Long-Horizon Direct Model Predictive Control: Modified Sphere Decoding for Transient Operation

Petros Karamanakos, *Member, IEEE*, Tobias Geyer, *Senior Member, IEEE*, and Ricardo P. Aguilera, *Member, IEEE*

Abstract—In this paper we present modifications to the sphere decoder initially introduced in [2] and modified in [3] that significantly reduce the computation times during transients. The relative position of the unconstrained solution of the integer quadratic program underlying model predictive control (MPC) with respect to the convex hull of the (truncated) lattice of integer points is examined. If it is found that the unconstrained solution does not lie within the convex hull—a phenomenon that is observed mostly during transients—then a projection is performed onto the convex hull. By doing so, a new sphere that guarantees feasibility and includes a significant smaller number of candidate solutions is computed. This reduces the computation time by up to three orders of magnitude when solving the optimization problem at hand. Nonetheless, the reduction of the computational burden comes at a cost of (mild) suboptimal results. The effectiveness of the proposed algorithm is tested with a variable speed drive system consisting of a three-level neutral point clamped (NPC) voltage source inverter and a medium-voltage induction machine. Based on the presented results, the sphere decoding algorithm with the proposed refinements maintains the very fast transient responses inherent to direct MPC. Moreover, it is observed that the occasional implementation of suboptimal solutions does not lead to a deterioration of the system performance.

Index Terms—AC drive, model predictive control (MPC), integer least-squares (ILS) problem, integer programming, medium-voltage (MV) drive, three-level neutral-point inverter, sphere decoding.

I. INTRODUCTION

OVER the past decade model predictive control (MPC) [4] has paved its way in becoming one of the most attractive control alternatives in power electronics [5]–[7]. MPC is formulated as a constrained optimization problem, meaning that it can operate the system at its physical limits, and—as a result—achieve the best possible performance. Moreover, it can easily handle systems with nonlinear dynamics, and/or multiple inputs and outputs since it is formulated in the (discrete) time domain. These features of MPC, combined with the vast computational power readily available, justify its widespread acceptance from the power electronic community. Prominent examples include, among others, current

control of inverters [8], [9], rectifiers [10] and cycloconverters [11], power control of grid-connected converters [12], and torque [13]–[15], current [16], [17], and flux [18], [19] control of electrical drives.

The most utilized MPC strategy in power electronics is the so-called finite control set MPC (FCS-MPC) [5]. The output reference tracking and the modulation problem are formulated in one computational stage, i.e., akin to a direct control strategy. Although this approach seems to be straightforward and intuitive, from a mathematical optimization perspective, it is computationally very challenging. The underlying optimization problem is a (mixed) integer program, which is known to be NP-hard, i.e., it can become computationally intractable as the number of decision variables (i.e., control inputs) or the length of the prediction horizon increase. Given that the latter has been documented to be necessary for an improved system performance [20]–[23], the commonly used brute-force method of exhaustive enumeration—according to which all candidate solutions are enumerated to determine the solution—cannot be considered as a realistic option for horizons longer than one step.

To allow for longer horizons, several techniques have been proposed that either utilize nontrivial prediction horizons [24] by implementing concepts such as move blocking [25] (e.g., [26]) and extrapolation [7] (e.g., [27]), or branch-and-bound methods [28] based on smart heuristics [20]. With regards to the latter, a dedicated branch-and-bound algorithm, developed in the field of communications and referred to as sphere decoder [29], [30], has been recently introduced in the field of power electronics [2], [3], [21]. Although it has been shown to be reasonably effective, since it manages to significantly reduce the computational complexity of the problem at steady-state operation as well as to tellingly deal with state/output constraints [31], there exist challenges that have not yet been fully addressed.

Among the most prominent issues yet to be resolved is the system operation during transient phenomena. As shown in this paper, the initial sphere used in the optimization stage may become large during transients, making the bounding in the sphere decoding less effective. To avoid the associated large computational burden, this paper proposes refinements in the sphere decoding algorithm. These refinements guarantee that the center of the computed sphere is located within the convex hull of the feasible set of the integer program underlying MPC. This is achieved by projecting the unconstrained solution (i.e., the center of the to-be-computed sphere) onto the aforementioned convex hull. Moreover, the geometry of the

optimization problem is exploited to compute the projected point in an efficient manner. As a result, the computational burden during transients is similar to that during steady-state operation. However, a small degree of suboptimality may result.

The issue of large spheres during transients was first examined in [32] for a three-phase three-level H-bridge (HB) inverter with an RL load. Therein, the projection problem was formulated as a quadratic program (QP) and the effectiveness of the approach was tested with the aforementioned time-invariant plant, i.e., the HB inverter, and a five-step horizon. Motivated by that work, the projection problem is formulated as a *box-constrained* QP in this paper. By doing so, reliable solvers can be employed that efficiently compute the solution. Moreover, the root cause of the above-mentioned suboptimality is identified and discussed in detail, along with a study on the optimality of the proposed method. Finally, a thorough computational analysis of the optimization procedure is provided.

To highlight the effectiveness of the modified sphere decoder, a variable speed drive system, consisting of a three-level neutral point clamped (NPC) voltage source inverter driving a medium-voltage (MV) induction machine (IM), is chosen as a case study. As it is shown, the computational complexity of the algorithm—in terms of real-time floating point operations (flops)—can be reduced by more than 97.5% when long horizons, such as ten steps, and a wide range of switching frequencies are considered.

This paper is structured as follows. The model of the drive system and the MPC problem are presented in Section II. The formulation of the optimization problem as an integer least-squares (ILS) one is outlined in Section III. The derivation of the sphere decoding algorithm applicable to transient operation follows in Section IV, whereas its computational complexity is analyzed in Section V. In Section VI, the performance of the MPC strategy is demonstrated for the chosen case study. Conclusions are drawn in Section VII.

II. OPTIMAL CONTROL PROBLEM

The problem examined relates to the control of the stator current of an IM when driven by a three-level NPC voltage source inverter with a constant dc-link voltage V_{dc} and a fixed neutral point potential (Fig. 1).

In the sequel, the mathematical model of the examined system and the formulation of the optimization problem are presented. Both these tasks are performed in the stationary orthogonal $\alpha\beta$ system. 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., $\boldsymbol{\xi}_{\alpha\beta} = \boldsymbol{K}\boldsymbol{\xi}_{abc}$, with

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

A. Control Model

The output potential of each phase of the three-level NPC inverter can obtain three discrete voltage levels $-\frac{V_{\mathrm{dc}}}{2}$, 0,
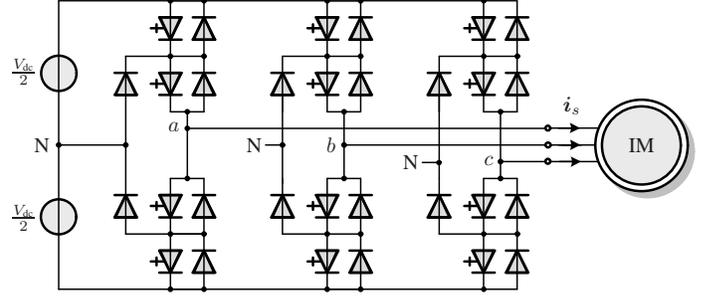
Fig. 1: Three-level three-phase neutral point clamped (NPC) voltage source inverter driving an induction motor (IM) with a fixed neutral point potential.

$\frac{V_{\mathrm{dc}}}{2}$, depending on the position of the power switches in the respective phase [33]. As a result, the inverter output voltage—equal to the voltage applied to the machine terminals $\boldsymbol{v}_{s,\alpha\beta}$—is[1]

$$\boldsymbol{v}_{\alpha\beta} = \frac{V_{\mathrm{dc}}}{2}\,\boldsymbol{u}_{\alpha\beta} = \frac{V_{\mathrm{dc}}}{2}\boldsymbol{K}\boldsymbol{u}\,, \tag{1}$$

where $\boldsymbol{u} = [u_a\ u_b\ u_c]^T \in \boldsymbol{\mathcal{U}} = \mathcal{U}\times\mathcal{U}\times\mathcal{U} = \mathcal{U}^3$ is the three-phase switch position, with the integer variable $u_x \in \mathcal{U} = \{-1,0,1\}$ denoting the switch position in phase $x \in \{a,b,c\}$.

With regards to the squirrel-cage IM, the stator current $\boldsymbol{i}_{s,\alpha\beta}$ and the rotor flux $\boldsymbol{\psi}_{r,\alpha\beta}$ in the $\alpha\beta$ plane as well as the angular speed of the rotor ω_r are adopted as state variables to fully describe its dynamics. Given the model parameters, i.e., the stator R_s and rotor R_r resistances, the stator X_{ls}, rotor X_{lr} and mutual X_m reactances, the moment of inertia H, and the mechanical load torque T_ℓ, the differential equations that govern the machine are[2] [34]

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

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

$$\frac{\mathrm{d}\omega_r}{\mathrm{d}t} = \frac{1}{H}(T_e - T_\ell)\,. \tag{2c}$$

In (2), the stator time constant is $\tau_s = X_r\Phi/(R_sX_r^2 + R_rX_m^2)$ and the rotor time constant $\tau_r = X_r/R_r$. Moreover, $\Phi = X_sX_r - X_m^2$ is a constant with $X_s = X_{ls} + X_m$ and $X_r = X_{lr} + X_m$, and T_e is the electromagnetic torque. Finally, $\boldsymbol{I}$ is the identity matrix of appropriate dimension (here two-dimensional).

For the MPC algorithm developed in Section II-B the internal model of the drive system is required so that its future behavior can be predicted. To this end, the stator current and the rotor flux in the $\alpha\beta$ plane constitute the stator vector $\boldsymbol{x} = [i_{s\alpha}\ i_{s\beta}\ \psi_{r\alpha}\ \psi_{r\beta}]^T$. Note that due to the slower dynamics the mechanical speed is assumed to be constant within the prediction horizon. Hence, the rotor angular speed

---

[1]Throughout the paper, vectors in the $\alpha\beta$ plane are denoted with the corresponding subscript. For vectors in the abc plane the subscript is omitted.

[2]To simplify the notation in (2) the subscript $\alpha\beta$ is dropped from the vectors $\boldsymbol{i}_s$, $\boldsymbol{\psi}_r$ and $\boldsymbol{v}_s$.
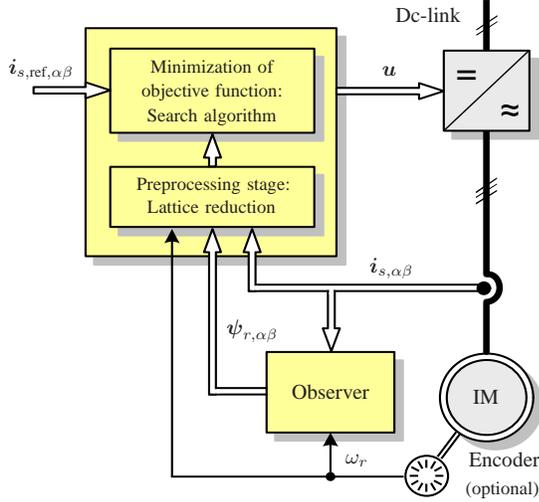
Fig. 2: Model predictive current control with reference tracking for the three-phase three-level NPC inverter with an induction machine.

ω_r is not a state variable, but a time-varying parameter instead. The input to the system is considered to be the three-phase switch position $\boldsymbol{u}$, whereas the stator current is the output variable, i.e., $\boldsymbol{y} = \boldsymbol{i}_{s,\alpha\beta}$. Therefore, by taking into account (1) and (2), the continuous-time state-space representation of the drive can be written as

$$\frac{\mathrm{d}\boldsymbol{x}(t)}{\mathrm{d}t} = \boldsymbol{D}\boldsymbol{x}(t) + \boldsymbol{E}\boldsymbol{K}\boldsymbol{u}(t) \tag{3a}$$

$$\boldsymbol{y}(t) = \boldsymbol{F}\boldsymbol{x}(t), \tag{3b}$$

where the dynamics $\boldsymbol{D}$, input $\boldsymbol{E}$ and output $\boldsymbol{F}$ matrices, are given in Appendix A.

Using exact Euler discretization, the discrete-time state-space model of the drive is

$$\boldsymbol{x}(k+1) = \boldsymbol{A}\boldsymbol{x}(k) + \boldsymbol{B}\boldsymbol{K}\boldsymbol{u}(k) \tag{4a}$$

$$\boldsymbol{y}(k) = \boldsymbol{C}\boldsymbol{x}(k), \tag{4b}$$

with $\boldsymbol{A} = \mathrm{e}^{\boldsymbol{D}T_s}$, $\boldsymbol{B} = -\boldsymbol{D}^{-1}(\boldsymbol{I} - \boldsymbol{A})\boldsymbol{E}$ and $\boldsymbol{C} = \boldsymbol{F}$. Moreover, $\boldsymbol{I}$ here is the 4×4 identity matrix, e the matrix exponential, T_s the sampling interval, and $k \in \mathbb{N}$.

B. Direct MPC With Current Reference Tracking

The discussed control algorithm aims to regulate the stator current $\boldsymbol{i}_{s,\alpha\beta}$ along its reference $\boldsymbol{i}_{s,\mathrm{ref},\alpha\beta}$, while operating the drive at low switching frequency. The latter objective is fairly 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 both objectives by computing and applying the control signals (i.e., the switch positions) in one stage, i.e., a modulation stage is bypassed.

By introducing the current (i.e., output) tracking error term $\boldsymbol{i}_{s,\mathrm{err},\alpha\beta}(k) = \boldsymbol{i}_{s,\mathrm{ref},\alpha\beta}(k) - \boldsymbol{i}_{s,\alpha\beta}(k)$, and the switching (i.e., control) effort term $\Delta\boldsymbol{u}(k) = \boldsymbol{u}(k) - \boldsymbol{u}(k-1)$, the aforementioned control objectives are mapped into a scalar, i.e.,

$$J(k) = \sum_{\ell=k}^{k+N-1} ||\boldsymbol{i}_{s,\mathrm{err},\alpha\beta}(\ell+1|k)||_2^2 + \lambda_u ||\Delta\boldsymbol{u}(\ell|k)||_2^2. \tag{5}$$

Based on the state of the current error and switching effort at step k, function (5) penalizes their evolution over the finite prediction horizon of length N time steps. Note that the weighting factor $\lambda_u > 0$ is introduced to adjust the trade-off between the two terms, i.e., the trade-off between the current tracking ability of the controller and the switching frequency.

To find the optimal sequence of control actions $\boldsymbol{U}^*(k) = [\boldsymbol{u}^{*T}(k)\ \boldsymbol{u}^{*T}(k+1)\ \dots\ \boldsymbol{u}^{*T}(k+N-1)]^T$ that results in the most desirable system behavior, i.e., the one that minimizes (5), problem

$$\begin{aligned} &\underset{\boldsymbol{U}(k)}{\text{minimize}} && J(k) \\ &\text{subject to} && \boldsymbol{x}(\ell+1) = \boldsymbol{A}\boldsymbol{x}(\ell) + \boldsymbol{B}\boldsymbol{u}(\ell) \\ & && \boldsymbol{y}(\ell+1) = \boldsymbol{C}\boldsymbol{x}(\ell+1), \forall\ell = k, \dots, k+N-1 \\ & && \boldsymbol{U}(k) \in \mathbb{U} \end{aligned}$$
$$\tag{6}$$

is solved in real time. In (6), $\boldsymbol{U}(k) = [\boldsymbol{u}^T(k)\ \boldsymbol{u}^T(k+1)\ \dots\ \boldsymbol{u}^T(k+N-1)]^T$ is the optimization variable and $\mathbb{U} = \mathcal{U}^N \subset \mathbb{Z}^n$, with $n = 3N$, is the feasible set defined as the N-times Cartesian product of the set $\mathcal{U}$.

III. EQUIVALENT INTEGER LEAST-SQUARES PROBLEM

By relaxing the feasible set in (6), from $\mathbb{U}$ to $\mathbb{R}^n$, one can easily compute the so-called unconstrained solution $\bar{\boldsymbol{U}}_{\mathrm{unc}}(k) = \boldsymbol{H}\boldsymbol{U}_{\mathrm{unc}}(k) \in \mathbb{R}^n$, with $\boldsymbol{U}_{\mathrm{unc}} \in \mathbb{R}^n$ [2]. Based on $\bar{\boldsymbol{U}}_{\mathrm{unc}}$—and after some algebraic manipulations shown in Appendix B—problem (6) becomes

$$\begin{aligned} &\underset{\boldsymbol{U}(k)}{\text{minimize}} && ||\bar{\boldsymbol{U}}_{\mathrm{unc}}(k) - \boldsymbol{H}\boldsymbol{U}(k)||_2^2 \\ &\text{subject to} && \boldsymbol{U}(k) \in \mathbb{U}, \end{aligned} \tag{7}$$

which is an ILS problem [2]. The nonsingular, upper triangular matrix $\boldsymbol{H} \in \mathbb{R}^{n\times n}$ in (7), is known as the lattice generator matrix that generates the discrete space wherein the solution lies.

ILS problems are known to be NP-hard [35], meaning that if $\boldsymbol{H}$ is ill-conditioned, then solving (7) may become computationally intractable. To avoid this, algorithms can be used that reshape the search space (i.e., they modify the entries of $\boldsymbol{H}$) with the eventual goal of having a lattice as close to orthogonal as possible, while ensuring that the length of the basis vectors that generate it (i.e., the Euclidean norm of the columns of the lattice generator matrix) is relatively small. To this end, the Lenstra-Lenstra-Lovász (LLL) lattice basis reduction algorithm [36] is employed to generate the reduced lattice generator matrix $\widetilde{\boldsymbol{H}}$. The resulting equivalent ILS problem is of the form [3]

$$\begin{aligned} &\underset{\boldsymbol{U}(k)}{\text{minimize}} && ||\widetilde{\boldsymbol{U}}_{\mathrm{unc}}(k) - \widetilde{\boldsymbol{H}}\widetilde{\boldsymbol{U}}(k)||_2^2, \\ &\text{subject to} && \boldsymbol{U}(k) \in \mathbb{U}, \end{aligned} \tag{8}$$

with $\widetilde{\boldsymbol{U}}_{\mathrm{unc}}(k) = \widetilde{\boldsymbol{H}}\ \boldsymbol{M}^{-1}\boldsymbol{U}_{\mathrm{unc}}(k)$, $\widetilde{\boldsymbol{U}}(k) = \boldsymbol{M}^{-1}\boldsymbol{U}(k)$ and $\widetilde{\boldsymbol{H}} = \boldsymbol{V}^T\boldsymbol{H}\boldsymbol{M}$. Finally, matrix $\boldsymbol{V} \in \mathbb{R}^{n\times n}$ is orthogonal and matrix $\boldsymbol{M} \in \mathbb{Z}^{n\times n}$ unimodular (i.e., $\det\boldsymbol{M} = \pm 1$).

Interpreting (8), one has to find the (integer) n-dimensional lattice point that is closest (in Euclidean sense) to the (real-valued) unconstrained solution $\widetilde{\boldsymbol{U}}_{\mathrm{unc}}(k)$. Equivalently, one has

to find the smallest hypersphere (n-dimensional sphere) of radius ρ centered at $\widehat{U}_{\text{unc}}(k)$ that includes at most one lattice point. To this end, the branch-and-bound method known as sphere decoder [29] can be utilized. With sphere decoding the search space is limited to those lattice points (i.e., nodes) included in the computed hypersphere, meaning that only these points are considered as candidate solutions, and thus evaluated, while optimality is still guaranteed. As a result, this search algorithm proves to be highly computationally efficient, especially compared with the brute-force approach of the exhaustive enumeration, see [21] and [3].

As can be understood, the initial radius ρ of the computed hypersphere determines the efficacy of the sphere decoder. The *initial* sphere should be small enough to include as few nodes as possible, whereas it should be nonempty in order to avoid feasibility issues. According to [37], the initial radius can be set equal to the minimum of two options, i.e.,

$$\rho(k) = \min\{\rho_a(k), \rho_b(k)\}, \tag{9}$$

with the options being

$$\rho_a(k) = || \ \widetilde{U}_{\text{unc}}(k) - \widetilde{H}\widetilde{U}_{\text{bab}}(k)||_2 , \tag{10a}$$
$$\rho_b(k) = || \ \widetilde{U}_{\text{unc}}(k) - \widetilde{H}\widetilde{U}_{\text{ed}}(k)||_2 . \tag{10b}$$

The radii in (10) are computed based on different estimated solutions (i.e., initial guesses) $\widetilde{U}_{\text{est}} = M^{-1}U_{\text{est}}$. Radius ρ_a is computed based on the so-called Babai estimate [38], i.e., $U_{\text{est}} = U_{\text{bab}}$, which results by rounding the unconstrained solution to the closest integer vector, i.e., $U_{\text{bab}}(k) = \lfloor U_{\text{unc}}(k) \rceil$. The second option (see (10b)) is computed based on an educational guess ($U_{\text{est}} = U_{\text{ed}}$), which results by taking into account the previously computed solution $U^*(k-1)$ shifted by one time step and with a repetition of the last switch position (see (40) in [2]). Note that $\widetilde{U}_{\text{bab}}(k) = M^{-1}U_{\text{bab}}(k)$ and $\widetilde{U}_{\text{ed}}(k) = M^{-1}U_{\text{ed}}(k)$.

With an initial tight sphere of radius (9) the vast majority of the lattice points are a priori excluded from being considered candidate solutions. The task of the sphere decoder is to visit the remaining nodes enclosed in the sphere in order to conclude to the optimal solution. To do so, and as can be seen in Algorithm 1 (lines 15–28), the optimizer traverses a search tree of n levels by visiting one-by-one the m-dimensional nodes, $m = 1, \ldots, n$, in a depth-first search manner. Starting form the root (top), n-dimensional node, the algorithm goes through the branches—that constitute the candidate elements of the solution—with a goal to reach the bottom level of the tree (i.e., the leaf nodes that are one-dimensional) as quickly as possible. As long as the assembled candidate solution results in a smaller intermediate sphere than the initial one, the algorithm keeps visiting nodes at the lower levels of tree; if not, it backtracks to visit higher-level nodes and find a new path to the bottom level. When a leaf node is reached, a complete candidate solution is assembled, and a new, tighter sphere is computed. The optimizer continues searching for different sequences that may result in a tighter sphere until it gets a "certificate" that the optimal solution has been found.

IV. Sphere Decoder for Transient Operation

As mentioned before—and as shown in [21] and [3]—the sphere decoder is significantly more computationally efficient than the exhaustive enumeration when the steady-state operation of the system is examined. Even though the sphere decoding principle is also applicable to transient operation, the computationally complexity of the sphere decoder itself increases (that of the exhaustive enumeration remains the same since all candidate solutions need to be *always* evaluated). The reason for that is that the unconstrained solution under transient operating conditions can be far from the (truncated) lattice, implying that the initial radius (9) can be large with a considerable number of nodes included in the resulting sphere. More specifically, radius ρ_b (see (10)) is clearly not a good guess since the previously applied solution (i.e., the one applied before the transient occurs) is expected to be very different compared to the one computed when the transient phenomenon begins, whereas ρ_a—although a better choice—appears to be a poor initial radius as well.

To overcome this, and to keep the complexity of the sphere decoding algorithm low, one could project the unconstrained solution $\widehat{U}_{\text{unc}}(k)$ onto the convex hull of the lattice in the space generated by $\widetilde{H}$. According to [39], the convex hull is defined as

$$\mathbf{conv}\,\widetilde{\mathbb{U}} = \{\theta_1(\widetilde{H}\widetilde{U}_1) + \cdots + \theta_j(\widetilde{H}\widetilde{U}_j) \mid \widetilde{U}_i = M^{-1}U_i,$$
$$U_i \in \mathbb{U}, \ \theta_i \geq 0, \ i = 1, \ldots, j, \ j = 3^n,$$
$$\theta_1 + \cdots + \theta_j = 1\}. \tag{11}$$

To this end, the following QP needs to be solved in real time

$$\underset{\widetilde{U}(k)}{\text{minimize}} \quad || \ \widetilde{U}_{\text{unc}}(k) - \widetilde{H}\widetilde{U}(k)||_2^2$$
$$\text{subject to} \quad \widetilde{U}(k) \in \mathbf{conv}\,\widetilde{\mathbb{U}} , \tag{12}$$

the optimal value of which is the (Euclidean) distance $\mathbf{dist}(\widetilde{U}_{\text{unc}}(k), \mathbf{conv}\,\widetilde{\mathbb{U}})$, and the optimal point is the projection of $\widetilde{U}_{\text{unc}}(k)$ on $\mathbf{conv}\,\widetilde{\mathbb{U}}$ [39].

Having found the (real-valued) projected point $\widehat{U}_{\text{rlx}}(k) \in \mathbf{conv}\,\widetilde{\mathbb{U}}$ a new sphere can be used by the sphere decoder. This smaller sphere is centered at $\widehat{U}_{\text{rlx}}$ and has radius $\hat{\rho}$ that is given by (9), with the difference that the first option results by rounding $U_{\text{rlx}}(k) = M\,\widetilde{H}^{-1}\widetilde{U}_{\text{rlx}}(k)$ to the nearest integer point, i.e., similarly to (10a) with the difference that $\lfloor U_{\text{rlx}}(k) \rceil$ is the estimated solution.

The downside of this approach is that the shape of the convex hull $\mathbf{conv}\,\widetilde{\mathbb{U}}$ is nontrivial (it is an n-dimensional polyhedron) that depends on $\widetilde{H}$. Since $\widetilde{H}$ depends on the rotor speed (see Appendix B), the convex hull needs to be recomputed when this parameter changes. This increases the computational burden of the approach rendering its benefits less tangible.

A workaround is to solve the equivalent problem of (12) in the original space, as motivated by the following remark.

Remark 1: The objective function $|| \ \widetilde{U}_{\text{unc}}(k) - \widetilde{H}\widetilde{U}(k)||_2^2$ can be written as $(U_{\text{unc}}(k) - U(k))^T Q(U_{\text{unc}}(k) - U(k))$ with $Q = H^T H$.

Proof: According to the definitions in Section III, it is true that

$$\widetilde{U}_{\text{unc}}(k) = \widetilde{H} \, M^{-1} U_{\text{unc}}(k), \qquad (13)$$

$$\widetilde{U}(k) = M^{-1} U(k), \qquad (14)$$

$$\widetilde{H} = V^T H M. \qquad (15)$$

Then for the function $J(k) = || \widetilde{U}_{\text{unc}}(k) - \widetilde{H}\widetilde{U}(k)||_2^2$, it holds that

$$
\begin{aligned}
J(k) &= || \widetilde{U}_{\text{unc}}(k) - \widetilde{H}\widetilde{U}(k)||_2^2 \\
&= ||V^T H M M^{-1} U_{\text{unc}}(k) - V^T H M M^{-1} U(k)||_2^2 \\
&\qquad\qquad\qquad\qquad\qquad\qquad \text{(from (13)–(15))} \\
&= ||V^T H U_{\text{unc}}(k) - V^T H U(k)||_2^2 \quad (\det M = \pm 1) \\
&= ||H U_{\text{unc}}(k) - H U(k)||_2^2 \qquad\qquad (V^T = V^{-1}) \\
&= (U_{\text{unc}}(k) - U(k))^T H^T H (U_{\text{unc}}(k) - U(k)) \\
&= (U_{\text{unc}}(k) - U(k))^T Q (U_{\text{unc}}(k) - U(k)) \\
&\qquad\qquad\qquad\qquad\qquad\qquad\qquad \text{(from (20))}
\end{aligned}
$$

∎

The convex hull in the original space **conv** $\mathbb{U}$ is trivial since it is a hypercube (n-dimensional cube), centered at the origin and of side length 2. Therefore the projection of $U_{\text{unc}}(k)$ on **conv** $\mathbb{U}$, i.e., $U_{\text{rlx}}(k) \in$ **conv** $\mathbb{U}$ can be found by solving the following *box-constrained* QP

$$
\begin{aligned}
\underset{U(k)}{\text{minimize}} \quad & (U_{\text{unc}}(k) - U(k))^T Q (U_{\text{unc}}(k) - U(k)) \\
\text{subject to} \quad & \begin{bmatrix} I & 0 \\ 0 & -I \end{bmatrix} \begin{bmatrix} U(k) \\ U(k) \end{bmatrix} \preceq \begin{bmatrix} 1 \\ -1 \end{bmatrix},
\end{aligned}
$$

$$(16)$$

where I and 0 are the n-dimensional identity and zero matrices, respectively, 1 is an n-dimensional vector with all components one, and $\preceq$ denotes the componentwise inequality. Because **conv** $\mathbb{U}$ is nonempty, problem (16) is by definition feasible. Furthermore, it is convex with n optimization variables and $2n$ inequality constraints, thus relatively small and easy to solve [39]. More specifically, a plethora of effective solvers is readily available, and any of iterative projection algorithms, such as [40], [41], gradient projection methods, e.g., [42]–[44], or derivatives [45]–[47] can be employed with negligible differences in computational time, given the small size of the problem at hand.

With the projection point U_{rlx} obtained, the next step is to map it into the $\widetilde{H}$-generated space, i.e., $\widetilde{U}_{\text{rlx}} = \widetilde{H} \, M^{-1} U_{\text{rlx}}$. Subsequently, the radius $\hat{\rho}$ of the new sphere, centered at $\widetilde{U}_{\text{rlx}}$, is computed as explained above, and the optimization process begins. The pseudocode of the proposed algorithm is presented in Algorithm 1. The initial values of the arguments are computed as explained in Section III.

Projecting the infeasible unconstrained solution on the convex hull (i.e., the feasible set) greatly reduces the computational complexity of the sphere decoder, as shown in Section VI. However, it does not guarantee optimality. This means that the lattice point that is actually closest to $\hat{U}_{\text{unc}}(k)$ may not be the one closest to $\widetilde{U}_{\text{rlx}}(k)$. This is explained with two illustrative examples in Figs. 3 and 4, where the

---

Algorithm 1 Long-Horizon Direct MPC

1: **function** $U^* = \text{DMPCC}(U_{\text{unc}}, U_{\text{ed}}, U_{\text{bab}})$
2:      **if** $U_{\text{unc}} \notin \textbf{conv}\,\mathbb{U}$ **then**
3:          $U_{\text{rlx}} \leftarrow \arg\min_{U \in \textbf{conv}\,\mathbb{U}} ||U_{\text{unc}} - U||_Q$
4:          $U_{\text{bab}} \leftarrow \lfloor U_{\text{rlx}} \rceil$
5:          $U_{\text{unc}} \leftarrow U_{\text{rlx}}$
6:      **end if**
7:      $\widetilde{U}_{\text{unc}} \leftarrow \widetilde{H} \, M^{-1} U_{\text{unc}}$
8:      $\widetilde{U}_{\text{bab}} \leftarrow M^{-1} U_{\text{bab}}$
9:      $\widetilde{U}_{\text{ed}} \leftarrow M^{-1} U_{\text{ed}}.$
10:     $\rho \leftarrow \min\{\rho_a, \rho_b\}$ $\triangleright$ see (10)
11:     $\text{SPHDEC}([\,], \widetilde{U}_{\text{unc}}, 0, n, \rho^2)$
12:     $U^* \leftarrow M \, \widetilde{U}^*$
13: **end function**
14:
15: **function** $\widetilde{U}^* = \text{SPHDEC}(\widetilde{U}, \widetilde{U}_{\text{unc}}, d^2, i, \rho^2)$
16:      **for** each $\tilde{u} \in \mathcal{U}$ **do**
17:          $\widetilde{U}_i \leftarrow \tilde{u}$
18:          $d'^2 \leftarrow || \widetilde{U}_{\text{unc}_i} - \widetilde{H}_{(i,i:n)} \widetilde{U}_{i:n}||_2^2 + d^2$
19:          **if** $d'^2 \leq \rho^2$ **then**
20:              **if** $i > 1$ **then**
21:                  $\text{SPHDEC}(\widetilde{U}, \widetilde{U}_{\text{unc}}, d'^2, i-1, \rho^2)$
22:              **else**
23:                  $\widetilde{U}^* \leftarrow \widetilde{U}$
24:                  $\rho^2 \leftarrow d'^2$
25:              **end if**
26:          **end if**
27:      **end for**
28: **end function**

---

two main reasons for suboptimality are presented, namely, a significantly skewed lattice, and a lattice with significantly uneven sides. As can be seen in Fig. 3, when the angle between the vectors that form the basis of $\widetilde{H}$ is far from orthogonal, then the proposed approach is prone to suboptimality. The same phenomenon is observed when the length of the vectors of the basis of $\widetilde{H}$ is greatly unequal, as depicted in Fig. 4. Moreover, combinations of both sources of subotimality can further aggravate the phenomenon under discussion. It is worth mentioning, nonetheless, that thanks to the well-conditioned $\widetilde{H}$ (i.e., the lattice is almost orthogonal and all sides are of comparable lengths) the probability for the optimal solution to be inside the sphere centered at $\widetilde{U}_{\text{rlx}}$ is very high, see Section VI. In addition, given that the projection algorithm is mostly active only during transients, the suboptimality does not affect the steady-state performance of the system. Consequently, the benefits of long-horizon direct MPC [3], [21] are still achieved.

V. COMPUTATIONAL COMPLEXITY

Analyzing the computational complexity of Algorithm 1, the main effort is put into solving (16) (see line 3) and the sphere decoder (lines 15–29). As mentioned in Section IV, problem (16) is a box-constrained QP which means that it can be solved in polynomial time. For example, in [40], [45], it is shown that the proposed schemes exhibit a global

(a)

(b)
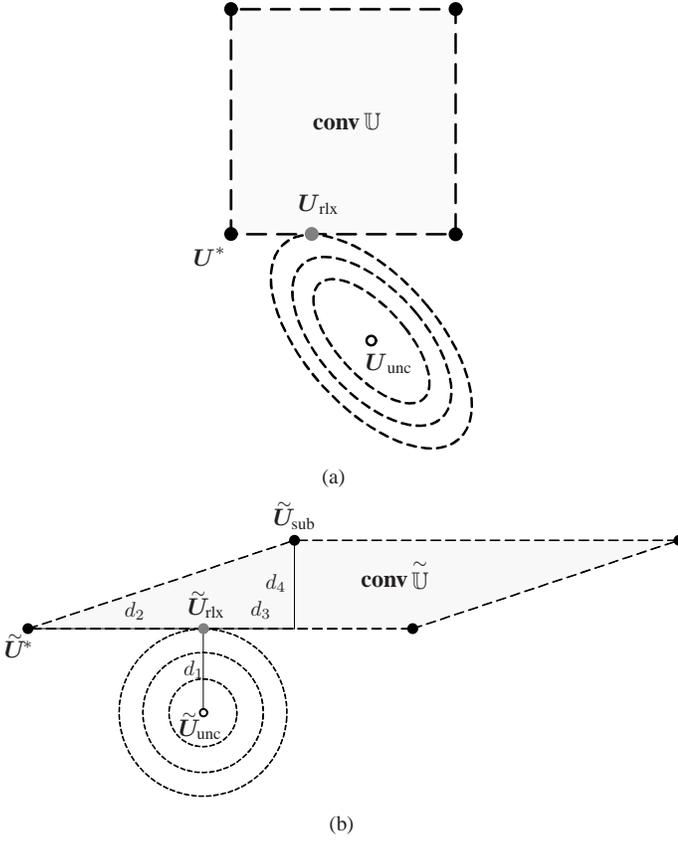
Fig. 3: Example of the projection principle and the underlying suboptimality due to a skewed lattice. Convex hull (shown shaded) of four points (shown as black solid circles) in the two-dimensional space. (a) Original space: The level sets of the objective function of problem (16) are shown as dashed ellipses. The projection of $\boldsymbol{U}_{\mathrm{unc}}$ (shown as circle) on **conv** $\mathbb{U}$ is the point $\boldsymbol{U}_{\mathrm{rlx}}$ (shown as gray solid circle), i.e., the point where the level sets "touch" the convex hull. (b) $\widetilde{\boldsymbol{H}}$-generated space: The level sets of the function of problem (12) are shown as dashed circles. The projection of $\widehat{\boldsymbol{U}}_{\mathrm{unc}}$ on **conv** $\widetilde{\mathbb{U}}$ is the point $\widehat{\boldsymbol{U}}_{\mathrm{rlx}}$. Consider the following case with lengths $d_1 = 0.2$, $d_2 = 0.6$, $d_3 = 0.3$ and $d_4 = 0.4$. The lattice point closest to $\widetilde{\boldsymbol{U}}_{\mathrm{unc}}$ (i.e., the optimal solution) is the one labeled as $\widetilde{\boldsymbol{U}}^*$, since $|| \widetilde{\boldsymbol{U}}_{\mathrm{unc}} - \widetilde{\boldsymbol{U}}^* || = \sqrt{0.4} < || \widehat{\boldsymbol{U}}_{\mathrm{unc}} - \widetilde{\boldsymbol{U}}_{\mathrm{sub}} || = \sqrt{0.45}$. On the other hand, the point closest to the projected point $\widetilde{\boldsymbol{U}}_{\mathrm{rlx}}$ is the point labeled as $\widehat{\boldsymbol{U}}_{\mathrm{sub}}$, since $|| \widetilde{\boldsymbol{U}}_{\mathrm{rlx}} - \widetilde{\boldsymbol{U}}^* || = 0.6 > || \widehat{\boldsymbol{U}}_{\mathrm{rlx}} - \widetilde{\boldsymbol{U}}_{\mathrm{sub}} || = 0.5$. Thus, $\widehat{\boldsymbol{U}}_{\mathrm{sub}}$ is clearly a suboptimal option.

convergence rate $O(1/\kappa^2)$, where κ is the number of iterations. Therefore, (16) can be easily solved within 12–18 iterations on average, depending on the size of the problem which varies linearly with the length of the prediction horizon.

With regards to the computational complexity of the sphere decoding algorithm, this is analyzed based on the flops performed in real time. As can be seen in Algorithm 1, when an m-dimensional node, with $m = 1, \ldots, n$, is visited, $n - m + 1$ additions[3] are required for the computation and update of the intermediate (squared) radius d ($n - m$ for the computation of the radius at level m, i.e., $\widetilde{\boldsymbol{H}}_{(m,m:n)} \, \widehat{\boldsymbol{U}}_{m:n}$, and one for the update $|| * ||_2^2 + d^2$), see line 18. Moreover one subtraction is required. Regarding the multiplications, only one is required per node (for squaring the Euclidean norm $|| * ||_2^2$ regardless of its dimension; since $\boldsymbol{U} \in \mathbb{U}$ the result of each one of the $n - m$ multiplications required to compute

[3]Except when $m = n$, where no additions are performed.

(a)

(b)
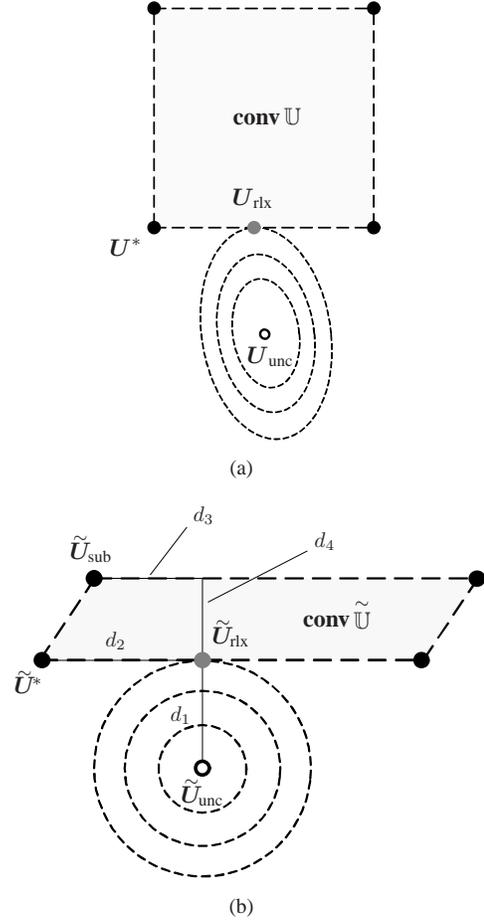
Fig. 4: Example of the projection principle and the underlying suboptimality due to a lattice with (greatly) unequal sides. The same notation and illustration principles are adopted as in Fig. 3. (a) Original space. (b) $\widetilde{\boldsymbol{H}}$-generated space: Consider the following case with lengths $d_1 = 0.2$, $d_2 = 0.3$, $d_3 = 0.2$ and $d_4 = 0.15$. $\widetilde{\boldsymbol{U}}^*$ is the optimal solution since $|| \widetilde{\boldsymbol{U}}_{\mathrm{unc}} - \widetilde{\boldsymbol{U}}^* || = \sqrt{0.13} < || \widetilde{\boldsymbol{U}}_{\mathrm{unc}} - \widetilde{\boldsymbol{U}}_{\mathrm{sub}} || = \sqrt{0.165}$. Point $\widetilde{\boldsymbol{U}}_{\mathrm{sub}}$ is the point closest to the projected point $\widehat{\boldsymbol{U}}_{\mathrm{rlx}}$ since $|| \widehat{\boldsymbol{U}}_{\mathrm{rlx}} - \widetilde{\boldsymbol{U}}^* || = 0.3 > || \widehat{\boldsymbol{U}}_{\mathrm{rlx}} - \widetilde{\boldsymbol{U}}_{\mathrm{sub}} || = 0.25$. Therefore, $\widehat{\boldsymbol{U}}_{\mathrm{sub}}$ is a suboptimal option.

the product $\widetilde{\boldsymbol{H}}_{(m,m:n)} \widetilde{\boldsymbol{U}}_{m:n}$ is either $\tilde{h}_{m,i}$, $-\tilde{h}_{m,i}$, or 0, with $m \leq i \leq n$. Finally, since the cardinality of the input set $\mathcal{U}$ is three, this implies that for each node visited both its sibling nodes need to be checked to ascertain whether they are inside the hypersphere or not.

Given the above analysis, the real-time additions N_a, subtractions N_s, and multiplications N_m are summarized as follows

$$N_a = 3 \left(\mu - 1 + \sum_{\nu=1}^{\mu} \left(n - m(\nu) \right) \right), \\ N_s = 3\mu, \\ N_m = 3\mu, \tag{17}$$

where $1 \leq m \leq n$ indicates the dimension of a node visited (i.e., level of the tree), and μ the maximum number of nodes explored. A more detailed analysis of the operations performed by the direct long-horizon MPC scheme and the employed optimizer can be found in [3], [48].

(a) Electromagnetic torque (solid line) T_e and its reference (dash-dotted line).

(b) Three-phase stator current i_s (solid lines) and their references (dash-dotted lines).

(c) Three-phase switch position u.

(d) Electromagnetic torque (solid line) T_e and its reference (dash-dotted line).

(e) Three-phase stator current i_s (solid lines) and their references (dash-dotted lines).

(f) Three-phase switch position u.

Fig. 5: Torque reference steps for direct MPC with a ten-step horizon ($N = 10$) at nominal speed. The sampling interval is $T_s = 25\,\mu$s and the switching frequency is approximately $300\,$Hz ($\lambda_u = 0.1$). (Upper row) Simulated waveforms with the original sphere decoder (see [3]). (Lower row) Simulated waveforms with the modified sphere decoder (i.e., the proposed approach).

VI. PERFORMANCE EVALUATION

To obtain the simulation results presented in this section, an MV drive (Fig. 1) consisting of a squirrel cage IM with $3.3\,$kV rated voltage, $356\,$A rated current, $2\,$MVA rated power, $50\,$Hz nominal frequency, 0.25 p.u. total leakage inductance, and a three-level NPC with constant dc-link voltage $V_{dc} = 5.2\,$kV and a fixed neutral point N, is considered. For all cases examined, the controller was operated with the sampling interval $T_s = 25\,\mu$s. All results are shown in the p.u. system.

The performance of the proposed direct MPC scheme is examined during torque transients to examine its dynamical behavior in terms of settling time and reference regulation. The horizon $N = 10$ case is investigated; the weighting factor $\lambda_u = 0.1$ is chosen, such that a switching frequency of approximately $300\,$Hz results. While operating at rated speed, reference torque steps of magnitude one are imposed. The response of the drive controlled with the sphere decoder in [3] (i.e., optimality is guaranteed) is shown in the upper row of Fig. 5, whereas that of the proposed approach in the lower row of the same figure. As can be seen, the electromagnetic torque in both cases tracks the new desired values as quickly as possible, with the settling time being limited only by the available dc-link voltage (see Figs. 5(a) and 5(d), respectively). In effect the controller behaves like a deadbeat controller. As for the currents, they accurately track their new references (the torque steps on the torque reference are translated into the corresponding current steady-state references), as shown in Figs. 5(b) and 5(e). Finally, in Figs. 5(c) and 5(f) the three-phase switching sequences are shown. It is worthwhile to mention that, as shown in [21], the limiting factor for the dynamical performance of direct MPC is the available dc-link voltage. The performance of the system during transients,

however, is the same, regardless of the prediction horizon length.

In Table I the proposed approach is compared with the sphere decoder in [3] in terms of (sub)optimality for different lengths of the prediction horizon and switching frequencies. Specifically, the weighting factor λ_u is adjusted such that the resulting average switching frequency is changed in a stepwise manner (with a step of $50\,$Hz) from 100 to $700\,$Hz. The same scenario as in Fig. 5 is examined, i.e., operation under step changes in the torque reference. As can be seen, thanks to the well-conditioned $\widetilde{H}$ optimality is sacrificed only for longer prediction horizons ($N \geq 5$) and very low switching frequencies, and even then marginally, without any significant effect on the drive performance, as shown in Fig. 5.

As for the computational complexity, a first rough analysis for the two aforementioned techniques and the strategy of exhaustive enumeration in terms of the nodes evaluated as a function of the prediction horizon length is presented in Table II, assuming same operating conditions as in Fig. 5. Moreover, the flops required by these three strategies—when the same conditions are considered—are depicted in Fig. 6. As can be seen, the sphere decoding algorithm with the refinements proposed here significantly reduces the number of examined nodes and real-time flops, even compared with the approach proposed in [3]. For example, the discussed method can visit less nodes and perform less operations by up to $99.7\,\%$ when a ten-step horizon and a step-up torque reference change are examined.

To provide further insight in the computational complexity of the proposed algorithm, Table III presents a comprehensive comparison with the sphere decoder in [3] in terms of flops for different prediction horizons and switching frequencies

TABLE I: Probability in percent that the solution computed by the proposed approach is the (global) optimal solution $\boldsymbol{U}^*$ for different prediction horizons and switching frequencies f_{sw} (the latter refer to steady-state operation).

Switching frequency f_{sw} [Hz]	Prediction horizon N						
	1	2	3	4	5	7	10
100	100	100	100	100	98.7	95.2	91.1
150	100	100	100	100	99.5	96.7	95.7
200	100	100	100	100	99.6	98.8	96.8
250	100	100	100	100	99.7	99.0	97.7
300	100	100	100	100	99.8	99.3	98.5
350	100	100	100	100	99.9	99.5	99.3
400	100	100	100	100	100	99.7	99.6
450	100	100	100	100	100	100	100
500	100	100	100	100	100	100	100
550	100	100	100	100	100	100	100
600	100	100	100	100	100	100	100
650	100	100	100	100	100	100	100
700	100	100	100	100	100	100	100

TABLE II: Maximum number of nodes $\mu_{\max}$ explored by (a) the exhaustive search algorithm, (b) the sphere decoder in [3], and (c) the proposed algorithm, during the step torque reference changes shown in Fig. 5 for different prediction horizons and switching frequency $f_{\mathrm{sw}} = 300\,\mathrm{Hz}$.

		Prediction horizon N						
		1	2	3	4	5	7	10
Exhaustive enumeration		39	1'092	29'523	797'160	21'523'359	$> 1.5 \cdot 10^{10}$	$> 3 \cdot 10^{14}$
Sphere decoder [3]	$\mu_{\max}$	7	23	43	165	460	1'579	36'092
Proposed approach		5	14	18	26	32	61	114

f_{sw}.[4] As before, for a given prediction horizon, the penalty on the switching transitions λ_u is tuned such that the desired switching frequency results. Following, the maximum number of flops $N_o^{\max}$ performed by the optimizer in real time is recorder during the torque reference changes, as presented in Fig. 5. As can be observed, owing to the projection onto the convex hull, the number of real-time flops during transient operation is greatly reduced, regardless of the nature of the step change.

In a last step, the transient performance of the proposed method is examined at low-speed operation, taking 10% speed as an example. While operating at a switching frequency of about $300\,\mathrm{Hz}$, and with a ten-step horizon, the torque reference is stepped down from its nominal value of $1\,\mathrm{p.u.}$ to zero at time instant $t = 50\,\mathrm{ms}$. Following, at $t = 100\,\mathrm{ms}$, the torque reference is stepped back up to $1\,\mathrm{p.u.}$. Observing the system behavior, as shown in Fig. 7, it can be claimed that the controller performance is as desired. It is important to point out that at low speeds optimality is not affected. In other words, both the original sphere decoder [3] as well as the algorithm proposed here provide the same solution in all problem instances[5]. Thus, the system performance with both optimizers is identical.

[4]Based on the results in Table II and Fig. 6, it is evident that exhaustive enumeration is not a feasible option, thus it is not considered in the comparisons that follow.

[5]An intuitive explanation for this argument is that the speed-dependent entries of $\boldsymbol{A}$ are dominant and get smaller as the speed decreases. Consequently, the off-diagonal entries of $\boldsymbol{Q}$ are also reduced, see Appendix B. The same applies to the off-diagonal entries of $\boldsymbol{H}$ (and $\widetilde{\boldsymbol{H}}$), as can be concluded from (20) and [49, Chapter 3]. The latter implies that the set of the lattice basis vectors, i.e., the columns of $\boldsymbol{H}$ (or $\widetilde{\boldsymbol{H}}$), is more orthogonal, while the basis vectors are of a smaller length.
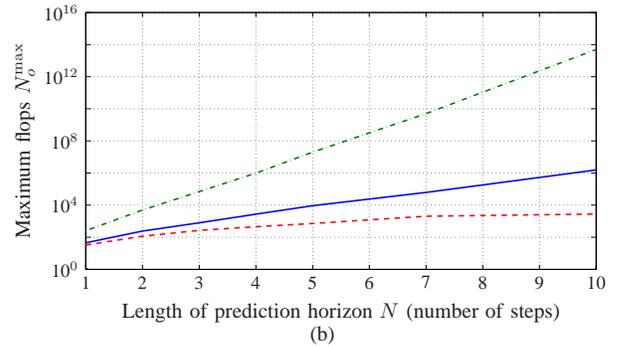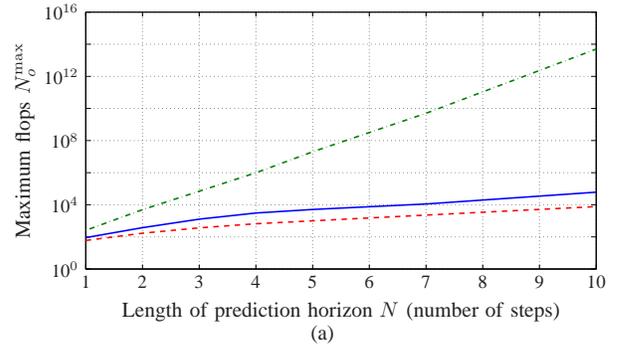
Fig. 6: Maximum number of flops $N_o^{\max}$ preformed during the search process when the (a) step-down and (b) step-up torque reference changes in Fig. 5 occur as a function of the prediction horizon N. The solid (blue) line refers to the sphere decoder in [3], the dashed (red) line to the presented algorithm, and the dash-dotted (green) line to the exhaustive enumeration.

VII. CONCLUSIONS

This paper proposes refinements for the sphere decoding algorithm employed to solve the long-horizon direct model predictive control (MPC) problem for transient operation. By

TABLE III: Maximum number of flops $N_o^{\max}$ preformed during the search process when torque reference changes are applied for different prediction horizons and switching frequencies f_{sw}. The flops performed by the proposed approach are denoted by $N_{o1}^{\max}$, whereas those by the sphere decoder in [3] by $N_{o2}^{\max}$.

Switching frequency f_{sw} [Hz]	Prediction horizon N													
	1		2		3		4		5		7		10	
	$N_{o1}^{\max}$	$N_{o2}^{\max}$	$N_{o1}^{\max}$	$N_{o2}^{\max}$	$N_{o1}^{\max}$	$N_{o2}^{\max}$	$N_{o1}^{\max}$	$N_{o2}^{\max}$	$N_{o1}^{\max}$	$N_{o2}^{\max}$	$N_{o1}^{\max}$	$N_{o2}^{\max}$	$N_{o1}^{\max}$	$N_{o2}^{\max}$
100	60	90	180	391	339	639	600	1'371	810	3'552	1'693	7'938	3'879	12'213
150	60	120	216	403	357	788	633	2'772	834	4'257	1'899	12'912	4'425	$> 1 \cdot 10^5$
200	60	132	237	414	366	951	663	3'045	895	4'695	2'067	18'598	6'213	$> 2 \cdot 10^5$
250	60	162	237	532	366	1'262	663	3'628	912	5'943	2'169	39'703	6'943	$> 6 \cdot 10^5$
300	60	177	237	651	366	1'578	663	3'936	996	6'839	2'289	58'054	7'728	$> 1 \cdot 10^6$
350	60	177	237	708	417	1'785	678	4'038	1'005	9'401	2'694	95'457	12'035	$> 7 \cdot 10^6$
400	60	177	240	900	474	2'004	712	5'769	1'035	17'452	3'645	$> 1 \cdot 10^5$	16'875	$> 2 \cdot 10^7$
450	60	177	240	948	474	3'624	753	9'126	1'044	22'911	4'086	$> 3 \cdot 10^5$	19'672	$> 9 \cdot 10^7$
500	60	177	240	1'149	479	4'011	785	9'555	1'093	34'274	4'914	$> 8 \cdot 10^5$	25'873	$> 6 \cdot 10^8$
550	60	177	243	1'233	486	4'161	802	10'327	1'101	35'882	5'031	$> 6 \cdot 10^6$	28'388	$> 2 \cdot 10^9$
600	60	177	243	1'461	486	5'244	831	12'086	1'140	41'082	5'073	$> 1 \cdot 10^7$	30'503	$> 8 \cdot 10^9$
650	60	177	249	1'641	493	5'525	927	14'266	1'376	46'596	5'839	$> 9 \cdot 10^7$	31'458	$> 4 \cdot 10^{10}$
700	60	177	249	1'779	493	6'108	930	15'311	1'629	77'154	6'927	$> 5 \cdot 10^8$	36'521	$> 2 \cdot 10^{11}$
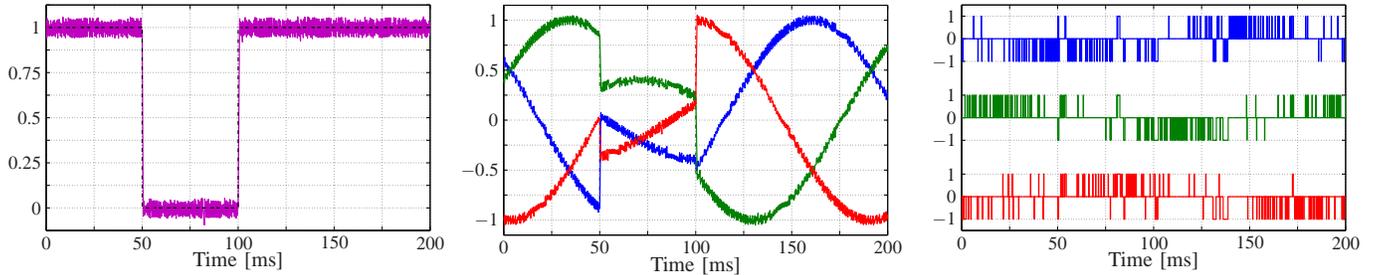
(a) Electromagnetic torque (solid line) T_e and its reference (dash-dotted line).

(b) Three-phase stator current i_s (solid lines) and their references (dash-dotted lines).

(c) Three-phase switch position u.

Fig. 7: Torque reference steps for direct MPC when operating at 0.1p.u. speed. A ten-step horizon ($N = 10$) is used, the sampling interval is $T_s = 25\,\mu$s and the weighting factor $\lambda_u = 0.15$ such that a switching frequency of approximately 300 Hz results. The results with the original sphere decoder (see [3]) and the modified one, i.e., the proposed approach, are identical.

exploiting the geometry of the underlying quadratic program (QP) a new, tighter sphere is computed that, although it sacrifices optimality (to some degree) when longer horizons are of concern, it can significantly reduce the computational complexity of the integer problem. Thanks to the proposed modifications, and given the presented case study, i.e., a three-level converter driving an induction machine, the computational burden can be reduced by more than 97.5% for a wide range of switching frequencies and for long horizons, compared to that required for the search algorithm in [3]. It can be concluded that the proposed modifications render the optimization problem of concern computationally tractable, thus facilitating the implementation of the sphere decoder for medium-voltage (MV) drives since continuous operation of the system—even during transients—is ensured. This has been *experimentally* verified with a field programmable gate array (FPGA) based control platform in [50]. Therein, the performance—both steady-state and transient—of a scaled-down prototype of a three-level neutral point clamped (NPC) inverter with an RL load is examined.

APPENDIX A
CONTINUOUS-TIME MODEL OF THE DRIVE

The matrices D, E, and F of the continuous-time state-space model of the drive (3) are

$$D = \begin{bmatrix} -\frac{1}{\tau_s} & 0 & \frac{X_m}{\tau_r \Phi} & \omega_r \frac{X_m}{\Phi} \\ 0 & -\frac{1}{\tau_s} & -\omega_r \frac{X_m}{\Phi} & \frac{X_m}{\tau_r \Phi} \\ \frac{X_m}{\tau_r} & 0 & -\frac{1}{\tau_r} & -\omega_r \\ 0 & \frac{X_m}{\tau_r} & \omega_r & -\frac{1}{\tau_r} \end{bmatrix},$$

$$E = \frac{X_r}{\Phi} \frac{V_{\text{dc}}}{2} \begin{bmatrix} 1 & 0 \\ 0 & 1 \\ 0 & 0 \\ 0 & 0 \end{bmatrix}, \quad F = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix}.$$

APPENDIX B
DERIVATION OF THE ILS PROBLEM

By introducing $Y(k) = [y^T(k+1) \ \dots \ y^T(k+N)]^T$ and $Y_{\text{ref}}(k) = [y_{\text{ref}}^T(k+1) \ \dots \ y_{\text{ref}}^T(k+N)]^T$ to denote the output and the corresponding output reference sequences over the horizon, respectively, function (5) can be written in vector form as

$$J = ||\Gamma x(k) + \Upsilon U(k) - Y_{\text{ref}}||_2^2 + \lambda_u ||S U(k) - \Xi u(k-1)||_2^2, \tag{18}$$

where it was used the fact that $\mathbf{Y}(k) = \mathbf{\Gamma}\mathbf{x}(k) + \mathbf{\Upsilon}\mathbf{U}(k)$, with the matrices $\mathbf{\Upsilon}$, $\mathbf{\Gamma}$, $\mathbf{S}$ and $\mathbf{\Xi}$ being

$$\mathbf{\Upsilon} = \begin{bmatrix} \mathbf{CBK} & \mathbf{0} & \cdots & \mathbf{0} \\ \mathbf{CABK} & \mathbf{CBK} & \cdots & \mathbf{0} \\ \vdots & \vdots & & \vdots \\ \mathbf{CA}^{N-1}\mathbf{BK} & \mathbf{CA}^{N-2}\mathbf{BK} & \cdots & \mathbf{CBK} \end{bmatrix},$$

$$\mathbf{\Gamma} = \begin{bmatrix} \mathbf{CA} \\ \mathbf{CA}^2 \\ \vdots \\ \mathbf{CA}^N \end{bmatrix}, \ \mathbf{S} = \begin{bmatrix} \mathbf{I} & \mathbf{0} & \cdots & \mathbf{0} \\ -\mathbf{I} & \mathbf{I} & \cdots & \mathbf{0} \\ \mathbf{0} & -\mathbf{I} & \cdots & \mathbf{0} \\ \vdots & \vdots & & \vdots \\ \mathbf{0} & \mathbf{0} & \cdots & \mathbf{I} \end{bmatrix}, \ \mathbf{\Xi} = \begin{bmatrix} \mathbf{I} \\ \mathbf{0} \\ \mathbf{0} \\ \vdots \\ \mathbf{0} \end{bmatrix},$$

where $\mathbf{0}$ is the zero matrix of appropriate dimensions.

After some algebraic manipulations, (18) takes the form

$$J(k) = ||\mathbf{U}(k) + \mathbf{Q}^{-1}\mathbf{\Lambda}(k)||_{\mathbf{Q}}^2 + \underbrace{\zeta(k) - \mathbf{\Lambda}^T(k)\mathbf{Q}^{-T}\mathbf{\Lambda}(k)}_{\text{const}(k)}. \tag{19}$$

where

$$\zeta(k) = ||\mathbf{\Gamma}\mathbf{x}(k) - \mathbf{Y}_{\text{ref}}(k)|| + \lambda_u||\mathbf{\Xi}\mathbf{u}(k-1)||,$$
$$\mathbf{Q} = \mathbf{\Upsilon}^T\mathbf{\Upsilon} + \lambda_u\mathbf{S}^T\mathbf{S},$$
$$\mathbf{\Lambda}(k) = \mathbf{\Upsilon}^T\big(\mathbf{\Gamma}\mathbf{x}(k) - \mathbf{Y}_{\text{ref}}(k)\big) - \lambda_u\mathbf{S}^T\mathbf{\Xi}\mathbf{u}(k-1).$$

Following, by noticing that $\lambda_u > 0 \Rightarrow \mathbf{Q} \succ 0$, then $\mathbf{Q}$ can be decomposed as

$$\mathbf{Q} = \mathbf{H}^T\mathbf{H}. \tag{20}$$

Using (20) and the unconstrained solution of (19), i.e., $\mathbf{U}_{\text{unc}}(k) = -\mathbf{Q}^{-1}\mathbf{\Lambda}(k)$, and by neglecting the constant term since it is independent of $\mathbf{U}(k)$, (19) is written as

$$J = ||\bar{\mathbf{U}}_{\text{unc}}(k) - \mathbf{H}\mathbf{U}(k)||_2^2, \tag{21}$$

which is the ILS function of problem (7). ∎

REFERENCES

[1] P. Karamanakos, T. Geyer, and R. P. Aguilera, "Computationally efficient long-horizon direct model predictive control for transient operation," in *Proc. IEEE Energy Convers. Congr. Expo.*, Cincinnati, OH, Oct. 2017, pp. 4642–4649.

[2] 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.

[3] P. Karamanakos, T. Geyer, and R. Kennel, "Reformulation of the long-horizon direct model predictive control problem to reduce the computational effort," in *Proc. IEEE Energy Convers. Congr. Expo.*, Pittsburgh, PA, Sep. 2014, pp. 3512–3519.

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

[5] 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.

[6] T. Geyer, "A comparison of control and modulation schemes for medium-voltage drives: Emerging predictive control concepts versus PWM-based schemes," *IEEE Trans. Ind. Appl.*, vol. 47, no. 3, pp. 1380–1389, May/Jun. 2011.

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

[8] J. Rodríguez, J. Pontt, C. A. Silva, P. Correa, P. Lezana, P. Cortés, and U. Ammann, "Predictive current control of a voltage source inverter," *IEEE Trans. Ind. Electron.*, vol. 54, no. 1, pp. 495–503, Feb. 2007.

[9] P. Karamanakos, A. Ayman, and R. Kennel, "A variable switching point predictive current control strategy for quasi-Z-source inverters," *IEEE Trans. Ind. Appl.*, vol. DOI 10.1109/TIA.2017.2765302, to appear, 2017.

[10] L. Tarisciotti, P. Zanchetta, A. Watson, J. C. Clare, M. Degano, and S. Bifaretti, "Modulated model predictive control for a three-phase active rectifier," *IEEE Trans. Ind. Appl.*, vol. 51, no. 2, pp. 1610–1620, Mar./Apr. 2015.

[11] L. Tarisciotti, J. Lei, A. Formentini, A. Trentin, P. Zanchetta, P. Wheeler, and M. Rivera, "Modulated predictive control for indirect matrix converter," *IEEE Trans. Ind. Appl.*, vol. 53, no. 5, pp. 4644–4654, Sep./Oct. 2017.

[12] Y. Zhang, Y. Peng, and C. Qu, "Model predictive control and direct power control for PWM rectifiers with active power ripple minimization," *IEEE Trans. Ind. Appl.*, vol. 52, no. 6, pp. 4909–4918, Nov./Dec. 2016.

[13] T. Geyer and S. Mastellone, "Model predictive direct torque control of a five-level ANPC converter drive system," *IEEE Trans. Ind. Appl.*, vol. 48, no. 5, pp. 1565–1575, Sep./Oct. 2012.

[14] S. Mariéthoz, A. Domahidi, and M. Morari, "High-bandwidth explicit model predictive control of electrical drives," *IEEE Trans. Ind. Appl.*, vol. 48, no. 6, pp. 1980–1992, Nov./Dec. 2012.

[15] Q. Liu and K. Hameyer, "Torque ripple minimization for direct torque control of PMSM with modified FCSMPC," *IEEE Trans. Ind. Appl.*, vol. 52, no. 6, pp. 4855–4864, Nov./Dec. 2016.

[16] 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.

[17] Y. Zhang, D. Xu, J. Liu, S. Gao, and W. Xu, "Performance improvement of model-predictive current control of permanent magnet synchronous motor drives," *IEEE Trans. Ind. Appl.*, vol. 53, no. 4, pp. 3683–3695, Jul./Aug. 2017.

[18] T. Geyer, N. Oikonomou, G. Papafotiou, and F. D. Kieferndorf, "Model predictive pulse pattern control," *IEEE Trans. Ind. Appl.*, vol. 48, no. 2, pp. 663–676, Mar./Apr. 2012.

[19] N. Oikonomou, C. Gutscher, P. Karamanakos, F. D. Kieferndorf, and T. Geyer, "Model predictive pulse pattern control for the five-level active neutral point clamped inverter," *IEEE Trans. Ind. Appl.*, vol. 49, no. 6, pp. 2583–2592, Nov./Dec. 2013.

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

[21] 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.

[22] 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.

[23] P. Karamanakos, T. Geyer, and R. Kennel, "A computationally efficient model predictive control strategy for linear systems with integer inputs," *IEEE Trans. Control Syst. Technol.*, vol. 24, no. 4, pp. 1463–1471, Jul. 2016.

[24] 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.

[25] R. Cagienard, P. Grieder, E. C. Kerrigan, and M. Morari, "Move blocking strategies in receding horizon control," *J. of Process Control*, vol. 17, no. 6, pp. 563–570, Jul. 2007.

[26] P. Karamanakos, T. Geyer, and S. Manias, "Direct voltage control of dc-dc boost converters using enumeration-based model predictive control," *IEEE Trans. Power Electron.*, vol. 29, no. 2, pp. 968–978, Feb. 2014.

[27] T. Geyer, G. Papafotiou, and M. Morari, "Model predictive direct torque control—Part I: Concept, algorithm and analysis," *IEEE Trans. Ind. Electron.*, vol. 56, no. 6, pp. 1894–1905, Jun. 2009.

[28] E. L. Lawler and D. E. Wood, "Branch-and-bound methods: A survey," *Op. Res.*, vol. 14, no. 4, pp. 699–719, Jul./Aug. 1966.

[29] 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.

[30] B. Hassibi and H. Vikalo, "On the sphere-decoding algorithm I. Expected complexity," *IEEE Trans. Signal Proc.*, vol. 53, no. 8, pp. 2806–2818, Aug. 2005.

[31] P. Karamanakos, T. Geyer, and R. Kennel, "Constrained long-horizon direct model predictive control for power electronics," in *Proc. IEEE Energy Convers. Congr. Expo.*, Milwaukee, WI, Sep. 2016, pp. 1–8.

[32] R. Baidya, R. P. Aguilera, P. Acuna, R. Delgado, T. Geyer, D. Quevedo, and T. Mouton, "Fast multistep finite control set model predictive control for transient operation of power converters," in *Proc. IEEE Ind. Electron. Conf.*, Florence, Italy, Oct. 2016, pp. 5039–5045.

[33] S. N. Manias, *Power electronics and motor drive systems*. Cambridge, MA: Academic Press, 2016.

[34] 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.

[35] D. Micciancio, "The hardness of the closest vector problem with preprocessing," *IEEE Trans. Inf. Theory*, vol. 47, no. 3, pp. 1212–1215, Mar. 2001.

[36] A. K. Lenstra, H. W. Lenstra, Jr., and L. Lovász, "Factoring polynomials with rational coefficients," *Math. Ann.*, vol. 261, no. 4, pp. 515–534, 1982.

[37] P. Karamanakos, T. Geyer, and R. Kennel, "Suboptimal search strategies with bounded computational complexity to solve long-horizon direct model predictive control problems," in *Proc. IEEE Energy Convers. Congr. Expo.*, Montreal, QC, Canada, Sep. 2015, pp. 334–341.

[38] L. Babai, "On Lovász' lattice reduction and the nearest lattice point problem," *Combinatorica*, vol. 6, no. 1, pp. 1–13, 1986.

[39] S. Boyd and L. Vandenberghe, *Convex Optimization*. Cambridge, UK: Cambridge Univ. Press, 2004.

[40] Y. Nesterov, "A method of solving a convex programming problem with convergence rate $O(1/k^2)$," *Soviet Mathematics Doklady*, vol. 27, no. 2, pp. 372–376, Feb. 1983.

[41] J. Barzilai and J. M. Borwein, "Two-point step size gradient methods," *IMA J. of Numer. Anal.*, vol. 8, no. 1, pp. 141–148, Jan. 1988.

[42] M. Raydan, "On the barzilai and borwein choice of steplength for the gradient method," *IMA J. of Numer. Anal.*, vol. 13, no. 3, pp. 321–326, Jul. 1993.

[43] A. Friedlander, J. M. Martinez, and M. Raydan, "A new method for large-scale box constrained convex quadratic minimization problems," *Optim. Methods and Software*, vol. 5, no. 1, pp. 57–74, Jan. 1995.

[44] A. R. Conn, N. I. M. Gould, and P. L. Toint, "Global convergence of a class of trust region algorithms for optimization with simple bounds," *SIAM J. on Numer. Anal.*, vol. 25, no. 2, pp. 433–460, Mar./Apr. 1988.

[45] A. Beck and M. Teboulle, "Fast gradient-based algorithms for constrained total variation image denoising and deblurring problems," *IEEE Trans. Image Proc.*, vol. 18, no. 11, pp. 2419–2434, Nov. 2009.

[46] B. Hauska, H. J. Ferreau, and M. Diehl, "An auto-generated real-time iteration algorithm for nonlinear MPC in the microsecond range," *Automatica*, vol. 47, no. 10, pp. 2279–2285, Oct. 2011.

[47] P. Zometa, M. Kögel, T. Faulwasser, and R. Findeisen, "Implementation aspects of model predictive control for embedded systems," in *Proc. Am. Control Conf.*, Montreal, QC, Canada, Jun. 2012, pp. 1205–1210.

[48] P. Karamanakos, T. Geyer, T. Mouton, and R. Kennel, "Computationally efficient sphere decoding for long-horizon direct model predictive control," in *Proc. IEEE Energy Convers. Congr. Expo.*, Milwaukee, WI, Sep. 2016, pp. 1–8.

[49] G. W. Stewart, *Matrix algorithms, Volume I: Basic decompositions*. Philadelphia, PA: SIAM, 1998.

[50] M. Dorfling, H. Mouton, P. Karamanakos, and T. Geyer, "Experimental evaluation of sphere decoding for long-horizon direct model predictive control," in *Proc. Eur. Power Electron. Conf.*, Warsaw, Poland, Sep. 2017, pp. P.1–P.10.

Petros Karamanakos (S'10–M'14) received the Diploma and the Ph.D. degrees in electrical and computer engineering from the National Technical University of Athens (NTUA), Athens, Greece, in 2007, and 2013, respectively.

From 2010 to 2011 he was with the ABB Corporate Research Center, Baden-Dättwil, Switzerland, where he worked on model predictive control strategies for medium-voltage drives. From 2013 to 2016 he was a PostDoc Research Associate in the Chair of Electrical Drive Systems and Power Electronics, Technische Universität München, Munich, Germany. He is currently an Assistant Professor in the Faculty of Computing and Electrical Engineering, Tampere University of Technology, Tampere, Finland. His main research interests lie at the intersection of optimal control, mathematical programming and power electronics, including model predictive control for power electronic converters and ac drives.

Dr. Karamanakos received the 2014 Third Best Paper Award of the IEEE Transactions on Industry Applications and the First Prize Paper Award of the Industrial Drives Committee at the 2013 IEEE Energy Conversion Congress and Exposition.

Tobias Geyer (M'08–SM'10) received the Dipl.-Ing. and Ph.D. degrees in electrical engineering from ETH Zurich, Zurich, Switzerland, in 2000 and 2005, respectively, and the Habilitation degree in power electronics from ETH Zurich, Zurich, Switzerland, in 2017.

After his Ph.D., he spent three years at GE Global Research, Munich, Germany, and another three years at the University of Auckland, Auckland, New Zealand. In 2012, he joined ABB's Corporate Research Centre, Baden-Dättwil, Switzerland, where he is currently a Senior Principal Scientist for power conversion control. He was appointed as an extraordinary Professor at Stellenbosch University, Stellenbosch, South Africa, from 2017 to 2020. He is the author of more than 100 peer-reviewed publications, 30 patent applications, and the book "Model predictive control of high power converters and industrial drives" (Wiley, 2016). He teaches a regular course on model predictive control at ETH Zurich. His research interests include model predictive control, medium-voltage drives and utility-scale power converters.

Dr. Geyer was the recipient of the 2014 Third Best Paper Award of the Transactions on Industry Applications and of two Prize Paper Awards at conferences. He served as an Associate Editor of the Transactions on Industry Applications. Since 2013 he has been serving as an Associate Editor for the Transactions on Power Electronics.

Ricardo P. Aguilera (S'01–M'12) received the B.Sc. degree in electrical engineering from the Universidad de Antofagasta, Antofagasta, Chile; the M.Sc. degree in electronics engineering from the Universidad Tecnica Federico Santa Maria, Valparaíso, Chile; and the Ph.D. degree in electrical engineering from The University of Newcastle (UoN), Newcastle, NSW, Australia, in 2003, 2007, and 2012, respectively.

From 2012 to 2013, he was a Research Academic at UoN, where he was part of the Centre for Complex Dynamic Systems and Control. From 2014 to 2016, he was a Senior Research Associate in The University of New South Wales, Sydney, NSW, Australia, where he was part of the Australian Energy Research Institute. Since September 2016, he has been with the School of Electrical and Data Engineering, University of Technology Sydney, Sydney, NSW, Australia, where he currently holds a Lecturer position. His main research interests include theoretical and practical aspects on model predictive control with application to power electronics, renewable energy integration, and microgrids.