

Constrained Long-Horizon Direct Model Predictive Control for Synchronous Reluctance Motor Drives

L. Ortombina*, E. Liegmann†, P. Karamanakos‡, F. Tinazzi*, M. Zigliotto*, R. Kennel†

* University of Padova, Dept. of Management and Engineering, 36100 Vicenza, Italy
ludovico.ortombina@phd.unipd.it {mauro.zigliotto,fabio.tinazzi}@unipd.it

† Technical University Munich, Chair of Electrical Drive Systems and Power Electronics, 80333 Munich, Germany
eyke.liegmann@tum.de ralph.kennel@tum.de

‡ Tampere University of Technology, Faculty of Computing and Electrical Engineering, 33101 Tampere, Finland
p.karamanakos@ieee.org

Abstract—A finite control set model predictive control strategy for the control of the stator currents of a synchronous reluctance motor driven by a three-level neutral point clamped inverter is presented in this paper. The presented algorithm minimizes the stator current distortions while operating the drive system at switching frequencies of a few hundred Hertz. Moreover, the power electronic converter is protected by overcurrents and/or overvoltages owing to a hard constraint imposed on the stator currents. To efficiently solve the underlying integer nonlinear optimization problem a sphere decoding algorithm serves as optimizer. To this end, a numerical calculation of the unconstrained solution of the optimization problem is proposed, along with modifications in the algorithm proposed in [1] so as to meet the above-mentioned control objectives. Simulation results show the effectiveness of the proposed control algorithm.

I. INTRODUCTION

The permanent magnet (PM) price crisis in 2010 [2] pushed academic and industrial interests about synchronous reluctance (SynR) motor drives. The main advantages of SynR motors are their inherent high efficiency, high dynamic performance and low manufacturing and maintenance cost. The drawback of SynR motors is the nonlinear relation between currents and flux linkages, i.e., their magnetic model. To achieve good performance and to fully exploit the capabilities of SynR motors, an accurate identification of the magnetic model is needed [3]. Consequently, if the relation between stator currents and flux linkages is known, nonlinear control techniques are more effective and, thus, preferable.

Model predictive control (MPC) [4] is a constrained optimal control method that can deal with nonlinear dynamics as well as multiple control objectives in one computational stage. This is achieved by introducing a cost function, i.e., a performance index, that maps the control objectives into a scalar which quantifies the system performance for a given control input. To find the optimal control action, the cost function is minimized while being subjected to constraints imposed on the variables of interest (e.g., stator current). With respect to more conventional proportional-integral-derivative (PID) or linear-quadratic (LQ) controllers, MPC can easily incorporate soft

and hard constraints on the state variables, giving the design a more systemic approach. Motivated by these advantages of MPC, a cascaded MPC scheme was proposed to control the stator currents of a SynR motor in [5].

MPC in power electronics is frequently implemented as a direct control method, i.e., a modulator is not required. This yields faster dynamics and intuitive implementation. Nevertheless, the computational complexity of that kind of MPC algorithms, commonly referred to as finite control set MPC (FCS-MPC), grows exponentially with the length of the prediction horizon, making the brute-force approach of exhaustive search infeasible when long horizons are employed.

However, as shown in e.g., [6], long horizons have positive effect on the system performance. Therefore, research towards strategies that can make the optimization problem computationally tractable is of interest. For example, in [1] a heuristic branch-and-bound algorithm, namely the sphere decoding algorithm (SDA) [7], used in, e.g., communications and cryptography, is utilized for power electronic systems. Specifically, based on the unconstrained solution of the optimization problem (that is computed analytically) one can formulate the optimization problem as an integer least-squares one. Subsequently, the SDA performs an efficient search for the best—integer-valued—switch position (i.e., the control input) to be applied to the converter at the next control period.

It is worthwhile to note that the approach described in [1] was developed for an induction motor drive, i.e., a linear isotropic system. As a key-feature, this paper extends the SDA principle to a SynR motor including its nonlinear magnetic model. Moreover, a *constrained* optimization problem is proposed by imposing limitations on the stator current so as to protect the inverter. Such a constraint, akin to that proposed in [8] for the (linear) induction motor drive system, is properly formulated to tackle the nonlinear dynamics of the SynR motor drive.

II. PHYSICAL MODEL OF THE MOTOR DRIVE

The system consists of a three-level neutral point clamped (3L-NPC) voltage source inverter, driving a SynR motor, as shown in Fig. 1. It is assumed that the dc-link voltage V_{dc}

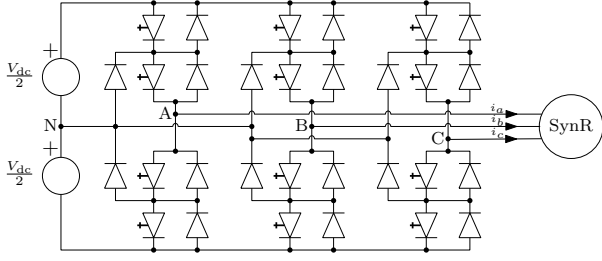


Fig. 1. Three-level NPC voltage source inverter with a synchronous reluctance motor. The inverter has a constant neutral point potential.

is constant and the neutral point potential v_N is fixed to zero. Moreover, variables in the three-phase (abc) system $\zeta^{abc} = [\zeta^a \zeta^b \zeta^c]^T$ are transformed into independent (orthogonal) dc variables $\zeta^{dq} = [\zeta^d \zeta^q]^T$ in the dq plane—which rotates with angular speed ω_{me} via $\zeta^{dq} = \mathbf{T}(\vartheta)\zeta^{abc}$, where

$$\mathbf{T}(\vartheta) = \frac{2}{3} \begin{bmatrix} \cos(\vartheta) & \cos(\vartheta - \frac{2}{3}\pi) & \cos(\vartheta - \frac{4}{3}\pi) \\ -\sin(\vartheta) & -\sin(\vartheta - \frac{2}{3}\pi) & -\sin(\vartheta - \frac{4}{3}\pi) \end{bmatrix} \quad (1)$$

and ϑ is the angle between the d and a axis. For the inverse mapping, i.e., from the dq to the abc plane, the matrix $\tilde{\mathbf{T}}(\vartheta) = 3/2\mathbf{T}^T(\vartheta)$ is employed. Thus, the operation $\zeta^{abc} = \tilde{\mathbf{T}}(\vartheta)\zeta^{dq}$ is performed.

A. Model of the inverter

The *single-phase* switch position of a 3L-NPC is modeled with the integer variables $u^a, u^b, u^c \in \mathcal{U} \triangleq \{-1, 0, 1\}$. Each leg of the 3L-NPC can output a phase voltage of $-V_{dc}/2$, 0 , $V_{dc}/2$. Introducing the vector $\mathbf{u}^{abc} = [u^a \ u^b \ u^c]^T \in \mathcal{U} = \mathcal{U} \times \mathcal{U} \times \mathcal{U}$ to denote the *three-phase* switch position, the output voltage of a three-level inverter in the three-phase (abc) frame can be expressed as

$$\mathbf{v}^{abc} = [v^a \ v^b \ v^c]^T = \frac{V_{dc}}{2} \mathbf{u}^{abc}. \quad (2)$$

B. Model of the motor

Most of MPC schemes rely on an accurate model of the system in order to determine the future control action. It is fundamental to find a trade-off among the accuracy and the complexity of the model, to make the state prediction feasible in real time.

The voltage equation of the SynR motor in the dq rotating frame is

$$\frac{d\boldsymbol{\lambda}^{dq}}{dt} = \mathbf{v}^{dq} - R\mathbf{i}^{dq} - j\omega_{me}\boldsymbol{\lambda}^{dq} \quad (3)$$

where $\boldsymbol{\lambda}^{dq} = \lambda^d + j\lambda^q$, $\mathbf{i}^{dq} = i^d + ji^q$, and $\mathbf{v}^{dq} = v^d + jv^q$ are the flux linkage, current and voltage space vectors, respectively, R is the stator resistance, and $\omega_{me} = p\omega_m$ is the electromechanical speed, with p denoting the number of motor pole pairs and ω_m is the mechanical rotor speed. To simplify the analysis, only the saturation effects are taken into account and the magnetic cross-coupling is neglected. The two flux

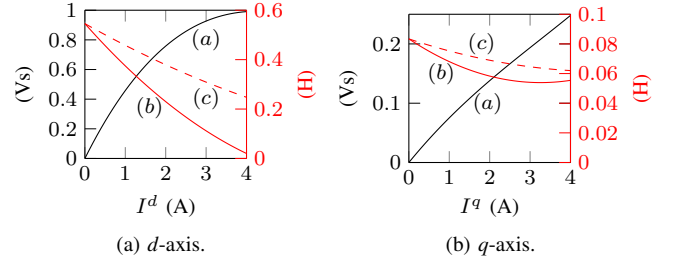


Fig. 2. Magnetic maps of a SynR motor: (a) flux linkages; (b) differential and (c) apparent inductances.

components λ^d and λ^q are modeled as two nonlinear functions, each depending on the relative axis coordinate, i.e.,

$$\lambda^d = f^d(i^d), \quad \lambda^q = f^q(i^q). \quad (4)$$

The two functions (4) and the associated differential and apparent inductances are shown in Fig. 2. In the following, since the mechanical dynamics are much slower than the electrical ones, a constant motor speed is assumed during a control cycle.

The implementation of the control strategy requires the discretization of the continuous model. In this paper, the method developed in [5] for a SynR motor is adopted, as a trade-off between accuracy and complexity. Let ξ_k indicate the sampled version of a generic continuous variable $\xi(t)$ at the sampling instant $t = kT_s$, with T_s the sampling interval. Introducing the average operator

$$\langle \xi \rangle_k \triangleq \frac{1}{T_s} \int_{kT_s}^{(k+1)T_s} \xi(t) dt, \quad (5)$$

the integration of (3) over the interval $[kT_s, (k+1)T_s]$ yields

$$\boldsymbol{\lambda}_{k+1}^{dq} = \boldsymbol{\lambda}_k^{dq} + T_s \langle \mathbf{v} \rangle_k^{dq} - RT_s \langle \mathbf{i} \rangle_k^{dq} - jT_s \langle \omega_{me} \boldsymbol{\lambda} \rangle_k^{dq}. \quad (6)$$

Recalling the constant speed assumption over a sampling period, imposing a constant voltage vector \mathbf{v}^{dq} over the whole sampling period and assuming that

$$\langle \mathbf{i} \rangle_k^{dq} \approx \frac{\mathbf{i}_{k+1}^{dq} + \mathbf{i}_k^{dq}}{2}, \quad \langle \omega_{me} \boldsymbol{\lambda} \rangle_k^{dq} \approx \omega_{me} \frac{\boldsymbol{\lambda}_{k+1}^{dq} + \boldsymbol{\lambda}_k^{dq}}{2}, \quad (7)$$

expression (6) can be further simplified to

$$\boldsymbol{\lambda}_{k+1}^{dq} \approx \boldsymbol{\lambda}_k^{dq} + T_s \mathbf{v}_k^{dq} - RT_s \frac{\mathbf{i}_{k+1}^{dq} + \mathbf{i}_k^{dq}}{2} - j\omega_{me} T_s \frac{\boldsymbol{\lambda}_{k+1}^{dq} + \boldsymbol{\lambda}_k^{dq}}{2}. \quad (8)$$

Given the nonlinear relation between the current vector \mathbf{i}_{k+1}^{dq} and the flux linkage $\boldsymbol{\lambda}_{k+1}^{dq}$, (8) can be solved only numerically, which is undesirable when real-time applications are of concern. However, accepting that the average current $\langle \mathbf{i} \rangle_k^{dq}$ in (8) is approximated by its instantaneous value \mathbf{i}_k^{dq} (due to the small sampling interval), the following closed-form solution for $\boldsymbol{\lambda}_{k+1}^{dq}$ can be obtained

$$\boldsymbol{\lambda}_{k+1}^{dq} \approx \frac{1 - j\omega_{me}T_s/2}{1 + j\omega_{me}T_s/2} \boldsymbol{\lambda}_k^{dq} + T_s \frac{\mathbf{v}_k^{dq} - R\mathbf{i}_k^{dq}}{1 + j\omega_{me}T_s/2}. \quad (9)$$

When devising MPC algorithms it is useful to have the model in its state-space representation [4]. Defining the state vector as $\mathbf{x}_k = [\lambda_k^d \lambda_k^q]^T$ and the input vector as $\mathbf{v}_k^{dq} = [v_k^d v_k^q]^T$, (9) can be rewritten as

$$\mathbf{x}_{k+1} = \mathbf{A}_k \mathbf{x}_k + \tilde{\mathbf{B}} \mathbf{v}_k^{dq}, \quad (10)$$

where the time-varying state matrix is

$$\mathbf{A}_k = \frac{1}{1 + M^2} \begin{bmatrix} 1 - M^2 - \frac{RT_s}{L_k^{d,\text{app}}} & 2M - \frac{RT_s M}{L_k^{q,\text{app}}} \\ -2M + \frac{RT_s M}{L_k^{d,\text{app}}} & 1 - M^2 - \frac{RT_s}{L_k^{q,\text{app}}} \end{bmatrix}, \quad (11)$$

and the input matrix is

$$\tilde{\mathbf{B}} = \frac{1}{1 + M^2} \begin{bmatrix} T_s & MT_s \\ -MT_s & T_s \end{bmatrix} \quad (12)$$

where $M = T_s \omega_{\text{me}}/2$. Moreover, $L_k^{d,\text{app}}$ and $L_k^{q,\text{app}}$ are the d and q apparent inductances computed at the current \mathbf{i}_k^{dq} . The time-varying output matrix \mathbf{C}_k is defined as

$$\mathbf{C}_k = \begin{bmatrix} \frac{1}{L_k^{d,\text{app}}} & 0 \\ 0 & \frac{1}{L_k^{q,\text{app}}} \end{bmatrix} \quad (13)$$

and the output equation is

$$\mathbf{y}_k = \mathbf{C}_k \mathbf{x}_k \quad (14)$$

with the stator current in the dq frame forming the output vector, i.e., $\mathbf{y}_k = \mathbf{i}_k^{dq}$.

The aim of this paper is to find the optimal switch position to be applied to the inverter, i.e., the $\mathbf{u}^{\text{opt},abc}$. Therefore, the model of the drive needs to be revised to reflect this. To this aim, (10) is written as

$$\mathbf{x}_{k+1} = \mathbf{A}_k \mathbf{x}_k + \tilde{\mathbf{B}} \mathbf{T}(\vartheta) \mathbf{v}_k^{abc} = \mathbf{A}_k \mathbf{x}_k + \mathbf{B}_k \mathbf{u}_k^{abc} \quad (15)$$

where the input matrix $\mathbf{B}_k = \tilde{\mathbf{B}} \mathbf{T}(\vartheta) V_{\text{dc}}/2$ becomes time-varying. With the proposed modeling the features of the drive model in the dq frame, i.e., a magnetic map independent of the position, are still present, while, at the same time, the three-phase switch position is modeled in a straightforward manner. Moreover, equations (13) and (15) represent a state model with the input expressed in the abc reference frame and state and output variables written in the rotating (dq) reference frame. This simplifies the derivation of the reference currents.

III. DIRECT MODEL PREDICTIVE CONTROL WITH REFERENCE TRACKING

As mentioned before, direct MPC is a control paradigm in which the switching signals are directly generated by the controller and fed into the inverter, i.e., without the interposition of a modulator. Each possible three-phase switch position is evaluated by the controller on the basis of the subsequent system performance, as quantified by a scalar performance index. The control action (i.e., the sequence of switch positions) that yields the minimum cost of the chosen performance index is selected as optimal. As can be understood, the definition of the performance index, also known as *cost function*, is one of the

key-elements of an MPC algorithm and it is formulated based on the control objectives. For the system under investigation, the control tasks are the following:

- the flux linkage λ^{dq} should track its reference λ^* accurately;
- the switching frequency has to be minimized;
- the inverter should be protected from overcurrents.

With regards to the first objective, by directly controlling flux linkage the stator current is indirectly controlled, as can be seen from (4). It is worth remembering that in a synchronous motor the relation among flux linkages and stator currents is a static mapping [3], and not a first-order dynamic relation as in induction motors [6]. Therefore, the flux is chosen to be controlled because this allows for the mathematical manipulations required when deriving the integer least-squares (ILS) problem in Section IV.

At time-step k , the first two control objectives are mapped into the cost function

$$J_k = \sum_{l=k}^{k+N-1} \left\| \lambda_k^*(l+1) - \lambda^{dq}(l+1) \right\|_{\mathbf{Q}}^2 + \left\| \Delta \mathbf{u}_k^{abc}(l) \right\|^2 \quad (16)$$

which penalizes the variables of interest over a finite prediction horizon of N time steps. The λ_k^* is the flux linkage reference trajectory and the term $\Delta \mathbf{u}^{abc}(l) = \mathbf{u}^{abc}(l) - \mathbf{u}^{abc}(l-1)$ accounts for the number of switching transitions involved between two consecutive time steps. By minimizing the number of transitions the switching losses (switching frequency) can be minimized as well. Matrix $\mathbf{Q} \in \mathbb{R}^{2 \times 2}$ is a diagonal, positive definite weighting matrix. Note that $\|\xi\|_{\mathbf{Q}}^2 \triangleq \xi^T \mathbf{Q} \xi$ denotes the squared norm of the vector ξ weighted by the matrix \mathbf{Q} . The choice of the diagonal entries of the weighting matrix sets the trade-off between the tracking accuracy of each flux linkage and the switching losses. In this work, all the diagonal terms of \mathbf{Q} are the same, meaning that the flux linkage tracking errors are equally penalized. In principle, it is possible to assign different weights. For instance, it could be interesting to track accurately the flux λ^d , as it is the main responsible for the torque response.

As listed above, the protection of the inverter from overcurrents has to be considered. This can be realized by imposing a constraint on the stator current as follows

$$\left\| \mathbf{i}_{k+1}^{dq} \right\|_2 \leq i_{\text{bnd}}, \quad (17)$$

where the positive scalar $i_{\text{bnd}} \in \mathbb{R}^+$ defines the boundary value of the stator current, as proposed in [8].

To find the optimal sequence of control actions (i.e., switch positions) at time-step k , the cost function (16) has to be minimized with respect to the so-called optimization variable, i.e., the switch sequence $\mathbf{U}_k = [\mathbf{u}_k^{abc,T} \mathbf{u}_{k+1}^{abc,T} \dots \mathbf{u}_{k+N-1}^{abc,T}]^T$, over the horizon N . The following problem needs to be solved in real time

$$\mathbf{U}_k^{\text{opt}} = \arg \underset{\mathbb{U}}{\text{minimize}} J_k \quad (18a)$$

$$\text{subject to } \mathbf{U}_k \in \mathbb{U} \quad (18b)$$

$$\|\Delta \mathbf{u}^{abc}(l)\|_{\infty} \leq 1, \forall l = k, \dots, k + N - 1 \quad (18c)$$

$$\|\mathbf{i}_{k+1}^{dq}\|_2 \leq i_{\text{bnd}}. \quad (18d)$$

with $\mathbb{U} = \mathbf{U} \times \dots \times \mathbf{U}$ being the N -times Cartesian product of the set \mathbf{U} , representing the feasible input set. Note that constraint (18c) allows to avoid a shoot-through in the inverter. According to the *receding horizon* principle [4], only the first element $\mathbf{u}_k^{abc,T}$ of the sequence $\mathbf{U}_k^{\text{opt}}$ is applied to the system, whereas the rest elements are discarded. Following, at step $k+1$ the whole procedure is repeated over a shifted prediction horizon after acquiring new measurements and/or estimates.

IV. INTEGER LEAST-SQUARES PROBLEM

The integer optimization problem (18) is clearly nonlinear. However, since its form somewhat resembles the (linear) problem solved in [1]—motivated by [9], [10] where a nonlinear problem was transformed to a linear one—we aim to reformulate it so as to solve it with the SDA. This algorithm is a smart branch-and-bound method which can find the solution of the long-horizon direct MPC problem in a computationally efficient manner [11]. The underlying idea is that the integer solution of the direct MPC problem is in the hypersphere (n -dimensional sphere) of radius ρ_k centered at the *unconstrained* solution. This allows the a-priori exclusion of several candidate switch positions, thus, effectively reducing the number of possible solutions to be tested in real time. Only those sequences the elements of which correspond to nodes that are inside the hypersphere are evaluated.

In brief, the main goal is to explore at least one complete branch, i.e., a complete switch sequence \mathbf{U}_k , from the root of the tree to the last node (leaf). Every time a complete branch is visited, the radius of the hypersphere gets smaller and the procedure is repeated as long as all possible paths to the leaf nodes are explored. By doing so, the optimal switch sequence—that leads to the most desirable system behavior—is found. To speed up the search process, the initial radius ρ_k of the hypersphere should be carefully chosen; it should be large enough in order to contain at least one complete switch sequence, but sufficiently small to reduce the number of candidate solutions that need to be evaluated. Moreover, when computing the radius the current constraint (17) should be taken into account. This makes the initial choice of ρ_k more challenging since all switch positions that result in a current that exceeds its respective limit i_{bnd} should be discarded from the feasible set. To achieve this, the initial radius is defined according to [8]. For more details on the operation of the SDA, the reader is referred to [1] and [11].

The extension of the SDA to a nonlinear system can be done starting from the linear formulation [1]. Given the un-

constrained solution $\mathbf{U}_k^{\text{unc}}$ (see Section IV-A), the cost function can be rewritten as follows

$$J_k = (\mathbf{U}_k - \mathbf{U}_k^{\text{unc}})^T \mathbf{H}_k (\mathbf{U}_k - \mathbf{U}_k^{\text{unc}}), \quad (19)$$

where \mathbf{H}_k is symmetric, positive definite matrix defined in Section IV-A. A unique invertible and lower triangular matrix $\mathbf{V}_k \in \mathbb{R}^{3N \times 3N}$ exists and satisfies $\mathbf{V}_k^T \mathbf{V}_k = \mathbf{H}_k$. The problem (18) can be rewritten as the equivalent constrained ILS problem

$$\mathbf{U}_k^{\text{opt}} = \arg \underset{\mathbb{U}}{\text{minimize}} \|\bar{\mathbf{U}}_k^{\text{unc}} - \mathbf{V}_k \mathbf{U}_k\|_2^2 \quad (20a)$$

$$\text{subject to } \mathbf{U}_k \in \mathbb{U} \quad (20b)$$

$$\|\Delta \mathbf{u}^{abc}(l)\|_{\infty} \leq 1, \forall l = k, \dots, k + N - 1 \quad (20c)$$

$$\|\mathbf{i}_{k+1}^{dq}\|_2 \leq i_{\text{bnd}}. \quad (20d)$$

where $\bar{\mathbf{U}}_k^{\text{unc}} = \mathbf{V}_k \mathbf{U}_k^{\text{unc}}$. Note that in order to extend the SDA to the nonlinear system under examination, a linearization of the system along the unconstrained solution has been accomplished, see Section IV-A.

A. Unconstrained solution

As mentioned, (18) represents a nonlinear quadratic function, the minimum of which cannot be computed by a closed-form expression. As an alternative, in this paper the Gauss-Newton algorithm (GNA) [12] was adopted to find the unconstrained solution of the cost function, by first ignoring the constraints (18b)–(18d). The GNA can only be used to minimize a sum of squared function values, but it has the advantage that the second derivatives, which can be challenging to compute, are not required. The iterative algorithm is of the form

$$\mathbf{U}_k^{\text{unc}}(z+1) = \mathbf{U}_k^{\text{unc}}(z) - [\mathbf{J}(z)^T \mathbf{J}(z)]^{-1} \mathbf{J}(z)^T \boldsymbol{\epsilon}(z) \quad (21)$$

$$z \in \{0, \dots, Z\},$$

where $Z \in \mathbb{N}$ is the number of iterations needed to find the global minimum with the given accuracy. $\mathbf{U}_k^{\text{unc}}(z)$ denotes the (real-valued) switch position at iteration z , while $\mathbf{J}(z)$ is the Jacobian matrix of the problem and $\boldsymbol{\epsilon}(z)$ is the error vector which has the same arguments as the cost function, i.e., is a function of the flux linkage and the switching effort. It should be pointed out that both $\mathbf{J}(z)$ and $\boldsymbol{\epsilon}(z)$ are calculated with respect to $\mathbf{U}_k^{\text{unc}}(z)$.

Given the error vector, and by splitting the two terms in (16), the following expression is obtained

$$J_k = \sum_{l=k}^{k+N-1} \left\| \boldsymbol{\lambda}_k^*(l+1) - \boldsymbol{\lambda}^{dq}(l+1) \right\|_{\mathbf{Q}}^2 + \sum_{l=k}^{k+N-1} \|\Delta \mathbf{u}^{abc}(l)\|^2 = \boldsymbol{\epsilon}^{dq,T} \boldsymbol{\epsilon}^{dq} + \boldsymbol{\epsilon}^{\mathbf{u}^{abc},T} \boldsymbol{\epsilon}^{\mathbf{u}^{abc}} = \boldsymbol{\epsilon}^T \boldsymbol{\epsilon}, \quad (22)$$

where $\boldsymbol{\epsilon} = [\boldsymbol{\epsilon}^{dq,T} \boldsymbol{\epsilon}^{\mathbf{u}^{abc},T}]^T \in \mathbb{R}^{5N \times 1}$ is the error vector; its structure facilitates the GNA execution.

The minimization procedure (21) has to be computed at each time step. Therefore, it needs to be as computationally efficient as possible. To this end, the previously computed unconstrained solution $\mathbf{U}_{k-1}^{\text{unc}}$ is utilized according to the receding horizon principle of MPC. It is reasonable to assume that the optimal solution at step k will be close to that at step $k-1$, at least at steady-state conditions. Therefore, at each step the GNA is initialized using the previous solution and the system is linearized along the same trajectory. The initial guess of $\mathbf{U}_k^{\text{unc}}(0)$ can be obtained by shifting \mathbf{U}_{k-1} by one time step and repeating the last switch position

$$\mathbf{U}_k^{\text{unc}}(0) = \mathbf{R}\mathbf{U}_{k-1} \quad (23)$$

where the matrix \mathbf{R} is defined in the appendix. Given the guess for the initial input, it is possible to calculate the associated initial flux linkage trajectory $\lambda_k^{dq}(0)$ using (15) and (23). The initial error vector $\epsilon(0)$ can be computed as

$$\epsilon(0) = \begin{bmatrix} \sqrt{\tilde{\mathbf{Q}}} (\lambda_k^* - \lambda_k^{dq}(0)) \\ \mathbf{U}_k^{\text{unc}}(0) - \mathbf{U}_{k-1} \end{bmatrix} = \begin{bmatrix} \sqrt{\tilde{\mathbf{Q}}} (\lambda_k^* - \lambda_k^{dq}(0)) \\ -\mathbf{S}^T \mathbf{U}_{k-1} \end{bmatrix} \quad (24)$$

where $\tilde{\mathbf{Q}} \in \mathbb{R}^{2N \times 2N}$ is a block diagonal matrix $\tilde{\mathbf{Q}} = \text{diag}(\mathbf{Q}, \dots, \mathbf{Q})$ and the time-invariant matrix \mathbf{S} is defined in the appendix.

Matrix $\mathbf{J}(z) \in \mathbb{R}^{5N \times 3N}$ is the Jacobian matrix of function (16), calculated along the trajectory $\lambda_k^{dq}(z)$. The entries of $\mathbf{J}(z)$ are defined as the derivative of each element of ϵ with respect to each element of $\mathbf{U}_k^{\text{unc}}(z)$. Thanks to the structure of the vector error ϵ , the Jacobian has the following advantageous structure

$$\mathbf{J}(z) = \begin{bmatrix} \mathbf{\Upsilon}(z) \\ \mathbf{S} \end{bmatrix}, \quad (25)$$

In addition, the time-varying matrix $\mathbf{\Upsilon}(z)$ is a lower triangular block matrix owing to the fact that the prediction error at step n is independent of the input at step m , with $m > n$. For this reason, $\mathbf{\Upsilon}(z)$ can be defined as

$$\mathbf{\Upsilon}(z) = \begin{cases} \begin{bmatrix} \sqrt{\tilde{\mathbf{Q}}} \begin{bmatrix} \frac{\partial e_m^d}{\partial u_n^a} & \frac{\partial e_m^d}{\partial u_n^b} & \frac{\partial e_m^d}{\partial u_n^c} \\ \frac{\partial e_m^q}{\partial u_n^a} & \frac{\partial e_m^q}{\partial u_n^b} & \frac{\partial e_m^q}{\partial u_n^c} \end{bmatrix} & \text{if } n \leq m \\ \mathbf{0}_{2,3} & \text{otherwise} \end{cases} \quad (26)$$

where $m, n = 1, \dots, N$. It is worth remembering that matrix $\mathbf{J}(z)$ and the error vector $\epsilon(z)$ need to be recalculated at each iteration z until the error vector becomes smaller than a predetermined threshold. The (real-valued) switch position that minimizes the error vector is the so-called unconstrained solution $\mathbf{U}_k^{\text{unc}}$.

After the preprocessing stage is over, i.e., matrix $\mathbf{\Upsilon}_k$ and the unconstrained solution $\mathbf{U}_k^{\text{unc}}$ are computed, matrix \mathbf{H}_k —introduced in (19)—is computed according to the following expression

$$\mathbf{H}_k = \mathbf{\Upsilon}_k^T \tilde{\mathbf{Q}} \mathbf{\Upsilon}_k + \mathbf{S}^T \mathbf{S}. \quad (27)$$

B. Hypersphere radius computation

The initial radius ρ_k of the hypersphere defines the upper bound of the search process. As mentioned above, a good upper bound should guarantee that problem (20) is feasible as well as that SDA can terminate (if possible) within the given time, i.e., the sampling interval T_s . To this aim, in [13], the initial radius is computed as¹

$$\rho_k = \min \{ \rho_k^1, \rho_k^2 \}, \quad (28)$$

where

$$\rho_k^1 = \left\| \bar{\mathbf{U}}_k^{\text{unc}} - \mathbf{V}_k \mathbf{U}_k^{\text{bab}} \right\|_2, \quad (29)$$

and

$$\rho_k^2 = \left\| \bar{\mathbf{U}}_k^{\text{unc}} - \mathbf{V}_k \mathbf{U}_k^{\text{ed}} \right\|_2. \quad (30)$$

Radius ρ_k^1 in (29) depends on $\mathbf{U}_k^{\text{bab}}$ that corresponds to the rounded unconstrained solution to the closest integer vector which respects the constraints [14], i.e.,

$$\mathbf{U}_k^{\text{bab}} = \lfloor \mathbf{U}_k^{\text{unc}} \rfloor \quad \text{s.t. } \mathbf{U}_k^{\text{bab}} \in \mathbb{U} \quad (31)$$

Radius ρ_k^2 , as can be seen in (30), is a function of \mathbf{U}_k^{ed} which is the previously applied solution \mathbf{U}_{k-1} shifted by one step [1], i.e., the initial guess for the GNA $\mathbf{U}_k^{\text{unc}}(0)$.

However, the computation of the radius ρ_k is more challenging when (17) has to be considered. More specifically, the hypersphere needs to include at least one switch position that leads to a current which does not violate its respective bound i_{bnd} . Calculating the radius based on (28) does not guarantee that the problem is feasible since the unconstrained solution is computed while neglecting the current constraint. As a consequence, the procedure for the computation of the initial radius needs to be revised. To this end, the constraint on the current—which is an output constraint since $\mathbf{y}_k = \mathbf{i}_k^{dq}$ —is translated into an input one, as follows. By using (14), (17) can be rewritten as

$$\left\| \mathbf{C}_k \mathbf{A}_k \mathbf{x}_k + \mathbf{C}_k \tilde{\mathbf{B}} \frac{V_{\text{dc}}}{2} \mathbf{u}_k^{dq} \right\|_2 \leq i_{\text{bnd}} \quad (32)$$

that is an ellipse \mathcal{E} in the rotating reference frame centered at

$$\mathbf{u}_k^{\text{constr}, dq} = -\frac{V_{\text{dc}}}{2} \tilde{\mathbf{B}}^{-1} \mathbf{A}_k \mathbf{x}_k \quad (33)$$

where \mathbf{u}_k^{dq} is the switch position in the rotating (dq) frame. The ellipse \mathcal{E} can be expressed in a matrix form as $\tilde{\mathbf{p}}^T \tilde{\mathbf{G}} \tilde{\mathbf{p}} = 0$ where the vector $\tilde{\mathbf{p}}$ is defined as $\tilde{\mathbf{p}} = [\mathbf{u}_k^{dq, T} \ 1]^T$ and the matrix $\tilde{\mathbf{G}}$ as

$$\tilde{\mathbf{G}} = \begin{bmatrix} \Theta & \Psi^T/2 \\ \Psi/2 & \Gamma \end{bmatrix}, \quad (34)$$

where

$$\Theta = \left(\mathbf{C}_k \tilde{\mathbf{B}} \frac{V_{\text{dc}}}{2} \right)^T \left(\mathbf{C}_k \tilde{\mathbf{B}} \frac{V_{\text{dc}}}{2} \right), \quad (35)$$

$$\Psi = 2\mathbf{x}_k^T (\mathbf{C}_k \mathbf{A})^T \left(\mathbf{C}_k \tilde{\mathbf{B}} \frac{V_{\text{dc}}}{2} \right), \quad (36)$$

¹Note that the superscripts in (28) are not powers but indices.

$$\Gamma = \mathbf{x}_k^T (\mathbf{C}_k \mathbf{A})^T (\mathbf{C}_k \mathbf{A}) \mathbf{x}_k - i_{\text{bnd}}^2. \quad (37)$$

The ellipse can be brought back to the three-phase (abc) frame $\mathbf{p}^T \mathbf{G} \mathbf{p} = 0$ by employing the transformation from dq to abc , i.e.,

$$\mathbf{G} = \bar{\mathbf{T}}^T(\vartheta) \tilde{\mathbf{G}} \bar{\mathbf{T}}(\vartheta) \quad (38)$$

being the vector \mathbf{p} defined as $\mathbf{p} = \mathbf{u}_k^{abc}$ and $\bar{\mathbf{T}}(\vartheta) = [\mathbf{T}(\vartheta); 1/2 \ 1/2 \ 1/2]$.

In order to meet constraint (17) the control input has to lie within the ellipse \mathcal{E} , as shown in Fig. 3 in the $\alpha\beta$ plane. It is worth remembering that there is a static relation between abc and $\alpha\beta$ frames. Formulating the output constraint (17) as an input one changes the feasible set of the integer-valued input \mathbb{U} to $\mathbb{U}^{\text{constr}} = \mathbf{U}^{\text{constr}_1} \times \mathbf{U}^{\text{constr}_2} \times \dots \times \mathbf{U}^{\text{constr}_N}$ with $\mathbf{U}^{\text{constr}_i} = \mathbf{U}$ for $i \in \{2, \dots, N\}$ and

$$\mathbf{U}^{\text{constr}_1} = \{ \mathbf{u}_k^{abc} | \mathbf{u}_k^{abc} \in \mathcal{E} \wedge \mathbf{u}_k^{abc} \in \mathbf{U} \}. \quad (39)$$

Having defined the feasible set, the computation of the initial radius ρ_k has to be refined accordingly; while maintaining the desired properties the initial radius should have—as explained above—feasibility has to be ensured as well. Therefore, the initial radius is computed as in [8]. To explain the concept of the refined feasible set and the revised computation of the radius, Fig. 6 shows a case where the feasible set does not include all possible switch positions, but rather a subset of them, as limited because of the current constraint (17). For simplicity, the visualization is done in the $\alpha\beta$ plane. As can be seen, the nearest point (i.e., switch position) to the unconstrained solution does not respect the current constraint. This implies that the hypersphere needs to be enlarged to include at least one feasible point. Hence, the radius ρ_k increases until it encloses until one feasible point, i.e., the point $\mathbf{u}_k^{\text{feas}, \alpha\beta}$, which corresponds to a switch position in the abc frame.

V. SIMULATION RESULTS

The proposed method was simulated for a 3L-NPC inverter with a constant dc bus voltage $V_{\text{dc}} = 540\text{V}$, driving a 1.6 kW SynR motor. The motor nominal current and speed were 4 A and 1500 rpm, respectively. All the simulations were carried out under the maximum torque per ampere (MTPA) condition. Accordingly, the nominal torque is produced by $i^d = 2.25\text{ A}$ and $i^q = 3.30\text{ A}$. The sampling interval was $T_s = 25\ \mu\text{s}$. The performance of direct MPC with long prediction horizon was investigated at steady-state, for a ten-step ($N = 10$) prediction horizon. To minimize the nonlinear cost function J_k , only one iteration of the proposed algorithm was needed. A matrix $\mathbf{Q} = \text{diag}(132, 132)$ was chosen by a trial-and-error procedure, to obtain a switching frequency of approximately 190 Hz.

The results over one fundamental period are depicted in Fig. 4. Fig. 4a shows the three-phase switch positions, i.e., the output voltage of direct MPC. The corresponding stator current i^{abc} and the fluxes (in the rotating (dq) reference frame) along with their references are depicted in Fig. 4b and Fig. 4d, respectively. Both flux linkages and stator currents follow accurately their reference values. For the sake of

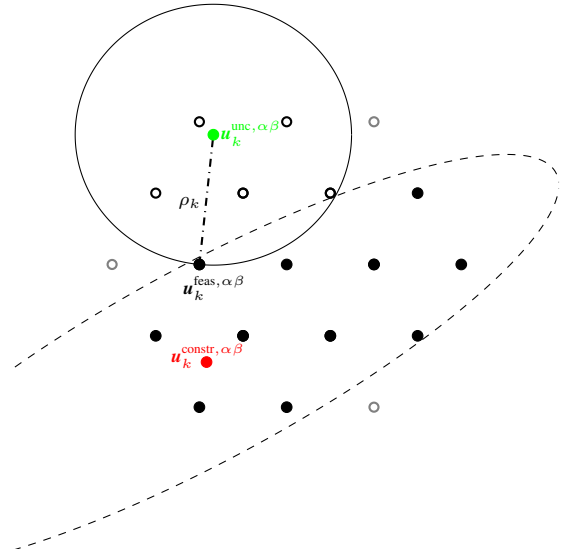


Fig. 3. Feasible set in the $\alpha\beta$ plane when the current constraint (17) is active. The switch positions that satisfy the current constraint are the black solid circles. In this example, only one solid circle lies inside the hypersphere and meets the constraint at the same time; this is the solution to the problem, and the corresponding three-phase switch position is applied to the converter.

completeness, the harmonic content of the stator currents is shown in Fig. 4c. A current total harmonic distortion (THD) of 5.01 % is achieved, a satisfactory result if compared with the THD = 7.62 % of the conventional asymmetric space vector modulation (SVM) at the same switching frequency, see also Fig. 5.

Fig. 5 presents the performance with different horizon lengths ($N = 1, 5, 10$) compared to SVM. Each dot corresponds to a single simulation carried out with a different value for the diagonal, nonzero entries of \mathbf{Q} , to test different switching frequencies. In this work, the fundamental frequency was 50 Hz. It is worth noting that the proposed direct MPC shows its advantages with respect to SVM at lower switching frequencies or, more generally, when the ratio between the switching frequency and the fundamental one is less than 10 (Fig. 5). One more thing observed in this figure, is that the current THD reduces as the length of the prediction horizon increases. This is in line with the published literature, see, e.g., [15] and [6]. However, it should be mentioned that the computational requirements increase with the length of the prediction horizon, thus, there is a trade-off between system performance and prediction horizon that can be implemented [13].

Fig. 6 shows the effectiveness of the current limitation around $1.05 I_N$, envisaging the unconstrained case (Fig. 6a) and the constrained one (Fig. 6b); the stator current is depicted in the stationary ($\alpha\beta$) plane. As can be seen in Fig. 6a, where the current constraint (17) is inactive, such control actions (i.e., the switch position) are taken by the MPC algorithm that result—occasionally—in an instantaneous current that exceeds the bound, shown as a red dashed circle. On the other hand, when the current constraint is taken into account by

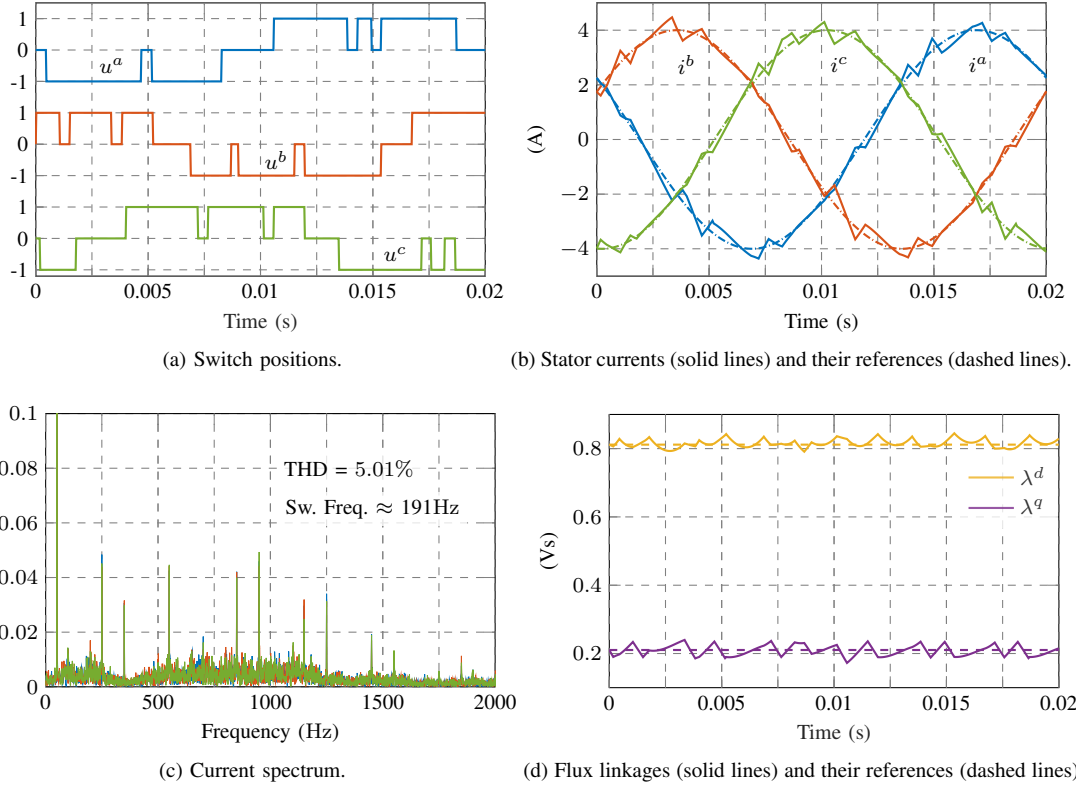


Fig. 4. Direct MPC with flux linkages reference tracking at steady-state condition, at full speed, rated torque and in MTPA condition.

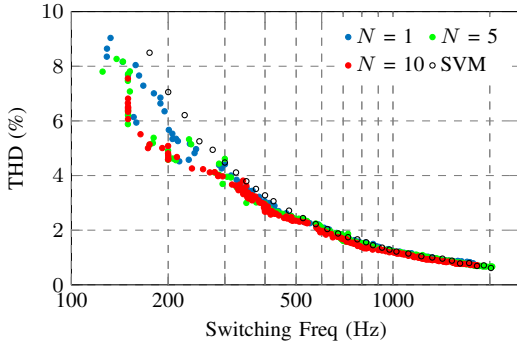


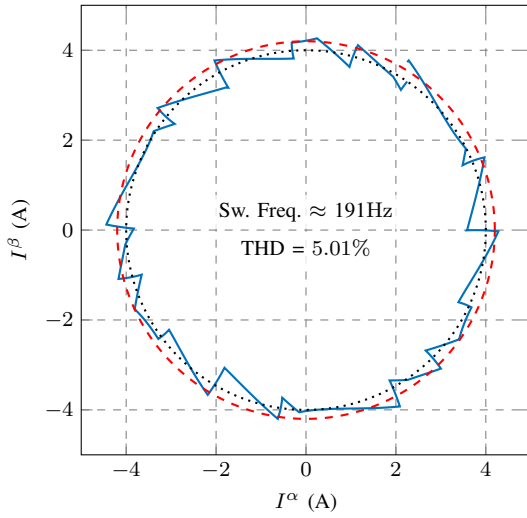
Fig. 5. THD vs switching frequency obtained with the MPC and different prediction horizons. The THD obtained with the asymmetric SVM is added for comparison.

the optimization problem, the current always remains within the current limitation (Fig. 6b). A side effect of this, is that the MPC algorithm forces the converter to switch more frequently to keep the current within its bound, so that the switching frequency increases from 190 Hz to 227 Hz. The subsequent increase in the switching losses is balanced by a lower current distortion, which reduces from THD = 5.01 % to THD = 4.12 %.

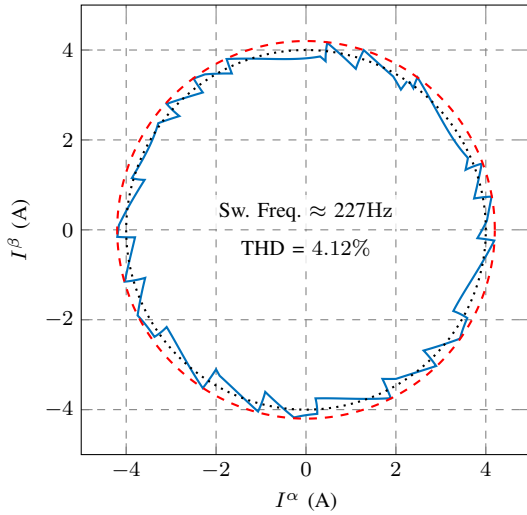
VI. CONCLUSIONS

In this paper a model predictive control (MPC) based current control algorithm for a synchronous reluctance (SynR) motor

drive was presented. It extends the work proposed in [1]; therein the controller is developed for an induction motor-based drive, thus limited to a linear system. On the contrary, in this work the nonlinear magnetic model of the machine is taken into account when modeling the system and formulating the subsequent optimal control problem. Moreover, an explicit stator current constraint is added to protect the inverter from overcurrents. To solve the underlying integer nonlinear optimization problem in a computationally efficient manner a smart branch-and-bound algorithm is employed akin to that proposed in [1]. This algorithm relies on the so-called unconstrained solution of the problem which is computed numerically. Based on the presented results, it can be seen that the proposed constrained long-horizon MPC improves the performance of the drive when compared with conventional control methods based on, e.g., space vector modulation (SVM). This improvement, as quantified by the stator current total harmonic distortion (THD), is more prominent at low switching frequencies.



(a) Without any current constraint.



(b) With maximum current constraint.

Fig. 6. Effects of the current limitation constraint. The solid lines represent the trajectory of the real current vector in the stationary $(\alpha\beta)$ reference frame, whereas the dotted and dashed circles show its reference trajectory and bound, respectively.

APPENDIX

The matrices used in the paper are

$$S = \begin{bmatrix} I_3 & \mathbf{0}_3 & \mathbf{0}_3 & \cdots & \mathbf{0}_3 \\ -I_3 & I_3 & \mathbf{0}_3 & \ddots & \vdots \\ \mathbf{0}_3 & -I_3 & I_3 & \ddots & \mathbf{0}_3 \\ \vdots & \ddots & \ddots & \ddots & \mathbf{0}_3 \\ \mathbf{0}_3 & \cdots & \mathbf{0}_3 & -I_3 & I_3 \end{bmatrix} \quad (40)$$

$$R = \begin{bmatrix} \mathbf{0}_3 & I_3 & \mathbf{0}_3 & \cdots & \mathbf{0}_3 \\ \mathbf{0}_3 & \mathbf{0}_3 & I_3 & \ddots & \vdots \\ \vdots & & \ddots & \ddots & \mathbf{0}_3 \\ \mathbf{0}_3 & \cdots & \cdots & \mathbf{0}_3 & I_3 \\ \mathbf{0}_3 & \cdots & \cdots & \mathbf{0}_3 & I_3 \end{bmatrix} \quad (41)$$

REFERENCES

- [1] T. Geyer and D. E. Quevedo, "Multistep finite control set model predictive control for power electronics," *IEEE Transactions on Power Electronics*, vol. 29, no. 12, pp. 6836–6846, Dec. 2014.
- [2] G. Pellegrino, T. M. Jahns, N. Bianchi, W. Soong, and F. Cupertino, *The Rediscovery of Synchronous Reluctance and Ferrite Permanent Magnet Motors*. Springer, 2016.
- [3] L. Ortombina, F. Tinazzi, and M. Zigliotto, "Magnetic modeling of synchronous reluctance and internal permanent magnet motors using radial basis function networks," *IEEE Transactions on Industrial Electronics*, vol. 65, no. 2, pp. 1140–1148, Feb. 2018.
- [4] J. B. Rawlings and D. Q. Mayne, *Model Predictive Control: Theory and Design*. Madison, WI: Nob Hill, 2009.
- [5] R. Antonello, M. Carraro, L. Peretti, and M. Zigliotto, "Hierarchical scaled-states direct predictive control of synchronous reluctance motor drives," *IEEE Transactions on Industrial Electronics*, vol. 63, no. 8, pp. 5176–5185, Aug. 2016.
- [6] T. Geyer, P. Karamanakos, and R. Kennel, "On the benefit of long-horizon direct model predictive control for drives with LC filters," in *2014 IEEE Energy Conversion Congress and Exposition (ECCE)*, Sep. 2014, pp. 3520–3527.
- [7] B. Hassibi and H. Vikalo, "On the sphere-decoding algorithm I. Expected complexity," *IEEE Transactions on Signal Processing*, vol. 53, no. 8, pp. 2806–2818, Aug. 2005.
- [8] P. Karamanakos, T. Geyer, and R. Kennel, "Constrained long-horizon direct model predictive control for power electronics," in *2016 IEEE Energy Conversion Congress and Exposition (ECCE)*, Sep. 2016, pp. 1–8.
- [9] E. Liegmann, P. Karamanakos, T. Geyer, T. Mouton, and R. Kennel, "Long-horizon direct model predictive control with active balancing of the neutral point potential," in *2017 IEEE International Symposium on Predictive Control of Electrical Drives and Power Electronics (PRECEDE)*, Sept 2017, pp. 89–94.
- [10] F. Grimm, Z. Zhang, F. Wang, and R. Kennel, "Multistep predictive control of three-level npc converters using weak derivative linearization," in *2017 Chinese Automation Congress (CAC)*, Oct 2017, pp. 4672–4677.
- [11] P. Karamanakos, T. Geyer, and R. Kennel, "Reformulation of the long-horizon direct model predictive control problem to reduce the computational effort," in *2014 IEEE Energy Conversion Congress and Exposition (ECCE)*, Sep. 2014, pp. 3512–3519.
- [12] Å. Björck, *Numerical Methods for Least Squares Problems*. Society for Industrial and Applied Mathematics, 1996. [Online]. Available: <https://epubs.siam.org/doi/abs/10.1137/1.9781611971484>
- [13] P. Karamanakos, T. Geyer, and R. Kennel, "Suboptimal search strategies with bounded computational complexity to solve long-horizon direct model predictive control problems," in *2015 IEEE Energy Conversion Congress and Exposition (ECCE)*, Sep. 2015, pp. 334–341.
- [14] L. Babai, "On Lovász' lattice reduction and the nearest lattice point problem," *Combinatorica*, vol. 6, no. 1, pp. 1–13, 1986.
- [15] T. Geyer and D. E. Quevedo, "Performance of multistep finite control set model predictive control for power electronics," *IEEE Transactions on Power Electronics*, vol. 30, no. 3, pp. 1633–1644, Mar. 2015.