Constrained Long-Horizon Direct Model Predictive Control for Power Electronics

Petros Karamanakos, *Member, IEEE*, Tobias Geyer, *Senior Member, IEEE*, and Ralph Kennel, *Senior Member, IEEE*

***Abstract*—The direct model predictive control (MPC) problem for linear systems with integer inputs, such as many power electronic systems, can be formulated as an integer least-squares (ILS) optimization problem. However, solving this problem when state and/or output constraints are explicitly included is challenging. In this paper, a method that allows to effectively use the sphere decoder—even in the presence of the aforementioned constraints—is proposed. This is done by computing a new hypersphere based on the feasible control input set, as defined by the imposed state/output constraints. A variable speed drive system with a three-level voltage source inverter serves as an illustrative example to demonstrate the performance of the proposed algorithm.**

I. Introduction

Model predictive control (MPC) [1] is usually implemented in power electronics as a direct control strategy, i.e., the switching signals are computed and applied directly to the converter in one computational stage, hence an additional modulation stage is not required [2]–[5]. Modeling, however, the switch positions with integer variables comes at a cost. The formulated problem is a (mixed) integer program which is computationally very demanding since its complexity grows exponentially with the number of the controlled variables (e.g., the number of switch positions) and the prediction steps [6]–[8]. This limits the applicability of MPC in power electronics, especially when considering that not only long prediction horizons are often required for good closed-loop performance [9], but also the underlying optimization problem is predominantly solved with the brute-force approach of exhaustive enumeration [10]. Despite a few methods that allow for MPC with long (nontrivial) horizons to be implemented in a real-time system [11], [12], computational challenges still exist that need to be addressed.

Recently, a dedicated branch-and-bound algorithm that allows one to solve the underlying optimization problem—formulated as an integer least-squares (ILS) problem—in a computationally efficient manner was proposed in [13] and evaluated in [14]. This strategy, called sphere decoding [15], [16], has shown promising results when applied to linear systems with integer inputs [17], [18], i.e., systems governed by linear differential equations with integer inputs (e.g., on/off switches), such as power electronics [13], [14], [19], [20]. Moreover, despite the nature of the optimization problem (it is NP-hard), it was shown that by (occasionally) allowing for suboptimal solutions real-time termination guarantees can be provided without significantly deteriorating the performance of the system [21]. These improvements, along with a refined optimization stage to speed it up [22], facilitate the implementation of long-horizon MPC schemes to meet performance requirements, especially when higher-order systems are concerned [23].

For the latter systems, though, employing the sphere decoder is not straightforward. The highly correlated dynamics of such systems result in overshoots (e.g., overcurrents and/or overvoltages) that may damage the hardware, meaning that bounds on the controlled and/or state variables need to be imposed in the form of safety constraints. When state and/or output constraints have to be explicitly taken into consideration during the formulation of the problem, the controller design becomes more complicated. The reason is that the design of the bounded set that includes the candidate solutions to the *constrained* ILS problem is nontrivial since it should include at least one point that does not violate the constraints.

This paper presents a method that translates the output constraints into input constraints. By doing so, a new feasible input set is defined and used to compute a new—as small as possible—hypersphere that includes candidate solutions that do not violate the introduced constraints. Moreover, it is guaranteed that the points within the sphere that do violate the constraints are not considered as candidate solutions, and are thus effectively discarded. To present the proposed method as well as its effectiveness in a simple yet meaningful manner, a first-order system is chosen as a case study. More specifically, the algorithm is tested on 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). As shown, the imposed stator current constraints are fully respected, thus current excursions that may lead to overcurrents are avoided. That way damages are prevented and a trip due to overcurrent conditions in the drive system is avoided.

II. Constrained Optimal Control Problem

Consider the current control problem of an IM 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).

P. Karamanakos and R. Kennel are with the Institute for Electrical Drive Systems and Power Electronics, Technische Universität München, 80333 Munich, Germany; e-mails: p.karamanakos@ieee.org, kennel@ieee.org

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

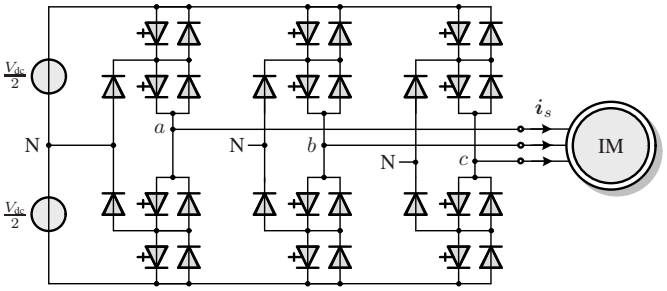


Fig. 1: Three-level three-phase neutral point clamped (NPC) voltage source inverter driving an induction motor (IM) with a fixed neutral point potential.

A. Control Model

In this section, the mathematical model of the examined case study is derived. To do so, the system variables are transformed from the three-phase system (abc) to the stationary orthogonal $\alpha\beta$ system. Thus, any variable in the abc-plane $\boldsymbol{\xi}_{abc} = [\xi_a \ \xi_b \ \xi_c]^T$ is transformed to a variable in the $\alpha\beta$-plane $\boldsymbol{\xi}_{\alpha\beta} = [\xi_\alpha \ \xi_\beta]^T$ via the transformation matrix $\boldsymbol{K}$, i.e., $\boldsymbol{\xi}_{\alpha\beta} = \boldsymbol{K}\boldsymbol{\xi}_{abc}$, with

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

The three-level NPC inverter can produce at each phase the three discrete voltage levels $-\frac{V_{\mathrm{dc}}}{2}$, 0, $\frac{V_{\mathrm{dc}}}{2}$, depending on the position of the corresponding phase switches. These switch positions can be described by the integer variable $u_x \in \mathcal{U} = \{-1, 0, 1\}$ with $x \in \{a, b, c\}$. By introducing the vector $\boldsymbol{u} = [u_a \ u_b \ u_c]^T \in \boldsymbol{\mathcal{U}} = \mathcal{U} \times \mathcal{U} \times \mathcal{U} = \mathcal{U}^3$ to denote the three-phase switch position, then the inverter output voltage—equal to the voltage applied to the machine terminals $\boldsymbol{v}_{s,\alpha\beta}$—is given by[1]

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

To accurately describe the dynamics of a squirrel-cage IM it is convenient to use the stator current $\boldsymbol{i}_{s,\alpha\beta}$ and the rotor flux $\boldsymbol{\psi}_{r,\alpha\beta}$ in the $\alpha\beta$-plane as state variables. The mechanical speed is assumed to be constant, thus the rotor angular speed ω_r is considered to be a time-varying parameter and not a state variable. The continuous-time state equations are[2] [24]

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

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

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

[1]Throughout the paper, the subscript $\alpha\beta$ is used to denote vectors in the $\alpha\beta$-plane. For vectors in the abc-plane the subscript is omitted.

[2]Since all vectors in (2) are in the $\alpha\beta$-plane we drop the subscripts to simplify the notation.

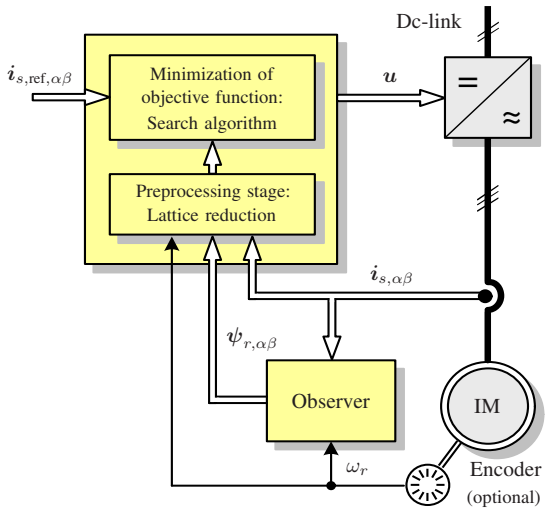


Fig. 2: Model predictive current control with reference tracking for the three-phase three-level NPC inverter with an induction machine.

where $\boldsymbol{I}$ is the identity matrix of appropriate dimension (here two-dimensional). Moreover, 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 are the model parameters. In (2) the constant Φ is defined as $\Phi = X_s X_r - X_m^2$, with $X_s = X_{ls} + X_m$ and $X_r = X_{lr} + X_m$, whereas $\tau_s = X_r\Phi/(R_s X_r^2 + R_r X_m^2)$ and $\tau_r = X_r/R_r$ denote the stator and rotor time constants, respectively. Finally, T_e is the electromagnetic torque.

Based on (1) and (2), the state-space representation of the drive system in the continuous-time domain 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 stator current and the rotor flux in the $\alpha\beta$-plane form the state vector, i.e., $\boldsymbol{x} = [i_{s\alpha} \ i_{s\beta} \ \psi_{r\alpha} \ \psi_{r\beta}]^T$, the three-phase switch position $\boldsymbol{u} = [u_a \ u_b \ u_c]^T$ constitutes the input vector, and the stator current is considered as the output of the system, i.e., $\boldsymbol{y} = \boldsymbol{i}_{s,\alpha\beta}$. The matrices $\boldsymbol{D}$, $\boldsymbol{E}$, and $\boldsymbol{F}$ are given in the appendix.

The discretized version of the continuous-time system (3) is derived by using exact Euler discretization. This takes the form

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

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

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

B. Constrained Direct Model Predictive Control With Current Reference Tracking

The block diagram of the presented MPC strategy is shown in Fig. 2. As can be deduced, the main control objective of the discussed control algorithm is to eliminate the stator

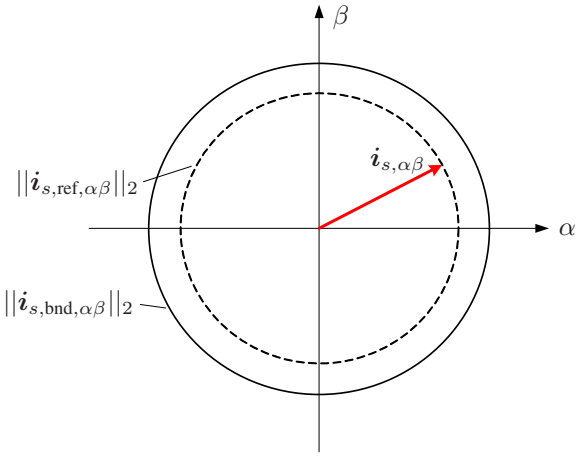


Fig. 3: Bounds on the stator current in the $\alpha\beta$-plane.

current error by directly manipulating the inverter switches, i.e., without the use of a modulator. Moreover, considering that for MV drives the switching power losses should be kept as low as possible, an additional goal is to operate the inverter at switching frequencies of a few hundred Hertz in order to indirectly reduce the aforementioned losses.

By defining the current (output) error as the difference between the (measured) stator current $\boldsymbol{i}_{s,\alpha\beta}$ and its demanded value $\boldsymbol{i}_{s,\text{ref},\alpha\beta}$, i.e., $\boldsymbol{i}_{s,\text{err},\alpha\beta}(k) = \boldsymbol{i}_{s,\text{ref},\alpha\beta}(k) - \boldsymbol{i}_{s,\alpha\beta}(k)$, and the switching (control) effort as the difference between two consecutive three-phase switch positions, i.e., $\Delta\boldsymbol{u}(k) = \boldsymbol{u}(k) - \boldsymbol{u}(k-1)$, the objective function

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

is formulated. Function (5) maps the evolution of the two above-mentioned terms over the finite prediction horizon of length N (time-steps) into a scalar. To set the trade-off between the stator current tracking accuracy and the switching frequency f_{sw}, the factor $\lambda_u > 0$ is introduced.

Before formulating the optimal control problem, an additional task needs to be taken into account. More specifically, the aforementioned control goals have to be met while protecting the inverter from overcurrents. To achieve this, the controlled variable(s), i.e., the stator current, should not exceed specified bounds. This can be expressed in the form of the inequality

$$||\boldsymbol{i}_{s,\alpha\beta}||_2 \leq i_{s,\text{bnd}}\,, \quad (6)$$

where the positive scalar $i_{s,\text{bnd}} \in \mathbb{R}^+$ defines the boundary value of the stator current. The current limitation in the $\alpha\beta$-plane is illustrated in Fig. 3.

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 best system performance—as quantified by (5)—while respecting the system constraints—as expressed by the plant model (4) and the current constraints (6), see Fig. 3—the following problem is formulated and solved at time-step k

$$\underset{\boldsymbol{U}(k)\in\mathbb{U}}{\text{minimize}} \quad J(k) \quad (7a)$$

$$\text{subject to} \quad \boldsymbol{x}(\ell+1) = \boldsymbol{A}\boldsymbol{x}(\ell) + \boldsymbol{B}\boldsymbol{K}\boldsymbol{u}(\ell) \quad (7b)$$

$$\boldsymbol{y}(\ell+1) = \boldsymbol{C}\boldsymbol{x}(\ell+1)\,,\ \forall\,\ell = k,\dots,k+N-1 \quad (7c)$$

$$||\boldsymbol{y}(k+1)||_2 \leq y_{\text{bnd}}\,. \quad (7d)$$

In (7), $\boldsymbol{U}(k) = [\boldsymbol{u}^T(k)\ \boldsymbol{u}^T(k+1)\dots\boldsymbol{u}^T(k+N-1)]^T$ is the optimization variable, and $\mathbb{U}$ is the feasible set defined as $\mathbb{U} = \boldsymbol{\mathcal{U}}^N \subset \mathbb{Z}^n$, i.e., the N-times Cartesian product of the set $\boldsymbol{\mathcal{U}} = \mathcal{U}\times\mathcal{U}\times\mathcal{U} = \mathcal{U}^3$, with $n = 3N$. Note that constraint (7d) corresponds to the current limitation, with $y_{\text{bnd}} = i_{s,\text{bnd}}$.

III. Equivalent Constrained Integer Least-Squares Problem

As shown in [13], the unconstrained solution $\boldsymbol{U}_{\text{unc}}(k) = [\boldsymbol{u}_{\text{unc}}^T(k)\ \dots\ \boldsymbol{u}_{\text{unc}}^T(k+N-1)]^T$ of (7) can be easily found by neglecting, in a first step, constraint (7d), and relaxing the integer constraint $\boldsymbol{U}(k) \in \mathbb{U}$, i.e., allowing $\boldsymbol{U}(k) \in \mathbb{R}^n$. By doing so, problem (7) can be written as the equivalent constrained ILS problem

$$\begin{aligned} \underset{\boldsymbol{U}(k)\in\mathbb{U}}{\text{minimize}} \quad & ||\bar{\boldsymbol{U}}_{\text{unc}}(k) - \boldsymbol{H}\boldsymbol{U}(k)||_2^2\,, \\ \text{subject to} \quad & ||\boldsymbol{y}(k+1)||_2 \leq y_{\text{bnd}}\,, \end{aligned} \quad (8)$$

where $\bar{\boldsymbol{U}}_{\text{unc}}(k) = \boldsymbol{H}\boldsymbol{U}_{\text{unc}}(k) \in \mathbb{R}^n$, and $\boldsymbol{H} \in \mathbb{R}^{n\times n}$ is a nonsingular, upper triangular matrix, known as the lattice generator matrix, since it generates the n-dimensional discrete space (lattice) one point of which is the solution to problem (8).

Problem (8), though, may be ill-conditioned. To have a well-conditioned one, the basis vectors, i.e., the columns of the lattice generator matrix, should be (nearly) orthogonal and of (relatively) small length. For this, the Lenstra-Lenstra-Lovász (LLL) lattice basis reduction algorithm [25] is employed that transforms $\boldsymbol{H}$ to the reduced lattice generator matrix $\widetilde{\boldsymbol{H}}$. As a result, the ILS problem is written as [19]

$$\begin{aligned} \underset{\boldsymbol{U}(k)\in\mathbb{U}}{\text{minimize}} \quad & ||\,\widetilde{\boldsymbol{U}}_{\text{unc}}(k) - \widetilde{\boldsymbol{H}}\widetilde{\boldsymbol{U}}(k)||_2^2\,, \\ \text{subject to} \quad & ||\boldsymbol{y}(k+1)||_2 \leq y_{\text{bnd}}\,, \end{aligned} \quad (9)$$

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

Since problem (9) is an integer optimization problem, branch-and-bound methods can be employed to solve it faster. One such an algorithm is the sphere decoding algorithm [15] that—as shown in [14] and [19]—appears to be computationally very efficient when employed to solve the long-horizon direct MPC problem. The sphere decoder computes a relatively tight upper bound from the beginning of the optimization procedure, and as a result only a few candidate solutions need to be evaluated in real time. Specifically, the set of candidate

solutions consists of these n-dimensional lattice points that are included in a hypersphere (n-dimensional sphere) of radius ρ centered at the unconstrained solution $\widehat{\boldsymbol{U}}_{\text{unc}}(k)$. To decide which of these points is the optimal solution, the algorithm traverses the generated search tree—consisting of branches that correspond to the elements of the sequence of control actions $\boldsymbol{U}$, and of m-dimensional nodes, $m = 1, \ldots, n$—in a depth-first search manner. The goal is to explore at least one complete branch of the tree and reach the bottom level in order to get a candidate solution; at that point, the radius of the hypersphere is updated (is getting smaller) resulting in a tighter sphere. To achieve this, the search procedure moves from the root node and the higher levels of the tree towards the lower levels and the one-dimensional (leaf) nodes. Every time a child node is visited branching occurs. On the other hand, when the leaf nodes or a dead end are reached backtracking occurs; then the algorithm moves to higher levels to examine unvisited nodes.

As mentioned before, the initial radius ρ of the sphere defines the upper bound of the search process. Therefore, the choice of the initial radius is crucial since the desired initial hypersphere includes the smallest possible (nonzero) number of lattice points (i.e., nodes). In [21], the initial radius is computed as

$$\rho(k) = \min\{\rho_1(k), \rho_2(k)\}\,, \tag{10}$$

with

$$\rho_1(k) = ||\,\widetilde{\boldsymbol{U}}_{\text{unc}}(k) - \widetilde{\boldsymbol{H}}\widetilde{\boldsymbol{U}}_{\text{bab}}(k)||_2\,, \tag{11}$$

and

$$\rho_2(k) = ||\,\widetilde{\boldsymbol{U}}_{\text{unc}}(k) - \widetilde{\boldsymbol{H}}\widetilde{\boldsymbol{U}}_{\text{ed}}(k)||_2\,. \tag{12}$$

Radius ρ_1 in (11) depends on $\widetilde{\boldsymbol{U}}_{\text{bab}}(k) = \boldsymbol{M}^{-1}\boldsymbol{U}_{\text{bab}}(k)$, where $\boldsymbol{U}_{\text{bab}}$, also referred to as the Babai estimate [26], [27], is the rounded unconstrained solution to the closest integer vector, i.e.,

$$\boldsymbol{U}_{\text{bab}}(k) = \lfloor \boldsymbol{H}^{-1}\bar{\boldsymbol{U}}_{\text{unc}}(k) \rceil = \lfloor \boldsymbol{U}_{\text{unc}}(k) \rceil\,, \tag{13}$$

On the other hand, as can be seen in (12), radius ρ_2 is a function of $\widetilde{\boldsymbol{U}}_{\text{ed}}(k) = \boldsymbol{M}^{-1}\boldsymbol{U}_{\text{ed}}(k)$, where $\boldsymbol{U}_{\text{ed}}$ is the previously applied solution $\boldsymbol{U}^*(k-1)$ shifted by one time step, i.e.,

$$\boldsymbol{U}_{\text{ed}}(k) = \begin{bmatrix} \boldsymbol{0} & \boldsymbol{I} & \boldsymbol{0} & \ldots & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{I} & \ddots & \vdots \\ \vdots & & \ddots & \ddots & \boldsymbol{0} \\ \boldsymbol{0} & \ldots & \ldots & \boldsymbol{0} & \boldsymbol{I} \\ \boldsymbol{0} & \ldots & \ldots & \boldsymbol{0} & \boldsymbol{I} \end{bmatrix} \boldsymbol{U}^*(k-1)\,. \tag{14}$$

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

IV. Solving the Constrained Integer Least-Squares Problem

However, in this work the computation of the initial radius ρ turns out to be more complicated because of the constraint (7d). The reason is that if the sphere is computed based on (10), then there is a possibility that the unconstrained solution and/or all candidate solutions included in the sphere violate (7d). To avoid this, the computation of the initial radius needs to be revised. This is done by translating the output constraint (7d) into an input constraint.

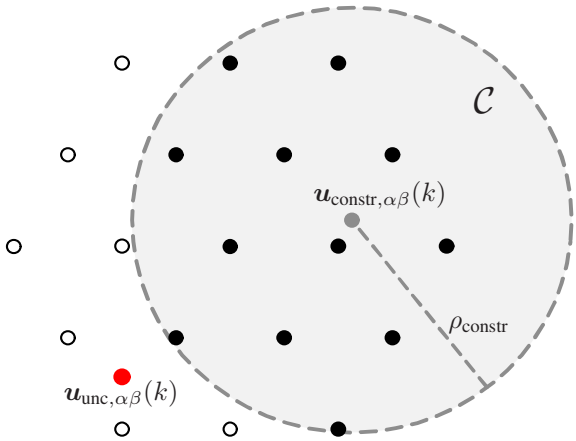


Fig. 4: Bounds on the control input in the $\alpha\beta$-plane (in this example it is assumed that the unconstrained solution is infeasible). The feasible points (i.e., those enclosed in $\mathcal{C}$, denoted by the lightly shaded circle area) are indicated with black solid circles. The infeasible points are shown as black circles.

By utilizing (4), the output constraint (7d) can be written as

$$||\boldsymbol{CAx}(k) + \boldsymbol{CBu}_{\alpha\beta}(k)||_2 \leq y_{\text{bnd}}\,, \tag{15}$$

with $\boldsymbol{u}_{\alpha\beta}(k) = \boldsymbol{Ku}(k)$. Noticing that the diagonal elements $\gamma_i,\ i = 1, 2$ of the matrix $\boldsymbol{CB} = \text{diag}(\gamma) \succ 0 \in \mathbb{R}^{2\times 2}$ are equal, (15) takes the form

$$\begin{aligned} &||\boldsymbol{CAx}(k) + \gamma\boldsymbol{I}\boldsymbol{u}_{\alpha\beta}(k)||_2 \leq y_{\text{bnd}} \Leftrightarrow \\ &\left|\left|\frac{\boldsymbol{CAx}(k)}{\gamma} + \boldsymbol{u}_{\alpha\beta}(k)\right|\right|_2 \leq \frac{y_{\text{bnd}}}{\gamma}\,, \end{aligned} \tag{16}$$

which is of the form

$$||\boldsymbol{u}_{\text{constr},\alpha\beta}(k) + \boldsymbol{u}_{\alpha\beta}(k)||_2 \leq \rho_{\text{constr}}\,, \tag{17}$$

i.e., a circle $\mathcal{C}$ centered at

$$\boldsymbol{u}_{\text{constr},\alpha\beta}(k) = -\frac{\boldsymbol{CAx}(k)}{\gamma} \tag{18}$$

and with the (constant) radius

$$\rho_{\text{constr}} = \frac{y_{\text{bnd}}}{\gamma}\,. \tag{19}$$

Hence, in order to meet the output constraint (7d), the control input (in the $\alpha\beta$-plane) should lie within the circle $\mathcal{C}$. The input constraint in the $\alpha\beta$-plane is shown in Fig. 4.

Based on the above, the feasible set of the integer-valued input vector $\boldsymbol{U}$ is defined as $\mathbb{U}_{\text{constr}} = \boldsymbol{\mathcal{U}}_{\text{constr}_1} \times \boldsymbol{\mathcal{U}}_{\text{constr}_2} \times \ldots \times \boldsymbol{\mathcal{U}}_{\text{constr}_N}$, with $\boldsymbol{\mathcal{U}}_{\text{constr}_i} = \boldsymbol{\mathcal{U}}$ for $i \in \{2, \ldots, N\}$ and

$$\boldsymbol{\mathcal{U}}_{\text{constr}_1} = \{\boldsymbol{u}(k) \mid \boldsymbol{Ku}(k) \in \mathcal{C}, \boldsymbol{u}(k) \in \boldsymbol{\mathcal{U}}\}\,.$$

Having defined the input feasible set, the initial radius ρ of the hypersphere needs revision. While on the one hand the radius should meet the same objective as in the unconstrained ILS-problem case, i.e., to be as small as possible, an additional

goal is that the new hypersphere should include at least one feasible point. To satisfy both criteria, the new radius is computed by writing (10) as

$$\rho(k) = \min\{\hat{\rho}_1(k), \hat{\rho}_2(k)\}\,. \tag{20}$$

In (20), $\hat{\rho}_1$ is defined as

$$\hat{\rho}_1(k) = ||\,\widetilde{\boldsymbol{U}}_{\text{unc}}(k) - \widetilde{\boldsymbol{H}}\widetilde{\boldsymbol{U}}_{\text{rnd}}(k)||_2\,, \tag{21}$$

where $\widetilde{\boldsymbol{U}}_{\text{rnd}}(k) = \boldsymbol{M}^{-1}\boldsymbol{U}_{\text{rnd}}(k)$, with the initial guess $\boldsymbol{U}_{\text{rnd}}(k) = [\boldsymbol{u}^T_{\text{rnd}}(k)\ \ldots\ \boldsymbol{u}^T_{\text{rnd}}(k+N-1)]^T$ given by

$$\boldsymbol{U}_{\text{rnd}}(k) = \begin{bmatrix} \boldsymbol{u}_{\text{feas}}(k) \\ \boldsymbol{0}_3 \\ \boldsymbol{0}_3 \\ \vdots \\ \boldsymbol{0}_3 \end{bmatrix} + \begin{bmatrix} \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} & \ddots & \boldsymbol{0} \\ \boldsymbol{I} & \boldsymbol{0} & \boldsymbol{0} & \ddots & \vdots \\ \boldsymbol{0} & \boldsymbol{I} & \boldsymbol{0} & \ddots & \vdots \\ \vdots & & \ddots & \ddots & \vdots \\ \boldsymbol{0} & \ldots & \ldots & \boldsymbol{I} & \boldsymbol{0} \end{bmatrix} \boldsymbol{U}_{\text{bab}}(k)\,, \tag{22}$$

where $\boldsymbol{0}_3$ is the three-dimensional zero vector. Hence, the intial guess $\boldsymbol{U}_{\text{rnd}}(k)$ is equal to the Babai estimate from time-step $k+1$ up to $k+N-1$, whereas the estimate at time-step k results by guessing the $\alpha\beta$-plane feasible candidate solution $\boldsymbol{u}_{\text{feas},\alpha\beta}(k) = \boldsymbol{K}\boldsymbol{u}_{\text{feas}}(k)$, with $\boldsymbol{u}_{\text{feas}}(k) \in \boldsymbol{\mathcal{U}}_{\text{constr}_1}$, closest to the unconstrained solution $\boldsymbol{u}_{\text{unc},\alpha\beta}(k) = \boldsymbol{K}\boldsymbol{u}_{\text{unc}}(k)$. For that, the following procedure is adopted.

Step 1: The vector

$$\boldsymbol{g}(k) = \boldsymbol{u}_{\text{constr},\alpha\beta}(k) - \boldsymbol{u}_{\text{unc},\alpha\beta}(k)$$

that spans the line passing through the unconstrained solution $\boldsymbol{u}_{\text{unc},\alpha\beta}(k)$ and the center of $\mathcal{C}$ is computed.

Step 2: The intersection points, $\boldsymbol{u}_{\text{int},\alpha\beta_1}$ and $\boldsymbol{u}_{\text{int},\alpha\beta_2}$, of the aforementioned line and $\mathcal{C}$ are calculated. The one closer to the unconstrained solution is given by

$$\boldsymbol{u}^*_{\text{int},\alpha\beta}(k) = \arg\min ||\boldsymbol{u}_{\text{int},\alpha\beta_i}(k) - \boldsymbol{u}_{\text{unc},\alpha\beta}(k)||_2\,,\ i = 1, 2\,.$$

Step 3: The feasible input in the $\alpha\beta$-plane closer to the intersection point $\boldsymbol{u}^*_{\text{int},\alpha\beta}(k)$ is found. This is done by examining the sign of the elements of $\boldsymbol{g}$, i.e., the direction of the gradient in each dimension of the space. Then, the appropriately scaled $\boldsymbol{u}^*_{\text{int},\alpha\beta}(k)$ is rounded to either the smallest following or the largest previous feasible integer value, depending on the direction of the gradient, i.e.,

$$u_{\text{feas}_i}(k) = \begin{cases} \lceil u_{\text{int}_i}(k) \rceil & \text{if } g_i \geq 0 \\ \lfloor u_{\text{int}_i}(k) \rfloor & \text{if } g_i < 0 \end{cases}\,, \tag{23}$$

for $i = 1, 2$. This step is visualized in Fig. 5.

Step 4: The rescaled $\alpha\beta$-plane candidate solution $\boldsymbol{u}_{\text{feas},\alpha\beta}(k)$ is mapped onto the—always feasible—three-phase switch position which is the initial guess at time-step k, $\boldsymbol{u}_{\text{feas}}(k)$.

Regarding radius $\hat{\rho}_2(k)$ in (20), it depends on the educated guess $\boldsymbol{U}_{\text{ed}}$ as defined in Section III. However, this radius is finite only when $\boldsymbol{U}_{\text{ed}}$ is feasible, i.e.,

$$\hat{\rho}_2(k) = \begin{cases} \rho_2(k) & \text{if } \boldsymbol{U}_{\text{ed}} \in \mathbb{U}_{\text{constr}} \\ \infty & \text{if } \boldsymbol{U}_{\text{ed}} \notin \mathbb{U}_{\text{constr}} \end{cases}\,. \tag{24}$$

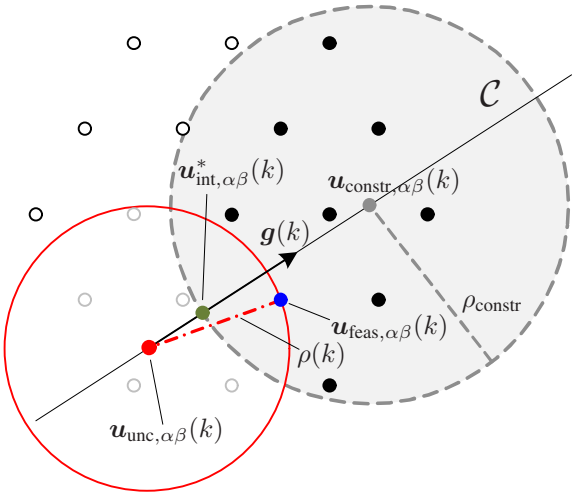


Fig. 5: Choice of the initial feasible guess in the $\alpha\beta$-plane. The gradient, as depicted by the (scaled-down) vector $\boldsymbol{g}(k)$, is positive in both dimensions. Consequently, the intersection point $\boldsymbol{u}^*_{\text{int},\alpha\beta}(k)$ (indicated as solid circle) closer to the unconstrained solution $\boldsymbol{u}_{\text{unc},\alpha\beta}(k)$ (shown as solid circle) is rounded up in both dimensions. Based on the feasible candidate solution $\boldsymbol{u}_{\text{feas},\alpha\beta}(k)$ (shown as solid circle) the radius ρ (dash-dotted line) of the hypersphere (shown as a circle) is determined. The infeasible points enclosed in the hypersphere are indicated with light gray circles, whereas the black circles are infeasible points that are not of concern. Finally, the feasible points that are not evaluated by the sphere decoder are shown with black solid circles.

After computing the feasible input set and the initial radius, the sphere decoder is called. The algorithm is the same as the one presented in [19], with the difference that for an n-dimensional point enclosed in the hypersphere to be considered as a candidate solution should be in the feasible input set $\mathbb{U}_{\text{constr}}$, see Algorithm 1. This implies that there might be points within the hypersphere that are infeasible, as shown in Fig. 5, and thus they should be discarded. This also becomes clear in Section V where an illustrative example is presented.

For the pseudocode of the Algorithm 1 the initial values of the arguments are $\widehat{\boldsymbol{U}} \leftarrow [\,]$, i.e., the empty vector, $\widetilde{\boldsymbol{U}}_{\text{unc}} \leftarrow \widetilde{\boldsymbol{H}}\ \boldsymbol{M}^{-1}\boldsymbol{U}_{\text{unc}}(k)$, $\rho \leftarrow \rho(k)$—see (20), $d \leftarrow 0$, and $j \leftarrow n$.

V. 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 inverter with the constant dc-link voltage $V_{\text{dc}} = 5.2\,\text{kV}$ and a fixed neutral point N is considered. For all cases examined, the controller was operated with the sampling interval $T_s = 25\,\mu\text{s}$. All results are presented in the p.u. system.

To provide meaningful insight into the operation of the algorithm, an illustrative example of one problem instance is presented here. The output (current) reference value is $||\boldsymbol{y}_{\text{ref}}||_2 \equiv ||\boldsymbol{i}_{s,\text{ref},\alpha\beta}||_2 = 1$, and the respective bound is set to $y_{\text{bnd}} \equiv i_{s,\text{bnd}} = 1.07$. Moreover, to simplify the exposition, the one-step horizon ($N = 1$) case is examined. Finally, the weighting factor λ_u is set equal to $4.8{\cdot}10^{-3}$ so that a switching

Algorithm 1 Sphere Decoder

function $\widetilde{\boldsymbol{U}}^*$ = ConstrSphDec($\widetilde{\boldsymbol{U}}$, $\widetilde{\boldsymbol{U}}_{\text{unc}}$, ρ^2, d^2, j)
  for each $\tilde{u} \in \mathcal{U}$ **do**
    $\widetilde{U}_j \leftarrow \tilde{u}$
    $d'^2 \leftarrow || \widetilde{U}_{\text{unc}_j} - \widetilde{\boldsymbol{H}}_{(j,j:n)} \widetilde{\boldsymbol{U}}_{j:n} ||_2^2 + d^2$
    if $d'^2 \leq \rho^2$ **then**
      if $j > 1$ **then**
        SphDec($\widetilde{\boldsymbol{U}}, d'^2, j-1, \rho^2, \widetilde{\boldsymbol{U}}_{\text{unc}}$)
      else
        if $\widetilde{\boldsymbol{U}} \in \mathbb{U}_{\text{constr}}$ **then**
          $\widetilde{\boldsymbol{U}}^* \leftarrow \widetilde{\boldsymbol{U}}$
          $\rho^2 \leftarrow d'^2$
        else
          SphDec($\widetilde{\boldsymbol{U}}, d'^2, j-1, \rho^2, \widetilde{\boldsymbol{U}}_{\text{unc}}$)
        end if
      end if
    end if
  end for
end function

frequency of about 150 Hz results in the case without the output constraint.

In Fig. 6(a) the case without the output constraint is shown in the $\alpha\beta$-plane. The transformed points are shown as black solid circles; these are the points that correspond to the input set $\mathbb{U}$ after applying the abc to $\alpha\beta$ transformation, see Section II. For example, the point closer to the top-left corner corresponds to the three-phase switch position $\boldsymbol{u} = [-1\ 1\ \ -1]^T$ in the original abc-plane, the point closer to the bottom-right corner to the switch position $\boldsymbol{u} = [1\ \ -1\ 1]^T$, and so on.

The unconstrained solution in the $\alpha\beta$-plane has the coordinates

$$\boldsymbol{u}_{\text{unc},\alpha\beta}(k) = [-0.7017\ \ -0.6780]^T .$$

This corresponds to the solution

$$\boldsymbol{u}_{\text{unc}}(k) = [-0.7017\ \ -0.2363\ 0.9380]^T$$

in the original abc-plane, and it is predicted to result in the stator current

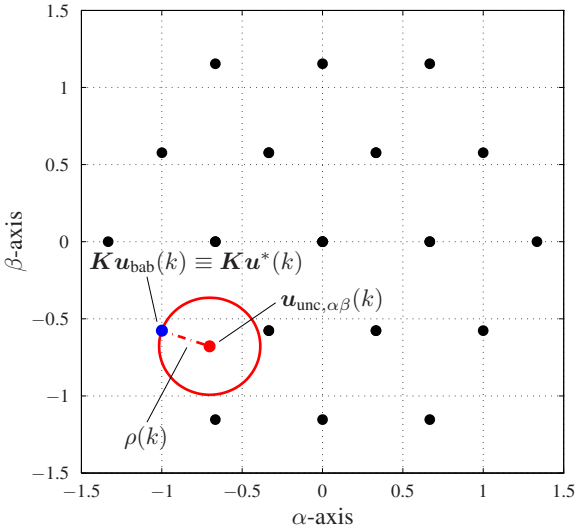
$$\boldsymbol{i}_{s,\alpha\beta}(k+1) = [-1.0645\ \ -0.1373]^T ,$$

of the Euclidean length

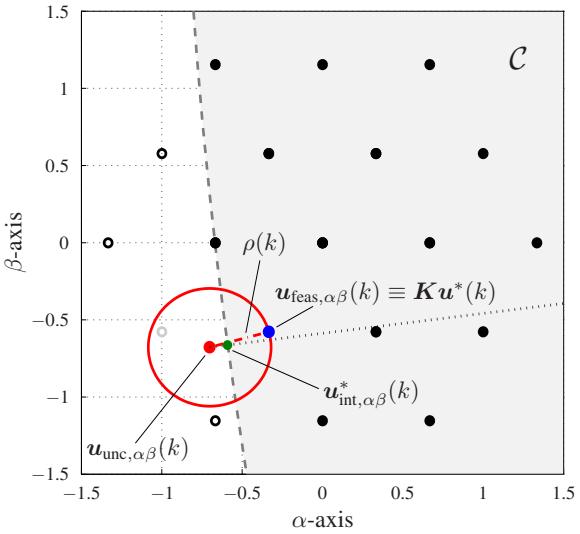
$$||\boldsymbol{i}_{s,\alpha\beta}(k+1)||_2 = 1.0734 .$$

This means that the unconstrained solution is infeasible. By rounding the unconstrained solution (i.e., by computing the Babai estimate $\boldsymbol{u}_{\text{bab}}(k)$) the initial radius[3] and the resulting sphere are computed. As can be seen, in this problem instance the optimal solution $\boldsymbol{u}^*(k)$ is equal to the initial guess since

[3]In this example the initial radius computed based on the Babai estimate is equal to the radius computed based on the educated guess $\boldsymbol{u}_{\text{ed}}(k)$, see ρ_2 in (12).



(a) Optimization problem without output constraint.



(b) Optimization problem with output constraint.

Fig. 6: Visualization of the sphere decoder in the $\alpha\beta$-plane for the horizon $N = 1$ case.

there are no other points in the sphere. It turns out that the applied solution in the original abc-plane is

$$\boldsymbol{u}^*(k) = [-1\ 0\ 1]^T$$

(notice that the Babai estimate shown in this figure corresponds to that point in the original abc-plane). This solution would result in the current

$$\boldsymbol{i}_{s,\alpha\beta}(k+1) = [-1.0734\ \ -0.1343]^T \Rightarrow$$
$$||\boldsymbol{i}_{s,\alpha\beta}(k+1)||_2 = 1.0818 ,$$

thus the constraint would be violated.

If, on the other hand, constraint (7d) is taken into account, the resulting $\alpha\beta$-plane looks like the one shown in Fig. 6(b). The unconstrained solution is the same. The feasible input set, computed according to the procedure described in Section IV, is (partly) shown as shaded (light gray) area; (an arc of) the

circle $\mathcal{C}$ that encloses that area is indicated with the (darker) gray dashed circle. The coordinates of the center of $\mathcal{C}$ are

$$\boldsymbol{u}_{\text{constr},\alpha\beta}(k) = [35.0985\ 3.9408]^T$$

and the radius is

$$\rho_{\text{constr}} = 35.9841\,.$$

As can be understood from the line segment (dotted line) that connects $\boldsymbol{u}_{\text{unc},\alpha\beta}(k)$ and $\boldsymbol{u}_{\text{constr},\alpha\beta}(k)$, vector $\boldsymbol{g}$ is elementwise positive, i.e., $\boldsymbol{g} \geq \mathbf{0}_2$, with $\mathbf{0}_2$ being the two-dimensional zero vector. This indicates that the intersection point

$$\boldsymbol{u}^*_{\text{int},\alpha\beta}(k) = [-0.5898\ \ -0.6636]^T$$

of the line segment and circle $\mathcal{C}$ should be rounded up in both dimensions. The feasible point that is closer to $\boldsymbol{u}^*_{\text{int},\alpha\beta}(k)$ has the coordinates

$$\boldsymbol{u}_{\text{feas},\alpha\beta}(k) = [-0.3333\ \ -0.5774]^T\,.$$

Mapping $\boldsymbol{u}_{\text{feas},\alpha\beta}(k)$ into the abc-plane, the three-phase switch position that serves as the initial guess is

$$\boldsymbol{u}_{\text{feas}}(k) = [0\ 0\ 1]^T\,.$$

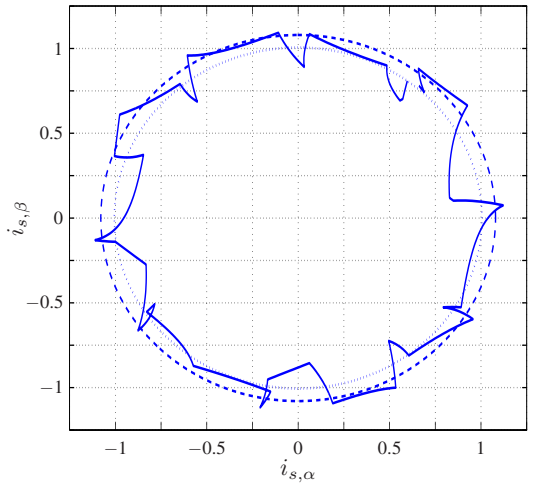
Based on that point the new radius ρ (see (20) and (21)) and the resulting sphere are computed. As can be seen, the initial guess is the optimal solution applied to the inverter; it produces the current

$$\begin{aligned}&\boldsymbol{i}_{s,\alpha\beta}(k+1) = [-1.0536\ \ -0.1343]^T \Rightarrow\\ &||\boldsymbol{i}_{s,\alpha\beta}(k+1)||_2 = 1.0621\,,\end{aligned}$$

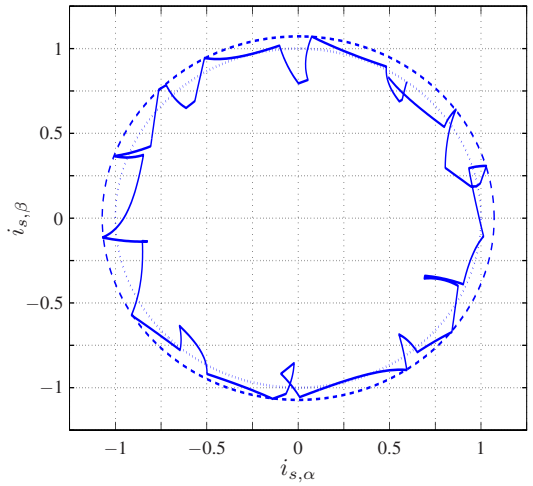
hence the constraint is met. It is worthwhile to mention that the computed hypersphere includes more than one point. However, the other point (indicated with a light gray circle) is infeasible and is thus not considered as a candidate solution.

In Fig. 7, the resulting stator current in the $\alpha\beta$-plane is shown over one fundamental period, along with its reference and bound. As can be seen, when the current constraint is not taken into account (Fig. 7(a)) the instantaneous current regularly reaches relatively high values, above the to-be-considered bound. On the other hand, when the constraint is included in the optimization problem, it is fully respected and the stator current always remains within the imposed bound, see Fig. 7(b).

Finally, the three-phase stator current waveforms produced by the unconstrained and constrained direct MPC algorithm are illustrated in Figs. 8(a) and 8(b), respectively. In both cases the currents accurately track their references, however, the ripples are higher when the output constraint is not taken into account. This results in a slightly higher stator current total harmonic distortion (THD) of 9.98% compared to the THD of 9.83% the constrained MPC produces. However, it should be pointed out that in the latter case the semiconductor switches are operated at a slightly higher frequency of approximately $192\,\text{Hz}$, compared to the switching frequency of about $150\,\text{Hz}$ obtained with the unconstrained MPC. This can be justified by the fact that the MPC scheme with the output constraint occasionally triggers switching transitions to keep the current within the desired range of values. This implies that the decisions the MPC algorithm makes when the output constraint is active would correspond to suboptimal solutions in the case where the constraint is not considered.



(a) Stator current $\boldsymbol{i}_{s,\alpha\beta}$ without output constraint.

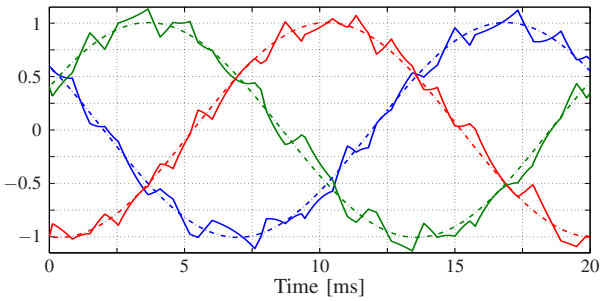


(b) Stator current $\boldsymbol{i}_{s,\alpha\beta}$ with output constraint.

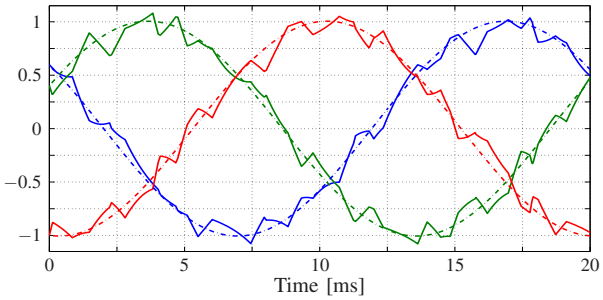
Fig. 7: Simulated waveforms of the stator current (solid line) in the $\alpha\beta$-plane produced by the direct model predictive controller with current reference tracking at steady-state operation, at full speed and rated torque. The current reference ($||\boldsymbol{i}_{s,\text{ref},\alpha\beta}||_2 = 1$) and bound ($i_{s,\text{bnd}} = 1.07$) are shown as dotted and dashed circles, respectively. A one-step horizon ($N = 1$) is used, the sampling interval is $T_s = 25\,\mu\text{s}$ and the weighting factor is $\lambda_u = 4.8 \cdot 10^{-3}$.

VI. Conclusions

This paper proposes an algorithm that translates the output constraints imposed on the variables of interest—in the form of safety constraints—into input constraints. Based on the redefined input feasible set, the computation of the hypersphere is refined. The new hypersphere includes at least one feasible point, thus ensuring that the solution set is always nonempty. Thanks to the proposed modifications, the sphere decoder can still perform effectively, while it manages to find the optimal solution that lies within the input feasible set. As a result, the imposed constraints are fully respected allowing



(a) Three-phase stator current $\boldsymbol{i}_s$ without output constraint. The switching frequency is approximately 150 Hz and the current THD is 9.98%.



(b) Three-phase stator current $\boldsymbol{i}_s$ with output constraint. The switching frequency is approximately 192 Hz and the current THD is 9.83%.

Fig. 8: Simulated waveforms of the three-phase stator current $\boldsymbol{i}_s$ (solid lines) and their references (dash-dotted lines) produced by the direct model predictive controller with current reference tracking at steady-state operation, at full speed and rated torque. The simulation parameters are the same as those in Fig. 7.

the plant to operate at its physical limits without deteriorating its performance.

Appendix

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

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

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

References

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

[2] 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.

[3] S. Kouro, P. Cortés, R. Vargas, U. Ammann, and J. Rodríguez, "Model predictive control—A simple and powerful method to control power converters," *IEEE Trans. Ind. Electron.*, vol. 56, no. 6, pp. 1826–1838, Jun. 2009.

[4] J. Rodríguez, M. P. Kazmierkowski, J. R. Espinoza, P. Zanchetta, H. Abu-Rub, H. A. Young, and C. A. Rojas, "State of the art of finite control set model predictive control in power electronics," *IEEE Trans. Ind. Informat.*, vol. 9, no. 2, pp. 1003–1016, May 2013.

[5] S. Vazquez, J. I. Leon, L. G. Franquelo, J. Rodríguez, H. A. Young, A. Marquez, and P. Zanchetta, "Model predictive control: A review of its applications in power electronics," *IEEE Ind. Electron. Mag.*, vol. 8, no. 1, pp. 16–31, Mar. 2014.

[6] T. Geyer, "Low complexity model predictive control in power electronics and power systems," Ph.D. dissertation, Autom. Control Lab. ETH Zurich, Zurich, Switzerland, 2005.

[7] P. Karamanakos, "Model predictive control strategies for power electronics converters and ac drives," Ph.D. dissertation, Elect. Mach. and Power Electron. Lab. NTU Athens, Athens, Greece, 2013.

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

[9] M. Morari and J. H. Lee, "Model predictive control: Past, present and future," *Comput. and Chemical Eng.*, vol. 23, no. 4, pp. 667–682, May 1999.

[10] 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.

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

[12] 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.

[13] 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.

[14] ——, "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.

[15] 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.

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

[17] P. Karamanakos, T. Geyer, and R. Kennel, "Computationally efficient optimization algorithms for model predictive control of linear systems with integer inputs," in *Proc. IEEE Conf. Decis. Control*, Osaka, Japan, Dec. 2015, pp. 3663–3668.

[18] ——, "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.

[19] ——, "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.

[20] R. P. Aguilera, R. Baidya, P. Acuna, S. Vazquez, T. Mouton, and V. G. Agelidis, "Model predictive control of cascaded H-bridge inverters based on a fast-optimization algorithm," in *Proc. IEEE Ind. Electron. Conf.*, Yokohama, Japan, Nov. 2015, pp. 4003–4008.

[21] 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.

[22] ——, "Computationally efficient sphere decoding for long-horizon direct model predictive control," in *Proc. IEEE Energy Convers. Congr. Expo.*, Milwaukee, WI, Sep. 2016.

[23] 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.

[24] 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.

[25] 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.

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

[27] M. Grötschel, L. Lovász, and A. Schrijver, *Geometric Algorithms and Combinatorial Optimization*, 2nd ed. New York: Springer-Verlag, 1993.