Long-horizon direct model predictive control with active balancing of the neutral point potential

Long-Horizon Direct Model Predictive Control With Active Balancing of the Neutral Point Potential

Eyke Liegmann, *Student Member, IEEE*, Petros Karamanakos, *Member, IEEE*,
Tobias Geyer, *Senior Member, IEEE*, Toit Mouton, *Member, IEEE*, and Ralph Kennel, *Senior Member, IEEE*

***Abstract*—In this paper we present modifications to the sphere decoder initially introduced in [1] to include the control of the neutral point (NP) potential of a three-level neutral point clamped (NPC) inverter. By linearizing the system model, the nonlinearities introduced by the dynamics of the NP potential are discarded. As a result, the optimization problem underlying direct model predictive control (MPC) can be formulated as an integer least-squares (ILS) one, and solved in a computationally efficient manner with a refined sphere decoding algorithm. As shown, thanks to the utilization of long prediction horizons, the system performance can be significantly improved. This is demonstrated with a variable speed drive consisting of a three-level NPC inverter and a medium-voltage induction machine.**

I. Introduction

In recent years, model predictive control (MPC) [2] has reached increasing popularity in the field of power electronics due to its ability to handle nonlinear systems with multiple inputs and outputs [3]. The control objectives are mapped into a scalar by an objective function and can be met by solving the formulated optimization problem. Moreover, incorporation of soft and hard constraints in the problem facilitates operation at the physical limitations of the system while ensuring optimal performance and, ultimately, allowing for smaller safety margins. As the name MPC suggests, a model of the plant is used to predict its future behavior based on which the optimal control action is chosen.

In this paper, we focus on the so called finite control set MPC (FCS-MPC) [3]. According to this method, the control inputs are modeled as integers that translate directly to the switch positions of the inverter, bypassing any modulation stage. This implies that the underlying optimization problem is an integer program. The latter is most often solved with the brute-force approach of exhaustive enumeration in which all candidate solutions are tested before concluding to the optimal one. With that optimizer, however, and given that a long horizon is beneficial for the overall performance of the system under study [4]–[7], the problem can become computationally intractable since the size of the candidate set increases exponentially with the horizon. One possible way to render the problem computationally feasible is to employ an effective branch-and-bound method referred to as sphere decoding [8]. This algorithm excludes a large number of candidate solutions from the very early stages of the search process, thus significantly reducing the computational burden.

In previous publications on the sphere decoder [1], [6], [7]—dealing with a three-level neutral point clamped (NPC) converter—it is assumed that the neutral point (NP) potential is fixed. The reason for that is that the sphere decoder can be employed only if the optimization problem is formulated as an integer least-squares (ILS) one. To achieve the latter, the plant should be modeled as a linear dynamical system. However, this assumption does not fully represent the physical behavior of a three-level NPC. In reality, the NP floats and its potential depends on the switch positions as well as the load of the inverter. Without proper control strategy the NP potential deviates which can lead to a large voltage unbalance and may create an overvoltage across one or more of the switches.

To overcome this issue, in this paper the dynamics of the NP are taken into account [9]–[11], and in a subsequent step are linearized. By doing so, the derived linearized model can be used to formulate an ILS problem that can be solved by a modified sphere decoder. To highlight the effectiveness of the proposed control strategy, a variable speed drive system, consisting of a three-level NPC voltage source inverter and a medium-voltage (MV) induction machine (IM), is chosen as a case study. The presented simulation results show the improved performance of the plant as compared to that achieved with the short-horizon FCS-MPC that utilizes the nonlinear model of the drive.

II. Control Model

The problem examined relates to the control of the stator current of an IM and the NP potential of the three-level NPC inverter that drives the machine (Fig. 1). In the sequel, the mathematical model of the system is derived. This task is performed in the stationary orthogonal $\alpha\beta$-plane. Therefore, any variable in the abc-plane $\boldsymbol{\xi}_{abc} = [\xi_a \ \xi_b \ \xi_c]^T$ is mapped into a two-dimensional vector $\boldsymbol{\xi}_{\alpha\beta} = [\xi_\alpha \ \xi_\beta]^T$ in the $\alpha\beta$-plane via the transformation matrix $\boldsymbol{K}$, i.e., $\boldsymbol{\xi}_{\alpha\beta} = \boldsymbol{K}\boldsymbol{\xi}_{abc}$[1]

$$\boldsymbol{K} = \frac{2}{3}\begin{bmatrix} 1 & -\frac{1}{2} & -\frac{1}{2} \\ 0 & \frac{\sqrt{3}}{2} & -\frac{\sqrt{3}}{2} \end{bmatrix}. \tag{1}$$

[1]Hereafter, the subscript the $\alpha\beta$ is dropped from vectors in $\alpha\beta$-plane to simplify the notation.

E. Liegmann and R. Kennel are with the Chair of Electrical Drive Systems and Power Electronics, Technical University Munich, 80333 Munich, Germany; e-mail: eyke.liegmann@tum.de and ralph.kennel@tum.de

P. Karamanakos is with the Faculty of Computing and Electrical Engineering, Tampere University of Technology, 33101 Tampere, Finland; e-mail: p.karamanakos@ieee.org

T. Geyer is with ABB Corporate Research, 5405 Baden-Dättwil, Switzerland; e-mail: t.geyer@ieee.org

T. Mouton is with the Department of Electrical and Electronic Engineering, University of Stellenbosch, 7602 Stellenbosch, South Africa; e-mail: dtmouton@sun.ac.za

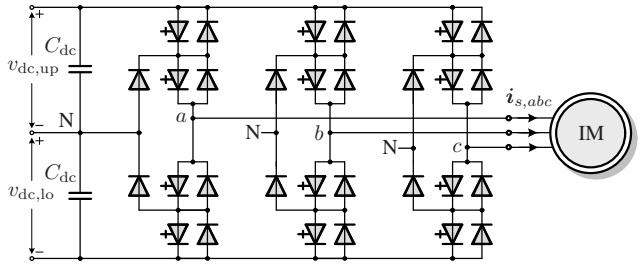


Fig. 1: Three-level three-phase neutral point clamped (NPC) voltage source inverter driving an induction motor (IM).

A. Nonlinear Model of the Drive System

Using the stator current $\boldsymbol{i}_s$ and the rotor flux $\boldsymbol{\psi}_r$ as state variables, the dynamics of the IM are described by[2] [12]

$$\frac{\mathrm{d}\boldsymbol{i}_s}{\mathrm{d}t} = -\frac{1}{\tau_s}\boldsymbol{i}_s + \left(\frac{1}{\tau_r}\boldsymbol{I} - \omega_r \begin{bmatrix} 0 & -1 \\ 1 & 0 \end{bmatrix}\right)\frac{X_m}{D}\boldsymbol{\psi_r} + \frac{X_r}{D}\boldsymbol{v}_s \tag{2a}$$

$$\frac{\mathrm{d}\boldsymbol{\psi}_r}{\mathrm{d}t} = \frac{X_m}{\tau_r}\boldsymbol{i}_s - \frac{1}{\tau_r}\boldsymbol{\psi_r} + \omega_r \begin{bmatrix} 0 & -1 \\ 1 & 0 \end{bmatrix}\boldsymbol{\psi_r}\,. \tag{2b}$$

Note that the rotor angular speed ω_r is considered to be a time-varying parameter. The input to the drive system is the three-phase switch position $\boldsymbol{u}_{abc} = [u_a\ u_b\ u_c]^T$, with $u_a, u_b, u_c \in \mathcal{U} = \{-1, 0, 1\}$. Moreover, $\boldsymbol{u}'$ is introduced to model the componentwise absolute value of the input vector, i.e.,

$$\boldsymbol{u}' = \begin{bmatrix} u'_\alpha \\ u'_\beta \end{bmatrix} = \boldsymbol{K}\begin{bmatrix} |u_a| \\ |u_b| \\ |u_c| \end{bmatrix}\,. \tag{3}$$

Moreover, v_n is the NP potential, defined as

$$v_n = v_{\mathrm{dc,lo}} - v_{\mathrm{dc,up}}\,, \tag{4}$$

where $v_{\mathrm{dc,lo}}$ and $v_{\mathrm{dc,up}}$ are the voltages of the lower and upper dc-link capacitors C_{dc}, respectively (Fig. 1). Here, the dc-link voltage $V_{\mathrm{dc}} = v_{\mathrm{dc,lo}} + v_{\mathrm{dc,up}}$ is assumed to be constant.

As can be seen in (2), the evolution of the stator current depends on the applied stator voltage $\boldsymbol{v}_s$, i.e., the output voltage of the inverter, defined as

$$\boldsymbol{v}_s = \boldsymbol{u}\,\frac{V_{\mathrm{dc}}}{2} - \boldsymbol{u}'\,\frac{v_n}{2}\,, \tag{5}$$

with $\boldsymbol{u} = \boldsymbol{K}\boldsymbol{u}_{abc}$.

The evolution of the NP potential depends on the current flowing into the center point of the capacitor bank [9], i.e.,

$$\frac{\mathrm{d}v_n}{\mathrm{d}t} = \frac{1}{C_{\mathrm{dc}}}(i_{sa}|u_a| + i_{sb}|u_b| + i_{sc}|u_c|) \overset{(3)}{=} \frac{1}{C_{\mathrm{dc}}}(i_{s\alpha}u'_\alpha + i_{s\beta}u'_\beta)\,. \tag{6}$$

B. Linearization and Discretization

The dynamics of the drive system include nonlinearities, as can be seen in (2) (see (5)) and (6). Specifically, they include multiplications of the state variables, i.e., v_n and $\boldsymbol{i}_s$, with $\boldsymbol{u}'$, i.e., the absolute value of the input vector. To derive a linear model of the system—which is a prerequisite for the optimization problem to be formulated as an ILS one so that it can be solved in a computational efficient manner [1]—these nonlinear terms should be removed. To this end, the aforementioned nonlinear terms are linearized using the first order Taylor expansion at $t = t_0$. For example, linearizing the term $i_{s\alpha}(t)u'_\alpha(t)$ in (6) yields

$$\begin{aligned} i_{s\alpha}(t)u'_\alpha(t) &\approx i_{s\alpha}(t_0)u'_\alpha(t_0) + i_{s\alpha}(t_0)\Delta u'_\alpha(t) + \Delta i_{s\alpha}(t)u'_\alpha(t_0) \\ &\approx i_{s\alpha}(t_0)u'_\alpha(t_0) + i_{s\alpha}(t_0)\Delta u'_\alpha(t) + \big(i_{s\alpha}(t) - i_{s\alpha}(t_0)\big)u'_\alpha(t_0) \\ &\approx i_{s\alpha}(t)u'_\alpha(t_0) + i_{s\alpha}(t_0)\Delta u'_\alpha(t)\,, \end{aligned} \tag{7}$$

where $\Delta u'_\alpha(t) = u'_\alpha(t) - u'_\alpha(t_0)$ and $\Delta i_{s\alpha}(t) = i_{s\alpha}(t) - i_{s\alpha}(t_0)$. In a similar manner the remaining nonlinear terms $i_\beta(t)u'_\beta(t)$ in (6) and $\boldsymbol{u}'v_n$ in (5) are linearized at time-instant t_0.

In a next step, and in order to derive the linearized state-space model of the drive, we define the state vector to be $\boldsymbol{x} = [i_{s\alpha}\ i_{s\beta}\ \psi_{r\alpha}\ \psi_{r\beta}\ v_n]^T$, which consists of the stator current $\boldsymbol{i}_s$, the rotor flux $\boldsymbol{\psi}_r$, and the NP potential v_n. Moreover, we introduce the augmented input vector $\boldsymbol{u}_{\mathrm{aug}} = [u_a\ u_b\ u_c\ \Delta|u_a|\ \Delta|u_b|\ \Delta|u_c|]^T$, where the first three entries are the three-phase switch position $\boldsymbol{u}_{abc}$ and last three entries are defined as the difference of the absolute values of two consecutive switchings, i.e.,

$$\Delta|u_x(k)| = |u_x(k)| - |u_x(k-1)|\,, \tag{8}$$

with $x \in \{a, b, c\}$ and $k \in \mathbb{N}$. We refer to these entries in (8) as "pseudo-inputs". Since they are fully defined by the current and previous switch position, they are not independent inputs to the system. Rather they are introduced to formulate the system dynamics in a compact state-space representation. Note that the reasons behind the choice of this—augmented—input vector will become more apparent in Section III-A. Finally, the stator current and NP potential are chosen as the output variables, i.e., $\boldsymbol{y} = [i_{s\alpha}\ i_{s\beta}\ v_n]^T$.

Based on the above, the resulting linearized, continuous-time state-space model is of the form

$$\frac{\mathrm{d}\boldsymbol{x}(t)}{\mathrm{d}t} = \boldsymbol{F}(t_0)\,\boldsymbol{x}(t) + \boldsymbol{G}(t_0)\,\boldsymbol{u}_{\mathrm{aug}}(t)\ ,\quad \boldsymbol{y}(t) = \boldsymbol{C}\,\boldsymbol{x}(t)\,, \tag{9}$$

where matrices $\boldsymbol{F}(t_0)$, $\boldsymbol{G}(t_0)$,[3] and $\boldsymbol{C}$ are defined as

$$\boldsymbol{F} = \begin{bmatrix} & \boldsymbol{F}_{\mathrm{IM}} & & -\frac{X_r}{2D}u'_\alpha(t_0) \\ & & & -\frac{X_r}{2D}u'_\beta(t_0) \\ & & & 0 \\ & & & 0 \\ \frac{1}{C_{\mathrm{dc}}}u'_\alpha(t_0) & \frac{1}{C_{\mathrm{dc}}}u'_\beta(t_0) & 0 & 0 \end{bmatrix}, \tag{10}$$

$$\boldsymbol{G} = \begin{bmatrix} \frac{X_r V_{\mathrm{dc}}}{2D} & 0 & -\frac{X_r}{2D}v_n(t_0) & 0 \\ 0 & \frac{X_r V_{\mathrm{dc}}}{2D} & 0 & -\frac{X_r}{2D}v_n(t_0) \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & \frac{1}{C_{\mathrm{dc}}}i_{s\alpha}(t_0) & \frac{1}{C_{\mathrm{dc}}}i_{s\beta}(t_0) \end{bmatrix}\boldsymbol{K}_{\mathrm{aug}},$$

[2]The definition of all parameters in (2) can be found in [1].

[3]Matrices $\boldsymbol{F}(t_0)$, and $\boldsymbol{G}(t_0)$ are time-varying. To simplify the notation, for these matrices—as well as all time-varying matrices in this work—the time dependency is dropped after the first indication.

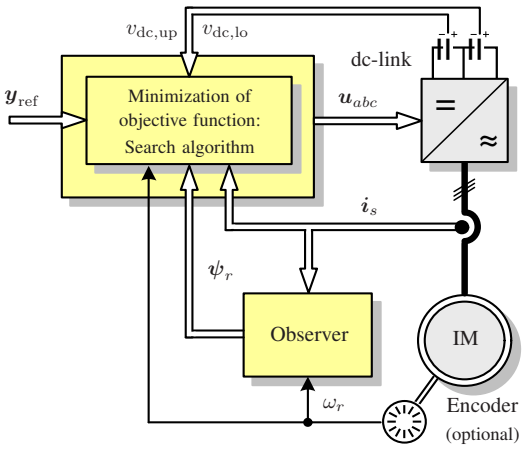


Fig. 2: Model predictive current control with reference tracking for the three-phase three-level NPC inverter with an IM.

$$\boldsymbol{K}_{\mathrm{aug}} = \begin{bmatrix} \boldsymbol{K} & \boldsymbol{0}_{2\mathrm{x}3} \\ \boldsymbol{0}_{2\mathrm{x}3} & \boldsymbol{K} \end{bmatrix}, \quad \boldsymbol{C} = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix}.$$

In (10), the 4×4 matrix $\boldsymbol{F}_{\mathrm{IM}}$ includes the dynamics of the IM (2), as defined in the appendix of [1]. Moreover, $\boldsymbol{0}$ is the zero matrix of dimensions defined by the corresponding subscript.

The next step is to discretize (9). All time-varying quantities—i.e., $\boldsymbol{u}'(t_0)$, $\boldsymbol{i}_s(t_0)$, and $v_n(t_0)$—are sampled at time-instant $t_0 \equiv k$. We assume that the measured quantities and the resulting matrices remain constant over the prediction horizon. Note that the switch position is held constant over one sampling interval which allows for the simplification $\boldsymbol{u}'(t_0) = \boldsymbol{u}'(k)$.

By using exact Euler discretization, the discrete-time state-space model of the drive becomes

$$\boldsymbol{x}(k+1) = \boldsymbol{A}\,\boldsymbol{x}(k) + \boldsymbol{B}\,\boldsymbol{u}_{\mathrm{aug}}(k), \;\; \boldsymbol{y}(k) = \boldsymbol{C}\,\boldsymbol{x}(k)\,, \quad (11)$$

where matrices $\boldsymbol{A}(k)$ and $\boldsymbol{B}(k)$ are

$$\boldsymbol{A} = \mathbf{e}^{\boldsymbol{F}T_s}, \;\; \boldsymbol{B} = \begin{cases} \displaystyle\int_0^{T_s} \mathbf{e}^{\boldsymbol{F}\tau}\mathrm{d}\tau\,\boldsymbol{G} & \text{if } \boldsymbol{u}' = \boldsymbol{0} \\ -\boldsymbol{F}^{-1}(\boldsymbol{I}-\boldsymbol{A})\boldsymbol{G} & \text{otherwise,} \end{cases} \quad (12)$$

where $\boldsymbol{I}$ is the identity matrix of appropriate dimensions and $\mathbf{e}$ the matrix exponential. It is important to point out that when $|u_a(k)| = |u_b(k)| = |u_c(k)|$, i.e., $\boldsymbol{u}' = \boldsymbol{0}$, the state matrix $\boldsymbol{F}$ —according to (10)—becomes singular. This is equivalent to the three-phase switch positions $\boldsymbol{u}_{abc}$ that have no influence on the NP potential [9]. To overcome this and still be able to derive $\boldsymbol{B}$, the latter is calculated in two different ways depending on $\boldsymbol{u}'(k)$, see (12). Finally, it should be noted that the linearization and discretization have to be recalculated whenever the controller is called since the state matrices are time-varying.

C. Direct MPC With Current Reference Tracking and NP Potential Balancing

The discussed control algorithm aims to regulate the stator current $\boldsymbol{i}_s$ along its reference $\boldsymbol{i}_{s,\mathrm{ref}}$ with the goal to achieve as low a current total harmonic distortion (THD) as possible. Moreover, the voltages over the upper and lower dc-link capacitors, $v_{\mathrm{dc,up}}$ and $v_{\mathrm{dc,lo}}$, respectively, should be kept balanced by minimizing the deviation of the NP potential v_n. Finally, the aforementioned objectives should be met while operating the drive at a low switching frequency. The latter objective is important when MV drives are of concern, since it is directly related to the switching power losses which have to be kept low for an increased efficiency of the converter. As can be seen in Fig. 2, the optimal controller meets all objectives by computing and applying the control signals (i.e., the switch positions) in one stage, i.e., a modulation stage is not required.

At time step k, the control objectives are mapped into a scalar by the objective function[4]

$$J(k) = \sum_{\ell=k}^{k+N-1} ||\boldsymbol{y}_{\mathrm{ref}}(\ell+1) - \boldsymbol{y}(\ell+1)||^2_{\boldsymbol{Q}} + \lambda_u ||\Delta\boldsymbol{u}_{abc}(\ell)||^2_2\,. \quad (13)$$

Based on the state of the current error and NP potential at step k, (13) penalizes their evolution over the finite prediction horizon of N time steps. The term $\Delta\boldsymbol{u}_{abc}$ penalizes the switching effort. The vector $\boldsymbol{y}_{\mathrm{ref}} = [i_{\alpha,\mathrm{ref}}\; i_{\beta,\mathrm{ref}}\; v_{n,\mathrm{ref}}]^T \in \mathbb{R}^3$ holds the reference values for the three output variables. The diagonal entries of the positive definite, diagonal weighting matrix $\boldsymbol{Q} = \mathrm{diag}(1, 1, \lambda_{\mathrm{dc}}) \in \mathbb{R}^{3\mathrm{x}3}$ penalize the deviation of the output variables from their respective references, i.e., $\boldsymbol{i}_s - \boldsymbol{i}_{s,\mathrm{ref}}$ and $v_n - v_{n,\mathrm{ref}}$.[5] Note that the weighting factors $\lambda_u, \lambda_{\mathrm{dc}} > 0$ are introduced to set the trade-off between the three terms, i.e., the trade-off between the current tracking ability of the controller, the NP potential balancing, and the switching frequency f_{sw}.

To find the optimal sequence of control actions $\boldsymbol{U}^*(k) = [\boldsymbol{u}^{*T}_{\mathrm{aug}}(k)\; \boldsymbol{u}^{*T}_{\mathrm{aug}}(k+1)\; \ldots\; \boldsymbol{u}^{*T}_{\mathrm{aug}}(k+N-1)]^T$ that results in the most desirable system behavior, the following problem needs to be solved in real time

$$\underset{\boldsymbol{U}(k)\in\mathbb{U}}{\text{minimize}} \;\; J(k) \quad (14a)$$

$$\text{subject to} \;\; \begin{aligned} &\boldsymbol{x}(\ell+1) = \boldsymbol{A}\boldsymbol{x}(\ell) + \boldsymbol{B}\boldsymbol{u}_{\mathrm{aug}}(\ell) \\ &\boldsymbol{y}(\ell+1) = \boldsymbol{C}\boldsymbol{x}(\ell+1) \end{aligned} \quad (14b)$$

$$||\Delta\boldsymbol{u}_{abc}(\ell)||_\infty \le 1\,, \;\; \forall\, \ell = k, \ldots, k+N-1\,. \quad (14c)$$

In (14), $\boldsymbol{U}(k) = [\boldsymbol{u}^T_{\mathrm{aug}}(k)\; \boldsymbol{u}^T_{\mathrm{aug}}(k+1)\; \ldots\; \boldsymbol{u}^T_{\mathrm{aug}}(k+N-1)]^T$ is the optimization variable and $\mathbb{U} = \boldsymbol{\mathcal{U}}^N \subset \mathbb{Z}^n$, with $n = 6N$, is the feasible set defined as the N-times Cartesian product of the set $\boldsymbol{\mathcal{U}} = \mathcal{U} \times \mathcal{U} \times \mathcal{U} \times \mathcal{U} \times \mathcal{U} \times \mathcal{U} = \mathcal{U}^6$. Moreover, constraint (14c) is added to prevent a solution that would cause a shoot-through in the converter.

Out of the optimal control sequence $\boldsymbol{U}^*(k)$ only the actions at time-step k, i.e., $\boldsymbol{u}^*_{\mathrm{aug}}(k)$ are of interest. More specifically, the first three elements $\boldsymbol{u}^*_{abc}(k)$ of $\boldsymbol{u}^*_{\mathrm{aug}}(k)$ are those that

[4]Note that $||\boldsymbol{\xi}||^2_{\boldsymbol{Q}} \;\widehat{=}\; \boldsymbol{\xi}^T\boldsymbol{Q}\boldsymbol{\xi}$ denotes the squared norm of the vector $\boldsymbol{\xi}$ weighted with the positive definite matrix $\boldsymbol{Q}$.

[5]The reference for the NP potential $v_{n,\mathrm{ref}}$ is zero.

are applied at the current time-step k, whereas all remaining elements are discarded. Then, the optimization is repeated over a one time-step shifted horizon based on new measurements and estimations. This principle is referred to as receding horizon control [2].

III. Equivalent Integer Least-Squares Problem

In this section, we formulate the optimization problem (14) in a vector form, and present it as a truncated ILS problem.

A. Optimization Problem in Vector Form

By denoting the output sequence over the prediction horizon as $\boldsymbol{Y}(k) = [\boldsymbol{y}^T(k+1) \ \dots \ \boldsymbol{y}^T(k+N)]$, it follows that

$$\boldsymbol{Y}(k) = \boldsymbol{\Gamma}(k)\,\boldsymbol{x}(k) + \boldsymbol{\Upsilon}(k)\,\boldsymbol{U}(k)\,, \tag{15}$$

where the matrices $\boldsymbol{\Gamma}(k)$ and $\boldsymbol{\Upsilon}(k)$ are given in the appendix. Since $\boldsymbol{U} \in \mathbb{U}$ and $||\Delta \boldsymbol{u}_{abc}(\ell)||_\infty \leq 1$, it is implied that the difference—in the Euclidean sense—of two consecutive "pseudo-inputs" yields the same cost as the difference of the corresponding switch positions, i.e., $||\Delta \boldsymbol{u}_{abc}(k)||_2^2 = ||\,\Delta|\boldsymbol{u}_{abc}(k)|\,||_2^2$. This allows us to include the full augmented input vector $\boldsymbol{u}_{\mathrm{aug}}(k)$ into the switching effort term of the function (13). By doing so, however, each switching event is counted twice. Therefore, the weighting factor is chosen equal to $\lambda_u' = \lambda_u/2$ in order to keep the contribution of the switching effort term to the total cost the unchanged. Note that including the "pseudo-inputs" into the objective function is important for employing the sphere decoder later on, since this allows the Hessian matrix in (19) to be nonsingular.

Using (15) and the output reference sequence over the prediction horizon $\boldsymbol{Y}^*(k)$, (13) becomes[4]

$$\begin{aligned} J = {} & ||\boldsymbol{\Gamma}\boldsymbol{x}(k) + \boldsymbol{\Upsilon}\boldsymbol{U}(k) - \boldsymbol{Y}^*(k)||_{\widetilde{\boldsymbol{Q}}}^2 \\ & + \lambda_u' ||\boldsymbol{S}\boldsymbol{U}(k) - \boldsymbol{E}\boldsymbol{u}_{\mathrm{aug}}(k-1)||_2^2\,, \end{aligned} \tag{16}$$

where $\widetilde{\boldsymbol{Q}} = \mathrm{diag}(\boldsymbol{Q}, \dots, \boldsymbol{Q})$, is a block diagonal matrix and

$$\boldsymbol{E} = \begin{bmatrix} \boldsymbol{L}_{6\mathrm{x}6} \\ \boldsymbol{0}_{6\mathrm{x}6} \\ \vdots \\ \boldsymbol{0}_{6\mathrm{x}6} \end{bmatrix}, \ \boldsymbol{S} = \begin{bmatrix} \boldsymbol{I}_{6\mathrm{x}6} & \boldsymbol{0}_{6\mathrm{x}6} & \dots & \boldsymbol{0}_{6\mathrm{x}6} \\ -\boldsymbol{L}_{6\mathrm{x}6} & \boldsymbol{I}_{6\mathrm{x}6} & \dots & \boldsymbol{0}_{6\mathrm{x}6} \\ \boldsymbol{0}_{6\mathrm{x}6} & -\boldsymbol{L}_{6\mathrm{x}6} & \dots & \boldsymbol{0}_{6\mathrm{x}6} \\ \vdots & \vdots & & \vdots \\ \boldsymbol{0}_{6\mathrm{x}6} & \boldsymbol{0}_{6\mathrm{x}6} & \dots & \boldsymbol{I}_{6\mathrm{x}6} \end{bmatrix}, \tag{17}$$

$$\text{with } \boldsymbol{L} := \begin{bmatrix} \boldsymbol{I}_{3\mathrm{x}3} & \boldsymbol{0}_{3\mathrm{x}3} \\ \boldsymbol{0}_{3\mathrm{x}3} & \boldsymbol{0}_{3\mathrm{x}3} \end{bmatrix}.$$

After some algebraic manipulations, (16) becomes

$$J = \big(\boldsymbol{U}(k) + \boldsymbol{H}^{-1}\boldsymbol{\Theta}(k)\big)^T \boldsymbol{H}\,\big(\boldsymbol{U}(k) + \boldsymbol{H}^{-1}\boldsymbol{\Theta}(k)\big) + \mathrm{const}(k), \tag{18}$$

where the Hessian matrix $\boldsymbol{H}(k)$ is defined as

$$\boldsymbol{H}(k) = \boldsymbol{\Upsilon}^T \widetilde{\boldsymbol{Q}} \boldsymbol{\Upsilon} + \lambda_u' \boldsymbol{S}^T \boldsymbol{S} \tag{19}$$

and

$$\boldsymbol{\Theta}(k) = \boldsymbol{\Upsilon}^T \widetilde{\boldsymbol{Q}} \big(\boldsymbol{\Gamma}\boldsymbol{x}(k) - \boldsymbol{Y}^*(k)\big) - \lambda_u' \boldsymbol{S}^T \boldsymbol{E}\boldsymbol{u}_{\mathrm{aug}}(k-1)\,.$$

Note that the constant term in (18) can be omitted, since it is independent of $\boldsymbol{U}(k)$ and therefore has no influence on the optimal solution. The interested reader is referred to [13] for a more detailed derivation.

B. Solution in Terms of the Unconstrained Solution

By relaxing the feasible set in (14) from $\mathbb{U}$ to $\mathbb{R}^n$, and with similar algebraic manipulations as done in [1], the problem (14) can be reformulated as a truncated ILS problem of the form

$$\begin{aligned} \underset{\boldsymbol{U}(k)\in\mathbb{U}}{\text{minimize}} \quad & ||\bar{\boldsymbol{U}}_{\mathrm{unc}}(k) - \boldsymbol{V}\boldsymbol{U}(k)||_2^2 \\ \text{subject to} \quad & ||\Delta\boldsymbol{u}(\ell)||_\infty \leq 1\,,\ \forall\,\ell = k, \dots, k+N-1\,, \end{aligned} \tag{20}$$

with $\bar{\boldsymbol{U}}_{\mathrm{unc}}(k) = \boldsymbol{V}\boldsymbol{U}_{\mathrm{unc}}(k)$ and $\boldsymbol{V}(k)$ being a lower triangular, known as *lattice generator* matrix defined as $\boldsymbol{V}^{-1}\boldsymbol{V}^{-T} = \boldsymbol{H}^{-1}$ [1]. The integer solution $\boldsymbol{U}^*(k)$ of (20) represents the lattice point with the shortest Euclidean distance to the unconstrained solution $\boldsymbol{U}_{\mathrm{unc}}(k)$.

To find the solution in a computationally efficient manner, a branch-and-bound algorithm known as sphere decoding [8] is utilized. According to its principle, a hypersphere (i.e., n-dimensional sphere) of radius ρ centered at the unconstrained solution $\boldsymbol{U}_{\mathrm{unc}}(k)$ is computed. By doing so, only the candidate solutions inside the sphere have to be evaluated. The goal of the optimizer is to tighten the radius ρ incrementally until only the optimal solution remains inside the sphere, whereas all other candidate solutions are excluded since they constitute suboptimal options. The initial radius ρ_{ini} should be chosen such that the resulting sphere is as small as possible for the majority of lattice points to be excluded a priori, but not too small so that at least one lattice point is enclosed. To this end, the initial radius is computed based on an educated guess $\boldsymbol{U}_{\mathrm{ed}}$, as introduced in (40) in [1]

$$\rho_{\mathrm{ini}}(k) = ||\bar{\boldsymbol{U}}_{\mathrm{unc}}(k) - \boldsymbol{V}\boldsymbol{U}_{\mathrm{ed}}(k)||_2\,. \tag{21}$$

Having computed the initial sphere, the optimization process proceeds by evaluating the lattice points inside the sphere to extract the optimal solution. A search tree of n levels is generated, in which the branches resemble the candidate elements of the solution. The sphere decoder traverses this tree in a depth-first search manner with a goal to find the optimal solution as quickly as possible. For more details on the functionality of the sphere decoder the reader is referred to [1] and [13].

C. Modified Sphere Decoding Algorithm

The algorithm proposed in [1] is modified to include the additional "pseudo-inputs" in the implementation. Similar to the code in [1], the algorithm builds the switching sequence $\boldsymbol{U}$ component by component. The recursive algorithm presented in Algorithm 1 is evoked with the initial values of the arguments $\boldsymbol{U}$, d^2, i, $\rho_{\mathrm{ini}}^2(k)$, $\bar{\boldsymbol{U}}_{\mathrm{unc}}(k)$ and $\boldsymbol{u}_{\mathrm{aug}}^*(k-1)$ being the empty set $\emptyset$, 0, 1, the (squared) initial radius, the unconstrained solution, and the previously computed optimal solution, respectively.

Algorithm 1 Modified sphere decoding algorithm

1: **function** MSPHDEC($\boldsymbol{U}, d^2, i, \rho^2, \bar{\boldsymbol{U}}_{\mathrm{unc}}, \boldsymbol{u}^*_{\mathrm{aug}}(k-1)$)
2: $\quad\boldsymbol{u}_{\mathrm{temp}} = \left[\boldsymbol{u}^{*T}_{\mathrm{aug}}(k-1)\ \boldsymbol{U}^T\right]^T$ ▷ append arrays
3: $\quad$**if** $1 \leq \mathrm{mod}(i,6) \leq 3$ **then** ▷ i is switch position
4: $\quad\quad u \leftarrow u \in \mathcal{U}$ ▷ $\mathcal{U} = \{-1, 0, 1\}$
5: $\quad$**else** ▷ i is "pseudo-input"
6: $\quad\quad u \leftarrow \Delta|u_i|$ ▷ $|u_{\mathrm{temp},i+3}| - |u_{\mathrm{temp},i-3}|$
7: $\quad$**end if**
8: $\quad$**for** each u **do**
9: $\quad\quad U_i \leftarrow u$
10: $\quad\quad d'^2 \leftarrow ||\bar{u}_{\mathrm{unc},i} - \boldsymbol{V}_{(i,1:i)}\boldsymbol{U}_{1:i}||_2^2 + d^2$
11: $\quad\quad$**if** $d'^2 \leq \rho^2$ **then**
12: $\quad\quad\quad$**if** $i < 6N$ **then**
13: $\quad\quad\quad\quad$MSphDec($\boldsymbol{U}$,d'^2,i+1,ρ^2,$\bar{\boldsymbol{U}}_{\mathrm{unc}}$,$\boldsymbol{u}^*_{\mathrm{aug}}(k-1)$)
14: $\quad\quad\quad$**else**
15: $\quad\quad\quad\quad$**if** $\boldsymbol{U}$ meets (14c) **then**
16: $\quad\quad\quad\quad\quad\boldsymbol{U}^* \leftarrow \boldsymbol{U},\ \ \rho^2 \leftarrow d'^2$
17: $\quad\quad\quad\quad$**end if**
18: $\quad\quad\quad$**end if**
19: $\quad\quad$**end if**
20: $\quad$**end for**
21: **end function**

According to the pseudocode, the modified sphere decoder first creates a temporary vector $\boldsymbol{u}_{\mathrm{temp}}$ comprised of the previously calculated optimal solution $\boldsymbol{u}^*_{\mathrm{aug}}(k-1)$ and the currently assembled switching sequence $\boldsymbol{U}$. Subsequently, it is checked if the currently evaluated node $i = 1, \ldots, 6N$ is a "pseudo-input". If this is not the case, the current node corresponds to a switch position of one phase leg and all three possibilities in $\mathcal{U}$ will be evaluated. If, on the other hand, the current node corresponds to a "pseudo-input", only one value can be assigned to node i, depending on the difference between the current and previous *phase* switch position, as defined in (8). Note that $\boldsymbol{u}^*_{\mathrm{aug}}(k-1)$ is covering the first six entries of $\boldsymbol{u}_{\mathrm{temp}}$.

The remaining part is identical to the pseudo-code in [1] except that in its refined version the iteration variable u (line 9) is of dimension one in case of a "pseudo-input".

IV. Performance Evaluation

The simulation results presented in this section are obtained based on an MV drive (Fig. 1) consisting of a three-level NPC with constant dc-link voltage $V_{\mathrm{dc}} = 5.2\,\mathrm{kV}$ with a dc-link capacitor value C_{dc} of 7 mF that is connected to a squirrel cage IM with rated values 3.3 kV, 356 A, 2 MVA, 50 Hz nominal frequency, and 0.25 p.u. total leakage inductance. For all cases examined, the sampling interval $T_s = 25\,\mu\mathrm{s}$ was used. All results are shown in the p.u. system.

A. Steady-State Operation

The performance of the proposed direct MPC scheme with the linearized model is examined at steady-state operation for a five-step ($N = 5$) prediction horizon, see Fig. 3. The weighting factors λ_u and λ_{dc} are chosen such that a switching frequency of approximately 200 Hz results and the NP potential remains balanced, as can be seen in Fig. 3(a). The three-phase stator currents $\boldsymbol{i}_{s,abc}$ and their references are shown in Fig. 3(b), whereas the resulting harmonic spectra are presented in Fig. 3(c). As can be seen, the currents accurately track their reference values; the current THD being equal to 5.49% is relatively low given the low switching frequency. Finally, Fig. 3(d) depicts the three-phase switch position for one fundamental period.

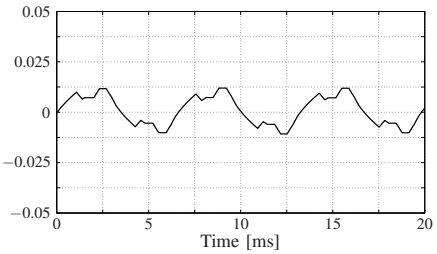
In a second step, the influence of the horizon length on the stator current THD and the rms value of the NP potential deviation is investigated while the switching frequency is kept equal to 200 Hz. As point of reference, a one-step ($N = 1$) horizon MPC, with the nonlinear model solved with exhaustive enumeration, produces stator current THD of 7.58%, whereas the rms of the NP potential deviation is 0.0222 p.u.. Fig. 4 summarizes the evolution of (a) the current THD, and (b) the rms value of $v_{n,\mathrm{err}}$ over the horizon length. One can clearly see significant improvements in both metrics for a horizon $N \geq 2$. For example, when a ten-step horizon is considered, the current THD decreases by about 27% compared to the reference case (it becomes 5.47%), while the rms value of the NP potential is about 60% smaller (0.0080 p.u.).

The presented results indicate that the linearized model can be effectively utilized by the proposed MPC scheme for controlling both stator current as well as the NP potential. Consistent with the findings in [4], [6], [7], a longer prediction horizon improves the performance of the controller in terms of stator current THD and the NP potential remains balanced.

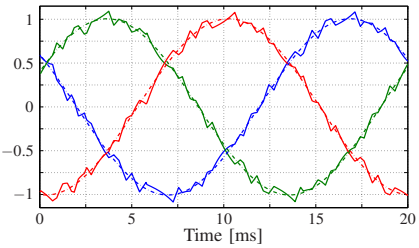
B. Discussion of Suboptimality and Calculation Effort

The derived linearized model of the converter introduces a mismatch between the prediction model used by the control algorithm and the actual, nonlinear model. Consequently, the optimal solution acquired based on the linearized model might differ from that resulting when the exact model is used. To obtain the optimal solution based on the nonlinear model, an exhaustive enumeration algorithm—evaluating all possible switch combinations—is utilized. This issue of suboptimality, i.e., how often the solutions of MPC with the nonlinear and the linearized model differ, is examined in the second column of Table I. For a ten-step horizon ($N = 10$) a suboptimal solution is chosen in less than 2.1% of the cases.

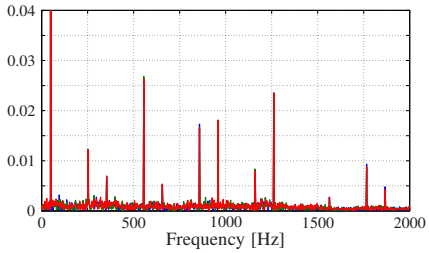
The other columns in Table I present a first rough analysis on the computational complexity of the proposed method in terms of the nodes visited during the search process before concluding in the optimal solution. More specifically, the maximum number of nodes visited by the proposed sphere decoder as well as—for comparison purposes—the exhaustive search algorithm with the nonlinear model are shown. However, the computational effort required to update the state matrices $\boldsymbol{F}(t_0)$ and $\boldsymbol{G}(t_0)$ as well as the subsequent calculation of the Hessian matrix $\boldsymbol{H}(k)$ and the lattice generator matrix $\boldsymbol{V}(k)$, is not analyzed in this paper. As can be understood, the somewhat computationally demanding preprocessing stage makes the real-time implementation of the proposed method more challenging.



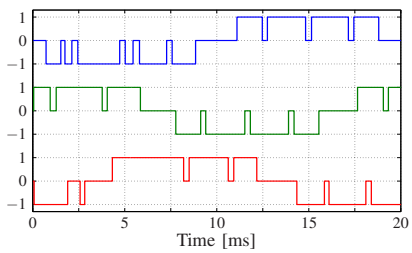
(a): Neutral point potential v_n.



(b): Three-phase stator current $\boldsymbol{i}_{s,abc}$ (solid lines) and their references (dash-dotted lines).

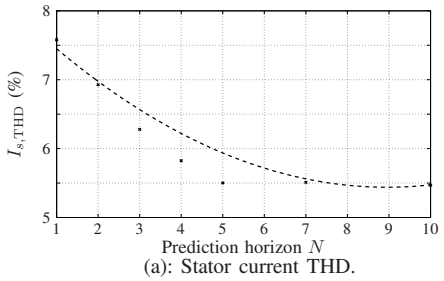


(c): Stator current spectrum. The THD is 5.49%.

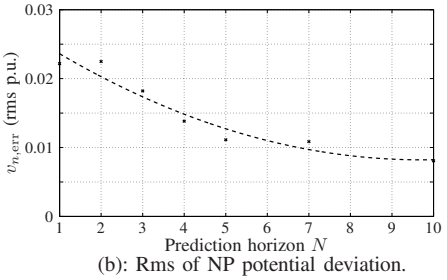


(d): Three-phase switch position $\boldsymbol{u}_{abc}$.

Fig. 3: Simulated waveforms produced by the direct model predictive control problem with current reference tracking and NP potential balancing at steady-state operation, at full speed and rated torque. A five-step horizon ($N = 5$) is used, the sampling interval is $T_s = 25\,\mu$s and the switching frequency is 200 Hz ($\lambda'_u = 0.02$, $\lambda_{\mathrm{dc}} = 15$).



(a): Stator current THD.



(b): Rms of NP potential deviation.

Fig. 4: Stator current THD and rms value of NP potential deviation as a function of the prediction horizon N. The switching frequency is fixed at 200 Hz. The data points relate to individual simulations, which were approximated using a polynomial function of second order.

TABLE I: Second column: The percentage of times the computed solution by the proposed approach $\boldsymbol{U}_{\mathrm{appl}}$ is the (global) optimal solution $\boldsymbol{U}^*$ for different prediction horizons. Other columns: Maximum number of nodes visited by (a) the exhaustive search algorithm and (b) the proposed algorithm.

Prediction Horizon N	$\boldsymbol{U}_{\mathrm{appl}} = \boldsymbol{U}^*$ %	Exhaustive Search	Proposed Approach
1	99.9	39	29
2	99.8	517	56
3	99.1	7,371	119
4	99.0	103,518	254
5	98.9	1,455,000	425
7	98.1	$> 3 \cdot 10^8$	1,084
10	97.9	$> 2 \cdot 10^{12}$	2,489

V. Conclusions

This paper proposes refinements to the sphere decoding algorithm introduced in [1] to include the balancing of the neutral point (NP) potential of a three-level NPC inverter to the control problem. To this end, a linearized model of the drive system is derived and utilized. By doing so, the sphere decoder can be employed and consequently the pronounced computational complexity of the integer problem underlying long-horizon model predictive control (MPC) can be kept at bay. As it was shown, thanks to the long horizon—and despite the simplifications introduced in the system modeling stage—the proposed algorithm outperforms the conventional one-step horizon FCS-MPC that uses the accurate (nonlinear) drive model.

Appendix

The matrices used in (15) are the following, with $\widehat{\boldsymbol{B}} = \boldsymbol{B}\boldsymbol{K}_{\mathrm{aug}}$

$$\boldsymbol{\Gamma} = \begin{bmatrix} \boldsymbol{C}\boldsymbol{A} \\ \boldsymbol{C}\boldsymbol{A}^2 \\ \vdots \\ \boldsymbol{C}\boldsymbol{A}^N \end{bmatrix}, \quad \boldsymbol{\Upsilon} = \begin{bmatrix} \boldsymbol{C}\widehat{\boldsymbol{B}} & \boldsymbol{0} & \dots & \boldsymbol{0} \\ \boldsymbol{C}\boldsymbol{A}\widehat{\boldsymbol{B}} & \boldsymbol{C}\widehat{\boldsymbol{B}} & \dots & \boldsymbol{0} \\ \vdots & \vdots & & \vdots \\ \boldsymbol{C}\boldsymbol{A}^{N-1}\widehat{\boldsymbol{B}} & \boldsymbol{C}\boldsymbol{A}^{N-2}\widehat{\boldsymbol{B}} & \dots & \boldsymbol{C}\widehat{\boldsymbol{B}} \end{bmatrix}.$$

References

[1] T. Geyer and D. E. Quevedo, "Multistep finite control set model predictive control for power electronics," *IEEE Trans. Power Electron.*, vol. 29, no. 12, pp. 6836–6846, Dec. 2014.

[2] J. B. Rawlings and D. Q. Mayne, *Model Predictive Control: Theory and Design*. Madison, WI: Nob Hill, 2009.

[3] P. Cortés, M. P. Kazmierkowski, R. M. Kennel, D. E. Quevedo, and J. Rodríguez, "Predictive control in power electronics and drives," *IEEE Trans. Ind. Electron.*, vol. 55, no. 12, pp. 4312–4324, Dec. 2008.

[4] T. Geyer, "Computationally efficient model predictive direct torque control," *IEEE Trans. Power Electron.*, vol. 26, no. 10, pp. 2804–2816, Oct. 2011.

[5] P. Karamanakos, T. Geyer, N. Oikonomou, F. D. Kieferndorf, and S. Manias, "Direct model predictive control: A review of strategies that achieve long prediction intervals for power electronics," *IEEE Ind. Electron. Mag.*, vol. 8, no. 1, pp. 32–43, Mar. 2014.

[6] T. Geyer and D. E. Quevedo, "Performance of multistep finite control set model predictive control for power electronics," *IEEE Trans. Power Electron.*, vol. 30, no. 3, pp. 1633–1644, Mar. 2015.

[7] T. Geyer, P. Karamanakos, and R. Kennel, "On the benefit of long-horizon direct model predictive control for drives with *LC* filters," in *Proc. IEEE Energy Convers. Congr. Expo.*, Pittsburgh, PA, Sep. 2014, pp. 3520–3527.

[8] U. Fincke and M. Pohst, "Improved methods for calculating vectors of short length in a lattice, including a complexity analysis," *Math. Comput.*, vol. 44, no. 170, pp. 463–471, Apr. 1985.

[9] H. du T. Mouton, "Natural balancing of three-level neutral-point-clamped PWM inverters," *IEEE Trans. Ind. Electron.*, vol. 49, no. 5, pp. 1017–1025, Oct. 2002.

[10] T. Geyer, "Model predictive direct current control: Formulation of the stator current bounds and the concept of the switching horizon," *IEEE Ind. Appl. Mag.*, vol. 18, no. 2, pp. 47–59, Mar./Apr. 2012.

[11] P. Stolze, P. Karamanakos, R. Kennel, S. Manias, and C. Endisch, "Effective variable switching point predictive current control for ac low-voltage drives," *Int. J. of Control*, vol. 88, no. 7, pp. 1366–1378, Jul. 2015.

[12] J. Holtz, "The representation of ac machine dynamics by complex signal flow graphs," *IEEE Trans. Ind. Electron.*, vol. 42, no. 3, pp. 263–271, Jun. 1995.

[13] T. Geyer, *Model predictive control of high power converters and industrial drives*. Hoboken, NJ: Wiley, 2016.