \frac{1}{2}i_{\text{sb}} - \frac{1}{2}i_{\text{sc}}\right) \\ i_{s\beta} &= \mathfrak{Im}\{\boldsymbol{i}_s\} = \frac{1}{\sqrt{3}}\left(i_{\text{sb}} - i_{\text{sc}}\right) \end{aligned} \tag{2.2}$$

The magnitudes of these two current components $i_{s\alpha}$, $i_{s\beta}$ depend on the respective windings currents, while their phase angles are fixed. As a result, the total current density in the space may vary in the magnitude and as well as in phase angle. The stator current space vector $\boldsymbol{i}_s$ in Figure 2.7 is shown in the stationary coordinate system (*αβ-reference frame*), having its real axis fixed to the stator winding of phase *a*, and the vector $\boldsymbol{i}_s$ rotating at stator frequency ω_s with respect to the stationary coordinates.

Supplying the stator windings of three-phase AC machine with three-phase balanced ac currents, of 120° phase difference, the resulting total stator current density $\boldsymbol{A}_s$ and therefore the space vector current $\boldsymbol{i}_s$ inside the machine will provide a constant amplitude and rotate at the stator currents frequency ω_s. This space vector representation of the sinusoidal signals can be further extended to all other sinusoidal machine quantities like fluxes and voltages (equation (2.3)).

$$\begin{aligned} \boldsymbol{u}_s &= \frac{2}{3}\left(\mathbf{1}.u_{sa} + \boldsymbol{a}.u_{sb} + \boldsymbol{a}^2.u_{sc}\right) \\ \boldsymbol{\psi}_s &= \frac{2}{3}\left(\mathbf{1}.\psi_{sa} + \boldsymbol{a}.\psi_{sb} + \boldsymbol{a}^2.\psi_{sc}\right) \\ \boldsymbol{a} &= e^{(j\,2\pi/3)} \end{aligned} \tag{2.3}$$

where $\boldsymbol{u}_s$ is the complex representation of the stator voltages in the complex plane. Similarly, $\boldsymbol{\psi}_s$ is the complex representation of the stator flux linkage in the complex plane. Here the flux linkage vector represents the linkage of the magnetic field with the stator windings.

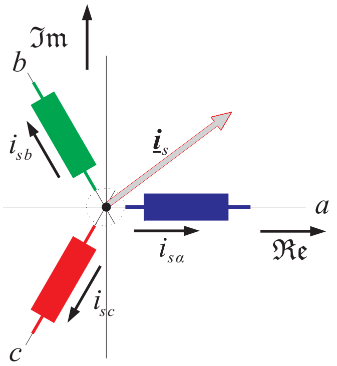


Figure 2.6: Three-phase AC machine currents in the stator windings

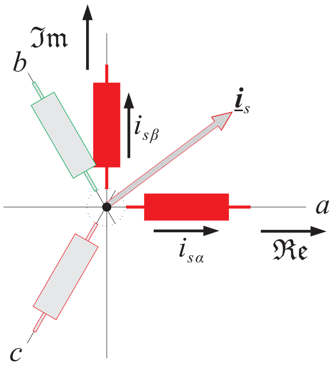


Figure 2.7: The equivalent two-phase currents in the stator windings

The discussion introduced above was done for the stator windings only, and therefore for the stator currents, voltages, and fluxes. Similar discussion is also valid for the rotor, by considering the induced currents in the rotor windings and therefore the resulting rotor flux. Actually, whatever the rotor construction of the AC machines is - solid iron rotor, wound-rotor or permanent magnet - all these structures can be equally substituted by sinusoidal-distributed three-phase rotor windings, in analogue way to the stator structure. Under this condition, the rotor of an induction machine with squirrel cage can be modeled in the same fashion as the stator. Consequently, the stator equations (2.4) for currents, voltages, and fluxes hold for the rotor when the subscript $\square_s$ is being replaced by $\square_r$. Observing these quantities from the rotor, they rotate at the slip frequency ω_r. Hence, the induced currents in the rotor windings of an induction machine can be represented by a space vector $\boldsymbol{i}_r$ in a complex plane and rotates at slip frequency ω_r with respect to the rotor.

$$\begin{aligned}
\boldsymbol{i}_r &= \frac{2}{3}\left(\mathbf{1}.\,i_{ra} + \boldsymbol{a}.\,i_{rb} + \boldsymbol{a}^2.\,i_{rc}\right)\\
\boldsymbol{u}_r &= \frac{2}{3}\left(\mathbf{1}.u_{ra} + \boldsymbol{a}.u_{rb} + \boldsymbol{a}^2.u_{rc}\right)\\
\boldsymbol{\psi}_r &= \frac{2}{3}\left(\mathbf{1}.\psi_{ra} + \boldsymbol{a}.\psi_{rb} + \boldsymbol{a}^2.\psi_{rc}\right)\\
\boldsymbol{a} &= e^{(j\,2\pi/3)}
\end{aligned} \tag{2.4}$$

Here the flux linkage vector $\boldsymbol{\psi}_r$ represents the linkage of the magnetic field with the rotor windings.

The contribution of the rotor field to the stator flux linkage, and vice versa, has not been considered until now in the equations above, and such contribution depends on the machine type and the rotor construction. Under this condition, and taking the induced currents in rotor windings as an example, these currents generate a sinusoidal magnetomotive force on the rotor side of the air gap. Their contribution on the main flux, and hence to the stator flux linkage, is determined by the main inductance l_m , according to the following equations.

$$\begin{aligned}
\boldsymbol{\psi}_s &= l_s\boldsymbol{i}_s + l_m\boldsymbol{i}_r\\
\boldsymbol{\psi}_r &= l_r\boldsymbol{i}_r + l_m\boldsymbol{i}_s
\end{aligned} \tag{2.5}$$

The respective parameters l_s and l_r in the equation represent the stator and rotor inductances, respectively; l_m is the main inductance.

All the space vectors in equation (2.5) are represented in the stationary coordinate system □$_s^{(S)}$. From geometry point of view, the space vectors can be redefined in an arbitrary rotating reference frame □$_s^{(K)}$, rotating with an angular velocity ω_k. Doing that requires a rotation or transformation angle δ_k , which is used to transform the original system coordinates (*αβ-reference frame*) into the new reference frame (*k-reference frame*). Using the transformation angle δ_k , all system variables represented in the original frame can be projected on the new coordinate system as follows:

$$\boldsymbol{i}_s^{(K)} = \boldsymbol{i}_s^{(S)}.e^{-j\delta_k} \tag{2.6}$$

Figure 2.9 shows the new representation of the stator current space vector $\boldsymbol{i}_s^{(S)}$, represented in the complex stationary reference frame, in a new reference frame has the same coordinate's origin and rotated at angular velocity $\omega_k = d\delta_k / d\tau$.

Under the definition of the system coordinate transformation introduced above, the stator voltage in the stationary reference frame (equation (2.7)) can be retransformed to the general *k-reference frame* (equation (2.8)) as follows:

$$\boldsymbol{u}_s = r_s\boldsymbol{i}_s + \frac{d\boldsymbol{\psi}_s}{d\tau} \tag{2.7}$$

The voltage space vectors $r_s\boldsymbol{i}_s$, $d\boldsymbol{\psi}_s / d\tau$ represent the voltage-drops across the stator windings resistance, and the induced voltage, respectively.

$$\boldsymbol{u}_s = r_s \boldsymbol{i}_s + \frac{d\boldsymbol{\psi}_s}{d\tau} + j\,\omega_k \boldsymbol{\psi}_s \tag{2.8}$$

Note that this transformation introduces a new term (equation(2.8)), namely the motion-induced voltage $\left(j\,\omega_k \boldsymbol{\psi}_s\right)$. It is contributed by the rotation of the stator windings with respect to the new coordinate system.

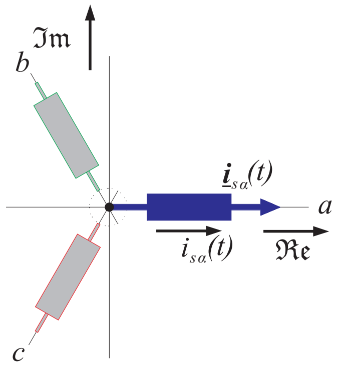


Figure 2.8: Phase-current space vector

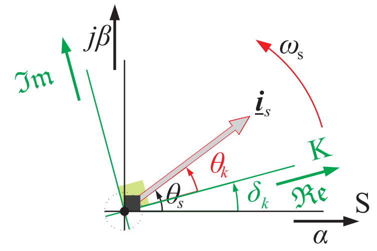


Figure 2.9: Reference frame transformation

The reason for using a different representation of the AC machine than in the three-axis coordinate system (*abc-reference frame*), is that the three axes are not linearly independent of each other, a fact that complicates the mathematically description of the machine dynamics [76]. The stationary coordinate system (*αβ-reference frame*) representation, however, simplifies the description but it does not solve the problem completely. Since the windings in the stator are fixed, and the others in the rotor rotate with rotor-shaft angular velocity, at least one of the windings and then their associated machine variables need to be represented in a rotating reference frame. To solve this problem and assess the interactions between the stator and the rotor of the AC machine, the respective dynamic equations must be represented in a common coordinate system. Therefore, it is preferred to use a general *k*-coordinate system for the derivation of the AC machine equations [50]. The following equations introduce the complex representation of the AC machines in a general *k*-coordinate system:

$$\begin{aligned}
\boldsymbol{u}_s &= r_s \boldsymbol{i}_s + \frac{d\boldsymbol{\psi}_s}{d\tau} + j\,\omega_k \boldsymbol{\psi}_s \\
\boldsymbol{u}_r &= r_r \boldsymbol{i}_r + \frac{d\boldsymbol{\psi}_r}{d\tau} + j\,(\omega_k - \omega_m)\boldsymbol{\psi}_r \\
\boldsymbol{\psi}_s &= l_s \boldsymbol{i}_s + l_m \boldsymbol{i}_r \\
\boldsymbol{\psi}_r &= l_m \boldsymbol{i}_s + l_r \boldsymbol{i}_r \\
m_e &= k_r \left|\boldsymbol{\psi}_r \mathrm{x}\, \boldsymbol{i}_s\right|_z \\
\tau_m \frac{d\omega_m}{d\tau} &= m_e - m_l \\
\omega_r &= (\omega_k - \omega_m)
\end{aligned} \tag{2.9}$$

The parameters r_s, r_r represent the stator and rotor resistance, respectively. The term $k_r = l_m / l_r$ represents the coupling factor of the rotor, τ_m is the mechanical time constant of the machine, and m_l is the (variable) load torque.

From current control point of view, it is preferred to reduce the number of state variables to the stator machine current $\boldsymbol{i}_s$ and rotor field $\boldsymbol{\psi}_r$. From equation (2.9) results:

$$\begin{aligned}
&\boldsymbol{i}_s + \tau_\sigma^{'} \frac{d\boldsymbol{i}_s}{d\tau} = -j\omega_k \tau_\sigma^{'} \boldsymbol{i}_s + \frac{k_r}{r_\sigma}\left(\frac{1}{\tau_r} - j\omega_m\right)\boldsymbol{\psi}_r + \frac{1}{r_\sigma}\boldsymbol{u}_s \\
&\boldsymbol{\psi}_r + \tau_r \frac{d\boldsymbol{\psi}_r}{d\tau} = -j\tau_r\left(\omega_k - \omega_m\right)\boldsymbol{\psi}_r + l_m \boldsymbol{i}_s
\end{aligned} \tag{2.10}$$

In which:

$$\begin{aligned}
&\tau_s = \frac{l_s}{r_s}, \tau_r = \frac{l_r}{r_r}, \sigma = 1 - \frac{l_h^2}{l_s l_r}, l_s' = \sigma l_s, l_r' = \sigma l_r \\
&\tau_s' = \frac{\sigma l_s}{r_s}, \tau_r' = \frac{\sigma l_r}{r_r}, \tau_\sigma' = \frac{\sigma l_s}{r_\sigma}, r_\sigma = r_s + r_r k_r^2 \\
&k_r = \frac{l_m}{l_r}, k_s = \frac{l_h}{l_s}
\end{aligned}$$

The machine quantities in the equations above (2.9) and (2.10) are normalized according to the normalization procedure in the Appendix-B, and the associated normalized machine values are also introduced in the Appendix-A.

This complex notation of the AC machines will be used in the future discussion and analysis.

The Need for a Normalized Machine Model

Applying the normalization procedure [110] on AC machine equations has the advantage that the final implementation of controlled system on the evaluation boards (like DSP, FPGA, dSPACE) will be simplified, more efficient, and all the quantities are expressed in per unit or percent. In addition, collected results and measured signals will be comparable for machines of different sizes, which lead to a fair comparison and understanding to permissible and non-permissible values of different machine variables. Furthermore, formulating the optimization problem in the MPT toolbox (as it will be discussed later in section 5.3) has to be done in a way that all the modeled quantities do not exceed a specific range [-10, 10]. Owing to this limitation of the MPT toolbox, the normalization task on the machine model should be done beforehand. Solving the optimization problem of the Model-Based Predictive Controller (MPC) for a time-normalized machine model intends to use the MPT-toolbox as an effective modeling tool of the MPC problems, irrespective of the machine size.

2.2.3 Induction Machine Modeling

As the squirrel-cage induction machine has short-circuited rotor windings ($\boldsymbol{u}_r = 0$) the rotor voltage equation of this kind of AC machines becomes (2.11):

$$0 = r_r \boldsymbol{i}_r + \frac{d\boldsymbol{\psi}_r}{d\tau} + j(\omega_k - \omega_m)\boldsymbol{\psi}_r \tag{2.11}$$

To fulfill the application requirements, it is useful to describe the respective windings in the stator and in the rotor of an AC machine in a common reference frame. The angular orientation of such coordinate system maybe either fixed to the stator or considered rotating in synchronism with the rotating magnetic fields. Figure 2.10 shows the signal flow graph of an induction machine in a general *k-reference frame*.

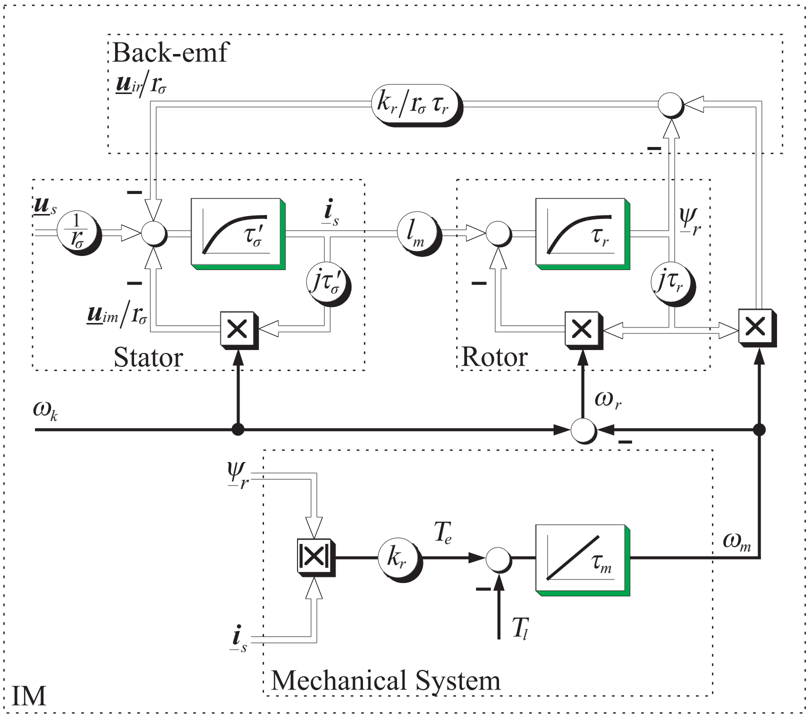


Figure 2.10: Signal flow graph of an induction machine in general reference frame [50]

According to the AC machine model introduced above (equation (2.9)) in a general coordinate system, the mathematical representation of the induction machine can be projected on both stationary as well as synchronous reference frame as follows:

Induction Machine Model in *dq-reference frame*

In this representation, the general *k-reference frame* rotates in synchronism with machine rotor field at ω_s angular velocity. Due to this new coordinate system the stator current space vector $\boldsymbol{i}_s$ can be split into two DC current components: a torque producing current component i_{sq} , and a flux producing current component i_{sd}. Here, both current components can be controlled independently of each other like in the DC machine. The control scenario of the DC current components will be introduced with field-oriented control structure in section 3.3, and with the indirect model-based predictive control in section 6.5. With respect to the rotor field orientation the following assignment $\omega_k = \omega_s , \psi_{rq} = 0$ are set.

$$\begin{aligned}
& i_{sd} + \tau'_\sigma \frac{di_{sd}}{d\tau} = \omega_s \tau'_\sigma i_{sq} + \frac{k_r}{r_\sigma \tau_r}\psi_{rd} + \frac{1}{r_\sigma}u_{sd} \\
& i_{sq} + \tau'_\sigma \frac{di_{sq}}{d\tau} = -\omega_s \tau'_\sigma i_{sd} - \frac{k_r}{r_\sigma}\omega_m \psi_{rd} + \frac{1}{r_\sigma}u_{sq} \\
& \psi_{rd} + \tau_r \frac{d\psi_{rd}}{d\tau} = l_m i_{sd} \\
& 0 = -\left(\omega_s - \omega_m\right)\tau_r \psi_{rd} + l_m i_{sq} \\
& m_e = k_r \psi_{rd} i_{sq} \\
& \tau_m \frac{d\omega_m}{d\tau} = m_e - m_l
\end{aligned} \tag{2.12}$$

Formulating the machine model in the *dq-reference frame* (equation (2.12)) introduces a non-linear coupling effect between the stator machine currents, which makes solving the optimization problem with MPC controller very complex, (see section 6.4.1).

The importance of the equation $0 = -(\omega_s - \omega_m)\tau_r\psi_{rd} + l_m i_{sq}$ appears in getting the rotor model of the machine [117]. This model provides the estimated angular velocity of the rotor field, and hence the coordinate transformation angle δ .

$$\begin{aligned}
&\psi_{rd} + \tau_r \frac{d\psi_{rd}}{d\tau} = l_m i_{sd} \\
&\omega_r = \frac{l_m i_{sd}}{\tau_r \psi_{rd}} \\
&\omega_s = \omega_r + \omega_m \\
&\delta = \int \omega_s \, .d\tau
\end{aligned} \tag{2.13}$$

Induction Machine Model in *αβ-reference frame*

Representing the AC machine model in a coordinate system fixed to the stator, is simply possible by setting $\omega_k = 0$ in the general form of equation (2.9), resulting in:

$$\begin{aligned}
&i_{s\alpha} + \tau_\sigma^{'} \frac{di_{s\alpha}}{d\tau} = \frac{k_r}{r_\sigma \tau_r}\psi_{r\alpha} + \frac{k_r}{r_\sigma}\omega_m \psi_{r\beta} + \frac{1}{r_\sigma}u_{s\alpha} \\
&i_{s\beta} + \tau_\sigma^{'} \frac{di_{s\beta}}{d\tau} = \frac{k_r}{r_\sigma \tau_r}\psi_{r\beta} - \frac{k_r}{r_\sigma}\omega_m \psi_{r\alpha} + \frac{1}{r_\sigma}u_{s\beta} \\
&\psi_{r\alpha} + \tau_r \frac{d\psi_{r\alpha}}{d\tau} = -\omega_m \tau_r \psi_{r\beta} + l_m i_{s\alpha} \\
&\psi_{r\beta} + \tau_r \frac{d\psi_{r\beta}}{d\tau} = \ \ \omega_m \tau_r \psi_{r\alpha} + l_m i_{s\beta}
\end{aligned} \tag{2.14}$$

Here the direct cross-coupling effect between the stator machine currents disappears. Since the model inputs $u_{s\alpha}$, $u_{s\beta}$ of this representation do not need any coordinate transformation before their applying on the inverter, this model is very suitable in the direct MPC control (section 6.8).

2.2.4 Synchronous Machine Modeling

The general dynamic representation of the AC machine in equation (2.9) can be extended to the SM and therefore the PMSM. Considering that the PMSM has embedded permanent magnets $\boldsymbol{\psi}_m$ in the rotor.

$$\boldsymbol{\psi}_s^{(K)} = l_s \boldsymbol{i}_s^{(K)} + \boldsymbol{\psi}_m^{(K)} \tag{2.15}$$

To work in synchronism with a reference frame aligned to the rotor or the rotor field, $\omega_k = \omega_s$ is set.

$$\omega_k = \omega_s = \omega_{me} = P_p \omega_m \tag{2.16}$$

where the P_p refers to the number of the machine pole-pairs.

Accordingly, normalized PMSM equations (2.17) in rotor-fixed reference frame will be:

$$
\begin{aligned}
&\frac{1}{\tau_s}\boldsymbol{i}_s + \frac{d\boldsymbol{i}_s}{d\tau} = \frac{1}{\tau_s r_s}\boldsymbol{u}_s - j\,\omega_{me}\boldsymbol{i}_s - j\,\frac{\omega_s}{\tau_s r_s}\hat{\boldsymbol{\psi}}_m \\
&\boldsymbol{\psi}_s = l_s \boldsymbol{i}_s + \hat{\boldsymbol{\psi}}_m \\
&\hat{\boldsymbol{\psi}}_m = \psi_{md} + j\,0 \\
&l_s = \begin{bmatrix} l_{sd} & 0 \\ 0 & l_{sq} \end{bmatrix} \\
&m_e = \hat{\psi}_{md} i_{sq} + \left(l_{sd} - l_{sq}\right) i_{sd} i_{sq} \\
&\tau_m \frac{d\,\omega_{me}}{d\tau} = m_e - m_l
\end{aligned}
\qquad (2.17)
$$

The signal flow graph Figure 2.11 represents the PMSM model. Just for simplicity of the future discussion of the MPC controller for PMSM in section 6.9, the machine inductances in the new reference frame $l_s = l_{sd} = l_{sq}$ are set.

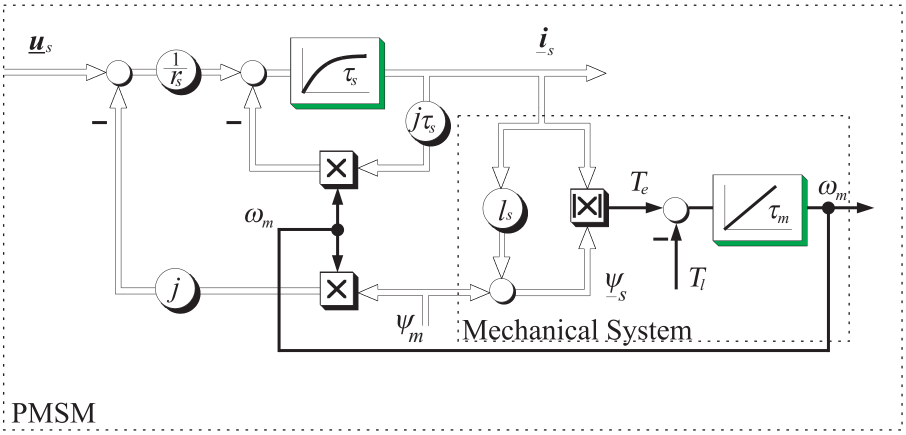


Figure 2.11: Signal flow graph of a PMSM machine in rotor reference frame [53]

2.3 Industrial Inverters

Multilevel inverters have got significant attention from the power electronics industry during the last decade. The main driving force behind multilevel inverter technologies is the ability of these inverters to operate at high voltage and power levels, with significantly reduced harmonics in the output voltage waveforms. Multilevel inverters produce output voltages which are much more similar to sinusoidal waveforms than those of standard two-level inverters. In applications where sinusoidal waveforms are essential (e.g. power supply of rural settlements or farms or electrical drives with low torque ripple requirements), this would save a lot of investments concerning output filters. By having several power switches in series within each inverter leg, the stacked power devices together could carry more voltage. Thus the blocking voltage of each power device in a multilevel inverter can be chosen smaller than in standard two-level inverters. The prices of several power devices for low voltage are still higher than the price for a single high voltage power device. The price gap, however, decreases and even if it will not disappear completely, the application of multilevel inverters becomes more probable due to the advantages mentioned above. The main disadvantages of multilevel inverters in comparison with standard two-level inverters, preventing them from being applied very often in industry, are the higher number of power semiconductors and a more sophisticated control scheme required. Researches on modern voltage source multilevel inverters started with the introduction of neutral point clamped (NPC) three level inverters in 1981 [1], and flying capacitor multilevel inverter in 1992 [2], [16]; for more details about other topologies the author refers the reader to [4]. Although an initial simulation study about the direct model-based predictive control was introduced on the three-level flying-capacitor inverter in [82], this inverter topology is not further discussed in this work. The introduced work was evaluated on a three-level neutral point diode-clamped inverter [27]; therefore the NPC topology is only explained and discussed in the following sections.

2.3.1 Neutral Point Diode-Clamped Inverters

Generally speaking, an *m-level* diode-clamped inverter consists of *m-1* series-connected DC-bus capacitors and produces *m* different levels of the output voltages at its outputs. Active power switches and clamping diodes in NPC topology, however, have same voltage ratings, namely u_{dc}*/(m-1)*. The associated free-welling diodes with the power switches serve to sharing the DC-link voltage to equal values, across the turned-off power switches.

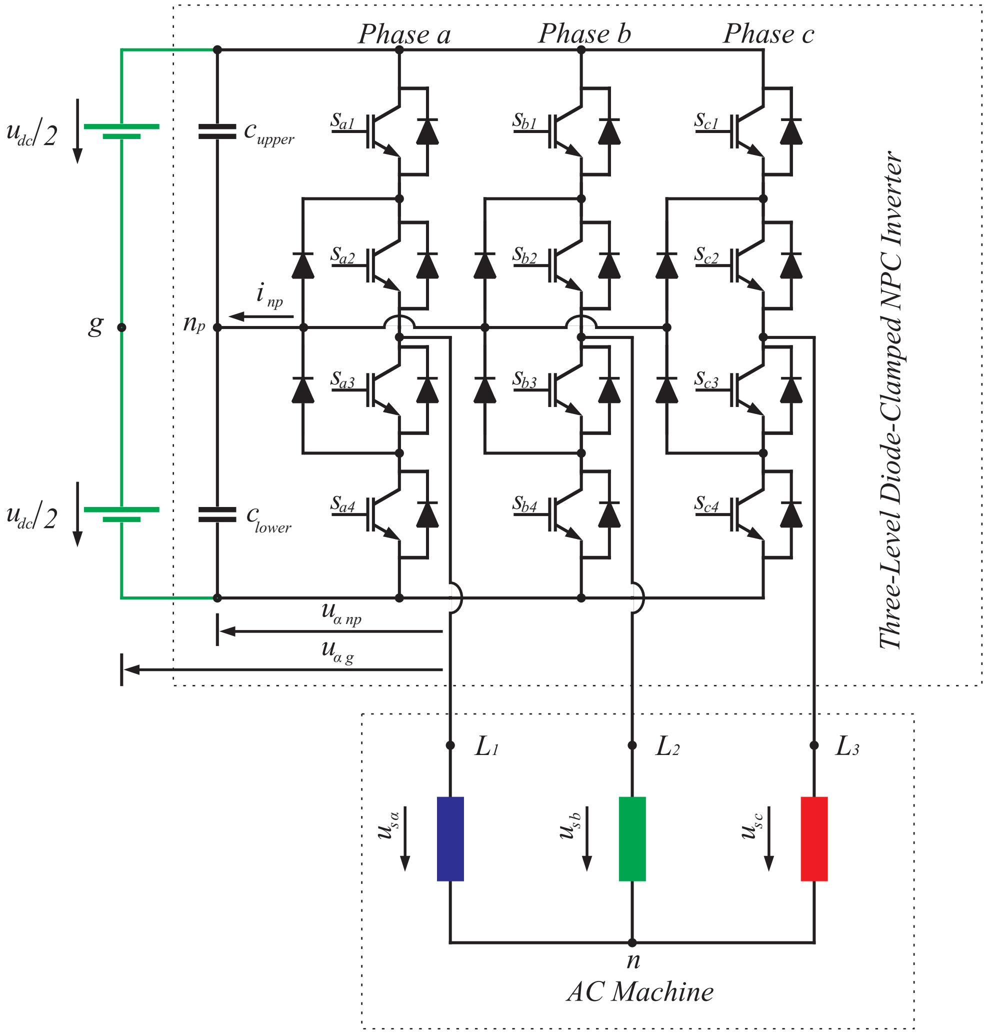


Figure 2.12: Circuit diagram of a three-level NPC inverter connected to an AC machine

Figure 2.12 shows a circuit diagram of three-level diode-clamped inverter connected to a three-phase AC machine. The inverter is fed by a DC source of voltage u_{dc} ; the DC-link voltage is normalized according to the voltage base value, see Appendix-B. The power semiconductors of the Insulated-Gate Bipolar transistor (IGBT) type are used in the designed three-level NPC inverter [27]. The provided DC-link at the inverter input is split into three voltage levels ($+u_{dc}/2$, 0, or $-u_{dc}/2$) using a series bank of capacitors (c_{upper} and c_{lower}), Figure 2.13. The inverter control gives the possibility to connect machine terminals to one of these three voltage levels ($+u_{dc}/2$, 0, or $-u_{dc}/2$), Table 2.I, thus generating three-level switched phase voltages. The power switches are controlled through triggering signals at their gates. The upper and lower power switches in each phase leg/ bridge are turned on and off in a predefined switching sequence and at given time instants. Actually, not all possible switching combinations are allowed in this topology, as shown in Table 2.I. However, the power switches in each inverter leg are to be switched in a complementary way. The complementary function is defined such that turning on one of the switches of the complementary pair will exclude the other from being turned on. This principle of work has to be projected in the implementation stage by keeping a sufficient interlock time while turning on the switches, to avoid a possible short circuit. For example, in the phase leg a, the switch s_{a1} has to be switched in a complementary way to the switch s_{a3}; and also s_{a2} has to be switched in a complementary way to s_{a4}. It follows, only three switching combinations are allowed in each phase leg. Having three legs in the NPC inverter leads to a total number of $3^3 = 27$ different arrangements of the switches are possible. Each switching arrangement ($s_a\ s_b\ s_c$) stands for a respective switching state voltage vector (e.g.: $\boldsymbol{u}_Z$, $\boldsymbol{u}_A$, $\boldsymbol{u}_B$, …) in Figure 2.14. The output voltage space vectors generated by the inverter are defined by:

$$\boldsymbol{u}_s = \frac{2}{3}\frac{u_{dc}}{2}\left(\mathbf{1}.s_a + \boldsymbol{a}.s_b + \boldsymbol{a}^2.s_c\right) : s_{a,b,c} \in \{-1,0,+1\},\ \boldsymbol{a} = e^{j2\pi/3} \tag{2.18}$$

where the variables s_a , s_b , and s_c denote the instantaneous value of the selected switching state of each phase.

The switching state vectors of the three-level NPC inverter are shown in Figure 2.14. The associated notation of the switching state-vector $\boldsymbol{u}_J = \boldsymbol{u}_J^{(0+-)}$ for example, indicates that the phase a is connected to the neutral point (n_p), the phase b is connected to the positive DC-rail, and the phase c is connected to the negative DC-rail. The Table 2.I summarizes the allowed switching states and the resulting phase voltage level, with respect to the neutral point n_p voltage as a reference.

Table 2.I: Switching possibilities of a three-phase NPC inverter

s_{a1}	s_{a2}	s_{a3}	s_{a4}		s_a	u_a
On	On	Off	Off		+1	$u_{dc}/2$
Off	On	On	Off		0	0
Off	Off	On	On		-1	$-u_{dc}/2$

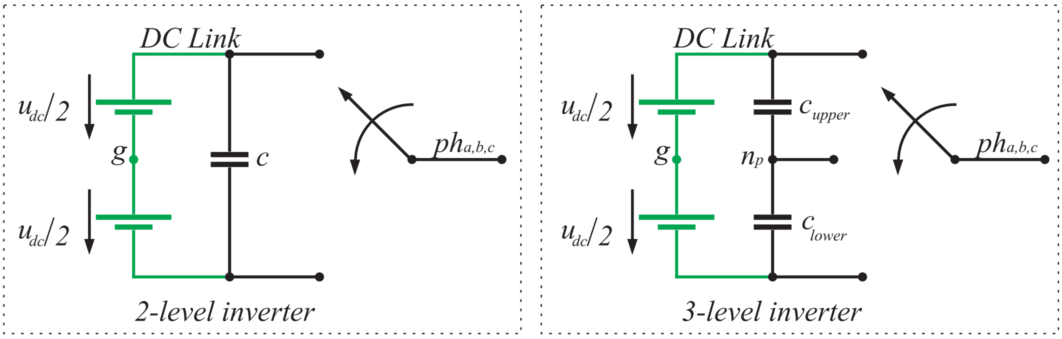


Figure 2.13: Phase voltage switching possibilities

a) for a 2-level inverter b) for a 3-level inverter

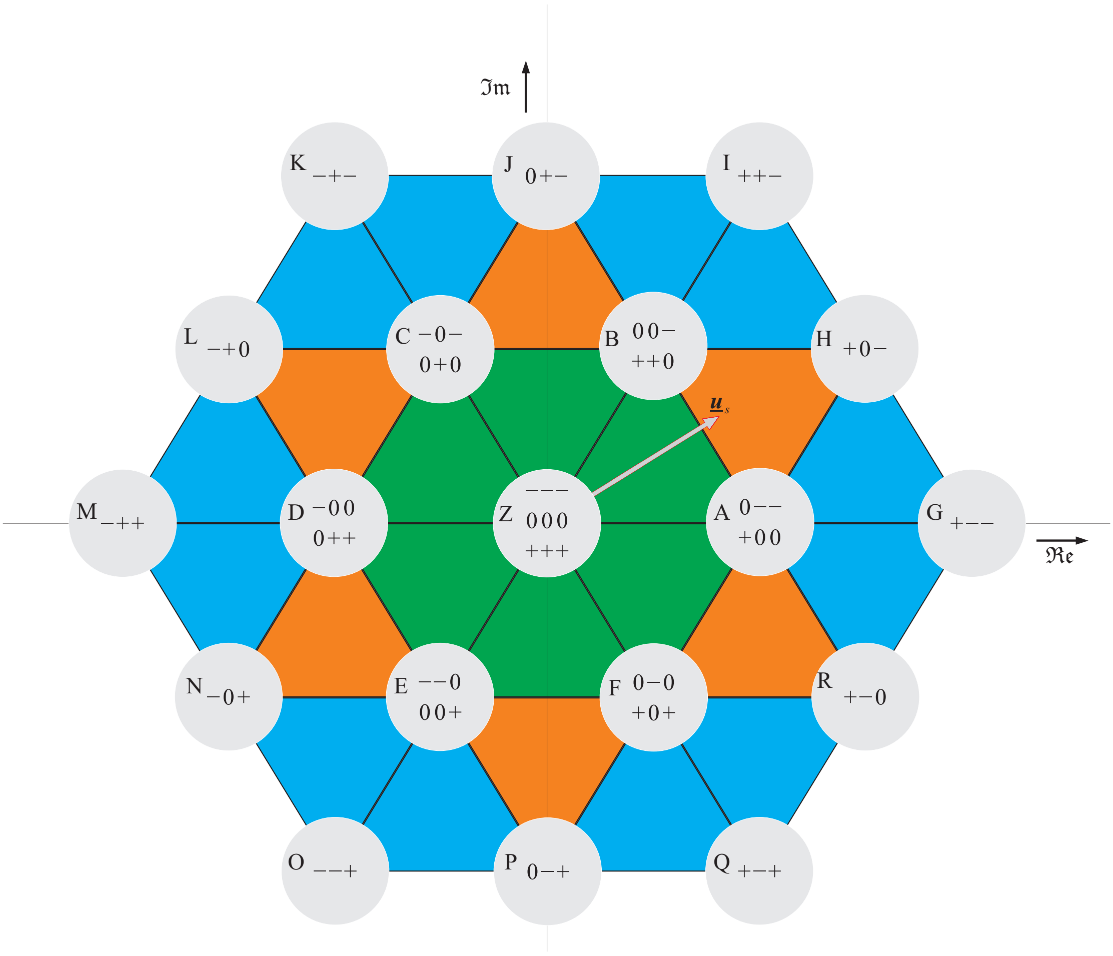


Figure 2.14: Switching state combinations of a three-level NPC voltage source inverter

2.3.1.1 Three-Level Inverter Modulation Strategies

The reason behind using pulse width modulation (PWM) techniques in drive applications is to get fundamental voltage waveforms at the outputs of the inverter with different amplitudes and frequency. Thus, the machine can be driven at different speed levels using a DC voltage source. The PWM modulator generates three-phase switched voltage waveforms whose fundamental components equal in amplitude and frequency to the voltage references. The voltage references result from the used controllers, and then they are sampled at double the switching frequency, $f_{sampling} = 2.\ f_{sw}$. The switching frequency can be defined as the number of commutations of each power switch within a fundamental period multiplied with the respective fundamental frequency. Pulse width modulation strategies can be classified into open and closed-loop control methods. The open loop modulation strategies are the most popular ones, whereas the closed-loop techniques are very complex and used in high performance drive applications [3]. In the following only the two most widely used open loop PWM schemes for multilevel inverters are introduced:

- Carrier-based sinusoidal pulse width modulation (SPWM) technique is very popular open loop modulation method in the industrial applications. For n level inverter, the SPWM involves comparisons of the three reference voltage signals with n-1 level-shifted carriers [17]. The carrier signals could be in phase or shifted from each other with different resulting harmonic spectrum. However, different contributions were presented in [17] depending on the phasor relationship between the carrier signals. It has been shown in [20] that the phase deposition (PD) modulation generates superior harmonic spectrum with less harmonic components. For a three-level inverter, two carrier signals arranged in an identical phase disposition of an f_{sw} frequency are involved in the SPWM modulation strategy. The modulation procedure starts by comparing the sampled reference signals with the carrier ones, Figure 2.15.

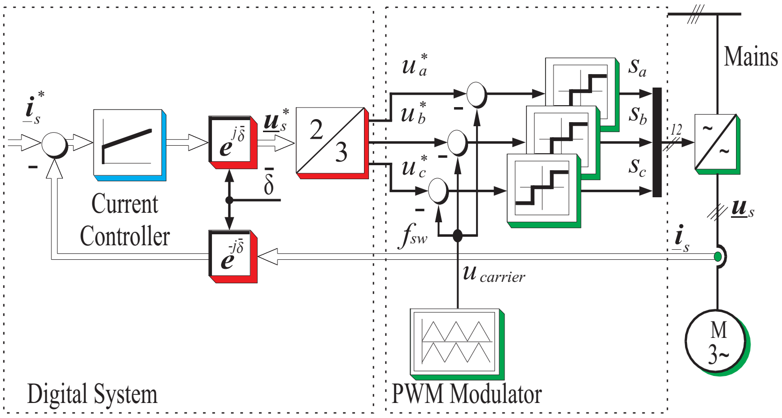


Figure 2.15: Carrier-based sinusoidal pulse width modulation, signal flow graph

The power semiconductors of the inverter in each leg are turned on and off according to the respective comparator outputs. Extension the linear range of SPWM is possible by adding an offset voltage to the three reference signals, which does not produce a phase voltage distortion or affect the load average currents [57]:

$$u_{offset} = -(u_{max} + u_{min})/2 \tag{2.19}$$

where u_{max} is the maximum amplitude among the three reference signals, while the u_{min} the minimum amplitude.

An offset addition makes the SPWM equivalent to the apace vector modulation SVM [57], [15] in both the maximum linear modulation range and the resulting switching sequence for two-level inverters. Actually it is not the case for multilevel inverters; for more information refers to [15]. For multilevel inverters, this addition helps only in increase of the maximum linear operation range of SPWM, but it will not result in the same switching sequence.

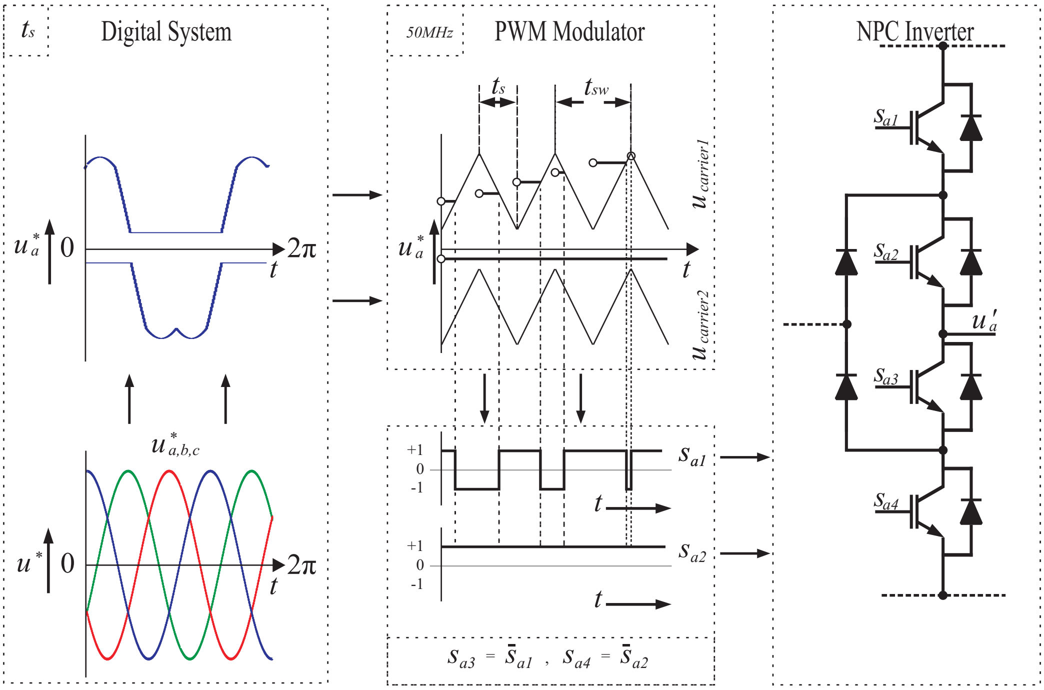


Figure 2.16: Carrier-based sinusoidal PWM procedure for a three-level NPC inverter

- Space vector modulation (SVM) is based on the space vector representation of the three reference voltages as a space vector $\boldsymbol{u}^*$ in stationary reference frame. The rotating reference voltage vector is sampled at regular time intervals T_0.

$$\int_{T_0} \boldsymbol{u}^* dt = \int_{t_a} \boldsymbol{u}_a dt + \int_{t_b} \boldsymbol{u}_b dt + \int_{t_c} \boldsymbol{u}_c dt$$
$$T_0 = t_\mathrm{a} + t_\mathrm{b} + t_\mathrm{c} \qquad (2.20)$$

which equate the volt-second value of the reference voltage sample to that of the switching sequence of a sub-cycle T_0.

Each allowed switching state possibility of the inverter corresponds to a specific output voltage vector. A three-phase three-level NPC inverter provides 19 active switching state vectors, while the remaining vectors are either zero or redundant vectors. The active vectors divide the plane in 24 triangular regions, Figure 2.14. Being the reference voltage vector in one of these regions, then only the adjacent switching state vectors are elected to compose the desired reference vector $\boldsymbol{u}^*$, Figure 2.17. The respective on-duration times are obtained by solving the equation (2.20). Working in low modulation range (i.e. in the inner hexagon), only the small voltage vectors (e.g. $\boldsymbol{u}_A$, $\boldsymbol{u}_B$, and $\boldsymbol{u}_Z$) are used in the switching sequence. Due to the redundancy of the small vectors, different possibilities of the switching state sequence can be introduced to produce the same voltage vector, equation (2.21). However, their effects on the neutral point potential are different. Increase of the modulation index to work in the outer hexagon involves medium and the large vectors to be used in composing the reference vector and here also only the nearest vector are recommended, Figure 2.17.

$$S \equiv u_Z \langle t_0 / 2 \rangle \; ... \; u_A^{+/-} \langle t_a \rangle \; ... \; u_B^{+/-} \langle t_b \rangle \; ... \; u_Z \langle t_0 / 2 \rangle$$

$$\begin{aligned} & e.g.: u_B^{+} = u_B^{(++0)} \;, \; u_B^{-} = u_B^{(00-)} \\ & T_0 = t_0 + t_a + t_b \end{aligned} \tag{2.21}$$

It can be seen that the order of the switching states in the switching sequence and the use of the nearest space vectors help to reduce the commutation between two consecutive switching states and therefore the switching losses.

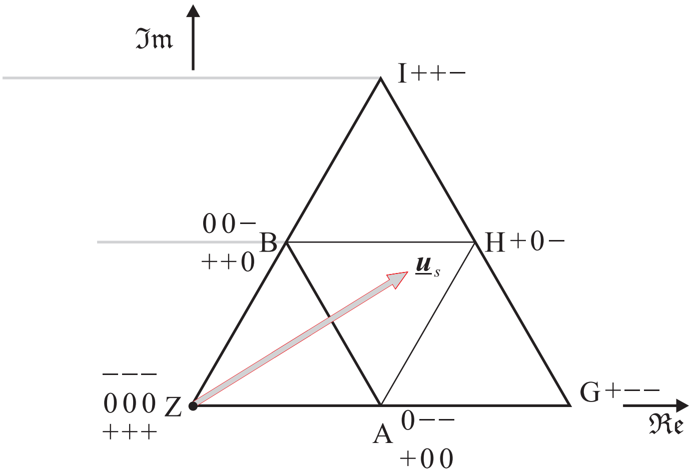


Figure 2.17: Space vector modulation, regional division

2.3.1.2 Neutral Point Balancing Mechanism

The DC-capacitors of the three-level inverters serve as a voltage divider of the full DC-link (u_{dc}). Generally, there is no physical connection between the two points (g) and (n_p) which helps to stabilize the capacitor voltages, thus the neutral point potential floats. Neutral point (n_p) potential value (u_δ) changes in proportion to the integral of the neutral point current (i_{np}). Value and sign of the neutral point current are determined by the applied switching state to the inverter and the measured phase currents (i_{sa} , i_{sb} , and i_{sc}), equation (2.22). By considering the neutral point current flowing into the neutral point (n_p) it follows [5]:

$$\begin{aligned} & u_\delta = u_{c_{lower}} - u_{c_{upper}} \\ & \frac{du_\delta}{d\tau} = \frac{1}{c_d} i_{np} \\ & i_{np} = (i_{sa} |s_a| + i_{sb} |s_b| + i_{sc} |s_c|) \end{aligned} \tag{2.22}$$

As the outputs of the inverter might be connected to the neutral point (n_p), connecting the inverter to unbalanced load (as the case in transients or connecting two of the machine windings to the neutral point) leads to an unbalanced voltage division on the DC-capacitors (c_{upper} and c_{lower}). The unbalanced voltage-division between the DC-capacitors causes an application of more voltage stress on the active power switches and on the DC-capacitors. High voltage excursions affect the actual voltages on the machine terminals and cause either increase or decrease of the commanded voltages, whenever the neutral point potential error is exist. It follows that the respective machine currents will be also affected and will not follow their reference signals. Such interaction and dependency relationship could lead to an unstable operation of the drive system. Therefore a special attention has to be paid here to maintain the neutral point potential in a tolerance band, around 1.4% of the full DC-link voltage [13].

The voltage space vectors, in Figure 2.14, associated with the respective switching state, can be divided into four groups, according to their effects on the n_p-potential:

1. Three zero vectors ($\boldsymbol{u}_Z$), do not cause a current flow throughout the neutral point, and hence its potential is not affected, e.g. $\boldsymbol{u}_Z^+ = \boldsymbol{u}_Z^{(+++)}$.

2. Twelve small vectors, available in six redundant pairs ($\boldsymbol{u}_A$, $\boldsymbol{u}_B$, $\boldsymbol{u}_C$, $\boldsymbol{u}_D$, $\boldsymbol{u}_E$, and $\boldsymbol{u}_F$), positive $\boldsymbol{u}^+$ and negative $\boldsymbol{u}^-$ vectors. Each small vector of a redundant pair has reverse effect on the n_p-current, as shown in Table 2.II. Although the n_p-current has the same magnitude for both small vectors, its sign reverses depending on the selected redundant vector, e.g., both small vectors $\boldsymbol{u}_B^+ = \boldsymbol{u}_B^{(++0)}$ and $\boldsymbol{u}_B^- = \boldsymbol{u}_B^{(00-)}$ have the same voltage magnitude u_B and opposite effect on the n_p-current and therefore n_p-potential, $i_{np\,B}^{+} = -i_{np\,B}^{-} = -i_c$. Small voltage vectors that connect machine terminals between the positive DC-rail and the neutral point are called positive vectors. These vectors could increase or decrease the unbalancing by charging the upper DC-capacitor. The other ones connecting the machine terminals between the neutral point and the negative DC-rail are called negative small vectors with opposite effect on the n_p-potential. So applying the respective switching states of these small vectors alternately helps to keep the DC-capacitors balanced. The majority of the proposed n_p-balancing schemes relies on the redundant small vectors and applies some form of manipulations of these vectors in a pair to keep the n_p-potential in some limit. The relative on-duration of positive and negative small vectors in a pair is usually controlled to eliminate the n_p-potential error completely.

3. Six medium vectors ($\boldsymbol{u}_H$, $\boldsymbol{u}_J$, $\boldsymbol{u}_L$, $\boldsymbol{u}_N$, $\boldsymbol{u}_P$, and $\boldsymbol{u}_R$), generate n_p-current in the neutral point and change n_p-potential by connecting one of the inverter outputs to this point, making the n_p-potential dependent in a part of the loading condition e.g.$\boldsymbol{u}_H = \boldsymbol{u}_H^{(+0-)}$. As these vectors do not have redundant pairs, they could be considered besides the small vectors as a new source of the capacitor voltage ripple and reason of unbalancing in the outer hexagon. As only one inverter output is connected to the neutral point and the others are connected to positive and negative DC-rails, the effect of the medium vectors comparing with the small vectors is minor and could be compensated by using the small vector redundancy properly, section 6.8. For that reason, medium vectors could be referred to as uncontrollable components and the small vectors are controllable components.

4. Six large vectors ($\boldsymbol{u}_G$, $\boldsymbol{u}_I$, $\boldsymbol{u}_K$, $\boldsymbol{u}_M$, $\boldsymbol{u}_O$, and $\boldsymbol{u}_Q$), do not cause a connection of the neutral point to any of the inverter outputs and therefore no n_p-current flows. Thus the neutral point potential is left unaffected, e.g. $\boldsymbol{u}_K = \boldsymbol{u}_K^{(-+-)}, i_{npK} = 0$.

Thus, a nonzero n_p-potential error (u_δ) and therefore n_p-current (i_{np}) exists whenever either a small or a medium vector is a part of the applied switching sequence. Figure 2.18 shows some examples of the switching combinations of three-level NPC inverter and their impact on the neutral point.

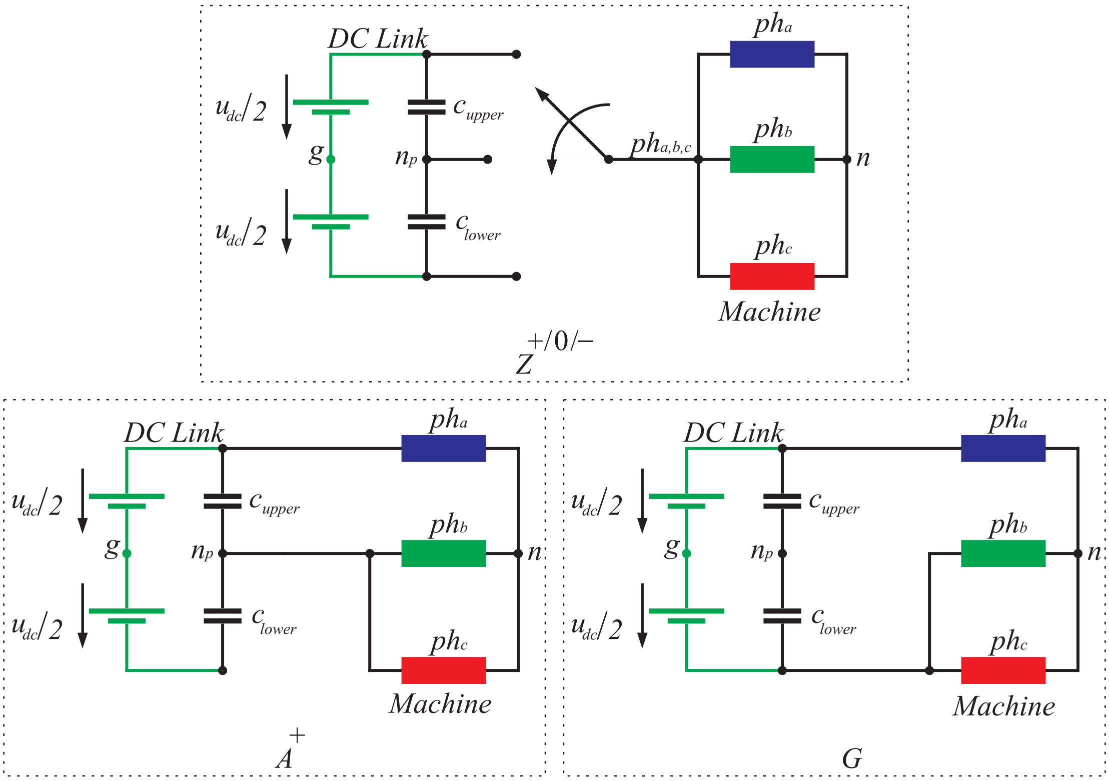


Figure 2.18: Circuit topology of a three-level NPC inverter for one leg

a) Zero vector b) Small vector d) Large vector

Table 2.II: The effect of small vectors on the neutral point current

Positive Small Vectors		i_{np}	Negative Small Vectors		i_{np}
$\boldsymbol{u}_A^+$	+ 0 0	i_{sa}	$\boldsymbol{u}_A^-$	0 – –	$-i_{sa}$
$\boldsymbol{u}_B^+$	+ + 0	$-i_{sc}$	$\boldsymbol{u}_B^-$	0 0 –	i_{sc}
$\boldsymbol{u}_C^+$	0 + 0	i_{sb}	$\boldsymbol{u}_C^-$	– 0 –	$-i_{sb}$
$\boldsymbol{u}_D^+$	0 + +	$-i_{sa}$	$\boldsymbol{u}_D^-$	– 0 0	i_{sa}
$\boldsymbol{u}_E^+$	0 0 +	i_{sc}	$\boldsymbol{u}_E^-$	– – 0	$-i_{sc}$
$\boldsymbol{u}_F^+$	+ 0 +	$-i_{sb}$	$\boldsymbol{u}_F^-$	0 – 0	i_{sb}

2.3.1.3 Control of the Neutral Point Potential

The investigation done in [5] on three-level NPC inverter with space vector modulation [3] and using the alternative selection of the redundant small vectors proves the natural balancing of the DC-capacitors in steady state. The reason behind this natural mechanism is that an existing of a neutral point potential error causes generating a positive n_p-current while the n_p-potential is positive, and vice versa [13]. It results in a normal operation, the average n_p-current drawn from the neutral point over one modulation cycle is zero and the n_p-potential remains constant. Furthermore, paper [5] addressed two main factors affect the capacitor balancing in steady state: The impedance of the connected load, and spectra of the switching functions in frequency domain. Hence, the natural balancing mechanism of three-level inverters is valid in the steady state and any sudden change in the applied voltage space vector creates deviations of the neutral point potential. As the resulting potential error could be accumulated to an extreme overvoltage, a fast balancing control strategy should be applied. Figure 2.19 shows the gradual elimination of n_p-potential error by exploiting the natural balancing mechanism of a three-level NPC inverter. This result was collected on an induction machine working at 75% of its nominal speed. A carrier-based SPWM modulator was used to modulate the outputs of the PI current controllers in (1).

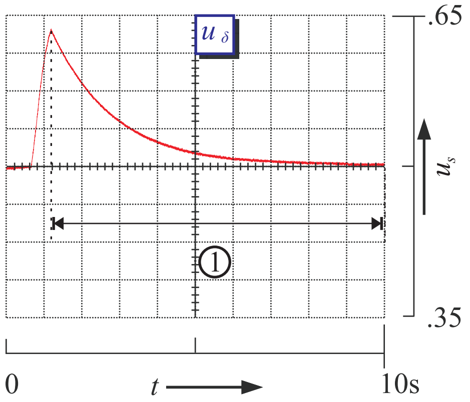


Figure 2.19: Natural balancing of a three-level NPC inverter

The following comparative discussion considers mainly the space vector modulation strategy for its dominance and simple explanation of its effect on the neutral point potential. According to that, different approaches can be distinguished:

- Exploit the intrinsic natural balancing mechanism of the three-level NPC inverters and enhance this mechanism by adding a small passive balancing circuit [8], [5]. This circuit in [5], for example, provides a low-impedance path for harmonics at half the sampling frequency. Another simple possibility could be done to reduce the neutral point potential fluctuation by enlarging the value of the DC-capacitors above the standard value given in [6]. Both methods increase the hardware cost but reduce the programming effort and simplify the control structure.

- A passive control of the n_p-potential: The positive and negative small vectors are applied alternatively in each new sampling cycle and irrespective of having or not an n_p-potential error. This method would have difficulties to recover from a sudden load change [18].

- Hysteresis type control: The controller is activated when the n_p-potential error violates the limits of the provided tolerance band. According to the used modulation strategy two cases can be distinguished here:

    - In carrier-based SPWM modulation and depending on the sign of the n_p-potential and the direction of the n_p-current, a zeros sequence voltage component is either added to or subtracted from the phase reference voltages. This addition/subtraction of the zeros sequence voltage inverts the direction of the n_p-current and keeps the n_p-potential within the limits [18], [7], [19], [22].

    - In conventional SVM approach the output reference voltage is generated by utilizing the nearest triangle vectors. Based on the effect of the positive and negative small vectors of the triangular work region on the direction of the n_p-current, the most appropriate small vector can be selected to keep the n_p-potential error in the provided band.

  This method gives good results for a small n_p-potential error up to 1.5% of the DC-link [18], [13]. However, acquiring the n_p-current direction has proved to be problematic at low switching frequency, where the current is highly distorted [21]. Therefore making the decision depending on the measurements of DC-capacitor voltages is preferred [12].

- Active control scheme: In this control scheme the controller reacts to the n_p-potential error by controlling the on-durations of the used redundant switching state vectors [9], [10], e.g.:

$$\begin{aligned} S \equiv\ & \boldsymbol{u}_A^{(0--)}\left\langle t_\mathrm{a}/2\right\rangle ... \boldsymbol{u}_B^{(00-)}\left\langle t_\mathrm{b}\right\rangle ... \boldsymbol{u}_Z^{(000)}\left\langle t_0/2\right\rangle ... \boldsymbol{u}_A^{(+00)}\left\langle t_\mathrm{a}/2\right\rangle ... \\ & \boldsymbol{u}_A^{(+00)}\left\langle t_\mathrm{a}/2\right\rangle ... \boldsymbol{u}_Z^{(000)}\left\langle t_0\right\rangle ... \boldsymbol{u}_B^{(00-)}\left\langle t_\mathrm{b}\right\rangle ... \boldsymbol{u}_A^{(0--)}\left\langle t_\mathrm{a}/2\right\rangle \end{aligned} \tag{2.23}$$

  This switching state sequence contains two small vectors $\boldsymbol{u}_A^{(0--)}$, $\boldsymbol{u}_A^{(+00)}$ with opposite effect on the n_p-potential and therefore on the n_p-current direction. Therefore changing the on-duration (t_a) of the applied redundant pairs $\boldsymbol{u}_A^{+/-}$ affects the n_p-potential. Controlling the on-durations could be further applied to the carrier-based SPWM method by adding/ subtracting a DC-offset to the phase reference voltages to decrease/ increase the on-durations, respectively [7]. The benefit of these schemes is in their capability to balance the neutral point exactly. Generally, these schemes require measuring the n_p-potential, n_p-current, and amplitude of phase voltages [11].

  However, acquiring the phase current polarities has proved to be problematic when the high current distortion exists at low switching frequency [21]. Furthermore, in consequence of using the small vectors and their redundant in each switching cycle, these schemes increase the switching loses. Maximum possible time extension of this method however is limited to the minimum on-duration of the semiconductor devices, equation (2.20).

- Sub-bridge selection method: It could be classified under “Hysteresis type” control scheme with some improvements and modifications on the proposed space vector modulation presented in [3], [10]. This method is suitable in the space of the inner hexagon, where only small and zero vectors form a part of the switching sequences.

The proposed method in [13] exploits the existence of two redundant sub-bridges in three-level inverters of the NPC topology. It modifies the proposed method in [3] and suggests two new switching sequences, equation (2.24). The first switching sequence S^{+} uses only the semiconductor devices of the positive sub-bridge (positive small vectors). The second switching sequence S^{-} uses only the semiconductor devices of the negative sub-bridge (negative small vectors).

$$\begin{aligned} S^{+} \equiv\, & \boldsymbol{u}_Z^{(000)}\langle t_0/2\rangle \ldots\ \boldsymbol{u}_A^{(+00)}\langle t_{\boldsymbol{a}}\rangle \ldots\ \boldsymbol{u}_B^{(++0)}\langle t_{\boldsymbol{b}}\rangle .. \boldsymbol{u}_Z^{(+++)}\langle t_0\rangle \ldots\ \boldsymbol{u}_B^{(++0)}\langle t_{\boldsymbol{b}}\rangle \ldots \\ & \boldsymbol{u}_A^{(+00)}\langle t_{\boldsymbol{a}}\rangle \ldots \boldsymbol{u}_Z^{(000)}\langle t_0/2\rangle \\ S^{-} \equiv\, & \boldsymbol{u}_Z^{(000)}\langle t_0/2\rangle \ldots \boldsymbol{u}_B^{(00-)}\langle t_{\boldsymbol{b}}\rangle\ \ldots \boldsymbol{u}_A^{(0--)}\langle t_{\boldsymbol{a}}\rangle\ \ldots \boldsymbol{u}_Z^{(---)}\langle t_0\rangle .. \boldsymbol{u}_A^{(0--)}\langle t_{\boldsymbol{a}}\rangle\ \ldots \\ & \boldsymbol{u}_B^{(00-)}\langle t_{\boldsymbol{b}}\rangle\ \ldots \boldsymbol{u}_Z^{(000)}\langle t_0/2\rangle \end{aligned} \tag{2.24}$$

Each sub-bridge selection has an opposite effect on the n_p-current, and the alternated selection of the positive and negative sub bridges leads to a n_p-potential balancing. The control is activated when the neutral point potential violates its limit by using the particular sub-bridge as many as needed to eliminate the error. As both switching sequences start and end with the same switching state $\boldsymbol{u}_Z^{(000)}$, there are no additional commutations required at the transient. Furthermore, the order of switching states in the proposed sub-bridge assures minimum commutations; where there is only one commutation is required between two consecutive switching states. Thus the maximum possible reduction of the commutations and hence the switching loses inside the inner hexagon can be achieved by following this method.

In this work it was sufficient to use large DC-capacitors to reduce the effect of the unbalancing on the behavior of indirect controllers[4]; especially when the used sampling frequency is around 10 kHz. However, working at low and variable switching frequency, as the case with direct predictive controller[5], an increase of the capacitor sizes is not sufficient and a fast balancing control method should be developed. The proposed balancing algorithm for direct model-based predictive control of a three-level NPC inverter is reported in section 6.8.

2.4 Conclusion

A brief description of the complex representation of AC machines was introduced in this chapter. Complex machine models for both induction and permanent magnet synchronous machines were provided and explained to be used in the following chapters. The used inverter in this thesis has been built under collaboration and support from Stellenbosch University in South Africa to work as a neutral-point diode-clamped (NPC) inverter or as a flying-capacitor (FC) inverter; according to the application goal. In this thesis the inverter was driven to work as a NPC inverter and its functionality was verified by different applications and algorithms. The design procedure of the three-level NPC inverter is explained in details with all the schematic, layout, and source codes (VHDL, C) in [27].

[4] The terminology “Indirect control” is used to refer to the need of a PWM modulator to drive the inverter.

[5] The terminology “Direct control” is used to refer to the possibility of driving the inverter without need for a PWM modulator.

3 VARIABLE SPEED CONTROL OF AC MACHINES

3.1 Introduction

Based on the space vector representation of the sinusoidal machine quantities (section 2.2) the variable speed control methods of AC machines can be distinguished into scalar and vector control [57]. In scalar control only the magnitude and the angular frequency of stator voltages, currents, and flux linkage space vectors are controlled. Under steady state conditions, these vectors describe a circular trajectory with time, and rotate with a constant stator angular frequency. Thus scalar control does not respond optimally to the transient variation of these vectors or their angles. In contrast, vector control does not only influence the magnitude and the angular frequency of these quantities, but also their instantaneous positions. Therefore the vector control provides correct orientation for both steady states and transients of the space vectors. The speed control methods introduced in the next sections were experimentally verified for the induction machine, whose parameters are introduced in the Appendix-A, connected to a three-level NPC inverter [27].

3.2 Open Loop V/F Control

The open loop speed control based on *v/f* principle [56], [57] is a popular method in industrial applications for its simplicity; where there are no requirements related to fast dynamic response are required. This is practically the case, when no rapid speed change is required or no sudden load torque is applied. Therefore it is usually used for variable speed pumps, fans and ventilators. In this control scheme there is no need for correct orientation of the machine quantities (space vectors) and it is to keep magnitude of the flux space vector (in the equations (2.8)) constant at its rated value and rotates at the stator angular frequency. This condition can be satisfied by keeping the applied machine voltage and frequency, in steady state, at constant rate, *v/f = const.*

$$\begin{aligned} &\boldsymbol{u}_s = r_s \boldsymbol{i}_s + j\,\omega_s \boldsymbol{\psi}_s \\ &: \frac{d\,|\boldsymbol{\psi}_s|}{d\,\tau} = 0 \end{aligned} \tag{3.1}$$

Equation (3.1) is based on the assumption that the flux amplitude is constant in steady state operation.

The normalized space vector magnitude of stator voltage can be calculated as [57]:

$$\left|\boldsymbol{u}_s\right| = \sqrt{\left(r_s i_s\right)^2 + \left(\omega_s \psi_s\right)^2} \tag{3.2}$$

For normalized constant stator flux linkage $\boldsymbol{\psi}_s = 1$ [pu], the relation between the magnitude of the applied reference voltage and its angular frequency is given in the following linear equation:

$$G = \frac{\left|\boldsymbol{u}_s^*\right|}{\omega_s^*} = 1 \tag{3.3}$$

Here the stator resistance was set $r_s = 0$, and G is the voltage generating function, proportionally to the reference angular frequency.

For practical implementation the voltage drop across the stator resistance has to be considered, especially at very low frequency. A boost voltage (around 0.05 [pu]) was supplied here allowing the flux linkage to be available at its rated value and thus the resulting machine torque is kept constant.

$$\left|\boldsymbol{u}_s^*\right| = u_o + \omega_s^* \tag{3.4}$$

The boost voltage u_o stands for compensating the stator resistive voltage drop $r_s \boldsymbol{i}_s$.

The block diagram of open loop *v/f* control scheme is shown in Figure 3.1.

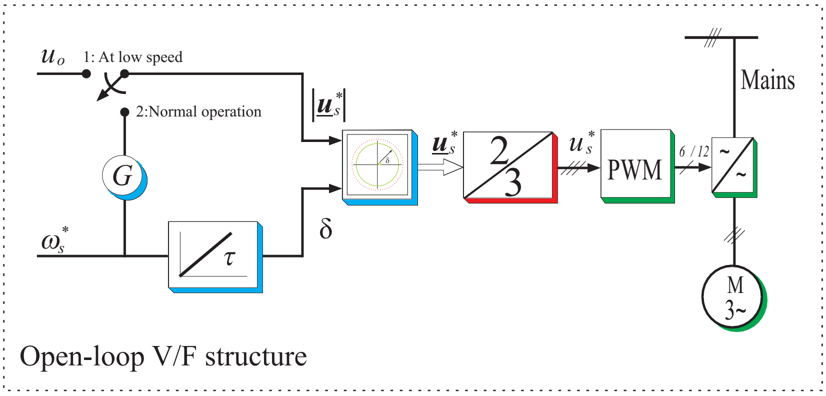


Figure 3.1: Signal flow graph of *v/f* control scheme

The block [2/3] in Figure 3.1 refers to the coordinate transformation from *αβ-reference frame* to *abc-reference frame*.

The control algorithm calculates the reference voltage amplitude, proportional to the command speed value, and its angle is obtained from the speed signal integration (Figure 3.2). Hence the mechanical speed command ω_s^* determines both the amplitude $|\boldsymbol{u}_s^*|$ and angular frequency $\omega_s^* = 2\pi f_s^*$ of the applied voltage vector $\boldsymbol{u}_s^*$. In this control scheme there is no feedback signal (open loop control structure), and hence both the machine speed and currents are left uncontrolled during transient. To avoid undesirable high slip values during the transient, a ramped speed signal has to be used as a reference. Figure 3.4 shows the machine current under different speed conditions.

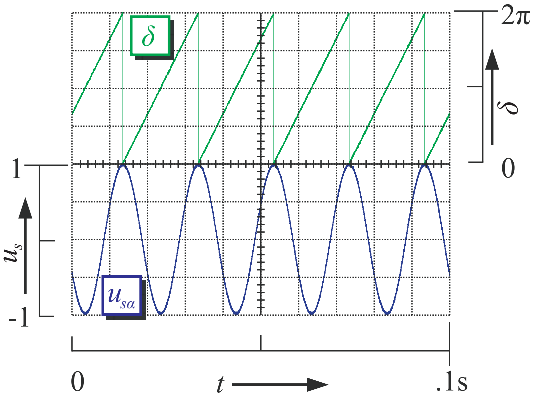


Figure 3.2: IM phase voltage $u_{s\alpha}$ with the associated angle δ

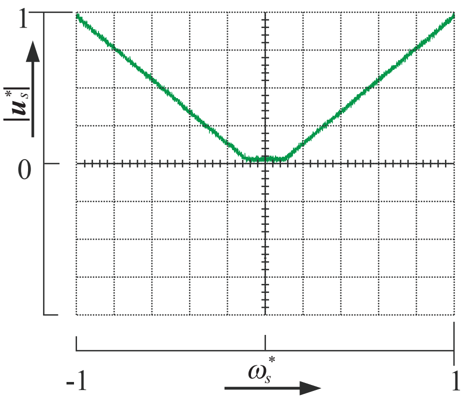


Figure 3.3: IM reference voltage and speed, $\left|\boldsymbol{u}_s^*\right|$ vs ω_s^*, in XY-coordinates

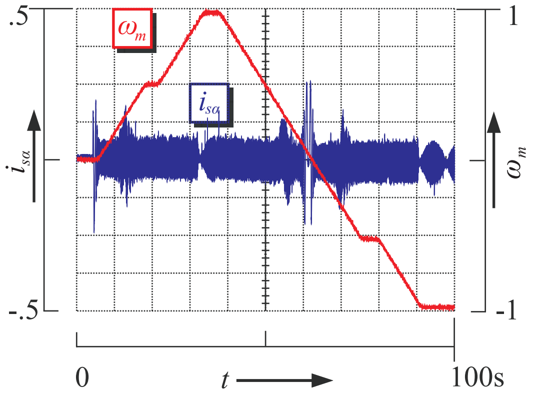


Figure 3.4: IM current $i_{s\alpha}$ and the mechanical speed ω_m, by changing the speed reference ω_s^*

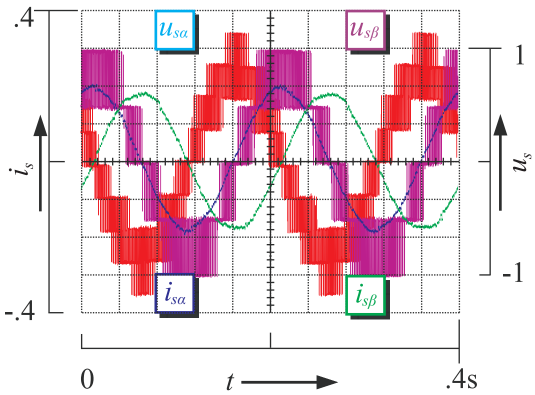


Figure 3.5: IM currents ($i_{s\alpha}$, $i_{s\beta}$) and voltages ($u_{s\alpha}$, $u_{s\beta}$) in stationary coordinates

For an induction machine connected to a three-level NPC inverter, the Figure 3.5 shows the machine currents and voltages in *αβ–reference frame* at full speed.

The experimental results introduced in this section evaluate the control algorithm and prove the high efficiency of the designed digital system (according to chapter 4) and the three-level NPC inverter [27]. The results presented in [26] were collected for the same induction machine connected to a two-level inverter.

3.3 Vector Control

As mentioned before, in vector control using the space vector representation of the AC machines, and regardless the selected reference frame, each of amplitude, angular velocity and position of the space vectors are to be controlled in both steady state and in transients. Such control arises in high-performance drive applications, where the controller should respond to the fast changes of the machine speed and torque. These requirements are fulfilled by imposing a precise closed-loop control on the stator current space vector $\boldsymbol{i}_s$. Actually, the stator currents, voltages, and the rotor angular velocity are the main machine quantities that can be sensed without need for high cost measuring tools; while this is not the case for other rotor quantities and the machine torque, which are to be estimated if necessary. Controlling the machine quantities in stationary reference frame means that the reference voltage $\boldsymbol{u}_s^*$, generated at the output of the current controller, alternates sinusoidally with the time and then the used PI current controller will deliver a steady state error. Thus, using the PI current controller in the stationary coordinates is not really preferred. In contrast, the MPC controller, as it will be discussed later, could achieve this task effectively [38]. Going further with PI controller requires getting constant current components, which can be only obtained for AC machines by representing the machines quantities in rotating reference fame. To remedy this problem the author proposed a PI controller in [55] in rotating coordinate system with number of coordinates transformation tasks. The rotated reference frame is usually chosen to rotate in synchronism with one of the machine fluxes, with some advantages and disadvantages of the methods over each other's [57]. The most popular and well-known method is by choosing the reference frame rotates in synchronism with the rotor flux at angular frequency ω_s. Following this alignment, the projection of the stator current space vector $\boldsymbol{i}_s$ on the reference frame axes delivers two current components (i_{sd} , i_{sq}). Since the selected reference frame rotates at the same angular frequency ω_s of the stator current space vector $\boldsymbol{i}_s$, the scalar PI (SPI) current controllers then receive DC-current components (i_{sd} , i_{sq}), and hence a zero current error in steady state could be achieved. Controlling the machine currents in this way introduces to *field-oriented control structure*. The *field-oriented control structure* is kind of cascaded control form, where the current controller is located in the inner control loop, followed by speed and flux control loops, and if necessary with a position control loop, Figure 3.6.

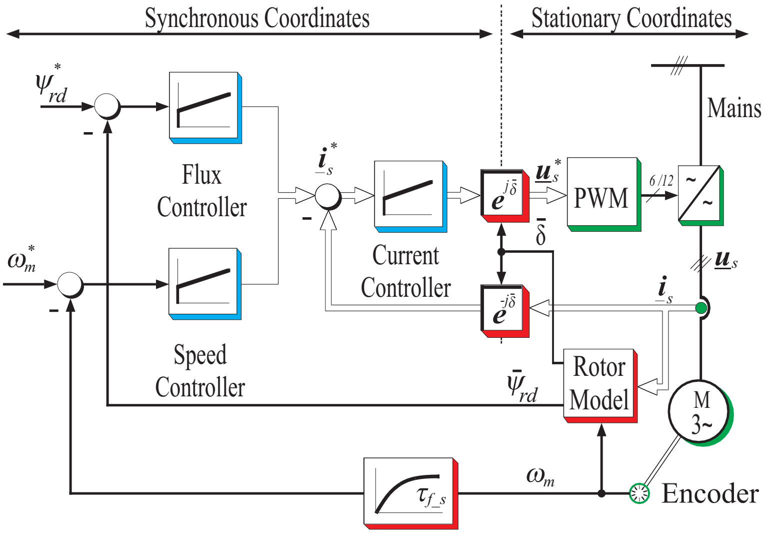


Figure 3.6: Signal flow graph of field-oriented control scheme

In the current control loop two scalar PI (SPI) controllers in a Single-Input Single-Output (SISO) form are separately used to control both the machine current components (i_{sd} , i_{sq}), and hence a full control for the associated produced terms can be achieved, namely the machine flux and torque. For flux control and instead of using an extra explicit flux control loop, necessary to work in field-weakening region, a steady state reference value for the flux producing component i_{sd} of the stator current vector $\boldsymbol{i}_s$ is used. The assigned current value

$i_{sd} = |\boldsymbol{\psi}_{sN}|/l_m$ assures that the machine is always excited with the nominal machine flux $\boldsymbol{\psi}_{sN}$. In the following, the design procedure of the current and speed PI controllers are introduced. To have a fair comparison with the proposed MPC control structure, both conventional and complex PI current controllers are introduced in follows.

3.3.1 Modeling the System Delay

The system delay caused by the inverter switching as well as the used digital system is a common element in all drive applications. In this section the modeling procedure of this delay element is introduced for its passive effect on the dynamic performance of the closed control loop [52]. Actually, the system delay is related to the inverter switching frequency f_{sw} used in the modulation procedure; i.e. driving the inverter at low switching frequency increases the effect of time delay on the system stability, and vice versa. Therefore, the time delay compensation of this element should be done beforehand in the controller design stage.

Using carrier-based SPWM technique [39] to modulate the calculated reference voltage $\boldsymbol{u}_s^*$, the current controller does not have an immediate effect on the stator machine currents, owing to the modulation time delay τ_{inv}. Furthermore the machine response to the applied control effort will be available on the controller input at the next sampling time only; i.e. one-sampling time delay caused by the time–discrete signal processing τ_{sys}. The equivalent time constant τ_d compromises both the time delay caused by the inverter switching $\tau_{inv} = 0.5/2f_{sw}$ and the time delay caused by the digital system $\tau_{sys}=1/2f_{sw}$, hence $\tau_d= \tau_{inv}+ \tau_{sys} = \frac{1.5}{2f_s} = 1.5\,\tau_{sampling}$, where f_{sw} is the normalized switching frequency and $\tau_{sampling} = 1/2f_{sw}$ is the normalized sampling time.

The system-delay is conventionally approximated by a first-order delay element, equation (3.5):

$$\boldsymbol{u}_s^{(S)} = \frac{\boldsymbol{u}_s^{*(S)}}{\left(1+\tau_d \boldsymbol{s}\right)} \tag{3.5}$$

where $\boldsymbol{u}_s^{*(\mathrm{S})}$ is the reference voltage vector generated by the current controller in stationary reference frame.

If the current controller is designed in field coordinates the above formula is to be also transformed accordingly by multiplying both sides with $e^{(-j\omega_s\tau)}$, equation (3.6):

$$\boldsymbol{u}_s^{(F)} = \frac{\boldsymbol{u}_s^*}{\left(1+j\,\omega_s\tau_d + \tau_d \boldsymbol{s}\right)} \tag{3.6}$$

The imaginary term ($j\,\omega_s\,\tau$) results from the representation of the delay element in the field coordinates, which is usually omitted by designing the current controller using the conventional PI controller. Furthermore, in systems operate at switching frequency much higher than the eigenfrequencies of the machine; system delay compensation is usually omitted in the current control design [54]. Such simplification is not possible when driving the inverter at low switching frequency [52], where the delay element has a dominated effect on the closed control loop. The complex form of system delay in equation (3.6) is used for designing the complex PI controller, in section 3.3.3.

3.3.2 Conventional PI Controller

Controlling the stator current space vector $\boldsymbol{i}_s$ of the induction machines, whose parameters are introduced in Appendix-A, is done in *field-oriented control scheme*. By representing the machine equations in *dq-reference frame*, equation (3.7), arises a nonlinear coupling relationship between the current components terms (motion-induced voltage $-j\,\omega_s\,\tau$) and an input-disturbance term (represented by the rotor-induced voltage $\left(\,k_r(1/\tau_r - j\omega_m)\psi_r\right)$). The rotor induced voltage depends on two main quantities: rotor time constant τ_r and mechanical speed ω_m. Here the reaction of the rotor to the changes in the stator current i_s is delayed by large rotor time constant τ_r. This means that the rotor flux vector does not follow the fast changes that the stator current i_s may take. Also the rotational speed ω_m changes at a lower rate than the stator current i_s. Therefore the rotor-induced voltage can be considered as disturbance added to the command voltage $\boldsymbol{u}_s^*$ and considered in later step. The motion induced voltage $\left(-j\,\omega_s\,l_\sigma\,i_s\right)$ in the stator windings is also considered as a disturbance. As this voltage changes as fast as the stator current does, its influence on the system dynamic in transient is much higher than the rotor induced voltage and it should be compensated and added to the output of the current controller.

$$\left(\tau_\sigma' \boldsymbol{s} + 1\right)\begin{bmatrix} i_{sd} \\ i_{sq} \end{bmatrix} = \left(\frac{1}{r_\sigma}\right)\left(\begin{bmatrix} u_{sd} \\ u_{sq} \end{bmatrix} - \overbrace{\begin{bmatrix} 0 & -l_\sigma \omega_s \\ l_\sigma \omega_s & 0 \end{bmatrix}}^{motion-induced\ voltage} \cdot \begin{bmatrix} i_{sd} \\ i_{sq} \end{bmatrix} - \overbrace{\begin{bmatrix} -\dfrac{k_r}{\tau_r}\psi_{rd} \\ k_r \omega_m \psi_{rd} \end{bmatrix}}^{rotor-induced\ voltage}\right) \tag{3.7}$$

Hence, the compensation of the rotor induced voltage could be done at the output of the controller as an input-disturbance without affecting current controller dynamic. Whereas the compensation of the current cross-coupling terms requires an accurate decoupling procedure.

Decoupling Procedure:

Designing the decoupling network [58] for such a system (equation (3.7)) intends to remove the coupling-effect from one system output on the others, and to have the possibility to design the controllers for each system output independently from the other controllers.

The mathematical representation of AC machines in *dq-reference frame* (equation (3.7)) leads actually to a Multi-Inputs Multi-Outputs system (MIMO) with cross-coupled effects between the machine currents.

$$\frac{d}{d\tau}\begin{bmatrix} i_{sd} \\ i_{sq} \end{bmatrix} = \begin{bmatrix} -\dfrac{1}{\tau_\sigma'} & \omega_s \\ -\omega_s & -\dfrac{1}{\tau_\sigma'} \end{bmatrix} \cdot \begin{bmatrix} i_{sd} \\ i_{sq} \end{bmatrix} + \begin{bmatrix} \dfrac{1}{l_\sigma} & 0 \\ 0 & \dfrac{1}{l_\sigma} \end{bmatrix} \cdot \left(\begin{bmatrix} u_{sd} \\ u_{sq} \end{bmatrix} - \begin{bmatrix} -\dfrac{k_r}{\tau_r}\psi_{rd} \\ k_r \omega_m \psi_{rd} \end{bmatrix}\right) \tag{3.8}$$

The cross-coupling between the stator currents is represented by the off-diagonal elements in equation (3.8). As it has small percentage effect on both currents, the controller stability will be not be heavy affected and only decentralized control can be here used; which is the usual case. In de-central control, the cross-coupling term and for its small effect on the system outputs can be ignored and only two main Single-Input Single-Output current controllers (SISO), based on the diagonal elements, are here necessary $F_m(\boldsymbol{s}) = \dfrac{i_{sd,q}}{u^*_{sd,q}} = \dfrac{1}{r_\sigma\left(\tau_\sigma' \boldsymbol{s} + 1\right)}$

: *F_m(s)* represents the transfer function of the machine stator-model. But for perfect currents' decoupling in transient, two more decoupling-controllers are required to eliminate/ compensate the effect the cross-coupling terms (*Decoupling Control*). The following equation (3.9) represents the machine model in frequency domain by neglecting the rotor-induced voltage.

$$\begin{aligned} i_{sd}(\boldsymbol{s}) &= \frac{1}{r_\sigma\left(\tau'_\sigma \boldsymbol{s}+1\right)}\left[u_{sd}(\boldsymbol{s})+l_\sigma \omega_s i_{sq}(\boldsymbol{s})\right] \\ i_{sq}(\boldsymbol{s}) &= \frac{1}{r_\sigma\left(\tau'_\sigma \boldsymbol{s}+1\right)}\left[u_{sq}(\boldsymbol{s})-l_\sigma \omega_s i_{sd}(\boldsymbol{s})\right] \end{aligned} \tag{3.9}$$

Generally speaking, MIMO systems can be represented in one of the following forms, *P or V canonic structure*, as shown in Figure 3.7. Each of them preserves some characteristics make it more favorable in some applications and not in others. For example, *V-form*, in Figure 3.7-b, can be easily extracted from the mathematical machine model in equation (3.9), Figure 3.7-a. This structure, in usual case, is very close to the real system and comes up to the system physical meaning. In contrast, *P-form*, Figure 3.7-c, is far from the physical interpretation of the real system, but on the other hand its formulation is identical to the general matrix formulation of MIMO systems, equation (3.10). This makes it most popular form for the decoupling procedure. Furthermore, the *V-structured decoupling network*, Figure 3.8, is more efficient than the *P-one*.

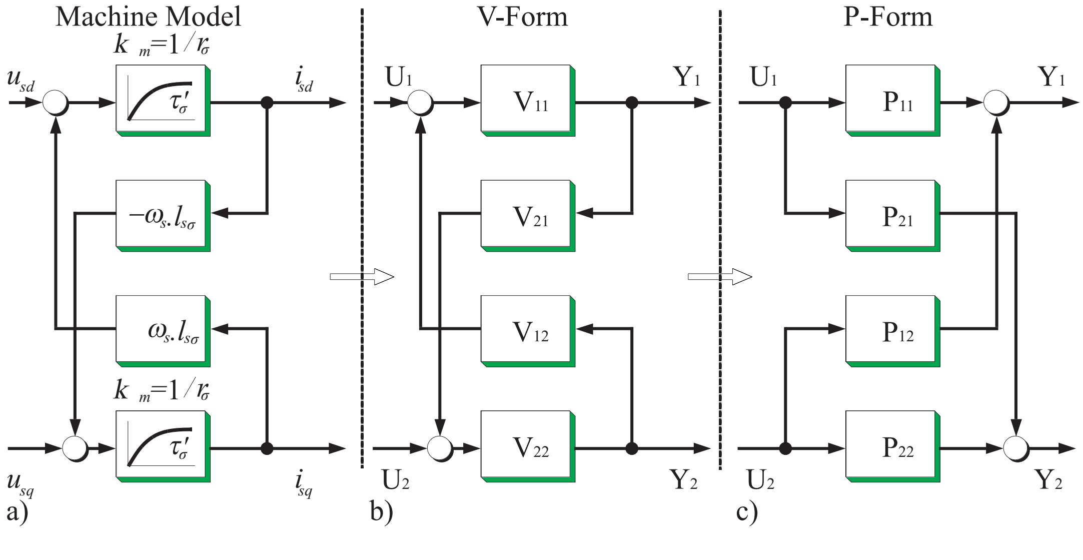


Figure 3.7: AC machine model representation using canonical forms

The general transfer function form for a MIMO system in matrix formulation is given as:

$$\begin{bmatrix} Y_1(s) \\ Y_2(s) \end{bmatrix} = \begin{bmatrix} F_{11}(s) & F_{12}(s) \\ F_{21}(s) & F_{22}(s) \end{bmatrix} \cdot \begin{bmatrix} U_1(s) \\ U_2(s) \end{bmatrix} \tag{3.10}$$

where $F_{11}(s)$, $F_{22}(s)$ represent direct transfer functions in frequency domain between the input U_1, U_2 and the output signals Y_1, Y_2, respectively, $F_{12}(s)$, $F_{21}(s)$ represent the coupling effects of the input signals U_2, U_1 on the output signals Y_1, Y_2, respectively.

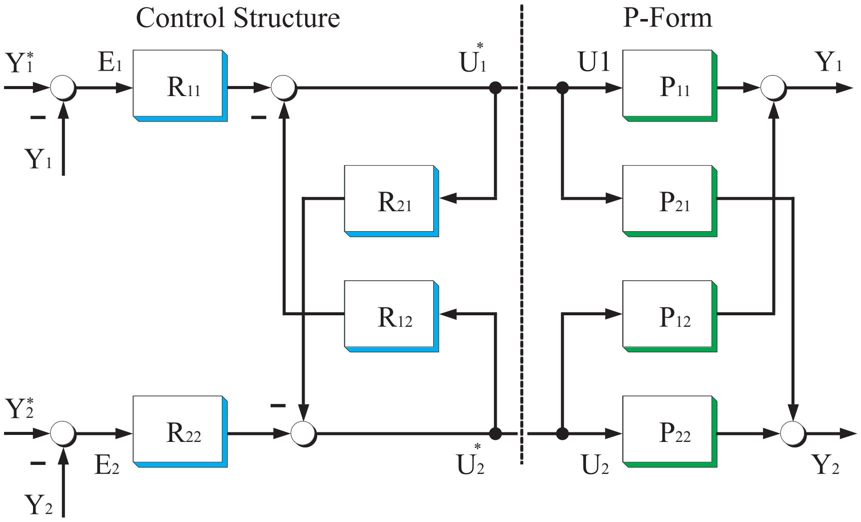


Figure 3.8: Decoupling MIMO control structure

Necessary steps to do the decoupling procedure can be summarized in the following steps[6]:

1. The *V-structured machine model* has to be translated first into *P-form*, Figure 3.7.

As stated above, the general matrix formulation of MIMO systems is exactly the same for the *P-canonic form*. It follows that the *V-formulated machine structure* in Figure 3.7-a has to be translated into *P-form*, and the associated translation matrices shall be determined.

The matrix representation of the *P-canonic system structure* is:

$$\begin{bmatrix} Y_1(s) \\ Y_2(s) \end{bmatrix} = \begin{bmatrix} P_{11}(s) & P_{12}(s) \\ P_{21}(s) & P_{22}(s) \end{bmatrix} \cdot \begin{bmatrix} U_1(s) \\ U_2(s) \end{bmatrix} \tag{3.11}$$

From the *V-canonic system structure* in Figure 3.7-b results:

$$\begin{aligned} Y_1 &= V_{11}\left[U_1 + V_{12}Y_2\right] \\ Y_2 &= V_{22}\left[U_2 + V_{21}Y_1\right] \end{aligned} \tag{3.12}$$

In which, the following assignments to the machine equations are valid:

$$\begin{bmatrix} Y_1(s) \\ Y_2(s) \end{bmatrix} = \begin{bmatrix} i_{sd} \\ i_{sq} \end{bmatrix}, \quad \begin{bmatrix} U_1(s) \\ U_2(s) \end{bmatrix} = \begin{bmatrix} u_{sd} \\ u_{sq} \end{bmatrix}$$

$$V_{11}(s) = V_{22}(s) = \frac{1}{r_\sigma\left(\tau_\sigma^{'} s + 1\right)} \tag{3.13}$$

$$V_{12}(s) = -V_{21}(s) = l_\sigma \omega_s$$

[6] For lack of space, the brackets with Laplace operator are might omitted in some places of the equations.

With some mathematically manipulations on the matrix formulations of both structure, equation (3.11) and equation (3.12), results:

$$\begin{bmatrix} Y_1 \\ Y_2 \end{bmatrix} = \frac{1}{1 - V_{11}V_{12}V_{21}V_{22}} \begin{bmatrix} V_{11} & V_{11}V_{12}V_{22} \\ V_{11}V_{21}V_{22} & V_{22} \end{bmatrix} \cdot \begin{bmatrix} U_1 \\ U_2 \end{bmatrix}$$

$$= \begin{bmatrix} P_{11} & P_{12} \\ P_{21} & P_{22} \end{bmatrix} \cdot \begin{bmatrix} U_1 \\ U_2 \end{bmatrix} \tag{3.14}$$

2. Do the decoupling procedure for the *P-structured system model*, Figure 3.8.

Decoupling procedure means design the main and the decoupling controllers for the *P-formed system model* in Figure 3.8. The main controllers R_{11}, R_{22} stand for the system elements P_{11}, P_{22} and control the system outputs Y_1, Y_2, respectively. Similarly, the decoupling controllers R_{12}, R_{21} stand to eliminate the coupling effect of the P_{12}, P_{21} on the system outputs Y_1, Y_2, respectively. Depending on the *V-formed decoupling network structure* in Figure 3.8, the controllers can be easily determined using the matrix representation:

$$\begin{bmatrix} U_1^*(s) \\ U_2^*(s) \end{bmatrix} = \begin{bmatrix} E_1(s)R_{11}(s) - U_2^*(s)R_{12}(s) \\ E_2(s)R_{22}(s) - U_1^*(s)R_{21}(s) \end{bmatrix}$$

$$= \frac{1}{1 - R_{12}(s)R_{21}(s)} \begin{bmatrix} R_{11}(s) & -R_{12}(s)R_{22}(s) \\ -R_{11}(s)R_{21}(s) & R_{22}(s) \end{bmatrix} \cdot \begin{bmatrix} E_1(s) \\ E_2(s) \end{bmatrix} \tag{3.15}$$

where E_1, E_2 are the controllers' inputs, U_1^*, U_2^* are the main controllers' outputs.

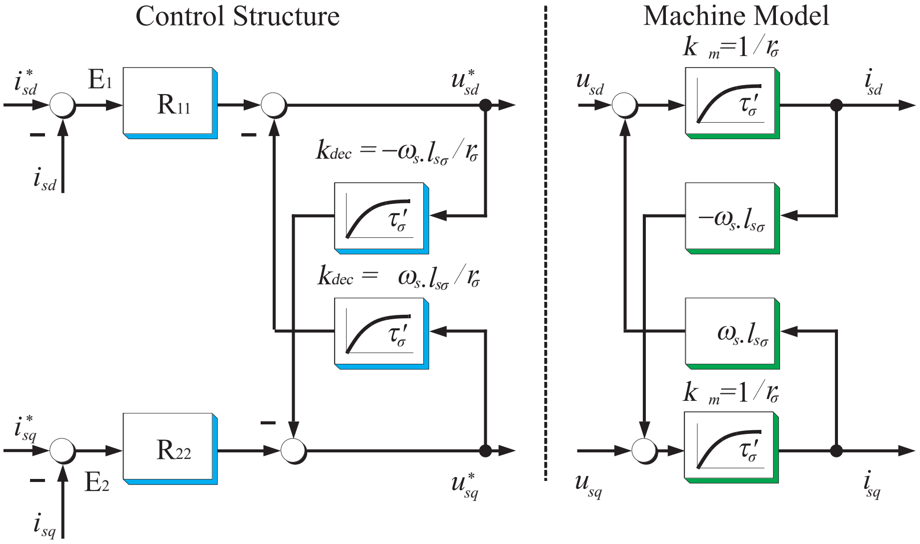


Figure 3.9: Decoupled AC machine model

Now, by combining both *P-formed machine model* (in equation (3.14)) with the resulting *V-formed control structure* (in equation (3.15)) results in equation (3.16).

$$\begin{bmatrix} Y_1 \\ Y_2 \end{bmatrix} = \begin{bmatrix} P_{11} & P_{12} \\ P_{21} & P_{22} \end{bmatrix} \frac{1}{1-R_{12}R_{21}} \begin{bmatrix} R_{11} & -R_{12}R_{22} \\ -R_{11}R_{21} & R_{22} \end{bmatrix} \cdot \begin{bmatrix} E_1 \\ E_2 \end{bmatrix}$$

$$\Downarrow \qquad (3.16)$$

$$\begin{bmatrix} Y_1 \\ Y_2 \end{bmatrix} = \frac{1}{1-R_{12}R_{21}} \begin{bmatrix} R_{11}\left[P_{11}-P_{12}R_{21}\right] & \left[P_{12}-P_{11}R_{12}\right]R_{22} \\ R_{11}\left[P_{21}-P_{22}R_{21}\right] & \left[P_{22}-P_{21}R_{21}\right]R_{22} \end{bmatrix} \cdot \begin{bmatrix} E_1 \\ E_2 \end{bmatrix}$$

Following the goal of the decoupling procedure, the effect of the input signals on the output ones has to be direct and diagonal represented. This can be achieved by set the off-diagonal elements in the combination matrix of system and controllers in the equation (3.16) to zeros.

$$\left[P_{12}-P_{11}R_{12}\right] \overset{!}{=} 0 \Rightarrow R_{12} = \frac{P_{12}}{P_{11}} = V_{12}V_{22} \Rightarrow \boxed{R_{12} = \frac{l_\sigma \omega_s}{r_\sigma\left(\tau_\sigma' s + 1\right)}}$$

$$(3.17)$$

$$\left[P_{21}-P_{22}R_{21}\right] \overset{!}{=} 0 \Rightarrow R_{21} = \frac{P_{21}}{P_{22}} = V_{21}V_{11} \Rightarrow \boxed{R_{21} = \frac{-l_\sigma \omega_s}{r_\sigma\left(\tau_\sigma' s + 1\right)}}$$

The above introduced equation (3.17) delivers directly the necessary decoupling controllers R_{12}, R_{21} in the decoupling network for an AC machine.

$$\begin{bmatrix} Y_1 \\ Y_2 \end{bmatrix} = \frac{1}{1-\frac{P_{12}}{P_{11}}\frac{P_{21}}{P_{22}}} \begin{bmatrix} P_{11}R_{11}\left[1-\frac{P_{12}}{P_{11}}\frac{P_{21}}{P_{22}}\right] & \left[1-\frac{P_{11}}{P_{12}}\frac{P_{12}}{P_{11}}\right]P_{12}R_{22} \\ P_{21}R_{11}\left[1-\frac{P_{22}}{P_{21}}\frac{P_{21}}{P_{22}}\right] & \left[1-\frac{P_{21}}{P_{11}}\frac{P_{12}}{P_{22}}\right]P_{22}R_{22} \end{bmatrix} \cdot \begin{bmatrix} E_1 \\ E_2 \end{bmatrix}$$

$$(3.18)$$

$$\frac{Y_1}{E_1} = \boxed{R_{11}P_{11}}$$

$$\frac{Y_2}{E_2} = \boxed{R_{22}P_{22}}$$

Design the main controllers R_{11}, R_{22} is left for the next step.

3. Design the main controllers for the *P-structured system model*, Figure 3.9.

Note that using the *P-canonic system structure* and *V-canonic controller structure* simplifies the decoupling procedure too much and results in designing four controllers, independently of each other's. Design the main controllers (R_{11} , R_{22}), as they are unaffected by the decoupling ones (R_{12} , R_{21}), can be done using conventional scalar PI controllers or MPC controllers; having the possibility to control both DC currents components (i_{sd} , i_{sq}) independently of each other's, Figure 3.10.

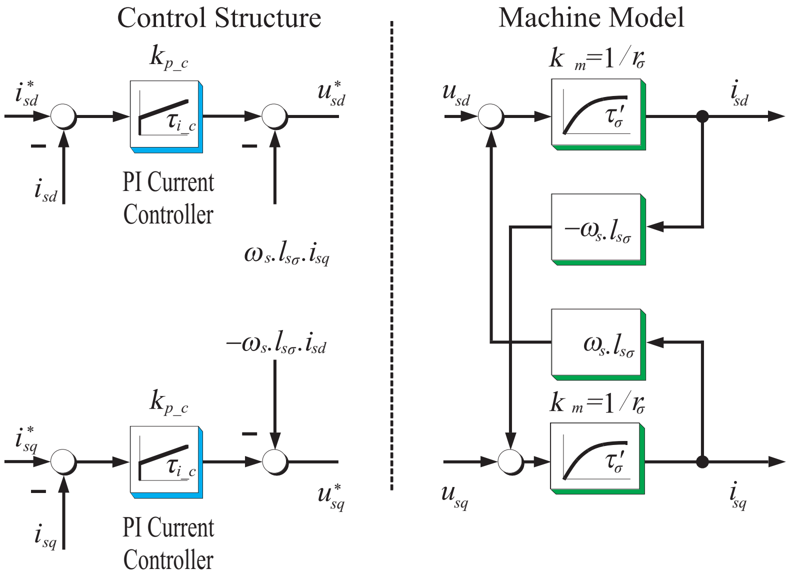


Figure 3.10: Scalar PI control structure with feed-forward compensation

The decoupling network acts at the output of the main controllers and eliminates the effect of the currents' cross-coupling in term of *feed-forward compensation (ffc)*. The *ffc* procedure is a model-dependent procedure and its validity comes with correct estimation of the machine quantities and parameters, as it will be discussed later in section 6.4.2. The design procedure of the PI controller (R$_{11}$, R$_{22}$) for both current components (i_{sd}, i_{sq}) is identical and delivers the same controller parameters. This procedure requires first determines the transfer functions in frequency domain of all models, which are involved in this closed-loop structure, Figure 3.11.

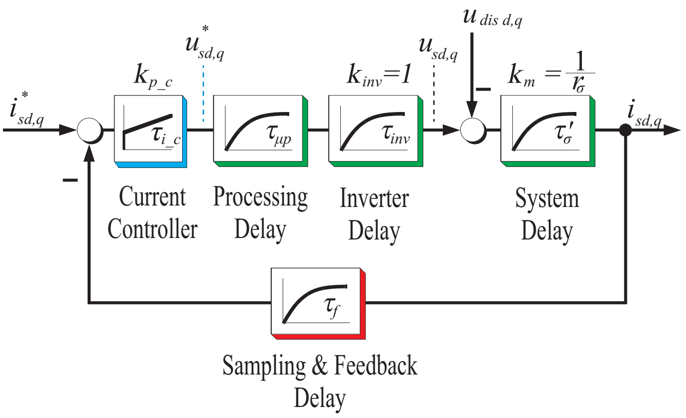


Figure 3.11: Current control loop, general form

All system elements, which cause amplification and a time delay in the current signal path, are represented by first-order systems with the associated coefficients. The time constants $\tau_f, \tau_{\mu p}, \tau_{inv}$ stand for the signal-feedback, micro-processor calculation-time, and inverter switching delays, respectively. These sub-systems are summed up together, as shown in Figure 3.12, in one equivalent first-order system with time constant $\tau_\Sigma = \tau_d$, which stands for the total delay. The time constant $\tau_\sigma^{'}$ stands for the machine time constant.

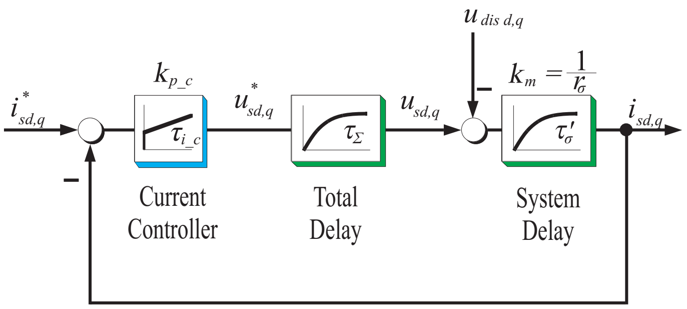


Figure 3.12: Current control loop

A scalar PI controller is assigned to each current component with following formulation:

$$F_{ctrl}(\boldsymbol{s}) = \frac{\boldsymbol{u}_s^*(\boldsymbol{s})}{\Delta \boldsymbol{i}_s(\boldsymbol{s})} = k_{pc}\frac{\tau_{ic}\boldsymbol{s}+1}{\tau_{ic}\boldsymbol{s}} \tag{3.19}$$

The resulting complete open-loop transfer function for current control is:

$$F_o(\boldsymbol{s}) = \overbrace{k_{pc}\frac{\tau_{ic}\boldsymbol{s}+1}{\tau_{ic}\boldsymbol{s}}}^{PI\ controller} \cdot \overbrace{\frac{1}{\tau_d\boldsymbol{s}+1}}^{System\ Delay} \cdot \overbrace{\frac{1}{r_\sigma}\frac{1}{\tau'_\sigma\boldsymbol{s}+1}}^{Machine\ Dynamic} \tag{3.20}$$

Two main factors are to be considered by adjusting PI controller parameters, namely the used sampling frequency and existing of an integrator in the system structure. In this work all the experimental results were collected at 10 kHz sampling frequency. The high sampling rate resulted in system-delay time constant $\tau_d = 1.5\tau_{sampling}$ much smaller than the stator time constant τ'_σ by a factor of about 50. For this reason and as the complete system model (machine model + system delay model) does not include an integrator in its structure, adjusting the PI current controller can be done according to modulus optimum, root locus, or symmetrical optimum [59]. Following the first two methods to adjust the PI current controller, Figure 3.12, the controller time constant will be chosen to eliminate the machine pole at $-1/\tau'_\sigma$. While the controller gain, will be adjusted to achieve a critical damping $k_{pc} = r_\sigma \tau'_\sigma / 2\tau_d$ Existing of an integral in the mechanical system structure restricts these two methods to the current controller; where eliminating the time constant of the current equivalent-system structure, in Figure 3.15, of speed control loop leads to an oscillatory behavior. Symmetrical optimum method can be introduced here to deal with both current and speed controllers. Therefore, the following results were collected for the induction machine, whose parameters are introduced in the Appendix-A, by using symmetrical optimum method [59] as flows: $\tau_{ic} = 4\tau_d\tau'_\sigma / (\tau'_\sigma + 3\tau_d)$, $k_{pc} = r_\sigma \tau'_\sigma / 2\tau_d$.

The Figure 3.13, shows a step response of the torque producing current ($i_{sq} = 0 \rightarrow 0.4$) by keeping a full magnetization of the induction machine ($i_{sd} = 0.35$).

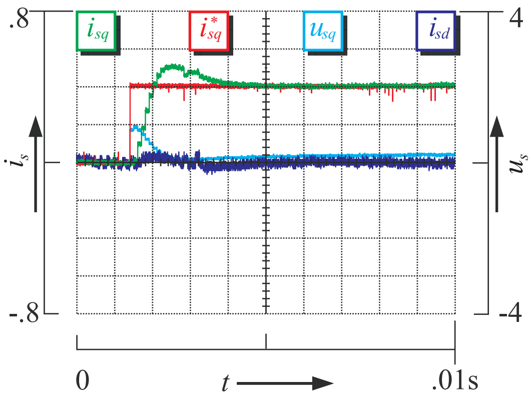


Figure 3.13: IM current response using a SPI$_{dq}$ controller

The current PI controller with symmetrical optimum adjustment exhibits a fast dynamic with unavoidable overshooting. In conventional PI controller, the constraints and limitations of the voltage source inverter on the controller outputs can be only partially considered. Reducing the effect of the controller saturation and associated instability can be done by imposing a voltage-limiter on the controller output in one of anti-wind up schemes [93], [94], e.g. Figure 3.14. In contrast to the conventional PI controller, the MPC controller considers the system constraints in its structure and optimizes its output such that a fast dynamic without overshooting can be achieved, as it will be discussed in section 6.7.

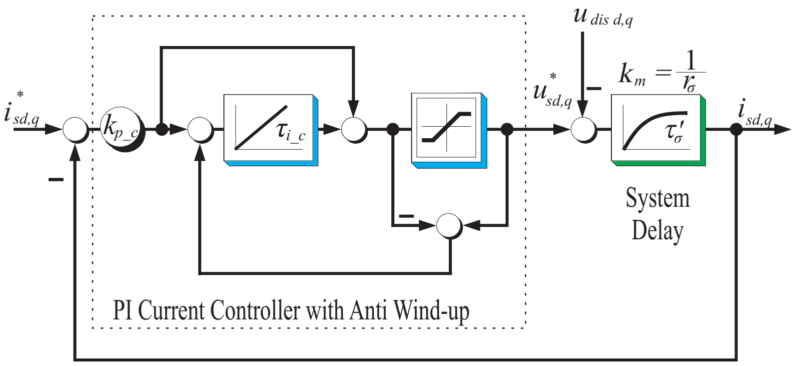


Figure 3.14: Anti wind-up PI control structure

For angular velocity control, the following speed closed-loop control structure in Figure 3.15 has to be considered. The design of the outer speed control loop requires the approximation of the inner current control loop behavior with an equivalent first order system, with a time constant τ_{eq} . The measured speed signal is to be further filtered with a sufficient time constant before it is applied to the speed controller, Figure 3.15. This will help to get a smooth speed signal and avoid the sluggish control behavior results of using a low resolution incremental encoder. The effect of using the speed filter on the feed-forward compensation procedure is discussed later in section 6.4.2 in more details. Similar to current controller and for having an integrator in the mechanical system model, the symmetrical optimum method has to be employed here.

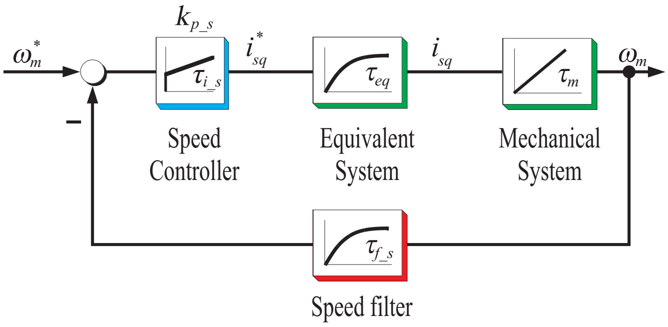
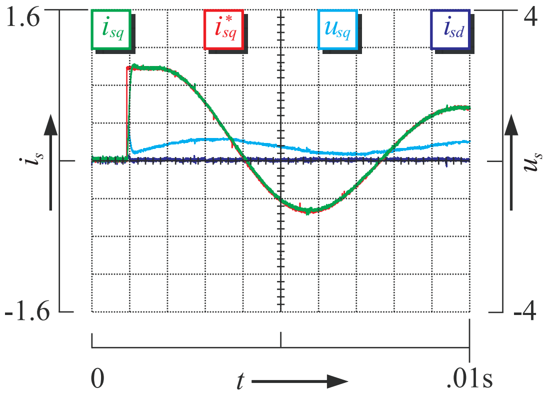


Figure 3.15: Speed control loop

To observe the angular velocity variation on the PI current controller, the Figure 3.16 shows the machine currents by applying a speed step ($\omega_m = 0 \rightarrow 50\%$ of the machine nominal speed). It is clear that the PI controller even without compensation of the induced-voltages could achieve a zero steady-state tracking error.

Figure 3.17 shows the machine currents and voltages at 60% of the nominal speed in the stationary reference frame. These results were collected for an induction machine fed with a three-level NPC inverter.



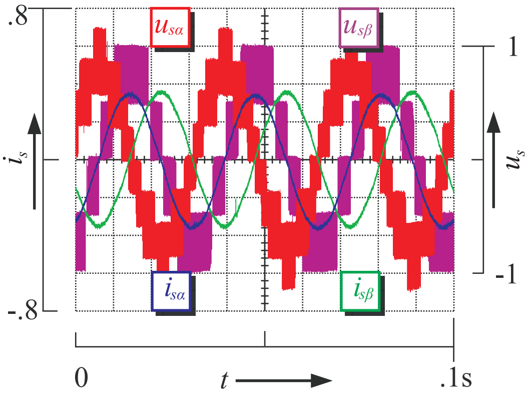


Figure 3.16: IM current response using a SPI$_{dq}$ controller by applying a 50% nominal speed step

Figure 3.17: IM currents and voltages in stationary coordinates controlled using a SPI$_{dq}$ controller at 60% nominal speed

3.3.2.1 Limitations of Using Scalar PI Current Control with Feed-Forward Compensation

The total leakage inductance l_σ of an induction machine is the main machine parameter used in the compensation procedure of the motion induce-voltage $\left(\pm l_\sigma\ \omega_s\, i_{sd,q}\right)$. Therefore it has to be correctly estimated from the feedback signals, and any error in the estimation could make the situation worse. Hence, the parameter sensitivity is a common drawback of this method. Some other factors affect the compensation procedure and make it in some situations not the right choice, can be stated in following: 1- The wrongly estimated total leakage inductance l_σ. 2- The delayed compensation terms due to the inverter switching. 3- Delayed speed signal ω_m. These factors together reduce the efficiency of this method, and make designing the controller depending on the complex machine model more important.

3.3.3 Complex PI Controller

Further investigation could be made on the scalar and complex PI controller [52] using Root Locus strategy, as an efficient tool to analyze and observe the system behavior with respect to one parameter of interest. The root locus is a method to sketch the location of the closed-loop roots of a system addressed by the characteristic equation $1+K_p\,.\,F_o(\mathrm{s})=0$ against some parameter of interest, usually the feedback gain K_p or other particular variable. $F_o(\mathrm{s})$ is the open-loop transfer function of a system. The resulting sketch shows how the closed-loop poles of the system move to the zeros or infinity.

To observe the cross-coupling effect by increasing angular velocity of the rotor shaft ω_m on the dynamic performance of an AC machine, it is wise to consider ω_m as the parameter of interest. When designing the current control system, $\boldsymbol{u}_s$ is considered as the input, and $\boldsymbol{i}_s$ is considered as the output variable of the machine. Thus the complex transfer function $F_m(\mathrm{s})$ of the induction machine is derived from the equation (2.10) in general *k-reference frame* as follows:

$$\begin{aligned} F_m^{(K)}(\boldsymbol{s}) &= \frac{\boldsymbol{i}_s^{(K)}(\boldsymbol{s})}{\boldsymbol{u}_s^{(K)}(\boldsymbol{s})} \\ &= \frac{1}{r_\sigma}\frac{(\tau_r\,.\,\boldsymbol{s}+1+j\left(\omega_k-\omega_m\right)\tau_r)}{\left((\boldsymbol{s}\,.\,\tau_\sigma^{'}+1+j\,\omega_k\,.\,\tau_\sigma^{'})(\tau_r\,.\,\boldsymbol{s}+1+j\left(\omega_k-\omega_m\right)\tau_r)-k_1\left(1-j\,\tau_r\,.\,\omega_m\right)\right)} \end{aligned} \tag{3.21}$$

Working in the synchronous coordinate system $\omega_k=\omega_s$, two cases can be distinguished for an induction machine: Unloaded induction machine [52]; the assumption $\omega_m=\omega_s$ is valid. Loaded induction machine; here the slip frequency $\omega_r\neq 0$ is to be considered.

In this work, no-load assumption, $\omega_r=\omega_s-\omega_m=0$, and the equation (3.22) were just used to draw the root locus graphs for simplicity, whereas the equation (3.23) without any assumption was used to design the complex controller.

An unloaded induction machine model, equation (3.22):

$$\begin{aligned} F_m(\boldsymbol{s}) &= \frac{\boldsymbol{i}_s^{(F)}(\boldsymbol{s})}{\boldsymbol{u}_s^{(F)}(\boldsymbol{s})} = \frac{1}{r_\sigma}\frac{(\tau_r\,\boldsymbol{s}+1)}{\left(\tau_r\tau_\sigma^{'}\boldsymbol{s}^2+\left(\tau_\sigma^{'}+\tau_r+j\,\omega_m\tau_\sigma^{'}\tau_r\right)\boldsymbol{s}+(1-k_1+j\,\omega_m\,(\tau_\sigma^{'}+k_1\tau_r))\right)} \\ &: \omega_k=\omega_s=\omega_m \end{aligned} \tag{3.22}$$

A loaded induction machine model, equation (3.23):

$$\begin{aligned} F_m(\boldsymbol{s}) &= \frac{\boldsymbol{i}_s^{(F)}(\boldsymbol{s})}{\boldsymbol{u}_s^{(F)}(\boldsymbol{s})} = \frac{1}{r_\sigma}\frac{\left(\tau_r\boldsymbol{s}+1+j\omega_r\tau_r\right)}{\left(\tau_\sigma^{'}\boldsymbol{s}+1+j\omega_s\tau_\sigma^{'}\right)\left(\tau_r\boldsymbol{s}+1+j\omega_r\tau_r\right)-k_1\left(1-j\omega_m\tau_r\right)} \\ &: \omega_r\neq 0 \end{aligned} \tag{3.23}$$

Refereeing to the complex representation of the delay-system in equation (3.6), controlling the complete complex-represented system model with a scalar PI controller will take the following formulation, equation (3.24).

$$F_o(\boldsymbol{s}) = \frac{\boldsymbol{i}_s^{(F)}(\boldsymbol{s})}{\boldsymbol{u}_s^{(F)}(\boldsymbol{s})} = F_{ctrl}(\boldsymbol{s}).F_s(\boldsymbol{s}).F_m(\boldsymbol{s}) =$$

$$\overbrace{k_{pc}\frac{\tau_{ic}\boldsymbol{s}+1}{\tau_{ic}\boldsymbol{s}}}^{PI\ controller}.\overbrace{\frac{1}{\left(1+j\,\omega_s\tau_d+\tau_d\boldsymbol{s}\right)}}^{System\ Delay}.\overbrace{\frac{1}{r_\sigma}\frac{\left(\tau_r\boldsymbol{s}+1+j\,\omega_r\tau_r\right)}{\left(\tau_\sigma^{'}\boldsymbol{s}+1+j\,\omega_s\tau_\sigma^{'}\right)\left(\tau_r\boldsymbol{s}+1+j\,\omega_r\tau_r\right)-k_1\left(1-j\,\omega_m\tau_r\right)}}^{Induction\ Machine\ Model} \qquad (3.24)$$

$$:\omega_r \neq 0$$

Considering the transfer function between the current and the voltage, equation (3.22), the root loci of the complete open loop system when the mechanical speed ω_m changes from 0 to 1 is presented in Figure 3.18.

Using a simple scalar PI current controller for controlling such a complex system, equation (3.24), all signals that are multiplied with imaginary gain coefficients are to be neglected. With this approximation, the respective imaginary and real components of the space vector currents results independent of each other and two single PI controllers could be built for each current component. This kind of controller eliminates only the real poles and not the imaginary ones. The scalar PI controller with feed forward compensation exhibits tendency to instability of the controlled machine, especially at higher than the nominal speed and low switching frequency [52], Figure 3.21. Working at high frequencies, e.g. of several 5 kHz, results in a very small system time delay τ_d. This permits the controller, with forward compensation of the imaginary terms, to enhance the machine dynamic and leads finally to a good performance, Figure 3.19.

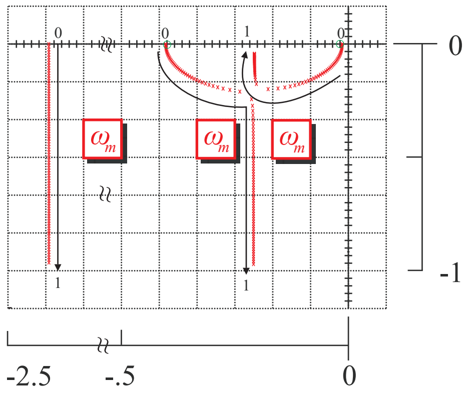


Figure 3.18: Root loci of an IM plus a delay-system in synchronous coordinates; par: ($\omega_m = 0 \rightarrow 1$)

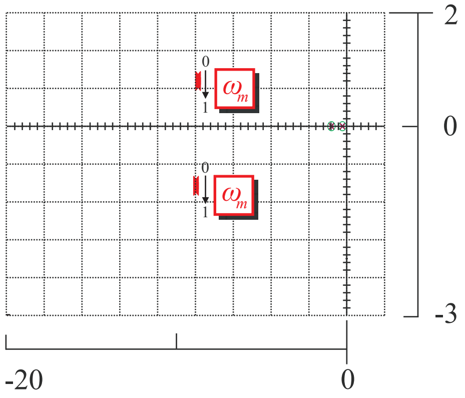


Figure 3.19: Root loci of the closed-loop system, using a SPI$_{dq}$ controller, in synchronous coordinates at f_s = 5 kHz; par: ($\omega_m = 0 \rightarrow 1$ [pu])

On the other side, working at low switching frequency like 500 Hz or even lower, the influence of the imaginary terms in the system model will appear very clear. Figure 3.20 shows how the cross-coupling between the currents dominates the response and leads the closed-loop system to be unstable, at speed higher than the nominal one, Figure 3.21. In this case, the dynamics of the PI controller could not be improved even by using the feed forward compensation method.

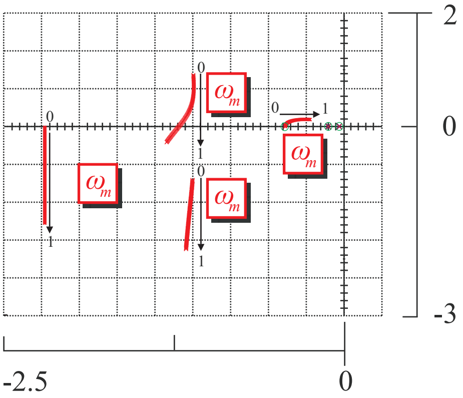


Figure 3.20: Root loci of the closed-loop system, using a SPI$_{dq}$ controller, in synchronous coordinates at f_s = 500 Hz; par: ($\omega_m = 0 \rightarrow 1$ [pu])

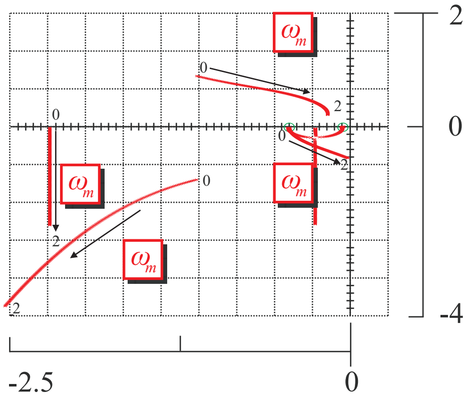


Figure 3.21: Root loci of the closed-loop system, using a SPI$_{dq}$ controller plus feed-forward compensation, in synchronous coordinates at f_s = 500 Hz; par: ($\omega_m = 0 \rightarrow 2$ [pu])

In contrast to simple PI controller, the complex one considers the complex system model without any simplification and tries to compensate all the system poles, real and imaginary ones. Therefore the complex controller can be seen as kind of compensator with the following complex formulation:

$$F_o(\boldsymbol{s}) = \frac{\boldsymbol{i}_s^{(F)}(\boldsymbol{s})}{\boldsymbol{u}_s^{(F)}(\boldsymbol{s})} = F_{ctrl}(\boldsymbol{s}).F_s(\boldsymbol{s}).F_m(\boldsymbol{s}) =$$

$$F_{ctrl}(\boldsymbol{s}) = k_o \frac{(\tau_{i3}\boldsymbol{s}+1+j\,\omega_s\tau_{i3})}{\tau_{i3}\boldsymbol{s}+1} . \frac{(\tau_{i1}\boldsymbol{s}+1+j\,\omega_s\tau_{i1})(\tau_{i2}\boldsymbol{s}+1+j\,\omega_r\tau_{i2})-k_1(1-j\,\omega_m\tau_{i2})}{\tau_{i1}\boldsymbol{s}(\tau_{i2}\boldsymbol{s}+1+j\,\omega_r\tau_{i2})} \qquad (3.25)$$

$$:\tau_{i1}=\tau'_\sigma, \tau_{i2}=\tau_r, \tau_{i3}=\tau_d$$

Following the time constants substitution in the equation (3.25) $\tau_{i1}=\tau'_\sigma, \tau_{i2}=\tau_r, \tau_{i3}=\tau_d$, results in an open loop transfer function with real poles, where all the complex dynamics associated with the angular velocity variation are completely canceled. The term $(\tau_{i3}\,\boldsymbol{s}+1+j\,\omega_s\tau_{i3}) : \tau_{i3}=\tau_d$ compensates the complex delay-system and generates a chain of zeros as function of ω_s which replaces the poles in the delay-system. Furthermore, the term $((\tau_{i1}\boldsymbol{s}+1+j\,\omega_s\,\tau_{i1})/(\tau_{i1}\,\boldsymbol{s}))$: $\tau_{i1}=\tau'_\sigma$ aims at compensating the eigenbehaviour of the stator, and contributes with a chain of zeros at $(-1/\tau'_\sigma - j\,\omega_s)$as function of ω_s and also adds a pole in the origin, which serves as integrator. The remaining part of the controller compensates the rotor, and with the term $(k_1\,(1-j\,\omega_m\,\tau_r)/(\tau'_\sigma\,\boldsymbol{s}\,(\tau_r\,\boldsymbol{s}+1+j\,\omega_r\,\tau_r)))$ it compensates the dynamic interaction between the stator and the rotor of the machine. It adds also zeros at $(-1/\tau_r - j\,\omega_m)$as functions of ω_m and poles at $(-1/\tau_r - j\omega_r)$ as a function of ω_r. In its interaction with last term, it displaces the zeros of the last term and its own zeros to be exactly located on the typical single-complex eigenvalues of the induction motor and adds poles at $(-1/\tau_r - j\,\omega_r)$ to eliminate the zeros in the machine equation, Figure 3.22.

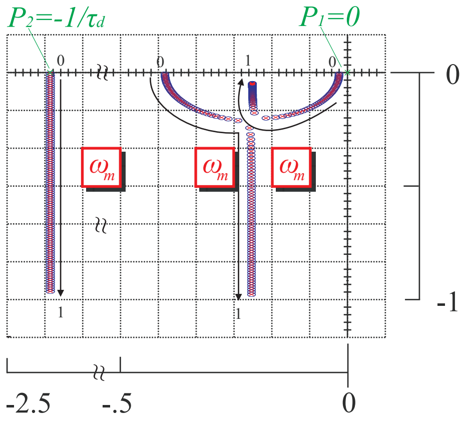


Figure 3.22: Roots and poles of the plant and the complex current controller, in synchronous coordinates at $f_s = 500$ Hz; par: ($\omega_m = 0 \rightarrow 1$ [pu])

Finally, the resulting open-loop transfer function will be:

$$F_o\left(s\right) = k_o \frac{1}{\tau'_\sigma \, s\,(\tau_d s + 1)} \tag{3.26}$$

It has only two poles P_1 , P_2 on the left side of real axis. The closed-loop system is designed for critical damping $D = \sqrt{2}/2$ which determine the open loop gain $k_o = r_\sigma \tau'_\sigma / 2\tau_d$. Note that the effect of both the rotor induced voltage and the motion induced voltage are totally compensated. Hence, the complex current controller could be considered as Complex-represented PI controller plus a compensator.

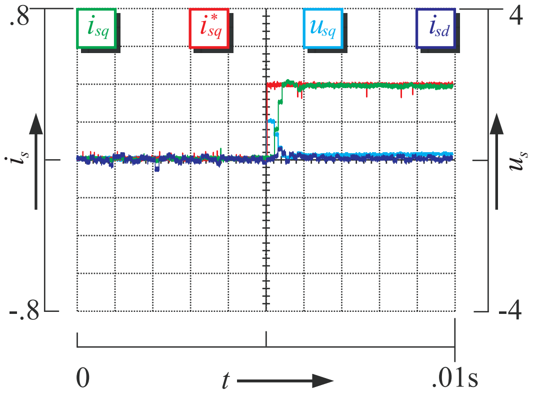


Figure 3.23: IM current response using a CPI$_{dq}$ controller

By compensating the voltage drops (represented by motion- and rotor- induced voltages) inside the controller structure, the controller could react actively on the variation of these drops and results in no overshooting response, Figure 3.23. Figure 3.23 presents fast dynamic and small settling time comparing with the scalar PI controller. On that account, this work distinguishes between two kinds of compensation: Active and passive compensations. In contrast to the active compensation, the passive one is done outside the controller and then the controller reacts passively on it; as the case for scalar PI controller with feed-forward compensation. The signal flow graph in Figure 3.24 shows the complex PI controller structure with internal compensation of the induced voltages.

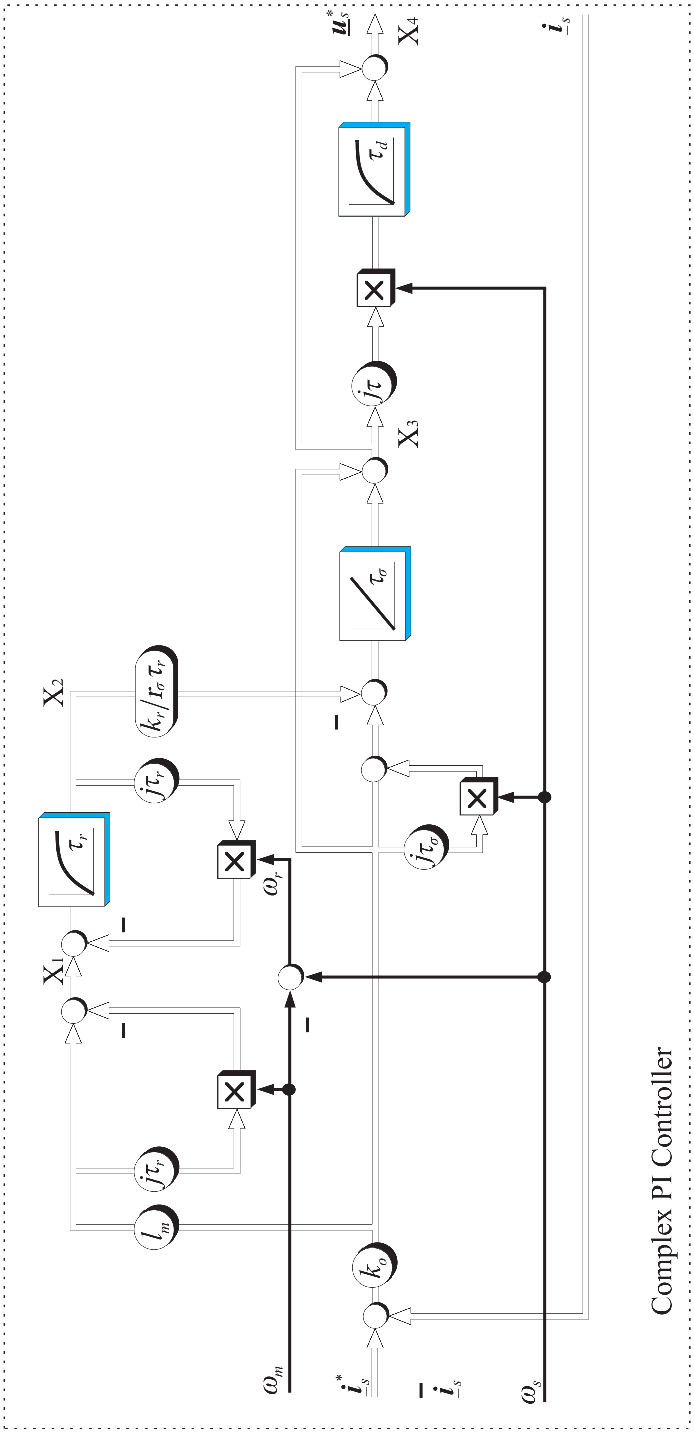


Figure 3.24: Signal flow graph of the complex current controller

The symbolic auxiliary variables ($X_1 \ldots X_4$) help to conclude the final formulation of the complex PI controller for both current components.

Figure 3.25 shows the induction machine currents and voltages in stationary coordinates at ($\omega_m = 25$ % of the nominal speed) by using the complex PI controller.

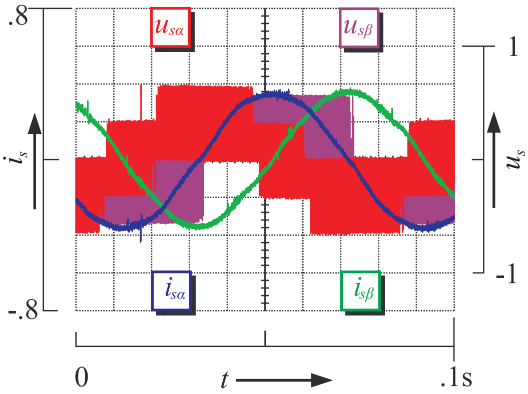


Figure 3.25: IM currents and voltages in stationary coordinates controlled using a CPI$_{\mathrm{dq}}$ controller at $\omega_m = 25$ % nominal speed

3.4 Conclusion

The AC machine model is a complex MIMO system with cross-coupled signals. Therefore, working with this structure as it is without simplifications is much preferred. The complex PI controller considers the complex machine structure and tries to eliminate all the system poles and adjust the resulting transfer function to get a critical damped response. The active compensation and the interaction between the controller parts result in a high efficient control structure. Whereas designing a simple scalar PI controller depends on some simplifications of the complex system model and passive reaction of the controller to the compensated terms at its output. Therefore it could fail to assure the control stability in some operating points and could not achieve a complete decoupling of the controlled signals.

In the previous chapters the author tries to summarize the introduced works by Prof. Holtz regarding the complex representation of the AC machine and designing the complex PI controller. The above introduced AC machine models were the fundamental ones and the space harmonic and torque ripple are not considered. Further discussions and details can be found on the following publications [50], [51], [52], [53], and [117]. The physical interpretation and the main reason to use a complex control structure were addressed in [51]. Introduction and implementation of the complex PI controller in this work intends to show the way of complete and active compensation of the motion- and rotor-induced voltages. Such compensation procedure inside the control structure is used with the MPC controller without any simplifications, section 6.5.4.

4 THE DIGITAL SYSTEM

4.1 Introduction

The aim of this chapter is to present briefly the design procedure of a small size, low cost and high efficiency rapid prototype system, which was used as hard- and software platform for the introduced work in this thesis. For further information about the proposed digital system, the author refers to the publications [1], [24], [25], [26], [81], [38], which explain the design and evaluation procedure with some examples and case studies in more details. Furthermore, the introduced work in [1] contains all the Schematic/ Layout files, C and VHDL source code. The introduced digital system depends basically on the personal computer (PC) processor without the need for any external processor (e.g. digital signal processor (DSP)) to get fast execution of control algorithms, and to support floating point operations. The PC-based rapid-prototyping system is dedicated for laboratory purposes and consists of an extension kit with all required interface boards to do the simulation and implementation of electrical drive algorithms. Almost all designed boards are based on programmable-logic devices (PLD) to reduce cost and size. To operate the hardware of the proposed system a free available open source real-time operating system (RTOS) kernel under Linux is used. The introduced work of this thesis could be considered as case study from electric drive area with numerous experimental results that demonstrate the good functionality of the conceived system.

4.2 State-Of-The-Art of Digital Control Systems

Nowadays digital control techniques are mostly carried out with microcontrollers or digital signal processors (DSPs), because almost all famous operating systems are desktop-oriented operating systems and do not support hard real-time applications. Thus, DSP controllers are considered by many engineers as an appropriate solution [28]. These components have an arithmetic logic unit particularly dedicated to the real-time computation. They also integrate peripheral units like analog-to-digital converters (ADCs) and timers, which are adapted to the needs of electrical drives. Nevertheless, the developed rapid-prototyping system has to contain all required software and packages to enhance the real-time applications. In such systems the PC is only used for developing, compiling the control software, collecting and showing the results, whereas the real-time tasks run completely on the DSP hardware. The required software to compile and transfer the output binary file to the DSP system is always associated with cost and license. High cost and license constraints associated with the High efficiency commercial prototype system like dSPACE [29] and Speedgoat, are a big problem in universities and technical institutes, because this cost appears for each researcher set-up [30], [31]. Because of this reason, teaching staffs at the universities always try to build their own set-up [27] and prototypes which fulfill their requirements [32], [33]. To overcome cost problem associated with such controller boards, the idea comes here to design a rapid-prototyping system with small size, low cost and high efficiency, which depends on programmable logic devices (PLD) and free available real-time application interfaces under Linux. This work introduces a fully digital, hard real-time system depending basically on the PC-processor, Figure 4.1. It is designed to meet the requirements of modern rapid control prototyping and to perform the necessary simulation and implementation of electrical drives algorithms. The PC-based proposed system does not need any external floating-point processor (as in other prototypes) to support the complex arithmetic operations like in [34], [35], and it provides a wide selection of interfaces, like analogue to digital converters (ADC) board, digital to analogue converter (DAC) board, pulse width modulation (PWM) board, encoder interface board etc. The proposed system is also supported by high efficiency RTOS, a user-friendly programming environment and high level programming language. The PC-based rapid-prototyping system was developed in the Electrical Machines and Drives laboratory at Wuppertal University and many researches and publications were applied on it ranging from different and complicated control algorithms such as field-oriented control [35], sensorless control [36], grid applications and control [37], and predictive control [38], [81]. This chapter is organized as follows. In section 4.3, the construction and properties of the proposed system are firstly presented. Then, section 4.4 introduces the chosen operating system and the developed programming environment.

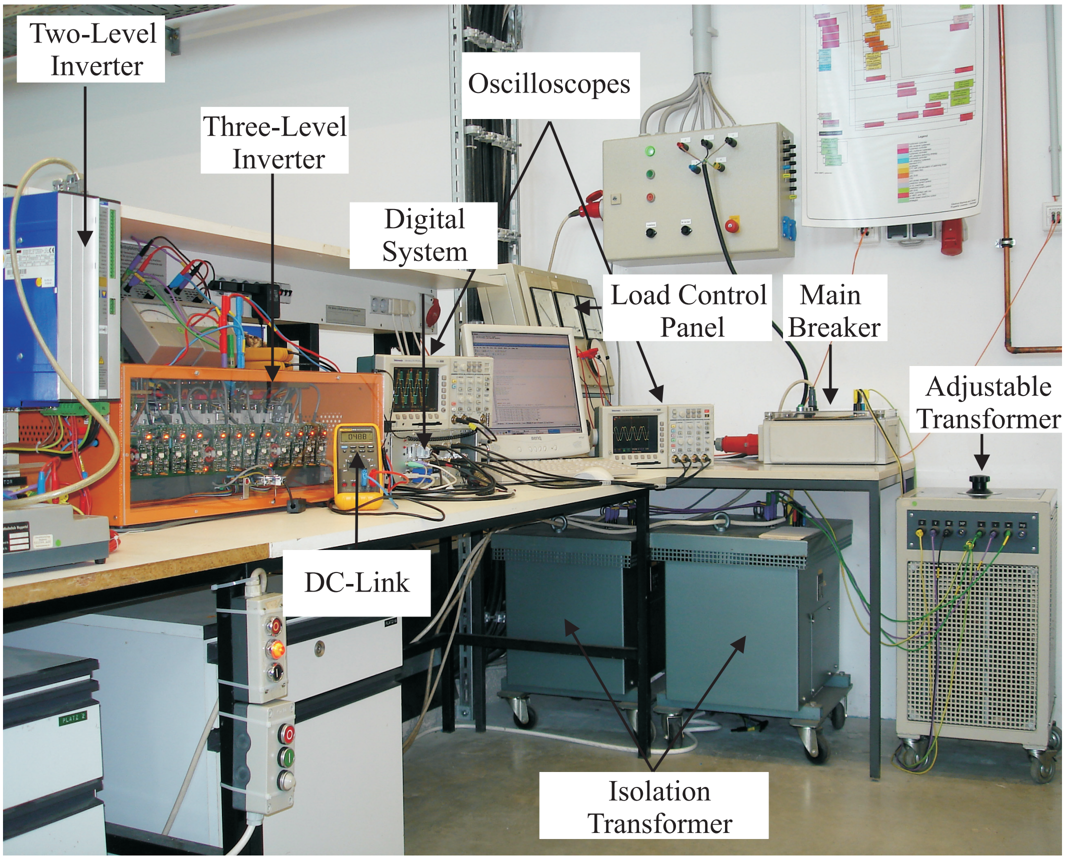


Figure 4.1: The hardware set-up

4.3 Construction and Properties of the Proposed System

The designed system, Figure 4.1, consists of all required interface boards for electric drives, which assure safe connections and adaptation between set-up equipments (e.g.: differential voltage probe, current sensor, position sensor, inverter and oscilloscope) and the PC. These interface boards are connected by an extension kit, which guarantees the parallel communication between the PC-mother board and other interface boards by the industrial standard adapter (ISA). A new PC with high efficiency and high processor speed is not necessary. It is sufficient to use a Pentium2- or a Pentium3 –PC with ISA slot. Since PCs of this type are not always available in the market or university storage, the PC interface board of the proposed system is designed to be connected with an industrial PC (PC/104-equiped with ISA slot) as well, which is always available in market at acceptable price. The extension kit consists of 13 slots, which are equipped to be connected with 13 various interface boards. In the following paragraphs, main components of this system are described.

4.3.1 Extension Board

It is the back bone of the proposed system and designed to provide safe parallel communication and data exchange between PC interface board and other interface boards. In combination with a suitable power supply attached, the extension board provides the required voltages (5V, +/- 15V) for other boards. Extension board gives the system more flexibility and extendibility by containing 12 slots with different addresses. Each slot is equipped with 2-physical addresses to communicate with PC and one of three available interrupt signals Irq3, 5, 10. One extra slot is added without addresses but with an interrupt signal Irq3. This slot can be used to insert an interrupt generating board. As shown in Figure 4.3 the first slot is designed to be connected with PC-interface board.

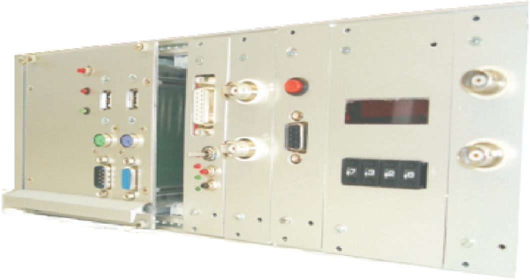


Figure 4.2: The developed digital system

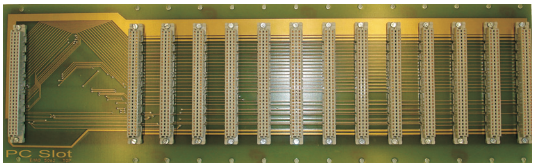


Figure 4.3: The extension board

4.3.2 PC and PC/104 Interface Board

The designed PC interface board is so adapted to be connected to ISA slot of a normal desktop PC or of an industrial PC (PC/104 module or PC/104Plus module) if the first one is not easily available. The interface board supplies the extension kit with 16-bit data bus, 24 enable signals to address 12 various interface boards and all other required interrupt and control (IOREAD, IOWRITE) signals. PLD-IC is used in this board as address decoder to reduce the cost, size and design complexity.

In the following Table 4.I a list of the industrial PC specifications, which was used in implementation phase of the introduced work.

Table 4.I: PC/104 Specifications

Specifications	PC/104 Module
Processor Module	Em 104-i613
Processor Specifications	Celeron 650 MHz + 256 KB Cache Memory
Extension Bus	1 PC/104 for ISA interface
RAM Memory	Max. 512 MB
SSD Interface	Max. 1 GB CFC
Display Controller	Savage4 2D/3D, 42 MB Shared Memory
Network Interface Controller	Realtek 8100BL chip
Ports	2 COM/2 USB
Power requirements	+5 V @ 2.1 A
Board dimensions	90 mm x 96 mm

4.3.3 Pulse Width Modulation (PWM) Board

The most important expansion board in the proposed system is PWM board. It contains a field-programmable gate array (FPGA) that can be reprogrammed on board by VHDL. The designed board is responsible to produce switching signals for driving two or multilevel voltage source inverters (VSI). VSIs are nowadays commonly used in variable-speed AC motor drives to produce variable, three-phase AC output voltages from a constant DC voltage. For more information about design, structure and function of the used inverter the author refers to technical report [27] of design 3-level VSI. In this board, the desirable PWM modulation strategy can be implemented to control the average output voltages of a VSI over a sufficiently small period, called switching period, by producing switching pulses of variable duty-cycles (s_{a1} , s_{a2} , s_{a3} , s_{a4} , s_{b1} , …, and s_{c4}) to control the associated power switches (see Figure 4.4). The reference voltages, which are computed on the PC, are refreshed exactly at the triangular PWM carrier vertex instants. Therefore, the mean values of the applied VSI voltages are equal to the reference ones [39], [41]. The circuit diagram in Figure 4.4 presents a conventional three-level IGBT voltage source inverter connected to a three-phase AC machine. A DC voltage u_{dc} is supplied at the inverter input by a DC power supply, where the inverter outputs are connected to the machine terminals, accordingly. In between, the inverter contains three half-bridge units of stacked power-switches, such that the switches in each leg are alternatingly turned on and off at given time instances according to the switching pulses. According to used modulation strategy on the PWM board, the machine terminals can be connected to either the positive DC-link voltage potential or to the negative potential, and the AC machine will be finally supplied with three-phase voltages with variable amplitude and frequency. PWM board has further functionality, namely a dead-/ interlock-time insertion between switching signals. Inserted dead-time protects the power switches of the inverter against short circuit, when two switches on one leg are to be conducting at the same time. To set the wanted dead-time, a group of dip-switches are available on the board surface. Besides, protection mechanism is implemented by VHDL program, and can easily be activated and modified. The third assigned task to the PWM board is an interrupt signal generation. As a result of this interrupt, the PC-processor scans the associated interrupt service routine. The interrupt frequency can be given in the main C-program.

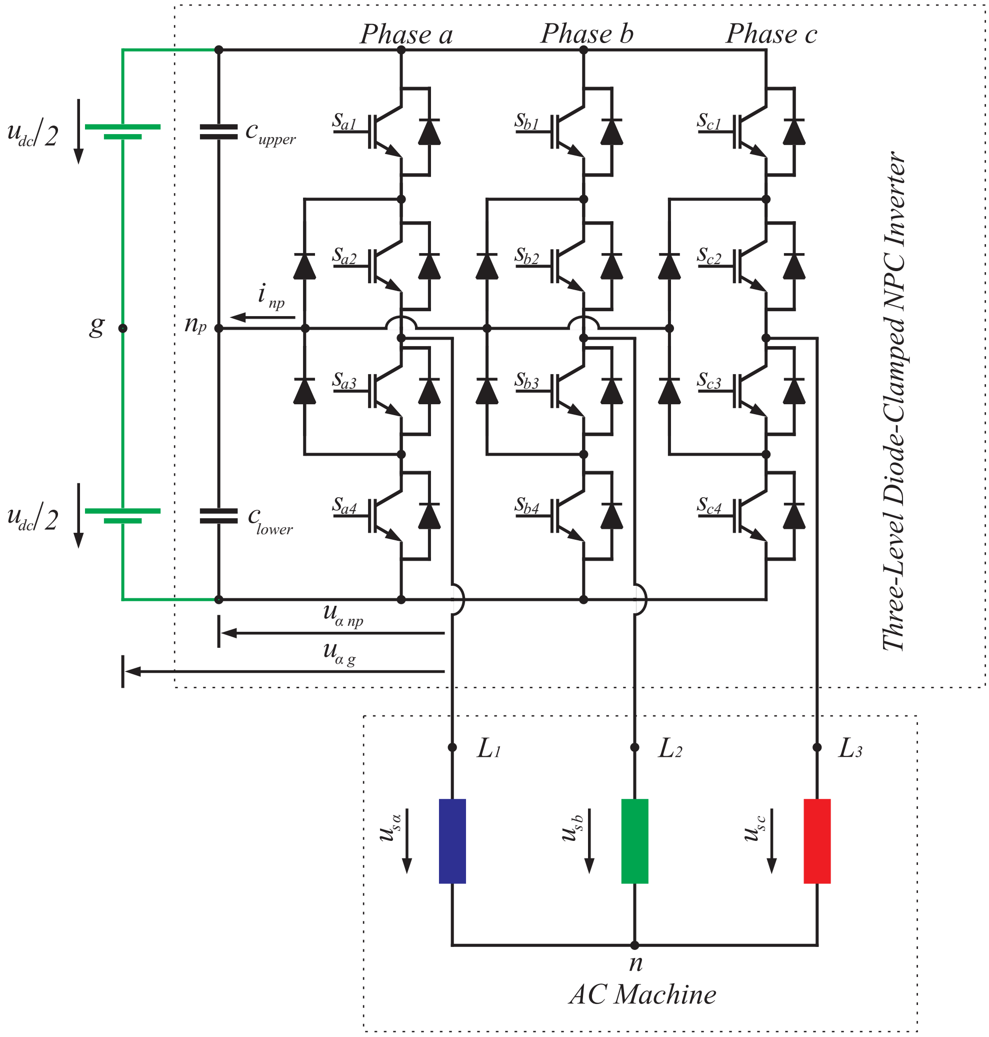


Figure 4.4: The controlled system (AC machine plus three-level NPC inverter)

4.3.4 Analogue to Digital Converter

In every controlled system some of analogue values (like currents and voltages) have to be sensed or measured from the control system. To satisfy this need, high resolution ADC interface board is introduced. Each ADC interface board contains two channels with 1.625µsec conversation time and 4.992 mV resolution. A low pass filter can be activated in the input channels, and the amplitude of the analogue input signals can reach +/-10 V. The two channels have different addresses and different operating modes:

- First, they can be enabled from hardware; synchronized with an internal interrupt signal or with other signals.
- Second, they can be enabled by software commands to activate one or two channels at the same time.

4.3.5 Digital to Analogue Converter

For control and test purposes during simulation and implementation stages, there are always some signals or values in C-program to be showed on the oscilloscope. To response to this purpose DAC interface board is designed. Each DAC interface board contains two channels with 3µsec settling time and with 4.885mV resolution. Each channel has a different address and the conversation process is initialized by software commands. For signal adjusting purpose, DAC interface board is equipped with trim resistors.

4.3.6 Hex-Value Entering and Displaying Board

Hex-board is equipped with four hex-values seven segments display units and four hex-values dip switches. The hexadecimal entered- and displayed- values are software adapted and enable the programmer to change the operating condition of the controlled machine. To reduce the cost and size of this board, PLD-ICs are used for 7-segments decoding operation.

4.3.7 Incremental Encoder Interface Board

Rotary incremental encoders are common in position and motion sensing and they are preferred when low cost is important, or when only relative position is needed. Rotary encoders typically produce two TTL-outputs and their inverse. These output signals are shifted 90º from each other and their sequence decides the rotation direction of motor shaft. Incremental rotary encoders often have a third reference signal, which comes one time in each shaft rotation. All three output signals and their inverse are considered in the developed board to remove the possible noise in work area.

The designed interface board deals with incremental encoders with resolution of 16384 pulses/ cycle maximum (14-bit). To protect the encoder and the rest of the board from possible short circuit, an optocoupler isolation circuit is integrated. An interface board with another type of encoders like absolute encoders or resolvers can be developed in the same way and inserted in the proposed system if it is required. Speed and position sensors for servo drives are discussed in [42].

4.4 Software

4.4.1 Operating System

Since a desktop or an industrial PC is employed in the proposed digital system, it is necessary to choose a proper operating system, which allows achieving: high efficiency, real-time application support, low interrupt latency, user-friendly interface and low cost. Among all operating systems (Table 4.II) available from Microsoft [43], Linux, and Macintosh; Linux is selected as an appropriate OS for the proposed system, due to the following reasons [44]:

- It is subject to generalized public license (GPL) (free available, open source).
- Many variations are available by its different distributions.
- It depends on UNIX platform, which provides high security and performance.
- Its kernel can be adapted to support real-time applications with very low interrupt latency.
- It supports loadable modules:
    - Kernel modules can be inserted in kernel space by commands in user space.
    - No need to restart the PC or reload the kernel after modifying the program.

These favorable characteristics of Linux as desktop oriented (general purposes) operating system are not sufficient for RT-applications. Linux as known is based on UNIX, which is designed as time sharing system to optimize its average performance (weighted by priorities), and Linux preserves this nature. This means that the RT-processes will suffer the most if the load increases. Currently Linux does not support an interface for informing the OS that a task is real-time (RT) or informing it of any timing constraints on a task. The basic Linux scheduler is time slice based on priorities, which make Linux not recommended for RT-applications for the following reasons:

- The time slice resolution may be higher than required for the RT-tasks and so it may be impossible to meet the timing demands.
- Raising the priority of real-time tasks is not sufficient because lower priority tasks will cause missing the deadlines anyway.
- Scheduler's fairness is not a desirable quality in real-time systems.
- Linux does not provide a reliable mechanism to wake a task up at a certain time. The sleep timer can only promise to wake a task up after a certain period.
- External interrupts are disabled at several sections in the kernel to protect OS data from corruption. This adds unpredictability to the amount of time that takes to respond to real-time I/O.

Nevertheless, it is possible to obtain hard RT-Linux from general purposes operating system with saving most of advantages of Linux environments [45]. The main idea is to add a hard RT-scheduler, which stands for scheduling of RT-tasks. These tasks will be granted higher priorities than non RT tasks as described in the following (Figure 4.6). The RT-scheduler splits the RT-programs into two parts: small light parts with hard RT constraints and larger parts doing most of the processing. The light RT-parts are scheduled in the RT-scheduler, while the bigger parts run as normal processes under Linux. The two parts can communicate through a RT-FIFO or a non-blocking queue. RT-tasks are implemented as kernel modules, where their init-functions send all information about their release times, periods and deadlines to the RT-scheduler. RT-Linux scheduler bases on EDF (Earliest Deadline First) algorithm and runs the basic operating system process as the lowest priority process and will execute it as long as no RT-process is available. While Linux is running, it uses its own scheduler to schedule running tasks. The Linux kernel runs as a process under the RT-scheduler and as a result can be interrupted by it at any time. The main performance characteristic of a RTOS is how fast it responses to internal or external events. These events can be internal software interrupts, external hardware interrupts etc. Standard criteria can be introduced to compare the available RTOSs; interrupt latency, which can be defined as the time between an interface device requesting service (by raising an interrupt flag) and the time that the CPU needs to start the interrupt process. All variants of RT-Linux have introduced modifications at the kernel level to reduce the interrupt latency between periodic interrupts to the microsecond range, allowing for faster response to external events and higher timing resolution. Table 4.II shows the average interrupt latency for standard windows and all RT-Kernels under Linux. Depending on the performance criteria, the proper kernel can be selected. These kernels include a standard RT-kernel as described in IEEE 1003.1d [46], commercial RTOS-kernels, Micro-kernels with subject to patents [47] as well as free available Nano-kernel ADEOS. The adaptive domain environment for operating systems (ADEOS), released as an open source project in June 2002, provides an interface between the standard kernel and the hardware [48].

The free available ADEOS kernel is similar to the micro kernel approach mentioned above, but it is implemented in a way to prevent conflicts with the claims of its patent. The background history and design for ADEOS is described in a white paper published in 2001 [49]. ADEOS kernel was chosen and used in the proposed system.

Table 4.II: Comparison between different RTOSs

Operating System OS	**Application**	**Average Latency**
Standard Windows	Non real-time	100 µsec to 100 msec
Standard Linux	Soft real-time	> 10 msec
IEEE 1003.1d Linux	Hard real-time	10 to 100 µsec
○ KURT: Kansas		
○ Time Sys		
Commercial RTOS Kernels	Hard real-time	1 to 10 µsec
○ Concurrent Computer Corporation		
○ FSM Labs Inc.		
○ Monta Vista Software Inc.		
○ Quality Real Time System QRTS		
○ RED Sonic Inc.		
Micro-Kernel Linux	Hard real-time	1 to 10 µsec
○ RT Linux: New Mexico		
○ RTAI: Milano		
Nano-Kernel	Hard real-time	1 to 10 µsec
○ ADEOS		

After patching the kernel as explained in [45] on the selected Linux distribution (like Suse) some measurements are done on the hardware with the mathematical machine model [50] before the real model is taken in operation. These measurements prove the efficiency of this kernel with the proposed hardware as a stable hard RT system. Interrupt latency test is done on the proposed system by using the PWM board as interrupt generator. The operating system receives this interrupt signal and starts to run the related interrupt service routine (ISR), which starts its instructions with output command of logic ‘1’ on digital to analogue board, output (‘1’, d2a_adr). Through the digital to analogue board and available interrupt test point on PWM board, the interrupt latency can be measured on the oscilloscope.

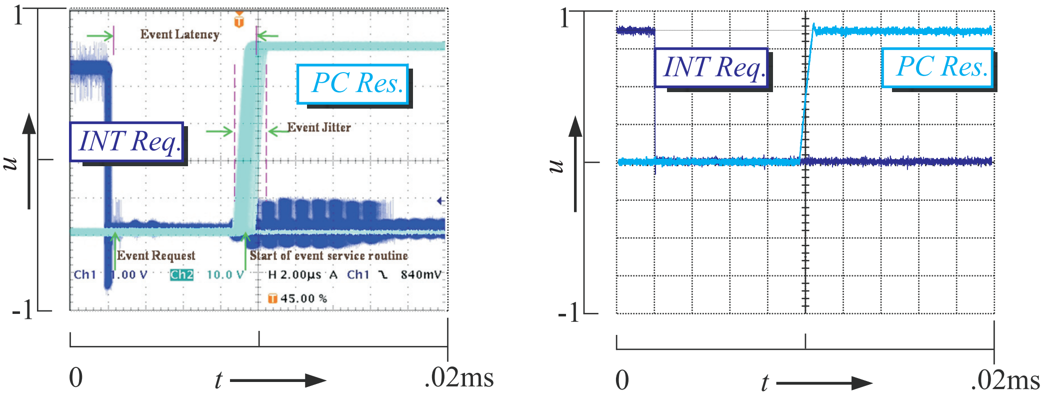


Figure 4.5: Interrupt latency time of the digital system

a) Jitter effect, b) Average latency time

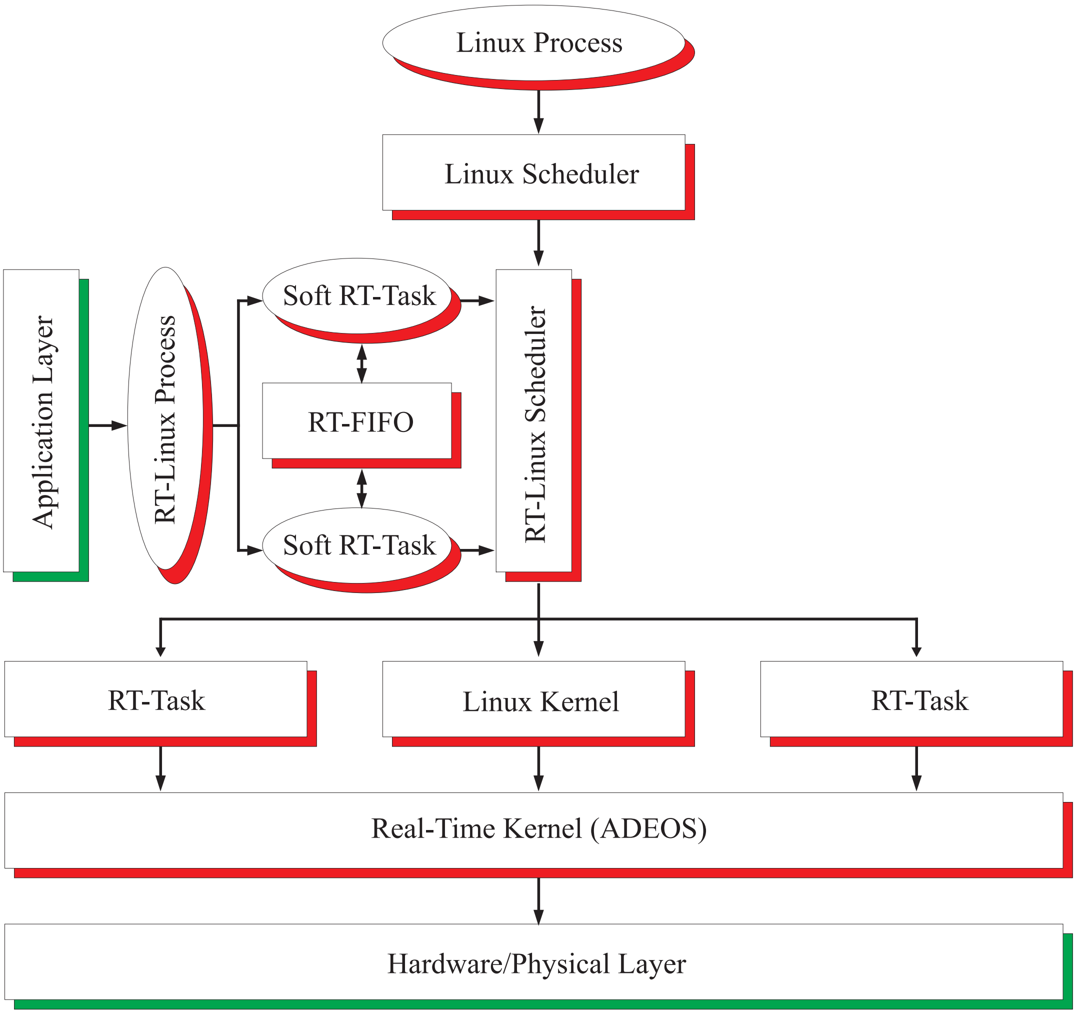


Figure 4.6: RTOS structure

This measurement was done under the following conditions:

- All system boards are inserted in their slots.
- 2PI current controllers are applied in the ISR (this means, there are some calculations in ISR).
- There are no programs running in the background.
- 650MHz Pentium3 Processor of an industrial PC with 512 MB RAM, Table 4.I.

Figure 4.5 shows the measured interrupt-latency time, which equal to 8µsec including DA conversion time (settling time of DAC is equal to 3µsec). These measurements were repeated many times to show the average latency in different load conditions of the operating system.

4.4.2 Programming Environment

This section introduces an easy and user-friendly programming environment that is supported by a high level programming language (standard C). All required routines to communicate with designed interface boards are included in this framework to keep the user or researcher far away from hardware layer complications. Thus, in order to make the design of control algorithms more manageable and less intuitive, the designer has to strictly follow a set of steps and rules, which consist of an efficient design methodology. The main characteristics of this kind of methodologies are the reusability of the already made designs, the consideration of the control performances, and finally, the reduction of the development time. The structure environment (Figure 4.7) is organized in such way, that it lets the researcher easily insert an algorithm code in the Interrupt Service Routine (ISR) with few modifications on other files. Start and stop files are responsible to initialize the system and save/ restore the status of processor registers before entering/ exiting the ISR. ISR should contain the main program or the implemented algorithm. Some definitions of assigned interface board addresses, required RT-libraries, and I/O functions are given in "Libraries .h". All required instructions to compile and execute source files are implemented in "Exec. Files". There is no compiler or programming environment to be bought like visual studio from Microsoft, but the system depends on free available GCC compiler and RT-libraries from Linux under GPL agreement. And the portability characteristic of the developed programming environment gives the user the right to choose the proper Linux distribution freely from any constraints, but of course with some necessary modifications in the code.

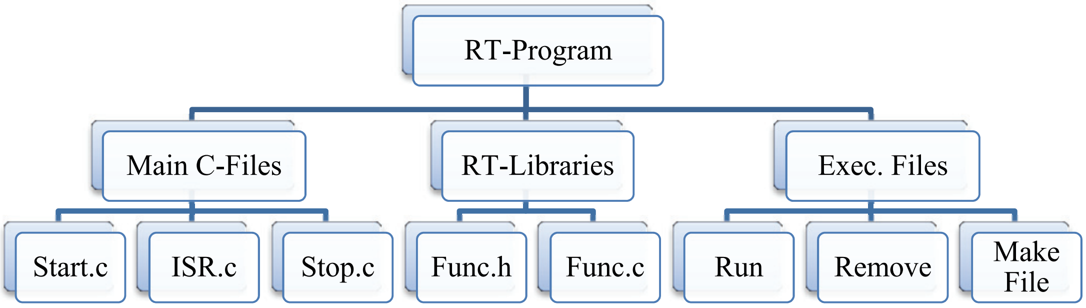


Figure 4.7: RT-program structure

4.5 Conclusion

In this chapter a low cost high efficiency rapid-prototyping system designed to support research applications in electrical drives has been presented. It is based on personal/ industrial computer and has benefited from its capabilities and high processor frequency to support sophisticated algorithm implementations and floating point operations. Concerning the operating system, a free available, RT-Linux kernel with high efficiency and very low interrupt latency time has been used. The proposed system has been supported with an easy environment for a high level programming language (C language). Finally, case studies from the electrical drive area are listed in the next chapters with numerous experimental results to give the proof that such solution can reach the level of their commercial counterparts, keeping at the same time a high degree of flexibility, programmability, and extendibility.

5 MODEL-BASED PREDICTIVE CONTROL

5.1 Overview

Model-based Predictive Control (MPC) is an optimal-control based method for constrained feedback control, where the optimization problem is solved at each time step starting from the current state and over a finite horizon called the prediction horizon N_p . With respect to all constraints on states, outputs, and inputs only the first element (**u**) of the resulting optimal control sequence (**U**) is applied to the plant while the rest is discarded. At the following time step, this computation is repeated with a new state and over a shifted horizon. Therefore, an MPC problem can be addressed as a constrained optimization problem (with a linear, quadratic, or infinite norm) with receding horizon control (RHC) policy. Unfortunately, MPC still suffers some limitations which make it not widely used in fast industrial applications; RHC policy with short horizon possibly also steer the state to a part of the state space where no solution satisfies the constraints. Therefore, the feasibility and stability of RHC are not guaranteed in general. The lack of feasibility and stability of RHC is addressed in literature intensively [92] and solved by adding the terminal set constraints for a finite time optimization problem, or by solving the problem for an infinite time $N_p = Inf.$

The main limitation of MPC to be a standard in electric drive applications is the extensive on-line computation of the optimization problem, which is increased dramatically with number of inputs and the increased prediction horizon N_p . In [65], the authors proposed multi-parametric programming (MPP) approach as a solution to move the time-consuming part of the model predictive control problem off-line. Solving the MPC problem using the multi-parametric approach, the state variables are treated as free parameters and the control variables as optimization variables, and then the optimization problem is solved for all possible values of these free parameters. Thus, the implicit solution of the MPC's optimization problem is converted into an explicit solution, and the state space of the free parameters is divided into polytopes[7] and inside each polytope, one single linear control law being valid. Depending on the Karush-Kuhn-Tucker (KKT) optimality conditions for stability of constrained linear systems [66], the resulting control action will be a piecewise affine (PWA) function of the state variables. The explicit solution has the same characteristics concerning control performance and stability of the full on-line solution of the MPC problem.

[7] Polytope: is a convex, bounded, and closed n-dimensional set, which can be represented as an intersection of a finite number of half-spaces [77], [67], [67].

For the implementation, there are various methods introduced in literature to accelerate the online execution depending on the system type (with continuous or discrete input and state variables) and on the optimization norm (with a linear, quadratic, or infinite norm) [68], [69], [70], [72], [73], [74], [75]. The resulting explicit solution will be as a look-up-table containing all feasible regions in state space and the associated control laws. To control the plant, there is no optimization problem to be solved online at each sampling time. Rather, the controller simply has to find out in real time, in which polytope the current state-vector is located, and to evaluate the appropriate affine control law. The optimal behavior of the controlled system is determined in advance either by pre-calculation or by selection of the optimal trajectory. Even if the chosen behavior does not fulfill all demands of the application, the user can be sure that this is the best possible behavior.

The explicit solution of model-based predictive control (Exp-MPC) for complex systems such as electric drive applications results in some cases in a high number of regions. Increase the number of regions means more memory space is needed to store the explicit solution of MPC (represented by the generated regions and associated optimal control laws) as well as the necessary time to evaluate the proper affine control laws will be longer; which limits the implementation of the Exp-MPC in fast sampling rate applications [82].

As a solution to reduce the online complexity, the polynomial approximation introduced in chapter 7 and summarized in [111] intends to reformulate a combination of the optimal affine control laws in several regions with only one polynomial of higher degree and covers those regions. Doing this approximation will reduce definitely the accuracy of the optimal controller, but on the other hand it will make the implementation of the Exp-MPC in more complex cases more feasible; if the stability and feasibility of the problem are still guaranteed.

In this chapter the philosophy and formulation of the MPC problem with all necessary stability constraints is introduced first, and followed by introduction to the multi-parametric programming and multi-parametric toolbox in MPC control. Finally, Section 5.4.1 refers to some of possible implementation strategies and highlights the most efficient one.

5.2 MPC Philosophy

Different approaches are introduced in the literature to deal with the model-based predictive control MPC and the way of formulating the optimization problem. The main difference between different strategies to formulate the MPC problem could be in the way of formulating the controlled model using the impulse response, step response, transfer function and state space representation. They also differ in the way of describing the disturbance model or even in the cost function with different norms [83]. The introduced formulation of MPC in this work depends on the state space representation of the MPC problem for its applicability to deal with SISO and MIMO systems, stability criteria is well known and established, and it is widely used in the academic research. However, the methodology behind all these different approaches is the same and it can be summarized as shown in Figure 5.1. Here and depending on the available knowledge of the state- and input-values until time step t= k, the system model is used to predict the future behavior of the physical process at each sampling time and for pre-determined steps, called prediction horizon N_p . The predicted outputs are used in an optimization procedure to minimize a cost function, which can be formulated with different norms to achieve the required goal, like tracking a reference signal or minimization of some losses. As a result of the optimization problem, a set of optimal control values until a specific horizon, called control horizon N_u , are determined $\mathbf{U} = \left\{ \mathbf{u}_{\mathrm{k}}^{*}, ..., \mathbf{u}_{\mathrm{k+N_u-1}}^{*} \right\}$. Following the receding horizon policy, only the first control action of this set has to be applied to the plant whereas the rest is ignored $\mathbf{u}_k^*$. This optimization procedure is repeated at each sampling time with respect to all constraints and conditions on the system variables.

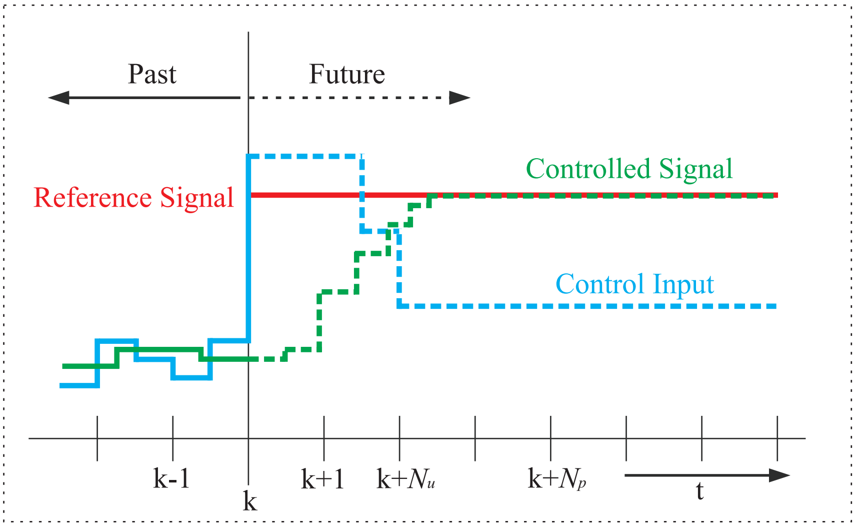


Figure 5.1: MPC strategy

5.3 MPC Formulation

Discretization:

The continuous-time state space representation of a linear time-invariant system model (5.1) can be transformed into the discrete form (5.3) by using the forward Euler discretization method (5.2), as follows:

$$\begin{aligned} &\frac{d}{dt}\mathbf{x}(t) = \mathbf{A}.\,\mathbf{x}(t) + \mathbf{B}.\,\mathbf{u}(t) \\ &\mathbf{y}(t) = \mathbf{C}.\,\mathbf{x}(t) \end{aligned} \tag{5.1}$$

where **A**, **B**, **C** are the system matrices, and **x**, **u**, **y** are the state, input, and output vectors, respectively.

The difference quotation of forward Euler discretization method:

$$\frac{d}{dt}\mathbf{x}(t) = \frac{\mathbf{x}(k+1) - \mathbf{x}(k)}{\Delta t} \tag{5.2}$$

where Δt is the time difference or the normalized sampling time in our case.

By substituting the (5.2) in (5.1) results the discrete-time formulation:

$$\begin{aligned} \mathbf{x}_{k+1} &= \mathbf{A}_d .\, \mathbf{x}_k + \mathbf{B}_d .\, \mathbf{u}_k \\ \mathbf{y}_k &= \mathbf{C}_d .\, \mathbf{x}_k \end{aligned} \tag{5.3}$$

where

$$\begin{aligned} \mathbf{A}_{\mathrm{d}} &= T_0\,\mathbf{A} + \mathbf{I} \\ \mathbf{B}_{\mathrm{d}} &= T_0\,\mathbf{B} \\ \mathbf{C}_{\mathrm{d}} &= \mathbf{C} \end{aligned} \tag{5.4}$$

with $T_0 = \Delta\mathrm{t} = \tau_{\mathrm{sampling}}$ being the normalized sampling time.

The introduced analysis in chapter 7 of [76] proved by experimental results that the forward Euler approximation can be used in machine drive field with an acceptable approximation error, for its simplicity and its low calculation effort.

Just for simplicity, the associated subscript ($\Box_{\mathrm{d}}$) with discrete formulated matrices is omitted in the future analysis and representation.

The following equations (5.5) describe the behavior of a discrete-time linear time-invariant system:

$$\begin{aligned} \mathbf{x}_{\mathrm{t+1}} &= \mathbf{A}.\,\mathbf{x}_{\mathrm{t}} + \mathbf{B}.\,\mathbf{u}_{\mathrm{t}} \\ \mathbf{y}_{\mathrm{t}} &= \mathbf{C}.\,\mathbf{x}_{\mathrm{t}} \end{aligned} \tag{5.5}$$

with $\mathbf{A} \in \mathbb{R}^{\mathrm{nxn}}$ and $\mathbf{B} \in \mathbb{R}^{\mathrm{nxm}}$ and the pair (**A**, **B**) is stabilizable, and the pair (**A, C**) is detectable.

While fulfilling the state, input, and output constraints, equation (5.6):

$$\begin{aligned} &\mathbf{x} \in \boldsymbol{X} \subset \mathbb{R}^{\mathrm{n}}, \mathbf{u} \in \boldsymbol{U} \subset \mathbb{R}^{\mathrm{m}}, \mathbf{y} \in \boldsymbol{Y} \subset \mathbb{R}^{\mathrm{p}} \\ &\boldsymbol{X} = \left\{\mathbf{x} | \mathbf{x}_{\mathrm{t}} \in \boldsymbol{X}, \mathrm{t} \geq 0\right\} \\ &\boldsymbol{U} = \{\mathbf{u} \mid \mathbf{u}_{\mathrm{t}} \in \boldsymbol{U}, \mathrm{t} \geq 0\} \\ &\boldsymbol{Y} = \{\mathbf{y} \mid \mathbf{y}_{\mathrm{t}} \in \boldsymbol{Y}, \mathrm{t} \geq 0\} \end{aligned} \tag{5.6}$$

Let $\mathbf{x}_{\mathrm{t}}$ denote the state-vector at time t and $\mathbf{x}_{\mathrm{t+k/t}}$ denote the predicted state-vector at time (t+k) for the given state-vector at time t. For brevity, let denote $\mathbf{x}_{\mathrm{k/0}}$ as $\mathbf{x}_{\mathrm{k}}$ and let $\mathbf{u}_{\mathrm{k}}$ be the computed input vector for the time k, given **x**(0). Where $\mathbf{x}_{\mathrm{t}} \in \mathbb{R}^{\mathrm{n}}$, $\mathbf{u}_{\mathrm{t}} \in \mathbb{R}^{\mathrm{m}}$, and $\mathbf{y}_{\mathrm{t}} \in \mathbb{R}^{\mathrm{p}}$ are the state, input, and output vectors, respectively. These constraints are to be provided at first as inequalities or as a closed, bounded and convex sets or polyhedral sets. In MPC formulas the system equation (5.5) is used also to predict the future behavior of the controlled system. Here we assume the full measurements of the state-vector $\mathbf{x}_{\mathrm{t}}$ are available at the current time t and no observer is required. The linear MPC problem for regulating the state-vector in (5.5) to the origin is expressed as following quadratic programming problem (QP):

$$\boldsymbol{J}\,(\mathbf{U}, \mathbf{x}_{\mathrm{t}}) = \min_{\mathbf{U}} \left(\mathbf{x}^{\mathrm{T}}{}_{\mathrm{t}+N_p|\mathrm{t}}.\,\mathbf{P}_{\mathrm{N}}.\,\mathbf{x}_{\mathrm{t}+N_p|\mathrm{t}} + \sum_{\mathrm{k}=0}^{N_p-1} \mathbf{x}^{\mathrm{T}}{}_{\mathrm{t+k|t}}.\,\mathbf{Q}.\,\mathbf{x}_{\mathrm{t+k|t}} + \mathbf{u}^{\mathrm{T}}{}_{\mathrm{t+k|t}}.\,\mathbf{R}.\,\mathbf{u}_{\mathrm{t+k|t}} \right) \tag{5.7}$$

$$\mathbf{Q} = \mathbf{Q}^{\mathrm{T}} \geq 0, \mathbf{R} = \mathbf{R}^{\mathrm{T}} > 0, \mathbf{P}_{\mathrm{N}} \geq 0 \tag{5.8}$$

And subject to the following constraints:

$$\begin{aligned}
&\mathbf{x}_{\min} \leq \mathbf{x}_{\mathrm{t+k|t}} \leq \mathbf{x}_{\max} && ,\mathrm{k}=1....N_c \\
&\mathbf{y}_{\min} \leq \mathbf{y}_{\mathrm{t+k|t}} \leq \mathbf{y}_{\max} && ,\mathrm{k}=1....N_c \\
&\mathbf{u}_{\min} \leq \mathbf{u}_{\mathrm{t+k}} \leq \mathbf{u}_{\max} && ,\mathrm{k}=0....N_c \\
&\mathbf{x}_{\mathrm{t|t}} = \mathbf{x}_{\mathrm{t}}
\end{aligned} \qquad (5.9)$$

$$\begin{aligned}
&\mathbf{x}_{\mathrm{t+k+1|t}} = \mathbf{A}.\,\mathbf{x}_{\mathrm{t+k|t}} + \mathbf{B}.\,\mathbf{u}_{\mathrm{t+k|t}} && ,\forall\ \mathrm{k} \geq 0 \\
&\mathbf{y}_{\mathrm{t+k|t}} = \mathbf{C}.\,\mathbf{x}_{\mathrm{t+k|t}} && ,\forall\ \mathrm{k} \geq 0 \\
&\mathbf{u}_{\mathrm{t+k}} = \mathbf{K}_{\mathrm{LQR}}.\,\mathbf{x}_{\mathrm{t+k|t}} && ,N_u \leq \mathrm{k} < N_p
\end{aligned}$$

$$\begin{aligned}
&\mathbf{K}_{\mathrm{LQR}} = -(\mathbf{R} + \mathbf{B}'.\,\mathbf{P}_{\mathrm{N}}.\,\mathbf{B})^{-1}.\,\mathbf{B}'.\,\mathbf{P}_{\mathrm{N}}.\,\mathbf{A} \\
&\mathbf{P} = (\mathbf{A} + \mathbf{B}.\,\mathbf{K}_{\mathrm{LQR}})'.\,\mathbf{P}_{\mathrm{N}}.(\mathbf{A} + \mathbf{B}.\,\mathbf{K}_{\mathrm{LQR}}) + \mathbf{K}'_{\mathrm{LQR}}.\,\mathbf{R}.\,\mathbf{K}_{\mathrm{LQR}} + \mathbf{Q}
\end{aligned} \qquad (5.10)$$

where $\mathbf{U} = \left\{\mathbf{u}_{\mathrm{t}},....,\mathbf{u}_{\mathrm{t}+N_{\mathrm{u}}-1}\right\} \in \mathbb{R}^{\mathrm{m\,x}N_u}$, N_u , N_p , N_c are input, output/ prediction, and constraint horizon, respectively, such that $N_p \geq N_u$, $N_c \leq N_p$-1, and $\mathbf{K}_{\mathrm{LQR}}$ is stabilizing state feedback gain.

This optimization problem has to be solved at each time step t, where $\mathbf{x}_{\mathrm{t+k/t}}$ is the predicted state-vector at time instance t+k with respect to the initial state at the time t. Equation (5.10) implies that after N_u time steps, the control is switched to unconstrained linear quadratic regulator (LQR) depending on the feedback gain $\mathbf{K}_{\mathrm{LQR}}$. The terminal constraint $\mathbf{P}_{\mathrm{N}}$ results as solution of discrete Riccati equation [90], [91]. For constrained infinite-time optimal controllers, the constraints are to be applied to the predicted states and on the calculated control laws for a finite constraint horizon N_c in equation (5.9) . Therefore, the constraints satisfaction in this horizon and system stability should be guaranteed.

Hence, stability of MPC problem depends mainly on the proper choice of N_u , $N_{p,}$ N_c , $\mathbf{P}_{\mathrm{N}}$, $\mathbf{Q}$ and $\mathbf{R}$; therefore, reducing this dependency is useful if the stability and feasibility is still assured. This can be done by imposing some constraints on the N_u , N_p ., N_c , and $\mathbf{P}_{\mathrm{N}}$ to enforce the state trajectory with the time to reach some invariant set at the end of prediction horizon, and let $\mathbf{Q} \geq 0, \mathbf{R} > 0$ to be freely chosen as tuning parameters affect the performance index [92].

The way introduced in the [95] can be followed for the tuning procedure of MIMO systems. If $N_u = N_p \leq N_\infty$ is set (N_∞ is the associated horizon with the maximal controllable invariant set), this condition gives a stable system for a controllable invariant set covers all the bounded state space. It is not necessary here to increase the prediction horizon N_p over N_∞ since that will increase also the complexity of the controller. The above stated constrained optimization problem (equations (5.5)-(5.10)) has to be solved repetitively at each sampling time and for a predetermined prediction horizon N_p . However, only the first optimal control input $\mathbf{u}_{\mathrm{t}}^{*}$ of the resulting N_u optimal control actions $\mathbf{U} = \left\{\mathbf{u}_{\mathrm{t}}^{*},....,\mathbf{u}_{\mathrm{t}+N_u-1}^{*}\right\}$ is to be applied to the plant at the next time step.

Introducing the following relation, derived from (5.5),

$$\mathbf{x}_{\mathrm{t+k|t}} = \mathbf{A}^{\mathrm{k}}.\,\mathbf{x}_{\mathrm{t}} + \sum_{\mathrm{j=0}}^{\mathrm{k-1}} \mathbf{A}^{\mathrm{j}}.\,\mathbf{B}.\,\mathbf{u}_{\mathrm{t+k-1-j}} \tag{5.11}$$

in (5.7) results the following quadratic programming problem (QP):

$$\boldsymbol{J}(\mathbf{x}_{\mathrm{t}}) = \min_{\mathbf{U}} \left(\frac{1}{2}\mathbf{U}^{\mathrm{T}}.\,\mathbf{H}.\,\mathbf{U} + \mathbf{x}_{\mathrm{t}}.\,\mathbf{F}.\,\mathbf{U} + \frac{1}{2}\mathbf{x}_{\mathrm{t}}^{\mathrm{T}}.\,\mathbf{Y}.\,\mathbf{x}_{\mathrm{t}} \right) \tag{5.12}$$

$$\text{s.t. } \mathbf{G}.\,\mathbf{U} \le \mathbf{W} + \mathbf{E}.\,\mathbf{x}_{\mathrm{t}}$$

where $\mathbf{U} = \left\{ \mathbf{u}_t^T, \ldots., \mathbf{u}_{t+N_u-1}^T \right\}^T \in \mathbb{R}^{\mathrm{m\,x}\,N_u}$ is the vector of the optimization variables, and **G**, **W**, **E**, **F**, **H**, and **Y** are obtained from **Q**, **R** and from (equations (5.7)-(5.10)).

Solving this QP problem results implicitly in control actions as function of the states and again only the first optimal control input has to be applied to the controlled system.

The problem in the equation (5.7) with its linear constraints describes the constrained linear quadratic regulation problem, while the reformulated MPC problem in (5.12) is a quadratic programming (QP) optimization problem. Although efficient QP solvers are available to solve the problem (5.12), but evaluating the optimizer online each sampling step may require significant computationally effort. This is the main limitation for the MPC to be widely used in the fast sampling rate applications like electrical drive applications. The authors in [65] introduced the multi-parametric programming technique to move the online computation complexity offline.

5.3.1 Multi-Parametric Programming Approach

Multi-parametric programming (MPP) is a mathematical technique introduced in the control theory to solve constrained optimization problems, where some of the parameters vary in a specified range. Solving the optimization problem as a function of uncertain parameters contributes to the sensitivity analysis of the solution. In sensitivity analysis arise the questions: how the optimal solution varies when the uncertain parameters change and how the optimal solution changes with the right-hand side (RHS) of the constraints. Solving the optimization problem in term of these uncertain parameters is known as a multi-parametric optimization problem. The main distinctive feature of the MPP is in its ability to obtain both the optimization- and objective-variable as an affine function of the varying parameters, and to determine the regions of the space of these parameters where these functions are valid. Hence, the optimization problem needs not to be solved again if these parameters change since the solution has been found as a function of these uncertain parameters. This means, solving the MPC optimization problem using MPP results in dividing the feasible space of the state variables into several regions, and inside each region there is only one linear optimal control law is valid.

Transforming the above formulated QP problem in (5.12) into MPP-Optimization problem can be obtained by introducing a new auxiliary variable:

$$\mathbf{z} = \mathbf{U} + \mathbf{H}^{-1}.\,\mathbf{F}^{\mathrm{T}}.\,\mathbf{x}_{\mathrm{t}}\,;\ \mathbf{z} \in \mathbb{R}^{\mathrm{m\,x}\,N_u} \tag{5.13}$$

Then, the QP optimization problem is transformed to a minimization of the following multi-parametric quadratic problem (mp-QP), equation (5.14).

$$\boldsymbol{J}(\mathbf{x}_{\mathrm{t}})=\min_{\mathbf{z}}\left(\frac{1}{2}\mathbf{z}^{\mathrm{T}}.\,\mathbf{H}.\,\mathbf{z}\right) \tag{5.14}$$

$$\text{s.t. } \mathbf{G}.\,\mathbf{z}\le\mathbf{W}+\mathbf{S}.\,\mathbf{x}_{\mathrm{t}}\,,\ \ \mathbf{S}=\mathbf{E}+\mathbf{G}.\,\mathbf{H}^{-1}.\,\mathbf{F}^{\mathrm{T}} \tag{5.15}$$

where $\mathbf{z}$ is a new vector of the optimization variables, and $\mathbf{x}_t$ is the vector of the free parameters.

Now the free parameters $\mathbf{x}_t$ in this formulation only exist in RHS of the constraints, whereas they presented in both the cost function and the RHS of the constraints of the former formulation. Solving the optimization problem (5.14) results in **z** and therefore in **U**. Hence, the optimizer **U** can be obtained as affine functions of the state variables $\mathbf{x}_t$ for the complete feasible space of $\mathbf{x}_t$. This can be done by solving the optimization problem using Karush-Kuhn-Tucker (KKT) optimality conditions [66]. Finally and as solution of the MPP, the state space will be divided into subspaces or polytopes according to the active and non-active constraints and inside each polytope there is one optimal linear control law is valid:

$$\mathbf{u}_{\mathrm{t}}=\mathbf{u}^{*}_{\mathrm{t}}(\mathrm{x}_{\mathrm{t}})=\mathbf{F}^{\mathrm{i}}.\,\mathbf{x}_{\mathrm{t}}+\mathbf{G}^{\mathrm{i}} \tag{5.16}$$

The subscript □$^{\mathrm{i}}$ refers to region index, where this control law is valid. The matrices $\mathbf{F}^{\mathrm{i}},\mathbf{G}^{\mathrm{i}}$ are the control law coefficients.

For further information about dividing the feasible space of the free parameter in regions and finding the optimal control law the author refers to [65], [66].

5.3.2 Stability and Feasibility Constraints

The main problem of MPC using RHC policy is that it does not guarantee the stability of the system in general. In order to make a MPC problem stable, certain conditions (e.g., terminal set constraints) have to be added to the original optimization problem. The necessary stability constraints for MPC formulation can be categorized into Lyapunov and non-Lyapunov dependent stability constraints, as in Figure 5.2, and stated as follows:

Remark: According to Lyapunov stability definition, a system is called stable if it is possible to control the state trajectory for infinite time to the origin [96]. Here it is to refer that under this definition the oscillatory system is considered also as a stable system, which makes Lyapunov stability condition in many cases not suitable for drive applications (e.g. electrical vehicles).

- Terminal constraints [85]: The following stability constraints on the state variables are to be added: $\mathbf{x}_{t+Np|t} = 0$. The main drawback of using terminal constraints is that, the control effort required to control the state to the origin can be large (especially for a short prediction horizon) and could cause an infeasibility of the problem.
- Infinite output prediction horizon [84], [86], [87]: If the system is asymptotic stable and $N_p = \infty$, then no more stability constraints are required.
- Terminal weighting matrix [88]: By choosing the terminal weighting matrix $\mathbf{P}_N$ in the cost function as a solution of the Riccati inequality equation, the system stability can be guaranteed.

- Invariant terminal sets [89]: The idea here is to relax the already mentioned terminal constraint $\mathbf{x}_{t+Np|t} = 0$ into a set of membership constraints ($\mathbf{x}_{t+Np|t}$) belongs to an invariant set $\mathbf{\Omega}$, where $\mathbf{u}_{t+k|t} = -\mathbf{K}_{LQR} .\mathbf{x}_{t+k|t}$, for $k \geq N_p$. This condition assures also stability for a system with the uncertainty.
- Contraction constraints [87]: Rather than relying on the optimal cost function as a Lyapunov function to prove the stability, the idea here is to assure explicitly that the state $\mathbf{x}_t$ is decreasing (is forced to be decreased) with the time in some norm: $||\mathbf{x}_{t+1|t}|| \leq \alpha.||\mathbf{x}_t||$, $\alpha < \mathbf{1}$. It also guarantees the stability for a system with the uncertainty.
- Uncontrollable system matrix **A**: For an infinite prediction horizon and if the system matrix **A** is stable (i.e.: it leads to a controllable system), the optimization problem can be reformulated into a finite prediction horizon problem by adding the terminal state penalty matrix $\mathbf{P}_N$, which satisfies the discrete Lyapunov equation. In case that the matrix **A** is unstable, the open loop state trajectory should be restricted to the stable subspace of **A** for $k > N_p$. However, the unstable modes of **A** cannot be allowed to proceed without a feasible control action. In this case, some extra constraints on the basis vectors of unstable part of **A** are to be imposed, like $\mathbf{A}_u^T.\mathbf{x}_N = 0$, where the matrix $\mathbf{A}_u{}^T$ is composed of the basis vectors of the unstable subspace of system matrix **A**.

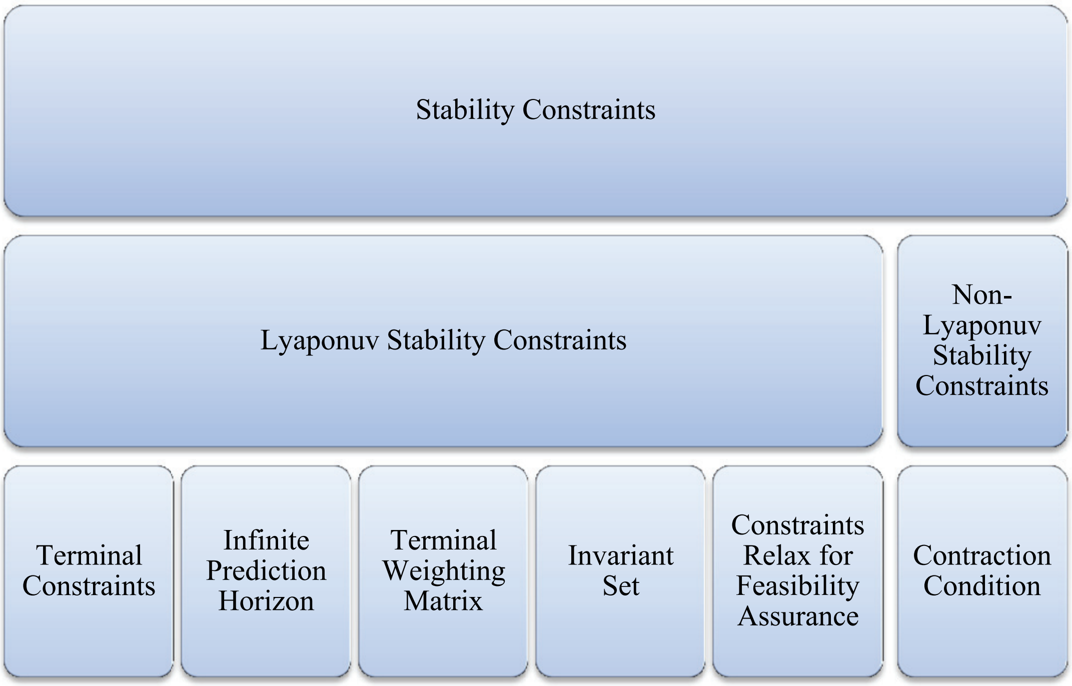

Stability Constraints
Lyaponuv Stability Constraints
Non-Lyaponuv Stability Constraints
Terminal Constraints
Infinite Prediction Horizon
Terminal Weighting Matrix
Invariant Set
Constraints Relax for Feasibility Assurance
Contraction Condition

Figure 5.2: MPC stability constraints

By considering the constraints on the inputs the feasibility issue arises, which is a necessary condition for the stability. The feasibility of a problem can be defined as the possibility of finding a control input of the input space which can drive the state trajectory to the controllable region of the state space. Solve the feasibility problem could be addressed in case that the (state, input) constraints cause the infeasibility; the state and output constraints can be considered as soft constraints and relaxed, whereas input constraints stay as hard constraints and should not be violated. Output constraints, however, frequently do not represent hard physical bound. Rather, they often represent desired ranges of operations that can be violated if necessary, where the input constraints represent physical limitations, [74], [84].

5.4 MPT Toolbox

Formulating the MPC problem and considering all the constraints on the system variables is quite time consuming to be done by hand for each problem, so it is necessary to employ a toolbox to reduce the consumed time during the preparation stage for the MPC problem. Multi-parametric toolbox (MPT) and hybrid systems description language (HYSDEL) toolbox developed at the automatic control laboratory in Zurich University ETH [77], [78], provide powerful tools to formulate the MPC problems in an intuitive and a user-friendly way.

MPT toolbox can be defined as a collection of packages works under Matlab from The Mathworks^{TM} designed to deal with the optimization problems with multi-parametric programming and introduces the necessary interfaces to some of the free and commercial solvers to solve linear, quadratic, integer and mixed-integer optimization problems. In addition to the control synthesis, the toolbox can also be employed for stability analysis, verification and simulation of MPC-based strategies [80]. Finally, the MPT toolbox is also capable to convert the MPC controller into a real time executable code to be used later in the implementation.

MPC problem formulation using the MPT toolbox can be categorized into two steps; in the first one, the system model has to be formulated in form of linear time-invariant (LTI) or piece-wise affine (PWA) model. To deal with hybrid systems (systems with continuous and discrete inputs) with more flexibility, HYSDEL toolbox could also be used in this step to generate PWA system structure introduced to MPT toolbox for later analysis and processing. In the next step, the definition of the problem structure (cost function with its weighting factors and norm), feasibility and stability constraints are to be done.

MPT toolbox is capable of dealing with two types of discrete-time dynamical systems:

- Linear Time-Invariant (LTI) dynamics:

Discrete-time LTI dynamics can be captured by the following linear relations:

$$\begin{aligned} \mathbf{x}_{k+1} &= \mathbf{A}.\,\mathbf{x}_k + \mathbf{B}.\,\mathbf{u}_k \\ \mathbf{y}_k &= \mathbf{C}.\,\mathbf{x}_k + \mathbf{D}.\,\mathbf{u}_k \end{aligned} \tag{5.17}$$

The constraints on the state, input, and output variables are to be provided as inequalities or polyhedral sets in the beginning.

$$\begin{aligned} &\mathbf{x} \in \mathbf{X} \subset \mathbb{R}^n, \mathbf{u} \in \mathbf{U} \subset \mathbb{R}^m, \mathbf{y} \in \mathbf{Y} \subset \mathbb{R}^p \\ &\mathbf{X} = \left\{ \mathbf{x} \mid \mathbf{x}_k \in \mathbf{X}, \mathrm{k} \geq 0 \right\} \\ &\mathbf{U} = \{ \mathbf{u} \mid \mathbf{u}_k \in \mathbf{U}, \mathrm{k} \geq 0 \} \\ &\mathbf{Y} = \{ \mathbf{y} \mid \mathbf{y}_k \in \mathbf{Y}, \mathrm{k} \geq 0 \} \end{aligned} \tag{5.18}$$

- Piecewise-Affine (PWA) dynamics:

PWA formulation to represent a non-linear system is more helpful than LTI if it is possible to reformulate the system dynamic in a combination of linear dynamics, where each dynamic D^{i} takes the following form in equation (5.19).

$$\begin{aligned}
\mathbf{x}_{k+1} &= \mathbf{A}^{i} \,.\, \mathbf{x}_{k} + \mathbf{B}^{i} \,.\, \mathbf{u}_{k} + \mathbf{f}^{i} \\
\mathbf{y}_{k} &= \mathbf{C}^{i} \,.\, \mathbf{x}_{k} + \mathbf{D}^{i} \,.\, \mathbf{u}_{k} + \mathbf{g}^{i} \\
&\text{if} \begin{bmatrix} \mathbf{x}_{k} \\ \mathbf{u}_{k} \end{bmatrix} \in \mathrm{D}^{i}
\end{aligned} \tag{5.19}$$

In the same way like in LTI systems, the constraints on the state, input, and output variables are to be provided as inequalities or polyhedral sets in the beginning.

$$\begin{aligned}
&\mathbf{x} \in \mathbf{X} \subset \mathbb{R}^{n}, \mathbf{u} \in \mathbf{U} \subset \mathbb{R}^{m}, \mathbf{y} \in \mathbf{Y} \subset \mathbb{R}^{p} \\
&\mathbf{X} = \left\{ \mathbf{x} \mid \mathbf{x}_{k} \in \mathbf{X},\ k \geq 0 \right\} \\
&\mathbf{U} = \{ \mathbf{u} \mid \mathbf{u}_{k} \in \mathbf{U},\ k \geq 0 \} \\
&\mathbf{Y} = \{ \mathbf{y} \mid \mathbf{y}_{k} \in \mathbf{Y},\ k \geq 0 \}
\end{aligned} \tag{5.20}$$

Moreover, new constraints are to be provided here to determine the area (subspace) of the introduced state-input space where the dynamic (D^{i}) is valid.

$$\text{if} \begin{bmatrix} \mathbf{x}_{k} \\ \mathbf{u}_{k} \end{bmatrix} \in \mathrm{D}^{i} \Leftrightarrow \mathbf{G}_{x}^{i} \,.\, \mathbf{x}_{k} + \mathbf{G}_{u}^{i} \,.\, \mathbf{u}_{k} \leq \mathbf{G}^{i} \tag{5.21}$$

Representing the model dynamics using PWA will help to represent the nonlinearity and the discrete nature of an inverter in an effective way. For example, as the two-level inverter has eight switching possibilities, hence the PWA formulation is useful here to define a specific dynamic for each switching state.

MPT-Toolbox Limitations

Although the MPT toolbox is powerful enough, all MPC formulations are not handled by the toolbox, and there are some exceptions where the toolbox is still limited. This is shown in the next hierarchical chart, Figure 5.3.

The second and third level of the chart represents the MPT possibilities, whereas the fourth one represents some of its limitations. For example, it is not possible to solve the MPC optimization problem for an infinite prediction horizon with discrete inputs. For practical aspects point of view, it is useful to formulate the MPC problem to consider some issues like free-offset reference tracking and delay compensation (with continuous/ discrete control inputs), which are not covered explicitly in the MPT, Figure 5.4. For that reason the state space representation of the system model and the optimization problem are to be reformulated again in a proper way before they are introduced to the MPT, as explained in section 6.3. Besides to these limitations of MPT, the introduced problem has to be formulated in a way that all the quantities do not exceed the range [-10, 10], [77]. Therefore, a scaling or normalization task for the machine model should be done beforehand [110]. Furthermore, integrating the input or output disturbance using MPT as shown in the equations (5.17), (5.19) is not possible as the case in the generalized predictive control (GPC) using controlled auto-regressive moving average model (CARIMA) model [76]. Therefore, reformulating the optimization problem as explained in section 6.5.4 is also necessary.

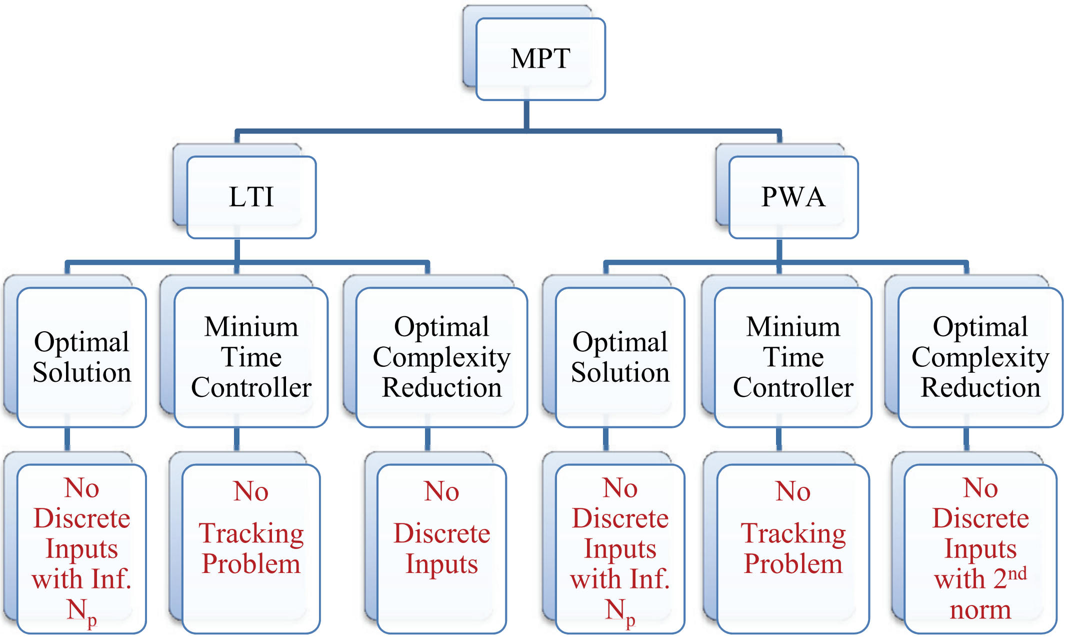

MPT
LTI
PWA
Optimal Solution
Minium Time Controller
Optimal Complexity Reduction
Optimal Solution
Minium Time Controller
Optimal Complexity Reduction
No Discrete Inputs with Inf. N_p
No Tracking Problem
No Discrete Inputs
No Discrete Inputs with Inf. N_p
No Tracking Problem
No Discrete Inputs with 2nd norm

Figure 5.3: MPT toolbox capabilities and limitations

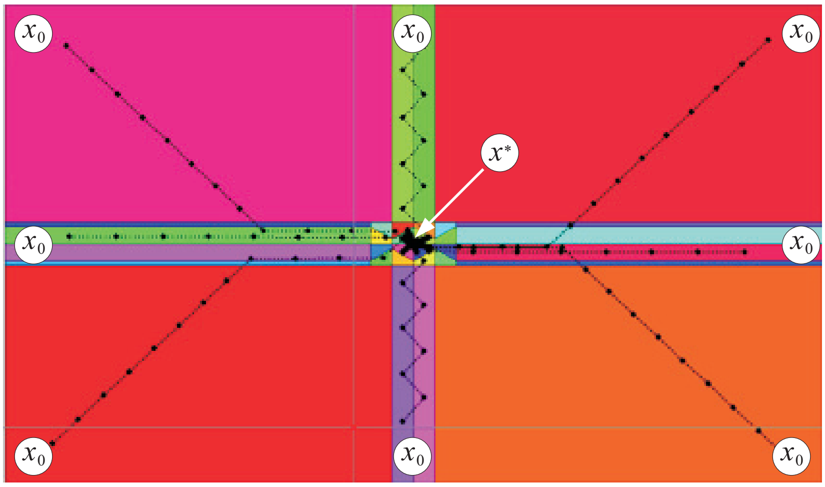


Figure 5.4: Reference tracking procedure using the MPT toolbox

Figure 5.4 shows the origin-tracking emulation procedure using the MPT toolbox from different initial points in the state space.

From here, formulating an MPC problem using the state space presentation can be stated in the following steps:

- Describe the plant as a linear time-invariant model or as a piece-wise affine model. This model is used later to predict the future behavior of the plant, where the state and output constraints should be not violated.
- Formulate the optimization problem as a cost function of a linear, a quadratic, or an infinite norm with a set of inequality constraints on the control input and state variables.
- Add the stability constraints.
- For an explicit MPC controller the linear/ quadratic programming optimization problem has to be further reformulated using the multi-parametric programming approach, as already explained.
- For an implicit MPC controller the optimization problem is to be solved at each sampling time with respect to all provided constraints.
- Follow the receding horizon concept to apply the first optimal control action.

More information about using MPT and HYSDEL toolboxes is available in the attached manual of the toolboxes [77], [78], and no need here to go through more details.

5.4.1 The Evaluation Procedure

Employing the multi-parametric programming technique to solve the MPC optimization problem explicitly, the optimal solution will result in a set of unsorted polytopes and the associated control laws, which can be reformulated for later implementation as look-up table (LUT). For online evaluation of the Exp-MPC, the controller has to determine in which region or polytope the current measured state variables are located and to evaluate the corresponding control law for that region. Different strategies were introduced in the literature to reduce the online complexity, which can be summarized in the Table 5.I [76].

Table 5.I: Comparison between different evaluation strategies

Algorithm	Controller Structure	Maximum Runtime
Exhaustive Search.	225 Polytopes	163 μsec
Optimal Complexity Reduction	60 Polytopes	94 μsec
Minimum Time Controller	9 Polytopes with 2 Iterations	27 μsec
Binary Search Tree	225 nodes, 10 levels	4.7 μsec

Table 5.I shows the maximum run time of different algorithms to evaluate the explicit solution of MPC problem for an induction machine controlled in synchronous coordinates. The evaluation procedure was done on the same control platform under identical conditions [76]. As the resulting polytopes are usually unsorted, the worst search case is the exhaustive search

by going through all the polytopes one after another searching for the location of the measured state variables. This search method and due the complete enumeration of all the polytopes is of course not time efficient.

The design procedure of both optimal complexity reduction controller and minimum time controller intends to solve the optimization problem sub-optimally to get a reduction in the final explicit solution (i.e.: the number of the resulting regions) of the MPC optimization problem.

The optimal complexity reduction method introduced in [71] focuses on finding a minimal representation of the PWA models, which is equivalent to the original one and results in a minimum number of regions. According to the receding horizon control policy the first control input of the resulting optimal control sequence is applied to the plant while the rest is discarded, and the optimization problem is solved again in the next sampling instance. As only the first control input is employed in the control task, it makes sense to combine the polytopes whose first control inputs of the optimal sequence are identical. However, the resulting polytope should be also convex. The proposed algorithm merges several regions of the same control laws as skillfully as possible under the consideration of the convexity constraints. However, this method is time consuming for both offline and online calculations and not applicable for discrete problem with quadratic cost function, Figure 5.3.

The online computation complexity of the Exp-MPC increases proportionally to the number of regions, which depends mainly on the length of the prediction horizon and the number of the control inputs [73]. However, number of the control inputs is a system-dependent parameter and cannot be chosen freely. Hence, the main factor to reduce the online complexity is to solve the optimization problem with a short prediction horizon. Minimum time controller algorithm contributed in [73] exploits this fact and solve 1-step optimization problem with a varying terminal set constraint iteratively. The algorithm starts by assigning the invariant set of unconstrained LQR problem as an initial terminal set constraint to the MPC optimization problem. The optimization problem is solved using the multi-parametric programming approach to get the explicit solution for the first iterative and divide the state space accordingly. In the next step a combination of the resulting regions or polytopes is introduced as a new terminal set and the optimization problem is solved again. This procedure is repeated until the terminal set constraint does not change or grow any more. As a result, several regions may overlap, and the current measured state variables could be contained in more than one region with a different iteration. To address this conflict and guarantee the feasibility and the stability of the optimization problem the control law associated with the region computed at smallest iteration number is selected. This method is actually much time efficient than the previous ones, but unfortunately does not work with tracking problems.

The binary search tree (BST) algorithm [69], [70] is applied on the resulting control structure (without affecting the optimal solution) and builds a binary search tree offline. The algorithm exploits the convexity of resulting polytopes and the dividing hyperplanes to build a balanced tree-like structure with a minimum depth. To traverse the tree, only one affine function has to be evaluated at each tree node. Depending on the sign of the evaluated function the decision is taken to select the right or the left sub-tree for the next branching. The optimal unique control laws are located in the final leaf nodes. This algorithm introduces the best possible reduction in the evaluation time of the explicit solution of MPC. However, it does not work for a discrete model with quadratic cost function. Choosing the Binary Search Tree (BST) for the final evaluation procedure, Table 5.II shows the offline computation time for an induction machine example and under different conditions and formulations. The simplified machine model (without considering the model nonlinearities) is introduced to the optimization problem with different norms and prediction horizons. The machine is controlled using a two-level inverter indirectly for the first three controllers (using a PWM modulator) and directly for the last one. More details about the system model and the controller structure are introduced in chapter 6.

Table 5.II: The offline computation time for different MPC examples

Type	Free-Offset Tracking	Delay Compensation	Norm	NP	Regions	Control laws	Average Offline Calculation Time
SMPC	No	No	2	2	63	7	12 sec
SMPC	No	Yes	2	2	146	7	12.5 sec
SMPC	No	Yes	2	3	372	7	4 min
DMPC	Yes	Yes	1	3	1476	25	22 Hours

The symbol ‘No’ means for example that the free-offset tracking formulation was not used for that case, ‘Yes’ refers to the case where it was used. It can be seen that the offline complexity of the Exp-MPC is exponentially increased; however the maximum evaluation time for all controllers using the BST does not exceed 5µsec. Figure 5.5 summarizes the complete design and evaluation cycle of the explicit solution of MPC.

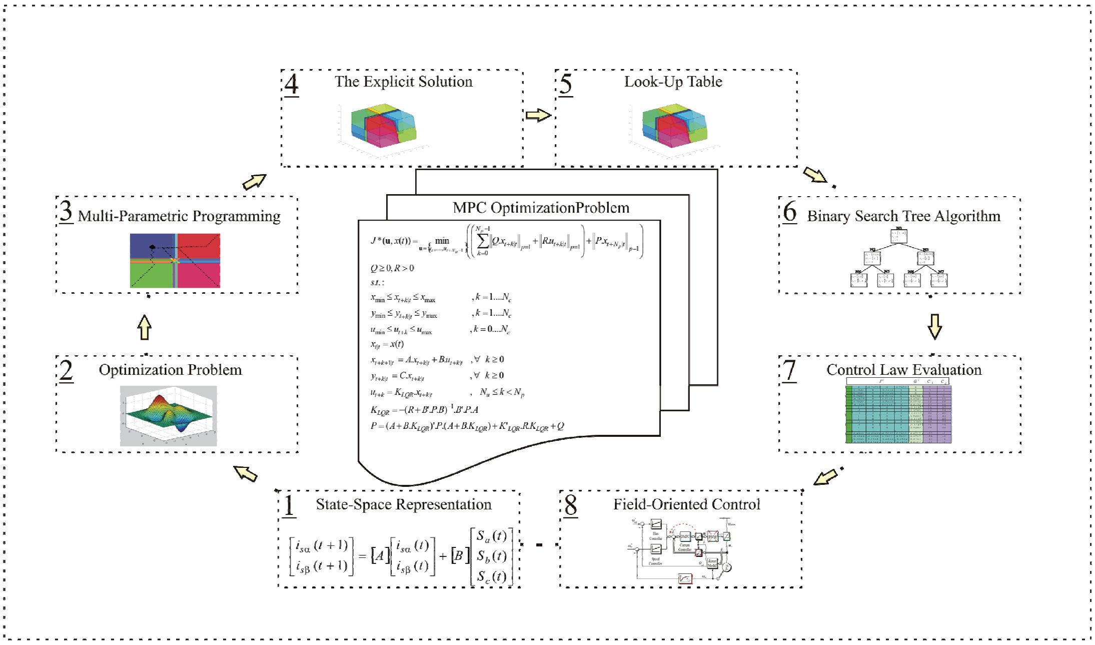


Figure 5.5: The explicit solution of MPC, design and evaluation procedure

5.5 Conclusion

The powerfulness of using the multi-parametric programming technique (MPP) to address the model-based predictive control (MPC) problems appears in moving the online computation complexity of the MPC problem offline, and no optimizer is ever called online. Therefore, the online computation effort is small and reduced to a mere function evaluation problem. As soon as the measurements of the states are available, the corresponding region and control law can be obtained by evaluating some of linear inequalities and an affine function, respectively. For more information about formulation and implementation procedures of the MPC optimization problem, the author refers to [76], [77], [80], and [81].

6 THE EXPLICIT SOLUTION OF MODEL-BASED PREDICTIVE CONTROL

6.1 Introduction

Solving the model-based predictive control problem using multi-parametric programming approach will reduce the online complexity of the implementation and move the more complex calculations offline. The offline procedure using the multi-parametric programming divides the state space of the controlled variables into several regions and only one optimal control law inside each region is valid. The region coordinates and the associated control laws are stored in a look-up table to be evaluated later. For the online implementation, one has to find where the current controlled state variables are located (i.e.: in which region of the feasible space), and to evaluate the associated control law. Using a high efficiency evaluation procedure such as binary search tree [69], the evaluation time is reduced reasonably [76]. It is not in scope of this work to explain the process of formulating and solving the explicit solution using MPT toolbox; for that the reader is directed to [77], [80] for basic information about MPT toolbox and basic concepts behind it supported with some examples. This chapter introduces to different issues related to delay compensation, free-offset tracking, and dealing with linear and non-linear systems with the Exp-MPC controller. Furthermore, direct and indirect control schemes of the AC machines are distinguished and discussed in this chapter with the Exp-MPC. This chapter does not describe the way of reducing the final explicit solution of MPC, which is covered in chapter 7, using the polynomial approximation [111].

6.2 State-Of-The-Art of Exp-MPC for Electric Drive Applications

The explicit solution of model-based predictive control (Exp-MPC) is introduced in the literature extensively, especially in last three years and for a different kind of machines and for different applications. Furthermore, similar ideas were repeated and applied in different applications for induction machines [100], [106], [107], synchronous machines [101], [103], [104], [99], with field-oriented control scheme (FOC) [76], or direct torque scheme (DTC) [103] with two or multilevel inverters in a direct [81], [100] or indirect way [105]. To go through the comparison between the contributions done in this direction so far, and to introduce to the new contribution of this work, some factors are to be considered and highlighted. First of all, working with Exp-MPC requires getting a linear time-invariant model of the plant with linear constraints on the state and input variables, adding the stability constraints, and solving the optimization MPC problem using the multi-parametric programming (MPP) approach [65]. Solving the MPP-Optimization problem will divide the state space into several regions and inside each region there is only one valid optimal control law. The resulting region coordinates and the associated control laws represent the explicit solution of the MPC problem, which is to be introduced to the implementation stage as a look-up table. The design complexity of the predictive controller and the final size of the resulting solution depend on many factors, like number of state and input variables, coupling and embedded relationship between state variables, system constraints, and finally weighting matrices in the cost functions. Complexity increase of the MPC optimization problem is associated with an increase of the number of the resulting regions. More generated regions means more memory space is required to store the explicit solution, as well as the evaluation procedure of the associated optimal control laws will be time consuming. In [76] a very good comparison was done between different evaluation strategies on the same control platform to introduce the binary search tree BST [69] as the best choice for drives applications. Still in some cases, where the final size of explicit solution is very high, the implementation using BST became very challenging [82]. To address this problem, a possible solution was presented in [111] by adopting the polynomial approximation. Generally speaking, constraints, prediction and control horizons are to be chosen in such a way that the feasibility and the stability of the controlled system are guaranteed [64]. For that, the complete dynamics of the system should be captured, by a proper selection of both prediction and control horizons N_p and N_u, respectively. That means N_p must be large enough to ensure the terminal states entering the chosen terminal set after N_p. Hence the main difference between the different contributions will be in the way of formulating the model of plant and dealing with some issues like: nonlinearities, delay, and free-offset tracking problems. The introduced works in [76] have covered some issues related to free-offset tracking and delay compensation using state space representation of the MPC controller. The main contribution of the work presented in this thesis is reflected in the way of dealing and formulating the non-linear terms appearing in the fundamental model of AC machines [112]. The cross-coupling and nonlinearities between the state variables in the fundamental model of AC machines arise in term of induced voltages. These voltages (motion- and rotor- induced voltages), are normally ignored [76], [24], [76], [100], considered as an additive disturbance [99], or linearized around some operating points [101], [102], [102] and introduced as a new equation to the system model. In this work, the nonlinear terms are considered without linearization and compensated either inside or outside the MPC controller structure in different configurations. The compensation terms are estimated/ calculated first irrespective of which compensation strategy is used in the final implementation. In [99] the authors got benefit of the existing observer for sensorless control to estimate motor states and input-disturbances. This observer helped to consider the induced voltages and the coupling effect between machine currents of PMSM in rotating coordinates in a similar way to the introduced work; however they are compensated here at once in a simpler and a direct way as it will be discussed with more details [112]. In both works, the nonlinear term is considered as it is without any kind of linearization and it is introduced to the system model in the same way like introducing the reference signal for tracking problem in state space representation. Here, it is assumed that the compensating term will not change within a single sampling period and over the prediction horizon. It follows, that this nonlinear term has to be calculated at each sampling period and introduced to controller as a new state variable. This formulation leads to an internal compensation of the induced voltages inside the controller structure. The basic idea was first applied in [99] and it is here further extended and generalized for different coordinate systems and compared to the external compensation [103]. The validity of the external and internal compensation is further verified and used for direct control of a three-level neutral-point diode-clamped inverter (NPC) in section 6.8, where the external compensation with discrete inputs is not possible any more. Furthermore, a comparative analysis concerning the compensation procedure of these terms in introduced between the proposed methods and other ones in conventional scalar and complex PI controllers.

6.3 MPC Formulation with Delay and Tracking

Delay Compensation:

The system delay caused by the inverter, calculation time, digital system and the measurement sensors will be reflected later on the system response in an oscillatory form and probably causes instability of the system, so it is necessary to consider and compensate this effect in advance.

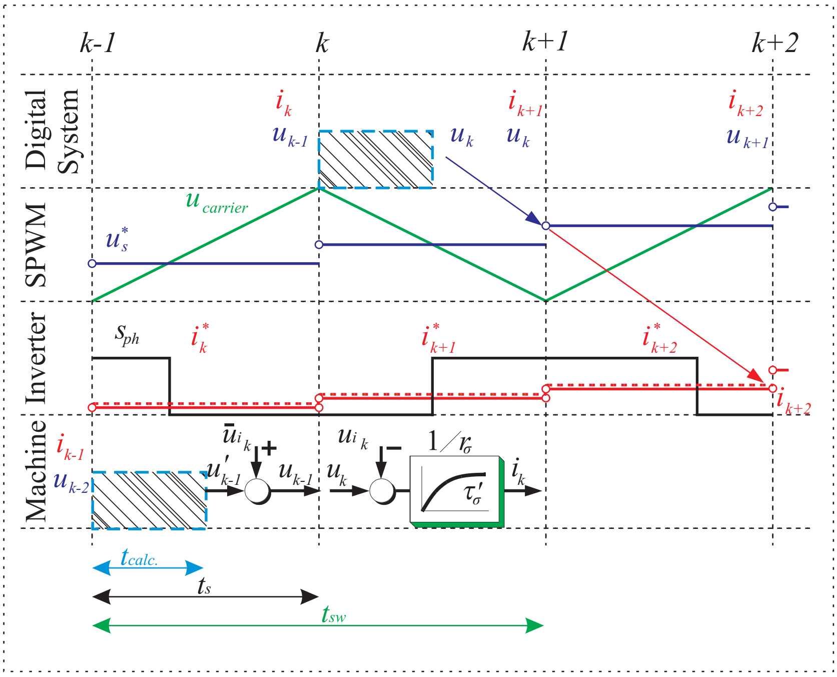


Figure 6.1: System delay occurrence and effect

Figure 6.1 summarizes the system delay occurrences; where the MPC controller with the available knowledge at time step (k) of the current references, the last applied control inputs, and the measured machine currents, has to calculate in advance the optimal control action to be applied to the system (PWM board, Inverter and machine) that the average value of inductor current at time step (k+2) will follow its reference. The MPC computed control input at time step (k) and after some computation delay will be firstly applied to the PWM board by the next sampling time (k+1). The PWM board and due to the high sampling rate (50 MHz), the modulated control input will be applied directly to the inverter and consequently to the machine. However, the effect of the applied average voltage and therefore the resulting average current in the machine inductors are first measured with the second sampling time (k+2). Therefore the complete system delay has to be considered in both the MPC system modeling and the induced-voltage compensation at time step (k), as it is explained later.

The problem with system delay compensation could be solved either online during the implementation as in [82], [104] (for simple cases and fraction of one sampling period), or by considering the delay as an integer number of the sampling periods in the design phase of the MPC problem as in [76]. The author in [99] got benefit of the existing observer for sensorless control to get best delay compensation even for delay within a fraction of sampling period.

To summarize, the delay compensation procedure can be done in one of the following ways:

- Reformulating the discrete LTI/ PWA machine model to include the time delay as an integer number of sampling periods, as introduced in [76].

$$\begin{bmatrix} \mathbf{x}_{k+1} \\ \mathbf{u}_k \end{bmatrix} = \begin{bmatrix} \mathbf{A} & \mathbf{B} \\ \mathbf{0} & \mathbf{0} \end{bmatrix} . \begin{bmatrix} \mathbf{x}_k \\ \mathbf{u}_{k-1} \end{bmatrix} + \begin{bmatrix} \mathbf{0} \\ \mathbf{I} \end{bmatrix} . [\Delta \mathbf{u}_k] \tag{6.1}$$

where **A**, **B** are the system and input matrices, respectively, **I** is the unit matrix.

- Add an estimator (like Kalman Filter) to predict the measured and unmeasured states by knowing the applied control inputs, so the state variables are now available without delay or noise [97], [99].
- An easier way than using the estimator to remove the oscillatory behavior (resulting due to the delay) is by following the first method, even if the time delay is not integer multiplication of the sampling period, and then reducing the controller proportional gain until the oscillation is suppressed completely [97].

Tracking Problem:

For practical aspects, solving the optimization problem for the state variables to follow fixed references (the origin or other fixed reference) can be rarely found in the industrial applications. The authors in [99] introduced fixed references as a way to reduce the complexity and the size of the solution but it is not a possible case in the majority of drive applications, where a time-varying reference signal has to be followed.

- Introducing both the controlled variables and their references as state variables, as in the equation below, and evaluate the cost function to minimize the tracking error is a sufficient procedure to achieve the tracking; but without guarantee for free-offset tracking in steady state.

$$\begin{bmatrix} \mathbf{x}_{k+1} \\ \mathbf{x}^*_{k+1} \end{bmatrix} = \begin{bmatrix} \mathbf{A} & \mathbf{0} \\ \mathbf{0} & \mathbf{I} \end{bmatrix} . \begin{bmatrix} \mathbf{x}_k \\ \mathbf{x}^*_k \end{bmatrix} + \begin{bmatrix} \mathbf{B} \\ \mathbf{0} \end{bmatrix} . [\mathbf{u}_k] \tag{6.2}$$

Under the assumption, that the reference signals do not change within the sampling time and over the prediction horizon, which is the case in steady state $\mathbf{x}^*_{k+1} = \mathbf{x}_k$.

- To fulfill the free-offset tracking requirements, an integral structure should be added to the state-space represented model in one of the following ways:
    - By introducing the last update of the control inputs $\mathbf{u}_{k-1}$ as new state variables and adding the control efforts $\Delta\mathbf{u}_k$ as new inputs to be evaluated later in the objective function [76], [77]. Here, not only the control input could be weighted in the cost function but also the control effort.

$$\begin{bmatrix} \mathbf{x}_{k+1} \\ \mathbf{u}_k \\ \mathbf{x}^*_{k+1} \end{bmatrix} = \begin{bmatrix} \mathbf{A} & \mathbf{B} & \mathbf{0} \\ \mathbf{0} & \mathbf{I} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{I} \end{bmatrix} . \begin{bmatrix} \mathbf{x}_k \\ \mathbf{u}_{k-1} \\ \mathbf{x}^*_k \end{bmatrix} + \begin{bmatrix} \mathbf{B} \\ \mathbf{I} \\ \mathbf{0} \end{bmatrix} . \left[\Delta\mathbf{u}_k\right] \tag{6.3}$$

- By introducing the integral of the state-reference error $\mathbf{e}_I = \int \mathbf{x} - \mathbf{x}^*$ as new state variable to be evaluated later in the cost function, [98].

$$\begin{bmatrix} \mathbf{x}_{k+1} \\ \mathbf{e}_{Ik+1} \\ \mathbf{x}^*_{k+1} \end{bmatrix} = \begin{bmatrix} \mathbf{A} & \mathbf{0} & \mathbf{0} \\ T_0\mathbf{I} & \mathbf{I} & -T_0\mathbf{I} \\ \mathbf{0} & \mathbf{0} & \mathbf{I} \end{bmatrix} . \begin{bmatrix} \mathbf{x}_k \\ \mathbf{e}_{Ik} \\ \mathbf{x}^*_k \end{bmatrix} + \begin{bmatrix} \mathbf{B} \\ \mathbf{0} \\ \mathbf{0} \end{bmatrix} . \left[\mathbf{u}_k\right] \tag{6.4}$$

where T_0 is the normalized sampling time.

Note: Integrating the error integral $\mathbf{e}_I$ as a new state variable and adjusting its weights in the cost function will help to do an online adaptation, force the actual signal to follow the reference one, and keep it inside some band. However, the number of state variables and the problem complexity increase.

Free-Offset Tracking with Delay Compensation:

As a combination of the last equations (6.1) and (6.3), this goal can be achieved with a minimum complexity by reformulating the system model as follows:

$$\begin{bmatrix} \mathbf{x}_{k+1} \\ \mathbf{u}_k \\ \mathbf{x}^*_{k+1} \end{bmatrix} = \begin{bmatrix} \mathbf{A} & \mathbf{B} & \mathbf{0} \\ \mathbf{0} & \mathbf{I} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{I} \end{bmatrix} . \begin{bmatrix} \mathbf{x}_k \\ \mathbf{u}_{k-1} \\ \mathbf{x}^*_k \end{bmatrix} + \begin{bmatrix} \mathbf{0} \\ \mathbf{I} \\ \mathbf{0} \end{bmatrix} . \left[\Delta\mathbf{u}_k\right] \tag{6.5}$$

Following this formulation to compensate the system delay and to eliminate the tracking offset in steady state, the multi-parametric MPC optimization problem has to be reformulated again accordingly. In this formulation the control effort Δ**u** is the new optimization control law and the original state-vector is augmented to include two more vectors, namely the last updated control inputs $\mathbf{u}_l$ = $\mathbf{u}_k$ -1 and the reference states $\mathbf{x}^*_k$. For sake of simplicity, only the new constraints and the modified equations of section 5.3 are repeated here:

$$\begin{aligned} &\mathbf{x}^* \in \boldsymbol{X}^{\mathrm{r}} \subset \mathbb{R}^{\mathrm{r}},\ \Delta\mathbf{u} \in \mathbf{U} \subset \mathbb{R}^{\mathrm{m}} \\ &\boldsymbol{X}^{\mathrm{r}} = \left\{\mathbf{x}^* | \mathbf{x}^*_{\mathrm{t}} \ \in \boldsymbol{X}^{\mathrm{r}}, \mathrm{t} \ge 0\right\} \\ &\mathbf{U} \ = \left\{\Delta\mathbf{u} | \Delta\mathbf{u}_{\mathrm{t}} \in \mathbf{U} \ \ , \mathrm{t} \ge 0\right\} \end{aligned} \tag{6.6}$$

where $\mathbf{x}^*$ is the newly added reference state-vector to the original state vector, and the X^{r} is bounding constraints on $\mathbf{x}^*$.

The control effort Δ**u** is the new optimizer, and **U** is the bounding constraints on Δ**u**.

$$\boldsymbol{J}(\mathbf{U},\mathbf{x}_{\mathrm{t}},\mathbf{x}^{*}_{\mathrm{t}},\mathbf{u}_{\mathrm{l}}) = \min_{\mathbf{U}} \left(\begin{array}{l} \mathbf{x}^{\mathrm{T}}_{\mathrm{t+N_p|t}} \cdot \mathbf{P}_{\mathrm{N}} \cdot \mathbf{x}_{\mathrm{t+N_p|t}} + \mathbf{x}^{*\mathrm{T}}_{\mathrm{t+N_p|t}} \cdot \mathbf{P}_{\mathrm{r}} \cdot \mathbf{x}^{*}_{\mathrm{t+N_p|t}} + \mathbf{u}^{\mathrm{T}}_{\mathrm{l\,t+N_p|t}} \cdot \mathbf{P}_{\mathrm{l}} \cdot \mathbf{u}_{\mathrm{l\,t+N_p|t}} + \\ \sum_{\mathrm{k=0}}^{\mathrm{N_p-1}} \mathbf{x}^{\mathrm{T}}_{\mathrm{t+k|t}} \cdot \mathbf{Q} \cdot \mathbf{x}_{\mathrm{t+k|t}} + \mathbf{x}^{*\mathrm{T}}_{\mathrm{t+k|t}} \cdot \mathbf{Q}_{\mathrm{r}} \cdot \mathbf{x}^{*}_{\mathrm{t+k|t}} + \mathbf{u}^{\mathrm{T}}_{\mathrm{l\,t+k|t}} \cdot \mathbf{Q}_{\mathrm{l}} \cdot \mathbf{u}_{\mathrm{lt+k|t}} + \Delta\mathbf{u}^{\mathrm{T}}_{\mathrm{t+k|t}} \cdot \mathbf{R}_{\Delta} \cdot \Delta\mathbf{u}_{\mathrm{t+k|t}} \end{array} \right) \tag{6.7}$$

$$\mathbf{Q}_{\mathrm{r}} = \mathbf{Q}_{\mathrm{r}}^{\mathrm{T}} \geq 0, \mathbf{P}_{\mathrm{r}} \geq 0 \tag{6.8}$$

where the matrices **Q**r, **P**r are thee associated weighting factors with the new added state variable $\mathbf{x}^{*}$.

$$\mathbf{Q}_{\mathrm{l}} = \mathbf{Q}_{\mathrm{l}}^{\mathrm{T}} \geq 0, \mathbf{P}_{\mathrm{l}} \geq 0 \tag{6.9}$$

where the matrices **Q**l, **P**l are the associated weighting factors with the last update of the control input **u**l.

$$\Delta\mathbf{R} = \Delta\mathbf{R}^{\mathrm{T}} > 0 \tag{6.10}$$

where the matrices Δ**R** are the associated weighting factors with the control effort Δ**u**, and the control input could be evaluated as follows: $\mathbf{u}_k = \mathbf{u}_{k-1} + \Delta\mathbf{u}_k$.

By treating the new augmented state-vector ($\mathbf{x},\mathbf{x}^{*},\mathbf{u}_{\mathrm{l}}$) as a free parameter, the resulting MPC formulation can be recast as a MPP-Quadratic optimization problem (mp-QP). The resulting solution will be a piece wise affine control function of the augmented state-vector with their corresponding regions where these control laws are valid. Hence, the resulting feasible regions will be with (n + m + r) dimensions and the resulting control law function will take the following formulation:

$$\begin{aligned} \Delta\mathbf{u}_{\mathrm{k}} &= \Delta\mathbf{u}^{*}_{\mathrm{k}}(\mathbf{x}_{\mathrm{k}},\mathbf{x}^{*}_{\mathrm{k}},\mathbf{u}_{\mathrm{k-1}}) = \mathbf{F}^{\mathrm{i}} \cdot \mathbf{x}_{\mathrm{k}} + \mathbf{F}^{\mathrm{i}}_{\mathrm{r}} \cdot \mathbf{x}^{*}_{\mathrm{k}} + \mathbf{F}^{\mathrm{i}}_{\mathrm{l}} \cdot \mathbf{u}_{\mathrm{k-1}} + \mathbf{G}^{\mathrm{i}} \\ \mathbf{u}_{\mathrm{k}} &= \mathbf{u}_{\mathrm{k-1}} + \Delta\mathbf{u}_{\mathrm{k}} \end{aligned} \tag{6.11}$$

where the subscript (i) refers to region index, in which this control law is valid, and the matrices $\mathbf{F}^{\mathrm{i}},\mathbf{F}^{\mathrm{i}}_{\mathrm{r}},\mathbf{F}^{\mathrm{i}}_{\mathrm{l}},\mathbf{G}^{\mathrm{i}}$ are the control law coefficients.

6.4 Predictive Current Control of an Induction Machine

6.4.1 Non-Linearity Problem in the Induction Machine Model

Recalling the induction machine equation, which describes the relationship between the stator currents and the stator voltages in rotor-flux coordinates:

$$\boldsymbol{i}_s + \tau'_\sigma \frac{d\boldsymbol{i}_s}{d\tau} = \frac{1}{r_\sigma}\left(\boldsymbol{u}_s - \overbrace{\left(j\omega_s l_\sigma \boldsymbol{i}_s - k_r \left(\frac{1}{\tau_r} - j\omega_m \right) \boldsymbol{\psi}_r \right)}^{\boldsymbol{u}_i} \right) \tag{6.12}$$

where $\tau'_\sigma = \sigma\tau_s$, $\sigma = 1 - \frac{l_m^2}{l_s l_r}, l_\sigma = \sigma l_s$, $\tau_s = \frac{l_s}{r_s}, \tau_r = \frac{l_r}{r_r}, k_r = \frac{l_m}{l_r}, r_\sigma = r_s + r_r k_r^2$.

In conventional PI control, the current controller design depends on using the simplified machine model, equation (6.13), by considering both the motion-induced voltage and the rotor-induced voltage as input-disturbances (the underlined terms in the equation (6.12) respectively). These voltage terms are to be compensated, if necessary, at the output of the controller as feed-forward compensation terms, section 3.3.2.

$$F_s(\mathrm{s}) = \frac{\boldsymbol{i}_s}{\boldsymbol{u}_s} = \frac{1}{r_\sigma}\frac{1}{(\tau_\sigma^{'}\boldsymbol{s}+1)} \tag{6.13}$$

Designing the MPC controller using the MPT toolbox requires introducing the system model to the optimization problem as linear or affine-linear equations with linear system constraints [77]. Furthermore, compensation of the induced-voltages in an MPC control structure formulated in state space representation, means extending the state variables to include both mechanical speed (ω_m) and the rotor flux ($\boldsymbol{\psi}_r$). Here arise the needs to do a linearization of the non-linear terms around some operating points, which restricts the optimality only around these points. However, the complexity of design and tuning procedure of MPC controller under the existing coupling relationship between the state variables will be increased definitely. To avoid this complexity, a possible simplification could be done here on the machine model in a similar way to the conventional PI controller. Here, a less number of states are to be considered, whereas the remaining terms of the equation should be compensated later outside the controller structure. Compensating the induced voltages inside the MPC controller structure (as in the complex PI controller structure) is also here possible, as it will be discussed later. In both cases (internal and external compensation), the machine currents ($\boldsymbol{i}_s$) are only controlled using a MIMO-MPC controller instead of two SISO-PI controllers in the cascaded field-oriented control structure, while the mechanical speed is controlled using PI controller as shown in Figure 6.2.

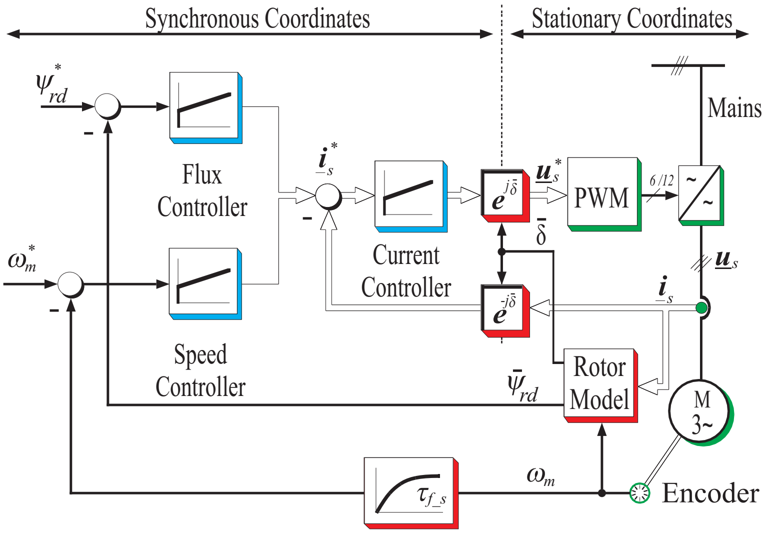


Figure 6.2: Field-oriented control structure using scalar PI controllers

In contrast to the PI controller, in case that the induced voltages are not considered as input-disturbances and not compensated at the output of the PI controller, the PI controller still able to compensate these disturbances by generating the required control input, Figure 6.3.

However, the MPC controller, as it is model-based controller, could not as shown in Figure 6.4. In both Figure 6.3 and Figure 6.4 a step in the ($i_{sq} = 0 \rightarrow 0.4$) current was applied, while the flux producing current was kept at its nominal value ($i_{sd} = 0.35$), and for a large time scale. The controlled induction machine was left unloaded, to show the effect of the back-emf on the tracking procedure when the machine starts rapidly to accelerate.

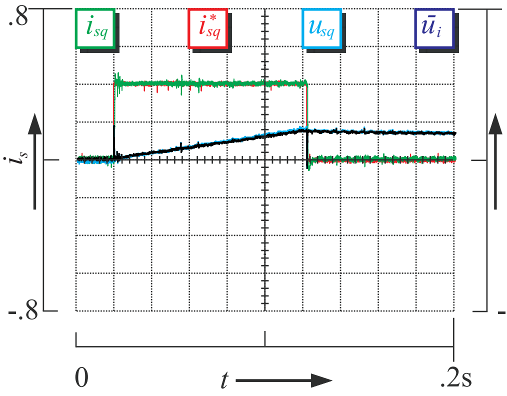


Figure 6.3: IM current response using a SPI$_{dq}$ controller without compensation

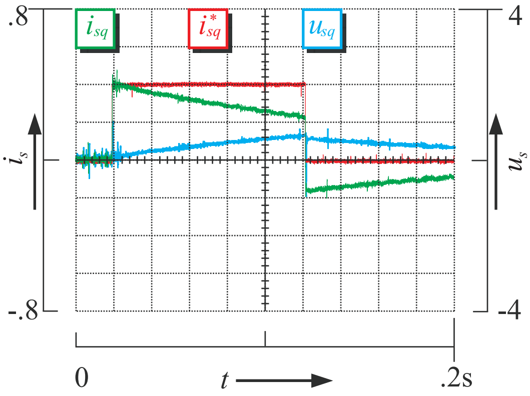


Figure 6.4: IM current response using a MPC$_{dq}$ controller without compensation

The SPI controller without feedforward compensation of the induced-voltages exhibits in Figure 6.3 a perfect tracking performance. Whereas the optimal control input, generated by the MPC controller in Figure 6.4 , was not sufficient to make the machine current (i_{sq}) follows its reference. The reason behind that is explained in the following scenario:

The MPC controller with delay and free-offset tracking formulation, equation (6.3), uses both the currents and the last applied control inputs to determine the region and the associated optimal control inputs, so neglecting the voltages drops on the last applied MPC control inputs leads as in Figure 6.4 to a wrong determination of the region and then to a wrong choice of the control input. Therefore the MPC controller without input-disturbance compensation or correction to the last applied control input results in loss of the tracking. In both cases (compensation or correction) a perfect estimation and compensation of the voltage drops is required to achieve free-offset-tracking MPC controller. As the estimation and compensation procedures of these disturbances are highly affected by the mechanical speed and the used speed filter, a proper estimation procedure independent from the measured speed has to be defined.

6.4.2 Speed Filter Effect on the Estimated Back-emf and the Cross-Coupling Terms

In the speed control loop, the fed back measured speed is to be smoothed via a low pass filter before it is fed to the speed controller. This filter helps to reduce the resulting noise from using a low resolution incremental encoder attached to the motor shaft, Figure 6.5. According to the filter time constant, the dynamic behavior of the speed controller will be affected; Using a small time constant leads to sluggish behavior of the PI speed controller (due to the associated noise), whereas using an extremely large time constant means that the low pass filter is the controlled part instead of the actual system. Finally a compromise should be found to choose the filter time constant. Since the used time constant causes a delay in the measured speed signal accordingly, this delay will affect the estimation procedure of unmeasured machine quantities; both measured speed ω_m and phase currents $\boldsymbol{i}_s$ are used in the rotor model to estimate the rotor flux ψ_{rd} and the coordinates-transformation angle δ, Figure 6.6. Therefore, having some delay in the measured quantities causes an inaccurate estimation of the unmeasured machine quantities, equation (6.14).

$$\begin{aligned} &\psi_{rd} + \tau_r \frac{d\psi_{rd}}{d\tau} = l_m\, i_{sd} \\ &\omega_r = \frac{l_m\, i_{sd}}{\tau_r\, \psi_{rd}} \\ &\omega_s = \omega_r + \omega_m \\ &\delta = \int \omega_s \,.\, d\tau \end{aligned} \tag{6.14}$$

Error in estimating this angle δ will cause an error in calculating the electrical machine quantities used later in control the machine and in calculating both the cross-coupling and the back-emf terms, equation (6.15).

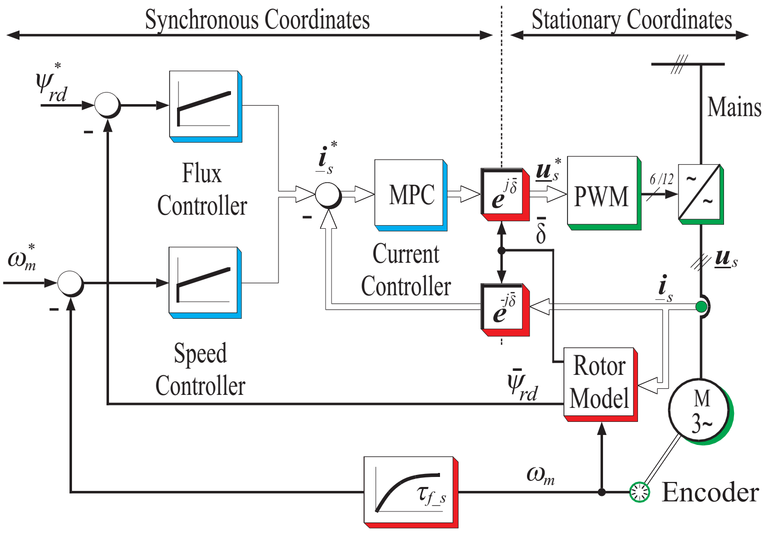


Figure 6.5: The proposed MPC current control structure

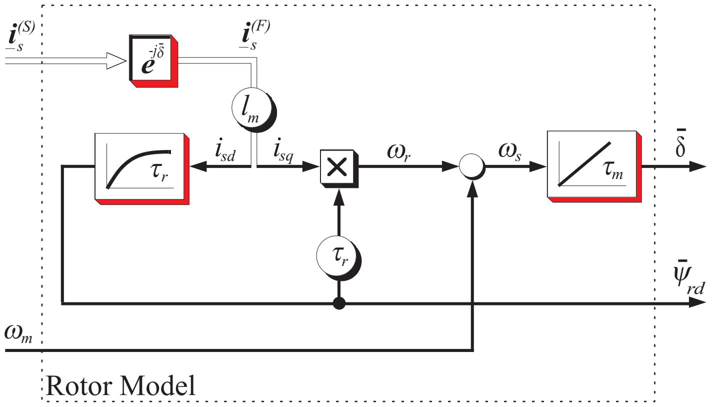


Figure 6.6: Rotor model of an induction machine

$$
\begin{aligned}
&Cross\text{ - }coupling\ term = j\, l_\sigma \omega_s \boldsymbol{i}_s \\
&B_{emf} = \boldsymbol{e}_s = -k_r \left(\frac{1}{\tau_r} - j\omega_m \right) \boldsymbol{\psi}_r \qquad (6.15)\\
&\boldsymbol{u}_i = \boldsymbol{u}_{s\ dis} = Cross\text{ -}coupling\ term + Back\ -emf\ term
\end{aligned}
$$

where ω_m is the filtered measured-speed signal, and $\boldsymbol{u}_i$ is the total induced voltage (motion- and rotor-induced voltages).

Here the following scenarios can be distinguished:

- Using a small time constant for the speed filter τ_{f_s}, leads to a correct estimation of the transformation angle δ, cross-coupling ($j\, l_\sigma \omega_s \boldsymbol{i}_s$) and back-emf ($\boldsymbol{e}_s$) terms, but on the other hand it leads to a sluggish control behavior.

- Using a large time constant for the speed filter τ_{f_s}, leads to a wrong estimation of the angle δ and consequently even the implementation of PI controller with feed-forward compensation (FFC) will not be possible any more.

6.4.3 Field-Oriented Control Structure with Two Speed Filters

Considering the above introduced scenarios arises the need to have two low pass filters on the speed signal with two different time constants, Figure 6.7. A low pass filter with a small time constant τ_{f_δ} is used to have a correct estimation of the transformation angle δ. Second low pass filter with a large time constant τ_{f_e} is required to get a smoothed speed signal used in the speed control loop and in the estimation procedure of the back-emf. However, a perfect estimation of the b-emf cannot be achieved here by using two low pass filters, and a small steady state tracking error is expected.

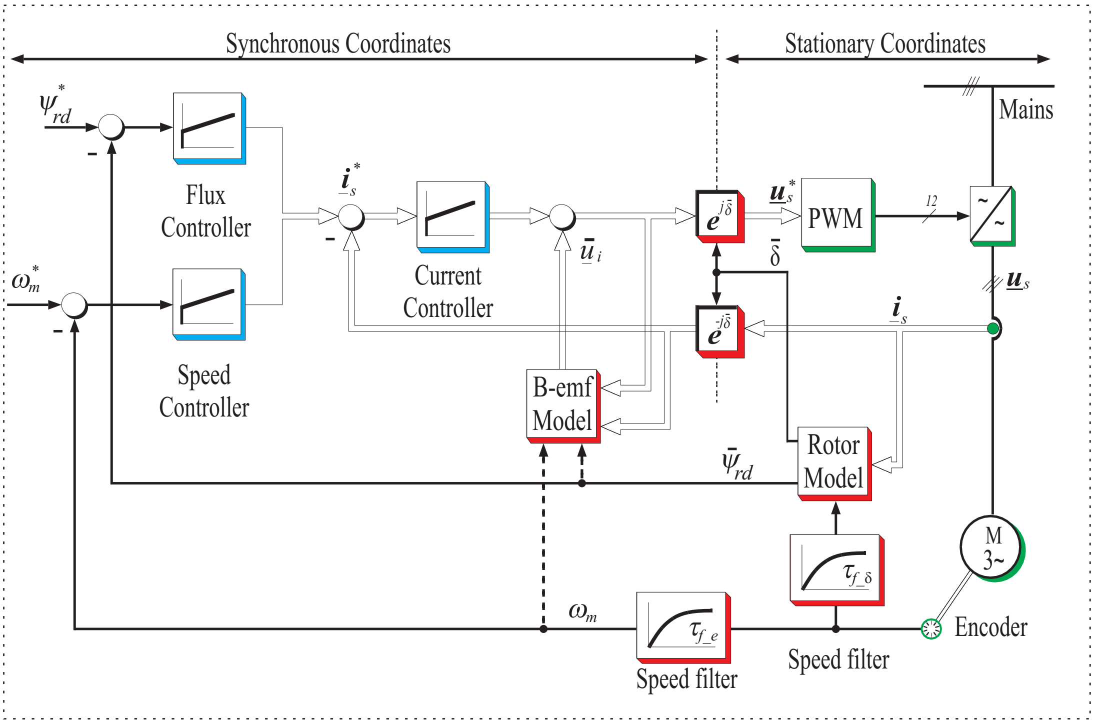


Figure 6.7: Field-oriented control structure with two speed filters

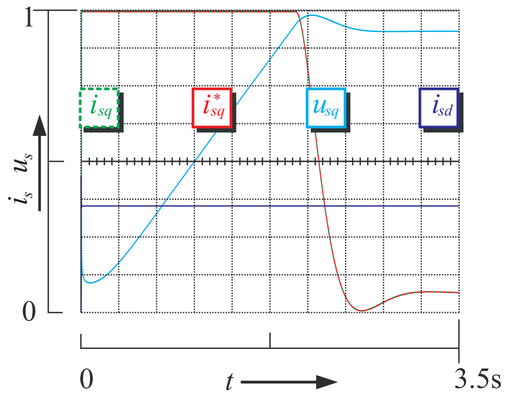


Figure 6.8: IM current response using a SPI_{dq} controller with external compensation

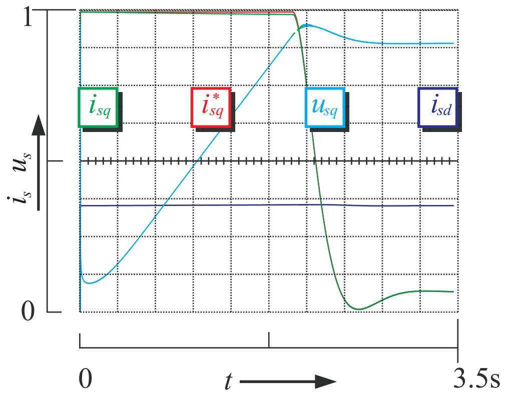


Figure 6.9: IM current response using a MPC$_{dq}$ controller with external compensation

The simulation results show that the SPI current controller with FFC and ($\tau_{f\_\delta} = 1, \tau_{f\_e} = 3$) could compensate this small tracking offset, Figure 6.8. Unfortunately the SMPC (as it is model-based controller and for the above mentioned explanation in section 6.4.1) with $\tau_{f\_\delta} = 1, \tau_{f\_e} = 3$ could not, Figure 6.9. Therefore another way independent from the delayed mechanical speed is required to estimate the back-emf. This observation is further verified by experimental results. The following experimental results show the clear effect of the speed filter time constant on the compensation procedure of the induced voltage in form of feed-forward compensation (FFC) at the output of the controller. Applying a step in the torque producing current ($i_{sq} = 0 \rightarrow 0.4$), while the flux producing current was kept at its nominal value ($i_{sd} = 0.35$) leads to accelerating the machine by the time. If the time window is large enough (time division is 20 msec) as in the following results, the effect of the induced voltage for an unloaded machine will be clearly reflected in the tracking procedure. As already mentioned, the scalar PI controller is able to compensate the voltage drop in the applied control input even if there is no FFC was applied as shown in Figure 6.10. Therefore, applying the compensation to the SPI controller will only reduce the control effort u_{sq} , Figure 6.11, and the coupling effect between the machine currents. Unfortunately, the MPC controller is highly affected with the accuracy of the compensation procedure and therefore with the speed filter time constant. The deviation between the controlled system output and its reference will increase as large as the speed filter time constant is. The MPC controller using the simplified machine model (SMPC) without compensation of the induced voltages, Figure 6.12, exhibits a large deviation between the controlled system output and its reference. Although FFC enhances the tracking procedure of SMPC and reduces the tracking error to some limit, but it is still with large time constants not accepted, Figure 6.13. Hence, reducing the speed filter time constants will reduce the tracking error significantly, Figure 6.14. Figure 6.15 shows the SMPC with the smallest possible speed filter time constants, where a small tracking error is still exist. In this figure, the control output is saturated at the end of the step and the closed control loop is turned to an open control loop, therefore a large tracking error can be noticed here. Form here it is clear, that using two speed filters with two different time constants will enhance the compensation procedure and reduces the tracking error to somehow, but it will not disappear completely. Therefore another estimation procedure independent from the measured mechanical speed is required to get a perfect compensation.

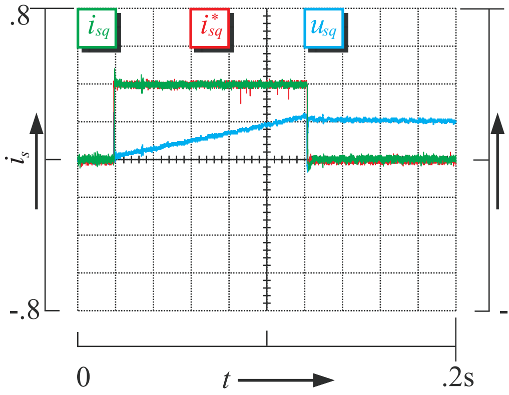


Figure 6.10: IM current response using a SPI$_{dq}$ controller without compensation $(\tau f_{\_\delta} = 16,\ \tau f_{\_e} = 16)$

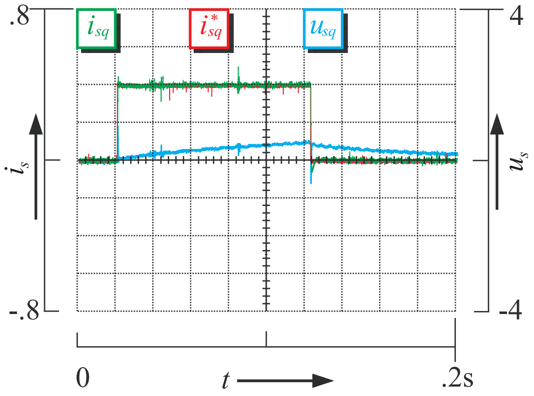


Figure 6.11: IM current response using a SPI$_{dq}$ controller with external compensation $(\tau f_{\delta} = 16,\ \tau f_{e} = 16)$

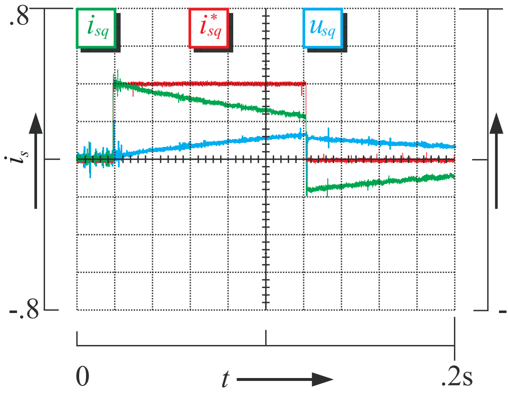


Figure 6.12: IM current response using a SMPC$_{dq}$ controller without compensation $(\tau f_{\delta} = 16,\ \tau f_{e} = 16)$

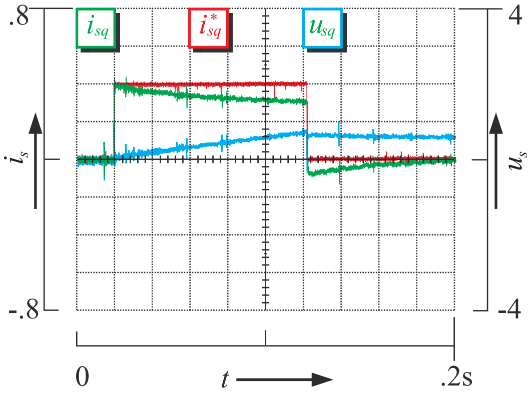


Figure 6.13: IM current response using a SMPC$_{dq}$ with external compensation $(\tau f_{\delta} = 16,\ \tau f_{e} = 16)$

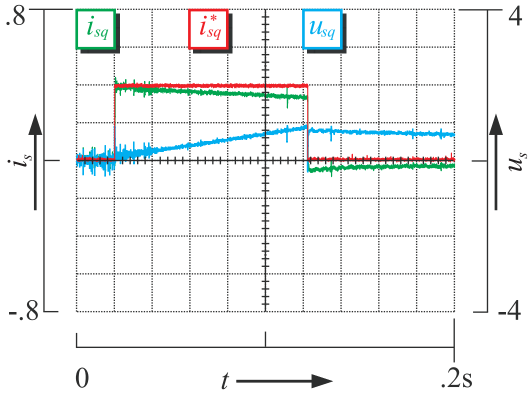


Figure 6.14: IM current response using a SMPC$_{dq}$ with external compensation $(\tau f_{\_\delta} = 1,\ \tau f_{\_e} = 3)$

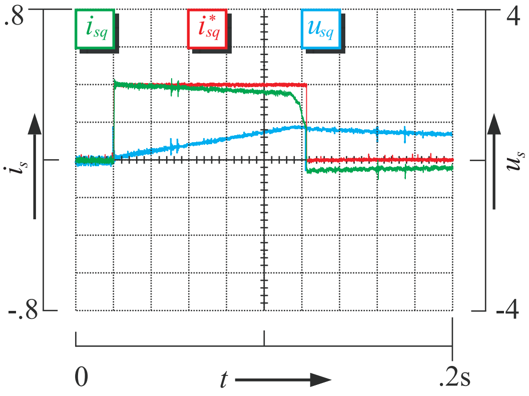


Figure 6.15: IM current response using a SMPC$_{dq}$ with external compensation $(\tau f_{\_\delta} = 1,\ \tau f_{\_e} = 1)$

6.4.4 Estimation Procedure of the Induced Voltages

Using the equivalent circuit of the induction machine [119] in Figure 6.16, the estimation procedure of both induced voltages ($\boldsymbol{u}_i$) (rotor-induced voltage (back-emf) and motion-induced voltage (cross-coupling terms)) at once without using the measured speed can be done by using the following equation (6.17):

$$\frac{d\boldsymbol{i}_s}{d\tau} = -\frac{1}{\tau_\sigma'}\boldsymbol{i}_s + \frac{1}{l_\sigma}(\boldsymbol{u}_s - \boldsymbol{u}_i) \tag{6.16}$$

Thus, the estimated total induced voltage is $\bar{\boldsymbol{u}}_i$:

$$\bar{\boldsymbol{u}}_i = \boldsymbol{u}_s - l_\sigma \frac{d\boldsymbol{i}_s}{d\tau} - r_\sigma \boldsymbol{i}_s \tag{6.17}$$

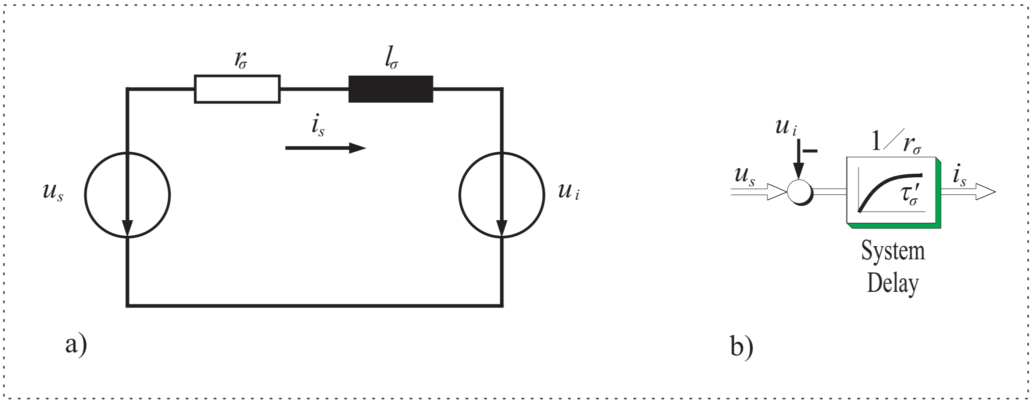


Figure 6.16: Simplified AC machine model

a) Equivalent circuit b) Signal flow graph

Note that all the quantities in this equation are resulting from the estimated angle $\bar{\delta}$ with a filter with small time constant τ_{f_δ}, equation (6.17). For the implementation, it is preferred to have a filter with small time constant on the resulting back-emf to reduce the harmonic of the measured currents at low switching frequencies.

6.5 Indirect Exp-MPC Current Controller for an Induction Machine in Synchronous Coordinates

Controlling the AC machine and the inverter can be done either in a direct or in an indirect way. In the direct control scheme, the controller generates the inverter switching states directly without need for any modulation strategy [3]. In the indirect control scheme, the controller generated control inputs are to be modulated by one of the modulation strategies before being applied to the inverter. The PWM modulator stands in this case to generate the respective inverter switching states. However, an indirect MPC control with solving the optimization problem completely online is time consuming [76]. Therefore, the online evaluation complexity of the implicit MPC controller was moved to be done off-line, using the multi-parametric programming approach, as already explained in section 5.3.1. Furthermore, according to the followed way to compensate the nonlinear terms appearing in the machine model, two cases can be introduced. Compensation of the model nonlinearities outside the controller structure as in FFC introduces to a “MIMO-SMPC controller with an external compensation”, relying on the simplified machine model without the nonlinear terms. While doing the compensation inside the controller structure in an extended formulation of the MPC optimization problem introduces to a “MIMO-EMPC controller with an internal compensation”.

6.5.1 SISO-SMPC Current Controller with External Compensation

The simplest way to design MPC controller is to follow the cascaded control structure in the same way like PI controller with two SISO-SMPC current controllers instead, Figure 6.5. However, designing two SISO-SMPC current controllers depending on the dynamic of stator currents, the motion- and rotor-induced voltages ($\boldsymbol{u}_i$) (underlined terms in the following equation (6.18)) have to be considered and compensated later outside the controller [107], as explained in section 6.5.3.

$$\frac{d}{d\tau}\boldsymbol{i}_s = \left[-\frac{1}{\tau_\sigma'}\right]\boldsymbol{i}_s + \left[\frac{1}{l_\sigma}\right]\cdot\left(\boldsymbol{u}_s - \underline{\left(j\omega_s l_\sigma \boldsymbol{i}_s - k_r\left(\frac{1}{\tau_r} - j\omega_m\right)\boldsymbol{\psi}_r\right)}\right) \tag{6.18}$$

The simplified scalar machine model for SISO control will be:

$$\frac{d}{d\tau}i_s = \left[-\frac{1}{\tau_\sigma'}\right]i_s + \left[\frac{1}{l_\sigma}\right]u_s \tag{6.19}$$

where i_s could be either i_{sd} , or i_{sq} and the same for u_s. The number of generated regions for this model is 17.

6.5.2 MIMO-SMPC Current Controller with External Compensation

Using two SISO-SMPC controllers instead of one MIMO-SMPC controller, the power of the MIMO control in considering the model nonlinearities effectively will be lost. The equation (2.12) introduces three possible state variables (i_{sd}, i_{sq}, ψ_{rd}) could be controlled using a MIMO controller to follow their references. According to that, two cases can be distinguished here:

A. Simplified Machine Model without Considering the Flux as a State Variable

Working with field-oriented control structure and in a non-field weakening region, considering the d-axis rotor flux ψ_{rd} as a state variable will just increase the complexity of tuning procedure of the MPC controller and without a significant improvement in its performance. The next equations describe the machine model with the currents as state variables:

$$\frac{d}{d\tau}\begin{bmatrix} i_{sd} \\ i_{sq} \end{bmatrix} = \begin{bmatrix} -\frac{1}{\tau_\sigma'} & 0 \\ 0 & -\frac{1}{\tau_\sigma'} \end{bmatrix}\cdot\begin{bmatrix} i_{sd} \\ i_{sq} \end{bmatrix} + \begin{bmatrix} \frac{1}{l_\sigma} & 0 \\ 0 & \frac{1}{l_\sigma} \end{bmatrix}\cdot\left(\begin{bmatrix} u_{sd} \\ u_{sq} \end{bmatrix} - \begin{bmatrix} -l_\sigma\omega_s i_{sq} - \frac{k_r}{\tau_r}\psi_{rd} \\ l_\sigma\omega_s i_{sd} + k_r\omega_m\psi_{rd} \end{bmatrix}\right) \tag{6.20}$$

The output equation is:

$$\begin{bmatrix} i_{sd} \\ i_{sq} \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}\cdot\begin{bmatrix} i_{sd} \\ i_{sq} \end{bmatrix} + \begin{bmatrix} 0 & 0 \\ 0 & 0 \end{bmatrix}\cdot\begin{bmatrix} u_{sd} \\ u_{sq} \end{bmatrix} \tag{6.21}$$

These terms $\begin{bmatrix} -l_\sigma \omega_s i_{sq} - \dfrac{k_r}{\tau_r}\psi_{rd} \\ l_\sigma \omega_s i_{sd} + k_r \omega_m \psi_{rd} \end{bmatrix}$ are neglected in the MIMO-SMPC problem formulation and compensated later in a proper way, as explained in section 6.5.3.

B. Introducing the d-axis Rotor Flux as a New State Variable

Extending the state-vector to include one more variable, namely the rotor flux, only as state variable and not as output variable, results in the following formulation:

$$\frac{d}{d\tau}\begin{bmatrix} i_{sd} \\ i_{sq} \\ \psi_{rd} \end{bmatrix} = \begin{bmatrix} -\dfrac{1}{\tau_\sigma^{'}} & 0 & \dfrac{k_r}{l_\sigma \tau_r} \\ 0 & -\dfrac{1}{\tau_\sigma^{'}} & 0 \\ \dfrac{l_m}{\tau_r} & 0 & -\dfrac{1}{\tau_r} \end{bmatrix} . \begin{bmatrix} i_{sd} \\ i_{sq} \\ \psi_{rd} \end{bmatrix} + \begin{bmatrix} \dfrac{1}{l_\sigma} & 0 \\ 0 & \dfrac{1}{l_\sigma} \\ 0 & 0 \end{bmatrix} . \left(\begin{bmatrix} u_{sd} \\ u_{sq} \end{bmatrix} - \begin{bmatrix} 0 & -l_\sigma & 0 \\ l_\sigma & 0 & k_r \\ 0 & 0 & 0 \end{bmatrix} . \begin{bmatrix} \omega_s \, i_{sd} \\ \omega_s \, i_{sq} \\ \omega_m \, \psi_{rd} \end{bmatrix} \right) \tag{6.22}$$

The output equation is:

$$\begin{bmatrix} i_{sd} \\ i_{sq} \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix} . \begin{bmatrix} i_{sd} \\ i_{sq} \\ \psi_{rd} \end{bmatrix} + \begin{bmatrix} 0 & 0 \\ 0 & 0 \end{bmatrix} . \begin{bmatrix} u_{sd} \\ u_{sq} \end{bmatrix} + \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} . \begin{bmatrix} \omega_s \, i_{sd} \\ \omega_s \, i_{sq} \\ \omega_m \, \psi_{rd} \end{bmatrix} \tag{6.23}$$

Here also the terms $\begin{bmatrix} 0 & -l_\sigma & 0 \\ l_\sigma & 0 & k_r \\ 0 & 0 & 0 \end{bmatrix} . \begin{bmatrix} \omega_s \, i_{sd} \\ \omega_s \, i_{sq} \\ \omega_m \, \psi_{rd} \end{bmatrix}$ are neglected in the SMPC problem formulation and they are to be compensated later in a proper way, as explained in section 6.5.3.

The tuning procedure for this controller with rotor flux is more difficult than for simple MIMO current controller and it will deliver much higher number of regions to get a similar performance.

In this formulation the voltage drops $\begin{bmatrix} 0 & -l_\sigma & 0 \\ l_\sigma & 0 & k_r \\ 0 & 0 & 0 \end{bmatrix} . \begin{bmatrix} \omega_s \, i_{sd} \\ \omega_s \, i_{sq} \\ \omega_m \, \psi_{rd} \end{bmatrix}$ on the control inputs are to be estimated/ calculated as already introduced using two speed filters, with expectation of small offset due to using the filtered speed signal in the compensation procedure. Otherwise, a complex estimation procedure as introduced in [99] should be used for these formulations. Furthermore, as the rotor flux is kept here constant at its nominal value (because of the used FOC control structure), using this formulation for working in non-field weakening area of the induction machine will not be a wise decision.

6.5.3 External Compensation Procedure

Here the simplified machine model is only considered in formulating the MPC optimization problem and the nonlinear terms in (6.18), (6.20), (6.22) are neglected in this formulation but compensated later in a proper way, passively or actively. These terms are time varying terms with much slower dynamic than the stator currents (rotor time constant $\tau_r \approx 60$ [pu] and mechanical time constant $\tau_m = 500$ [pu] comparing with the currents time constant $\tau_\sigma' \approx 2.5$ [pu]); hence they could be considered as input-disturbances and compensated outside the controller. The MPT toolbox in [77] introduces a method to deal with the additive input-disturbances and considers these terms in the design phase of the MPC controller. Here the constraints on these terms are to be provided as a bounded polytope to the MPT toolbox. The dimension of this polytope could be determined by observing these terms with a PI controlled system, Figure 6.3. Unfortunately, following the additive input-disturbance method using the MPT toolbox, the resulting feasible stat-space will be shrunk too much, in the same rate of the polytope dimensions. Having a feasible state space smaller than the state variable limitations will cause an infeasibility problem when these variables exceed the feasible space. Alternatively, the MPC controller can be designed considering only the simplified machine model, and these nonlinear terms are estimated and compensated later in the implementation phase outside the controller as voltage compensation terms. As the compensated terms represent voltage drops in the system inputs, the control inputs are corrected outside the controller structure: either before their feeding back to the controller (Active SMPC in Figure 6.17) or before their applying to the system as actuated variables (Passive MPC in Figure 6.18). Active SMPC controller responses actively to changes in the applied voltage and generates more voltage to compensate the voltage drop, accordingly, Figure 6.19. Having an internal limitation on the system variables, the active SMPC controller does not violate the input constraints.

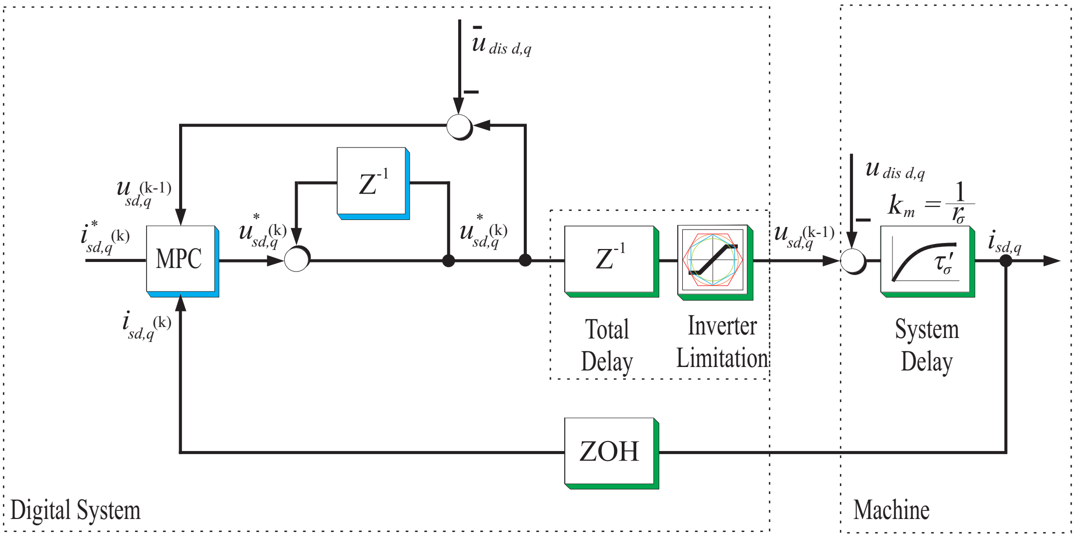


Figure 6.17: SMPC current control structure with an input correction

In the second case, the SMPC controller reacts passively to the later compensated voltage, similar to FFC as shown in Figure 6.20. Actually, there is no significant difference in the implementation results in transient and steady state between both compensation methods. Both exhibit a fast dynamic without overshoot and zero steady-state tracking error.

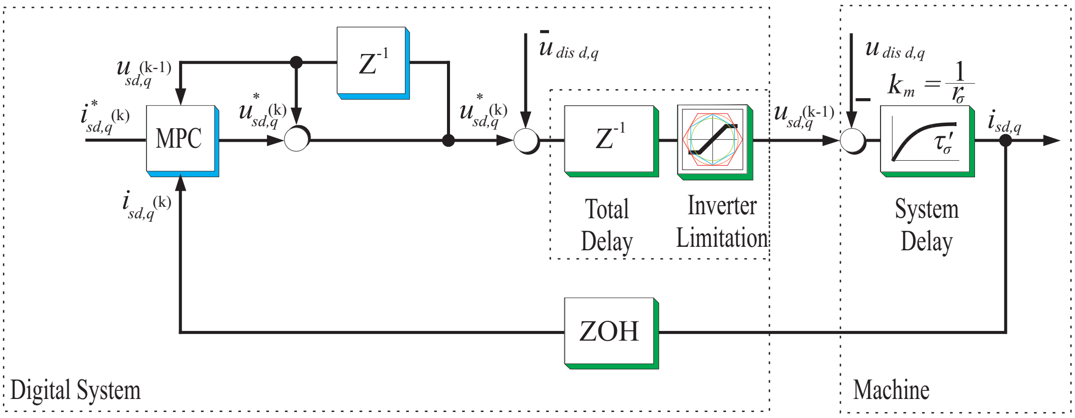


Figure 6.18: SMPC current control structure with feed-forward compensation

The following experimental results present the dynamic behavior of the SMPC current controller (active and passive SMPC) under similar conditions for an induction machine controlled using a 3-level NPC inverter. To observe the controller dynamic it is enough here to apply a step in the torque producing current (i_{sq} = 0→0.4), while the flux producing current was kept at its nominal value (i_{sd} = 0.35) and for a small time scale (time division is 1 msec). For better presentation the current signal i_{sd} was dragged in the figures to zero.

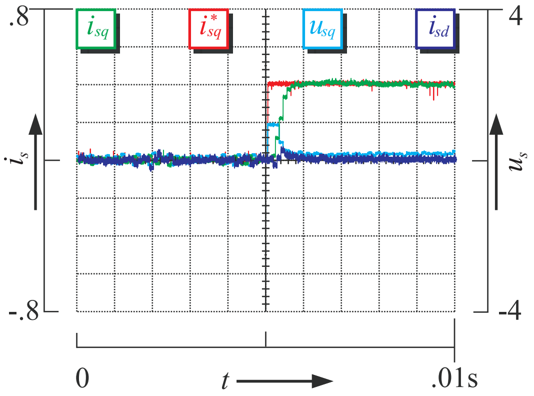


Figure 6.19: IM current response using an active SMPC$_{dq}$ controller

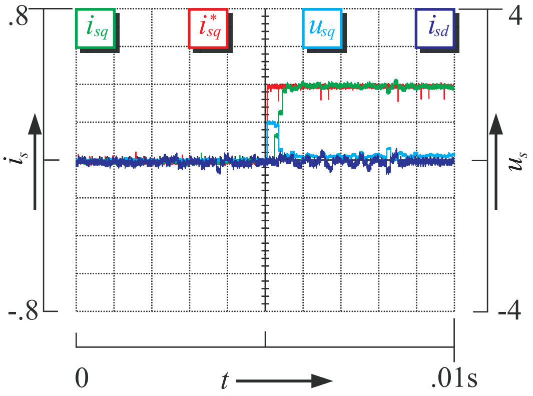


Figure 6.20: IM current response using a passive SMPC$_{dq}$ controller

6.5.4 MIMO-MPC Current Controller with Internal Compensation

In this paragraph, the nonlinear term is considered as it is (without any kind of linearization) and it is introduced to the system model in the same way like introducing the reference signal for tracking problems in state space representation. Assuming that this term will not change within the sampling period; it is calculated at each sampling instance and introduced to controller as a new state variable. Referring to equation (6.2) and to the standard way of considering the tracking problem for a time-varying reference, where a new state variable x_k^* is added for the reference signal as follows: $x_{k+1}^* = x_k^*$ or $\frac{dx^*(t)}{dt} = 0$. This formulation means that the reference signal does not change within the sampling period, which is accepted assumption in steady state and delivers very good implementation results. In the same way, terms that have to be compensated could be integrated inside the controller as a new input-disturbance terms $\boldsymbol{u}_{dist}$ added to the state variables. Whatever the new added state-vector $\mathbf{x}^{d} = -\,(1/l_{\sigma}).\boldsymbol{u}_{i}$ (Input-disturbance vector $\boldsymbol{u}_{dist} = \boldsymbol{u}_{i}$) is, the optimization problem (5.7) has to be reformulated again to consider the new modifications. As a way to compensate the input-disturbance terms in steady state by augmenting the state-vector in the system dynamic, the multi-parametric MPC optimization problem has to be reformulated again accordingly.

For sake of simplicity, only the new constraints and the modified equations of are repeated here:

$$\begin{aligned} &\mathbf{x}^{d} \in \boldsymbol{X}^{d} \subset \mathbb{R}^{d} \\ &\boldsymbol{X}^{d} = \left\{\mathbf{x}^{d} | \mathbf{x}_{t}^{d} \in \boldsymbol{X}^{d}, t \geq 0\right\} \end{aligned} \tag{6.24}$$

where $\mathbf{x}^{d}$ is the newly added input-disturbance state vector, and the $\boldsymbol{X}^{d}$ is bounding constraints on $\mathbf{x}^{d}$.

$$\boldsymbol{J}(\mathbf{U}, \mathbf{x}_{t}, \mathbf{x}_{t}^{d}) = \min_{\mathbf{U}} \left(\begin{array}{l} \mathbf{x}_{t+N_p|t}^{T} \cdot \mathbf{P}_{N} \cdot \mathbf{x}_{t+N_p|t} + \mathbf{x}_{t+N_p|t}^{d\,T} \cdot \mathbf{P}_{d} \cdot \mathbf{x}_{t+N_p|t}^{d} + \\ \sum\limits_{k=0}^{N_p-1} \mathbf{x}_{t+k|t}^{T} \cdot \mathbf{Q} .\ \mathbf{x}_{t+k|t} + \mathbf{x}_{t+k|t}^{d\,T} \cdot \mathbf{Q}_{d} \cdot \mathbf{x}_{t+k|t}^{d} + \mathbf{u}_{t+k|t}^{T} \cdot \mathbf{R} .\ \mathbf{u}_{t+k|t} \end{array} \right) \tag{6.25}$$

$$\mathbf{Q}_{d} = \mathbf{Q}^{T} \geq 0, \mathbf{P}_{d} \geq 0 \tag{6.26}$$

where the matrices $\mathbf{Q}_{d}$, $\mathbf{P}_{d}$ are the associated weighting factors with the new state vector $\mathbf{x}^{d}$.

By treating the new augmented state-vector as a free parameter, the resulting MPC can be recast as MPP-Quadratic optimization problem (mp-QP). The resulting solution will be piece wise affine control functions of the augmented state-vector with their corresponding regions, where these control laws are valid. Hence, the resulting feasible regions will be with (n + d) dimensions and the resulting control law function will take the following formulation:

$$\mathbf{u}_{t} = \mathbf{u}_{t}^{*}(x_{t}, x_{t}^{d}) = \mathbf{F}^{i} .\ \mathbf{x}_{t} + \mathbf{F}_{d}^{i} .\ \mathbf{x}_{t}^{d} + \mathbf{G}^{i} \tag{6.27}$$

where the subscript (i) refers to region index, where this control law is valid, and the matrices $\mathbf{F}^{i}$, $\mathbf{F}_{d}^{i}$, $\mathbf{G}^{i}$ are the control law coefficients.

Extending the MPC formulation to consider the delay compensation and free-offset tracking, equation (6.11), the final control law will take the following from:

$$\begin{aligned} &\Delta\mathbf{u}_{k} = \Delta\mathbf{u}_{k}^{*}(\mathbf{x}_{k}, \mathbf{x}_{k}^{d}, \mathbf{u}_{k-1}, \mathbf{x}_{k}^{*}) = \mathbf{F}^{i} .\ \mathbf{x}_{k} + \mathbf{F}_{d}^{i} .\ \mathbf{x}_{k}^{d} + \mathbf{F}_{l}^{i} .\ \mathbf{u}_{k-1} + \mathbf{F}_{r}^{i} .\ \mathbf{x}_{k}^{*} + \mathbf{G}^{i} \\ &\mathbf{u}_{k} = \mathbf{u}_{k-1} + \Delta\mathbf{u}_{k} \end{aligned} \tag{6.28}$$

Similar formulation can be concluded by considering the error integral formulation (6.4) instead of the du-formulation (6.3).

$$\mathbf{u}_{k} = \mathbf{u}_{k}^{*}(\mathbf{x}_{k}, \mathbf{x}_{k}^{d}, \mathbf{e}_{Ik}, \mathbf{x}_{k}^{*}) = \mathbf{F}^{i} .\ \mathbf{x}_{k} + \mathbf{F}_{d}^{i} .\ \mathbf{x}_{k}^{d} + \mathbf{F}_{e}^{i} .\ \mathbf{e}_{Ik} + \mathbf{F}_{r}^{i} .\ \mathbf{x}_{k}^{*} + \mathbf{G}^{i} \tag{6.29}$$

where $\mathbf{e}_{Ik}$ is the state-error integral at time step k, and $\mathbf{F}_{e}^{i}$ is the associated control law coefficient.

Here appears the formulation similarity between the classical linear quadratic regulator (LQR) [97] for unconstrained optimization problems and the MPC controller for constrained optimization problems, in the way of dealing with free-offset tracking and disturbance rejection. To achieve a free-offset tracking in steady state with an input-disturbance rejection, the LQR control structure in Figure 6.21, has to be reconstructed to consider both the state reference and the estimated input-disturbance, Figure 6.22.

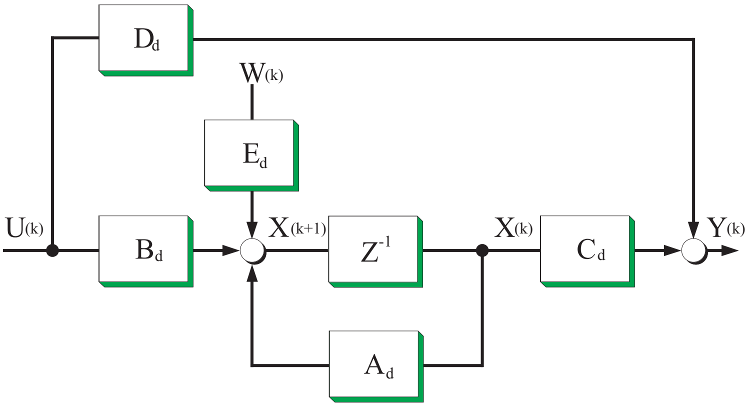


Figure 6.21: Block diagram of LQR control structure

The matrices ($\mathbf{A}_d$, $\mathbf{B}_d$, $\mathbf{C}_d$, $\mathbf{D}_d$, $\mathbf{E}_d$) are the discrete forms of the system matrices.

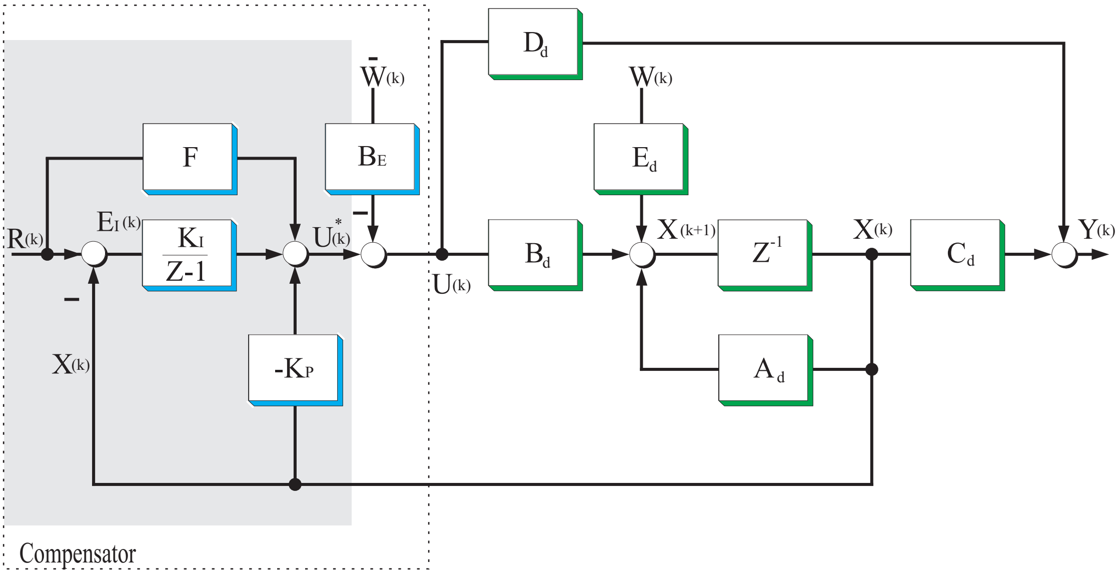


Figure 6.22: Modified block diagram of LQR control structure for tracking and input disturbance rejection

According to the introduced structure in Figure 6.22, the resulting control law for LQR can be stated as follows:

$$\begin{aligned} &\mathbf{u} = -\mathbf{K}_p \,.\, \mathbf{x} + \mathbf{B}_d \,.\, \overline{\mathbf{w}} + \mathbf{K}_I \,.\, \mathbf{e}_I + \mathbf{F} \,.\, \mathbf{x}^* \\ &\mathbf{e}_I = \int (\mathbf{x}^* - \mathbf{x}) \end{aligned} \tag{6.30}$$

Note that the control input is a function of the state, estimated input disturbance, error integral, and the reference state variable; which is the case by considering the MPC formulation with state-error integral, equation (6.29). For further information about the evaluation of the compensator parameters ($\mathbf{K}_{\mathrm{I}}$, $\mathbf{K}_{\mathrm{P}}$, $\mathbf{B}_{\mathrm{E}}$, $\mathbf{F}$) in Figure 6.22, the author refers to [97].

A. Introducing the d-axis Rotor Flux as a New State Variable

Using the equation (6.22) and considering the flux as a one of the state variables, the disturbance term will take the form:

$$\mathbf{x}^{\mathrm{d}} = -\frac{1}{l_\sigma}\boldsymbol{u}_i \Rightarrow \mathbf{x}^{\mathrm{d}} = \begin{bmatrix} x_d^d \\ x_q^d \\ x_\psi^d \end{bmatrix} = -\frac{1}{l_\sigma}\begin{bmatrix} 0 & -l_\sigma & 0 \\ l_\sigma & 0 & k_r \\ 0 & 0 & 0 \end{bmatrix} \cdot \begin{bmatrix} \omega_s\, i_{sd} \\ \omega_s\, i_{sq} \\ \omega_m\, \psi_{rd} \end{bmatrix} \tag{6.31}$$

This vector $\mathbf{x}^{\mathrm{d}}$, as the case for the time-varying reference tracking, has to be estimated/ calculated outside the controller and instead of compensating it outside the controller structure in an active or a passive way (as explained in section 6.5.3) the new formulated vector $\mathbf{x}^{\mathrm{d}}$ will be fed here to the MPC controller as a new state variable. As the term $x_\psi^d = 0$, the respective row could be deleted from compensation matrix.

Thus, the new extended system model will take this form:

$$\frac{d}{d\tau}\begin{bmatrix} i_{sd} \\ i_{sq} \\ \psi_{rd} \\ x_d^d \\ x_q^d \end{bmatrix} = \begin{bmatrix} -\frac{1}{\tau_\sigma^{'}} & 0 & \frac{k_r}{l_\sigma \tau_r} & 1 & 0 \\ 0 & -\frac{1}{\tau_\sigma^{'}} & 0 & 0 & 1 \\ \frac{l_m}{\tau_r} & 0 & -\frac{1}{\tau_r} & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix} \cdot \begin{bmatrix} i_{sd} \\ i_{sq} \\ \psi_{rd} \\ x_d^d \\ x_q^d \end{bmatrix} + \begin{bmatrix} \frac{1}{l_\sigma} & 0 \\ 0 & \frac{1}{l_\sigma} \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \end{bmatrix} \cdot \begin{bmatrix} u_{sd} \\ u_{sq} \end{bmatrix} \tag{6.32}$$

And the output equation:

$$\begin{bmatrix} i_{sd} \\ i_{sq} \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \end{bmatrix} \cdot \begin{bmatrix} i_{sd} \\ i_{sq} \\ \psi_{rd} \\ x_d^d \\ x_q^d \end{bmatrix} + \begin{bmatrix} 0 & 0 \\ 0 & 0 \end{bmatrix} \cdot \begin{bmatrix} u_{sd} \\ u_{sq} \end{bmatrix} \tag{6.33}$$

For an easy adjustment of the constraints on the disturbance-state-variables, it would be better to keep their multiplications with their coefficients inside the controller structure and consider the following formulation:

$$\mathbf{x}^{\mathrm{d}} = \begin{bmatrix} x_d^d \\ x_q^d \\ x_\psi^d \end{bmatrix} = -\begin{bmatrix} \omega_s\, i_{sd} \\ \omega_s\, i_{sq} \\ \omega_m\, \psi_{rd} \end{bmatrix} \tag{6.34}$$

$$\frac{d}{d\tau}\begin{bmatrix} i_{sd} \\ i_{sq} \\ \psi_{rd} \\ x_d^d \\ x_q^d \end{bmatrix} = \begin{bmatrix} -\frac{1}{\tau_\sigma^{'}} & 0 & \frac{k_r}{l_\sigma \tau_r} & 0 & 1 & 0 \\ 0 & -\frac{1}{\tau_\sigma^{'}} & 0 & -1 & 0 & -\frac{k_r}{l_\sigma} \\ \frac{l_m}{\tau_r} & 0 & -\frac{1}{\tau_r} & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \cdot \begin{bmatrix} i_{sd} \\ i_{sq} \\ \psi_{rd} \\ x_d^d \\ x_q^d \end{bmatrix} + \begin{bmatrix} \frac{1}{l_\sigma} & 0 \\ 0 & \frac{1}{l_\sigma} \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \end{bmatrix} \cdot \begin{bmatrix} u_{sd} \\ u_{sq} \end{bmatrix} \tag{6.35}$$

The discrete form of the extended-model MPC formulation (EMPC) will be:

$$\begin{bmatrix} i_{sd} \\ i_{sq} \\ \psi_{rd} \\ x_d^d \\ x_q^d \end{bmatrix}_{k+1} = \begin{bmatrix} 1-\frac{T_0}{\tau_\sigma^{'}} & 0 & \frac{T_0 k_r}{l_\sigma \tau_r} & 0 & T_0 & 0 \\ 0 & 1-\frac{T_0}{\tau_\sigma^{'}} & 0 & -T_0 & 0 & -\frac{T_0 k_r}{l_\sigma} \\ \frac{T_0 l_m}{\tau_r} & 0 & 1-\frac{T_0}{\tau_r} & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} i_{sd} \\ i_{sq} \\ \psi_{rd} \\ x_d^d \\ x_q^d \end{bmatrix}_k + \begin{bmatrix} \frac{T_0}{l_\sigma} & 0 \\ 0 & \frac{T_0}{l_\sigma} \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \end{bmatrix} \cdot \begin{bmatrix} u_{sd} \\ u_{sq} \end{bmatrix}_k \tag{6.36}$$

Following the first estimation procedure with two speed filters to estimate/ calculate and compensate the disturbance terms ($\mathbf{x}^d$) can be done here with expectation of small tracking offset.

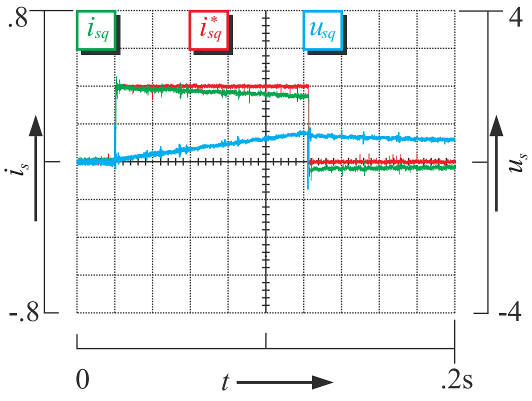


Figure 6.23: IM current response using an EMPC$_{dq}$ controller with internal compensation ($\tau_{f\_\delta} = 1$, $\tau_{f\_e} = 3$)

B. EMPC with a Perfect Internal Compensation

Perfect and easy compensation of the induced voltages without using the measured speed or the complex estimation procedure proposed in [99] can be done by considering the equivalent circuit of the induction machine in Figure 6.16, and reformulating the machine equations as follows:

$$\frac{d}{d\tau}\begin{bmatrix} i_{sd} \\ i_{sq} \end{bmatrix} = \begin{bmatrix} -\frac{1}{\tau_\sigma^{'}} & 0 \\ 0 & -\frac{1}{\tau_\sigma^{'}} \end{bmatrix} \cdot \begin{bmatrix} i_{sd} \\ i_{sq} \end{bmatrix} + \begin{bmatrix} \frac{1}{l_\sigma} & 0 \\ 0 & \frac{1}{l_\sigma} \end{bmatrix} \cdot \left(\begin{bmatrix} u_{sd} \\ u_{sq} \end{bmatrix} - \begin{bmatrix} u_{id} \\ u_{iq} \end{bmatrix} \right) \tag{6.37}$$

where $$\begin{bmatrix} u_{id} \\ u_{iq} \end{bmatrix} = \begin{bmatrix} -l_\sigma \omega_s i_{sq} - \frac{k_r}{\tau_r}\psi_{rd} \\ l_\sigma \omega_s i_{sd} + k_r \omega_m \psi_{rd} \end{bmatrix} \tag{6.38}$$

The output equation is:

$$\begin{bmatrix} i_{sd} \\ i_{sq} \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} i_{sd} \\ i_{sq} \end{bmatrix} + \begin{bmatrix} 0 & 0 \\ 0 & 0 \end{bmatrix} \cdot \begin{bmatrix} u_{sd} \\ u_{sq} \end{bmatrix} \tag{6.39}$$

Following the state augmentation procedure under the assumptions mentioned above, the estimated induced-voltages using the equation (6.17) can be used here directly $\mathbf{x}^{\mathrm{d}} = -\frac{1}{l_\sigma}\boldsymbol{u}_i$:

$$\frac{d}{d\tau}\begin{bmatrix} i_{sd} \\ i_{sq} \\ x_d^d \\ x_q^d \end{bmatrix} = \begin{bmatrix} -\frac{1}{\tau_\sigma^{'}} & 0 & 1 & 0 \\ 0 & -\frac{1}{\tau_\sigma^{'}} & 0 & 1 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} \cdot \begin{bmatrix} i_{sd} \\ i_{sq} \\ x_d^d \\ x_q^d \end{bmatrix} + \begin{bmatrix} \frac{1}{l_\sigma} & 0 \\ 0 & \frac{1}{l_\sigma} \\ 0 & 0 \\ 0 & 0 \end{bmatrix} \cdot \begin{bmatrix} u_{sd} \\ u_{sq} \end{bmatrix} \tag{6.40}$$

$$: \mathbf{x}^{\mathrm{d}} = -\frac{1}{l_\sigma}\boldsymbol{u}_i$$

The output equation is:

$$\begin{bmatrix} i_{sd} \\ i_{sq} \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix} \cdot \begin{bmatrix} i_{sd} \\ i_{sq} \\ x_d^d \\ x_q^d \end{bmatrix} + \begin{bmatrix} 0 & 0 \\ 0 & 0 \end{bmatrix} \cdot \begin{bmatrix} u_{sd} \\ u_{sq} \end{bmatrix} \tag{6.41}$$

Using forward Euler discretization introduced in section 5.3, the discrete form of the extended machine model will be:

$$\begin{bmatrix} i_{sd} \\ i_{sq} \\ x_d^d \\ x_q^d \end{bmatrix}_{k+1} = \begin{bmatrix} 1-\frac{T_0}{\tau_\sigma^{'}} & 0 & T_0 & 0 \\ 0 & 1-\frac{T_0}{\tau_\sigma^{'}} & 0 & T_0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} i_{sd} \\ i_{sq} \\ x_d^d \\ x_q^d \end{bmatrix}_k + \begin{bmatrix} \frac{T_0}{l_\sigma} & 0 \\ 0 & \frac{T_0}{l_\sigma} \\ 0 & 0 \\ 0 & 0 \end{bmatrix} \cdot \begin{bmatrix} u_{sd} \\ u_{sq} \end{bmatrix}_k \qquad (6.42)$$

The output equation of the discrete model is:

$$\begin{bmatrix} i_{sd} \\ i_{sq} \end{bmatrix}_k = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix} \cdot \begin{bmatrix} i_{sd} \\ i_{sq} \\ x_d^d \\ x_q^d \end{bmatrix}_k + \begin{bmatrix} 0 & 0 \\ 0 & 0 \end{bmatrix} \cdot \begin{bmatrix} u_{sd} \\ u_{sq} \end{bmatrix}_k \qquad (6.43)$$

where T_0 is the normalized sampling time.

This formulation introduces the extended machine model for an input-disturbance rejection without refereeing to the tracking or time delay compensation. For zero-offset tracking and time delay compensation, this equation has to be further reformulated according to the equations (6.5) in section 6.3. The proposed EMPC controller was evaluated on an induction machine controlled through three-level neutral-point diode-clamped inverter (NPC) in field coordinates. In case of existing a steady state error in the machine currents represented in field coordinates, this error will be reflected in an amplitude difference between the currents and their references in stator coordinates much clear. Figure 6.24 shows the implementation results of the induction machine currents and their references in stator coordinates using the EMPC controller at 50% of the nominal speed. The figure shows a very good tracking behavior of the EMPC current controller when the machine revolves.

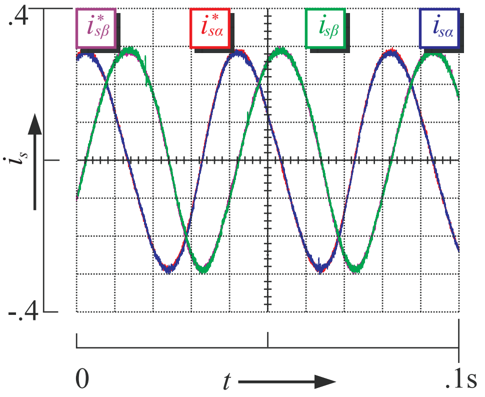


Figure 6.24: IM currents and their references in stationary coordinates controlled using an EMPC$_{dq}$ controller at 50% nominal speed

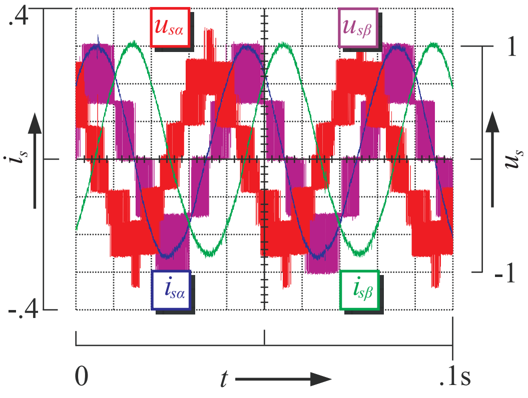


Figure 6.25: IM currents and voltages in stationary coordinates controlled using an EMPC$_{dq}$ controller at 50% nominal speed

Figure 6.25 shows both the machine currents and voltages in stator coordinates at 50% nominal speed using a three-level NPC inverter.

6.6 Indirect Exp-MPC Current Controller for an Induction Machine in Stationary Coordinates

In comparison with the conventional PI controller the MPC controller has the capability to control the systems with sinusoidal inputs without impact on the performance. Actually, one of the reasons to control the machine currents in synchronous reference frame comes from the steady state error associated with PI controller by controlling such systems in stationary reference frame with AC components. However, controlling the machine quantities in stationary reference frame simplifies the control structure and removes the need for the coordinate transformation [57], [109]. Furthermore the synchronous controller requires an explicit knowledge of the synchronous frequency ω_s. Controlling the machine currents in stationary reference frame was introduced in [60], [61] using a PR regulator (Proportional and Resonant regulator) with a zero-steady state error; however the control design is more complicated than that of a PI controller in synchronous coordinates. In the following scenario and just for simplicity, the field-oriented control structure with PI speed controller in synchronous coordinates is used. The machine currents are controlled using MIMO-MPC controller in the stationary coordinates, Figure 6.26.

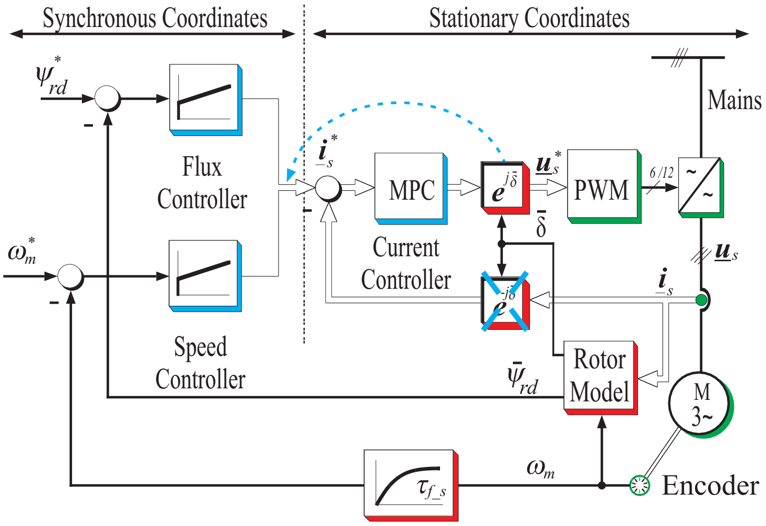


Figure 6.26: The proposed MPC current control structure in stationary coordinates

Controlling the machine currents in the stationary coordinates using the MPC controller is very similar to the case in synchronous coordinates except the way of dealing with the compensation terms and the system constraints. Following this discussion, both the system constraints and the compensation terms are to be transformed using the estimated transformation angle $\bar{\delta}$ into the stationary coordinates. Here the control inputs (machine voltages $u_{s\alpha}$, $u_{s\beta}$) must remain inside the outer hexagon (three-level inverter voltage-limitation, Figure 6.28) and the state variables (machine currents $i_{s\alpha}$, $i_{s\beta}$) must stay inside the maximum circular limitation of the machine currents, equation (6.44).

$$i_{s\,max} = \sqrt{\left(i_{s\alpha}\right)^2 + \left(i_{s\beta}\right)^2} \tag{6.44}$$

The current limitations should be introduced to the optimization problem as linear constraints on the state variables. However, these constraints could be provided as the largest area, may be defined as linear inequalities (rectangular (1), hexagon (2), or even octagon (3)) inside the maximum circular limitation, Figure 6.27.

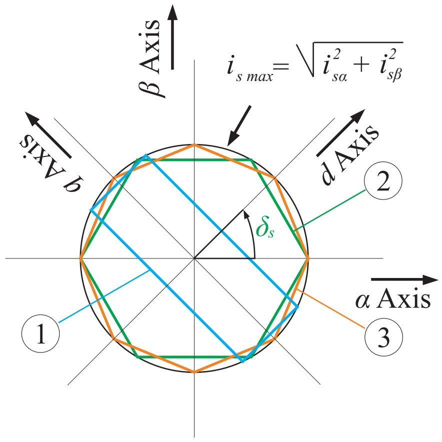


Figure 6.27: Current limitations of an AC Machine

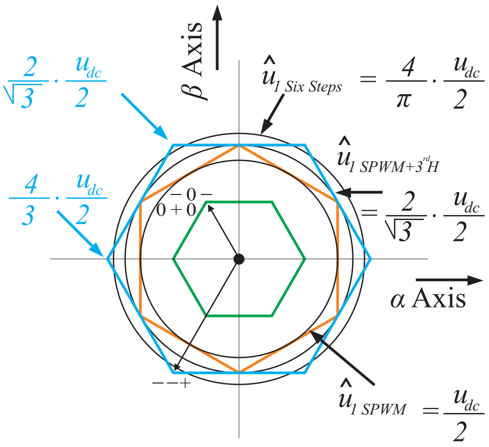

$\frac{2}{\sqrt{3}} \cdot \frac{u_{dc}}{2}$
β Axis
$\hat{u}_{1\ Six\ Steps} = \frac{4}{\pi} \cdot \frac{u_{dc}}{2}$
$\hat{u}_{1\ SPWM+3^{rd}H}$
$= \frac{2}{\sqrt{3}} \cdot \frac{u_{dc}}{2}$
$\frac{4}{3} \cdot \frac{u_{dc}}{2}$
− 0 −
0 + 0
α Axis
− − +
$\hat{u}_{1\ SPWM} = \frac{u_{dc}}{2}$

Figure 6.28: Voltage limitations of a three-level inverter

Figure 6.28 shows the voltage limitations of a three-level NPC inverter using different modulation strategies [39]. The notation $\hat{u}_1$ refers to peak value of the resulting voltage fundamental signal at the output of the NPC inverter using the associated modulation strategy.

A. SMPC Controller in Stationary Reference Frame

Referring to the induction machine equations in general k-coordinates, equation (2.10), and by substituting $\omega_k = 0$, the new system model in stationary reference frame[8] can be concluded, equation (6.45).

$$\boldsymbol{i}_s^{(S)} + \tau_\sigma' \frac{d\boldsymbol{i}_s^{(S)}}{d\tau} = \frac{1}{r_\sigma}\left(\boldsymbol{u}_s^{(S)} - \overbrace{\underline{\left(-k_r \left(\frac{1}{\tau_r} - j\omega_m \right) \boldsymbol{\psi}_r \right)}}^{\boldsymbol{u}_i^{(S)}} \right) \tag{6.45}$$

where $\tau_\sigma' = \sigma \tau_s$, $\sigma = 1 - \frac{l_m^2}{l_s l_r}$, $l_\sigma = \sigma l_s$, $\tau_s = \frac{l_s}{r_s}$, $\tau_r = \frac{l_r}{r_r}$, $k_r = \frac{l_m}{l_r}$, $r_\sigma = r_s + r_r k_r^2$.

Actually, the current cross-coupling effect appears as a result of representing the machine model in synchronous reference frame. Therefore, this coupling effect does not exist any more by considering the stationary reference frame, equation (6.45). The simplified machine model, equation (6.46), used to design the SMPC controller in both synchronous and stationary reference frame has an identical formulation, but with different notations. The nonlinear terms ($\boldsymbol{u}_i$) are to be compensated properly in a later step.

[8] The superscript (S) refers to the *αβ-reference frame*, and it is omitted in next equations for simplicity.

$$\begin{aligned}&\boldsymbol{i}_s + \tau_\sigma^{'}\frac{d\boldsymbol{i}_s}{d\tau} = \frac{1}{r_\sigma}\left(\boldsymbol{u}_s - \boldsymbol{u}_i\right)\\ &:\boldsymbol{u}_i = \left(\underline{j\omega_k l_\sigma \boldsymbol{i}_s - k_r\left(\frac{1}{\tau_r} - j\omega_m\right)\boldsymbol{\psi}_r}\right), \omega_k = \begin{cases}0 & \text{,in } \alpha\beta - \text{reference frame}\\ \omega_s & \text{,in } dq - \text{reference frame}\end{cases}\end{aligned} \tag{6.46}$$

Formulating the new optimization problem requires first introducing the machine model in state space representation as follows:

$$\frac{d}{d\tau}\begin{bmatrix} i_{s\alpha}\\ i_{s\beta}\end{bmatrix} = \begin{bmatrix} -\frac{1}{\tau_\sigma^{'}} & 0\\ 0 & -\frac{1}{\tau_\sigma^{'}}\end{bmatrix}\cdot\begin{bmatrix} i_{s\alpha}\\ i_{s\beta}\end{bmatrix} + \begin{bmatrix} \frac{1}{l_\sigma} & 0\\ 0 & \frac{1}{l_\sigma}\end{bmatrix}\cdot\begin{bmatrix} u_{s\alpha}\\ u_{s\beta}\end{bmatrix} \tag{6.47}$$

The output equation is:

$$\begin{bmatrix} i_{s\alpha}\\ i_{s\beta}\end{bmatrix} = \begin{bmatrix} 1 & 0\\ 0 & 1\end{bmatrix}\cdot\begin{bmatrix} i_{s\alpha}\\ i_{s\beta}\end{bmatrix} + \begin{bmatrix} 0 & 0\\ 0 & 0\end{bmatrix}\cdot\begin{bmatrix} u_{s\alpha}\\ u_{s\beta}\end{bmatrix} \tag{6.48}$$

In analogue discussion of compensating the voltage-drops in *dq-reference frame*, the induced-voltage (**$\boldsymbol{u}_i$**) in *αβ-reference frame* can be compensated either outside the SMPC controller structure (Active or Passive SMPC) or inside the controller structure (EMPC) by extending the state variables of the MPC controller to include these voltage-drops as new state variables. In both cases, the induced voltage, equation (6.49), has to be estimated/ calculated at each sampling time and compensated properly.

$$\boldsymbol{u}_i = \begin{bmatrix} u_{i\alpha}\\ u_{i\beta}\end{bmatrix} = -k_r\left(-\frac{1}{\tau_r} - \omega_m\begin{bmatrix} 0 & 1\\ -1 & 0\end{bmatrix}\right)\cdot\begin{bmatrix} \psi_{r\alpha}\\ \psi_{r\beta}\end{bmatrix} \tag{6.49}$$

For the similarity to the synchronous coordinates, more details about the formulation of SMPC and EMPC controllers in *αβ-reference frame* are omitted in this section and just some experimental results are presented here.

The results introduced below were obtained for SMPC controller working in stationary coordinates with a passive compensation of the induced voltages. Figure 6.29 shows a fast dynamic response by applying a step in the torque producing current component ($i_{sq} = 0 \rightarrow 0.4$) while keeping the machine fully magnetized ($i_{sd} = 0.35$). To evaluate the efficiency of the compensation procedure of the rotor-induced voltage, the same test is repeated in Figure 6.30 with a large time scale, where the machine is started to accelerate very fast. As the controller is working in stationary coordinates it was wise to apply a step ($\left|i_{s\alpha,\beta}\right| = 0.6 \rightarrow 0.2$) to the machine currents ($i_{s\alpha}$, $i_{s\beta}$) represented in stationary reference frame and observe the tracking procedure, Figure 6.31.

Finally, Figure 6.32 shows both machine currents and voltages in *αβ-reference frame* at 25% of the nominal speed using a three-level NPC inverter for steady state operation of the induction machine.

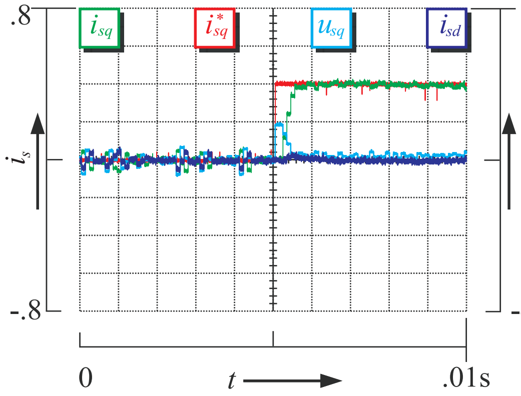


Figure 6.29: IM current response in synchronous coordinates using a passive SMPC$_{\alpha\beta}$ controller

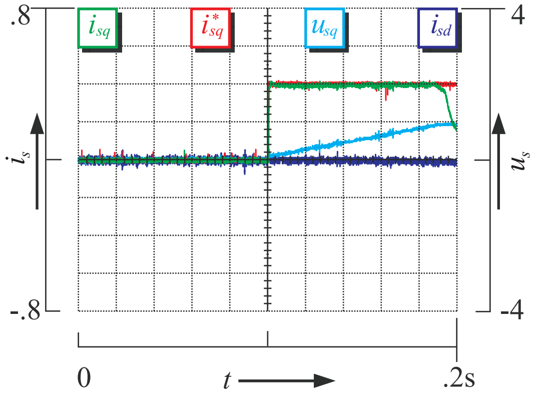


Figure 6.30: IM current response in synchronous coordinates using a passive SMPC$_{\alpha\beta}$ controller, with a large time scale

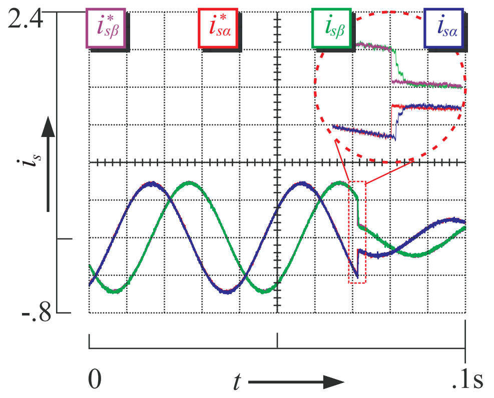


Figure 6.31: IM current responses in stationary coordinates using a passive SMPC$_{\alpha\beta}$ controller; steps in ($i_{s\alpha}$, $i_{s\beta}$) currents

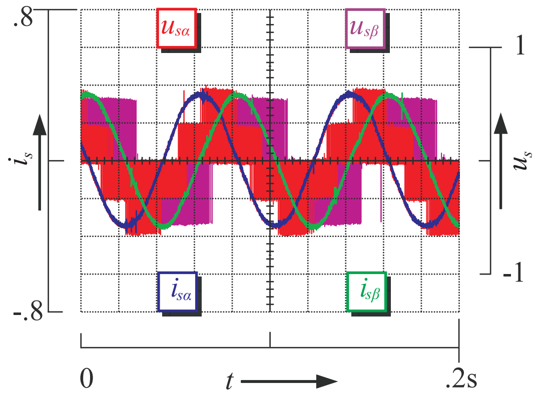


Figure 6.32: IM currents and voltages in stationary coordinates controlled using a passive SMPC$_{\alpha\beta}$ controller; normal operation at 25% nominal speed

B. EMPC Controller in Stationary Reference Frame

Following the state augmentation procedure, the estimated induced voltage in the stationary reference frame, equation (6.49), can be used here directly $\mathbf{x}^{\mathrm{d}} = \text{-}(1/l_\sigma).\boldsymbol{u}_i^{(S)}$ and compensated inside the controller structure as follows:

$$\frac{d}{d\tau}\begin{bmatrix} i_{s\alpha} \\ i_{s\beta} \\ x^d_\alpha \\ x^d_\beta \end{bmatrix} = \begin{bmatrix} -\frac{1}{\tau^{'}_\sigma} & 0 & 1 & 0 \\ 0 & -\frac{1}{\tau^{'}_\sigma} & 0 & 1 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} . \begin{bmatrix} i_{s\alpha} \\ i_{s\beta} \\ x^d_\alpha \\ x^d_\beta \end{bmatrix} + \begin{bmatrix} \frac{1}{l_\sigma} & 0 \\ 0 & \frac{1}{l_\sigma} \\ 0 & 0 \\ 0 & 0 \end{bmatrix} . \begin{bmatrix} u_{s\alpha} \\ u_{s\beta} \end{bmatrix} \tag{6.50}$$

$$: \mathbf{x}^{\mathrm{d}} = -\frac{1}{l_\sigma}\boldsymbol{u}_i^{(S)}$$

The output equation is:

$$\begin{bmatrix} i_{s\alpha} \\ i_{s\beta} \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix} \cdot \begin{bmatrix} i_{s\alpha} \\ i_{s\beta} \\ x_{\alpha}^{d} \\ x_{\beta}^{d} \end{bmatrix} + \begin{bmatrix} 0 & 0 \\ 0 & 0 \end{bmatrix} \cdot \begin{bmatrix} u_{s\alpha} \\ u_{s\beta} \end{bmatrix} \tag{6.51}$$

Similar discussion to the system model in synchronous coordinates can be applied here to adjust the system constraints and introduce the discrete-time LTI/ PWA model to the MPT toolbox. Some implementation results were collected for the EMPC$_{\alpha\beta}$ controller in synchronous and stationary coordinates as follows. Figure 6.33 shows controller dynamic in transient by applying a step in ($i_{sq} = 0 \rightarrow 0.4$) current component while keeping the flux producing current component constant. The efficiency of the internally compensation procedure in stationary reference frame of the rotor-induced voltage, can be evaluated by repeating the last test but for a large time scale, Figure 6.34, where the machine is started to accelerate very fast.

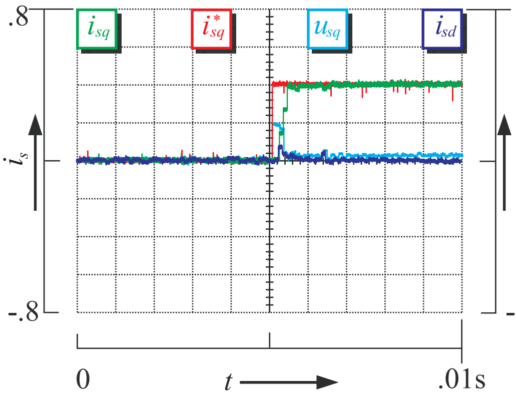


Figure 6.33: IM current response in synchronous coordinates using an EMPC$_{\alpha\beta}$ controller

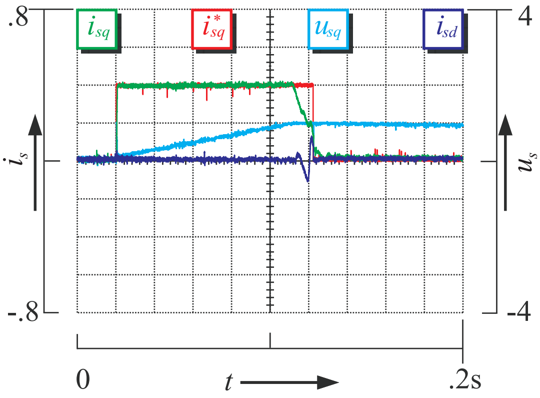


Figure 6.34: IM current response in synchronous coordinates using an EMPC$_{\alpha\beta}$ controller, with a large time scale

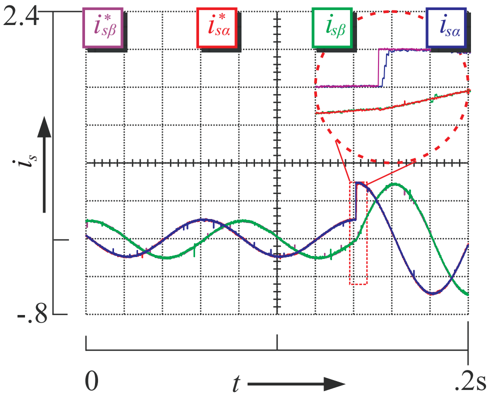


Figure 6.35: IM current responses in stationary coordinates using an EMPC$_{\alpha\beta}$ controller; steps in ($i_{s\alpha}$, $i_{s\beta}$) currents

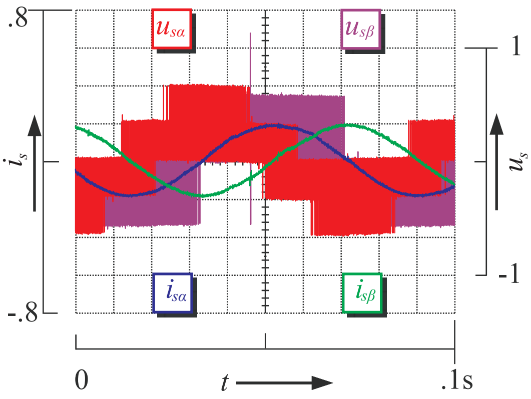


Figure 6.36: IM current and voltages in stationary coordinates controlled using an EMPC$_{\alpha\beta}$ controller; normal operation at 25% nominal speed

As the controller is working in stationary coordinates it was wise to apply a step ($|i_{s\alpha,\beta}| = 0.2 \rightarrow 0.6$) in the machine currents ($i_{s\alpha}$, $i_{s\beta}$) represented in stationary reference frame, while the machine was rotating at 25% of the nominal speed, and observe the tracking procedure, Figure 6.35. Finally, the Figure 6.36 shows the machine currents and voltages in *αβ-reference frame* at 25% of the nominal speed.

Formulating the induction machine model in the stationary reference frame simplifies the task of machine direct control, by replacing continuous inputs with discrete ones. In direct control scheme, the inverter model has to be integrated to the machine model and the MPC controller will generate directly the inverter switching states without any kind of modulation.

6.7 Indirect Exp-MPC Control and Comparative Analysis with PI Controllers

To summarize the indirect control scheme using a PWM modulator, different controllers are evaluated on the same platform and under the same conditions. The induction machine was driven by a three-level NPC inverter with one of the controllers discussed above. The program sampling frequency was set to 10 kHz, and the used PWM modulation strategy is the carried-based SPWM. The PWM modulator stands in this case to generate the respective inverter switching states.

The following experimental results present the dynamic behavior of the used current controllers working in both synchronous and stationary coordinates. To observe the controller dynamic it is enough here to apply a step in the torque producing current ($i_{sq} = 0 \rightarrow 0.4$), while the flux producing current was kept at its nominal value ($i_{sd} = 0.35$) for a small time window (time division is 1 msec). For better presentation the current signal i_{sd} was dragged in the figures to zero.

All the controllers (Complex PI (CPI), Passive SMPC$_{dq}$, Passive SMPC$_{\alpha\beta}$, EMPC$_{dq}$, and EMPC$_{\alpha\beta}$) present a high dynamic behavior with (0.5 msec) settling time and almost zero overshooting. Whereas the scalar PI (SPI) controller optimized according to symmetric optimum strategy exhibits longer settling time (2.5 msec) and much higher overshooting (25%) under the same conditions. the subscripts (□$_{dq}$, □$_{\alpha\beta}$) in the controller names refer to the system coordinates of the controlled signals.

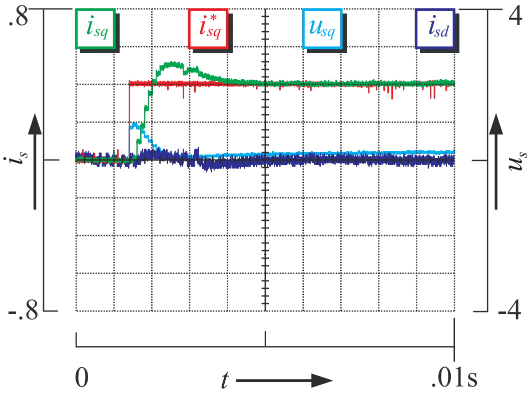


Figure 6.37: IM current response using a SPI$_{dq}$ controller

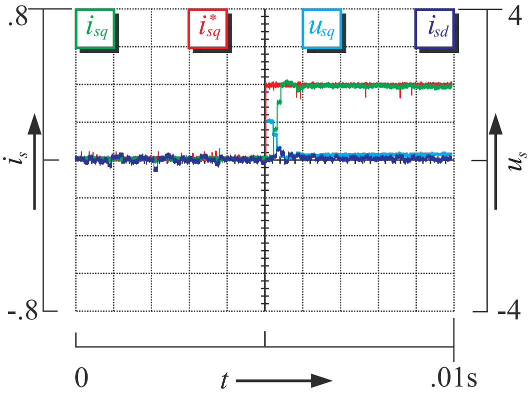
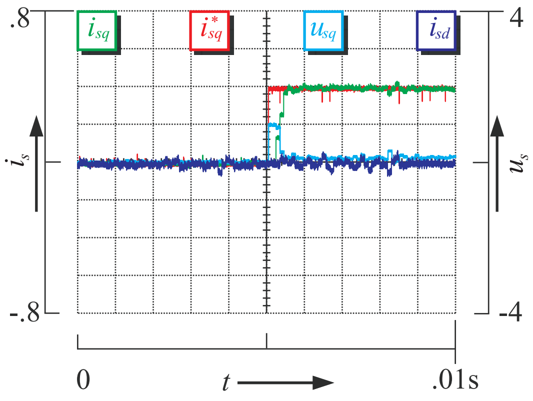


Figure 6.38: IM current response using a CPI$_{dq}$ controller



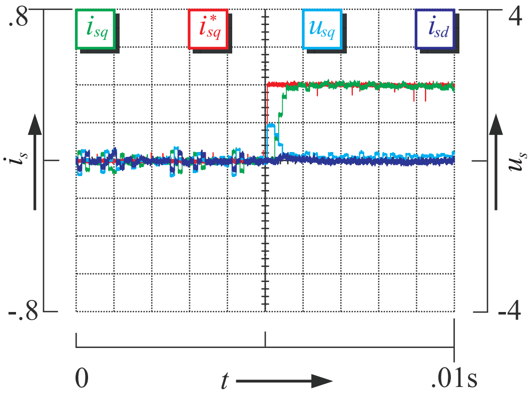
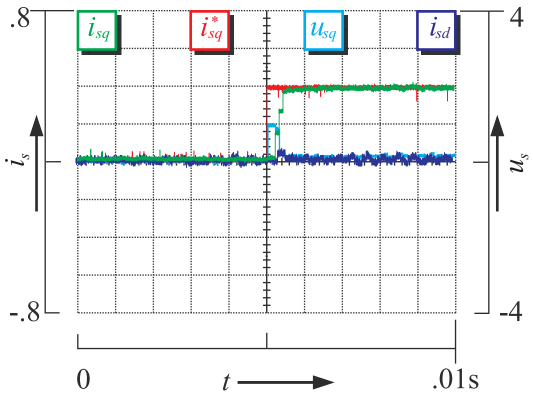


Figure 6.39: IM current response using a passive SMPC$_{dq}$ controller

Figure 6.40: IM current response using a passive SMPC$_{\alpha\beta}$ controller



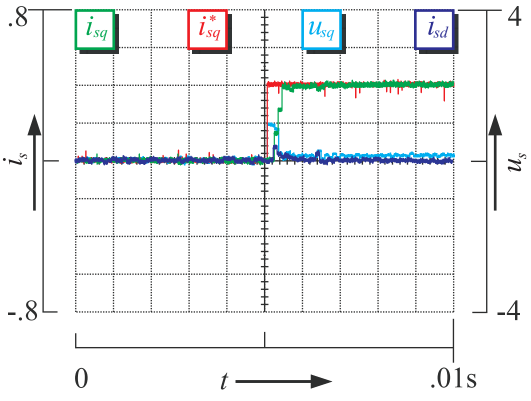


Figure 6.41: IM current response using an EMPC$_{dq}$ controller

Figure 6.42**:** IM current response using an EMPC$_{\alpha\beta}$ controller

Evaluation procedure of the different controllers on the proposed digital system results in an equivalent execution times. The following Table 6.I summarizes the times elapsed by evaluating the proposed MPC control structures comparing to the conventional scalar and complex PI controllers. The necessary times to evaluate a single MIMO-MPC controller is almost equal to that times to evaluate two SISO-SPI or SISO-CPI current controllers, but with remarkable performance.

Table 6.I: The online evaluation complexity of the proposed Exp-MPC controllers

Controller	**Q**	**R**	N_p	**Number of Regions**	**Size of Exp-MPC**	**Average Evaluation Time**
SISO-SPI	---	---	---	---	---	1.00 µsec
SISO-CPI	---	---	---	---	---	1.25 µsec
MIMO-SMPC	1	1e-99	3	80 R	40 KByte	2.25 µsec
MIMO-EMPC	1	3e-04	3	169 R	70 KByte	2.50 µsec

6.8 Direct Exp-MPC Controller for an Induction Machine

The indirect current controllers (conventional PI and MPC controllers) introduced above work without any modifications for both two and three-level inverters. The generated control inputs have to be modulated by a suitable PWM modulator. The PWM modulation board in the proposed digital system is responsible for driving two- or multilevel voltage source inverters by producing the switching signals properly. For direct control of the inverter both the AC machine and the inverter model are to be considered in the design phase of the MPC controller, Figure 6.43.

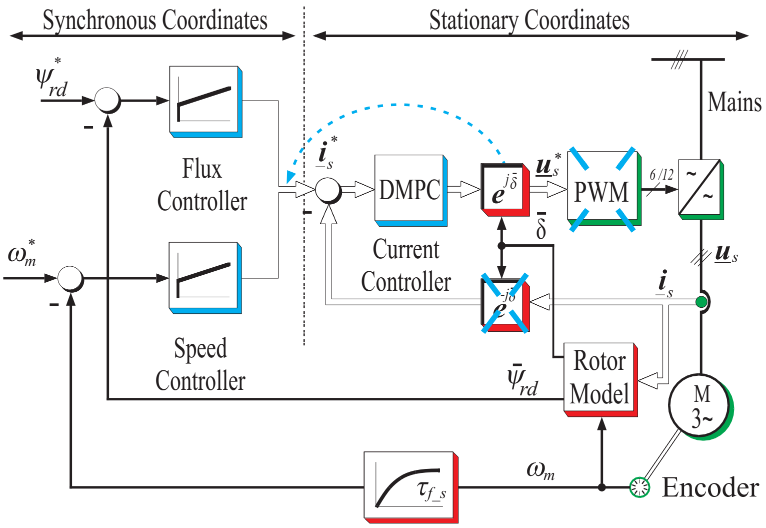


Figure 6.43: The proposed DMPC current control structure

The proposed control structure and following formulation are valid for both kinds of inverters with minor modifications.

For the purpose of a comparative discussion, a brief introduction is introduced here about the structure and function of a two-level voltage source inverter.

Figure 6.44 shows a circuit diagram of a two-level inverter connected to a three-phase AC machine. The inverter is fed by a DC source of voltage u_{dc} followed by one DC-bus capacitor. The power semiconductors of the Insulated-Gate Bipolar transistor (IGBT) type are usually used in the industrial two-level inverters. Active power switches and freewheeling diodes in this topology, however, have the same voltage ratings, namely u_{dc} . The inverter control gives the possibility to connect machine terminals to one of the DC-rails voltage levels ($+u_{dc}/2$, or $-u_{dc}/2$), thus generating two-level switched phase voltages. The power switches are controlled through triggering signals at their gates. The upper and lower power switches in each phase leg are turned on and off in a predefined switching sequence and at given time instants. However, the power switches in each inverter leg are to be switched in a complementary way, such that turning on one of the phase switches will exclude the other one from being turned on. This principle of work has to be projected in the implementation stage by keeping a sufficient interlock time while turning on the switches, to avoid a possible short circuit. It follows, only two switching combinations are allowed in each phase leg. Having three legs in the two-level inverter leads to a total number of $2^3 = 8$ different switching-state voltage vectors are possible. The output voltage space vectors generated by the two-level inverter are defined by:

$$\boldsymbol{u}_s = \frac{2}{3}\frac{u_{dc}}{2}\left(\mathbf{1}.s_a + \boldsymbol{a}.s_b + \boldsymbol{a}^2.s_c\right):\ s_{a,b,c} \in \{-1,+1\},\ \boldsymbol{a} = e^{j2\pi/3} \tag{6.52}$$

where the variables s_a, s_b, and s_c denote the instantaneous value of the selected switching state of each phase.

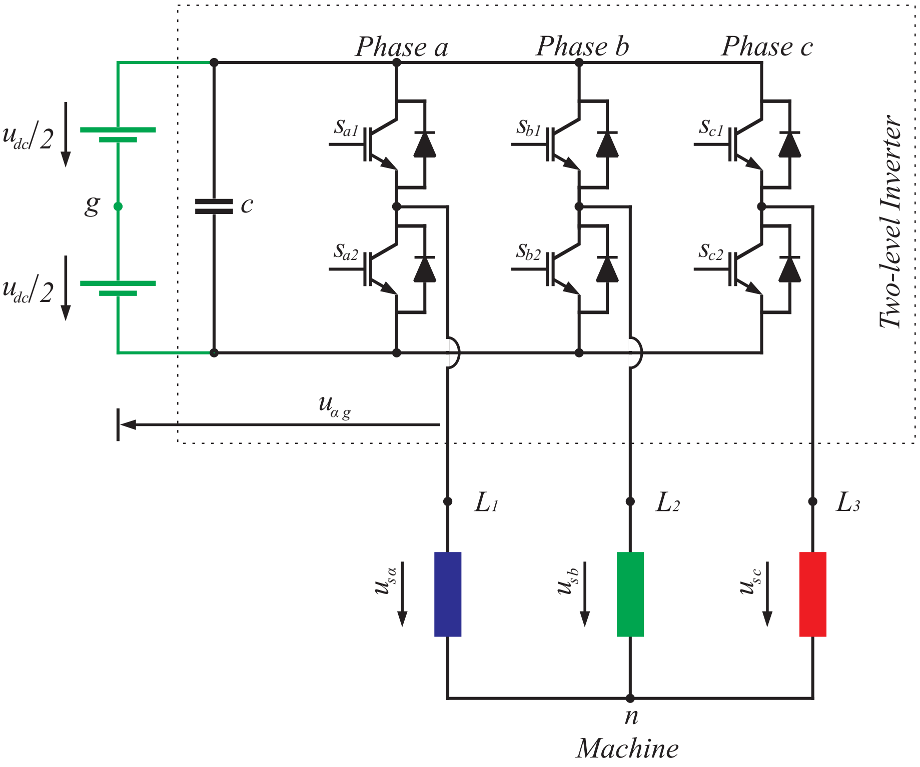


Figure 6.44: Circuit diagram of a two-level voltage source inverter connected to an AC machine

Controlling the two-level inverter using a direct MPC controller (DMPC) was introduced in the literature in implicit and explicit ways.

In the implicit control form, the optimization problem is to be solved online. At each sampling time, the DMPC controller has to evaluate all 8-switching possibilities and chooses the optimal switching state that minimizes a predefined cost function. The cost function aims at minimizing the state-reference tracking error in the next sampling step using the linear norm, equation (6.53). Both direct generalized predictive control (GPC) in [76] and finite-set MPC controller in [119] use the complete machine model, equation (6.45), by considering the induced voltages $\boldsymbol{u}_i$ as input-disturbances in the system model. However, following the proposed implicit solution in [76], [119] and solve the optimization problem online is complex and time consuming, especially for a prediction horizon longer than 2 steps.

$$\boldsymbol{J} = \left| i_{s\alpha} - i^*_{s\alpha} \right|_{k+1} + \left| i_{s\beta} - i^*_{s\beta} \right|_{k+1} \tag{6.53}$$

Employing the multi-parametric programming approach to solve the DMPC optimization problem using the MPT toolbox introduces to the explicit solution of direct MPC controller (Exp-DMPC). The time consuming part of the optimization problem is moved off-line and a mere function evaluation problem has to be evaluated online. Solving the hybrid structured optimization problem using the MPT toolbox subjects to some limitations and restrictions: Only a linear system model and a linear optimization norm are possible. However, solving the MPC optimization problem using the linear norm results in much higher region number than of using the quadratic norm. With a linear norm, around 1500 regions are expected for a two-level inverter with only two steps prediction horizon. Furthermore, the Exp-DMPC controller in [76], [81], and [100] relies on the simplified linear AC machine model in equation (6.54) and neglects the nonlinear terms. The hybrid structured system model used for designing the DMPC controller results in equation (6.56) by combining both the simplified machine model in equation (6.54) and the two-level inverter model in equation (6.55). The generated control inputs are discrete (integer) values, which represent the switching states (s_a , s_b , s_c) of the inverter. Whereas the compensation terms ($-(1/l_\sigma).\boldsymbol{u}_i$) are continuous values. For that reason, the compensation procedure of the induced voltages cannot be applied for this hybrid structure outside the controller, and in [76] and [100] it was not clear how the induced voltages are considered and compensated. Neglecting the compensation of the induced voltages in the design phase of the Exp-DMPC controller affects the tracking performance of current signals in [81]. The introduced results in [81] neglected this compensation, and the controller was designed using the simplified machine model in equation (6.54) without considering the induced voltages, where a tracking error was clearly noticed.

The simplified AC machine model in stationary coordinates is:

$$\frac{d}{d\tau}\begin{bmatrix} i_{s\alpha} \\ i_{s\beta} \end{bmatrix} = \begin{bmatrix} -\frac{1}{\tau'_\sigma} & 0 \\ 0 & -\frac{1}{\tau'_\sigma} \end{bmatrix} . \begin{bmatrix} i_{s\alpha} \\ i_{s\beta} \end{bmatrix} + \frac{1}{l_\sigma}\begin{bmatrix} u_{s\alpha} \\ u_{s\beta} \end{bmatrix} \tag{6.54}$$

Two-level inverter model is:

$$\begin{bmatrix} u_{s\alpha} \\ u_{s\beta} \end{bmatrix} = \mathbf{CM}.\begin{bmatrix} s_a \\ s_b \\ s_c \end{bmatrix} : s_{a,b,c} \in \{-1,+1\}; \mathbf{CM} = \frac{2}{3}\frac{u_{dc}}{2}\begin{bmatrix} 1 & \frac{-1}{2} & \frac{-1}{2} \\ 0 & \frac{\sqrt{3}}{2} & \frac{-\sqrt{3}}{2} \end{bmatrix} \tag{6.55}$$

The simplified hybrid DMPC model for two-level inverter is:

$$\frac{d}{d\tau}\begin{bmatrix} i_{s\alpha} \\ i_{s\beta} \end{bmatrix} = \begin{bmatrix} -\frac{1}{\tau'_\sigma} & 0 \\ 0 & -\frac{1}{\tau'_\sigma} \end{bmatrix} . \begin{bmatrix} i_{s\alpha} \\ i_{s\beta} \end{bmatrix} + \frac{1}{l_\sigma}\mathbf{CM}.\begin{bmatrix} s_a \\ s_b \\ s_c \end{bmatrix} \tag{6.56}$$

6.8.1 Direct Control of a Three-Level NPC Inverter

Some contributions were introduced in the literature for direct control of multilevel inverters. The higher number of possible switching states increases the online computation complexity. Highly time demand to evaluate the cost function for all of inverter switching possibilities limits the evaluation of the implicit DMPC to one step prediction horizon [119].

In the finite-set control scheme [119], the DMPC controller has to evaluate all 27-switching possibilities and chooses the optimal switching state that minimizes a predefined cost function [120]. The linear cost function in the NPC topology aims at minimizing the state-reference tracking error in the next sampling step, switching losses, n_p-potential deviation, or all together. Following this finite-set control scheme to evaluate the cost function with a quadratic norm or with more than one prediction step for multilevel inverters is time critical.

The pioneer work introduced in [74] makes the online evaluation of DMPC controller with long range prediction horizon realizable. This control scheme exclude from the cost function evaluation all the switching state sequences that make the predicted current trajectories violate the feasibility bounds. At time step k and using the last applied switching state vector u_{k-1}, all admissible inverter switching sequences are enumerated over a short provided switching horizon N_s. As a result 27^{Ns} possible switching sequences are obtained. These switching sequences are evaluated using the system model, equation (6.45), to build predicted current trajectories over the prediction horizon N_P. The switching sequences that lead to current trajectories violating the feasible bound of the machine currents are discarded. Finally, only the switching sequences that guarantee the problem feasibility for N_s are evaluated using the cost function. The cost function in [74], [114] intends to minimize the power switch commutations and the tracking error. Finally, only the first switching state vector of the optimal switching sequence is applied to the inverter at time step k+1 and the procedure is repeated again. However, solving the optimization problem with longer switching horizon is time extensive, for that reason a sup-optimal solution was proposed in [116]. For more information about this control scheme the author refers to the publications from T. Geyer [113], [114], [115], [116], [115], and [116].

The explicit solution of DMPC control for multilevel inverter was not contributed yet, and it is firstly introduced in this work. In the following discussion with direct control of the NPC inverter, the DMPC model composed of both machine and inverter models combined together in one hybrid structured system model. The proposed models below can be reformulated to work as well for a two-level inverter. However, the introduced results are obtained for a three-level NPC inverter only.

For a free-offset tracking of the current signals, it is most convenient to consider the extended machine model in the stationary reference frame, equation (6.50). The extended machine model assures the compensation of the induced voltages implicitly (inside the controller structure). The introduced model for NPC inverters in [5] is employed here, equation (6.57). The NPC inverter model introduces a new term, namely the n_p-potential related term. The effect of this term on the model appears whenever one of the inverter outputs is connected to the neutral point (n_p) and causes an unbalancing in the neutral point potential.

The following inverter model is valid for both two- and three-level NPC inverter, equation (6.57).

$$\begin{bmatrix} u_{s\alpha} \\ u_{s\beta} \end{bmatrix} = \mathbf{CM}.\left(\underbrace{\underbrace{\begin{bmatrix} s_a \\ s_b \\ s_c \end{bmatrix}}_{2\text{Level Inverter}} + \underbrace{\begin{bmatrix} 1-|s_a| \\ 1-|s_b| \\ 1-|s_c| \end{bmatrix} \frac{u_\delta}{u_{dc}}}_{np-\text{potential effect}}}_{3\text{Level Inverter}} \right) : s_{a,b,c} \in \begin{cases} \{-1,+1\} : \text{Two}-\text{level Inverter} \\ \{-1,0,+1\} : \text{Three}-\text{level Inverter} \end{cases} \tag{6.57}$$

The switching states take the discrete values {-1, +1} for a two-level inverter and {-1, 0, +1} for a three-level inverter. The notation (-1, 0, +1) means connecting the inverter outputs of phase (*a*, *b*, *c*) to one of the inverter voltage levels ($-u_{dc}/2, n_p, +u_{dc}/2$), accordingly; Figure 6.45, Figure 6.46. The effect of the n_p-potential term arises whenever a neutral point unbalancing exists. The introduced discrete model for the NPC inverter could be also used for a two-level inverter by neglecting the n_p-potential related term and considering the two switching possibilities {-1, +1}. The voltage level of the neutral point should be kept at zero volts. However, in a two-level inverter only one DC-capacitor exists, and there is no unbalancing problem.

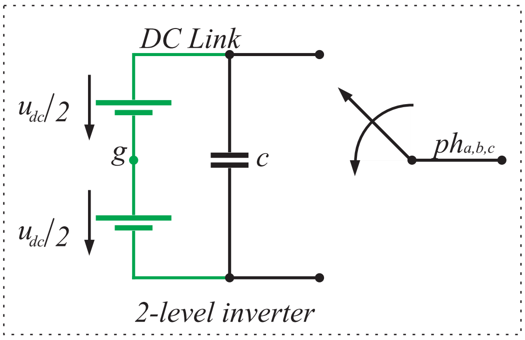


Figure 6.45: Phase voltage switching possibilities for a two-level inverter

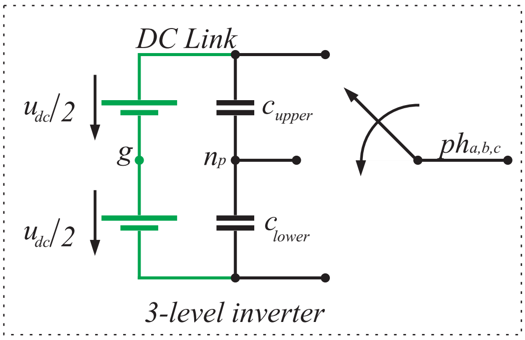


Figure 6.46: Phase voltage switching possibilities for a three-level NPC inverter

Furthermore, the n_p-potential related term in the equation (6.57) could be combined with the compensation term $\mathbf{x}^{\mathrm{d}}$ introduced before in the extended MPC model. Again the hybrid structured compensation term (induced voltages and n_p-potential effect) in equation (6.58) is less dynamic than the machine current model and its effect on the MPC current controller appears mainly in the steady state. Therefore, it could be considered as a new disturbance term added as before to the state variables. Introducing the n_p-potential related term to the compensation procedure enhances mainly the tracking procedure during the unbalancing without a significant improvement of the balancing procedure. The new formulated compensation term in equation (6.58) is to be estimated/ calculated at each sampling time and introduced to the DMPC controller structure, equation (6.59).

$$\mathbf{x}^{\mathrm{d}} = \frac{1}{l_\sigma} \left(\mathbf{CM}. \begin{bmatrix} 1-|s_a| \\ 1-|s_b| \\ 1-|s_c| \end{bmatrix} \frac{u_\delta}{u_{dc}} - \begin{bmatrix} u_{i\alpha} \\ u_{i\beta} \end{bmatrix} \right) \tag{6.58}$$

Composing the extended machine model in (6.50) and the NPC inverter model in (6.57) leads to a hybrid structured system model used to design the DMPC controller, equation (6.59).

$$\frac{d}{d\tau}\begin{bmatrix} i_{s\alpha} \\ i_{s\beta} \\ x_{\alpha}^{d} \\ x_{\beta}^{d} \end{bmatrix} = \begin{bmatrix} -\frac{1}{\tau_{\sigma}^{'}} & 0 & 1 & 0 \\ 0 & -\frac{1}{\tau_{\sigma}^{'}} & 0 & 1 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} . \begin{bmatrix} i_{s\alpha} \\ i_{s\beta} \\ x_{\alpha}^{d} \\ x_{\beta}^{d} \end{bmatrix} + \frac{1}{l_{\sigma}} \begin{bmatrix} & \mathbf{CM} & \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} . \left(\begin{bmatrix} s_a \\ s_b \\ s_c \end{bmatrix} \right)$$

$$s_{a,b,c} \in \begin{cases} \{-1,+1\} : \text{Two} - \text{level Inverter} \\ \\ \{-1,0,+1\} : \text{Three-level Inverter} \end{cases} \tag{6.59}$$

$$\mathbf{x}^{\mathrm{d}} = \begin{cases} \dfrac{1}{l_{\sigma}} \left(\mathbf{0} - \begin{bmatrix} u_{i\alpha} \\ u_{i\beta} \end{bmatrix} \right) : \text{Two-level Inverter} \\ \\ \dfrac{1}{l_{\sigma}} \left(\mathbf{CM} . \begin{bmatrix} 1 - |s_a| \\ 1 - |s_b| \\ 1 - |s_c| \end{bmatrix} \dfrac{u_{\delta}}{u_{dc}} - \begin{bmatrix} u_{i\alpha} \\ u_{i\beta} \end{bmatrix} \right) : \text{Three-level Inverter} \end{cases}$$

The output equation is:

$$\begin{bmatrix} i_{s\alpha} \\ i_{s\beta} \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix} . \begin{bmatrix} i_{s\alpha} \\ i_{s\beta} \\ x_{\alpha}^{d} \\ x_{\beta}^{d} \end{bmatrix} + \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} . \begin{bmatrix} s_a \\ s_b \\ s_c \end{bmatrix} \tag{6.60}$$

Tips and Tricks

The EDMPC optimization problem can be formulated using either the MPT or HYSDEL toolbox. Each of them has some capabilities and some limitations in formulating the constrained optimization problem. MPT toolbox supports the continuous and discrete variables, while HYSDEL toolbox supports the continuous and Boolean variables. HYSDEL toolbox gives more flexibility in formulating the system constraints using if-then-else rules, which is not easily achievable using the MPT toolbox. However, HYSDEL toolbox is a tool to formulate the MPC problem and delivers a PWA system structure. Whereas solving the optimization problem is only possible using the MPT toolbox. Therefore, it is advisable to formulate the MPC problem with HYSDEL and solve the resulting PWA problem using the MPT toolbox. Here some tips and tricks regarding the formulation of the EDMPC optimization problem using the MPT and HYSDEL toolboxes.

MPT toolbox [77]: Before introducing the system model in the equation (6.59) to the MPT toolbox, it has to be reformulated to consider the free-offset tracking case, equation (6.3). This formulation introduces a new vector to the state variables (the last updated control inputs s_{k-1}) and replaces the control input vector of the optimization problem with a new vector (control efforts Δ*s*$_k$). The resulting optimal control input to be applied to the inverter results as $s_k = s_{k-1} + \Delta s_k$. The constraints on the both vectors (new state vector and control effort vector) are to be introduced as in equation (6.61).

$$\begin{aligned} s_{k-1} &\in \{-1, 0, +1\} \\ \Delta s_k &\in \{-2, -1, 0, +1, +2\} \end{aligned} \tag{6.61}$$

where *s* is the switching state associated with phase *a*, *b*, or *c*.

The optimization problem is to be solved by finding the control effort Δ*s*$_k$ that minimizes the cost function without violating the system constraints. Giving the constraints introduced above to the MPT toolbox without further extension will results in a new control input that violates the state constraints $s_k \notin \{-1, 0, +1\}$, Figure 6.47. For example, being at the switching state $(s_k = 0)$ and getting a new optimal control effort $(\Delta s_k = 2)$ will result in a new control input $(s_k = 2)$, which violates the state constraints in equation (6.61).

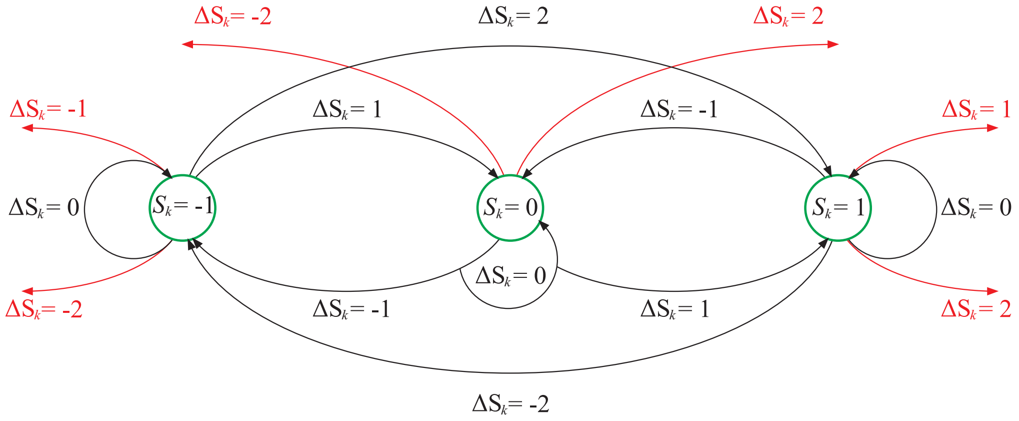


Figure 6.47: Finite-state machine for switching possibilities of a three-level NPC inverter

The switching possibilities that violate the state constraints are summarized in Table 6.II.

Table 6.II: Prohibited switching possibilities

State Variable *s*$_{k-1}$	**Control Effort Δ*s*$_k$**		**Control Input *s*$_k$**
-1	-1		-2
-1	-2		-3
0	-2		-2
0	-2		-2
1	1		2
1	2		3

Here new constraints are to be introduced to assure the right choice of the control effort. To address this conflict, the state constraints on the last applied control input s_{k-1} are to be relaxed, equation (6.62).

$$\begin{aligned} s_{k\text{-}1} &\in \left[-1-\xi, +1+\xi\right] : 0 < \xi < 1 \\ \Delta s_k &\in \left\{-2, -1, 0, +1, +2\right\} \end{aligned} \tag{6.62}$$

This space extension assures that the calculated control input s_k will be in the required range $s_k \in \left\{-1, 0, +1\right\}$. Similar discussion can be done for a two-level inverter to get the following constraints:

$$\begin{aligned} s_{k\text{-}1} &\in \left[-1-\xi, +1+\xi\right] : 0 < \xi < 1 \\ \Delta s_k &\in \left\{-1, 0, +1\right\} \end{aligned} \tag{6.63}$$

HYSDEL toolbox [78]: It provides more flexibility to describe the interaction between continuous dynamics described by differential equations (6.50) and logical components described by finite-state machines or if-then-else rules in Figure 6.47. However, HYSDEL toolbox does not support the discrete type of variables, where only continuous and Boolean variables are allowed. To address this issue, the discrete variables are to be encoded somehow using auxiliary Boolean variables. The inverter switching states (especially for multilevel inverter) are discrete variables (equation (6.64)) and then they are to be represented again in HYSDEL using auxiliary Boolean variables.

$$s_{a,b,c} \in \begin{cases} \left\{-1, +1\right\} : \text{Two-level Inverter} \\ \\ \left\{-1, 0, +1\right\} : \text{Three-level Inverter} \end{cases} \tag{6.64}$$

The inverter switching states could be encoded using two auxiliary Boolean variables as follows:

$$s = s_{pos} - s_{neg} \quad : s_{pos}, s_{neg} \in \left\{'0', '1'\right\} \tag{6.65}$$

Not all combinations of the new auxiliary variables are possible for two and three-level inverters as shown in Table 6.III and Table 6.IV, respectively.

Table 6.III: Switching state possibilities for a two-level inverter

Inverter Switches		**Switching Possibility**	**Auxiliary Variables**	
s_1	**s_2**	**s**	**s_{pos}**	**s_{neg}**
‘0’	‘0’	Prohibited	‘0’	‘0’
‘0’	‘1’	-1	‘0’	‘1’
‘1’	‘0’	+1	‘1’	‘0’
‘1’	‘1’	Prohibited	‘1’	‘1’

To exclude the prohibited switching possibilities, some constraints have to be introduced on the auxiliary Boolean variables. The following state constraints assure the correct generation of the discrete switching values for two and three-level inverter:

$$\begin{aligned} & s_{pos} + s_{neg} \overset{!}{\leq} 1 \\ & \left(s_{pos} + s_{neg} \overset{!}{\leq} 1 \right) \& \left(1 - \left(s_{pos} + s_{neg} \right) \overset{!}{\leq} 0 \right) \end{aligned} \tag{6.66}$$

where the first condition stands for three-level inverter, and the second one for two-level inverter.

Table 6.IV: Switching state possibilities for a three-level NPC inverter

Inverter Switches				**Switching Possibility**	**Auxiliary Variables**	
s_1	**s_2**	**s_3**	**s_4**	**s**	**s_{pos}**	**s_{neg}**
‘0’	‘1’	‘1’	‘0’	0	‘0’	‘0’
‘0’	‘0’	‘1’	‘1’	-1	‘0’	‘1’
‘1’	‘1’	‘0’	‘0’	+1	‘1’	‘0’
‘1’	‘1’	‘1’	‘1’	Prohibited	‘1’	‘1’

Experimental Results

The explicit solution of direct model-based predictive control using the extended machine model (Exp-EDMPC) results for a three-level NPC inverter in more than 3500 regions with a linear norm and two steps prediction horizon. Table 6.V shows the average evaluation time of the explicit solution of different MPC controllers evaluated on the same platform [26], [27].

Table 6.V: The online evaluation complexity of the proposed Exp-MPC controllers

Controller	**Q**	**R**	**N_p**	**Number of Regions**	**Size of Exp-MPC**	**Average Evaluation Time**
SISO-SPI	---	---	---	---	---	1.00 µsec
SISO-CPI	---	---	---	---	---	1.25 µsec
MIMO-SMPC	1	1e-99	3	80 R	40 KByte	2.25 µsec
MIMO-SMPC	1	3e-03	3	537 R	210 KByte	2.75 µsec
MIMO-EMPC	1	3e-04	3	169 R	70 KByte	2.50 µsec
MIMO-DMPC	1	1e-99	2	2566 R	430 KByte	2.75 µsec
MIMO-EDMPC	1	1e-99	2	3566 R	702 KByte	2.75 µsec

Increasing the prediction horizon one step more, around 10000 regions are expected. The following experimental results were evaluated using EDMPC controller with a linear norm and 2 steps prediction horizon. Figure 6.48 and Figure 6.49 show the clear effect of neglecting the compensation terms on the tracking performance. In Figure 6.48 a step in the torque producing current ($i_{sq} = 0 \rightarrow 0.4$) was applied, while the flux producing current was kept at its nominal value ($i_{sd} = 0.35$) and for a large time scale (time division is 20 msec). For better presentation the current signal i_{sd} was dragged in the figures to zero. The introduced results in Figure 6.49 are similar to that one presented in [81] but for different kind of inverters (three-level NPC inverter instead of a two-level inverter in [81]) and without considering the compensation. The figure shows the induction machine currents in stationary coordinates at 75% on the nominal speed. To prove the efficiency of the proposed controller structure in dynamic and steady state, different evaluation tests on the Exp-EMPC were applied. Figure 6.50 shows controller dynamic in transient by applying step in the torque producing current component ($i_{sq} = 0 \rightarrow 0.4$) while keeping the machine fully magnetized ($i_{sd} = 0.35$). The figure shows the high dynamic performance of the controller with (0.5 msec) settling time and no overshooting. The efficiency of the compensation procedure of the rotor-induced voltages is evaluated by applying a step ($\left|i_{s\alpha,\beta}\right| = 0.6 \rightarrow 0.2$) in both machine currents ($i_{s\alpha}$, $i_{s\beta}$) and observing the tracking procedure, while the machine was rotating at 50% of the nominal speed, Figure 6.51. Figure 6.52 and Figure 6.53 evaluate the controller behavior under load conditions. The machine currents in stationary reference frame are presented with their reference in normal operation at 75% of the nominal speed and 20% of the machine nominal torque. The Figure 6.52 shows the machine currents without compensation, where the induced voltages were compensated in Figure 6.53 internally.

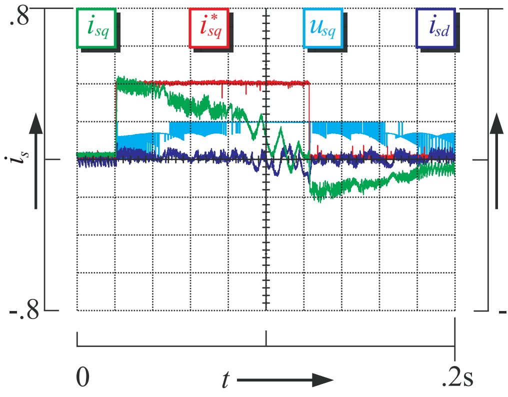


Figure 6.48: IM current response using a DMPC$_{dq}$ controller (without compensation of the induced voltages)

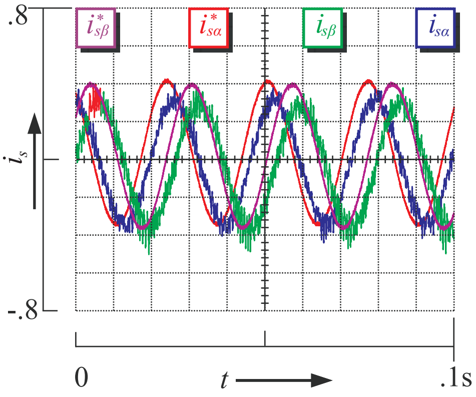


Figure 6.49: IM currents and their references controlled using a DMPC$_{\alpha\beta}$ controller; normal operation at 75% nominal speed

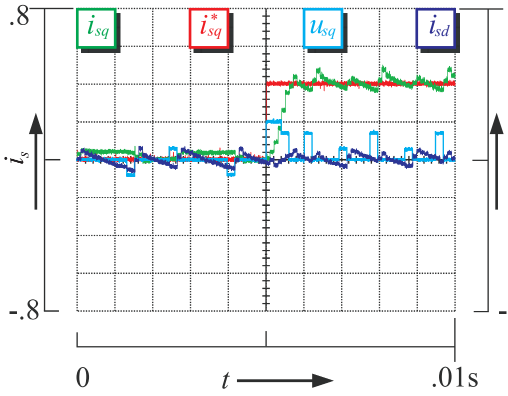


Figure 6.50: IM current response in synchronous coordinates using an EDMPC$_{\alpha\beta}$ controller

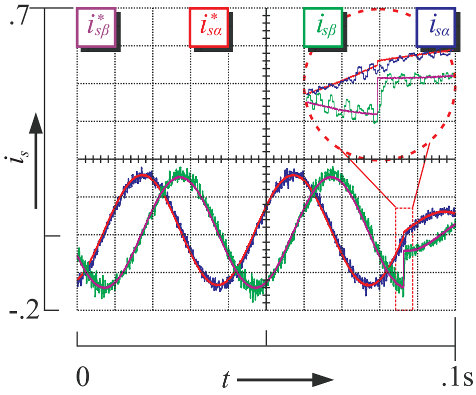


Figure 6.51: IM current responses in stationary coordinates using an EDMPC$_{\alpha\beta}$ controller, steps in ($i_{s\alpha}$, $i_{s\beta}$) currents

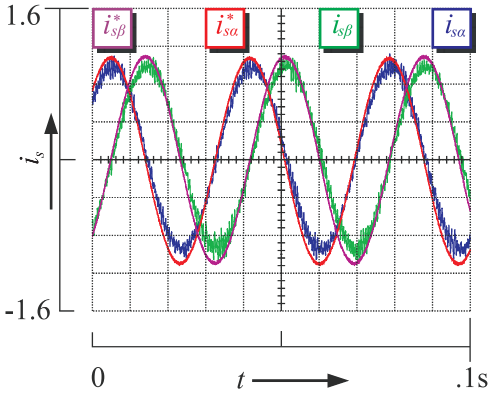


Figure 6.52: IM currents and their references controlled using a DMPC$_{\alpha\beta}$ controller; normal operation at 75% nominal speed and 20% nominal torque (without compensation of the induced voltages)

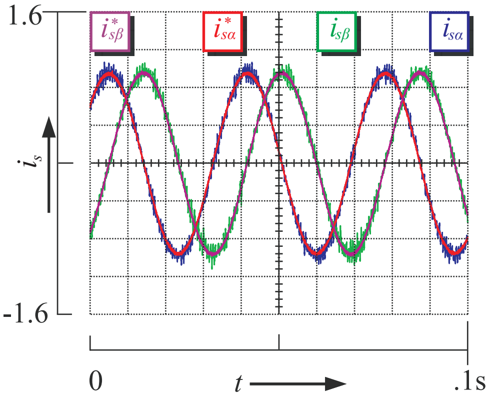


Figure 6.53: IM currents and their references controlled using an EDMPC$_{\alpha\beta}$ controller; normal operation at 75% nominal speed and 20% nominal torque (with compensation of the induced voltages)

6.8.2 Neutral Point Potential Balancing Mechanism

The DC-capacitors of three-level inverters serve as a voltage divider of the full DC-link (u_{dc}). As the outputs of the inverter might be connected to the neutral point (n_p), connecting the inverter to an unbalanced load (as the case in transients or connecting two of the machine windings to the neutral point) leads to an unbalanced voltage distribution between the DC-capacitors (c_{upper} and c_{lower}). The unbalanced voltage-division between the DC-capacitors causes an application of more voltage stress on the active power switches and on the DC-capacitors. High voltage excursions affect the actual voltages on the machine terminals and cause either increase or decrease of the commanded voltages, whenever the neutral point potential error is exist. As a consequence the respective machine currents will also be affected and will not follow their reference signals. Therefore a special attention has to be paid here to maintain the neutral point potential in a tolerance band $u_{\delta_{limit}}$, around 1.4% of the full DC-link voltage [13].

Neutral point (n_p) potential value (u_δ) changes in proportion to the integral of the neutral point current (i_{np}). Value and sign of the neutral point current are determined by the switching states applied to the inverter and the measured phase currents (i_{sa}, i_{sb}, and i_{sc}), equation (6.67).

$$\begin{aligned}\frac{du_\delta}{d\tau} &= \frac{1}{c_d}(i_{sa}\left|s_a\right| + i_{sb}\left|s_b\right| + i_{sc}\left|s_c\right|) \\ &= \frac{2}{u_{dc}}\frac{1}{c_d}\begin{bmatrix} i_{s\alpha} & i_{s\beta}\end{bmatrix}.\mathbf{CM}.\begin{bmatrix}\left|s_a\right| \\ \left|s_b\right| \\ \left|s_c\right|\end{bmatrix}\end{aligned} \tag{6.67}$$

Both machine currents and the switching states are already available in the control scheme. Extending the DMPC model to include the n_p-potential equation is possible, but it will increase the complexity of the controller design as well as the resulting explicit solution of the DMPC controller. However, with the available digital platform and to save the programmer time, it is advised to run the implementation with an explicit solution with less than 5000 regions. Having more than 5000 regions makes the compile procedure of C-code on the available digital platform time consuming. For that reason, the balancing procedure was implemented as a separate structure. However, the proposed balancing procedure does not affect the performance of DMPC controller for a free-offset tracking of the machine currents.

A nonzero n_p-potential error (u_δ) and therefore n_p-current (i_{np}) exist whenever either a small or a medium voltage vector is a part of the applied switching sequence. Wherever the commanded voltage vector is located (in the inner or outer hexagon), a redundant small vector pair forms a part of the used switching sequence to generate the commanded vector. Exploiting this fact, a use of the proper small vector of this pair (as long as it is required) helps to assure the n_p-balancing. The proposed n_p-balancing controller is a kind of hysteresis control, and it is activated whenever the n_p-potential violates the provided limits, Figure 6.54. The controller acts by replacing the DMPC control output (only small vector switching states) with its redundancy as often as needed to keep the n_p-potential inside the provided limits. As this control algorithm depends on the measured n_p-potential only, it is not affected with the noisy machine currents at low switching frequencies.

The following experimental results show the neutral point potential using different controllers under different conditions. Figure 6.55 presents the natural balancing mechanism of the three-level NPC inverter using a scalar PI controller and carrier-based SPWM modulation strategy with the added offset, see section 2.3.1.1. Under this condition the n_p-potential needs around (9 sec) to be recovered again from 15% of the full DC-link deviation. Figure 6.56 shows the necessary time to recover the n_p-balancing from 20% of the full DC-link unbalancing case using different controllers. In region (1) the neutral point was balanced using the natural balancing with the scalar PI controller. In region (2) the machine was controlled using EDMPC without the balancing controller; it is here clear how the EDMPC controller alone fails to keep the neutral point balanced. In region (3) the EDMPC was supported with the proposed balancing controller at its output, the n_p-balancing could be recovered very fast in (0.5 sec) without affecting the tracking performance.

Driving the induction machine in transient and under load conditions increases the effect of the unbalancing. In Figure 6.57 the n_p-potential was observed by driving the induction machine with and without load and under different speed levels (10% nominal load was applied on the induction machine).

For a better appearance of n_p-potential signal u_δ it is presented in Figure 6.60 with a different scaling than the inverter output voltages ($u_{s\alpha}$, $u_{s\beta}$). Its maximum amplitude is in the provided tolerance band $u_{\delta_{limit}}$. The full applied DC-Link is 480 volts, while the n_p-potential does not exceed 5 volts.

Finally, the induction machine currents and voltages are presented in Figure 6.59 and Figure 6.60 for different time scales evaluated on a three-level NPC inverter. The voltage signals ($u_{s\alpha}$, $u_{s\beta}$) are with a variable switching frequency, which does not exceed in this experiment 650 Hz. The experimental results introduced below prove the efficiency of the EDMPC controller with the proposed balancing mechanism.

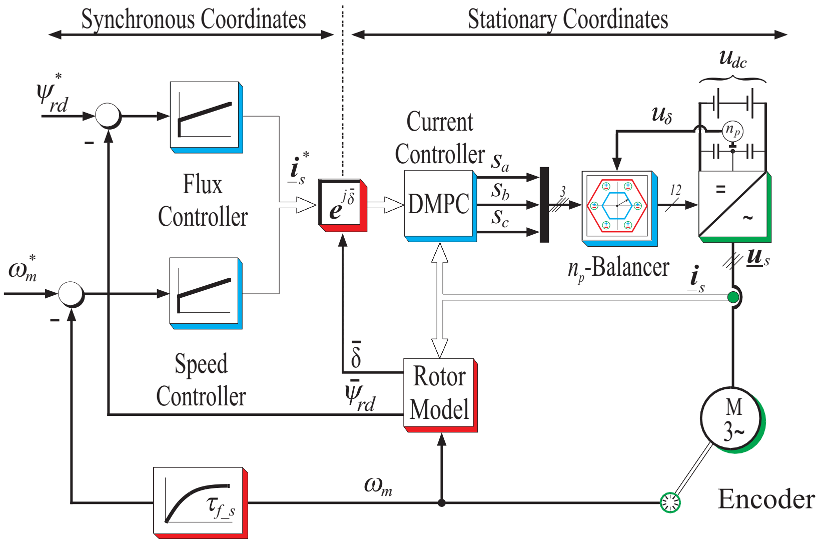


Figure 6.54: The proposed EDMPC current control structure for a three-level NPC inverter

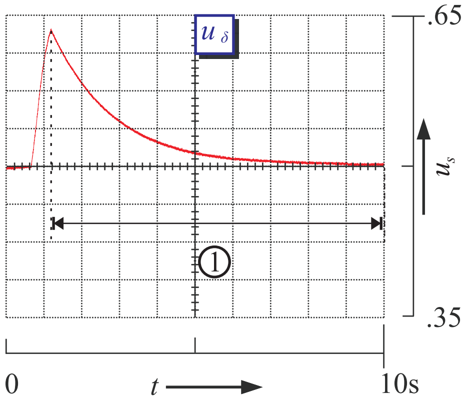


Figure 6.55: Natural balancing mechanism of a three-level NPC inverter using a SPI$_{dq}$ controller and SPWM modulation with offset

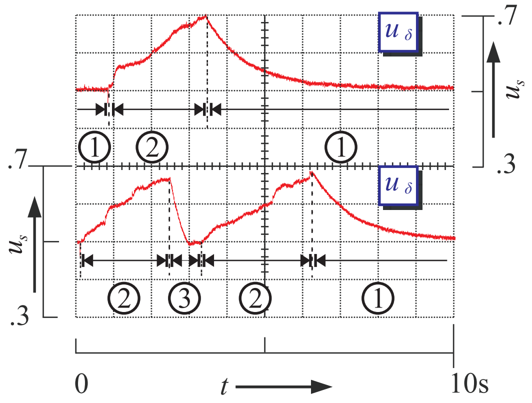


Figure 6.56: Neutral point balancing mechanism using different controllers

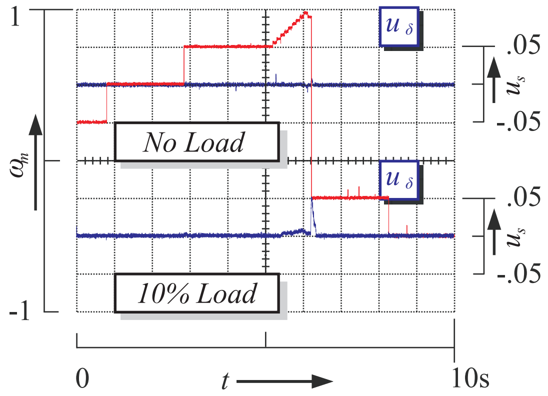


Figure 6.57: Natural balancing of a three-level NPC inverter using an EDMPC$_{\alpha\beta}$ and n_p-balancing controllers with and without loading the machine

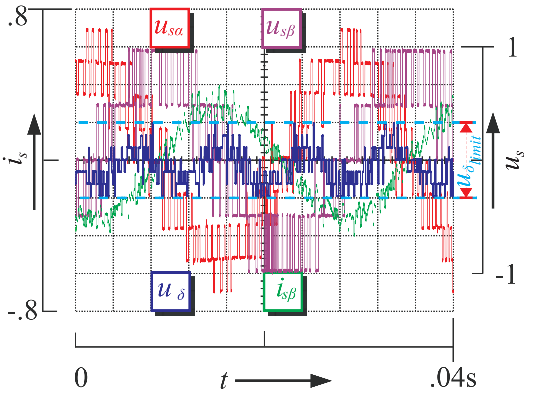


Figure 6.58: IM current and voltages as well as the n_p-potential using an EDMPC$_{\alpha\beta}$ and n_p-balancing controllers; normal operation at 75% of the nominal speed

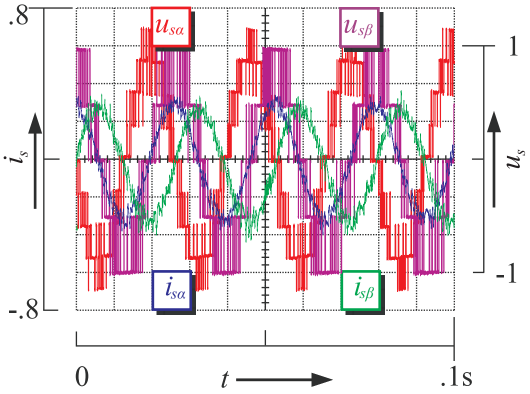


Figure 6.59: IM currents and voltages controlled using an EDMPC$_{\alpha\beta}$ controller; normal operation at 75% nominal speed

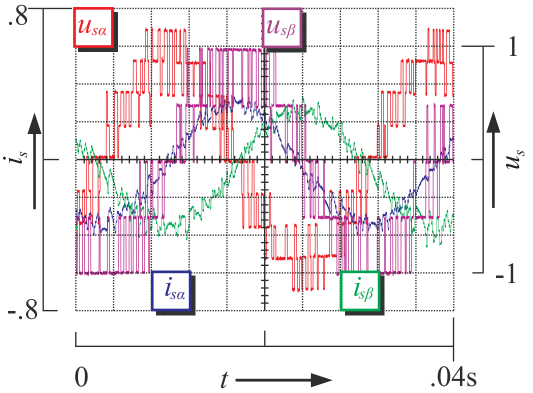


Figure 6.60: IM currents and voltages controlled using an EDMPC$_{\alpha\beta}$ controller; normal operation at 75% nominal speed, with a large time scale

6.9 Predictive Control of a PMSM Machine

6.9.1 Non-Linearity Problem in the PMSM Machine Model

In analogue way to the induction machine, the same compensation procedure can be applied here to the permanent magnet synchronous machine (PMSM). The PMSM machine parameters introduced in the Appendix-A, and the time-normalized machine equations (2.17) are used in the following simulation study. The PMSM machine in this study was controlled in the field-oriented control scheme as in Figure 6.2, using two PI controllers to control the machine currents, or as in Figure 6.5, using a MIMO-MPC current controller, and of course without need for the rotor model to estimate the slip frequency. Similar to the induction machine; adding the mechanical speed as new state variable will introduce to non-linear relationship between the state variables, in the equation (6.68), due to the cross-coupling between the currents themselves and between the currents and the mechanical speed. Therefore, it very advisable to consider it in same way like for the induction machine, either by compensating it outside the controller as voltage terms (6.70), as in the feed-forward compensation method and design a pure MIMO current controller for that (6.71), or by considering it partially (6.79) or completely (6.72) inside the controller as new disturbance terms added to the state variables.

Under the guidelines in section 5.3, different constraints for the MPC problem on the state and input variables were set as follows:

- Current constraints: for the current limitations a rectangular space was given on both dq-currents with some relaxation or extension on the lower and upper limits (zeros and nominal currents) to get a stable controller.
- Voltage constraints: Similar to the current limitations (a rectangular space inside the unit circle or other space defined by linear inequalities) the unit circle was given on both dq-voltages, without relaxing the limits (Hard constraints under the physical limitations of the inverter).

Recalling the PMSM machine equation describing the relationship between the stator currents and the stator voltages in rotor flux coordinates:

$$\boldsymbol{i}_s + \tau_s \frac{d\boldsymbol{i}_s}{d\tau} = \frac{1}{r_s}\left(\boldsymbol{u}_s - \overbrace{\underline{\left(jl_s\omega_s\boldsymbol{i}_s + j\omega_{me}\hat{\boldsymbol{\psi}}_m\right)}}^{\boldsymbol{u}_i}\right) \tag{6.68}$$

where $\tau_s = \frac{l_s}{r_s}$ is the stator time constant, and the $\hat{\boldsymbol{\psi}}_m$ permanent rotor flux, $\boldsymbol{u}_i$ is the induced voltage.

The underlined terms of the equation (6.68) represent both the motion- ($jl_s\omega_s\boldsymbol{i}_s$) and the rotor- ($j\omega_{me}\hat{\boldsymbol{\psi}}_m$) induced voltages $\boldsymbol{u}_i$, respectively. Following the strategy introduced for the induction machine, these terms are to be estimated and compensated either outside the controller in form of voltage feed-forward compensation terms $\boldsymbol{u}_i$ or inside the controller by augmenting the state-vector to contain these terms as input-disturbances $\mathbf{x}^{\mathrm{d}} = -\frac{1}{l_s}\boldsymbol{u}_i$.

The equation (6.69) introduces the discrete-time formulation of the PMSM machine model in synchronous coordinates.

$$\begin{bmatrix} i_{sd} \\ i_{sq} \end{bmatrix}_{k+1} = \begin{bmatrix} 1-\frac{T_0}{\tau_s} & 0 \\ 0 & 1-\frac{T_0}{\tau_s} \end{bmatrix} \cdot \begin{bmatrix} i_{sd} \\ i_{sq} \end{bmatrix}_k + \begin{bmatrix} \frac{T_0}{l_s} & 0 \\ 0 & \frac{T_0}{l_s} \end{bmatrix} \cdot \left(\begin{bmatrix} u_{sd} \\ u_{sq} \end{bmatrix}_k - \begin{bmatrix} u_{id} \\ u_{iq} \end{bmatrix}_k \right) \tag{6.69}$$

$$: \begin{bmatrix} u_{id} \\ u_{iq} \end{bmatrix}_k = l_s \omega_{s\,k} \begin{bmatrix} 0 & -1 \\ 1 & 0 \end{bmatrix} \cdot \begin{bmatrix} i_{sd} \\ i_{sq} \end{bmatrix}_k + \omega_{em\,k} \begin{bmatrix} 0 \\ \hat{\psi}_{m_{dq}} \end{bmatrix}$$

where T_0 is the normalized sampling time $\tau_{sampling}$, $\tau_s = \frac{l_s}{r_s}$ is the stator time constant, ω_{em} is the electrical value of the mechanical angular velocity, ω_s is the stator frequency, $\boldsymbol{u}_i$ is the induced voltage, and $\hat{\psi}_{m_{dq}}$ is the permanent rotor flux.

Working in rotor-flux aligned coordinates for a synchronous machine, the synchronous speed ω_s results directly from the measured mechanical speed ω_m, where there is no slip frequency to be estimated using a rotor model, then $\omega_s = \text{pole pairs} * \omega_m = \omega_{me}$.

The speed-related disturbance terms:

$$\mathbf{x}^{\mathrm{d}} = \begin{bmatrix} x_d^d \\ x_q^d \end{bmatrix} = -\frac{1}{l_s} \begin{bmatrix} u_{id} \\ u_{iq} \end{bmatrix} = -\frac{1}{l_s} \omega_{em} \begin{pmatrix} -l_s i_{sq} \\ l_s i_{sd} + \hat{\psi}_{m_{dq}} \end{pmatrix} \tag{6.70}$$

As it is stated before this disturbance term has to be estimated/ calculated outside the controller at each sampling time and used for the compensation procedure in the following ways.

6.9.2 MIMO-SMPC Current Controller with External Compensation

MPC current controller based on the simplified machine model of PMSM machine is introduced in the following equation (6.71), where the voltage-drops $\boldsymbol{u}_i$ in the applied control inputs u_s were compensated later at the output of the controller in a passive way in form of FFC terms. This formulation leads to 361 regions.

$$\begin{bmatrix} i_{sd} \\ i_{sq} \end{bmatrix}_{k+1} = \begin{bmatrix} 1-\frac{T_0}{\tau_s} & 0 \\ 0 & 1-\frac{T_0}{\tau_s} \end{bmatrix} \cdot \begin{bmatrix} i_{sd} \\ i_{sq} \end{bmatrix}_k + \begin{bmatrix} \frac{T_0}{l_s} & 0 \\ 0 & \frac{T_0}{l_s} \end{bmatrix} \cdot \begin{bmatrix} u_{sd} \\ u_{sq} \end{bmatrix}_k \tag{6.71}$$

As the induced voltage $\boldsymbol{u}_i$ contains the measured mechanical speed ω_{em} , the estimation procedure of these terms for PMSM machine is also affected with the used speed filter time constant. Following the estimation method of the total induced voltages $\boldsymbol{u}_i$ at once, as introduced in section 6.4.4, leads to a perfect estimation, as shown in the following simulation results. In Figure 6.61 a full current step was applied to the torque producing current $i_{sq} = 0 \rightarrow 1 \rightarrow 0$, while the flux producing current was kept at zero $i_{sd} = 0$. The compensation procedure of the induced voltages was done at the output of the simple MPC controller in a passive way.

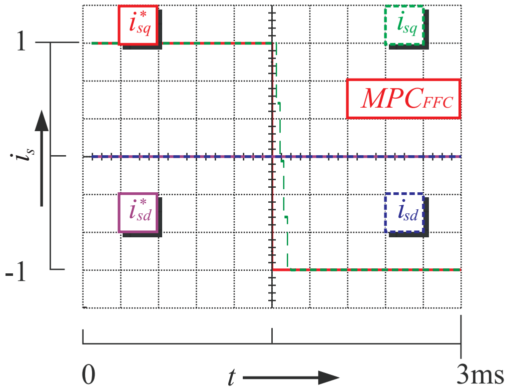


Figure 6.61: PMSM current response using a passive SMPC$_{dq}$ controller

6.9.3 MIMO-EMPC Current Controller with Internal Compensation

MPC current controller based on the extended machine model is introduced in the following equation (6.72), where the voltage-drops $\boldsymbol{u}_i$ in the applied control inputs $\boldsymbol{u}_s$ were estimated and compensated inside the control structure as an input-disturbance (6.70) by state-vector augmentation $\mathbf{x}^{\mathrm{d}} = \begin{bmatrix} x_d^d \\ x_q^d \end{bmatrix} = -\frac{1}{l_s}\begin{bmatrix} u_{id} \\ u_{iq} \end{bmatrix}$. This formulation leads to 725 regions.

$$\begin{bmatrix} i_{sd} \\ i_{sq} \\ \mathbf{x}^{\mathrm{d}} \end{bmatrix}_{k+1} = \begin{bmatrix} 1-\dfrac{T_0}{\tau_s} & 0 & \mathbf{I} \\ 0 & 1-\dfrac{T_0}{\tau_s} & \mathbf{I} \\ \mathbf{0} & \mathbf{0} & \mathbf{I} \end{bmatrix} \cdot \begin{bmatrix} i_{sd} \\ i_{sq} \\ \mathbf{x}^{\mathrm{d}} \end{bmatrix}_k + \begin{bmatrix} \dfrac{T_0}{l_s} & 0 \\ 0 & \dfrac{T_0}{l_s} \\ \mathbf{0} & \mathbf{0} \end{bmatrix} \cdot \begin{bmatrix} u_{sd} \\ u_{sq} \end{bmatrix}_k \tag{6.72}$$

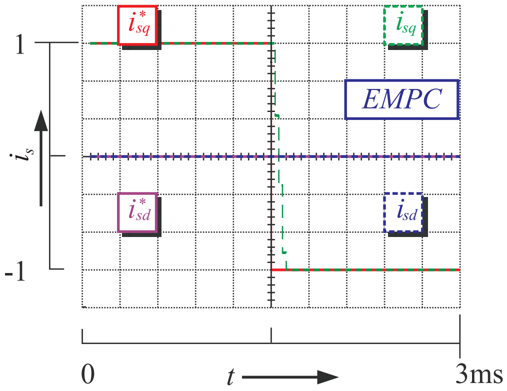


Figure 6.62: PMSM current response using an EMPC$_{dq}$ controller

In the simulation result of Figure 6.62, a full current step was applied to the torque producing current $i_{sq} = 0 \rightarrow 1 \rightarrow 0$, while the flux producing current was kept at zero value $i_{sd} = 0$. Both figures (Figure 6.62 and Figure 6.62) present a fast dynamic and zero offset in steady state. To evaluate the performance of the MPC controller in both cases, the PI controller optimized according to symmetric optimum method for the introduced PMSM machine is introduced in the next section.

6.9.4 Comparative Analysis with Scalar PI Controller

For a linear control of PMSM machine, the following simplified linear machine model, equation (6.73), is considered to adjust the simple PI controller parameters according to symmetric optimum method.

$$\boldsymbol{i}_s + \tau_s \frac{d\boldsymbol{i}_s}{d\tau} = \frac{1}{r_s}\boldsymbol{u}_s \qquad (6.73)$$

Although the PI controller is able to compensate the voltage drops in the system input and no need for further compensation, the compensation of the induced voltages $\boldsymbol{u}_i$ at the output of the PI controller will help to reduce the cross-coupling effect between the machine currents, especially at low switching frequency. The simulation results of a simple PI controller with feed-forward compensation are considered here as the basis for the comparison with the different MPC controller configurations. In Figure 6.63 a full current step was applied to the torque producing current $i_{sq} = 0 \rightarrow 1 \rightarrow 0$, while the flux producing current was kept at zero $i_{sd} = 0$. The PI controller exhibits a fast dynamic and small coupling effect of the applied step of i_{sq} in i_{sd} current, but on the other hand a large overshooting and large settling time can be observed here, comparing with both introduced MPC controllers.

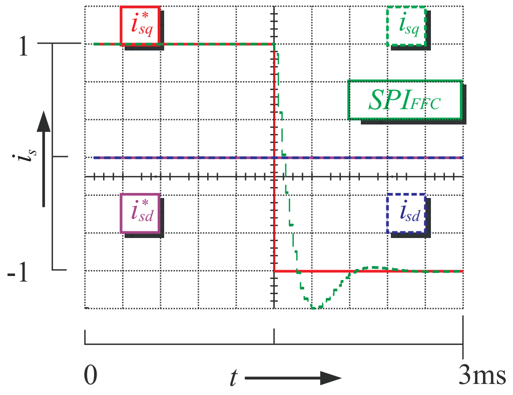


Figure 6.63: PMSM current response using a SPI_{dq} controller with FFC

To evaluate the tracking procedure when the machine accelerates and under load condition, the Figure 6.64 shows the machine currents by applying a full nominal speed step at time step (t_1) and a half nominal torque step at time step (t_2). All the controllers (SMPC_{FFC}, EMPC, and SPI_{FFC}) exhibit a perfect current tracking, where in Figure 6.64 it is difficult to distinguish between the reference signals and the actual ones.

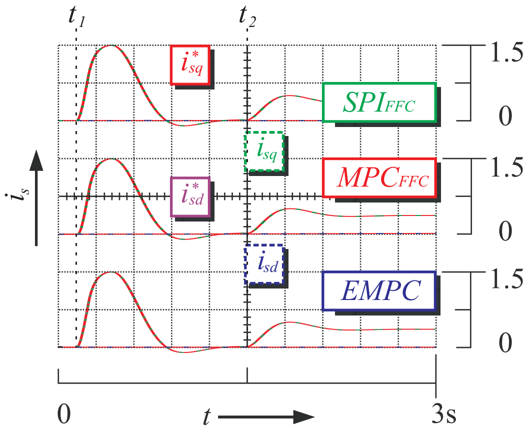


Figure 6.64: PMSM machine currents and their references in synchronous coordinates under load conditions using different controllers

a) SPI$_{dq}$ controller with FFC. b) SMPC$_{dq}$ controller with FFC. c) EMPC$_{dq}$ controller

Unfortunately, the cross-coupling effect between the machine currents in the above introduced figures was not sufficient to evaluate the performance of the different controllers.

Figure 6.65 shows again the same results for the time period when the machine speed reached its reference and the current i_{sq} starts going down to negative. The XY-Plot is drawn to show the cross-coupling effect between the machine currents i_{sd} , i_{sq} .

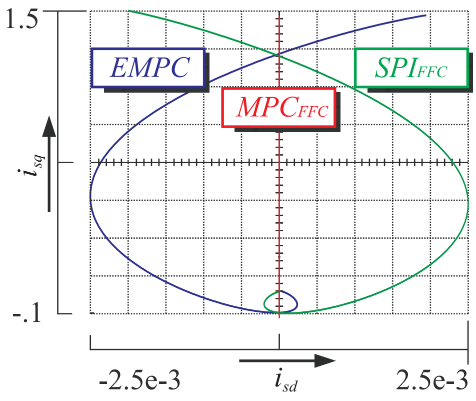


Figure 6.65: PMSM current response in XY-coordinates, comparative analysis using different current controllers, when the nominal speed is reached

With perfect compensation of the cross-coupling effect between the machine currents, the XY-plot shows a straight line (MPC$_{FFC}$). Increasing the coupling effect, the line starts to take the curve form (EMPC, SPI$_{FFC}$). All these simulation results are to be further verified by experimental results, which is out of the scope of this work.

6.9.5 MIMO-MPC (Speed, Current) Controller

As the rotor flux $\hat{\psi}_m$ of the PMSM machine results from the permanent magnet in the rotor, and therefore it is constant $\hat{\psi}_m = \text{const.}$, the extension of the state-vector of the PMSM machine model to include the mechanical speed can be done in a simpler way than in the induction machine model. Referring to the time-normalized mechanical equations (6.74), the speed as new state variable could be integrated into the MIMO controller in analogous way.

$$\frac{d\omega_{em}}{d\tau}=\frac{\hat{\psi}_m}{\tau_m}i_{sq} \tag{6.74}$$

Following the discretization method in section 5.3, the discrete-time linear time-invariant speed model takes the following formulation:

$$\omega_{em_{k+1}}=\omega_{em_k}+\frac{T_0\hat{\psi}_m}{\tau_m}i_{sq\,k} \tag{6.75}$$

where $\frac{\omega_{em_{k+1}}-\omega_{em_k}}{T_0}=\frac{\hat{\psi}_m}{\tau_m}i_{sq_k}$, T_0 is the normalized sampling time.

Note the linear relationship between the mechanical speed ω_{em} and the machine current i_{sq} . This new term (ω_{em}) is to be inserted in the equation (6.72) accordingly:

$$\begin{bmatrix} i_{sd} \\ i_{sq} \\ x_d^d \\ x_q^d \\ \omega_{em} \end{bmatrix}_{k+1}=\begin{bmatrix} 1-\frac{T_0}{\tau_s} & 0 & T_0 & 0 & 0 \\ 0 & 1-\frac{T_0}{\tau_s} & 0 & T_0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & \frac{T_0\hat{\psi}_{m_{dq}}}{\tau_m} & 0 & 0 & 1 \end{bmatrix}\cdot\begin{bmatrix} i_{sd} \\ i_{sq} \\ x_d^d \\ x_q^d \\ \omega_{em} \end{bmatrix}_k+\begin{bmatrix} \frac{T_0}{l_s} & 0 \\ 0 & \frac{T_0}{l_s} \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \end{bmatrix}\cdot\begin{bmatrix} u_{sd} \\ u_{sq} \end{bmatrix}_k \tag{6.76}$$

From the equation (6.70), the compensation terms in form of current are:

$$\mathbf{x}^{\mathrm{d}}=\begin{bmatrix} x_d^d \\ x_q^d \end{bmatrix}=-\frac{1}{l_s}\begin{bmatrix} u_{id} \\ u_{iq} \end{bmatrix}=-\frac{1}{l_s}\left(l_s\begin{bmatrix} 0 & -1 \\ 1 & 0 \end{bmatrix}\cdot\begin{bmatrix} i_{sd} \\ i_{sq} \end{bmatrix}+\begin{bmatrix} 0 \\ \hat{\psi}_{m_{dq}} \end{bmatrix}\right)\omega_{em} \tag{6.77}$$

This equation can be reformulated again to get the mechanical speed term separated:

$$\mathbf{x}^{\mathrm{d}}=\begin{bmatrix} x_d^d \\ x_q^d \end{bmatrix}=-\begin{bmatrix} 0 & -1 \\ 1 & 0 \end{bmatrix}\cdot\begin{bmatrix} i_{s_d} \\ i_{s_q} \end{bmatrix}\omega_{em}\overbrace{-\frac{1}{l_s}\begin{bmatrix} 0 \\ \psi_{m_{dq}} \end{bmatrix}}^{const.}\omega_{em} \tag{6.78}$$

As the rotor flux is constant $\hat{\psi}_m=\text{const.}$, the terms in $\mathbf{x}^{\mathrm{d}}$ can be reduced by integrating the constant part of the equation (6.78) into the equation (6.76).

$$\begin{bmatrix} i_{sd} \\ i_{sq} \\ x_d^d \\ x_q^d \\ \omega_{em} \end{bmatrix}_{k+1} = \begin{bmatrix} 1-\dfrac{T_0}{\tau_s} & 0 & T_0 & 0 & 0 \\ 0 & 1-\dfrac{T_0}{\tau_s} & 0 & T_0 & \dfrac{T_0\widehat{\psi}_{m_{dq}}}{\tau_m} \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & \dfrac{T_0\hat{\psi}_{m_{dq}}}{\tau_m} & 0 & 0 & 1 \end{bmatrix} . \begin{bmatrix} i_{sd} \\ i_{sq} \\ x_d^d \\ x_q^d \\ \omega_{em} \end{bmatrix}_k + \begin{bmatrix} \dfrac{T_0}{l_s} & 0 \\ 0 & \dfrac{T_0}{l_s} \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \end{bmatrix} . \begin{bmatrix} u_{sd} \\ u_{sq} \end{bmatrix}_k \tag{6.79}$$

Hence, the new terms that have to be compensated in this case are:

$$\mathbf{x}^{\mathrm{d}} = \begin{bmatrix} x_d^d \\ x_q^d \end{bmatrix} = \begin{bmatrix} i_{sq} \\ -i_{sd} \end{bmatrix} \omega_{em} \tag{6.80}$$

In similar way, these terms ($\mathbf{x}^{\mathrm{d}}$) are to be evaluated at each sampling time and feed to the controller as new state variables in the equation (6.79).

Finally, the introduced MIMO (speed, currents) machine model is ready to be used in designing the MPC controller with expectation of difficulty in tuning the controller and of a high number of the generated regions of the Exp-MPC controller.

6.10 Conclusions

This chapter addressed mainly the explicit solution of model-based predictive control from design, simulation and implementation point of views. A comparative study was presented between different approaches in the way of representing the system model, dealing with tracking and delay issues, and introducing the non-linear parts. The nonlinearity was considered in this chapter as it is, without any kind of linearization, and it was introduced to the extended system model, in state space representation, in the same way like introducing the reference signal for tracking problems. Assuming that these terms will not be changed within the sampling period and along the prediction horizon, they are to be estimated/ calculated in each sampling period and introduced to controller as new state variables. A comparative analysis concerning the compensation procedure of these terms was introduced between the proposed methods with MPC controllers and other ones in conventional scalar and complex PI controllers. The validity of the internal compensation is also verified and used for direct control of a three-level neutral-point diode-clamped inverter (NPC), where the external compensation with discrete inputs is not possible any more. Finally, the neutral point balancing mechanism for the direct control of three-level NPC inverter was covered. The experimental results for the induction machine and the simulation ones for the PMSM machine prove the efficiency of this method in transient and steady state and its validity for all the operating points. Some suggestions, regarding extending the state-vector of the PMSM machine model to contain the mechanical speed in the MIMO (speed, currents) controller, were highlighted for the future work in this direction.

7 Polynomial Approximation

7.1 Overview

In the last five years, a significant and increasing attention has been paid to use the explicit solution of model-based predictive control (Exp-MPC) in a wide range of electric drive applications, where a lot of literatures was published to reduce the online computation complexity of the MPC to make it more feasible for this field of applications [74], [75], [76]; For example, using the binary search tree (BST) strategy [69] to reduce the evaluation time plays a significant role in that direction. Nevertheless, there are still some limitations like memory size and the computation capability of the control set-up, which make Exp-MPC until now not the right choice and the right alternative for different kinds of controllers for AC drive applications [82]. This work proposes the polynomial approximation of the Exp-MPC as a new contribution to reduce the implementation complexity of Exp-MPC. Solving the MPC optimization problem using multi-parametric programming approach [65] divides the feasible state space of the control problem into a number of regions. Inside each region, there is one valid optimal control law, defined as an affine function of the state variables. The final solution is a look-up table of the region coordinates and the optimal control laws. The complexity of implementing the Exp-MPC in drive applications results in a higher number of generated regions. This increases the required memory space to store these regions and the coefficients of the associated optimal control laws, even when the solution is implemented using an efficient algorithm like the binary search tree (BST) [69]. The proposed method intends to replace the optimal linear/ affine control laws, defined over several regions of the feasible state space, with one polynomial. The polynomial will be multivariate, with higher order, and defined over a combination of feasible regions. Using the polynomial approximation, the necessary memory space, to store region coordinates and the coefficients of optimal control laws, is reduced significantly. Although the accuracy of the optimal controller is reduced the polynomial approximation, it makes the implementation more realistic in cases where original Exp-MPC is impossible or very difficult to implement; provided the stability and feasibility of the problem is still guaranteed.

7.2 Problem Description

The explicit solution of model-based predictive control becomes more complex when detailed motor models are considered while designing Exp-MPC controller. This complexity is reflected in the number of generated regions. Increase in the number of regions requires more memory space to store the explicit solution of MPC as well as the evaluation time of the proper affine control laws. These drawbacks limit the implementation of the Exp-MPC in fast sampling rate applications [70]. The polynomial approximation intends to reformulate a combination of the optimal affine control laws in several regions with only one multivariate polynomial of higher degree and covers those regions. This approximation will definitely reduce the accuracy of the optimal controller, but on the other hand, it will make the implementation of the Exp-MPC in much complex cases more simple and feasible; provided that the stability and the feasibility of the problem are still guaranteed. Complexity reduction of the Exp-MPC using the polynomial approximation, without considering the cross-product[9] terms, was introduced in the literature [119], [122], [123], [124]. The authors in these contributions use the sum-of-squares and the Polya's theorem[10] to find an approximate polynomial inside a stabilization set defined as Polytope. Building the stabilization set depends on the principle of existence of more than one Lyapunov function assure the stability of the optimization problem. On the other hand and due to the expensive symbolic computation of the extended Polya's polynomial, this formulation is just suitable for small problems. It works well for small problem size (up to 100 regions in 3 dimensions) of linear or infinite norm and with guarantee for stability and feasibility of the problem.

The novelty of the introduced work in this chapter is in introducing the multivariate polynomial approximation using the cross-product terms for drives applications. The simulation results carried on vector controlled motor drive prove the feasibility and the stability of the machine problem with the approximated control laws. This work reformulates the explicit solution of model-based predictive control using least-square curve fitting strategy.

7.3 Polynomial Formulation

Since the implementation of the explicit solution of model-based predictive control Exp-MPC is always limited by the number of the feasible regions, this work introduces curve fitting methods as effective ways to approximate a combination of several optimal affine control laws , equation (7.1), with only one approximate polynomial of higher order and covers those regions, equations (7.2), (7.4).

$$\mathbf{U}_t^* = \mathbf{F}^i \,.\, \mathbf{x}_t + \mathbf{G}^i \tag{7.1}$$

where $\mathbf{U}_t^*$ is the optimal control law at time instance t, the matrices $\mathbf{F}^i, \mathbf{G}^i$ contain the affine control law coefficients, associated with region i, $\mathbf{x}_t$ is a vector of the measured/ estimated state variables, which define the region i.

The used multivariate polynomial expression [125] takes the following form with two different formulations (7.2), (7.4) as given below:

- Considering the cross-product terms:

$$\tilde{\mathbf{U}}(x) = \sum C_{i_1 i_2 \ldots i_{n_x}} \cdot \prod_{j=1}^{n_x} x_j^{i_j} \tag{7.2}$$

where $\tilde{\mathbf{U}}(x)$ is the approximated control law, C are the polynomial coefficients, n_x is number of the polynomial variables (x), i_j is the degree of the polynomial variables, and P_o is

[9] The terminology "Cross products" is used in this chapter to refer to multiplications between the polynomial variables of various powers.

[10] Polya's theorem [124]: If a homogenous Polynomial $\tilde{\mathbf{U}}(C, x)$ is positive for all possible values of the polynomial variables (x) belong to the N-dimensional polytope, all the polynomial coefficients (C) are positive for a sufficiently large polynomial degree (P_O).

the polynomial degree. The multivariate polynomial in this expression can be introduced as a sum of terms, where each term is a multiplication of real coefficients with the polynomial variables of various degrees; e.g.:

$$P\left(x_1,\, x_2,\, x_3\right) \;=\; C_{000} + C_{100}.x_1 + C_{010}.x_2 + C_{001}.x_3 + C_{200}.x_1^{\;2} + C_{210}.x_1^{\;2}.x_2 + \ldots \qquad (7.3)$$

- Avoiding the cross-product terms:

$$\tilde{\mathbf{U}}(x) = \sum_{i=0}^{P_O}\sum_{j=1}^{n_x}\left[C_i\right]_j .\, x_j^i \qquad (7.4)$$

where $\tilde{\mathbf{U}}(x)$ is the approximated control law, C are the polynomial coefficients, P_o is the polynomial order/ degree. Avoiding the cross-product terms means avoiding the cross multiplication between the polynomial variables, and each term contains only one of the polynomial variables of different degree and multiplied with a real coefficient; e.g.:

$$P\left(x_1, x_2\right) = C_{00} + C_{10}.x_1 + C_{01}.x_2 + C_{20}.x_1^{\;2} + C_{02}.x_2^{\;2} + C_{30}.x_1^{\;3} + C_{03}.x_2^{\;3} + \ldots \qquad (7.5)$$

Having the cross products in the polynomial expression, equation (7.2), increases the accuracy of the approximation and delivers much better results than without, but it will increase the total number of polynomial coefficients (N_c). More coefficients mean more memory space is required to store these coefficients. In some cases this number could exceed the number of coefficients using BST of the Exp-MPC. However a compromise should be settled.

7.3.1 Requirements and The Pre-Processing Procedure

All steps introduced in section 5.3 to get the explicit solution of the MPC optimization problem are to be considered here before going further with the polynomial approximation, Figure 7.1.

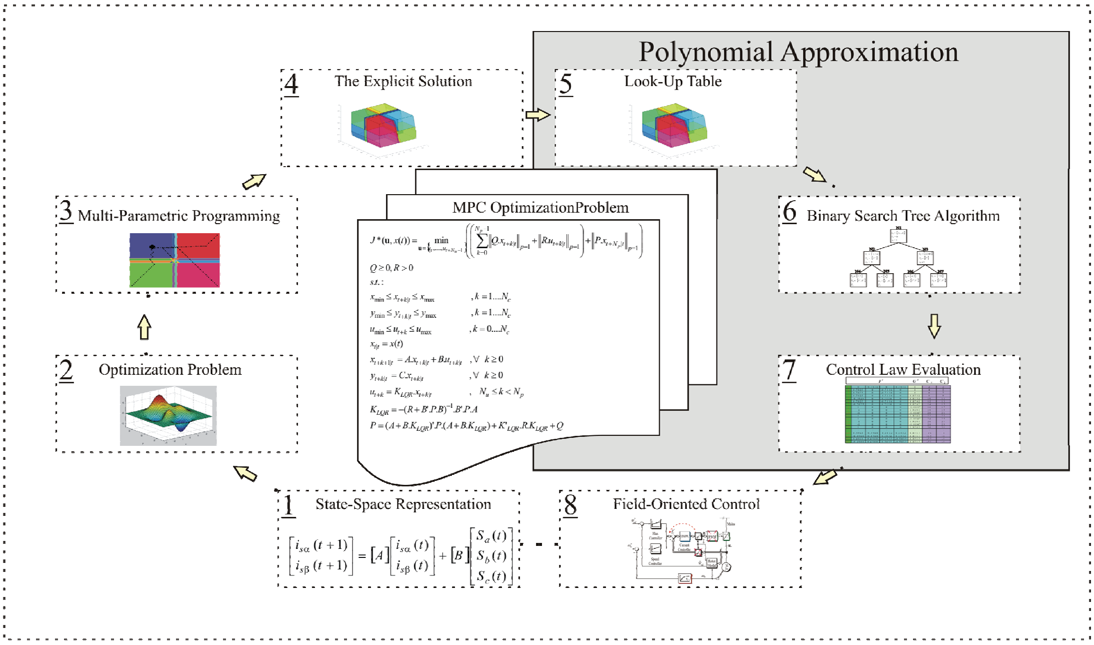


Figure 7.1: Polynomial approximation of the explicit solution of MPC, design and evaluation procedure

The preparation stage starts by generating the explicit solution from the MPC controller. The Exp-MPC represented as a structure should be calculated first and provided. To fulfill this requirement, the freely available Multi-Parametric Toolbox (MPT) from ETH-Zurich [77], [80] should be already installed with Matlab program from The Mathworks^{TM}. The Exp-MPC structure contains the coordinates of all regions and the associated control laws in its fields. The curve fitting procedure starts by selecting some fitting points of each region and evaluating the proper control law. Degree of computational complexity and the off-line preparation time depend basically on the number of fitting points and their locations in each region. For example, taking only the vertices and the Cheby-Ball center[11] of each region reduces the required time to extract these points significantly, but they will not suffice for a fine fitting. Whereas taking a fixed number of mesh points from each region will improve the fitting, but on contrary will increase the complexity of the preparation procedure. As it is desired to obtain a good approximation of the optimal control laws over the regions, both vertices and mesh points are used in the proposed method. The combination procedure of the regions associated with one polynomial are not covered in this work, where it is mathematically not possible to combine the regions in a way to get a convex one, which assures the continuity of the resulting polynomial. As an elementary solution for this problem, it can be tried to divide the state space with respect to the main state variables before solving the optimization problem. An initial solution for the combination procedure associated with some remarks for future research in this direction is introduced in section 7.6.

7.3.2 Least-Square Curve Fitting

This work reformulates the explicit solution of model-based predictive control using least-square curve fitting strategy. Generally, the goal of least-square curve fitting method is to keep a minimum distance (fitting error **E**) between the original curve and the approximated one. This problem is usually solved as a linear optimization problem aimed at finding the polynomial coefficients '**C**' which minimize the approximation error **E**.

$$\min_{C} \mathbf{E} = \min_{C} \sum_{n=1}^{n_p . n_r} [\mathbf{U}_n(x) - \tilde{\mathbf{U}}_n(x)]^2 \tag{7.6}$$

where C are the optimal polynomial coefficients, n_p is the number of the selected points for the fitting procedure in each region, n_r the total number of the regions, resulted by the explicit solution of the MPC problem, $\mathbf{U}_n(x)$ are the calculated control inputs according to Exp-MPC, and $\tilde{\mathbf{U}}(x)$ is the approximated control law.

7.3.3 Complexity Analysis

This section introduces the complexity reduction of the online evaluation procedure for both the Exp-MPC using binary search tree (BST) method and the polynomial one:

1. The online evaluation procedure of Exp-MPC using BST requires storing different matrices:

   - Branching Matrix (**BM**): This matrix stores the coefficients of all linear inequality-equations to be evaluated later in the branching procedure over the tree- levels. Its size can be roughly approximated as [69], [70], equation (7.7).

[11] Cheby-Ball center is the center of the largest ball inscribed inside the considered region.

$$\begin{aligned}&\mathrm{size}(\mathbf{BM})=\mathrm{n_r}.(\mathrm{n_x}+3)+\mathrm{n_c}.\mathrm{n_u}.(\mathrm{n_x}+1)\\&:\mathrm{n_c}\approx \mathrm{n_r}/4\end{aligned} \tag{7.7}$$

where n_r is the number of the Exp-MPC regions, n_c is number of the unique control laws, and n_x, n_u denote the number of state and the input variables, respectively.

- Control law matrices (**F**, **G**): Two more matrices (**F**, **G**) are to be stored, which contain the coefficients of the affine control laws. Their sizes can be given by the following equation:

$$\mathrm{size}(\mathbf{F})+\mathrm{size}(\mathbf{G})=\mathrm{n_r}.\mathrm{n_u}.(\mathrm{n_x}+1) \tag{7.8}$$

Thus the total number of coefficients in the Exp-MPC case using the BST is:

$$\mathrm{U_L}=\mathrm{size}(\mathbf{BM})+\mathrm{size}(\mathbf{F})+\mathrm{size}(\mathbf{G}) \tag{7.9}$$

This number could be considered as upper limit (U_L) for the polynomial strategy and should not be violated to say that the proposed strategy is effective enough and delivers a reduction in number of coefficients over the Exp-MPC solution.

2. Number of the coefficients (N_c) to be stored in polynomial case is:

- Without cross-product terms:

$$\begin{aligned}&\mathrm{N_c}=\mathrm{n_P}.\mathrm{n_u}.(\mathrm{P_O}.\mathrm{n_x}+1)+\ldots\\&\ldots\text{Coefficients of }\mathrm{n_P}\text{ regions' inequalities if }\mathrm{n_P}<\mathrm{n_r}\end{aligned} \tag{7.10}$$

- With cross-product terms:

$$\begin{aligned}&\mathrm{N_c}=\mathrm{n_P}.\mathrm{n_u}.\left(1+\sum_{i=1}^{i=\mathrm{P_O}}\frac{(i+(\mathrm{n_x}-1))!}{i!.(\mathrm{n_x}-1)!}\right)+\ldots\\&\ldots\text{Coefficients of }\mathrm{n_P}\text{ regions' inequalities if }\mathrm{n_P}<\mathrm{n_r}\end{aligned} \tag{7.11}$$

where n_u is the number of the control inputs, n_P is the number of the polynomials, P_O is the polynomial's order, n_r is the total number of regions, and n_x, n_u denote the number of state and the input variables, respectively.

This number (N_c) is increased proportionally with number of polynomials (n_P) and number of control inputs (n_u). To cover more regions with a lower number of polynomials, the polynomials' orders have to be increased ($P_O > 1$) gradually. But increase of polynomial order (P_O) will increase the evaluation time, especially when cross-product terms are considered with different power multiplications which makes the evaluation in some cases prohibitive. Hence a compromise should be found here between the above two approaches. The first approximation way without cross products was introduced in [75]. It easy to implement and delivers lower number of coefficients to be stored, but it does not lead to a good approximation. Employing the cross products in the approximation delivers much better results, but on contrary, it will increase the number of coefficients as well as the complexity of the evaluation procedure. In this work the polynomial expression (7.2) is used up to some degree, where the upper-limit of Exp-MPC using the BST, equation (7.9), was not violated. This compromise was considered in the Example-4, described in next section for an induction motor, where the cross-product terms are allowed up to the 3rd order, such that the upper-limit of Exp-MPC using BST is not violated.

7.4 Examples with Different Complexity

To demonstrate the concept of polynomial approximation, the simulation results are presented in this section consequently. It is applied to a simple example first and then it is extended to more complicated examples with different norms, having a large number of states and regions.

7.4.1 Example-1

The following equation represents a system model [119] with two dynamics, one state and one input variable:

$$x_{k+1} = \begin{cases} \frac{4}{5} x_k + 2u_k \,, if \quad x > 0 \\ \\ -\frac{6}{5} x_k + u_k \,, if \quad x \le 0 \end{cases} \tag{7.12}$$

With the associated system constraints: $\mathbf{x}_k \in [-4,4], \mathbf{u}_k \in [-1,1]$.

The explicit solution of MPC for this problem with a linear norm and an infinite prediction horizon delivers five regions $R_1 \ldots R_5$ in one dimension. For each region, one optimal control law u is valid; as shown in Figure 7.2.

By considering the extreme points only of each region (vertexes) to do the curve fitting using least-square method, the resulting curve will oscillate far away from the line segments and then comes back to match at the extreme points. This means that the approximated curve will deliver good results just for the extreme points but not in between. Thus an increase in number of fitting points in each region will improve the fitting results. In this example, 10 points (marked with ‘x’ in Figure 7.3a) from each region were considered to do the fitting. Also it is advisable to test the resulting polynomial with much more test points, as shows the green-continuous line in Figure 7.3b. For a univariate polynomial of 7th order, it is necessary to store eight coefficients instead of storing five regions plus 10 coefficients for the Exp-MPC control laws ($\mathrm{U_L} \approx 33$).

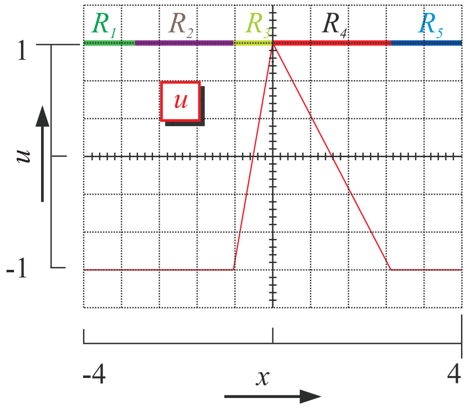


Figure 7.2: Controller regions R_i with the associated control laws u_i

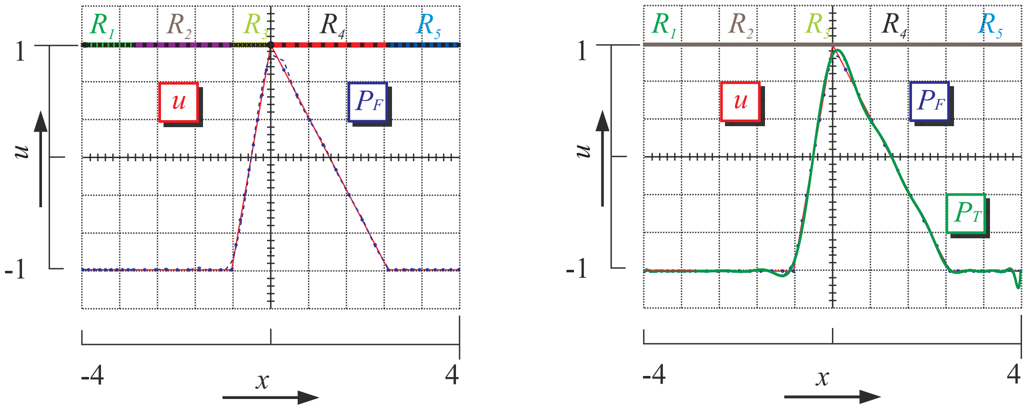


Figure 7.3: Controller regions R_i with the associated control laws u_i

a) The fitted polynomial curve P_F (blue curve). b) Testing the polynomial with more points P_T (green curve).

7.4.2 Example-2

To introduce to the multidimensional control structure of Exp-MPC, consider the following system model (7.13) with one dynamic, two state variables, and one input:

$$\begin{bmatrix} x_1 \\ x_2 \end{bmatrix}_{k+1} = \begin{bmatrix} 0.7326 & -0.0861 \\ 0.1722 & 0.9909 \end{bmatrix} \cdot \begin{bmatrix} x_1 \\ x_2 \end{bmatrix}_k + \begin{bmatrix} 0.0609 \\ 0.0064 \end{bmatrix} \cdot [u]_k$$

$$\begin{bmatrix} y_1 \\ y_2 \end{bmatrix}_k = \begin{bmatrix} x_1 \\ x_2 \end{bmatrix}_k \tag{7.13}$$

System variables subject to the following constraints: $\mathbf{x}_k \in [-1.5, 1.5], \mathbf{u}_k \in [-2, 2]$.

The explicit solution of the MPC controller for this problem with quadratic norm and 2 steps prediction horizon delivers five regions $R_1 \ldots R_5$ in two dimensions, Figure 7.4, where there is one valid optimal control law u in each region. Plotting the control input for two state variables in one figure, as in Figure 7.5, will not help to observe the fitting procedure; instead of that the evaluation procedure has been done in Simulink. The evaluation results of both Exp-MPC and polynomial controllers are shown in Figure 7.6, where the controllers have to find the optimal control law u to steer the state signals x_1, x_2 to the origin. By assigning one bivariate polynomial in each region, and if the fitting procedure works correctly, then both methods should deliver exactly the same results. This means, both controllers deliver same control input and hence the state and control signals will match, respectively, Figure 7.4. Figure 7.5 shows again the states and the control input for both Exp-MPC and polynomial method by using one bivariate polynomial for all the regions of 1st order; here the approximated control inputs are distinguished very clearly from the Exp-MPC.

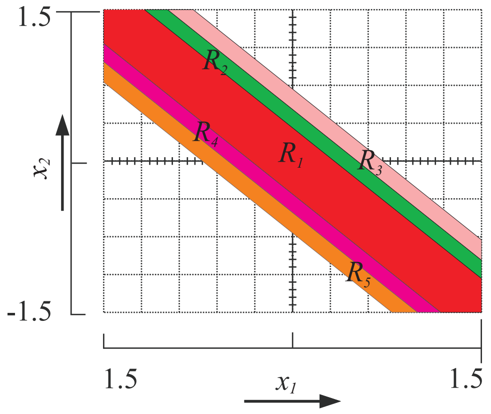


Figure 7.4: State space regional division

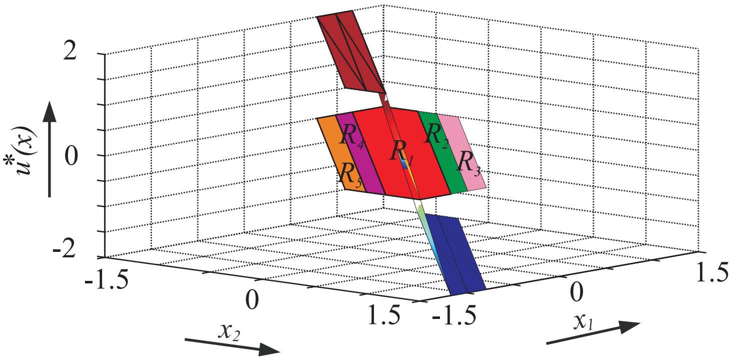


Figure 7.5: Controller regions R_i with the associated control laws u_i

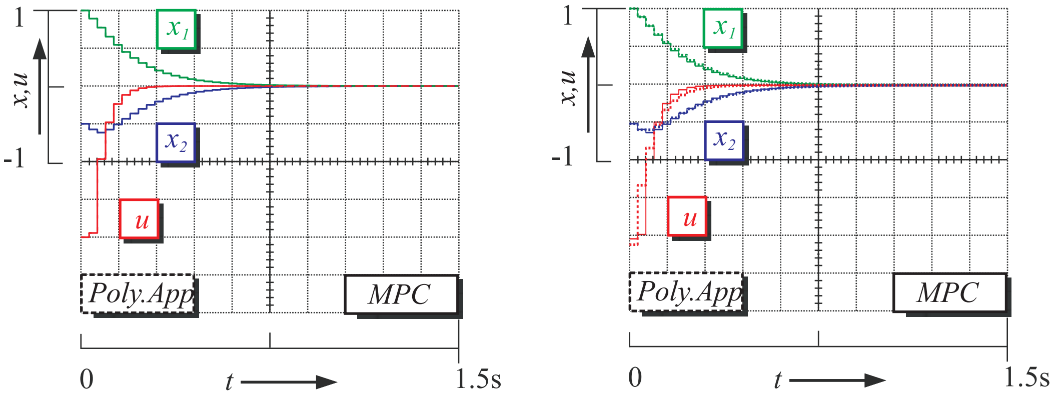


Figure 7.6: Exp-MPC and polynomial controllers' response

a) 1th order Polynomial in each region b) 1th order Polynomial for all the regions

7.4.3 Example-3

Machine example: 6 state variables, two inputs and 44 regions:

This is a more complex and complete example to control the currents of an induction machine in synchronous coordinates, where the approximation method still works effectively. The induction machine currents are controlled using one MIMO-MPC controller instead of 2 SISO-PI controllers in the Filed-Oriented Control scheme, Figure 7.7. The compensation procedure of the induced-voltages was done outside the MPC controller in a passive way as explained in section 6.5.3.

To consider the delay compensation and the free-offset tracking of the machine currents, as introduced in equation (6.5) of section 6.3, the original state variables (i_{sd} , i_{sq}) in the equation (7.14) are to be augmented to include the control inputs (u_{sd} , u_{sq}) from previous time step and the reference signals (i_{sd}^*, i_{sq}^*).

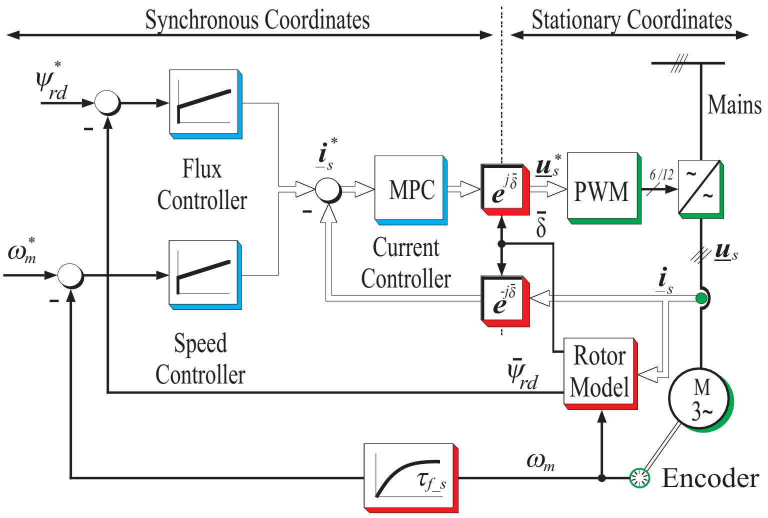


Figure 7.7: Field-oriented control structure using a MPC$_{\text{dq}}$ current controller

It results in 6 state variables in the equation (7.15).

$$\begin{bmatrix} i_{sd} \\ i_{sq} \end{bmatrix}_{k+1} = \begin{bmatrix} 1-\frac{T_0}{\tau_\sigma^{'}} & 0 \\ 0 & 1-\frac{T_0}{\tau_\sigma^{'}} \end{bmatrix} \cdot \begin{bmatrix} i_{sd} \\ i_{sq} \end{bmatrix}_k + \begin{bmatrix} \frac{T_0}{l_\sigma} & 0 \\ 0 & \frac{T_0}{l_\sigma} \end{bmatrix} \cdot \begin{bmatrix} u_{sd} \\ u_{sq} \end{bmatrix}_k \qquad (7.14)$$

where T_0 is the normalized sampling time. All machine quantities are normalized according to the normalization procedure in the Appendix-B.

$$\begin{bmatrix} i_{sd\,k+1} \\ i_{sq\,k+1} \\ u_{sd\,k} \\ u_{sq\,k} \\ i^*_{sd\,k+1} \\ i^*_{sq\,k+1} \end{bmatrix} = \left[\tilde{\mathbf{A}}\right] \cdot \begin{bmatrix} i_{sd\,k} \\ i_{sq\,k} \\ u_{sd\,k-1} \\ u_{sq\,k-1} \\ i^*_{sd\,k} \\ i^*_{sq\,k} \end{bmatrix} + \left[\tilde{\mathbf{B}}\right] \cdot \begin{bmatrix} \Delta u_{sd} \\ \Delta u_{sq} \end{bmatrix}_k \qquad (7.15)$$

The explicit solution of MPC for this problem with quadratic norm and only 1 step prediction horizon will deliver six-dimensional control structure defined over 44 regions. Hence the coordinates of the six-dimensional 44 regions (the matrix **BM**) plus the coefficients of the associated control laws (the matrixes **F**, **G**) are to be stored here.

$$\begin{aligned} U_L &= \underbrace{\text{size}(\mathbf{BM})}_{536} + \underbrace{\text{size}(\mathbf{F}) + \text{size}(\mathbf{G})}_{616} \\ &= 1152 \ \ \text{Coefficients}(\text{double}) \end{aligned} \tag{7.16}$$

This number has to be considered as an upper limit for the polynomial strategy and should not be violated to say that this strategy is effective enough and delivers a reduction in the Exp-MPC solution. The pre-processing procedure for this case is more complex than the former ones, where the fitting points are taken from a six-dimensional mesh grid constructed for each region plus the extreme points of that region. Assigning 44 polynomials of the 1st order for 44 regions should deliver exactly the same results for both the Exp-MPC and polynomial controller; this is shown in Figure 7.8a, Figure 7.8b. In these figures the machine currents were observed in different coordinate systems by applying a step in the nominal speed and nominal torque, respectively. The nominal torque step was applied after reaching the machine its nominal speed.

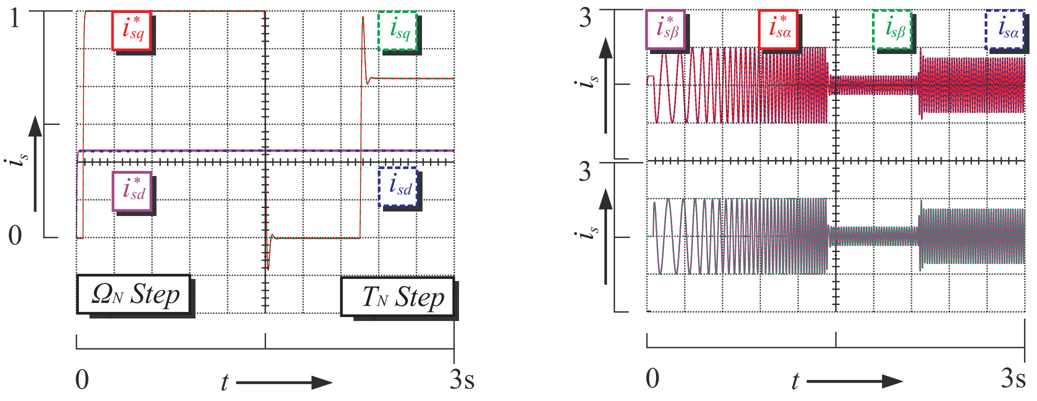


Figure 7.8: IM currents using the polynomial controller of the 1st order in each region

a) In synchronous coordinates b) In stationary coordinates

It can be seen from Figure 7.9 that it is still difficult to distinguish between both Exp-MPC and polynomial controller responses even if the control laws of four regions are replaced with one polynomial.

Assigning one polynomial of the 1st order, to cover more and more regions, will definitely increase the fitting error as shown in Figure 7.10a; therefore the polynomial's order has to be increased slightly, the corresponding results are shown in Figure 7.10b. Increase in the polynomial order will increase the number of coefficients as well as the complexity of the evaluation procedure. Figure 7.10b shows the results of using one polynomial of 3rd order to cover the 44 regions of the Exp-MPC. Here it is required to store 168 polynomial coefficients instead of storing 1152 coefficients for the region coordinates and the optimal control laws of Exp-MPC.

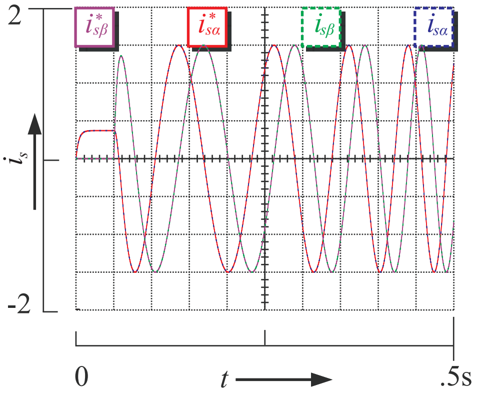


Figure 7.9: IM currents ($i_{s\alpha}$, $i_{s\beta}$) using one polynomial of the 1st order for each group of 4 regions

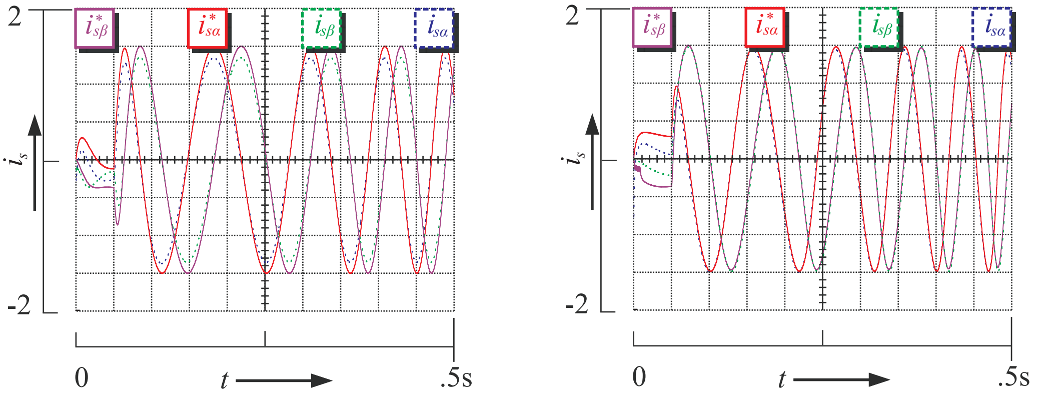


Figure 7.10: IM currents ($i_{s\alpha}$, $i_{s\beta}$) using polynomial controllers of different orders

a) 1st order polynomial for each group of 22 regions b) 3rd order polynomial for each group of 22 regions

7.4.4 Example-4

Machine example: 6 state variables, two inputs and 410 regions:

The above introduced problem in Example-3 is reformulated again with more constraints on the state variables and higher prediction horizon. Increasing the prediction horizon of the control problem and using a linear norm instead of the quadratic one will increase the number of regions for the explicit solution of MPC. The explicit solution of MPC for this problem with a linear norm and 4 steps prediction horizon delivers a 6-dimensional control structure defined over 410 regions. Hence the coordinates of the six-dimensional 410 regions (the matrix **BM**) plus the coefficients of the associated control laws (the matrixes **F**, **G**) are to be stored here.

$$U_L = \underbrace{\text{size}(\mathbf{BM})}_{5125} + \underbrace{\text{size}(\mathbf{F}) + \text{size}(\mathbf{G})}_{5740}$$
$$= 10865 \quad \text{Coefficients (double)} \tag{7.17}$$

This number is the upper limit for the polynomial approximation and should not be violated. The pre-processing procedure here requires extracting fitting points (the extreme and mesh-grid points) from the resulting 6-dimentional regions.

The expecting reduction in the online complexity of Exp-MPC depending on the number and the order of polynomials with and without cross products is summarized in the Table 7.I to be compared to the Exp-MPC with binary search tree strategy (BST).

The last row of the table means, for example, that if it is possible to build only one polynomial (with cross products) of 3rd order and covers all the regions, then the number of coefficients to be stored during the implementation will be 168 coefficients instead of 10865 required for the Exp-MPC and there will be no need to store any regions.

Table 7.I: Number of coefficients for the approximate polynomial, with and without cross products

Exp-MPC with polynomial approximation				
n_p	**P_o**		**N_c without Cross Products**	**N_c with Cross Products**
410	1		5740	5740
5	1		70	70
::	::		::	::
5	3		190	840
2	1		28	28
::	::		::	::
2	3		76	336
1	1		14	14
::	::		::	::
1	3		38	168

Figure 7.11 shows the machine currents and their references in synchronous and stationary reference frames, by applying a nominal speed step. The linear control laws in each group of 10 regions are approximated and replaced by one polynomial of 3rd order. Here 1680 coefficients are to be stored instead of storing 5740 coefficients of the linear control laws in

(**F**, **G**) of Exp-MPC. Furthermore the coordinates of resulting groups are to be stored (i.e.: 125 coefficients instead of 5125 BM coefficients).

Using only one multivariate polynomial (with cross products) of 3rd order and covers all the regions requires storing only 168 coefficients, instead of 10865 coefficients (U_L) required for the Exp-MPC. Furthermore, with the approximate polynomial there is no need to store the coordinates of any region. The evaluation procedure of the resulting polynomial exhibits a steady state error, which it is not acceptable in practice. Figure 7.12 shows the machine currents and their references in synchronous reference frame, when the machine accelerates from zero speed to the nominal speed and under different load conditions. Noting that, the error stays constant for different speed levels even by keeping applying the nominal torque; therefore it could be compensated later at the output of the controller. In Figure 7.12 the machine was driven under different speed conditions with and without applying the nominal torque. The lower part of the figure shows the speed steps, while the upper part shows the machine currents (i_{sd} , i_{sq}) with their references. The resulting constant error was tuned and compensated in the simulation manually.

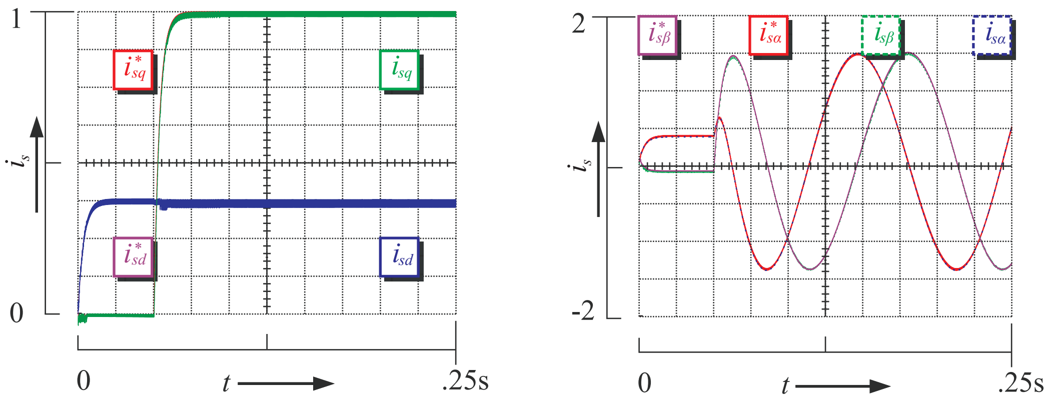


Figure 7.11: IM currents response using one polynomial of the 3rd order for each group of 41 regions, by applying nominal speed step

a) In synchronous coordinates b) In stationary coordinates

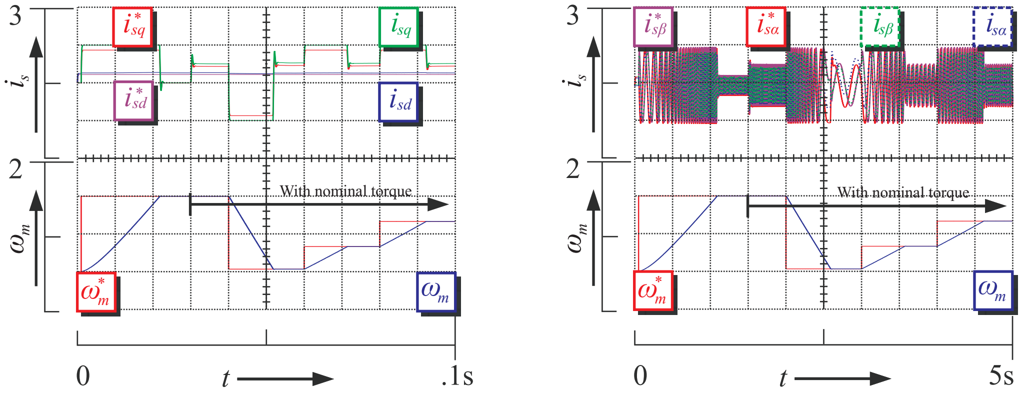


Figure 7.12: IM currents using one polynomial of the 3rd order for all the 410 regions

a) In synchronous coordinates system b) In stationary coordinates system

In Figure 7.13 a full current step was applied in the torque producing current (i_{sq} = 0→1), while keeping the machine fully magnetized (i_{sd} = 0.35). It shows the dynamic behavior of the machine currents using the polynomial controller against the MPC one. Figure 7.13 proves that the polynomial approximation method of the Exp-MPC solution exhibits a “very good” and an “acceptable” dynamic response as compared to the original Exp-MPC; having the stability and feasibility of the problem are still assured.

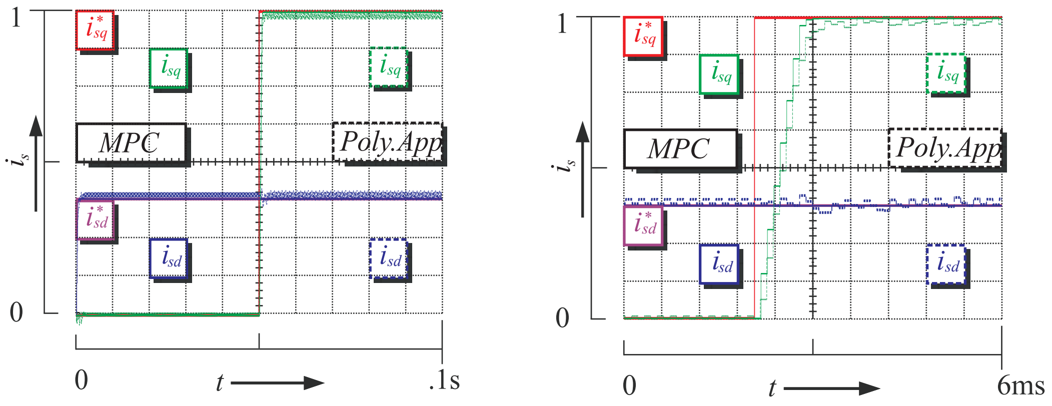


Figure 7.13: IM current response using an Exp- MPC controller and one polynomial of the 3rd order for all the 410 regions with different time windows

7.5 Conclusions

The multivariate polynomial approximation method using the cross-product terms was presented in this chapter as an effective way to accelerate the implementation procedure of the explicit solution of model-based predictive control with a very close dynamic performance to the Exp-MPC. The number of the coefficients to be stored during the evaluation procedure and consequently the memory footprints were reduced significantly. A comparison with the binary search tree algorithm (BST) was also introduced. BST could be considered as fastest and simplest method for implementing the Exp-MPC. This work reformulates the explicit solution of model-based predictive control using least-square curve fitting strategy, and does include neither experimental results nor the combination procedure of the regions which are covered with the resulting polynomials. The combination procedure could be found in section 7.6, with some suggestions and improvements on the presented procedure.

7.6 Future Work on the Polynomial Approximation

There are some points to be considered if the research is to be continued in this direction:

- Looking for another fitting strategy such as weighted least square or cubic Spline fitting method, which could be more effective, than the proposed one for the following reasons:
    - In the proposed method, all the fitting points (mesh points, vertexes plus the Cheby-ball center of the regions) have the same priority in the fitting procedure, which may not be completely true. Considering the mesh points plus the vertexes will lead to better results but at the same time, this will pull the fitting cure away from its optimal location at the edges as long as the vertexes are not considered with higher priority; as illustrated in Figure 7.3.

- Figure 7.3b in the first example exhibits oscillation of the approximated curve around the line segments even with higher polynomial's order, where it should be much smoother, and also it could not go through the edge even if the order is increased. The Cubic-Spline fitting method is more suitable to deal with this problem.
- Although, it has been shown in this chapter that the presented method works well, using different examples of varying complexity, but this may not be a feasible and acceptable solution, when large numbers of regions are merged into a single polynomial. Therefore a solution should be introduced to combine the regions in some groups and delivers one polynomial for each of them.

Combination Procedure

The combination procedure of the regions associated with one polynomial are not covered in this work, where it is until now not clear how to combine the feasible regions of Exp-MPC in a way to get a convex one, which assures the continuity of the polynomial. It is still a challenge in convex programming to combine convex regions and to get finally convex combination; "actually it is mathematically not possible". As the main goal of this method is to build one polynomial for a group of regions, then the area of the state space where this polynomial is valid should be determined; otherwise it will be no sense going through this method. In this chapter, 410 linear control laws were represented (as illustrated in Example-4) with one polynomial expression of higher order and with a good approximation, but it is not always the case. When the MPC controller complexity is increased to higher levels, this polynomial approximation procedure might tend to be very difficult or non-implementable. A good example for this case is the direct MPC controller for three-level inverter with NPC topology and with prediction horizons more than 2; here around 10000 regions are expected [82], which is a tough task to be implemented on DSP or FPGA platforms with a reasonable evaluation time. In this case, a possible way for the implementation can be followed as proposed in [82], by dividing the state space into some partitions before designing the MPC controller. As described in [82], the total number of regions of all the partitions (or quarters in this example) will be more than the 10000 regions. The implementation can be reduced to evaluate the control laws of 4000 regions for each quarter of the state space provided that the original state space is divided in 4 quarters; also this solution is possible to implement. The same procedure can be adopted for the more complex MPC problems to avoid the combination of the feasible regions generated by the Exp-MPC. This can be done in the following way:

I. Divide the state space into 4, 8 or much more partitions before solving the optimization problem, by setting the constraints on the state variables accordingly, as shown in Figure 7.14.

II. Build a simple branching tree to go through the partitions during the implementation in the same sequence as the state space is divided in the first step.

III. Build an iterative program to solve the MPC problem for each partition. Although the last step will generate total number of regions higher than if the MPC problem was solved for the complete state space at once, but it will lead to a better approximation.

IV. For each partition, build an approximated control law using polynomial approximation method.

V. Now, the steps above will deliver non-linear polynomial expressions defined over some convex and known subspaces or partitions. From here it is possible to combine these subspaces easily and do the approximation again for the resulting non linear control laws.

Again, the accuracy of the approximate polynomial controller as compared to the Exp-MPC controller will be the criteria to stop merging the subspaces.

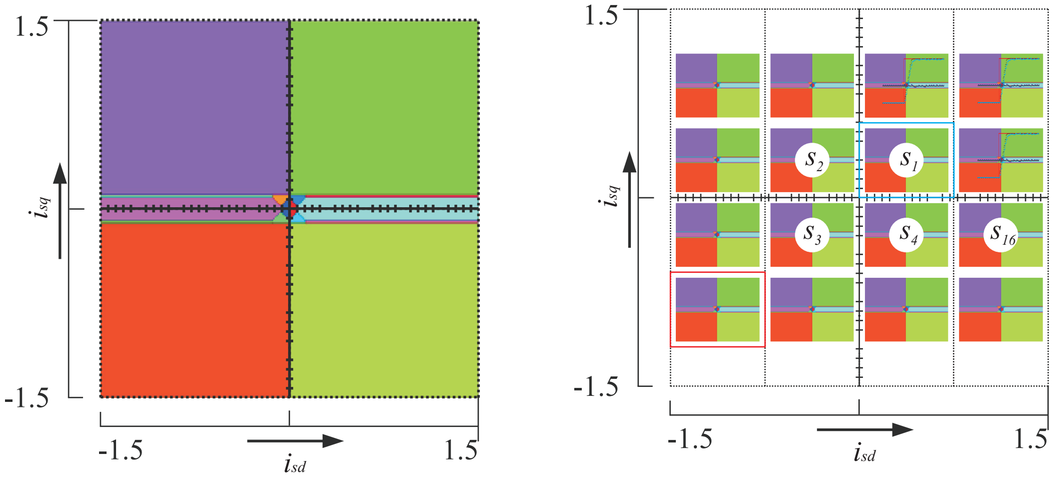


Figure 7.14: Regional division of the state space into 16 subspaces

- Finally, feasibility and stability issues should be more considered in future investigations. The introduced work in [119] and by keeping the approximate polynomial inside the stabilization set could be one of the possible solutions to assure the stability of the approximated optimization problem.

8 CONCLUSION AND FUTURE PERSPECTIVE

In this work the explicit solution of model based predictive control has been introduced for drive applications with two and three level inverters. More attention, however, is paid to the nonlinearities of the AC machine model and the voltage source inverter. The cross-coupling and nonlinearities between state variables in the fundamental model of AC machines arise in term of induced voltages. Different approaches are highlighted to compensate these nonlinearities either inside or outside the controller structure in an active or a passive way. The effect of these nonlinearities appears mainly in a steady-state offset and affects the tracking performance very clearly. The compensated terms have larger time constants than the machine current dynamics; hence the compensation of these terms outside the controller structure actively or passively is possible without affecting the controller performance. The external compensation for indirect control scheme is effective and the resulting explicit solution requires less evaluation time and lower memory usage, however the power of MIMO control structure is not utilized here properly. One of the possible extensions in the indirect control scheme is to include the PWM modulation strategy (Space vector modulation (SVM) [39], or synchronous optimal pulse width modulation [40]) in the current controller structure to build an optimal closed-loop PWM modulator according to the desired objectives. Using an optimal closed-loop PWM modulator fulfills different requirements: reduced current harmonics, eliminating the neutral point potential of the NPC inverter, and having a fixed switching frequency. The resulting optimal controller will generate in this case the optimal timing signals for the voltage vectors employed in the switching sequence for SVM, or the optimal switching angels in the synchronous optimal PWM instead of the continuous voltage inputs ($\boldsymbol{u}_s$) of the indirect current controller.

This kind of external compensation is not possible for direct control of the inverter. In direct control scheme, the controller has to generate the inverter switching states directly without using any modulation strategy. The nonlinear terms which are to be compensated are continuous signals; however the controller generated control inputs are discrete signals. For that reason, using the external compensation was not possible and instead of that the internal compensation was employed for the direct control scheme.

In general, augmenting the state vector is the only way to introduce a new parameter in the cost function of the off-line optimized MPC problem. Furthermore, solving the optimization problem using the multi-parametric programming approach is a virtue, where the state vector and hence the new added state variable will be considered as a free parameter and the optimization problem will be solved for the all possible values of this variable. Hence, weighting this variable in the cost function will affect the resulting optimal solution and change the optimization objective accordingly. This fact could be exploited to add new dummy or active state variables (like the number of switching, neutral point potential of the NPC inverter, and DC-link voltage) and weighting these variables in the cost function to change the optimization goal. For example, integrating the error integral as a new state variable to get a free-offset tracking (section 6.3) and adjusting its weights in the cost function will help to do online adaptation, force the actual current signal to follow the reference one, and keep it inside some band. Until now, however, there is no systematic way to adjust the weighting factors of the cost function optimally, which could be also a possible future trend.

HYSDEL toolbox provides a powerful description language for the nonlinear models, by representing the continuous and discrete system dynamics in one system model. Supporting the HYSDEL toolbox to the finite-state machine modeling makes it possible to represent the discrete nature of the inverter of different levels much effectively. Here, some of the prohibited switching possibilities could be excluded in the next prediction horizon to reduce the complexity of the resulting explicit solution and get a higher prediction horizon.

The validity of external and internal compensation of the nonlinear terms in the system model was proved in this work by simulation and experimental results. The proposed control structures are evaluated for two and three level inverter with NPC and FC [82] topologies, however they could be further extended in future works for more inverter topologies.

Finally, the polynomial approximation is a wide and a promising approach to reduce the online complexity of the explicit solution of MPC controller. It could be further extended and improved to remove the oscillation around the line segments and get a fine approximation with lower complexity. Avoiding the cross product terms in the polynomial expression reduces the computation effort necessary to evaluate the resulting polynomial as well as the memory space to store the polynomial coefficients.

Appendix-A

Quantity	**Symbol**	**Induction Machine**	**PMSM Machine**	**Unit**
Rated Phase Voltage	U_{phR}	220	220	[v]
Rated Phase Current	I_{phR}	4.9	2.9	[A]
Rated Power	P_R		1000	[W]
Rated Speed	ω_{mR}	2750	3000	[rpm]
Rated Torque	M_R	10	3	[Nm]
Rated Stator Frequency	F_s	50	50	[Hz]
Inertia	J	0.0572	0.008	[kgm^2]
Pole Pairs	P_p	1	4	pu
Stator Resistance	r_s	0.0447	0.0653	pu
Rotor Resistance	r_r	0.0443		pu
Stator Inductance	l_s	2.67	0.0607	pu
Rotor Inductance	l_r	2.67		pu
Mutual Inductance	l_m	2.56		pu
Permanent Flux	ψ_m		0.3061	pu

Appendix–B

The nominal phase values are used for the normalization:

Star connection	$U_{ph,R} = \frac{1}{\sqrt{3}} U_R$ $I_{ph,R} = I_R$
Y-connection	$U_{ph,R} = U_R$ $I_{ph,R} = \frac{1}{\sqrt{3}} I_R$
Frequency	$F_R = 50,\ 60\ \text{Hz}$

Thereby the following normalization base values result:

Voltage	$\sqrt{2}\, U_{ph,R}$
Current	$\sqrt{2}\, I_{ph,R}$
Impedance	$\frac{U_{ph,R}}{I_{ph,R}}$
Inductance	$\frac{U_{ph,R}}{\omega_{s,R} \,.\, I_{ph,R}}$
Flux Linkage	$\frac{\sqrt{2}\, U_{ph,R}}{\omega_{s,R}}$
Power	$3\, U_{ph,R} \,.\, I_{ph,R}$
Torque	$\frac{3p \,.\, U_{ph,R} \,.\, I_{ph,R}}{\omega_{s,R}}$
Rotating Speed	$\frac{\omega_{s,R}}{p}$
Frequency	F_R
Time	$\frac{1}{\omega_{s,R}}$

Definitions

General abbreviations

RHC	**R**eceding **H**orizon **C**ontrol Policy
GPC	**G**eneralized Model-Based **P**redictive **C**ontroller
CARIMA	**C**ontrolled **A**uto **R**egressive and **I**ntegrated **M**oving **A**verage Model
MPC	**M**odel-Based **P**redictive **C**ontroller
SMPC	**M**odel-Based **P**redictive **C**ontroller using the **S**implified Machine Model
EMPC	**M**odel-Based **P**redictive **C**ontroller using the **E**xtended Machine Model
DMPC	**D**irect **M**odel-Based **P**redictive **C**ontroller (using the **S**implified Machine Model)
EDMPC	**D**irect **M**odel-Based **P**redictive **C**ontroller using the **E**xtended Machine Model
Exp-MPC	The **Exp**licit Solution of **M**odel-Based **P**redictive **C**ontroller
Poly-App.	The **Poly**nomial **App**roximation of the Explicit Solution of Model-Based Predictive Controller
LTI	**L**inear **T**ime **I**nvariant System Model
PWA	**P**iece-**W**ise **A**ffine System Model
SISO	**S**ingle-**I**nput **S**ingle-**O**utput System
MIMO	**M**ultiple-**I**nput **M**ultiple-**O**utput System
MPP-	**M**ulti-**P**arametric **P**rogramming Approach
MPT	**M**ulti-**P**arametric Programming **T**oolbox
HYSDEL	**Hy**brid **S**ystem **De**scription **L**anguage Toolbox
KKT	**K**arush-**K**uhn-**T**ucker Optimality Conditions
LP	**L**inear **P**rogramming
QP	**Q**uadratic **P**rogramming
MIQP	**M**ixed **I**nteger **Q**uadratic **P**rogramming
BST	**B**inary **S**earch **T**ree Algorithm
PI	**P**roportional **I**ntegral Controller (general)
SPI	**S**calar **P**roportional **I**ntegral Controller
CPI	**C**omplex **P**roportional **I**ntegral Controller
LQR	**L**inear **Q**uadratic **R**egulator
FFC	**F**eed **F**orward **C**ompensation
VSI	**V**oltage **S**ource **I**nverter
IGBT	**I**nsulated **G**ate **B**ipolar **T**ransistor
FC VSI	**VSI** with **F**lying **C**apacitor Topology
NPC VSI	**VSI** with **N**eutral **P**oint **D**iode **C**lamped Topology
PWM	**P**ulse **W**idth **M**odulation Strategy
SPWM	**S**inus **PWM** Strategy

SVM	**S**pace **V**ector **M**odulation Strategy

V/f	Open Loop **V**oltage/ **f**requency Control Scheme
FOC	**F**ield-**O**riented **C**ontrol Scheme
DTC	**D**irect **T**orque **C**ontrol Scheme

IM	**I**nduction **M**otor
SM	**S**ynchronous **M**otor
Sync-RM	**Syn**chronous **R**eluctance **M**otor
PMSM	**P**ermanent **M**agnet **S**ynchronous **M**otor

ISA	**I**ndustry **S**tandard **A**rchitecture
ISR	**I**nterrupt **S**ervice **R**outine
DSP	**D**igital **S**ignal **P**rocessor
PLD-ICs	**P**rogrammable **L**ogic **D**evices
FPGA	**F**ield **P**rogrammable **G**ate **A**rray Devices
VHDL	**V**ery High Speed Integrated Circuit **H**ardware **D**escription **L**anguage
ADC	**A**nalogue to **D**igital **C**onverter
DAC	**D**igital to **A**nalogue **C**onverter

RTOS	**R**eal-**T**ime **O**perating **S**ystem
GPL	**G**eneral **P**ublic **L**icense
FIFO	**F**irst **I**n First **O**ut Scheduling/ Register
EDF	**E**arliest **D**eadline **F**irst Scheduling
ADEOS	**A**daptive **D**omain **E**nvironment for **O**perating **S**ystems

General nomenclatures

Q, q	Machine variable (general)
Q	Non-normalized variable
q	Normalized variable
q_R	Rated value of a variable
q_{ph}	Phase value of a variable
q_{base}	Base value of a variable
q_N	Nominal value of a variable
q_s	Stator variable
q_r	Rotor variable
q_1	Fundamental component
q_h	'h' Harmonic component
$\vec{q}$	Complex vector
$\hat{q}$	Peak value
$\overline{q}$	Estimated value
$q^{(K)}$	General k-coordinates
$q^{(S)}$	Stationary coordinates
$q^{(F)}$	Field coordinates
q_d	Direct axis component of a quantity
q_q	Quadrature axis component of a quantity
q_α	Real component of a quantity
q_β	Imaginary component of a quantity
δ	Coordinates transformation angle

Machine variables

i_s	Stator current
u_s	Stator voltage
r_s	Stator resistance
l_s	Stator inductance
x_s	Stator reactance
ψ_s	Stator flux linkage
ω_s	Stator angular velocity
f_s	Stator frequency
ω_k	Angular velocity of the general *k*-coordinates
ω_m	Mechanical angular velocity
l_m	Mutual inductance
x_m	Mutual reactance
i_r	Rotor current
u_r	Rotor voltage
r_r	Rotor resistance
l_r	Rotor inductance
x_r	Rotor reactance
ψ_r	Rotor flux
ω_r	Rotor angular velocity
f_r	Rotor frequency
ψ_m	Permanent rotor flux of PMSM
m_e	Electrical torque
m_l	Load torque
T_0, $\tau_{sampling}$	Normalized sampling time
$\omega_{s,R}$	Rated value of stator angular velocity
$\mathrm{P_P}$	Pole pairs
J	Rotor inertia
f_{sw}	Switching frequency
τ	Normalized time (general)
τ'_{σ}	Transient time constant
τ_d	Transient stator time constant
τ_r	Transient rotor time constant
τ_{sys}	Digital system time constant
τ_{inv}	Inverter time constant
τ_s	Digital system + Inverter time constant
τ_m	Mechanical time constant
τ_r	Rotor time constant
τ_s	Stator time constant
σ	Total leakage coefficient
r_{σ}	Effective resistance of both windings
k_s	Stator coupling factor
k_r	Rotor coupling factor
u_{dc}	DC-link voltage
i_{dc}	DC-link current

c_{upper}	Upper capacitor
c_{lower}	Lower capacitor
i_{np}	Neutral point current
u_{np}	Neutral Point voltage
s_a, s_b, s_c	Inverter switching states for *a, b, c* phases

PI Control variables

j	Complex operator
$\mathbf{s}$	Laplace operator
$F\,(\mathbf{s})$	Transfer function (general)
$F_s\,(\mathbf{s})$	Transfer function of the stator
$F_m\,(\mathbf{s})$	Transfer function of the machine
$F_{inv}\,(\mathbf{s})$	Transfer function of the inverter
$F_d\,(\mathbf{s})$	Transfer function of the delay-system
$F_o\,(\mathbf{s})$	Open-loop transfer function
$F_{ctrl}(\mathbf{s})$	Controller transfer function
t_i	PI controller, time constant
k_p	PI controller, proportional gain
t_{i1},t_{i2},t_{i3}	Complex controller time constants
g_o	Open loop gain
k_o	Complex controller, proportional gain
D	Damping factor

MPC Variables

N	Horizon (general)
N_p	Prediction horizon
N_c	Constraints horizon
N_u	Control horizon
$\mathbf{U}$	Control sequence

v	System variable (general)
v^*	Reference variable
v_{min}	Lower limit
v_{max}	Upper limit

$\mathbf{x}, x$	State vector, variable
$\mathbf{u}, u$	Input vector, variable
$\mathbf{y}, y$	Output vector, variable
$\mathbf{x}^*, x^*$	Reference vector, variable
$\mathbf{x}^d, x^d$	Disturbance vector, variable
k	Time instance
t	Time
K_{LQR}	Linear quadratic regulation, proportional gain

$\mathbf{M}$	System matrix (general) in discrete domain
$\mathbf{M}_k$, $\mathbf{M}_t$	System matrix in discrete domain
$\mathbf{M}^i$	System matrix for the i^{th}-dynamic in PWA formulation
$\mathbf{I}$	Identity matrix
$\mathbf{0}$	Zero matrix

$\mathbf{A}$	State matrix
$\mathbf{A}_{\mathrm{u}}$	Unstable part of state matrix
$\mathbf{B}$	Input matrix
$\mathbf{C}$	Output matrix
$\mathbf{D}$	Feedforward matrix
$\mathbf{f}$	System matrix in PWA formulation
$\mathbf{g}$	Output matrix in PWA formulation
$\mathbf{e}$	Error vector of the states
$\mathbf{e}_{\mathrm{I}}$	Integral of the error vector
$\mathbf{G}$	Guard lines constraints in PWA formulation
$\mathbf{Q}$	State variables weighting matrix
$\mathbf{R}$	Actuating variables weighting matrix
$\mathbf{P}$	Terminal state weighting matrix
$\boldsymbol{J}$	Cost function of the optimization problem
x_N	Terminal state variables
$\mathbf{P}_0$	Terminal state weighting matrix according to Riccati equation
F_{LQR}	Control action in LQR
$\boldsymbol{X}$	Polytopic constraints on the states
$\boldsymbol{U}$	Polytopic constraints on the inputs
$\boldsymbol{Y}$	Polytopic constraints on the outputs
$\boldsymbol{\Omega}$	Invariant set
R_i	Feasible region
U_i	Control law associated with region R_i
$\mathbf{BM}$	Branching matrix in binary search tree algorithm
$\mathbf{F_i}$	Matrix 1 of the affine control law
$\mathbf{G_i}$	Matrix 2 of the affine control law

References

[1] A. Nabae, I. Takahashi, and H. Akagi, “A New Neutral Point Clamped PWM Inverter,” *IEEE Transactions on Industry Applications*, Vol. IA-17, no. 5, Sept./ Oct. 1981, pp. 518-523.

[2] T. A. Meynard, and H. Foch, “Multilevel Conversion: High Voltage Choppers and Voltage-Source Inverters,” *Power Electronics Specialists Conference*, *Proc. IEEE-PESC*, Toledo, Spain, Vol. 1, 29 June- 3 July 1992, pp. 397-403.

[3] J. Holtz, “Pulsewidth Modulation - A Survey,” *IEEE Transactions on Industrial Electronics*, Vol. 30, no. 5, Dec. 1992, pp. 410-420.

[4] J. Rodriguez, L. Jih-Sheng, and P. Fang Zheng, “Multilevel Inverters: a Survey of Topologies, Controls, and Applications,” *IEEE Transactions on Industrial Electronics*, Vol. 49, no. 4, 2002, pp. 724-738.

[5] H. du Toit Mouton, “Natural Balancing of Three-Level Neutral-Point-Clamped PWM Inverters,” *IEEE Transactions on Industrial Electronics*, Vol. 49, no. 5, 2002, pp. 1017-1025.

[6] C. Newton, M. Sumner, and T. Alexander, “The Investigation and Development of a Multilevel Voltage Source Inverter,” *Sixth International Conference on Power Electronics and Variable Speed Drives*, Conf. Publ. no. 429, 1996, pp. 317-321.

[7] C. Newton, and M. Sumner, “Neutral Point Control for Multilevel Inverters: Theory, Design and Operational Limitations,” *Proceedings of IEEE IAS conference*, New Orleans, Louisiana, 5-9 Oct. 1997, pp. 1336-1343.

[8] C. Newton, and M. Sumner, “Novel Technique for Maintaining Balanced Internal DC Link Voltages in Diode Clamped Five-Level Inverters,” *IEE Proceedings on Electric Power Applications*, Vol. 146, no. 3, 1999, pp. 341-349.

[9] N. Celanovic, and D. Boroyevich, “A Comprehensive Study of Neutral-Point Voltage Balancing Problem in Three-Level Neutral Point Clamped Voltage Source PWM Inverters,” *IEEE Transactions on Power Electronics*, Vol. 15, no. 2, March 2000, pp. 242-249.

[10] T. Bruckner, and D. G. Holmes, “Optimal Pulse-Width Modulation for Three-Level Inverters,” *IEEE Transactions on Power Electronics*, Vol. 20, no. 1, Jan. 2005, pp. 82-89.

[11] K. Yamanaka, A. M. Hava, H. Kirino, Y. Tanaka, N. Koga, and T. Kume,” A Novel Neutral Point Potential Stabilization Technique Using the Information of Output Current Polarities and Voltage Vector,” *IEEE Transactions on Industry Applications*, Vol. 38, no. 6, Nov./ Dec. 2002, pp. 1572-1580.

[12] R. S. Kanchan, P. N. Tekwani, and K. Gopakumar, “Three-Level Inverter Scheme with Common Mode Voltage Elimination and DC Link Capacitor Voltage Balancing for an Open-End Winding Induction Motor Drive,” *IEEE Transactions on Power Electronics*, Vol. 21, no. 6, 2006, pp. 1676-1683.

[13] J. Holtz, and N. Oikonomou, "Neutral Point Potential Balancing Algorithm at Low Modulation Index for Three-Level Inverter Medium Voltage Drives," *Industry Applications Conference, 40th IAS Annual Meeting*, Vol. 2, 2005, pp. 1246-1252.

[14] L. M. Tolbert, and T. G. Habetler, "Novel Multilevel Inverter Carrier-Based PWM Method," *IEEE Transactions on Industry Applications*, Vol. 35, no. 5, Sept./ Oct. 1999, pp. 1098-1107.

[15] F. Wang, "Sine-Triangle Versus Space Vector Modulation for Three-Level PWM Voltage Source Inverters," *IEEE Transactions on Industry Applications*, Vol. 38, no. 2, Mar./ Apr. 2002, pp. 500-506.

[16] J-S. Lai, and F. Z. Peng, "Multilevel Converters- A New Breed of Power Converters," *IEEE Transactions on Industry Applications*, Vol. 32, no. 3, May- June 1996, pp. 509-517.

[17] G. Carrara, S. G. Gardella, M. Archesoni, R. Salutari, and G. Sciutto, "A New Multilevel PWM Method: A Theoretical Analysis," *IEEE Transactions on Power Electronics*, Vol. 7, no. 3, July 1992, pp. 497-505.

[18] J. K. Steinke, "Switching Frequency Optimal PWM Control of a Three-Level Inverter," *IEEE Transactions on Power Electronics*, Vol. 7, no. 3, July 1992, pp. 497-505.

[19] S. Ogasawara, and H. Akagi, "Analysis of Variation of Neutral Point Potential in Neutral-Point-Clamped Voltage Source PWM Inverters," *Proceedings of IEEE Industrial Applications Society conference recordings*, 1993, pp. 965-970.

[20] L. M. Tolbert, and T. G. Habetler, "Novel Multilevel Inverter Carrier-Based PWM Method," *IEEE Transactions on Industry Applications*, Vol. 35, no. 5, Sept./ Oct. 1999, pp. 1098-1107.

[21] F. Wang, "Multilevel PWM VSIs-Coordinated Control of Regenerative Three-Level Neutral Point Clamped Pulsewidth Modulated Voltage Source Inverter," *IEEE Magazine on Industrial Applications*, Vol. 10, no. 4, Jul./ Aug. 2004, pp. 51-58.

[22] P. Zhiguo, and P. Fang Zheng, "A Sinusoidal PWM Method with Voltage Balancing Capability for Diode-Clamped Five-Level Converters," *IEEE Transactions on Industry Applications*, Vol. 45, no. 3, 2009, pp. 1028-1034.

[23] N. AL_Sheakh Ameen, and R. Kennel, "Pentium System Set-up on Basis PC104 for the Experimental Research of a V/f-Control with SVPWM (Space Vector Pulse Width Modulation)," *Technical Report at EMAD Wuppertal, Germany,* June 2006.

[24] N. AL_Sheakh Ameen, A. A. Naassani, and A. B. Alkhodre, "Realization of a Digital System for Electrical Drive Applications," *Research Journal of Aleppo University*, Syria, June 2008, Vol. 62.

[25] N. AL_Sheakh Ameen, A. A. Naassani, and A. B. Alkhodre, "Use Open-Source Operating Systems (Linux) for Electrical Drive Applications," *Research Journal of Aleppo University*, Syria, June 2008, Vol. 62.

[26] N. AL_Sheakh Ameen, A. A. Naassani, and R. Kennel, "Design of a Digital System Dedicated for Electrical Drive Applications," *EPE Journal,* Vol. 20, no. 04, December 2010, pp. 37-44.

[27] R. S. Kanchan, N. AL_Sheakh Ameen, R. Kennel, and H. du Toit Mouton, "Three-Level Inverter Design with Neutral-Point Diode-Clamped and Flying Capacitor Topologies," *Technical Report at EMAD Wuppertal, Germany*, March 2008.

[28] C. Cecati, “Microprocessors for Power Electronics and Electrical Drives Applications,” *IEEE Industrial Electronics Society Newsletter*, Vol. 46, no. 3, Sep. 1999, pp. 5-9.

[29] K. Meah, S. Hietpas, and S. Ula, “Rapid Control Prototyping of a Permanent Magnet DC Motor Drive System Using dSPACE and Mathworks Simulink,” *Applied Power Electronics Conference, APEC’07*, 25 Feb.- 1 March 2007, pp. 856-861.

[30] A. Gebregergis, and P. Pillay, “Implementation of Fuel Cell Emulation on DSP and dSPACE Controllers in the Design of Power Electronic Converters,” *Industry Applications Society Annual Meeting*, *IAS '08,* Oct. 2008, pp. 1-8, 5-9.

[31] A. Rubaai, M. J. Castro-Sitiriche, and A. R. Ofoli, “DSP-Based Laboratory Implementation of Hybrid Fuzzy-PID Controller Using Genetic Optimization for High-Performance Motor Drives,” *IEEE Transactions on Industrial Applications*, Vol. 44, no. 6, Nov./ Dec. 2008, pp. 1977-1986.

[32] L. G. B. Rolim, R. M. Stephan, W. I. Suemitsu , and J. L. da Silva Neto , “UFRJ Power Electronics Teaching Lab: Ten Years,” *IEEE Power Electronics Education Workshop*, 2005, pp. 69-73.

[33] R. M. Fernandes , L. G. B. Rolim, and W. I. Suemitsu, “Design and Implementation of a Power-Electronic Remote-Laboratory (ELEPOT-rLab),” *ISIE'03-International Symposium on Industrial Electronics*. Anais do ISIE'03-*International Symposium on Industrial Electronics*, Vol. 1, 2003, pp. 307-311.

[34] S. Abourida, C. Dufour, J. Belanger, G. Murere, N. Lechevin, and Biao Yu, “Real-Time, PC-Based Simulator of Electric Systems and Drives,” *Applied Power Electronics Conference and Exposition, APEC2002 seventeenth Annual IEEE*, Vol. 1, 2002, pp. 433-438.

[35] A. Linder, “A Rapid-Prototyping System Based on Standard PCs with RTAI as Real-Time Operating System,” *Third Real-Time Linux Workshop*, Milano, Italy, 26-29 Nov. 2001.

[36] H. W. de Kock, M. J. Kamper, O. C. Ferreira, and R. M. Kennel, “Position Sensorless Control of the Reluctance Synchronous Machine Considering High Frequency Inductances,” *7th International Conference on Power Electronics and Drive Systems*, 2007, pp. 812-821.

[37] P. Szczupak, R. Kennel, and T. Boller, “Sensorless Control of 3-Phase PWM Rectifier in Case of Grid Phase Disconnection,” *Power Electronics Specialists Conference. PESC '05. IEEE 36th*, 2005, pp. 2019-2022.

[38] J. C. Ramirez Martinez, R. Kennel, and N. AL_Sheakh Ameen, “Comparative Analysis of On-Line and Off-Line Explicit Solutions, Applied in a Predictive Direct Current Control,” *5th International Conference on Power Electronics, Machines and Drives PEMD*, Brighton, UK, 2010.

[39] J. Holtz, “Pulse Width Modulation for Electronic Power Conversion,” *Proceedings of the IEEE*, Vol. 82, no. 8, Aug. 1994, pp. 1194-1214.

[40] J. Holtz, and N. Oikonomou, “Closed-Loop Control of Medium Voltage Drives Operated with Synchronous Optimal Pulsewidth Modulation,” *IEEE Transactions on Industrial Electronics*, Vol. 55, no. 2, Jan. 2008, pp. 115-123.

[41] F. Blaabjerg, P. C. Kjaer, P. O. Rasmussen, and C. Cossar, “Improved Digital Current Control Methods in Switched Reluctance Motor Drives,” *IEEE Transactions on Power Electronic*, Vol. 14, no. 3, May 1999, pp. 563-572.

[42] R. M. Kennel, “Encoders for Simultaneous Sensing of Position and Speed in Electrical Drives with Digital Control,” *IEEE Transactions on Industry Applications*, Vol. 43, no. 3, 2007, pp. 993-1000.

[43] M. Timmerman, B. V. Beneden, and L. Uhres “Windows NT Real-Time Extensions, Better or Worse?,” *Real-Time Magazine*, 3(98), 1998, pp.11-19. Available on: http://www.realtime-info.be.

[44] S. Hill, and B. Krishnamurthy, “Introduction to Linux for Real-Time Control,” *prepared for National Institute of Standards & Technology by Aeolean Inc.* Available on: http://www.isd.mel.nist.gov/projects/rtlinux/intro-rtl.pdf.

[45] RTAI Installation Guide, Available on: http://www.captain.at/programming/rtai/kernel-2.4.php.

[46] IEEE Std. 1003.1-2001 (Open Group Technical Standard, Issue 6). Standard for Information Technology, Portable Operating System Interface (POSIX). *Technical report, IEEE Press*, New York, 2001.

[47] V. Yodaiken, “Adding Real-Time Support to General Purpose Operating Systems,” 1999.

[48] ADEOS Project Homepage, 2002. Available on: http://www.opersys.com/adeos.

[49] K. Yaghmour, “Adaptive Domain Environment for Operating Systems,” *Technical report, OperSys.com*, 2001, Available on: http://opersys.com/ftp/pub/Adeos/adeos.pdf.

[50] J. Holtz, “The Representation of AC Machine Dynamics by Complex Signal Flow Graphs,” *IEEE Transactions on Industrial Electronics*, Vol. 42, no. 3, 1995, pp. 263-271.

[51] J. Holtz, “On the Spatial Propagation of Transient Magnetic Field in AC Machines,” *IEEE Transactions on Industrial Applications*, Vol. 32, no. 4, 1996, pp. 927-937.

[52] J. Holtz, J. Quan, G. Schmidt, J. Pontt, J. Rodriguez P., P. Newman, and H. Miranda, “Design of Fast and Robust Current Regulator for High Power Drives Based on Complex State Variables,” *IEEE Transactions on Industrial Applications*, Vol. 40, Sep./ Oct. 2004, no. 5, pp. 1388-1397.

[53] J. O. Krah, and J. Holtz, “High-Performance Current Regulation for Low Inductance Servo Motors,” *Industrial Applications Conference, Thirty-Third IAS Annual Meeting*, Vol. 1, 1998, pp. 490-499.

[54] F. Britz, M. W. Gegener, and R. D. Lorenz, “Dynamic Analysis of Current Regulators of for AC Motors Using Complex Vectors,” *IEEE Transactions on Industrial Applications*, Vol. 35, no. 6, Nov./ Dec. 1999, pp. 1424-1432.

[55] C. Schauder, and R. Caddy, “Current Control of Voltage-Source Inverter for Fast Four-Quadrant Drive Performance,” *IEEE Transaction on Industrial Applications*, Vol. IA-8, 1982, pp. 163-171.

[56] B. K. Bose, “Modern Power Electronics and AC Drives,” *Pearson Education*, India, 2002.

[57] M. P. Kazmierkowski, F. Blaabjerg, and R. Krishnan, “Control in Power Electronics, Selected Problem,” *Academic Press (Elsevier Science)*, USA, 2002.

[58] O. Nelles, “Grundbegriffe der Regelungstechnik,” Lecture Notes, 2011, Available on: www.uni-siegen.de/fb11/mrt/lehre/mrt2/mrt2-folien.pdf.

[59] F. Fröher, and F. Orttenburger, "Einführung in die elektronische Regelungstechnik," *Siemens Aktiengesellschaft*, 1976.

[60] D. N. Zmood, and D. G. Holmes, and G. H. Bode "Frequency- Domain Analysis of Three Phase Linear Current Regulators," *IEEE Transaction on Industrial Applications*, Vol. 37, no. 2, 2001, pp. 601-610.

[61] D. N. Zmood and D. G. Holmes, "Stationary Frame Current Regulation of PWM Inverters with Zero Steady-State Error," *IEEE Transaction on Power Electronics*, Vol. 18, no. 3, 2003, pp. 814-822.

[62] J. Holtz, and S. Statfeldt, "A Predictive Controller for the Stator Current Vector of AC Machines Fed from a Switched Voltage Source," *in IPEC Tokyo 1983 Conf. Record*, Vol.2, pp. 1665-1675.

[63] R. Kennel, and D. Schroeder, "Predictive Control Strategy for Converters," *Proc. of the third IFAC Symposium*, 1983, pp. 407-427.

[64] D. Q. Mayne, J. B. Rawlings, C. V. Rao, and P. O. M. Scokaert, "Constrained Model Predictive Control: Stability and Optimality," *Automatica*, Vol. 36(6), Jun. 2000, pp. 789-814.

[65] V. Dua, and E. N. Pistikopoulos, "An Algorithm for Multi-Parametric Mixed-Integer Linear Programming Problems," *Annals of operations research*, Springer, 2000.

[66] E. N. Pistikopoulos, M. C. Georgiadis, and V. Dua, "Multi-Parametric Model-Based Control," VILEY-VCH Verlag, 2007.

[67] K. Fukuda, "Polyhedral Computation FAQ, 2000," *Swiss Federal Institute of Technology*, Switzerland, Available on: http://www.ifor.math.ethz.ch/~fukuda/polyfaq/polyfaq.html.

[68] A. Bemporad, F. Borrelli, and M. Morari, "Piecewise Linear Optimal Controllers for Hybrid Systems," *American Control Conference ACC 2000*, Chicago, 2000, pp. 1190-1194.

[69] P. Tondel, T. A. Johansen, and A. Bemporad, "Evaluation of Piecewise Affine Control via Binary Search Tree," *Automatica*, Vol. 39, 2003, pp. 945-950.

[70] P. Tondel, and T. A. Johansen, "Complexity Reduction in Explicit Model Predictive Control," *IFAC World Congress*, Barcelona, 2002.

[71] T. Geyer, F. D. Torrisi, and M. Morrari, "Optimal Complexity Reduction of Piecewise Affine Models based on Hyperplane Arrangements," *Proceeding of the 2004 American Control Conference*, Boston, Massachusetts, Vol. 2, 30 June- 2 July 2004, pp. 1190-1195.

[72] T. A. Johansen, I. Petersen, and O. Slupphaug, "Explicit Suboptimal Linear Quadratic Regulation with State and Input Constraints," *Automatica*, Vol. 38, no. 7, Jul. 2002, pp. 1099-1111.

[73] P. Grieder, M. Kvasnica, M. Baotic, and M. Morari, "Low Complexity Control of Piecewise Affine Systems with Stability Guarantee," *In American Control Conference*, Boston, USA, June 2004.

[74] T. Geyer, "Low Complexity Model Predictive Control in Power Electronics and Power System," Ph.D Thesis, Cuvillier Verlag Göttingen, Germany, 2005.

[75] M. Kvasnica, J. Löfberg, M. Herceg, L. Cirka, and M. Fikar, "Low-Complexity Polynomial Approximation of Explicit MPC via Linear Programming," *American Control Conference (ACC)*, Jul. 2010, pp. 4713-4718

[76] A. Linder, R. S. Kanchan, R. M. Kennel, and P. Stolze, “Model-Based Predictive Control of Electric Drives,” Cuvillier Verlag Göttingen, Germany, 2010.

[77] M. Kvasnica, P.Grieder, M. Baotic, and M. Morari, “Multi-Parametric Toolbox (MPT),” Lecture Notes in Computer Science, Vol. 2993, March 2004, pp. 448-462.

[78] F. D. Torrisi, and A. Bemporad, “HYSDEL - A Tool for Generating Computational Hybrid Models for Analysis and Synthesis Problems,” *IEEE Transaction on Control Systems Technology*, Vol.12, no. 2, March 2004, pp. 235-249.

[79] F. Borrelli, A. Bemporad, and M. Morari, “Predictive Control for Linear and Hybrid Systems,” Cambridge University Press, 2011, In press.

[80] M. Kvasnica, and M. Fikar, “Design and Implementation of Model Predictive Control Using Multi-Parametric Toolbox and YALIMP,” *IEEE International Symposium on Computer-Aided Control System Design*, Yokohama, Japan, 8-10 Sep. 2010.

[81] N. AL_Sheakh Ameen, and R. Kennel, “Realization of the Explicit Solution of Model-Based Predictive Control for Electric Drive Applications Using RT-Linux,” *Real-Time Linux Workshop, RTL12 WS*, Nairobi, Kenya, Oct. 2010.

[82] P. Stolze, N. AL_Sheakh Ameen, and R. Kennel, “Predictive Control of Multilevel Inverters Using Online and Offline Optimization,” *SPS/IPC/Drives Conference*, Nürnberg, Germany, Nov. 2010.

[83] E. F. Camacho, and C. Bordons, “Model Predictive Control,” Springer Verlag, London, 1999.

[84] J. B. Rawlings and K. R. Muske, “The Stability of Constrained Receding Horizon Control,” *IEEE Transaction on Automatic Control*, Vol. 38, no. 10, Oct. 1993, pp. 512-516.

[85] W. Kwon, and A. Pearson, “A Modified Quadratic Cost Problem and Feedback Stabilization of a Linear System,” *IEEE Transaction on Automatic Control*, Vol. 22, 1997, pp. 838-842.

[86] S. S. Keerthi, and E.G. Gilbert, “Optimal Infinite-Horizon Feedback Control Laws for a General Class of Constrained Discrete-Time Systems: Stability and Moving-Horizon Approximations,” *Journal of Optimization Theory and Applications*, Vol. 57, no. 2, 1988, pp. 265-293.

[87] A. Zheng, and M. Morari, “Stability of Model Predictive Control with Mixed Constraints,” *IEEE Transactions Automatic Control*, Vol. 40, 1995, pp. 1818-1823.

[88] W. H. Kwon, and D. G. Byun, “Receding Horizon Tracking Control as a Predictive Control and its Stability Properties,” *International Journal of Control*, Vol. 50, no. 5, 1989, pp. 1807-1824.

[89] P. O. M. Scokaert, and J. B. Rawlings, “Infinite Horizon Linear Quadratic Control with Constraints,” *Proceedings of the 13th IFAC World Congress*, San Francisco, USA, Vol. 7a/041, 1996, pp. 109-114.

[90] D. J. Chmielewski, and V. Manousiouthakis, “On Constrained Infinite-Time Linear Quadratic Optimal Control,” *Systems & Control Letters*, Vol. 29/3, Nov. 1996, pp. 121-129.

[91] P. O. M. Scokaert, and J. B. Rawlings, “Constrained Linear Quadratic Regulation,” *IEEE Transactions on Automatic Control*, Vol. 43, no. 8, 1998, pp.1163-1169.

[92] A. Bemporad, and M. Morari, "Robust Model Predictive Control: a Survey," *Lecture Notes in Control and Information Sciences, Robustness in Identification and Control*, Vol. 245, 1999, pp. 207-226.

[93] J. W. Choi, "Anti Windup Strategy for PI-Type Speed Controller," *IEEE Transaction on Industrial Electronics*, Vol. 56, no. 6, June 2009, pp. 2039-2046.

[94] K. J. Aström, and R. M. Murray, "Feedback System, An Introduction for Scientists and Engineers," Princeton University Press, 30 Aug. 2011, an electronic edition is available on: http://www.cds.caltech.edu/~murray/books/AM08/pdf/am08-complete_30Aug11.pdf.

[95] A. E. Bryson, and Yu-Chi Ho, "Applied Optimal Control, Optimization, Estimation, and Control," John Wiley & Sons, 1975.

[96] K. Ogata, "Modern Control Engineering, 4th Edition," Prentice Hall, 2002.

[97] G. F. Franklin, J. D. Powell, and M. Workman, "Digital Control of Dynamic Systems," Third Edition, Addison-Wesley, 1998.

[98] S. Matsutani, T. Zanma, Y. Sumiyoshi, M. Ishida, A. Imura, and M. Fujitsuna, "Optimal Control of PMSMs Using Model Predictive Control with Integrator," *International Joint Conference, ICROS-SICE, Fukuoka International Congress Center*, Japan, 18-21 August 2009.

[99] S. Mariethoz, A.Domahidi, and M. Morari, "Sensorless Explicit Model Predictive Control of Permanent Magnet Synchronous Motors," *Electric Machines and Drives Conference, IEMDC'09*, 3-6 May 2009, pp. 1250-1257.

[100] A. Linder, and R. Kennel, "Direct Model Predictive Control - A New Direct Predictive Control Strategy for Electrical Drives," *11th European Conference on Power Electronics and Applications EPE*, Dresden, 10-15 Sep. 2005.

[101] Saverio Bolognani, Silverio Bolognani, L. Peretti, and M. Zigliotto, "Combined Speed and Current Model Predictive Control with Inherent Field-Weakening Features for PMSM Drives," *14th IEEE Electro technical Conference, MELECON,* 5-7 May 2008, pp. 472-478.

[102] N. Kutasi, A. Kelemen, and M. Imecs, "Vector Control of Induction Motor Drives with Model Based Predictive Current Controller," *International Conference on Computational Cybernetics*, 2008, pp. 21-26.

[103] M. T. Cychowski, R. Nalepa, and T. O'Mahony, "Explicit Model Predictive Control of a Permanent Magnet Synchronous Motor Drive," *40th Universities Power Engineering Conference*, Cork, 2005.

[104] F. Morel, X. Lin-Shi, J.-M. Retif, and B. Allard, "A Predictive Current Control Applied to a Permanent Magnet Synchronous Machine, Comparison with a Classical Direct Torque Control," *Electric Power Systems Research*, Vol. 78, no. 8, 2008, pp. 1437-1447.

[105] S. Mastellone, G. Papafotiou, and E. Liakos, "Model Predictive Direct Torque Control for MV Drives with LC Filters," *13th European Conference on Power Electronics and Applications, EPE'09*, Barcelona, 8-10 Sep. 2009, pp. 1-10.

[106] R. Kennel, and A. Linder "Predictive Control of Inverter Supplied Electrical Drives," *Power Electronics Specialists Conference, IEEE 31st Annual PESC,* Vol. 2, 2000, pp. 761.

[107] A. Linder, and R. Kennel, “Model Predictive Control for Electrical Drives,” *Power Electronics Specialists Conference, Proc. 36th IEEE PESC*, 16 June 2005, pp. 1793-1799.

[108] S. Bolognani, S. Bolognani, L. Peretti, and M. Zigliotto, “Design and Implementation of Model Predictive Control for Electrical Motor Drives,” *IEEE Transactions on Industrial Electronics*, Vol. 56, no. 6, 2009, pp. 1925-1936.

[109] E. Fuentes, J. Rodrigues, C. Silva, S. Diaz, and D. Quevedo, “Speed Control of a Permanent Magnet Synchronous Motor Using Predictive Current Control,” *in Proc. IEEE 6th Int. Power Electronics and Motion Control Conf. IPEMC ’09*, 2009, pp. 390-395.

[110] N. AL_Sheakh Ameen, “The Normalization Procedure of AC Machines under Different Aspects,” ABB Technical Report - SECRC/PT/EMMC, 2010.

[111] N. AL_Sheakh Ameen, B. S. Galal, R. M. Kennel, and R. S. Kanchan,“ The Polynomial Approximation of the Explicit Solution of Model-Based Predictive Controller for Drive Applications,” *IEEE Workshop on predictive control of electrical drives and power electronics*, Munich, Germany, 14-15 Oct. 2011, pp. 113-118.

[112] N. AL_Sheakh Ameen, R. M. Kennel, and R. S. Kanchan,“ The Explicit Solution of Model-Based Predictive Control by Considering the Nonlinearities in Drive Applications,” *IEEE Transactions on Power Electronics*, submitted.

[113] T. Geyer, G. Papafotiou, and M. Morari, “Model Predictive Direct Control-Part I: Concept, Algorithm, and Analysis,” *IEEE Transactions on Industrial Electronics*, Vol. 56, no. 6, 2009, pp. 1894-1905.

[114] T. Geyer, “Generalized Model Predictive Direct Torque Control: Long Prediction Horizons and Minimization of the Switching Losses,” *Proceeding of the 48th IEEE Conference on Decision and Control*, Shanghai, 15-18 Dec. 2009, pp. 6799-6804.

[115] T. Geyer, G. A. Beccuti, G. Papafotiou, and M. Morari, “Model Predictive Direct Control of Permanent Magnet Synchronous Motors,” *Energy Conversion Congress and Exposition (ECCE), IEEE*, 2010, pp. 199-206.

[116] T. Geyer, “Computationally Efficient Model Predictive Direct Torque Control,” *Energy Conversion Congress and Exposition (ECCE), IEEE*, 2010, pp. 207-214.

[117] J. Holtz, “Machine Analysis,” Lecture Notes, 1999.

[118] D. Schröder, “Elektrische Antriebe Regelung von Antriebssystemen 3. Auflage,” Springer, 2009.

[119] J. Rodriguez, J. Pontt, C. A. Silva, P. Correa, P. Lenzana, and P. Cortes, “Predictive Current Control of a Voltage Source Inverter,” *IEEE Transactions on Industrial Electronics*, Vol. 54, no. 1, Feb. 2007, pp. 495-503.

[120] P. Cortes, J. Rodriguez, R. Vargas, and U. Ammann, “Cost Function-Based Predictive Control for Power Converter,” *Conference of IEEE on Industrial Electronics, IECON 2006, 32nd Annual*, Paris, 6-10 Nov. 2006, pp. 2268-2273.

[121] M. Kvasnica, J. Löfberg, M. Herceg, L. Cirka, and M. Fikar, “Low-Complexity Polynomial Approximation of Explicit MPC via Linear Programming,” *American Control Conference*, Jul. 2010, pp. 4713-4718.

[122] M. Herceg, M. Kvasnica, M. Fikar, and L. Cirka, “Real-Time Control of a Thermo-Optical Device Using Polynomial Approximation of MPC Scheme,” *In Proceedings of the 17th*

International Conference on Process Control '09, Slovak University of Technology in Bratislava, Strbske Pleso, Slovakia, 2009, pp. 332-340.

[123] M. Kvasnica, F. J. Christophersen, M. Herceg, and M. Fikar, "Polynomial Approximation of Closed-form MPC for Piecewise Affine Systems," July 2008, *In Proceedings of the 17th World Congress of the International Federation of Automatic Control*, Seoul, Korea, pp. 3877-3882.

[124] M. Kvasnica, J. Löfberg, M. Herceg, L. Cirka, and M. Fikar, "Polynomial Approximation of Closed-Form MPC with *O*(1) Implementation and Storage," 2009, Available on: http://www.control.isy.liu.se/~johanl/2009_CDC_MPCAPPROXIMATION.pdf.

[125] H. J. Stetter, "Numerical Polynomial Algebra," Society for Industrial and Applied Mathematics (siam), 2004.