Predictive Control in Power Electronics and Drives: basic concepts, theory and methods

Daniel E. Quevedo, Ricardo P. Aguilera, and Tobias Geyer

Abstract In this chapter we revise basic principles and methods of model predictive control with a view towards applications in power electronics and drives. The simplest predictive control formulations use horizon-one cost functions, which can be related to well-established dead-beat controllers. Model predictive control using larger horizons has the potential to give significant performance benefits, but requires more computations at each sampling instant to solve the associated optimization problems. For particular classes of system models, we discuss practical algorithms, which make long-horizon predictive control suitable for power electronics applications.

1 Introduction

Model predictive control (MPC), also referred to as *receding horizon control*, has received significant attention. Applications and theoretical results abound, see, e.g., the books [17, 51, 52, 77, 108] and survey papers [84, 98]. An attractive feature of MPC is that it can handle general constrained nonlinear systems with multiple inputs and outputs in a unified and clear manner.

Particularly, in the field of power electronics, various embodiments of MPC principles have emerged as a promising control alternative for power converters and electrical drives [20, 66, 111, 112]. This is due to the fact that predictive control algorithms present several advantages that make it suitable for the control of power converters:

Daniel E. Quevedo · Ricardo P. Aguilera
School of Electrical Engineering & Computer Science, The University of Newcastle, NSW 2308, Australia, e-mail: dquevedo@ieee.org, raguilera@ieee.org

Tobias Geyer
ABB Corporate Research, ABB Switzerland Ltd, Power Electronic Systems, Segelhofstrasse 1 K, 5405 Baden-Dättwil, Switzerland, e-mail: t.geyer@ieee.org

1. Concepts are intuitive and easy to understand;
2. MPC can handle converters with multiple switches and states, e.g., current, voltage, power, torque, etc.;
3. constraints and nonlinearities can be easily included; and
4. the resulting controller is, in general, easy to implement.

The aim of this chapter is to provide an overview of the basic theoretical underpinnings of MPC and to illustrate their use in power converters. For that purpose, in Section 2, we begin by presenting the key element underlying MPC, namely, that of moving horizon optimization. In Section 3 we analyze, in detail, the special case of MPC with horizon one. These simple formulations often give good results and only require little computations, leading to their immense popularity in power electronics applications [112]. In particular, we establish relationships between dead-beat control and horizon-one MPC strategies. For implementations without a modulator, dealt with by so called *Finite Control-Set MPC* with a quadratic cost function, we derive the optimal control input by exploring the geometry of the underlying optimization problem. Section 4 focuses on the general case of MPC with longer horizons. In general, using long horizons yields better closed-loop performance than MPC with horizon one [34, 53]. However, solving the underlying on-line optimization problems can be highly demanding. To overcome these issues, we examine special purpose optimization algorithms, which allow one to implement long-horizon optimal solutions in practical power electronics and drive systems.

2 Basic Concepts

Various model predictive control methods have been proposed for controlling power electronics and drives. Here, one can distinguish between formulations that use system models governed by linear time invariant dynamics, and those that incorporate nonlinearities. Most MPC strategies are formulated in a discrete-time setting with a fixed sampling interval, say $h > 0$. System inputs are restricted to change their values only at the discrete sampling instants, i.e., at times $t = kh$, where $k \in \mathbb{N} \triangleq \{0, 1, 2, \dots\}$ denotes the sampling instants.

Since power electronics applications are often governed by nonlinear dynamic relations, it is convenient to represent the system to be controlled in discrete-time state space form via:

$$x(k+1) = f(x(k), u(k)), \quad k \in \mathbb{N}, \tag{1}$$

where $x(k) \in \mathbb{R}^n$ denotes the state value at time k and $u(k) \in \mathbb{R}^m$ is the plant input. Depending on the application at hand, the system state is a vector, which may contain capacitor voltages, inductor and load currents, and fluxes.

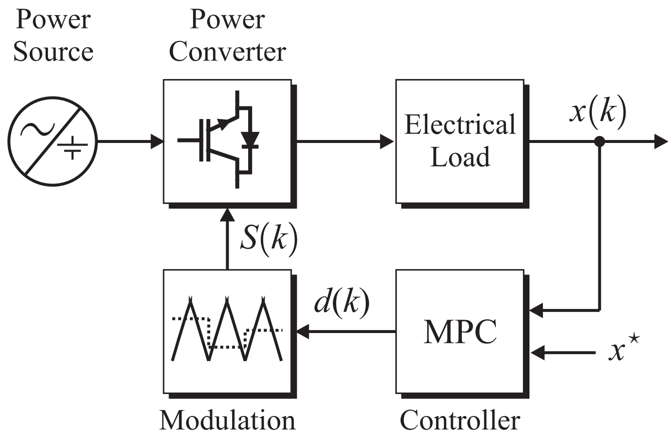


Fig. 1 MPC with continuous control set.

2.1 System Constraints

An interesting feature of the MPC framework is that it allows one to incorporate state and input constraints, say:

$$\begin{aligned} x(k) &\in \mathbb{X} \subseteq \mathbb{R}^n, \qquad k \in \{0,1,2,\dots\}, \\ u(k) &\in \mathbb{U} \subseteq \mathbb{R}^m, \qquad k \in \{0,1,2,\dots\}. \end{aligned} \tag{2}$$

State constraints can, for example, correspond to constraints on capacitor voltages in flying capacitor converters or neutral point clamped converters. Constraints on load currents can also be modeled as state constraints. Throughout this chapter we will focus on input constraints, since their form is peculiar to the nature of power converters.

Input constraints, $u(k) \in \mathbb{U}$, are related to the switch positions during the interval $(kh,(k+1)h]$. If the converter uses a modulator, then $u(k)$ will be constrained to belong to a bounded continuous set. For example, the components of $u(k)$ could correspond to duty cycles, $d(k)$, or PWM reference signals. In this case, the control input is constrained by

$$u(k) = d(k) \in \mathbb{U} \triangleq [-1,1]^m \subset \mathbb{R}^m, \qquad k \in \{0,1,2,\dots\}, \tag{3}$$

where m denotes the number of phases, see Fig. 1. Clearly, the above model can only approximate switching effects, see also [73]. Nevertheless, as we will see, several interesting and powerful controllers for power electronics and drives have been developed by using this simple setting.

On the other hand, in direct control-applications, where no modulator is used, $u(k)$ is constrained to belong to a finite set describing the available switch combinations. Such approaches have attracted significant attention in the power electronics

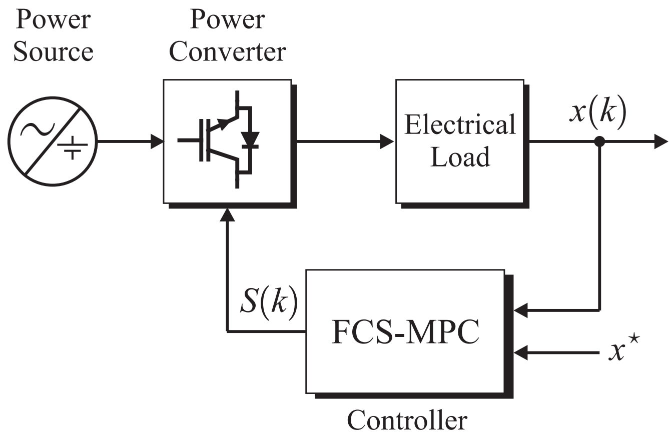


Fig. 2 MPC with finite control set.

community, often under term Finite Control Set MPC [112]. The main advantage of this predictive control strategy comes from the fact that switching actions, say $S(k)$, are directly taken into account in the optimization as constraints on the system inputs, see Fig. 2. Thus, the control input is restricted to belong to a finite set represented by

$$u(k) = S(k) \in \mathbb{U} \triangleq \{0,1\}^m \subset \mathbb{R}^m, \qquad k \in \{0,1,2,\dots\}, \tag{4}$$

where $\mathbb{U}$ is a boolean set obtained by combining the m switch values. For the control of some multi-level topologies, it is at times convenient to consider the resultant voltage level as the control input, without making the distinction at a switch level. For example, for a 5-level inverter, one would have $\{-2,-1,0,1,2\}^m$.

2.2 Cost Function

A distinguishing element of MPC, when compared to other control algorithms, is that at each time instant k and for a given (measured or estimated) plant state $x(k)$, a cost function over a finite horizon of length N is minimized. The following choice encompasses many alternatives documented in the literature:

$$V(x(k),\mathbf{u}'(k)) \triangleq F(x'(k+N)) + \sum_{\ell=k}^{k+N-1} L(x'(\ell),u'(\ell)). \tag{5}$$

Here, $L(\cdot,\cdot)$ and $F(\cdot)$ are weighting functions, which serve to penalize predicted system behaviour, e.g., differences between voltage references and predicted values, see Sect. 2.4. Predicted plant state values, $x'(\ell)$, are formed using the system model (1):

$$x'(\ell+1) = f(x'(\ell), u'(\ell)), \quad \ell \in \{k, k+1, \ldots, k+N-1\} \tag{6}$$

where

$$u'(\ell) \in \mathbb{U}, \quad \ell \in \{k, k+1, \ldots, k+N-1\}$$

refers to tentative plant inputs (to be decided). The recursion (6) is initialized with the current plant state measurement (or estimate), i.e.:

$$x'(k) \longleftarrow x(k).$$

Thus, (6) refers to predictions of the plant states that would result if the plant inputs at the update times $\{k, k+1, \ldots, k+N-1\}$ were set equal to the corresponding values in

$$\mathbf{u}'(k) \triangleq \big\{u'(k), u'(k+1), \ldots, u'(k+N-1)\big\}. \tag{7}$$

Both, the predicted plant state trajectory and the plant inputs are constrained in accordance with (2), i.e., we have:

$$\begin{aligned} u'(\ell) &\in \mathbb{U}, \quad \forall \ell \in \{k, k+1, \ldots, k+N-1\} \\ x'(\ell) &\in \mathbb{X}, \quad \forall \ell \in \{k+1, k+2, \ldots, k+N\}. \end{aligned}$$

Constrained minimization of $V(\cdot,\cdot)$ in (5) gives the optimizing control sequence at time k and for state $x(k)$:

$$\mathbf{u}^{\mathrm{opt}}(k) \triangleq \big\{u^{\mathrm{opt}}(k), u^{\mathrm{opt}}(k+1;k), \ldots, u^{\mathrm{opt}}(k+N-1;k)\big\}. \tag{8}$$

It is worth emphasizing here that, in general, plant state predictions, $x'(\ell)$, will differ from actual plant state trajectories, $x(\ell)$. This is a consequence of possible model inaccuracies and the moving horizon optimization paradigm described next.

2.3 Moving Horizon Optimization

Despite the fact that the optimizer $\mathbf{u}^{\mathrm{opt}}(k)$ in (8) contains feasible plant inputs over the entire horizon, $(kh, (k+N-1)h]$, in most MPC approaches, only the first element is used, i.e., the system input is set to

$$u(k) \longleftarrow u^{\mathrm{opt}}(k).$$

At the next sampling step, i.e., at discrete-time $k+1$, the system state $x(k+1)$ is measured (or estimated), the horizon is shifted by one step, and another optimization is carried out. This yields $\mathbf{u}^{\mathrm{opt}}(k+1)$ and its first element provides $u(k+1) = u^{\mathrm{opt}}(k+1)$, etc. As illustrated in Fig. 3 for a horizon length $N=3$, the horizon taken into account in the minimization of V slides forward as k increases. The design of observers for the system state lies outside the scope of the present

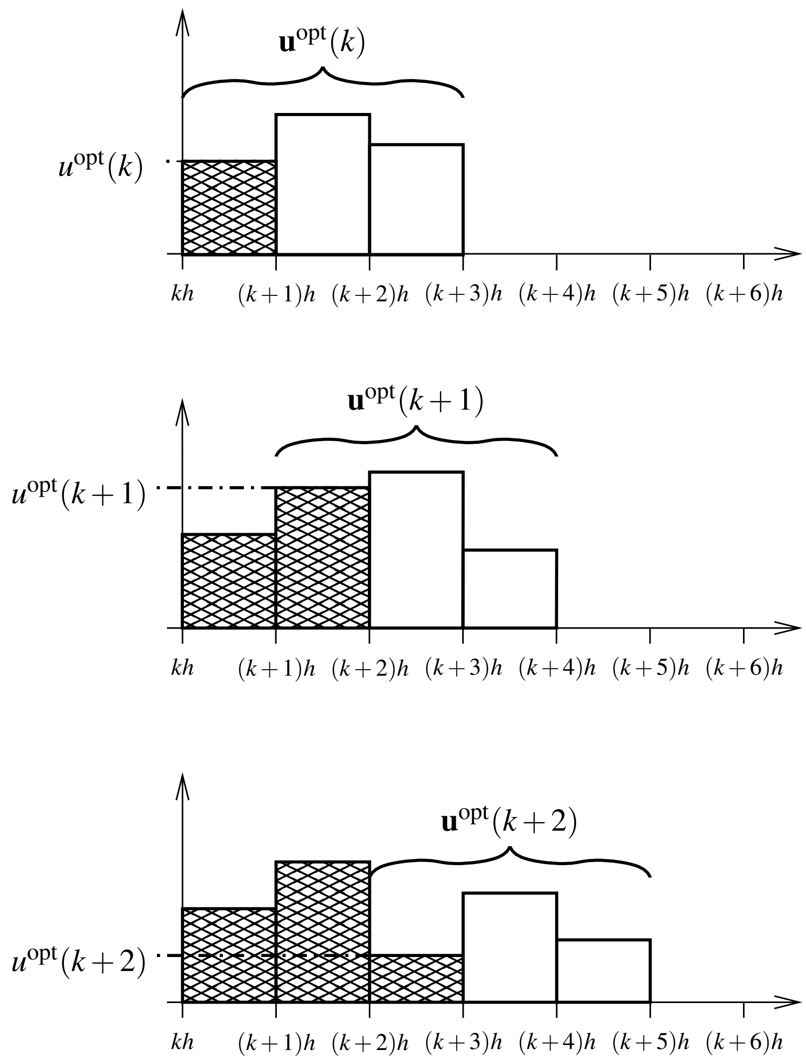


Fig. 3 Moving horizon principle with horizon $N = 3$.

chapter. The interested reader is referred to [2, 29, 41], which illustrate the use of Kalman filters for MPC formulations in power electronics.

2.4 Design Parameters

As seen above, MPC allows one to treat multi-variable nonlinear systems in an, at least conceptually, simple way. In addition to choosing the sampling interval h (which, amongst other things, determines the system model (1)), MPC design essentially amounts to selecting the cost function, i.e., the weighting functions $F(\cdot)$ and $L(\cdot,\cdot)$, and the horizon length N.

The design of the weighting functions $F(\cdot)$ and $L(\cdot,\cdot)$ should take into account the actual control objectives and may also consider stability issues [3, 4, 84]. [1] For example, tracking of desired output and internal voltages and currents can be accommodated into the MPC framework by choosing weights which penalize a measure of the difference between predicted and reference values.

For a given sampling frequency $1/h$ and, especially for systems with inverse response, larger values for the horizon length N will in general provide better performance, as quantified by the weighting functions $F(\cdot)$ and $L(\cdot,\cdot)$. Indeed, one can expect that, for large enough N, the effect of $u(k)$ on $x'(\ell)$ for $\ell > k+N$ will be negligible and, consequently, MPC will approximate the performance of an infinite horizon optimal controller [53]. On the other hand, the constrained optimization problem which, in principle, needs to be solved on-line to find the controller output, has computational complexity which, in general, increases with the horizon length. As a consequence, the optimization horizon parameter N allows the designer to trade-off performance versus on-line computational effort. Fortunately, excellent performance can often be achieved with relatively small horizons. In fact, in most applications of MPC to power electronics and electrical drives a horizon $N = 1$ is chosen. We will next present key aspects of *horizon-one MPC*. In Section 4, we will then discuss specific optimization algorithms which allow MPC with larger horizons to be implemented in practical power electronics and drive applications.

3 Horizon-One Predictive Control

In the academic field of power electronics, it is most common to focus on one-step horizon formulations when using predictive controllers. This comes from the fact that horizon-one solutions are easy to obtain and often give satisfactory results.

In this section, we present some basic concepts on one-step predictive control formulations used in power electronics. Our focus is on power converters and electrical drive systems, which can be modeled in discrete-time as

$$x(k+1) = Ax(k) + Bu(k), \tag{9}$$

where $x \in \mathbb{R}^n$ stands for the n-system states (e.g. voltages and currents) and $u \in \mathbb{R}^m$ stands for the m-control inputs (e.g. duty cycles or power switches).

[1] Note that the weighting functions should be chosen such that $V(\cdot,\cdot)$ depends on the decision variables contained in $\mathbf{u}'(k)$, see (7).

For our subsequent analysis, it is convenient to focus on regulation problems with a constant reference, say $x^\star \in \mathbb{R}^n$. By setting

$$x(k+1) = x(k) = x^\star,$$

it follows that

$$x^\star = Ax^\star + Bu^\star \Rightarrow x^\star = (I-A)^{-1}Bu^\star,$$

where $u^\star \in \mathbb{R}^m$ is the required input to maintain $x^\star$.

If we now introduce the system state and input tracking errors as

$$\begin{aligned} \hat{x} &\triangleq x - x^\star, \qquad \text{and} \\ \hat{u} &\triangleq u - u^\star, \end{aligned} \tag{10}$$

respectively, then it is easy to see that

$$\hat{x}(k+1) = x(k+1) - x^\star = A(\hat{x}(k) + x^\star) + B(\hat{u}(k) + u^\star) - x^\star.$$

Since $x^\star = Ax^\star + Bu^\star$, we obtain the system model:

$$\hat{x}(k+1) = A\hat{x}(k) + B\hat{u}(k), \tag{11}$$

In the sequel, we shall refer to $\hat{x} \in \mathbb{R}^n$ as the system state, whereas $\hat{u} \in \mathbb{R}^m$ is the control input. Consequently, the control goal becomes one of leading the system (11) to the origin. This is equivalent to leading the original system (9) to the desired reference, $x^\star$.

3.1 Deadbeat Control with a Modulator

One of the earliest control strategies referred to as “predictive control” in the power electronics community are deadbeat controllers. These use a discrete-time system model to calculate, at each sampling instant, the required control input to lead the system output to some desired value in a finite number of time steps. Generally, this input is in the form of a voltage reference, which is then modulated as described in Section 2.1. Deadbeat control has been applied to current control in three-phase inverters [70, 125, 127], rectifiers [78, 91], active filters [59, 87], DC-DC converters [114], and torque control of induction machines [19].

This control technique is normally used to govern power converters and electrical drives by obtaining the required input to achieve the desired system reference in only one sampling instant. Nevertheless, for some class of power converters, it is not possible to achieve the control target in just one sampling instant and more time-steps need to be considered. A useful concept to understand this issue is called reachability, which is defined as follows.

Definition 1 (Reachability [5]). The system (11) is reachable if it is possible to find a control sequence, say

$$\mathbf{u} = \begin{bmatrix} u(0) \\ u(1) \\ \vdots \\ u(\ell-1) \end{bmatrix}, \quad \ell \in \mathbb{N}$$

such that an arbitrary state, $x^\star$, can be reached from an initial state, $x(0)$, in a finite time, i.e., ℓ-sampling steps. ■

Now, we introduce the so-called reachability matrix, used to determined the system reachability, which is represented via

$$W_\ell = \left[A^{\ell-1}B \; \dots \; AB \; B\right]. \tag{12}$$

The following theorem follows from the preceding definition.

Theorem 1 (Reachability [5]). *The system* (11) *is reachable in ℓ sampling steps if and only if the reachability matrix W_ℓ has rank n, where n is the number of system states.* ■

Assuming that the initial state, $x(0)$, is known, the system state at the sampling time ℓ is given by

$$x(\ell) = A^\ell x(0) + W_\ell \mathbf{u}. \tag{13}$$

As stated in Theorem 1 above, if W_ℓ has rank n, then it is possible to obtain $\mathbf{u}$ by solving a system of n linearly independent equations. This sequence leads the initial system state, $x(0)$, to the desired final state value, $x^\star$.

It is important to emphasize that, based on Theorem 1, for the one-step case, where $\ell = 1$, we have that

$$W_1 = B.$$

Thus, if one wants to lead the system state from $x(0)$ to $x^\star$ in only one sampling instant, then the number of control inputs must be equal or larger than the number of system states, i.e., $m \geq n$. Notice that if there are more control inputs than system states, i.e., $m > n$, then the solution to $Ax(0) + W_1\mathbf{u} = 0$, see (13), is not unique.

Following, based on the above discussion, two particular cases are analyzed.

Invertible Matrix B

In this case, we focus on power converters which present the same number of system states as control inputs, i.e., $n = m$. This occurs for several converter topologies, e.g., a three-phase inverter in $\alpha\beta$ coordinates with an rl-load, which has 2-inputs and 2-outputs [113] .

Here, B is a square matrix which, if nonsingular, is invertible. Moreover, the one-step reachability matrix, $W_1 = B$, has rank n. Therefore, it is possible to lead the system (11) to $x^\star$ in one sampling instant. The one-step deadbeat control law for this case is expressed by:

$$\begin{aligned}\hat{u}(k) &= -B^{-1}A\hat{x}(k),\\ u(k) &= -B^{-1}A(x(k)-x^\star)+u^\star.\end{aligned} \tag{14}$$

Non-Invertible Matrix B

Here, we consider the case where the system (11) has less control inputs than system states, e.g., a three-phase inverter with an LC-filter [21]. Thus, matrix $B \in \mathbb{R}^{n\times m}$, with $n > m$, is not invertible. Moreover, the one-step reachability matrix $W_1 = B$ has rank smaller than n. However, if matrix B has rank m, then $B^TB \in \mathbb{R}^{m\times m}$ is invertible. Therefore, one can obtain the, so called, Moore-Penrose pseudoinverse matrix, which is given by

$$B^\dagger = (B^TB)^{-1}B^T;$$

see, e.g., [5]. Thus, the minimum time deadbeat control law can be expressed via:

$$\begin{aligned}\hat{u}(k) &= -B^\dagger A\hat{x}(k) = -(B^TB)^{-1}B^TA\hat{x}(k),\\ u(k) &= -(B^TB)^{-1}B^TA(x(k)-x^\star)+u^\star.\end{aligned} \tag{15}$$

The minimum number of sampling steps required to achieve a desired reference can be determined by adding terms of the form $A^\ell B$, until the reachability matrix W_ℓ in (12) has full rank n [2].

Notice that if the system inputs are constrained to a bounded set, e.g., $u(k) \in [-1,1]^m$, then it is necessary to saturate the control input when the system is far from the reference yielding:

$$\begin{aligned}\hat{u}(k) &= \text{sat}\left[-B^\dagger A\hat{x}(k)\right], & \hat{u} &\in [u_{\min}-u^\star, u_{\max}-u^\star]\\ u(k) &= \text{sat}\left[-B^\dagger A(x(k)-x^\star)+u^\star\right], & u &\in [u_{\min}, u_{\max}]\end{aligned}$$

Thus, the desired system reference may not be reached in one sampling instant whenever the control input is saturated.

While this method has been used when a fast dynamic response is required, being deadbeat-based, it is often fragile. Indeed, uncertainties such as model errors, unmodeled delays, and external disturbances may often deteriorate the closed-loop performance. In the literature, there exist some works addressing these issues. For example, in [86] an adaptive self tuning deadbeat controller has been proposed to deal with system parameter uncertainties and compensate the calculation delay. On the other hand, in [83] a disturbance observer has been included to improve the disturbance rejection of the closed-loop system.

[2] It is important to remark that in the case that $B \in \mathbb{R}^{n\times n}$, with B nonsingular, we have that $B^\dagger = B^{-1}$. Thus, the control law (14) is a particular case of (15).

3.2 Horizon-One MPC with a Finite Control Set

One of the most popular predictive control strategy for power converters and drives is FCS-MPC [20, 112]. This predictive control strategy explicitly considers the power switches, $S(k)$ in the optimization by means of a finite control set constraint. Modulation stages are not needed.

In general, large prediction horizons are preferable when using MPC. However, finding the optimal input sequence for the finite control set case typically requires one to solve a combinatorial optimization problem. Interestingly, for some topologies, one-step horizon MPC provides good closed-loop performance [66, 111]. In this section, we discuss the optimal solution to this predictive control strategy.

3.2.1 Cost Function

When using MPC for power electronics, it is often desirable to minimize the tracking error of the system state, which includes variables of different physical nature and order of magnitude, e.g., currents, voltages, torques, power. It is therefore convenient to adopt a cost function, which considers a weighted positive sum of the tracking errors of the controlled variables, see, e.g., [112]. This particular class of cost function can be, in general, expressed via:

$$V = \lambda_1(x_1(k+1)-x_1^\star)^2 + \lambda_2(x_2(k+1)-x_2^\star)^2 + \ldots + \lambda_n(x_n(k+1)-x_n^\star)^2, \quad (16)$$

where λ_i are the weighting factors, which allow the designer to trade-off among the different system state tracking errors. For example, for a two-level three-phase inverter, in $\alpha\beta$ orthogonal coordinates, one can use (see [113])

$$V_{2LI} = \lambda_1(i_\alpha(k+1)-i_\alpha^\star)^2 + \lambda_2(i_\beta(k+1)-i_\beta^\star)^2,$$

where $\lambda_1 = \lambda_2 = 1$. For a one-phase three-cell Flying Capacitor Converter (FCC) one can choose (see, e.g., [74])

$$V_{FCC} = \lambda_1(i_a(k+1)-i_a^\star)^2 + \lambda_2(v_{c1}(k+1)-v_{c1}^\star)^2 + \lambda_3(v_{c2}(k+1)-v_{c2}^\star)^2.$$

The above cost functions can be expressed as

$$V(x(k),u(k)) = \hat{x}^T(k+1)P\hat{x}(k+1) \quad (17)$$

where $\hat{x}(k) = x(k) - x^\star$ represents the state tracking errors of system (11), and $P = \mathrm{diag}\{\lambda_1,\ldots,\lambda_n\}$ is the weighting matrix.

In this case, it is also assumed that the power converter to be controlled is modeled as per (9), i.e.,

$$x(k+1) = Ax(k) + Bu(k), \quad (18)$$

where u represents the m-control inputs, which belong to a finite set of p-elements, i.e.,

$$u \in \mathbb{U} \triangleq \{u_1, \ldots, u_p\} \subset \mathbb{R}^m. \tag{19}$$

3.2.2 Unconstrained Optimum

To derive a closed form solution for FCS-MPC with horizon $N = 1$, we note that given (9), the quadratic cost function (17) can be expanded via

$$V(x(k), u(k)) = \hat{x}^T(k) A^T P A \hat{x}(k) + \hat{u}^T(k) B^T P B \hat{u}(k) + 2\hat{u}^T(k) B^T P A \hat{x}(k), \tag{20}$$

where $\hat{u}(k) = u(k) - u^\star$, as before. If there were no control constraints, i.e., $\hat{u}(k) \in \mathbb{R}^m$, then the unconstrained optimal solution can be obtained as follows:

$$\frac{\partial V(\hat{x}(k), \hat{u}(k))}{\partial \hat{u}(k)} = 2B^T P B \hat{u}(k) + 2B^T P A \hat{x}(k) = 0$$

Thus, the minimizer to (17), without taking into account any system constraints, is given by

$$\begin{aligned} \hat{u}_{uc}^{\mathrm{opt}}(k) &= -K\hat{x}(k), \\ u_{uc}^{\mathrm{opt}}(k) &= -K(x(k) - x^\star) + u^\star, \end{aligned} \tag{21}$$

where

$$K = (B^T P B)^{-1} B^T P A. \tag{22}$$

It is worth noting that this unconstrained solution, will normally, not belong to the finite set (19).

To obtain the constrained optimal solution, $u^{\mathrm{opt}}(k) \in \mathbb{U}$, it is convenient to introduce the following auxiliary variable:

$$z(k) \triangleq u(k) - u_{uc}^{\mathrm{opt}}(k) = \hat{u}(k) - \hat{u}_{uc}^{\mathrm{opt}}(k). \tag{23}$$

In terms of $z(k)$, the cost function in (20) can be expressed via:

$$V(x(k), u(k)) = g(x(k)) + z^T(k) H z(k), \tag{24}$$

where the term $g(x(k))$ is independent of $u(k)$ and

$$H \triangleq B^T P B. \tag{25}$$

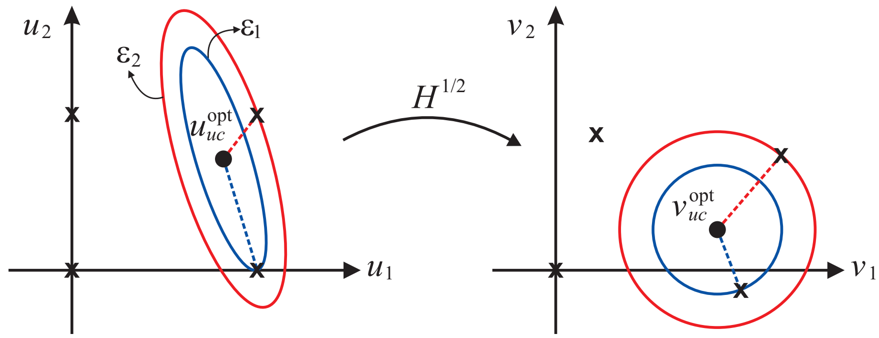


Fig. 4 Geometrical representation of FCS-MPC optimal solution; $u_1, u_2 \in \{0, 1\}$.

3.2.3 Closed-Form Solution

To obtain the optimal finite set constrained solution, one must find the control input which minimizes $V(x(k), u(k))$. From (24), it follows that level sets of the cost function are ellipsoids, where the eigenvectors of H define the principal directions of the ellipsoid. Thus, the constrained optimizer $u^{\text{opt}}(k)$ does not necessarily correspond to the nearest neighbour of $u_{uc}^{\text{opt}}(k)$ within the constraint set $\mathbb{U}$.

Example 1. Consider the case where a power converter, modeled as per (18), has 2 power switches, which can take only two values, i.e., $u_1, u_2 \in \{0, 1\}$. Thus, the control input belongs to the following finite set:

$$u \in \mathbb{U} \triangleq \left\{ \begin{bmatrix} 0 \\ 0 \end{bmatrix}, \begin{bmatrix} 0 \\ 1 \end{bmatrix}, \begin{bmatrix} 1 \\ 0 \end{bmatrix}, \begin{bmatrix} 1 \\ 1 \end{bmatrix} \right\} \subset \mathbb{R}^2. \tag{26}$$

The vectorial representation of the optimal solution is depicted in Fig. 4. Here, the ellipses, ε_i centered in u_{uc}^{opt}, represent all the points that lead to the same cost. Formally, if $a, b \in \varepsilon_i$ then, $V(x(k), a) = V(x(k), b)$.

As we move away from the centre, the ellipses become larger, increasing the cost function value, i.e., if $a \in \varepsilon_1$ and $b \in \varepsilon_2$ then, $V(x(k), a) < V(x(k), b)$. Thus, in this example, the optimal solution, which produces the minimum cost function value is $u^{\text{opt}} = [1\ 0]^T$, despite the nearest vector to the unconstrained solution being $u = [1\ 1]^T$. Clearly, the optimal solution is, in general, not the nearest neighbour to the unconstrained solution. ■

To obtain the optimal solution, we use the following transformation [104]:

$$v = H^{1/2} u, \quad v \in \mathbb{V} \triangleq H^{1/2} \mathbb{U}.$$

Now, the cost function (24) can be expressed as:

$$V(x(k), v(k)) \triangleq g(x(k)) + (v(k) - v_{uc}^{\text{opt}}(k))^T (v(k) - v_{uc}^{\text{opt}}(k)), \tag{27}$$

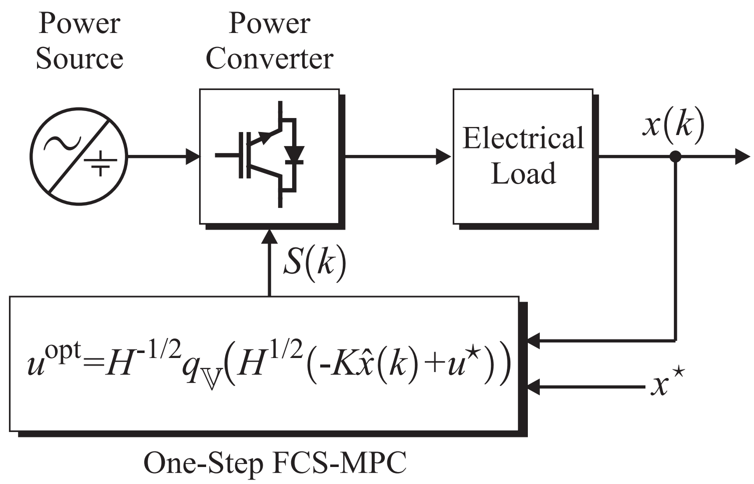


Fig. 5 One-step FCS-MPC closed-loop.

where

$$v_{uc}^{\text{opt}}(k) \triangleq H^{1/2}u_{uc}^{\text{opt}}(k).$$

Thus, using this transformation, the level sets of the cost function describe spheres centered in v_{uc}^{opt}, as depicted in Fig. 4. Therefore, in terms of these transformed variables, the nearest vector to the unconstrained solution, $v_{uc}^{\text{opt}}(k)$, is indeed the (constrained) optimal solution.

Definition 2 (Vector Quantizer (see e.g. [31])). Consider a set $\mathscr{A} \subseteq \mathbb{R}^n$ and a finite set $\mathscr{B} \triangleq \{b_1, \ldots, b_p\} \subset \mathbb{R}^n$. A function $q_{\mathscr{B}}(\cdot) : \mathscr{A} \to \mathscr{B}$ is an Euclidean vector quantizer if $q_{\mathscr{B}}(a) = b_i \in \mathscr{B}$ if and only if b_i satisfies that $|\, a - b_i \,| \leq |\, a - b_j \,|$, for all $b_j \neq b_i$, where $b_j \in \mathscr{B}$. ■

Using the vector quantizer presented in Definition 2, it follows that

$$v^{\text{opt}}(k) = q_{\mathbb{V}}\left(H^{1/2}u_{uc}^{\text{opt}}(k)\right), \quad v^{\text{opt}}(k) \in \mathbb{V}.$$

Finally, the actual optimal solution, which minimizes the cost function (17), is given by:

$$u^{\text{opt}}(k) = H^{-1/2}v^{\text{opt}}(k) = H^{-1/2}q_{\mathbb{V}}\left(H^{1/2}u_{uc}^{\text{opt}}(k)\right), \quad u^{\text{opt}}(k) \in \mathbb{U}. \tag{28}$$

A block diagram of the resulting one-step FCS-MPC closed-loop is depicted in Fig. 5. Notice that $q_{\mathbb{V}}(\cdot)$ in (28) implies that the quantization considers the vectors from the finite set $\mathbb{V} = H^{1/2}\mathbb{U}$, i.e., $q_{\mathbb{V}}(a) \in \mathbb{V}$.

3.2.4 Relationship to Deadbeat Control and Embellishments

It is interesting and instructive, to compare horizon-one predictive control with deadbeat control. Here, we first notice that (21) is a weighted version of (15). In fact, when $P = I$, where I denotes the identity matrix, both control values are the same. Thus, horizon-one FCS-MPC, with the simple cost function presented in (16) and (17), is a class of quantized deadbeat controller. Intuitively, such a predictive control methodology presents features akin to those of a deadbeat controller, i.e., a fast dynamic response but poor robustness.

To mitigate this problem, it is convenient to include control weighting in the cost function. This will lead to a less aggressive, more cautious, controller. In this case, the cost function can be formulated as[3]

$$V(x(k),u(k)) = \hat{u}^T(k)R\hat{u}(k) + \hat{x}^T(k+1)P\hat{x}(k+1), \tag{29}$$

see (10) and where R is a positive definite matrix, which can be used as a tuning parameter. With a larger matrix R we are seeking to apply an input close to $u^\star$. This allows one to reduce the control action that is applied to the system, leading to a slower dynamic response with often better robustness properties.

To obtain the optimal solution of this embellished horizon-one FCS-MPC formulation, one can roughly follow the analysis presented in Subsection 3.2.3. The optimal solution has the same structure as presented in (28) (see also (21)), but where

$$H = B^T PB + R, \tag{30}$$

and

$$K = H^{-1}B^T PA, \tag{31}$$

see also [104]. Notice that, the control gain matrix K is reduced as R is increased. Thus, for the same system tracking error, $\hat{x}(k)$, this new formulation produces a less aggressive actuation.

Example 2. If H is diagonal, i.e., $H = \alpha I$, with $\alpha \in \mathbb{R}$, then the cost in (24) becomes

$$\begin{aligned} V(x(k),u(k)) &= g(x(k)) + z^T(k)Hz(k), \\ &= g(x(k)) + \alpha^2(u(k) - u_{uc}^{\text{opt}}(k))^T(u(k) - u_{uc}^{\text{opt}}(k)). \end{aligned} \tag{32}$$

The level sets of this cost function are spheres centered in $u_{uc}^{\text{opt}}(k)$. Therefore, one can directly quantize the unconstrained solution $u_{uc}^{\text{opt}}(k)$, to obtain the finite optimal solution without performing any transformation, i.e.,

$$u^{\text{opt}}(k) = q_{\mathbb{U}}\left(u_{uc}^{\text{opt}}(k)\right) = q_{\mathbb{U}}\left(-K\hat{x}(k) + u^\star\right) \in \mathbb{U}.$$

Fig. 6 illustrates a block diagram for this particular horizon-one FCS-MPC closed-loop when $H = \alpha I$.

[3] Alternatively, one can also penalize the size of the increments of the control input via a term of the form $(u(k) - u(k-1))^T R(u(k) - u(k-1))$.

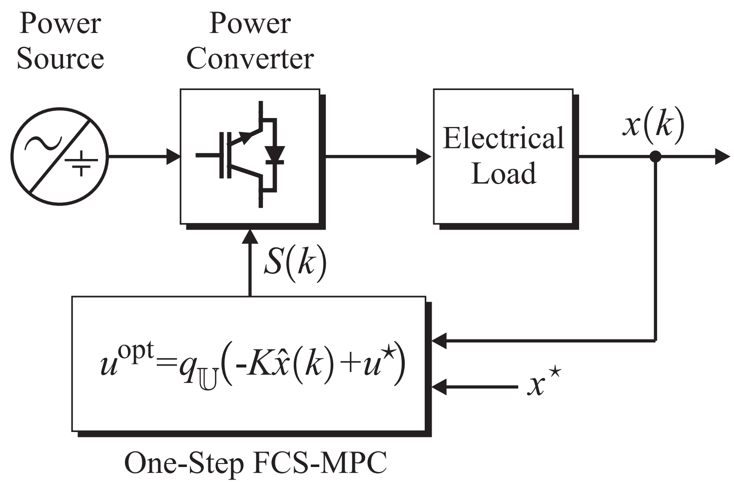


Fig. 6 One-step FCS-MPC closed-loop with a diagonal matrix H.

4 Predictive Control with Long Horizons

In the context of MPC, long prediction horizons yield in general a better closed-loop performance than short horizons. In particular, extending the length of the prediction horizons reduces the cost associated with the objective function in closed-loop operation [53, 103]. Indeed, the use of very long prediction horizons is typically preferred. In particular, the infinite horizon case often ensures closed-loop stability, provided that a solution with a finite cost exists [77, 108].

In the field of power electronics and electrical drives, the benefits of long prediction horizons can be highlighted using the following three examples. First, for the direct MPC formulation outlined in Section 4.5.2, very long prediction horizons of 100 and more time-steps yield a superior steady-state performance, in the sense that current THDs similar to the one obtained by employing optimized pulse patterns (OPPs) can be achieved, when assuming the same device switching losses. In contrast to that, very short horizons tend to lead to steady-state performance results that are inferior to carrier-based PWM [35, 115].

Second, for systems with an inverse behavior[4], long prediction horizons are required during transient operation to ensure the tracking of the reference signals and to avoid closed-loop instability. This was illustrated in [61] for a boost converter with a single voltage control loop (without an underlying current control loop) that directly manipulates the switch position.

Third, and as already mentioned in Section 3, very short horizons lead to a closed-loop behavior similar to the one typically obtained by deadbeat control, particularly when the penalty on the manipulated variable is small or set to zero. Deadbeat con-

[4] When considering linear time-invariant (LTI) systems, *inverse behavior* is equivalent to non-minimum phase behavior, i.e. systems with zeros in the right half-plane of the Laplace domain.

trol is known to be highly sensitive to measurement and estimation noise, as well as parameter uncertainties of the system model. Long prediction horizons, on the other hand, significantly reduce the sensitivity of the controller to noise and—as a result—improve the performance during steady-state operation [39].

Unfortunately, the computational burden associated with solving the optimization problem underlying MPC increases exponentially with the length of the prediction horizon. At the same time, the sampling intervals typically required in power electronic systems are very short, often amounting to a few tens of μs. From a computational point of view, this makes the solution of MPC problems with long prediction horizons very challenging. Indeed, the belief in the power electronics community is widespread that MPC problems with long horizons cannot be solved in real time. Nevertheless, control and optimization techniques are available that reduce the computational burden to a level, which allows the solution of MPC problems on today's available computational platforms within sampling intervals of less than $100\,\mu$s. In this section, several such techniques will be presented along with corresponding power electronics examples, for which they have been applied to. Selected experimental results are highlighted as well.

4.1 Linear Quadratic MPC for Converters with a Modulator

A particularly simple case of (5)–(6) arises when the cost function is quadratic and the system model is linear, i.e.:

$$\begin{aligned} V(x(k),\mathbf{u}(k)) &= x'^T(k+N)Px'(k+N) \\ &\qquad + \sum_{\ell=k}^{k+N-1}\left\{x'^T(\ell)Qx'(\ell)+u'^T(\ell)Ru'(\ell)\right\}, \\ x'(\ell+1) &= Ax'(\ell)+Bu'(\ell), \\ &\quad x'(\ell)\in\mathbb{X}\subseteq\mathbb{R}^n,\ u'(\ell)\in\mathbb{U}\subseteq\mathbb{R}^m, \quad \ell\in\{0,1,\dots N\}, \end{aligned} \tag{33}$$

where A and B denote the state-update matrices of a linear time-invariant system, and P, Q and R are penalty matrices of appropriate dimensions. More specifically, P and Q are positive semi-definite matrices and R is a positive definite matrix. The constraint sets $\mathbb{X}$ and $\mathbb{U}$ are polyhedra, given by the intersection of a finite number of half-spaces, which are defined by hyperplanes. In such a setup, a PWM is typically used.

By successively using the state-update equation in (33) and assuming a finite N, the state vector at time-step $\ell+1$ can be represented as a function of the state vector at time-step k and the control sequence as follows:

$$\begin{bmatrix} x'(k) \\ x'(k+1) \\ \vdots \\ \vdots \\ x'(k+N) \end{bmatrix} = \begin{bmatrix} I \\ A \\ \vdots \\ \vdots \\ A^N \end{bmatrix} x(k) + \begin{bmatrix} 0 & \cdots & \cdots & 0 \\ B & 0 & \cdots & 0 \\ AB & B & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ A^{N-1}B & A^{N-2}B^2 & \cdots & B \end{bmatrix} \begin{bmatrix} u'(k) \\ u'(k+1) \\ \vdots \\ \vdots \\ u'(k+N-1) \end{bmatrix}. \tag{34}$$

We rewrite this expression in compact form as

$$\mathbf{x}(k) = \mathbf{S}x(k) + \mathbf{T}\mathbf{u}(k), \tag{35}$$

with $\mathbf{S}$ and $\mathbf{T}$ appropriately defined. Introducing $\mathbf{Q} \triangleq \text{diag}(Q, \ldots, Q, P)$ and $\mathbf{R} \triangleq \text{diag}(R, \ldots, R)$, the cost function in (33) can be rewritten as

$$V(x(k), \mathbf{u}(k)) = \mathbf{x}^T(k)\mathbf{Q}\mathbf{x}(k) + \mathbf{u}^T(k)\mathbf{R}\mathbf{u}(k). \tag{36}$$

Substituting (35) into (36) leads to[5]

$$V(x(k), \mathbf{u}(k)) = \mathbf{u}^T(k)H\mathbf{u}(k) + 2x^T(k)F\mathbf{u}(k), \tag{37}$$

with

$$\begin{aligned} H &= \mathbf{T}^T\mathbf{Q}\mathbf{T} + \mathbf{R} \\ F &= \mathbf{S}^T\mathbf{Q}\mathbf{T}. \end{aligned}$$

Note that $\mathbf{u}(k)$ is the real-valued optimization variable and $H = H^T$ is the positive definite Hessian matrix.

4.1.1 Unconstrained Solution

Similar to the horizon-one case treated in Section 3.2.2, if the system inputs and states in (33) are unconstrained, i.e. $\mathbb{X} = \mathbb{R}^n$ and $\mathbb{U} = \mathbb{R}^m$, then the moving horizon optimization problem (33) with the reformulated cost function (37) can be solved algebraically. This yields the control sequence as a linear function of the state vector according to $\mathbf{u}(k) = -\mathbf{K}x(k)$ with $\mathbf{K} = -H^{-1}F^T$. The control input at time-step k is obtained by taking the first element of this sequence, i.e. the linear state-feedback controller is of the form

$$u(k) = -K_0 x(k) \tag{38}$$

with $\mathbf{K} = [K_0^T \ K_1^T \ldots \ K_{N-1}^T]^T$. Directly related to this is to concept of generalized predictive control (GPC) [18]. The unconstrained finite-horizon approach was investigated in [64, 76] for use in electrical drive applications.

[5] Note that (37) contains the third term $x^T(k)\mathbf{S}^T\mathbf{S}x(k)$. Since this term is constant and independent of $\mathbf{u}(k)$, it can be omitted in the cost function without affecting the result of the optimization problem.

Optimality, however, does not ensure stability, motivating the use of an infinite horizon N. The problem (33) is then referred to as the *linear quadratic regulator* (LQR), and $K = K_0 = \ldots = K_{N-1}$ is the solution of an algebraic Riccati equation. LQR controllers have been proposed for a number of power electronics applications; early examples of such applications include dc-dc converters [30] and electrical drives [88].

4.1.2 Constrained Solution

On the other hand, if the inputs and states are constrained to belong to polyhedra, then (33) can be rewritten[6] in the form

$$\begin{aligned} &\min_{\mathbf{u}(k)} V(x(k),\mathbf{u}(k)) = \mathbf{u}^T(k)H\mathbf{u}(k) + 2x^T(k)F\mathbf{u}(k) \\ &\text{subj. to } G\mathbf{u}(k) \leq w + Ex(k)\,. \end{aligned} \tag{39}$$

with G and E being matrices of appropriate dimensions and w denoting a column vector. For a given state vector $x(k)$, (39) can be simplified to

$$\begin{aligned} &\min_{\mathbf{u}(k)} V(\mathbf{u}(k)) = \mathbf{u}^T(k)H\mathbf{u}(k) + 2c^T\mathbf{u}(k) \\ &\text{subj. to } G\mathbf{u}(k) \leq g\,, \end{aligned} \tag{40}$$

with $c = F^T x(k)$ and $g = w + Ex(k)$.

If $\mathbf{u}(k)$ satisfies the constraints $G\mathbf{u}(k) \leq g$ in (40), then $\mathbf{u}(k)$ is said to be a *feasible* solution. The problem (40) is feasible, if there exists at least one such feasible solution, else the problem is *infeasible*. Assuming feasibility, the optimization variable, for which the minimal (i.e. the optimal) value of the cost function is obtained, is the *optimizer* $\mathbf{u}^{\text{opt}}(k)$.

The form (40) is a convex mathematical optimization problem with a quadratic objective function and linear constraints, a so called *quadratic program* (QP) [51]. Such problems can be solved efficiently. Specifically, with H being positive definite, (40) can be solved in polynomial time [67].

QPs are typically solved using the interior point method [62, 90]. Other solution approaches include the active set [28] and gradient methods [89]. Examples of QP solvers include SeDuMi [119], CPLEX [58] and IpOpt [124]. Recently, first efforts have been reported in the literature to solve QPs in embedded systems, particularly when running on field-programmable gate arrays (FPGAs), see e.g. [24, 60, 109, 110]. Moreover, a simplified QP is solved algebraically in [76] to derive a position controller for a brushless dc drive.

Another example is model predictive pulse pattern control (MP^{3}C), which manipulates online the switching instants of pre-computed optimized pulse patterns

[6] For the cost function the constant term in (37) has been neglected. The inequality constraints can be derived by substituting (35) into the state constraints and adding the input constraints.

(OPPs) [14, 97] to achieve very fast closed-loop control. The underlying QP can be solved easily using a variation of the active set method [39]. First experimental results on a 1.1 MVA five-level inverter with a 6 kV medium-voltage induction machine are available [92].

4.2 The Explicit State-Feedback Control Law for Linear Quadratic MPC with a Modulator

Despite the ever growing computational power available and recent advances in implementing QP solvers on embedded system architectures, solving the QP in real-time for power electronics applications poses a highly challenging problem. When using sampling intervals in the μs range, the computation times needed to solve the QP typically exceed the sampling interval—often by one or two orders of magnitude. Rather than solving the mathematical optimization problem in real-time for the given state vector at the current time-step, the optimization problem can be solved offline for *all possible* states. Specifically, the so-called (explicit) *state-feedback control law* can be computed for all states $x(k) \in \mathbb{X}$ [9]. The explicit control law can be stored in a look-up table and the optimal control input can be read from the look-up table in a computationally efficient manner. We refer to this methodology as *explicit MPC*, in contrast to MPC, which has traditionally been solved entirely online.

4.2.1 The State-Feedback Control Law

Using the coordinate transformation

$$\mathbf{z}(k) = \mathbf{u}(k) + H^{-1}F^T x(k)\,, \tag{41}$$

(39) can be rewritten as

$$\min_{\mathbf{z}(k)} V(\mathbf{z}(k)) = \mathbf{z}^T(k) H \mathbf{z}(k) \tag{42a}$$

$$\text{subj. to } G\mathbf{z}(k) \leq w + Sx(k) \tag{42b}$$

with $S = E + GH^{-1}F^T$. Note that the optimization variable $\mathbf{z}(k)$ includes, in a linear manner, the control sequence $\mathbf{u}(k)$ and the state vector $x(k)$. Moreover, the constraints in (42b) depend linearly on the state vector $x(k)$.

As previously, for a given $x(k)$, the optimizer $\mathbf{z}^{\text{opt}}(k)$ can be computed. Using the notion of *sensitivity analysis*, we are interested in exploring the sensitivity of $\mathbf{z}^{\text{opt}}(k)$ to small perturbation in $x(k)$. In general, unless the set of active constraints changes in (42b), a small perturbation in $x(k)$ will lead to a small variation in $\mathbf{z}^{\text{opt}}(k)$. Because of the linear dependency in (41), in a small neighborhood around $x(k)$, the variation in $\mathbf{z}^{\mathrm{opt}}(k)$ depends linearly on the modification in $x(k)$. As a result, in this neighborhood, $\mathbf{z}^{\mathrm{opt}}(k)$ is a linear function of $x(k)$ plus an offset, a so called *affine* function. Rewriting (41) to express $\mathbf{u}^{\mathrm{opt}}(k)$ as a function of the optimizer and the state vector, we conclude that the control sequence is also affine in the state vector.

We refer to the neighborhood around $x(k)$, in which $\mathbf{z}^{\mathrm{opt}}(k)$ is an affine function of $x(k)$, as a *critical region*. Using the Karush-Kuhn-Tucker (KKT) conditions for optimality [63, 69], the critical region and its shape can be computed. Due to the linearity of (42b), the boundaries of the critical region are hyperplanes. More specifically, it can be shown that the critical region is a polyhedron, with its facets being defined by the active constraints. An algorithm can be constructed, which iteratively explores the whole set of states, $\mathbb{X}$, computes all critical regions and the associated affine control laws. By doing so, we treat the state vector as a parameter, which gives rise to *multi-parametric programming* and yields the state-feedback control law. In particular, (42) is a *multi-parametric quadratic program* (mp-QP).

Hereafter, the main results of mp-QP are recapitulated. For this, a number of definitions are required.

Definition 3 (Polyhedron). A polyhedron $\mathscr{P}$ is a set that is equal to the intersection of a finite number of half-spaces defined by hyperplanes. ■

Definition 4 (Polyhedral Partition). A collection of polyhedra

$$\mathscr{P}_i \subseteq \mathbb{X}, \quad i \in \{1, 2, \ldots, n\},$$

is a *polyhedral partition* of the polyhedron $\mathbb{X}$, if and only if $\bigcup_{i=1}^{n} \mathscr{P}_i = \mathbb{X}$ and $\mathscr{P}_i \cap \mathscr{P}_j$ is lower dimensional $\forall i \neq j$. ■

Theorem 2. *The solution to the mp-QP in* (42) *is a state-feedback control law $u^{opt}(k)$ that is a continuous and piecewise affine function of the state vector $x(k)$ defined on a polyhedral partition of the feasible state-space $\mathbb{X}$.* ■

More specifically, the feasible state-space is partitioned into polyhedra, and for each polyhedron the optimal control law $u^{\mathrm{opt}}(k)$ is given as an affine function of the state. Note that $u^{\mathrm{opt}}(k)$ is the first element in $\mathbf{u}^{\mathrm{opt}}(k)$, as defined in (7).

Theorem 3. *The value function $V^{opt}(x(k)) = V(x(k), \mathbf{u}^{opt}(k))$ of the mp-QP* (42) *is continuous, convex and piecewise quadratic in the state.* ■

More details about multi-parametric programming for QPs and the proofs of the above theorems can be found in [9, 121]. Related results were obtained in [116]. When the cost function in (42a) is linear, a *multi-parametric linear program* (mp-LP) results. The state-feedback control law is, as in the mp-QP case, continuous and piecewise affine in the state. The value function, on the other hand, is convex and piecewise affine. For more details on mp-LPs, the reader is referred to [7, 13]. Multi-parametric programs can be solved efficiently using the multi-parametric toolbox (MPT) [71]. This versatile and numerically robust toolbox is available for free on http://control.ee.ethz.ch/ mpt/.

4.2.2 Implementation Aspects

The state-feedback control law, which is the result of mp-LP or mp-QP, is of the form

$$u^{\text{opt}}(k) = \begin{cases} K_1 x(k) + f_1 & \text{if } G_1 x(k) \leq g_1 \\ \vdots & \vdots \\ K_n x(k) + f_n & \text{if } G_n x(k) \leq g_n \end{cases} \tag{43}$$

with $K_i x(k) + f_i$ denoting the ith affine control law and $G_i x(k) \leq g_i$ the corresponding polyhedron. Such a state-feedback controller can be easily implemented and evaluated online. In a first step, given the state vector $x(k)$, the polyhedron needs to be determined, in which the estimated or measured state lies. The brute force approach is to go through all polyhedra and to check the corresponding inequalities. An alternative approach is to build a binary search tree as proposed in [122]. Such a search tree reduces the online computational demand at the expense of an increased memory requirement. In a second step, after the correct polyhedron has been identified, the affine control law is read out and the optimal control input $u^{\text{opt}}(k)$ is computed.

In general, polyhedra with the same control law form convex unions and can thus be merged and replaced by their unions. This leads to an equivalent piecewise affine control law that features a lower number of polyhedra and thus a reduced complexity. From an implementation point of view, such a representation is highly desirable, since it relaxes the memory requirements and reduces the computational burden. Indeed, by adopting the notion of hyperplane arrangements, an equivalent piecewise affine control law can be derived that is *minimal* in the number of polyhedra [48]. We refer to this concept as *optimal complexity reduction*. For control laws with hundreds of polyhedra, the number of polyhedra can often be reduced by an order of magnitude.

4.2.3 An Illustrative Example of the State-Feedback Control Law

To further illustrate the derivation and properties of the explicit state-feedback control law of MPC, consider a dc-dc buck converter, as shown in Fig. 7. Using the classic technique of averaging between the on and off modes of the circuit, the discrete-time system model

$$x(k+1) = Ax(k) + Bv_s d(k) \tag{44}$$

can be obtained, where v_s denotes the unregulated input voltage and $d(k)$ the duty cycle. The state vector contains the inductor current i_ℓ and the output voltage v_o, i.e. $x = [i_\ell \; v_o]^T$. The continuous-time system matrices are

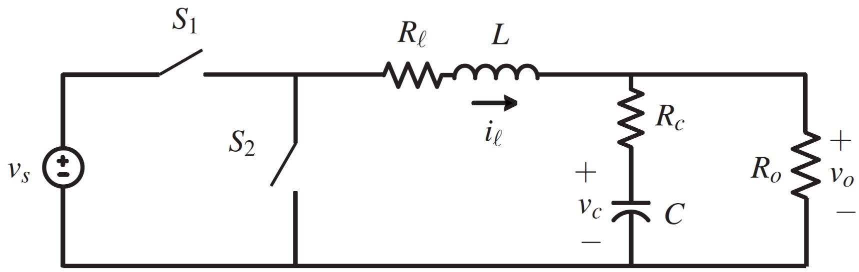


Fig. 7 Topology of the step-down synchronous converter

$$F = \begin{bmatrix} -\frac{R_\ell}{L} & -\frac{1}{L} \\ \frac{R_o}{R_o+R_c}\frac{L-R_cR_\ell C}{LC} & -\frac{1}{R_o+R_c}\frac{L+R_cR_oC}{LC} \end{bmatrix}, \quad G = \begin{bmatrix} \frac{1}{L} \\ \frac{R_o}{R_o+R_c}\frac{R_c}{L} \end{bmatrix}, \tag{45}$$

and their discrete-time representations are given by

$$A = e^{Fh}, \quad B = \int_0^h e^{F\tau} G \, d\tau, \tag{46}$$

where h denotes the sampling interval. Adopting the per unit (pu) system, the parameters in (45) are the inductor $L = 3\,\text{pu}$, capacitor $C = 20\,\text{pu}$ and output resistor $R_o = 1\,\text{pu}$. The internal resistor of the inductor is $R_\ell = 0.05\,\text{pu}$ and the equivalent series resistance of the capacitor is $R_c = 0.005\,\text{pu}$. The nominal input voltage is assumed to be $v_s = 1.8\,\text{pu}$.

To allow for variations in the input voltage, it is convenient to scale the system equations by v_s, as proposed in [42]. To this end, we define $\tilde{i}_\ell = i_\ell / v_s$, $\tilde{v}_o = v_o / v_s$ and $\tilde{x} = [\tilde{i}_\ell \ \tilde{v}_o]^T$, and rewrite (44) as

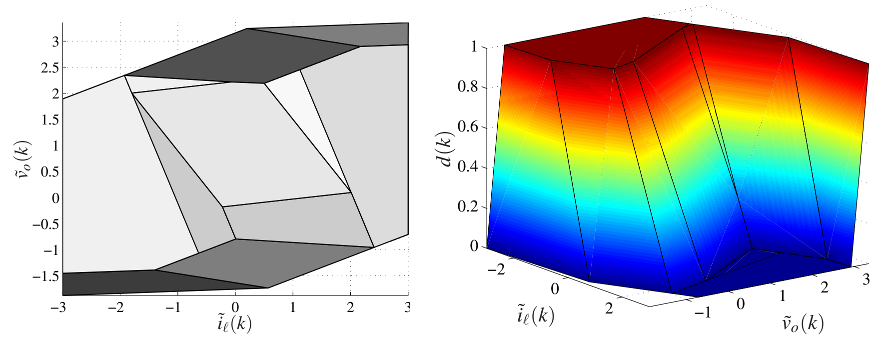
$$\tilde{x}(k+1) = A\tilde{x}(k) + Bd(k) \tag{47}$$

Note that, unlike (44), (47) is linear in the state vector and the duty cycle.

The control objective is to regulate the output voltage to its reference $v_o^\star$ and to maintain the inductor current below its maximal allowed limit $i_{\ell,\max}$ by manipulating the duty cycle. The latter is bounded between zero and one. This control problem can be captured by the optimization problem (cf., (29))

$$\begin{aligned} V(\tilde{x}(k), \mathbf{u}(k)) &= \sum_{\ell=k}^{k+N-1} \left\{ (\tilde{x}'(\ell) - \tilde{x}^\star)^T Q (\tilde{x}'(\ell) - \tilde{x}^\star) + R(u'(\ell))^2 \right\}, \\ \tilde{x}'(\ell+1) &= A\tilde{x}'(\ell) + Bu'(\ell), \\ \tilde{x}'(\ell) &\in \mathbb{X}, \ u'(\ell) \in \mathbb{U}, \quad \ell \in \{0, 1, \ldots N\}, \end{aligned} \tag{48}$$

where we set $Q = \text{diag}(0,1)$, $R = 0.1$, $\mathbb{X} = [-\tilde{i}_{\ell,\max}, \tilde{i}_{\ell,\max}] \times [-10, 10]$ and $\mathbb{U} = [0,1]$. Note that $\tilde{i}_{\ell,\max} = i_{\ell,\max}/v_s$ and $u = d$. To facilitate the regulation of the output



(a) Polyhedral partition of the state-space $\mathbb{X}$ (b) Control law $u^{\mathrm{opt}}(k)$

Fig. 8 Explicit state-feedback control law for the dc-dc buck converter over the state-space $\mathbb{X}$ spanned by the scaled inductor current $\tilde{i}_\ell(k)$ and the scaled output voltage $\tilde{v}_o(k)$

voltage to a non-zero reference, we define $\tilde{x}^\star = [0, \tilde{v}_o^\star]^T$ with $\tilde{v}_o^\star = v_o^\star / v_s$. We assume $\tilde{v}_o^\star = 0.5$ and choose the horizon $N = 3$.

The explicit control law can be computed using the MPT toolbox [71]. The two-dimensional state-space is partitioned into 20 polyhedra. Using optimal complexity reduction [48], an equivalent control law with 11 polyhedra can be derived, as shown in Fig. 8(a). The corresponding state-feedback controller $u^{\mathrm{opt}}(k)$ is shown in Fig. 8(b). Note that the duty cycle is limited by zero and one as a result of the design procedure. An additional patch, such as an anti-windup scheme, is not required.

A similar MPC scheme was proposed in [80]. This rather basic controller can be enhanced in various ways. In the context of dc-dc converters, it is usually preferred to penalize the *change* in the duty cycle rather than the duty cycle as such, by introducing $\Delta u(k) = u(k) - u(k-1)$ and penalizing $R(\Delta u(\ell))^2$ rather than $R(u(\ell))^2$ in (48). To enhance the voltage regulation at steady-state by removing any dc offset, an integrator state can be added [80]. Load variations can be addressed by a Kalman filter, see [43].

4.2.4 Application Examples of the State-Feedback Control Law

In the context of power electronics and drives applications, the notion of the state-feedback control law of MPC formulations has been studied extensively. One of the earliest references is [75], which proposes an explicit MPC controller in a field-oriented controller setting for an electrical drive. These initial results are extended in [81]. In [10], the speed and current control problem of a permanent-magnet synchronous machine is solved using MPC. Drives with flexible shafts are considered in [23] and active rectifier units with LC filters in [82].

4.3 Linear Quadratic MPC with a Finite Control Set

As for the continuous control set case studied above, when the control set is constrained to a finite set, the use of horizons larger than one will often give significant performance gains. This is well known in other application areas, where the use of MPC with finite decision variables has been examined; see, e.g., [57, 93, 100, 102, 105–107]. However, only few results have been documented in the literature regarding power converters, see [22, 45, 46, 101, 118].

We will next examine how to use prediction horizons longer than one for MPC formulations where the plant inputs are constrained to belong to a finite set $\mathbb{U}$. As in (48), we will focus on linear plant models and quadratic cost functions with reference tracking. To address computational issues, we will exploit the geometrical structure of the underlying MPC optimization problem and presents a practical optimization algorithm. The algorithm uses elements of sphere decoding [54] to provide optimal switching sequences, requiring only little computational resources, thus, enabling the use of longer prediction horizons [45, 46]. We will illustrate the ideas on a variable speed drive application consisting of a three-level neutral point clamped voltage source inverter driving an induction machine. Our results show that using prediction horizons larger than one does, in fact, provides significant performance benefits. In particular, at steady-state operation the current distortions and/or the switching frequency can be reduced considerably with respect to direct MPC with horizon one, as presented in Section 3.

The methods proposed and results obtained are directly applicable to both the machine-side inverter in an ac drive setting, as well as to grid-side converters. The ideas can also be used for other converter topology and are particularly promising for topologies with a high number of voltage levels.

4.3.1 Physical Model

As an illustrative example of a medium-voltage power electronic system, we consider a variable speed drive consisting of a three-level neutral point clamped (NPC) voltage source inverter (VSI) driving an induction machine (IM), as depicted in Fig. 9. The total dc-link voltage V_{dc} is assumed constant and the neutral point potential N is fixed.

Let the integer variables u_a, u_b, $u_c \in \{-1,0,1\}$ denote the switch positions in the three phase legs, corresponding to the phase voltages $-\frac{V_{\text{dc}}}{2}, 0, \frac{V_{\text{dc}}}{2}$, respectively. Thus, the voltage applied to the machine terminals in orthogonal coordinates is

$$v_{s,\alpha\beta} = \frac{1}{2} V_{\text{dc}}\, u_{\alpha\beta} = \frac{1}{2} V_{\text{dc}}\, \mathbf{P} u$$

with

$$\mathbf{P} \triangleq \frac{2}{3}\begin{bmatrix} 1 & -\frac{1}{2} & -\frac{1}{2} \\ 0 & \frac{\sqrt{3}}{2} & -\frac{\sqrt{3}}{2} \end{bmatrix}, \quad u \triangleq \begin{bmatrix} u_a \\ u_b \\ u_b \end{bmatrix} \in \mathbb{U}, \quad \mathbb{U} \triangleq \{-1,0,1\}^3. \tag{49}$$

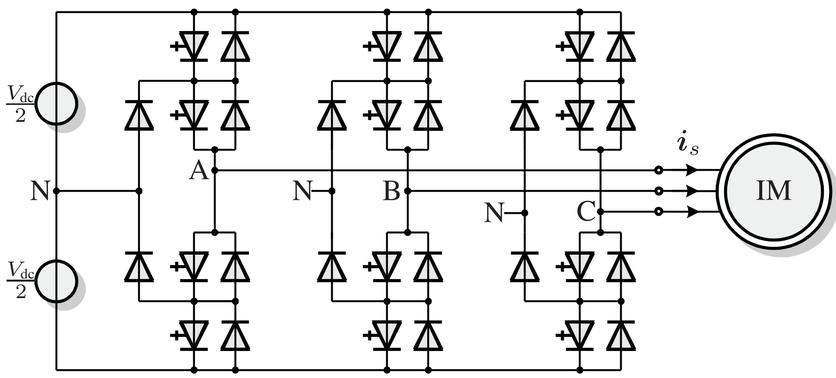


Fig. 9 Three-level three-phase neutral point clamped voltage source inverter driving an induction motor with a fixed neutral point potential

The state-space model of a squirrel-cage induction machine in the stationary $\alpha\beta$ reference frame is summarized hereafter. For the current control problem at hand, it is convenient to choose the stator currents $i_{s\alpha}$ and $i_{s\beta}$ as state variables. The state vector is complemented by the rotor flux linkages $\psi_{r\alpha}$ and $\psi_{r\beta}$, and the rotor's angular velocity ω_r. The model input are the stator voltages $v_{s\alpha}$ and $v_{s\beta}$. The model parameters are the stator and rotor resistances R_s and R_r, the stator, rotor and mutual reactances X_{ls}, X_{lr} and X_m, respectively, the inertia J, and the mechanical load torque T_ℓ. All rotor quantities are referred to the stator circuit. In terms of the above quantities, the continuous-time state equations are [56, 68]

$$\frac{\mathrm{d}i_s}{\mathrm{d}t} = -\frac{1}{\tau_s} i_s + \left(\frac{1}{\tau_r} - \omega_r \begin{bmatrix} 0 & -1 \\ 1 & 0 \end{bmatrix}\right) \frac{X_m}{D} \psi_r + \frac{X_r}{D} v_s \tag{50a}$$

$$\frac{\mathrm{d}\psi_r}{\mathrm{d}t} = \frac{X_m}{\tau_r} i_s - \frac{1}{\tau_r} \psi_r + \omega_r \begin{bmatrix} 0 & -1 \\ 1 & 0 \end{bmatrix} \psi_r \tag{50b}$$

$$\frac{\mathrm{d}\omega_r}{\mathrm{d}t} = \frac{1}{J}(T_e - T_\ell), \tag{50c}$$

where we have used[7]

$$\begin{aligned} X_s &\triangleq X_{ls} + X_m \\ X_r &\triangleq X_{lr} + X_m \\ D &\triangleq X_s X_r - X_m^2. \end{aligned} \tag{51}$$

The transient stator time constant and the rotor time constant are equal to

$$\tau_s \triangleq \frac{X_r D}{R_s X_r^2 + R_r X_m^2} \quad \text{and} \quad \tau_r \triangleq \frac{X_r}{R_r},$$

[7] To simplify the notation, in (50) we dropped $\alpha\beta$ from the vectors i_s, ψ_r and v_s.

whereas the electromagnetic torque is given by

$$T_e = \frac{X_m}{X_r}(\psi_{r\alpha} i_{s\beta} - \psi_{r\beta} i_{s\alpha}) . \tag{52}$$

4.3.2 MPC formulation

The control problem is formulated in the $\alpha\beta$ reference frame. Let

$$i_s^* \triangleq \begin{bmatrix} i_{s\alpha}^* \\ i_{s\beta}^* \end{bmatrix}$$

denote the reference of the instantaneous stator current. The objective of the current controller is to manipulate the three-phase switch position u, by synthesizing a switching sequence, such that the stator current i_s closely tracks its reference. At the same time, the switching effort, i.e., the switching frequency or the switching losses, are to be kept small.

It is convenient to describe the system by introducing the following state vector of the drive model:

$$x \triangleq \begin{bmatrix} i_{s\alpha} \\ i_{s\beta} \\ \psi_{r\alpha} \\ \psi_{r\beta} \end{bmatrix} . \tag{53}$$

The stator current is taken as the system output vector, i.e., $y = i_s$. The switch position $u_{\alpha\beta}$ in the orthogonal coordinate system constitutes the input vector, which is provided by the controller. Discretization of (50) yields a linear system model of the form

$$x(k+1) = Ax(k) + Bu(k).$$

To penalise current errors and the control effort, we adopt the cost function:

$$V(x(k), \mathbf{u}(k)) = \sum_{\ell=k}^{k+N-1} (i'_{e,\alpha\beta}(\ell+1))^T (i'_{e,\alpha\beta}(\ell+1)) + \lambda_u (\Delta u'(\ell))^T \Delta u'(\ell) , \tag{54}$$

where

$$i'_{e,\alpha\beta} \triangleq i^*_{s,\alpha\beta} - i'_{s,\alpha\beta}$$
$$\Delta u'(\ell) \triangleq u'(\ell) - u'(\ell-1)$$

and subject to (see (49))

$$\begin{aligned} \mathbf{u}(k) &\in \mathbb{U}^N \\ \|\Delta u'(\ell)\|_\infty &\leq 1, \quad \forall \ell \in \{k, k+1, \dots, k+N-1\}. \end{aligned} \tag{55}$$

The latter constraint is imposed since in each phase switching is only possible by one step up or down.

4.3.3 Obtaining the Switch Positions via Exhaustive Search

Due to the discrete nature of the decision variable **u**, minimizing (54) subject to (55) is difficult, except for short horizons. In fact, as the prediction horizon is enlarged and the number of decision variables is increased, the (worst-case) computational complexity grows exponentially, thus, cannot be bounded by a polynomial, see also [104]. The difficulties associated with minimizing J become apparent when using exhaustive search. With this method, the set of admissible switching sequences $\mathbf{u}(k)$ is enumerated and the cost function evaluated for each such sequence. The switching sequence with the smallest cost is (by definition) the optimal one and its first element is chosen as the control input. At every time-step k, exhaustive search entails the following procedure:

1. Given the previously applied switch position $u(k-1)$ and taking into account (55), determine the set of admissible switching sequences over the horizon.
2. For each of these switching sequences, compute the cost V according to (54).
3. Choose the switching sequence, $\mathbf{u}^{\text{opt}}(k)$, which minimizes the cost. Apply its first element, $u^{\text{opt}}(k)$, to the converter.

It is easy to see that exhaustive search is computationally feasible only for very small horizons N, such as one or two. For $N = 5$, assuming a three-level converter, the number of switching sequences amounts to $1.4 \cdot 10^7$.

Techniques from vector quantization [31] and from mathematical programming, such as branch and bound [36, 72, 85], can be used to reduce the computational burden. In particular, following akin to the method described in Section 3.2, an explicit state-feedback control law for FCS-MPC can be obtained which, as for the convex case in Section 4.2, induces a partition of the state-space [104]. In addition, off-the-shelf solvers such as CPLEX [58], include a wealth of smart heuristics and methods. However, none of the general methods take advantage of the particular structure of (54) and the fact that in MPC the solution is implemented in a moving horizon manner.

4.4 An Efficient Algorithm for Finite-Control Set MPC

We will next present a method for calculating the optimal switching sequences in FCS-MPC. The algorithm requires only little computations and is thereby attractive for applications in power electronics and drives.

4.4.1 Modified Sphere Decoding Algorithm

Direct algebraic manipulations akin to those mentioned in Sections 4.1 and 4.2 (see also [50, 104]), give that minimization of (54) amounts to finding

$$\mathbf{u}^{\text{opt}}(k) = \arg\min_{\mathbf{u}} (\mathbf{z} - \mathbf{H}\mathbf{u})^T (\mathbf{z} - \mathbf{H}\mathbf{u}), \quad \text{subject to (55)}, \tag{56}$$

where $\mathbf{H}$ is an invertible lower-triangular matrix. In (56),

$$\mathbf{z} = \mathbf{H}\mathbf{u}^{\text{uc}},$$

where $\mathbf{u}^{\text{uc}}$ is the sequence obtained from optimizing (54) *without constraints*, i.e., with $\mathbb{U} = \mathbb{R}^3$. Thus, we have rewritten the MPC optimization problem as a (truncated) *integer least-squares* problem. Interestingly, various efficient solution algorithms for (56) subject to finite-set constraints have been developed in recent years; see, e.g., [1, 54] and references therein. We will next show how to adapt the sphere decoding algorithm [27, 54] to find the optimal switching sequence $\mathbf{u}^{\text{opt}}(k)$.

The basic idea of the algorithm is to iteratively consider candidate sequences, say $\mathbf{u} \in \mathbb{U}^N$, which belong to a sphere of radius $\rho(k) > 0$ centered in $\mathbf{z}$,

$$(\mathbf{z} - \mathbf{H}\mathbf{u})^T (\mathbf{z} - \mathbf{H}\mathbf{u}) \leq \rho(k). \tag{57}$$

Especially in the case of multi-level converters (where $\mathbb{U}$ has many elements; see, e.g., [74]), the set of candidate sequences satisfying the above conditions is much smaller than the original constraint set $\mathbb{U}^N$. Not surprisingly, computation times can be drastically reduced compared to exhaustive search.

A key property used in sphere decoding is that, since $\mathbf{H}$ is triangular, for a given radius, identifying candidate sequences which satisfy (57) is very simple. In particular, for the present case, $\mathbf{H}$ is lower triangular, thus (57) can be rewritten as

$$\rho^2(k) \geq (z_1 - H_{(1,1)}u_1)^2 + (z_2 - H_{(2,1)}u_1 - H_{(2,2)}u_2)^2 + \dots \tag{58}$$

where z_i denotes the i-th element of $\mathbf{z}$, u_i is the i-th element of $\mathbf{u}$, and $H_{(i,j)}$ refers to the (i, j)-th entry of $\mathbf{H}$. Therefore, the solution set of (57) can be found by proceeding in a sequential manner akin to Gaussian elimination, in the sense that at each step only a one-dimension problem needs to be solved; for details, see [54].

The algorithm requires an initial value for the radius used at time k to determine $\mathbf{u}$. On the one hand, the radius $\rho(k)$ should be as small as possible, enabling us to remove as many candidate solutions *a priori* as possible. On the other hand, $\rho(k)$ must not be too small, to ensure that the solution set is non-empty. We propose to choose the initial radius by using the following *educated guess* for the optimal solution:

$$\mathbf{u}^{\text{sub}}(k) = \begin{bmatrix} 0 & I & 0 & \dots & 0 \\ 0 & 0 & I & \ddots & \vdots \\ \vdots & & \ddots & \ddots & 0 \\ 0 & \dots & \dots & 0 & I \\ 0 & \dots & \dots & 0 & I \end{bmatrix} \mathbf{u}^{\text{opt}}(k-1), \tag{59}$$

which is obtained by shifting the previous solution by one time-step and repeating the last switch position. This is in accordance with the moving horizon optimization paradigm. Since the optimal solution at the previous time-step satisfies the constraint, $\mathbf{u}^{\text{sub}}(k)$ is a feasible solution candidate of (54). Given (59), the initial value of $\rho(k)$ is then set to:

$$\rho(k) = (\mathbf{z} - \mathbf{H}\,\mathbf{u}^{\text{sub}}(k))^T(\mathbf{z} - \mathbf{H}\,\mathbf{u}^{\text{sub}}(k)). \tag{60}$$

At each time-step k, the controller first uses the current system state $\mathbf{x}(k)$, the future reference values, the previous switch position $\mathbf{u}(k-1)$ and the previous optimizer $\mathbf{u}^{\text{opt}}(k-1)$ to calculate $\mathbf{u}^{\text{sub}}(k)$, $\rho(k)$ and $\mathbf{z}$. The optimal switching sequence $\mathbf{u}^{\text{opt}}(k)$ is then obtained by invoking Algorithm 1:

$$\mathbf{u}^{\text{opt}}(k) = \text{MSPHDEC}(\emptyset, 0, 1, \rho^2(k), \mathbf{z}), \tag{61}$$

where $\emptyset$ is the empty set[8].

Algorithm 1 Modified sphere decoding algorithm

```
function u^OPT(k) = MSPHDEC(u, d^2, i, ρ^2, z)
    for each u ∈ {−1, 0, 1} do
        u_i ← u
        d'^2 ← (z_i − H_(i,1:i) u_1:i)^T (z_i − H_(i,1:i) u_1:i) + d^2
        if d'^2 ≤ ρ^2 then
            if i < 3N then
                MSPHDEC(u, d'^2, i+1, ρ^2, z)
            else
                if u meets (55) then
                    u^opt ← u
                    ρ^2 ← d'^2
                end if
            end if
        end if
    end for
end function
```

[8] The notation $\mathbf{H}_{(i,1:i)}$ refers to the first i entries of the i-th row of $\mathbf{H}$; similarly, $\mathbf{u}_{1:i}$ are the first i elements of the vector $\mathbf{u}$. Note that the matrix $\mathbf{H}$ is time-invariant and does not change when running the algorithm. Therefore, $\mathbf{H}$ can be computed once offline before the execution of the algorithm.

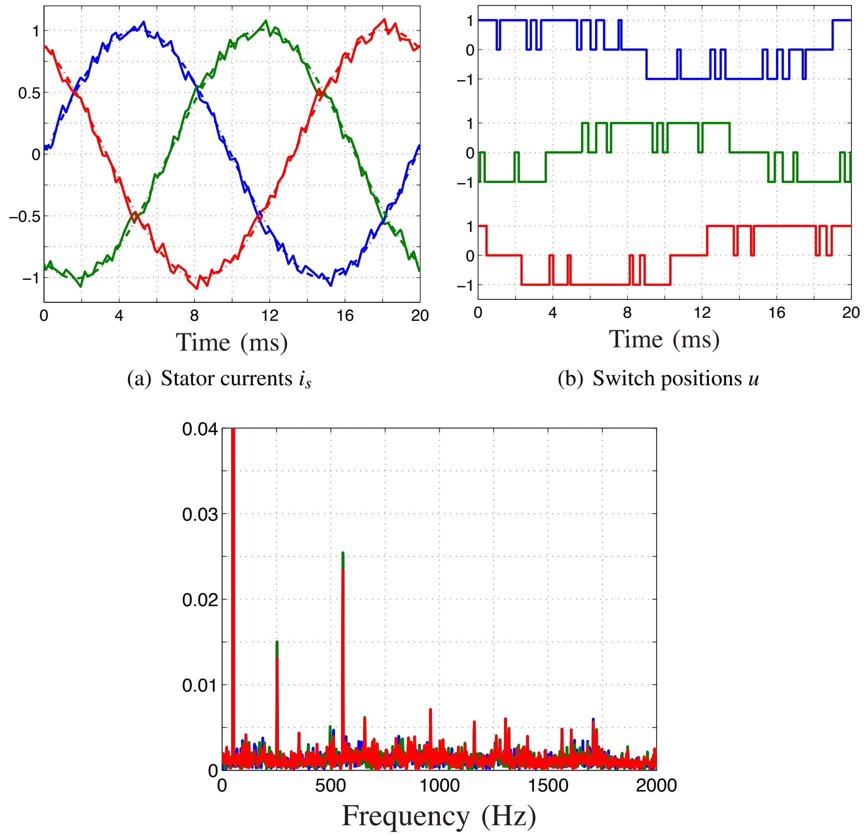
As can be seen in Algorithm 1, the proposed modification to sphere decoding operates in a recursive manner. Starting with the first component, the switching sequence $\mathbf{u}$ is built component by component, by considering the admissible single-phase switch positions in the constraint set $\{-1,0,1\}$. If the associated squared distance is smaller than the current value of ρ^2, then we proceed to the next component. If the last component,i.e., u_{3N}, has been reached, meaning that $\mathbf{u}$ is of full dimension $3N$, then $\mathbf{u}$ is a candidate solution. If $\mathbf{u}$ meets the switching constraint (55) and if the distance is smaller than the current optimum, then we update the incumbent optimal solution $\mathbf{u}^{\text{opt}}$ and also the radius ρ.

The computational advantages of this algorithm stem from adopting the notion of branch and bound [72, 85]. Branching is done over the set of single-phase switch positions $\{-1,0,1\}$; bounding is achieved by considering solutions only within the sphere of current radius. If the distance d' exceeds the radius, a certificate has been found that the branch (and all its associated switching sequences) provides only solutions worse than the incumbent optimum. Therefore, this branch can be pruned, i.e., removed from further consideration without exploring it. During the optimization procedure, whenever a better incumbent solution is found, the radius is reduced and the sphere thus tightened, so that the set of candidate sequences is as small as possible, but non-empty. The majority of the computational burden relates to the computation of d' via evaluating the terms $\mathbf{H}_{(i,1:i)}\mathbf{u}_{1:i}$. Thanks to (58), d' can be computed sequentially, by computing only the squared addition due to the ith component of $\mathbf{u}$. In particular, the sum of squares in d, accumulated over the layers 1 to $i-1$, does not need to be recomputed.

4.4.2 Performance Evaluation

Whilst the main advantages of using MPC, when compared to other methods such as optimized pulse patterns (OPPs), see e.g. [14], lie in the handling of transients, we will focus on the steady-state behaviour. We consider a three-level voltage source inverter driving an induction machine with a constant mechanical load. A 3.3 kV and 50 Hz squirrel-cage induction machine rated at 2 MVA with a total leakage inductance of 0.25 pu is used as an example of a typical medium-voltage induction machine. The dc-link voltage is $V_{\text{dc}} = 5.2\,\text{kV}$ and assumed to be constant.

The key control performance criteria are the device switching frequency f_{sw} and the current THD I_{THD}. In addition, we will also investigate the empirical *closed-loop* cost, V_{cl}, which in accordance with (54) captures the squared RMS current error plus the weighted averaged and squared switching effort. In a first step, the steady-state performance of MPC tracking the current reference is illustrated, using the sampling interval $h = 25\,\mu\text{s}$. The controller uses the cost function J with prediction horizon $N = 10$ and weighting factor $\lambda_u = 103 \cdot 10^{-3}$. This results in an average device switching frequency of $f_{\text{sw}} = 300\,\text{Hz}$, which is typical for medium-voltage applications, and a current THD of $I_{\text{THD}} = 5.03\%$. Fig. 10(a) illustrates three-phase stator current waveforms along with their (dash-dotted) references over one fundamental period. The colours blue, green and red correspond to phase a, b and c,



(a) Stator currents i_s

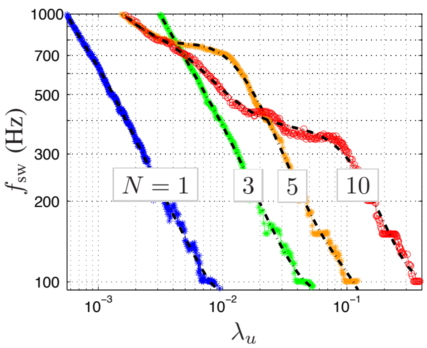
(b) Switch positions u

(c) Stator current spectrum

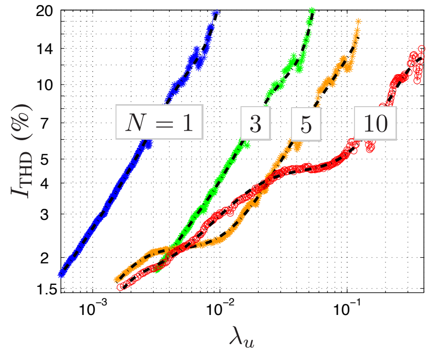
Fig. 10 Simulated waveforms for MPC with horizon $N = 10$ and the weight $\lambda_u = 103 \cdot 10^{-3}$

respectively. The spectrum of the stator current, computed with a Fourier transformation, is shown in Fig. 10(c). The three-phase switching sequence is depicted in Fig. 10(b). As can be seen, unlike PWM, the switching pattern lacks symmetry and repetitiveness, resulting in a non-discrete and predominantly flat spectrum. Nevertheless, non-tripled odd-order current harmonics such as the 5th, 7th, 11th, 13th and 19th harmonics are clearly identifiable.

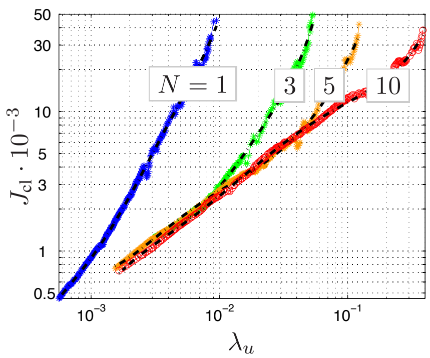
Next, the influence of λ_u on the switching frequency, the current THD and the cost is investigated. For each of the horizons $N = 1$, 3, 5 and 10 and for more than 1000 different values of λ_u, ranging between 0 and 0.5, steady-state simulations were run. Focusing on switching frequencies between 100 Hz and 1 kHz, and current THDs below 20%, the results are shown in Fig. 11, using a double logarithmic scale. Each simulation corresponds to a data point. Polynomial functions are overlaid, which approximate the individual data points. Figs. 11(a) and 11(b) suggest



(a) Average switching frequency f_{sw}



(b) Current THD I_{THD}



(c) Closed-loop cost

Fig. 11 Key performance criteria of MPC for the prediction horizons $N = 1, 3, 5, 10$. The switching frequency, current THD and closed-loop cost are shown as a function of the tuning parameter λ_u, using a double logarithmic scaling. The individual simulations are indicated using dots, their overall trend is approximated using dash-dotted polynomials

that, for small prediction horizons, the relationship between λ_u and the performance variables is approximately linear in double logarithmic scale; for larger values of N, the relationship is more complicated, but still monotonic. Fig. 11(c) illustrates the empirical closed-loop costs obtained. Clearly, the cost is reduced as the prediction horizon is increased, suggesting the use of horizons larger than one. For example, with $\lambda_u = 0.01$ and $N = 1$, we have $J_{\mathrm{cl}} \approx 50$, whereas with horizon $N = 3$, the closed-loop cost can be reduced to $J_{\mathrm{cl}} \approx 3$! We note that, for this value of λ_u, the empirical closed-loop cost achieved is almost optimal.

Prediction horizon N	Sphere decoding avg.	Sphere decoding max.	Exhaustive search avg.	Exhaustive search max.
1	1.18	5	11.8	18
2	1.39	8	171	343
3	1.72	14	2350	4910
5	2.54	35	467’000	970’000
10	8.10	220		

Table 1 Average and maximal number of switching sequences that need to be considered by the sphere decoding and exhaustive search algorithms to obtain the optimal result, depending on the length of the prediction horizon

4.4.3 Computational Burden

We next compare the computational burden of the algorithm presented in Section 4.4 with that of exhaustive search, see Section 4.3.3. The weight λ_u is tuned such that approximately the same switching frequency of $f_{\text{sw}} = 300\,\text{Hz}$ is obtained, irrespective of the chosen prediction horizon. As a measure of the computational burden, the number of switching sequences, which are investigated by the algorithm at each time-step when computing the optimum, is considered. Over multiple fundamental periods, the average as well as the maximal number of sequences is monitored, as summarized in Table 1. The table shows that, as the prediction horizon is increased, initially, the computational burden associated with Algorithm 1 grows slowly, despite being exponential, whilst exhaustive search becomes computationally intractable already for horizons of five and longer.

4.5 MPC for Switched Systems

Power electronic systems are hybrid systems, featuring different dynamics for different sets of binary switch positions. When considering currents, fluxes and voltages, power electronic systems constitute switched linear systems. However, when an electromagnetic torque, flux magnitude, or real and reactive power expression is used, the system turns into a switched nonlinear system. The switched linear behavior of power electronic systems can be directly captured by polyhedral piecewise affine (PWA) systems as described below. PWA systems also allow the approximation of switched nonlinear behavior with an arbitrary accuracy. Alternatively, solution approaches are available that directly address the switched nonlinear optimization problem, albeit in an approximate manner, as shown at the end of this section.

4.5.1 Piecewise Affine Systems

Polyhedral PWA systems [55, 117] are defined by partitioning the state-input space into polyhedra and associating with each polyhedron an affine state-update and output function

$$x(k+1) = A_{i(k)}x(k) + B_{i(k)}u(k) + f_{i(k)} \tag{62a}$$

$$y(k) = C_{i(k)}x(k) + D_{i(k)}u(k) + g_{i(k)} \tag{62b}$$

$$\text{with } i(k) \text{ such that } [x^T(k)\ u^T(k)]^T \in \mathscr{P}_{i(k)}, \tag{62c}$$

where $x \in \mathbb{X}$, $u \in \mathbb{U}$, $y \in \mathbb{Y}$ denote the state, input and output vectors, respectively. The state vector $x = [x_r^T\ x_b^T]^T$ encompasses, in general, real-valued components $x_r \in \mathbb{X}_r \subseteq \mathbb{R}^{n_r}$ as well as binary components $x_b \in \mathbb{X}_b \subseteq \{0,1\}^{n_b}$. The same applies to the inputs and outputs.

The polyhedra $\mathscr{P}_{i(k)}$ define a set of polyhedra $\{\mathscr{P}_i\}_{i\in I}$ on $\mathbb{X} \times \mathbb{U}$, and the real matrices $A_{i(k)}, B_{i(k)}, C_{i(k)}, D_{i(k)}$ and real vectors $f_{i(k)}, g_{i(k)}$ with $i(k) \in I$, I finite, are constant and have suitable dimensions. We refer to $i(k)$ as the *mode* of the system, which is associated with a binary state and a binary input.

PWA systems are so called *linear hybrid* systems. These are heterogenous systems that incorporate both continuous-valued components governed by difference equations, as well as discrete-valued components, such as finite state machines, if-then-else rules and on/off switches. PWA systems switch between different operating modes, with each mode being governed by a discrete-time affine dynamical law. Mode transitions are triggered by inputs or states crossing specific affine thresholds. PWA constraints can be imposed on states and inputs.

Modelling complex hybrid systems in PWA form is, in general, a tedious and highly non-trivial task. To facilitate the modelling process, the HYbrid Systems DEscription Language (HYSDEL) has been developed, which allows the designer to describe a hybrid system on a textual basis [123]. The HYSDEL modelling language is an integral part of the MPT toolbox. Tools, such as the mode enumeration algorithm, are available to translate HYSDEL code into PWA form [49].

Within the class of linear hybrid systems, a number of modelling frameworks are available, which are equivalent to each other [55]. Apart from PWA systems, another major representative is the *mixed logical dynamical* (MLD) framework [8]. The MLD framework extends linear discrete-time systems by augmenting the state and output equations by auxiliary real and binary variables and a mixed-integer linear inequality constraint. MLD models are very suitable for online MPC, since the equality and inequality constraints of the MLD model can be easily included in the optimization problem. Depending on whether the cost function is linear or quadratic, the optimization problem is either a *mixed-integer* linear or quadratic program (MILP or MIQP).

The State-Feedback Control Law

PWA models are the starting point to derive off-line the state-feedback control law [11]. Indeed, the notion of multi-parametric programming can be extended to PWA systems. Specifically, the formulation of a linear cost function subject to a PWA model gives rise to a *multi-parametric mixed-integer linear program* (mp-MILP). As proposed in [11,12], the state-feedback control law can be computed, by augmenting mp-LP with dynamic programming, i.e. by moving backwards in time using mp-LP.

Theorem 4. *The solution to mp-MILP is a state-feedback control law $u^{opt}(k)$ that is a piecewise affine function of the state vector $x(k)$ defined on a polyhedral partition of the feasible state-space $\mathbb{X}$.* ∎

Theorem 5. *The value function $V^{opt}(x(k)) = V(x(k), \mathbf{u}^{opt}(k))$ of the mp-MILP is convex and piecewise affine in the state.* ∎

The related proofs and additional details can be found in [26] for mp-MILPs, and in [11,25] for mp-MIQPs. Furthermore, [12] provides an in-depth analysis and description of multi-parametric programming for MILPs and MIQPs.

Application Examples

Consider the direct torque control (DTC) problem of ac machines, first addressed in [120]. In DTC, the electromagnetic torque T and the stator flux magnitude Ψ are directly controlled without using a modulator. A suitable voltage vector is selected that keeps the torque and flux magnitude within upper and lower bounds, which are imposed around their references. For a neutral point (NP) clamped three-level inverter, also the NP potential is to be balanced around zero. By approximating the nonlinearities relating to the torque, flux magnitude, machine rotation and NP potential by PWA functions, the DTC problem can be cast in the MLD framework using HYSDEL, as shown in [94], and then translated into PWA form.

Let $\Psi_{\max}$ and $\Psi_{\min}$ denote the upper and lower flux magnitude bounds, respectively, and Ψ^* its reference. We introduce the non-negative term

$$\varepsilon_\Psi(\ell) = \begin{cases} q_F(\Psi(\ell) - \Psi_{\max}) & \text{if } \Psi(\ell) \geq \Psi_{\max} \\ q_F(\Psi_{\min} - \Psi(\ell)) & \text{if } \Psi(\ell) \leq \Psi_{\min} \\ q_f|\Psi(\ell) - \Psi^*(\ell)| & \text{else} \end{cases} \tag{63}$$

that uses soft constraints with the weight q_F to heavily penalize violations of the bounds. A small penalty q_f, with $q_f \ll q_F$, is added to penalize deviations from the reference. Similarly, for the torque and the NP potential, the terms ε_T and ε_υ can be defined. The switching transitions are penalized by

$$\varepsilon_u(\ell) = q_u(\ell)\|u(\ell) - u(\ell-1)\|_1, \tag{64}$$

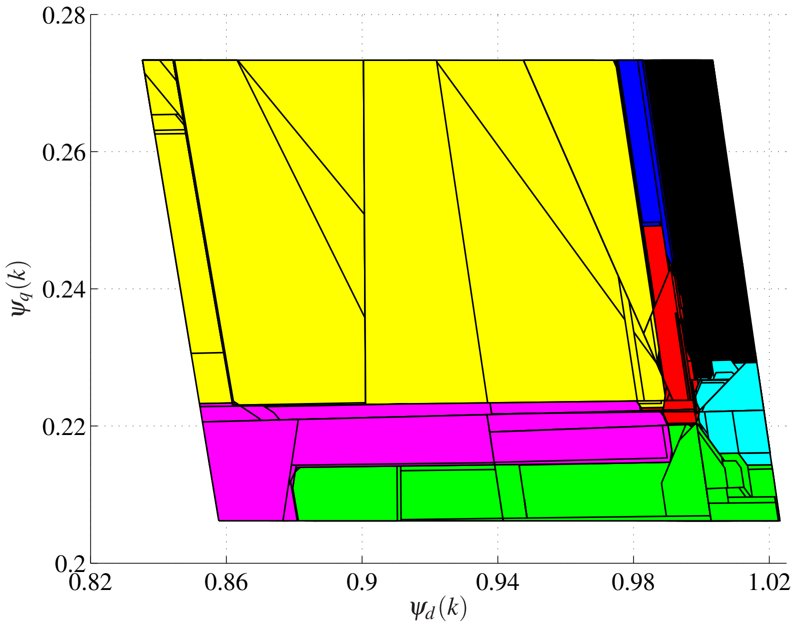


Fig. 12 The explicit state-feedback control law for the DTC drive with a two-level inverter for $u(k-1) = [+1\ \ -1\ \ -1]^T$ and a given angular position of the dq reference frame, where the colours correspond to the unique switch positions

where $q_f < q_u(\ell) < q_F$ is a time-varying weight that decays exponentially within the prediction horizon, providing an incentive to further reduce the switching frequency by postponing switching until at least one soft constraint is about to be violated.

Aggregating these terms results in the piecewise linear cost function

$$V(x(k), \mathbf{u}(k), u(k-1)) = \sum_{\ell=k}^{k+N-1} \| \left[\varepsilon'_T(\ell+1)\ \ \varepsilon'_\Psi(\ell+1)\ \ \varepsilon'_\upsilon(\ell+1)\ \ \varepsilon'_u(\ell) \right]^T \|_1 \,. \tag{65}$$

The state vector includes the stator flux components in a rotating dq reference frame, the angular position of the reference frame and the NP potential. Minimizing (65) subject to the integer constraints on the switch positions and the PWA model results in an MILP, which can be solved off-line by computing the state-feedback control [95]. A move blocking strategy [16] reduces the complexity of the solution, albeit it remains high. Considering a two-level inverter for the ease of visualization, Fig. 12 depicts the state-feedback control law in the stator flux plane for a specific angular position and for $u(k-1) = [+1\ \ -1\ \ -1]^T$. The colours refer to distinct switch positions $u(k)$, as explained in detail in [95]. In particular, the large (yellow) region refers to $u(k) = u(k-1)$. In this region, to maintain the controlled variables within their bounds, switching is not required and thus avoided.

Recognizing that the DTC problem of keeping the torque and stator flux magnitude within given bounds strongly relates to feasibility, by computing a semi-explicit control law a controller of lower complexity can be derived [40].

Similarly, also dc-dc converters can be modelled as PWA systems, and an MPC problem with a linear cost function can be formulated for the buck converter [42]. A Kalman filter can be used to account for load changes, and closed-loop stability can be proven by deriving a piecewise quadratic Lyapunov function [43]. The state-feedback control law can be easily computed and implemented on a DSP to obtain experimental results [41]. Similar results are available for boost converters [6, 79].

4.5.2 Switched Nonlinear Systems

Solving MPC problems involving switched *nonlinear* systems in real-time is a highly challenging task, since this amounts to solving a *mixed-integer nonlinear program*. Computing an explicit solution for such problems remains largely an open problem.

Optimization Problem

Nevertheless, for a subclass of MPC problems with switched nonlinear systems, an optimization algorithm can be constructed, which features a computational complexity that is suitable for implementation, albeit it solves the MPC problem only in an approximative manner. More specifically, consider a nonlinear system with integer inputs, whose output variables are to be regulated along given trajectories, by keeping the outputs within upper and lower bounds around their references. The second control objective is to minimize the switching effort, i.e. the switching frequency or the switching losses.

The (short-term) switching frequency is captured by the objective function

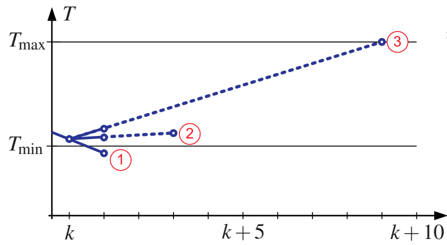
$$V(x(k), \mathbf{u}(k), u(k-1)) = \frac{1}{N} \sum_{\ell=k}^{k+N-1} ||\Delta u'(\ell)||_1 , \tag{66}$$

which represents the sum of the switching transitions (number of commutations) over the prediction horizon divided by the length of the horizon. Alternatively, the switching (power) *losses* can be directly represented through

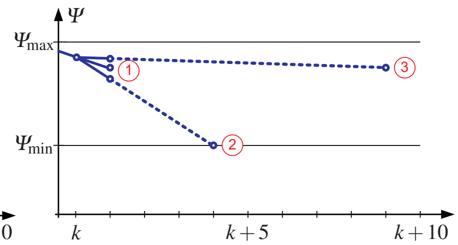
$$V(x(k), \mathbf{u}(k), u(k-1)) = \frac{1}{N} \sum_{\ell=k}^{k+N-1} E_{\mathrm{sw}}(x'(\ell), u'(\ell), u'(\ell-1)) , \tag{67}$$

which is the sum of the instantaneous switching (energy) losses E_{sw} over the prediction horizon[9].

[9] Note that, E_{sw} is a function of the inverter current i, which in turn is either a state variable or a linear combination of the state vector x.



(a) Electromagnetic torque



(b) Stator flux magnitude

Fig. 13 Linear extrapolation of the torque and stator flux trajectories using the samples at time-steps k and $k+1$. For each of the three switch positions $u(k)$ the trajectories are extrapolated until one of them hits a bound

For a drive system as in the previous section, for example, the optimization problem can be stated as

$$\mathbf{u}^{\text{opt}}(k) = \arg\min_{\mathbf{u}(k)} V(x(k), \mathbf{u}(k), u(k-1)) \tag{68a}$$

$$\text{s.t. } x'(\ell+1) = Ax'(\ell) + Bu'(\ell) \tag{68b}$$

$$y'(\ell+1) = g(x'(\ell+1)) \tag{68c}$$

$$y'(\ell+1) \in \mathscr{Y} \text{ or } \varepsilon_y'(\ell+1) < \varepsilon_y'(\ell) \tag{68d}$$

$$u'(\ell) \in \mathbb{U}, \; ||\Delta u'(\ell)||_\infty \leq 1 \tag{68e}$$

$$\forall \ell = k, \ldots, k+N-1\,, \tag{68f}$$

When using inductor currents, fluxes or capacitor voltages as states, the state-update equation is typically linear in power electronic systems, while the outputs y, such as the torque, stator flux magnitude and the NP potential, are a nonlinear function $g(\cdot)$ of the state vector. The outputs are to be maintained within their bounds, described by the set $\mathscr{Y}$ or, if a bound is violated, brought closer to this set at every time-step within the horizon.

An Algorithm based on Enumeration and Extrapolation

Attempting to solve the optimization problem (68) for a long horizon, say 80, leads for a three-level inverter to as many possible switching sequences as there are atoms in the observable universe, which is clearly a futile endeavour. By considering switching transitions only when one of the output variables is close to one of its bounds, the number of switching sequences to be considered can be greatly reduced. Depending on the horizon length, a few tens to a few thousand sequences are to be investigated. Since the switching effort is to be minimized, this heuristic turns out to have only a minor effect on the solution. Indeed, as shown in [35], optimality is only mildly affected, which is evidenced by the fact that the closed-loop performance in terms of the current THD and switching losses is, at least for very long prediction horizons, effectively equal to the one obtained by offline computed optimized pulse patterns [14]. The latter are widely considered to provide steady-state operating conditions that upper bound achievable performance.

Between switching instants, the concept of extrapolation is used, which is similar to an adaptive move blocking scheme. The notion of *linear* extrapolation is highlighted in Fig. 13, where two output variables are considered along with their—in this case—constant bounds. At time-step k, the model (68b)–(68c) is used to compute $y(k+1)$ for three different switching inputs $u(k)$. For each $u(k)$, based on $y(k)$ and $y(k+1)$, extrapolation is used to compute the number of steps a switch position can be applied to the inverter before a bound is violated. This operation is computationally very cheap.

Using enumeration of all possible switching transitions, an algorithm using extrapolation can be easily constructed, which relies on three key concepts:

1. The optimization problem is formulated in an open, rather than in a closed, form. The set of admissible switching sequences is constructed sequentially, and the corresponding output trajectories are computed forward in time.
2. In between of the switching events, the output trajectories are either computed using a model of the drive system or by extrapolating them. Typically, quadratic extrapolation is used, even though linear extrapolation is often sufficiently accurate.
3. The set of admissible switching sequences is controlled by the so-called switching horizon, which is composed of the elements ”S” and ”E” that stand for switch and extrapolate (or more generally extend), respectively.

It is important to distinguish between the switching horizon (number of switching instants within the horizon, i.e., the degrees of freedom) and the prediction horizon (number of time steps MPC looks into the future). Between the switching instants, the switch positions are frozen and the drive behavior is extrapolated until a bound is hit. The concept of extrapolation gives rise to long prediction horizons (typically, 30–200 time steps), while the switching horizon is short (usually one to three). Note that the prediction horizon directly relates to the steady-state performance, which, in this case, is the ratio between the switching effort and the current THD. The switching horizon, on the other hand, is proportional to the computational burden.

For an in-depth description and analysis of this algorithm, the reader is referred to [44] and [33]. It’s roots can be traced back to the 1980s [115]. Branch and bound techniques can be used to reduce the computation time by an order of magnitude [36]. Smart extrapolation methods can be used to increase the accuracy of the predictions [126]. Infeasible states, so called deadlocks, can be largely avoided, by adding terminal weights and terminal constraints [15]. A deadlock resolution strategy was proposed in [96]. Closed-loop and robust stability can be shown [38].

Application Examples

The algorithm described above was first proposed as model predictive direct torque control (MPDTC) to address the DTC problem of medium-voltage induction motor drives. During the past few years, MPDTC has been extended into the *model predictive direct control* family (MPDxC) with $x \in \{T, C, P, B\}$, representing the torque, current, power and balancing control problem, respectively.

More specifically, model predictive direct torque (MPDTC) was proposed in 2005 [32, 44] as a successor of DTC, tested on a 2.5 MVA drive in 2007 [96] and generalized in 2009 to arbitrary switching horizons [33]. At steady-state operating conditions, MPDTC provides switching losses and current distortions similar to the ones typically achieved by optimized pulse patterns [14], while during transients, its dynamic response is as fast as the one of DTC [35]. Model predictive direct current control (MPDCC) is a derivative of MPDTC [37], while model predictive direct power control (MPDPC) is the adaptation of MPDTC to grid-connected converters [47, 115]. Model predictive direct balancing control (MPDBC) is the most recent member of the family, being used to control the internal voltages of multilevel converters [65].

A control approach similar to MPDTC can be also applied to dc-dc boost converters. As shown in [61], the voltage control problem can be tackled with one control loop, by adopting the concepts of enumeration, move blocking as well as penalties on the tracking error and switching effort.

5 Conclusions

In this chapter, basic aspects and methods underlying model predictive control have been discussed. To clarify the concepts, horizon-one controllers have been analyzed in detail, including a derivation of the optimal solution and establishing relationships to dead-beat controllers. For MPC formulations with longer horizons, we have presented optimization algorithms, which allow one to implement long-horizon MPC in practical applications in power electronics and drives.

The presentation in this chapter has been kept at a basic system-theoretic level and illustrated on simple converter topologies. Some configurations like, e.g., Active-Front-End converters [99], require a more careful consideration of both control theoretic tools and also physical system knowledge for the design of high-performance model predictive controllers. Subsequent chapters in this book will serve to illustrate the synergy required.

Acknowledgements Research supported under Australian Research Council's Discovery Projects funding scheme (project number DP 110103074).

References

1. Agrell, E., Eriksson, T., Vardy, A., Zeger, K.: Closest point search in lattices. IEEE Trans. Inform. Theory **48**(8), 2201–2214 (2002)
2. Aguilera, R.P., Quevedo, D.E.: Capacitor voltage estimation for predictive control algorithm of flying capacitor converters. In: IEEE International Conference on Industrial Technology. Melbourne, Australia (2009)
3. Aguilera, R.P., Quevedo, D.E.: On stability and performance of finite control set MPC for power converters. In: Workshop on Predictive Control of Electrical Drives and Power Electronics (PRECEDE), pp. 55–62 (2011)
4. Aguilera, R.P., Quevedo, D.E.: Stability Analysis of Quadratic MPC with a Discrete Input Alphabet. IEEE Transactions on Automatic Control (2013)
5. Åström, K.J., Wittenmark, B.: Computer-Controlled Systems: Theory and Design, 3rd edn. Prentice Hall (1997)
6. Beccuti, A.G., Papafotiou, G., Morari, M.: Explicit model predictive control of the boost dc-dc converter. In: Proceedings of the 2nd IFAC Conf. on Analysis and Design of Hybrid Systems. Alghero, Italy (2006)
7. Bemporad, A., Borrelli, F., Morari, M.: Model predictive control based on linear programming — the explicit solution. IEEE Trans. Automat. Contr. **47**(12), 1974–1985 (2002)
8. Bemporad, A., Morari, M.: Control of systems integrating logic, dynamics and constraints. Automatica **35**(3), 407–427 (1999)
9. Bemporad, A., Morari, M., Dua, V., Pistikopoulos, E.N.: The explicit linear quadratic regulator for constrained systems. Automatica **38**(1), 3–20 (2002)
10. Bolognani, S., Bolognani, S., Peretti, L., Zigliotto, M.: Design and implementation of model predictive control for electrical motor drives. IEEE Trans. Ind. Electron. **56**(6), 1925–1936 (2009)
11. Borrelli, F.: Constrained Optimal Control of Linear and Hybrid Systems, *LNCIS*, vol. 290. Springer (2003)
12. Borrelli, F., Baotić, M., Bemporad, A., Morari, M.: Dynamic programming for constrained optimal control of discrete-time linear hybrid systems. Automatica **41**(10), 1709–1721 (2005)
13. Borrelli, F., Bemporad, A., Morari, M.: Geometric algorithm for multiparametric linear programming. Journal of Optimization Theory and Applications **118**(3), 515–540 (2003)
14. Buja, G.S.: Optimum output waveforms in PWM inverters. IEEE Trans. Ind. Appl. **16**(6), 830–836 (1980)
15. Burtscher, T., Geyer, T.: Deadlock avoidance in model predictive direct torque control. In: Proc. IEEE Energy Convers. Congr. Expo. Raleigh, NC, USA (2012)
16. Cagienard, R., Grieder, P., Kerrigan, E., Morari, M.: Move blocking strategies in receding horizon control. J. of Process Control **17**, 563–570 (2007)
17. Camacho, E.F., Bordons, C.: Model Predictive Control. Springer-Verlag, New York, N.Y. (1999)
18. Clarke, D.W., Mohtadi, C., Tuffs, P.S.: Generalized predictive control—part I. The basic algorithm. Automatica **23**(2), 137–148 (1987)
19. Correa, P., Pacas, M., Rodríguez, J.: Predictive Torque Control for Inverter-Fed Induction Machines. IEEE Transactions on Industrial Electronics **54**(2), 1073–1079 (2007)
20. Cortés, P., Kazmierkowski, M.P., Kennel, R.M., Quevedo, D.E., Rodríguez, J.: Predictive Control in Power Electronics and Drives. IEEE Transactions on Industrial Electronics **55**(12), 4312–4324 (2008)
21. Cortés, P., Ortiz, G., Yuz, J.I., Rodríguez, J., Vazquez, S., Franquelo, L.: Model Predictive Control of an Inverter With Output Filter for UPS Applications. IEEE Transactions on Industrial Electronics **56**(6), 1875–1883 (2009)
22. Cortés, P., Rodríguez, J., Vazquez, S., Franquelo, L.: Predictive control of a three-phase UPS inverter using two steps prediction horizon. In: Proc. IEEE Int. Conf. Ind. Technol., pp. 1283–1288. Viña del Mar, Chile (2010)

23. Cychowski, M., Szabat, K., Orlowska-Kowalska, T.: Constrained model predictive control of the drive system with mechanical elasticity. IEEE Trans. Ind. Electron. **56**(6), 1963–1973 (2009)
24. Domahidi, A., Zgraggen, A., Zeilinger, M., Morari, M., Jones, C.: Efficient interior point methods for multistage problems arising in receding horizon control. In: Proc. IEEE Conf. Decis. Contr., pp. 668–674. Maui, HI, USA (2012)
25. Dua, V., Bozinis, N.A., Pistikopoulos, E.N.: A multiparametric programming approach for mixed-integer quadratic engineering problems. Comput. and Chemical Eng. **26**(4–5), 715–733 (2002)
26. Dua, V., Pistikopoulos, E.N.: An algorithm for the solution of multiparametric mixed integer linear programming problems. Ann. of Op.-Res. pp. 123–139 (2000)
27. Fincke, U., Pohst, M.: Improved methods for calculating vectors of short length in a lattice, including a complexity analysis. Math. Comput. **44**(170), 463–471 (1985)
28. Fletcher, R.: A general quadratic programming algorithm. IMA J. Appl. Math. **7**(1), 76–91 (1971)
29. Fuentes, E.J., Silva, C.A., Yuz, J.I.: Predictive Speed Control of a Two-Mass System Driven by a Permanent Magnet Synchronous Motor. IEEE Transactions on Industrial Electronics **59**(7), 2840–2848 (2012)
30. Garofalo, F., Marino, P., Scala, S., Vasca, F.: Control of DC/DC converters with linear optimal feedback and nonlinear feedforward. IEEE Trans. Power Electron. **9**(6), 607–615 (1994)
31. Gersho, A., Gray, R.M.: Vector Quantization and Signal Compression. Kluwer Academic, Boston, MA (1992)
32. Geyer, T.: Low complexity model predictive control in power electronics and power systems. Ph.D. thesis, Autom. Control Lab. ETH Zurich (2005)
33. Geyer, T.: Generalized model predictive direct torque control: Long prediction horizons and minimization of switching losses. In: Proc. IEEE Conf. Decis. Control, pp. 6799–6804. Shanghai, China (2009)
34. Geyer, T.: A comparison of control and modulation schemes for medium-voltage drives: Emerging predictive control concepts versus PWM-based schemes. IEEE Trans. Ind. Applicat. **47**(47), 3 (2011)
35. Geyer, T.: A comparison of control and modulation schemes for medium-voltage drives: emerging predictive control concepts versus PWM-based schemes. IEEE Trans. Ind. Appl. **47**(3), 1380–1389 (2011)
36. Geyer, T.: Computationally efficient model predictive direct torque control. IEEE Trans. Power Electron. **26**(10), 2804–2816 (2011)
37. Geyer, T.: Model predictive direct current control: formulation of the stator current bounds and the concept of the switching horizon. IEEE Ind. Appl. Mag. **18**(2), 47–59 (2012)
38. Geyer, T., Aguilera, R.P., Quevedo, D.E.: On the stability and robustness of model predictive direct current control. In: Proc. IEEE Int. Conf. Ind. Technol. Cape Town, South Africa (2013)
39. Geyer, T., Oikonomou, N., Papafotiou, G., Kieferndorf, F.: Model predictive pulse pattern control. IEEE Trans. Ind. Appl. **48**(2), 663–676 (2012)
40. Geyer, T., Papafotiou, G.: Direct torque control for induction motor drives: A model predictive control approach based on feasibility. In: M. Morari, L. Thiele (eds.) Hybrid Syst.: Comput. and Control, *LNCS*, vol. 3414, pp. 274–290. Springer (2005)
41. Geyer, T., Papafotiou, G., Frasca, R., Morari, M.: Constrained optimal control of the step-down dc-dc converter. IEEE Trans. Power Electron. **23**(5), 2454–2464 (2008)
42. Geyer, T., Papafotiou, G., Morari, M.: On the optimal control of switch-mode dc-dc converters. In: R. Alur, G. Pappas (eds.) Hybrid Syst.: Comput. and Control, *LNCS*, vol. 2993, pp. 342–356. Springer (2004)
43. Geyer, T., Papafotiou, G., Morari, M.: Hybrid model predictive control of the step-down dc-dc converter. IEEE Trans. Contr. Syst. Technol. **16**(6), 1112–1124 (2008)
44. Geyer, T., Papafotiou, G., Morari, M.: Model predictive direct torque control—Part I: Concept, algorithm and analysis. IEEE Trans. Ind. Electron. **56**(6), 1894–1905 (2009). DOI 10.1109/TIE.2008.2007030

45. Geyer, T., Quevedo, D.E.: Multistep direct model predictive control for power electronics – Part 1: Algorithm. In: Proc. of the Energy Conversion Congress and Exposition (ECCE) (2013)
46. Geyer, T., Quevedo, D.E.: Multistep direct model predictive control for power electronics – Part 2: Analysis. In: Proc. of the Energy Conversion Congress and Exposition (ECCE) (2013)
47. Geyer, T., Scoltock, J., Madawala, U.: Model predictive direct power control for grid-connected converters. In: Proc. IEEE Ind. Electron. Soc. Annu. Conf. Melbourne, Australia (2011)
48. Geyer, T., Torrisi, F., Morari, M.: Optimal complexity reduction of polyhedral piecewise affine systems. Automatica **44**(7), 1728–1740 (2008)
49. Geyer, T., Torrisi, F., Morari, M.: Efficient mode enumeration of compositional hybrid systems. Int. J. of Control **83**(2), 313–329 (2010)
50. Goodwin, G., Mayne, D., Chen, K.Y., Coates, C., Mirzaeva, G., Quevedo, D.: An introduction to the control of switching electronic systems. Ann. Rev. in Control **34**(2), 209–220 (2010)
51. Goodwin, G.C., Serón, M.M., De Doná, J.A.: Constrained Control & Estimation – An Optimization Perspective. Springer Verlag, London (2005)
52. Grüne, L., Panne, J.: Nonlinear Model Predictive Control: Theory and Algorithms. Springer (2011)
53. Grüne, L., Rantzer, A.: On the infinite horizon performance of receding horizon controllers. IEEE Trans. Automat. Contr. **53**(9), 2100–2111 (2008)
54. Hassibi, B., Vikalo, H.: On the sphere-decoding algorithm I. Expected complexity. IEEE Trans. Sign. Process. **53**(8), 2806–2818 (2005)
55. Heemels, W.P.M.H., De Schutter, B., Bemporad, A.: Equivalence of hybrid dynamical models. Automatica **37**(7), 1085–1091 (2001)
56. Holtz, J.: The representation of AC machine dynamics by complex signal graphs. IEEE Trans. Ind. Electron. **42**(3), 263–271 (1995)
57. Hu, J.S., Chen, K.Y.: Analysis and design of the receding horizon constrained optimization for class-D amplifier driving signals. Digital Signal Processing **20**, 1511–1525 (2010)
58. IBM ILOG, Inc.: CPLEX 12.5 User Manual. Somers, NY, USA (2012)
59. Jeong, S.G., Woo, M.H.: DSP-based active power filter with predictive current control. IEEE Transactions on Industrial Electronics **44**(3), 329–336 (1997)
60. Jerez, J., Constantinides, G., Kerrigan, E.: An FPGA implementation of a sparse quadratic programming solver for constrained predictive control. In: Proc. ACM/SIGDA Int. Symp. FPGA, pp. 209–218 (2011)
61. Karamanakos, P., Geyer, T., Manias, S.: Direct voltage control of DC-DC boost converters using model predictive control based on enumeration. In: Proc. Int. Power Electron. and Motion Control Conf. Novi Sad, Serbia (2012)
62. Karmarkar, N.: A new polynomial-time algorithm for linear programming. In: Proc. ACM Symp. Theory Comput., pp. 302–311 (1984)
63. Karush, W.: Minima of functions of several variables with inequalities as side constraints. Master's thesis, Univ. of Chicago, Chicago, IL, USA (1939)
64. Kennel, R., Linder, A., Linke, M.: Generalized predictive control (GPC) – ready for use in drive applications ? In: Proc. IEEE Power Electronics Specialists Conference (PESC), pp. 1839–1844. Vancouver, Canada (2001)
65. Kieferndorf, F., Karamanakos, P., Bader, P., Oikonomou, N., Geyer, T.: Model predictive control of the internal voltages of a five-level active neutral point clamped converter. In: Proc. IEEE Energy Convers. Congr. Expo. Raleigh, NC, USA (2012)
66. Kouro, S., Cortés, P., Vargas, R., Ammann, U., Rodríguez, J.: Model Predictive Control—A Simple and Powerful Method to Control Power Converters. IEEE Transactions on Industrial Electronics **56**(6), 1826–1838 (2009)
67. Kozlov, M., Tarasov, S., Khachiyan, L.: The polynomial solvability of convex quadratic programming. USSR Comput. Math. and Math. Physics **20**(5), 223–228 (1980)

68. Krause, P.C., Wasynczuk, O., Sudhoff, S.D.: Analysis of Electric Machinery and Drive Systems, 2nd edn. Intersci. Publ. John Wiley & Sons Inc. (2002)
69. Kuhn, H., Tucker, A.: Nonlinear programming. In: Proc. 2nd Berkeley Symp. Math. Statistics and Probability, pp. 481–492 (1951)
70. Kukrer, O.: Discrete-time current control of voltage-fed three-phase PWM inverters. IEEE Transactions on Power Electronics **11**(2), 260–269 (1996)
71. Kvasnica, M., Grieder, P., Baotić, M., Morari, M.: Multi parametric toolbox (MPT). In: R. Alur, G. Pappas (eds.) Hybrid Syst.: Comput. and Control, *LNCS*, vol. 2993, pp. 448–462. Springer (2004). http://control.ee.ethz.ch/ mpt
72. Lawler, E.L., Wood, D.E.: Branch and bound methods: A survey. Op. Res. **14**(4), 699–719 (1966)
73. Lehman, B., Bass, R.M.: Extensions of averaging theory for power electronic systems. IEEE Trans. Power Electron. **11**(4), 542–553 (1996)
74. Lezana, P., Aguilera, R.P., Quevedo, D.E.: Model predictive control of an asymmetric flying capacitor converter. IEEE Trans. Ind. Electron. **56**(6), 1839–1846 (2009)
75. Linder, A., Kennel, R.: Model predictive control for electrical drives. In: Proc. IEEE Power Electronics Specialists Conference (PESC), pp. 1793–1799. Recife, Brazil (2005)
76. Low, K.S., Zhuang, X.: Robust model predictive control and observer for direct drive applications. IEEE Trans. Power Electron. **15**(6), 1018–1028 (2000)
77. Maciejowski, J.M.: Predictive Control with Constraints. Prentice-Hall, Englewood Cliffs, N. J. (2002)
78. Malesani, L., Mattavelli, P., Buso, S.: Robust dead-beat current control for PWM rectifiers and active filters. IEEE Transactions on Industry Applications **35**(3), 613–620 (1999)
79. Mariéthoz, S., Almér, S., Bâja, M., Beccuti, A., Patino, D., Wernrud, A., Buisson, J., Cormerais, H., Fujioka, H., Geyer, T., Jönsson, U., Kao, C.Y., Morari, M., Papafotiou, G., Rantzer, A., Riedinger, P.: Comparison of hybrid control techniques for buck and boost DC-DC converters. IEEE Trans. Contr. Syst. Technol. **18**(5), 1126–1145 (2010)
80. Mariethoz, S., Beccuti, A., Papafotiou, G., Morari, M.: Sensorless explicit model predictive control of the dc-dc buck converter with inductor current limitation. In: Proc. App. Power Electron. Conf. and Expo., pp. 1710–1715 (2008)
81. Mariéthoz, S., Domahidi, A., Morari, M.: High-bandwidth explicit model predictive control of electrical drives. IEEE Trans. Ind. Appl. **48**(6), 1980–1992 (2012)
82. Mariéthoz, S., Morari, M.: Explicit model predictive control of a PWM inverter with an LCL filter. IEEE Trans. Ind. Electron. **56**(2), 389–399 (2009)
83. Mattavelli, P.: An improved deadbeat control for UPS using disturbance observers. IEEE Transactions on Industrial Electronics **52**(1), 206–212 (2005)
84. Mayne, D.Q., Rawlings, J.B., Rao, C.V., Scokaert, P.O.M.: Constrained model predictive control: Optimality and stability. Automatica **36**(6), 789–814 (2000)
85. Mitten, L.G.: Branch–and–bound methods: General formulation and properties. Op. Res. **18**(1), 24–34 (1970)
86. Mohamed, Y.A.R.I., El-Saadany, E.F.: An Improved Deadbeat Current Control Scheme With a Novel Adaptive Self-Tuning Load Model for a Three-Phase PWM Voltage-Source Inverter. IEEE Transactions on Industrial Electronics **54**(2), 747–759 (2007)
87. Mossoba, J., Lehn, P.W.: A controller architecture for high bandwidth active power filters. IEEE Transactions on Power Electronics **18**(1), 317–325 (2003)
88. Murata, T., Tsuchiya, T., Takeda, I.: Vector control for induction machine on the application of optimal control theory. IEEE Trans. Ind. Electron. **37**(4), 283–290 (1990)
89. Nesterov, Y.: A method of solving a convex programming problem with convergence rate $O(1/k^2)$. Soviet Math. Dokl. **27**(2), 372–386 (1983)
90. Nesterov, Y., Nemirovskii, A.: Interior-point polynomial algorithms in convex programming. SIAM (1994)
91. Nishida, Y., Miyashita, O., Haneyoshi, T., Tomita, H., Maeda, A.: A predictive instantaneous-current PWM controlled rectifier with AC-side harmonic current reduction. IEEE Transactions on Industrial Electronics **44**(3), 337–343 (1997)

92. Oikonomou, N., Gutscher, C., Karamanakos, P., Kieferndorf, F., Geyer, T.: Model predictive pulse pattern control for the five-level active neutral point clamped inverter. In: Proc. IEEE Energy Convers. Congr. Expo. Raleigh, NC, USA (2012)
93. Østergaard, J., Quevedo, D.E., Jensen, J.: Real-time perceptual moving-horizon multiple-description audio coding. IEEE Trans. Signal Processing **59**(9), 4286–4299 (2011)
94. Papafotiou, G., Geyer, T., Morari, M.: Optimal direct torque control of three-phase symmetric induction motors. In: Proc. IEEE Conf. Decis. Control. Atlantis, Bahamas (2004)
95. Papafotiou, G., Geyer, T., Morari, M.: A hybrid model predictive control approach to the direct torque control problem of induction motors (invited paper). Int. J. of Robust Nonlinear Control **17**(17), 1572–1589 (2007)
96. Papafotiou, G., Kley, J., Papadopoulos, K.G., Bohren, P., Morari, M.: Model predictive direct torque control—Part II: Implementation and experimental evaluation. IEEE Trans. Ind. Electron. **56**(6), 1906–1915 (2009). DOI 10.1109/TIE.2008.2007032
97. Patel, H.S., Hoft, R.G.: Generalized techniques of harmonic elimination and voltage control in thyristor inverters: Part I–Harmonic elimination. IEEE Trans. Ind. Appl. **9**(3), 310–317 (1973)
98. Qin, S.J., Badgwell, T.A.: A survey of industrial model predictive control technology. Contr. Eng. Pract. **11**, 733–764 (2003)
99. Quevedo, D.E., Aguilera, R.P., Pérez, M.A., Cortés, P., Lizana, R.: Model predictive control of an AFE rectifier with dynamic references. IEEE Trans. Power Electron. **27**(7), 3128–3136 (2012)
100. Quevedo, D.E., Bölcskei, H., Goodwin, G.C.: Quantization of filter bank frame expansions through moving horizon optimization. IEEE Trans. Signal Processing **57**(2), 503–515 (2009)
101. Quevedo, D.E., Goodwin, G.C.: Control of EMI from switch-mode power supplies via multi-step optimization. In: Proc. Amer. Contr. Conf., vol. 1, pp. 390–395. Boston, MA (2004)
102. Quevedo, D.E., Goodwin, G.C.: Moving horizon design of discrete coefficient FIR filters. IEEE Trans. Signal Processing **53**(6), 2262–2267 (2005)
103. Quevedo, D.E., Goodwin, G.C.: Multistep optimal analog-to-digital conversion. IEEE Trans. Circuits Syst. I **52**(4), 503–515 (2005)
104. Quevedo, D.E., Goodwin, G.C., De Doná, J.A.: Finite constraint set receding horizon quadratic control. Int. J. Robust Nonlin. Contr. **14**(4), 355–377 (2004)
105. Quevedo, D.E., Goodwin, G.C., De Doná, J.A.: Multistep detector for linear ISI-channels incorporating degrees of belief in past estimates. IEEE Trans. Commun. **55**(11), 2092–2103 (2007)
106. Quevedo, D.E., Nešić, D.: Robust stability of packetized predictive control of nonlinear systems with disturbances and Markovian packet losses. Automatica **48**(8), 1803–1811 (2012)
107. Quevedo, D.E., Østergaard, J., Nešić, D.: Packetized predictive control of stochastic systems over bit-rate limited channels with packet loss. IEEE Trans. Automat. Contr. **56**(12), 2854–2868 (2011)
108. Rawlings, J.B., Mayne, D.Q.: Model Predictive Control: Theory and Design. Nob Hill Publishing, Madison, WI (2009)
109. Richter, S., Jones, C., Morari, M.: Real-time input-constrained MPC using fast gradient methods. In: Proc. IEEE Conf. Decis. Contr., pp. 7387–7393. Shanghai, China (2009)
110. Richter, S., Mariéthoz, S., Morari, M.: High-speed online MPC based on fast gradient method applied to power converter control. In: Proc. Am. Control Conf. Baltimore, MD, USA (2010)
111. Rodríguez, J., Cortés, P.: Predictive Control of Power Converters and Electrical Drives, 1 edn. Wiley-IEEE Press (2012)
112. Rodríguez, J., Kazmierkowski, M.P., Espinoza, J., Zanchetta, P., Abu-Rub, H., Young, H.A., Rojas, C.A.: State of the Art of Finite Control Set Model Predictive Control in Power Electronics. IEEE Transactions on Industrial Informatics **9**(2), 1003–1016 (2013)
113. Rodríguez, J., Pontt, J., Silva, C., Correa, P., Lezana, P., Cortés, P., Ammann, U.: Predictive Current Control of a Voltage Source Inverter. IEEE Transactions on Industrial Electronics **54**(1), 495–503 (2007)

114. Saggini, S., Stefanutti, W., Tedeschi, E., Mattavelli, P.: Digital Deadbeat Control Tuning for dc-dc Converters Using Error Correlation. IEEE Transactions on Power Electronics **22**(4), 1566–1570 (2007)
115. Scoltock, J., Geyer, T., Madawala, U.: A comparison of model predictive control schemes for MV induction motor drives. IEEE Trans. Ind. Inf. **9**(2), 909–919 (2013)
116. Serón, M., Goodwin, G.C., De Doná, J.A.: Characterisation of receding horizon control for constrained linear systems. Asian J. Contr. **5**(2), 271–286 (2003)
117. Sontag, E.: Nonlinear regulation: The piecewise linear approach. IEEE Trans. Automat. Contr. **26**(2), 346–358 (1981)
118. Stolze, P., Landsmann, P., Kennel, R., Mouton, T.: Finite-set model predictive control of a flying capacitor converter with heuristic voltage vector preselection. In: Proc. IEEE Int. Conf. on Power Electron. and ECCE Asia (2011)
119. Sturm, J.: Using SeDuMi 1.02, a MATLAB toolbox for optimization over symmetric cones (1998)
120. Takahashi, I., Noguchi, T.: A new quick response and high efficiency control strategy for the induction motor. IEEE Trans. Ind. Appl. **22**(2), 820–827 (1986)
121. Tøndel, P., Johansen, T.A., Bemporad, A.: An algorithm for multi-parametric quadratic programming and explicit MPC solutions. In: Proc. IEEE Conf. Decis. Control. Orlando, Florida (2001)
122. Tøndel, P., Johansen, T.A., Bemporad, A.: Evaluation of piecewise affine control via binary search tree. Automatica **39**(5), 945–950 (2003)
123. Torrisi, F.D., Bemporad, A.: Hysdel — a tool for generating computational hybrid models for analysis and synthesis problems. IEEE Trans. Contr. Syst. Technol. **12**(2), 235–249 (2004)
124. Wächter, A., Biegler, L.: On the implementation of a primal-dual interior point filter line search algorithm for large-scale nonlinear programming. Math. Program. **106**(1), 25–57 (2006)
125. Yang, S.M., Lee, C.H.: A deadbeat current controller for field oriented induction motor drives. IEEE Transactions on Power Electronics **17**(5), 772–778 (2002)
126. Zeinaly, Y., Geyer, T., Egardt, B.: Trajectory extension methods for model predictive direct torque control. In: Proc. App. Power Electron. Conf. and Expo., pp. 1667–1674. Fort Worth, TX, USA (2011)
127. Zeng, Q., Chang, L.: An Advanced SVPWM-Based Predictive Current Controller for Three-Phase Inverters in Distributed Generation Systems. IEEE Transactions on Industrial Electronics **55**(3), 1235–1246 (2008)