2018 European Control Conference (ECC)
June 12-15, 2018. Limassol, Cyprus

Model Predictive Direct Switching Control for Multi-Phase Permanent-Magnet Synchronous Motors

Michael Ringkowski[1], Stefan Gering[2], Maximilian Manderla[2], Eckhard Arnold[1] and Oliver Sawodny[1]

Abstract— **In model predictive direct switching control (MPDSC) approaches, the finite set of inverter switch positions for the control of electrical drives is taken as control variables within the MPC underlying optimization problem. The cost function can be designed to track a torque or current reference while also considering additional control goals like minimizing switching and conduction losses, as well as physical constraints. The challenge is to solve the resulting integer quadratically constrained quadratic program (IQCQP) with high sampling frequencies online in order to obtain the optimal switch positions at each fixed time step. This paper presents two new MPDSC algorithms, namely *relaxed barrier functions iteration scheme* (RBF) and *alternating direction method of multipliers heuristic* (ADMM) and compares them with two state-of-the-art algorithms, namely *full enumeration* (FE) and *multistep with sphere decoding* (MSD) for the example of a multi-phase permanent-magnet synchronous motor (PMSM) in simulations.**

I. INTRODUCTION

In recent years, the research interest for high-performance electrical drives has increased due to demanding requirements in automotive applications. Especially in the context of autonomous driving, efficient and fault-tolerant electrical drives pose a challenging problem. One important safety critical application is the PMSM for steering systems.

In order to address the requirements for fault tolerance by design, physical redundancy in form of multi-phase PMSMs can be introduced [1]. In case of a faulty subsystems, the remaining subsystems can be used to compensate for the faulty one. In this paper, the example of a 4×3-PMSM with separated topology is considered (Fig. 1). Each of the four subsystems is controlled by its own voltage source inverter (VSI).

The requirements for efficiency and fault-tolerant control can be targeted at the same time with modern MPDSC approaches, which in addition to classical control approaches for electrical drives, e.g., field-oriented control (FOC) or direct torque control (DTC) [2], offer more flexibility, especially in the case of fault, due to online optimization. In MPDSC approaches, the discrete inverter switch positions at each fixed time step constitute a finite control set within the underlying optimization problem. A cost function is designed such that a reference torque or current can be tracked while minimizing the switching and conduction losses. Moreover, the MPDSC problem formulation allows for the inclusion of physical current constraints [3], as well as the handling of multivariable systems [4] and promises an improved transient performance compared to classical approaches [5]. In the last decade, model predictive control (MPC) for power electronics is emerging in science and technology [4]. The reason for this is the increased available computational power of control platforms like digital signal processors and field programmable gate arrays for embedded applications, rendering hardware implementation of more complex control techniques feasible for very short sampling times, which are required for the control of electrical drives [6].
The MPC techniques for power electronics can be divided into continuous MPC and finite control set MPC (FCS-MPC). Continuous MPC requires a modulator step, which translates a continuous voltage signal into switching times and switch positions of the voltage source inverter (VSI) controlling the applied voltages, similar to classical FOC approaches. On the other hand, FCS-MPC techniques directly compute the switch positions at given sampling times and are in the following referred to MPDSC. These methods do not require a modulator step and allow for a compact problem formulation, especially in failure scenarios.

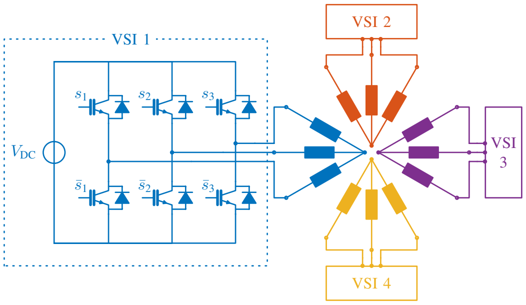

Fig. 1. 4×3 PMSM with separated topology, consisting of four conventional wye-connected three-phase subsystems, each controlled by its own voltage source inverter with switch positions s_i ($\bar{s}_i$ negated position).

In [3], trajectory based model predictive direct torque control (MPDTC) with a prediction horizon of one time step for a standard PMSM drive system is presented considering the physical current constraints and attraction regions for optimal operation below and above rated speed, respectively, yielding promising results. However, the optimal solution is found using exhaustive search, which is not feasible for multi-phase topologies, since thousands of switching sequences would have to be evaluated. For longer horizons, [7] presents a promising method for model predictive direct current control (MPDCC) called multistep with sphere decoding (MSD). A

[1]Michael Ringkowski, Eckhard Arnold and Oliver Sawodny are with the Institute for System Dynamics, Universität Stuttgart, Germany Michael.Ringkowski@isys.uni-stuttgart.de

[2]Stefan Gering and Maximilian Manderla are with the Corporate Research of Robert Bosch GmbH, Renningen, Germany Stefan.Gering@de.bosch.com

978-3-9524-2699-9 ©2018 EUCA

branch and bound method is employed in order to reduce the computational effort with the intention of enabling horizons greater than one. As a result, a large reduction of the computation time can be observed also for the horizon of one [5]. Several improvements of the method have been proposed. In [8] and [9], a preconditioning step and a partially offline polyhedral partition of the solution space, respectively, are applied in order to speed up MSD. However, both methods seem infeasible for the problem at hand due to a time-varying and non-sparse lattice generator matrix. In [10], a modified sphere decoding algorithm including a projection step is investigated, which yields reduced computation times during transients, at the cost of an increased mean computation time. Another way of improving the performance other than increasing the horizon is investigated in [11]. An offline calculated approximation of the terminal cost term is employed which yields superior results compared to the previous mentioned methods. However, the method seems not to be directly applicable to the considered multi-phase PMSM due to the time-variant model and non-sinusoidal currents in failure scenarios.

This paper presents two new MPDSC algorithms, namely *relaxed barrier functions iteration scheme* (RBF) and *alternating direction method of multipliers heuristic* (ADMM), which yield approximate solutions, and compares them with two state-of-the-art algorithms, namely *full enumeration* (FE) and *multistep with sphere decoding* (MSD), which yield the global solution, for the example of a multi-phase PMSM based on simulation results. To the best of the authors knowledge, MPDSC has not yet been investigated for the problem of designing a controller for multi-phase PMSMs.

The paper is structured as follows: In Section 2, the considered multi-phase PMSM and the MPDSC optimization problem for torque control are introduced. Section 3 presents the four considered algorithms. In Section 4, the algorithms are compared based on simulation results and a brief conclusion is given in Section 5.

II. Problem Setup

A. Multi-Phase Permanent-Magnet Synchronous Motors

The system dynamics for the wye-connected multi-phase PMSM with separated topology (Fig. 1) can be derived in stator-fixed abc-coordinates as

$$\dot{\mathbf{I}}_{\text{abc}} = \mathbf{L}_{\text{abc}}^{-1}\bar{\mathbf{K}}\left(\mathbf{U}_{\text{abc}} - \mathbf{R}_{\text{abc}}\mathbf{I}_{\text{abc}} - \mathbf{U}_{\text{i}}\right) \;, \tag{1a}$$

$$\mathbf{U}_{\text{i}} = \left(\frac{\mathrm{d}\mathbf{L}_{\text{abc}}}{\mathrm{d}\varphi_{\text{el}}}\mathbf{I}_{\text{abc}} + \frac{\mathrm{d}\boldsymbol{\Psi}_{\text{pm,abc}}}{\mathrm{d}\varphi_{\text{el}}}\right)\omega_{\text{el}} \;, \tag{1b}$$

$$\bar{\mathbf{K}} = \boldsymbol{I}_{n_{\text{sys}}} - \frac{1}{3}\left(\boldsymbol{I}_{n_{\text{sys}}} \otimes \mathbf{1}_{3\times 3}\right) \tag{1c}$$

where $\mathbf{I}_{\text{abc}}(t)$ denotes the currents, $\mathbf{U}_{\text{abc}}(t)$ the input voltages, $\mathbf{L}_{\text{abc}}(\varphi_{\text{el}}(t))$ the inductance matrix, $\mathbf{R}_{\text{abc}}$ the matrix of resistances, $\boldsymbol{\Psi}_{\text{pm,abc}}(\varphi_{\text{el}}(t))$ the magnetic flux, φ_{el} and ω_{el} the electrical motor angle and velocity, respectively, and $\otimes$ the Kronecker product. The italic letter $\boldsymbol{I}$ denotes an identity matrix and n_{sys} the number of subsystems. The parameters for this model can be obtained as Fourier coefficients from FEM calculations for the specific machine type yielding a complex model including higher harmonics suited for simulations.

A simplified model in rotor-fixed dq-coordinates is usually used for the controller design. The system dynamics in dq-coordinates are given by

$$\dot{\mathbf{I}}_{\text{dq}} = \mathbf{L}_{\text{dq}}^{-1}\left(\mathbf{U}_{\text{dq}} - \left(\mathbf{R}_{\text{dq}} + \omega_{\text{el}}\mathbf{Y}\mathbf{L}_{\text{dq}}\right)\mathbf{I}_{\text{dq}} - \omega_{\text{el}}\mathbf{Y}\boldsymbol{\Psi}_{\text{pm,dq}}\right) \tag{2}$$

where $\mathbf{Y} = \boldsymbol{I}_{n_{\text{sys}}} \otimes [0, -1; 1, 0]$ is a topology matrix using the Kronecker product and $\mathbf{L}_{\text{dq}} = \mathbf{T}_{\text{dq}}\mathbf{L}_{\text{abc}}\mathbf{T}_{\text{abc}}$ the Park-Clarke-transformed inductance matrix. However, only by neglecting higher harmonics, this dq-transformation yields a constant inductance matrix $\mathbf{L}_{\text{dq}}$. Due to the separated topology for the considered 4×3-PMSM with significant higher harmonics, this results in a relatively large model mismatch between the abc-model and the dq-model (Fig. 2), which is tolerated here for the sake of a simpler prediction model.

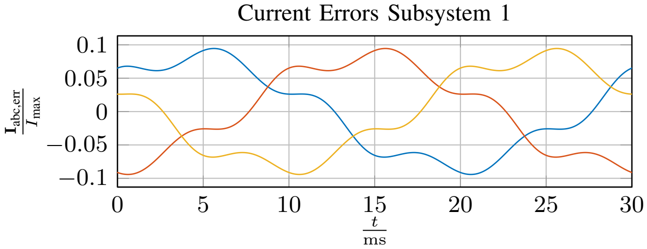

Fig. 2. Model mismatch $\mathbf{I}_{\text{abc,err}} = \mathbf{I}_{\text{abc}}^{\text{abc}} - \mathbf{I}_{\text{abc}}^{\text{dq}}$ between the abc-currents of the dq-model and the abc-model at full-load operation with 500 rpm for the first subsystem.

B. Optimization Problem

The MPDSC problem with a prediction horizon n_{h} is solved at time step k by minimizing a scalar cost function $J_k(\mathbf{s})$, which depends on the switch positions $\mathbf{s} = \mathbf{s}_{\text{abc}}$, subject to some constraints. For torque tracking, i.e. MPDTC, the optimization can be formulated as

$$\underset{\mathbf{s}_k,\dots,\mathbf{s}_{k+n_{\text{h}}-1}}{\text{minimize}} \quad J_k = \sum_{i=k}^{k+n_{\text{h}}-1} J_{\text{t},i+1} + J_{\text{c},i+1} + J_{\text{s},i} \tag{3a}$$

$$\text{subject to} \quad \mathbf{I}_{i+1} = \mathbf{A}\mathbf{I}_i + \mathbf{B}(\varphi_i)\mathbf{s}_i + \mathbf{E}\boldsymbol{\Psi}_{\text{pm}} \;, \tag{3b}$$

$$T_{i+1} = f(\mathbf{I}_{i+1}) \;, \tag{3c}$$

$$\left\|\mathbf{I}_{i+1}^{j}\right\|_2^2 \leq I_{\text{max}}^2 \;, \tag{3d}$$

$$\mathbf{s}_i \in \mathcal{S} = \{-1, 1\}^{3n_{\text{sys}}} \;, \tag{3e}$$

where the cost function consists of a torque tracking term $J_{\text{t},i+1} = \left\|T_{i+1}^* - T_{i+1}\right\|_2^2$, a term for energy optimality $J_{\text{c},i+1} = \lambda_{\text{i}}\left\|\mathbf{I}_{i+1}\right\|_2^2$ as well as a term to penalize switching losses $J_{\text{s},i} = \lambda_{\text{s}}\left\|\mathbf{s}_i - \mathbf{s}_{i-1}\right\|_2^2$. For the prediction, the system dynamics of the dq-model (3b) are employed and the torque is computed as a function of the predicted currents (3c). In order to get a quadratic cost function the quadratic torque equation has to be linearized (21) around the last current measurement, which is acceptable due to fast sampling frequencies and short prediction horizons. The physical current

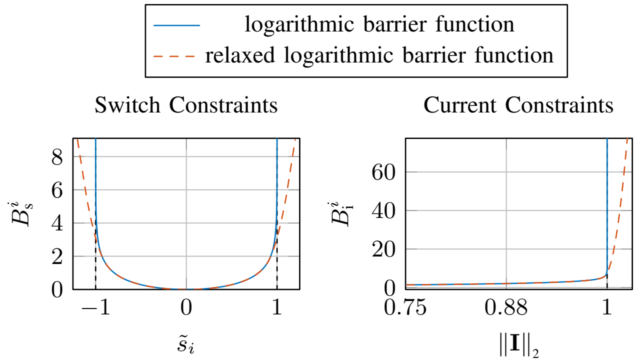

Fig. 3. Visualization of the logarithmic barrier function based switch constraints (left) and current constraints (right) with and without quadratic relaxation.

limits are considered using the squared two-norm of the current vector resulting in a quadratic constraint (3d) where $\mathbf{I}_{i+1}^{j}$ denotes the sub-vector, which corresponds to the i-th prediction step and the j-th subsystem. The discrete nature of the switch positions yields the integer constraint (3e).

Following the approach in [12], the problem can be rewritten in a condensed form as an IQCQP using stacked vectors for the switching sequence $\mathbf{S}_k = [\mathbf{s}_k^{\mathrm{T}}, \dots, \mathbf{s}_{k+n_\mathrm{h}-1}^{\mathrm{T}}]^{\mathrm{T}}$ yielding

$$\underset{\mathbf{S}_k}{\text{minimize}} \qquad J_k = \frac{1}{2}\mathbf{S}_k^{\mathrm{T}}\mathbf{P}(k)\mathbf{S}_k + \mathbf{q}(k)^{\mathrm{T}}\mathbf{S}_k \tag{4a}$$

$$\text{subject to} \qquad \left\|\mathbf{I}_{i+1}^{j}(\mathbf{S}_k)\right\|_2^2 \le I_{\text{max}}^2 \;, \tag{4b}$$

$$\mathbf{S}_k \in \mathcal{S} = \{-1, 1\}^{3 n_{\text{sys}} n_\mathrm{h}} \;, \tag{4c}$$

where $\mathbf{P}(k)$ and $\mathbf{q}(k)$ are defined in the appendix.

III. Algorithms

This section presents the considered algorithms, where the first two (RBF, ADMM) yield approximate solutions and were adapted for the MPDSC problem, whereas the last two (FE, MSD) yield the global optimum and are commonly used.

A. Relaxed Barrier Function Iteration Scheme

Another approach is to solve the MPDTC problem approximately by reformulating the IQCQP as an unconstrained convex problem. This can be achieved by replacing the input and current constraints by quadratically relaxed logarithmic barrier functions B^i [13] (Fig. 3). In order to be able to apply this concept to the given IQCQP (4), the set $\mathcal{S}$ in constraint (3e) has to be relaxed to the convex set $\tilde{\mathcal{S}} = [-1, 1]^{3 n_{\text{sys}} n_\mathrm{h}}$. The resulting nonlinear convex unconstrained problem can be solved using an iteration scheme based on line search. This approach, in the following referred to as RBF, is proposed in [14] where closed-loop stability is proved. The integer switch position vector can be obtained from the solution of the unconstrained problem by rounding or sum-up rounding [15]. The latter is a simple strategy that considers the sum of previous rounding errors in the rounding process. Here, the Newton search direction with a step size of one is chosen as proposed in [14].

B. Alternating Directions Method of Multipliers Heuristic

Alternating direction method of multipliers (ADMM) is a first-order method for solving convex problems of the form

$$\underset{\mathbf{S},\mathbf{z}}{\text{minimize}} \qquad J = f(\mathbf{S}) + g(\mathbf{z}) \tag{5a}$$

$$\text{subject to} \qquad \tilde{\mathbf{A}}\mathbf{S} + \tilde{\mathbf{B}}\mathbf{z} = \tilde{\mathbf{c}} \;, \tag{5b}$$

where f and g are convex [16]. The idea is to split the problem into easier subproblems and the augmented Lagrangian is used to solve the subproblems iteratively in an alternating manner. The method can also be applied to problems with non-convex constraint sets such as mixed-integer QPs [17]. However, since convergence and optimality cannot be guaranteed for non-convex problems, it is called a heuristic, although practical examples show promising results. In this paper, the heuristic is adapted for the given IQCQP (4) and referred to as ADMM for the sake of simplicity.

The optimization problem (4) can be reformulated to fit the structure of (5)

$$\underset{\mathbf{S},\mathbf{z}}{\text{minimize}} \qquad J = \underbrace{\frac{1}{2}\mathbf{S}^{\mathrm{T}}\mathbf{P}\mathbf{S} + \mathbf{q}^{\mathrm{T}}\mathbf{S}}_{f(\mathbf{S})} + \underbrace{\mathcal{I}(\mathbf{z})}_{g(\mathbf{z})} \tag{6a}$$

$$\text{subject to} \qquad \underbrace{\begin{bmatrix} \boldsymbol{I} \\ \boldsymbol{\Gamma}_\mathrm{s} \end{bmatrix}}_{\tilde{\mathbf{A}}} \mathbf{S} + \underbrace{\begin{bmatrix} \mathbf{0} \\ \boldsymbol{\Gamma}_\mathrm{r} \end{bmatrix}}_{-\tilde{\mathbf{c}}} + \underbrace{(-\boldsymbol{I})}_{\tilde{\mathbf{B}}} \underbrace{\begin{bmatrix} \mathbf{z}_\mathrm{s} \\ \mathbf{z}_\mathrm{c} \end{bmatrix}}_{\mathbf{z}} = \mathbf{0} \tag{6b}$$

with the indicator function

$$\mathcal{I}(\mathbf{z}) = \mathcal{I}\left(\begin{bmatrix} \mathbf{z}_\mathrm{s} \\ \mathbf{z}_\mathrm{c} \end{bmatrix}\right) = \begin{cases} 0 & \mathbf{z}_\mathrm{s} \in \mathcal{S} \wedge \left\|\mathbf{z}_\mathrm{c}^{j}\right\|_2 \le I_{\max} \;, \\ \infty & \text{else} \end{cases} . \tag{7}$$

Herein, $\mathbf{z}_\mathrm{s}$ and $\mathbf{z}_\mathrm{c}$ are copies of the optimization variables $\mathbf{S}$ and the currents $\mathbf{I} = \boldsymbol{\Gamma}_\mathrm{s}\mathbf{S} + \boldsymbol{\Gamma}_\mathrm{r}$, respectively. The constraints (4c) and (4b) are implicitly imposed on $\mathbf{z}$ via the indicator function.

The algorithms minimizes the scaled augmented Lagrangian

$$L_{\rho_\mathrm{p}} = f(\mathbf{S}) + g(\mathbf{z}) + \frac{\rho_\mathrm{p}}{2}\left\|\tilde{\mathbf{A}}\mathbf{S} + \tilde{\mathbf{B}}\mathbf{z} - \tilde{\mathbf{c}} + \mathbf{u}\right\|_2^2 - \frac{\rho_\mathrm{p}}{2}\|\mathbf{u}\|_2^2 \tag{8}$$

with the scaled dual variable $\mathbf{u}$ and the penalty parameter ρ_p iteratively for $\mathbf{s}$ and $\mathbf{z}$ in an alternating fashion resulting in the three update steps

$$\mathbf{S}\text{-update:} \quad \mathbf{S}_{i+1} = \arg\min_{\mathbf{S}} L_{\rho_\mathrm{p}} \tag{9a}$$

$$\mathbf{z}\text{-update:} \quad \mathbf{z}_{i+1} = \arg\min_{\mathbf{z}} L_{\rho_\mathrm{p}} \tag{9b}$$

$$\mathbf{u}\text{-update:} \quad \mathbf{u}_{i+1} = \mathbf{u}_i + \tilde{\mathbf{A}}\mathbf{S}_{i+1} - \mathbf{z}_{i+1} - \tilde{\mathbf{c}} \tag{9c}$$

at each iteration i. The $\mathbf{S}$-update (9a) is based on minimizing an unconstrained QP, which can be solved analytically using the first order optimality condition $\nabla_{\mathbf{S}} L_{\rho_\mathrm{p}} = \mathbf{0}$ as

$$\mathbf{S}_{i+1} = \left(\mathbf{P} + \rho_\mathrm{p}\left(\boldsymbol{I} + \boldsymbol{\Gamma}_\mathrm{s}{}^{\mathrm{T}}\boldsymbol{\Gamma}_\mathrm{s}\right)\right)^{-1} \cdot \left(\rho_\mathrm{p}\left(\mathbf{z}_{\mathrm{s},i} - \mathbf{u}_{\mathrm{s},i} + \boldsymbol{\Gamma}_\mathrm{s}{}^{\mathrm{T}}\left(-\boldsymbol{\Gamma}_\mathrm{r} + \mathbf{z}_{\mathrm{c},i} - \mathbf{u}_{\mathrm{c},i}\right)\right) - \mathbf{q}\right) . \tag{10}$$

The $\mathbf{z}$-update (9b) is realized by a suitable projection of the minimizer of the norm, i.e., $\mathbf{z} = \tilde{\mathbf{A}}\mathbf{S}_{i+1} - \tilde{\mathbf{c}} + \mathbf{u}_i$, onto the feasible set

$$\mathcal{Z}(\mathbf{z}) = \left\{\mathbf{z} | \mathbf{z}_{\mathrm{s}} \in \mathcal{S} \wedge \left\|\mathbf{z}_{\mathrm{c}}^{j}\right\|_2 \leq I_{\mathrm{max}}, j = 1, \dots, n_{\mathrm{sys}} n_{\mathrm{h}}\right\} , \tag{11}$$

which can be solved for $\mathbf{z}_{\mathrm{s}}$ and $\mathbf{z}_{\mathrm{c}}$ separately as

$$\mathbf{z}_{\mathrm{s},i+1} = \Pi_{\mathcal{S}}(\mathbf{S}_{i+1} + \mathbf{u}_{\mathrm{s},i}) = \lfloor \mathbf{S}_{i+1} + \mathbf{u}_{\mathrm{s},i} \rceil_{\mathcal{S}} \ , \tag{12a}$$

$$\begin{aligned} \mathbf{z}_{\mathrm{c},i+1} &= \Pi_{L^2}(\underbrace{\mathbf{\Gamma}_{\mathrm{s}}\mathbf{S}_{i+1} + \mathbf{\Gamma}_{\mathrm{r}} + \mathbf{u}_{\mathrm{c},i}}_{\mathbf{z}_{\mathrm{c},i+1}^{-}}) \qquad (12b) \\ &= \begin{cases} \mathbf{z}_{\mathrm{c},i+1}^{-} & , \left\|\mathbf{z}_{\mathrm{c},i+1}^{-,j}\right\|_2 \leq I_{\mathrm{max}} \\ \mathbf{z}_{\mathrm{c},i+1}^{-} \frac{I_{\mathrm{max}}}{\max_j \mathbf{z}_{\mathrm{c},i+1}^{-,j}} & , \text{else} \end{cases} . \end{aligned}$$

Here, $\lfloor \cdot \rceil_{\mathcal{S}}$ denotes rounding to the nearest integer in $\mathcal{S}$. The $\mathbf{u}$-update (9c) adds the equality constraint violations.

In the classical ADMM, usually the norms of the primal and dual residual serve as a stopping criterion. However, the actual convergence of the residuals can require many iterations and for the heuristic, a good approximate solution is found after only a few iterations. Moreover, the calculation of the norm of the residuals is computationally expensive and the convergence rate of the ADMM algorithm depends in the choice of the penalty parameter ρ_{p} as discussed in [18]. Here, ρ_{p} was manually tuned. Therefore, it seems reasonable to introduce a maximal number of iterations as stopping criterion. For an implementation on hardware, this limit is given by the available computation time.

Since the cost function value in ADMM is not necessarily monotonically decreasing, the cost function is evaluated at each iteration step and its minimizer is chosen.

C. Full Enumeration

One rather intuitive way of solving problem (4) is to compute the cost function value for all admissible switching sequences $\mathbf{S}_k$ and choose the minimizer $\mathbf{S}_k^* = \arg\min J_k$, which constitutes the global optimum. This exhaustive search approach is called *full enumeration* (FE) or *exhaustive enumeration* [7]. However, the number of admissible switching sequences, which have to be investigated, increases exponentially with the prediction horizon. Hence, for the considered 4×3-PMSM, FE is limited to short prediction horizons even in simulations (Table I) but is addressed here nonetheless in order to obtain baseline results.

TABLE I

NUMBER OF INPUT COMBINATIONS FOR DIFFERENT HORIZONS.

horizon n_{h}	1	2	3
combinations	$2^{12} = 4096$	$2^{24} \approx 16 \cdot 10^6$	$2^{36} \approx 68 \cdot 10^9$

D. Multistep with Sphere Decoding

The state-of-the-art method for solving finite control set MPC problems with large prediction horizons is called multistep with sphere decoding (MSD) [7]. The idea is to reformulate the condensed MPC problem to an integer least-squares problem

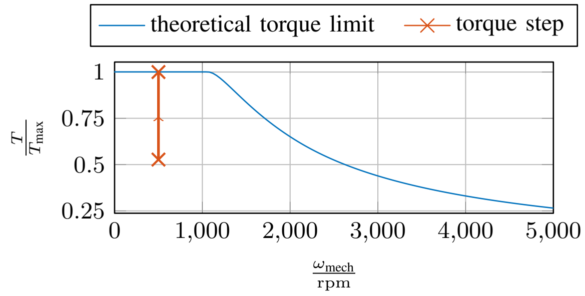

Fig. 4. Performance map showing the theoretical mean torque calculated for an electrical round over different velocities with the dq-model as well as the simulation scenario consisting of a torque step at medium velocity where the stationary operation points are marked with x.

$$\begin{aligned} \min_{\mathbf{S}_k} \quad & J_k = \left\|\mathbf{H}(k)\mathbf{S}_k + \mathbf{H}(k)\mathbf{P}^{-1}(k)\mathbf{q}(k)\right\|_2^2 & (13a) \\ \text{subject to} \quad & \mathbf{H}^{\mathrm{T}}(k)\mathbf{H}(k) = \mathbf{P}(k) \ , & (13b) \\ & \mathbf{S}_k \in \mathcal{S} = \{-1, 1\}^{3 n_{\mathrm{sys}} n_{\mathrm{h}}} \ . & (13c) \end{aligned}$$

The idea is to find the integer solution closest to the unconstrained solution $\mathbf{S}_{\mathrm{unc}}(k) = -\mathbf{P}(k)^{-1}\mathbf{q}(k)$ in transformed coordinates resulting from the multiplication with the triangular time-varying lattice generator matrix $\mathbf{H}(k)$. This can be achieved efficiently by exploiting the structure of $\mathbf{H}$ using a tailored branch and bound algorithm (sphere decoder). While in [7], the method is presented for MPDCC without current constraints, the method can be adapted for MPDTC with current constraints. The current constraints can be added as additional condition that is checked whenever a candidate solution is found. However, the choice of the initial sphere radius, which is crucial for the reduction of computation time, gets more involved.

IV. Simulation Results

The presented algorithms are tested in simulations for the example of a 4×3-PMSM with separated topology using Simulink. The considered scenario consists of a torque step selected based on the performance map of the machine (Fig. 4). For slower velocities, the maximal torque is bounded by a horizontal line due to current constraints. For higher velocities, the voltage limit of the dc source reduces the maximal torque with increasing velocity.

The torque results with the abc-model as simulation model and prediction horizon $n_{\mathrm{h}} = 1$ (Fig. 5) show that all four algorithms are able to track the torque reference accurately during the first steady-state operation. They demonstrate the same excellent transient performance during the torque reference step. After the torque step, a full-load operation point is reached and the periodic oscillations around the torque reference, which are due to the model mismatch between the prediction model (dq) and the simulation model (abc), are increased.

FE and MSD yield identical torque values and constitute the global optimal solution with respect to the cost function.

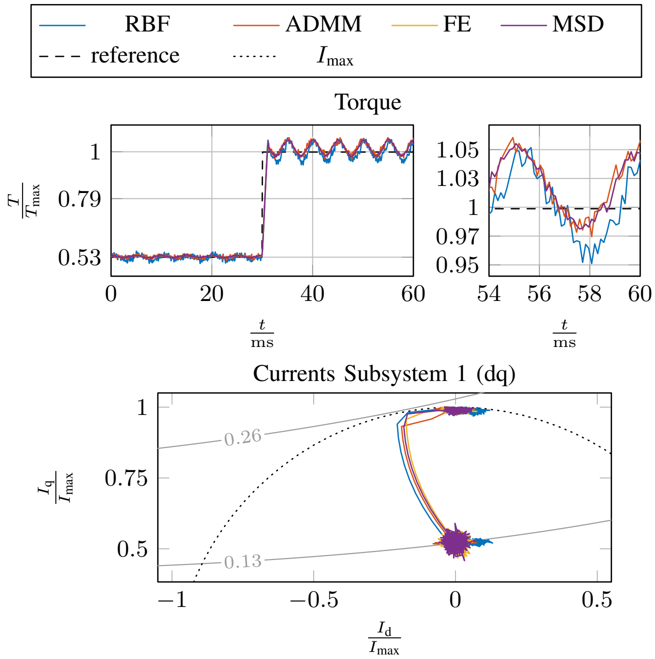

Fig. 5. MPDTC simulation results for the different algorithms. The torque reference is depicted with a dashed line and the current constraints with a dotted circle.

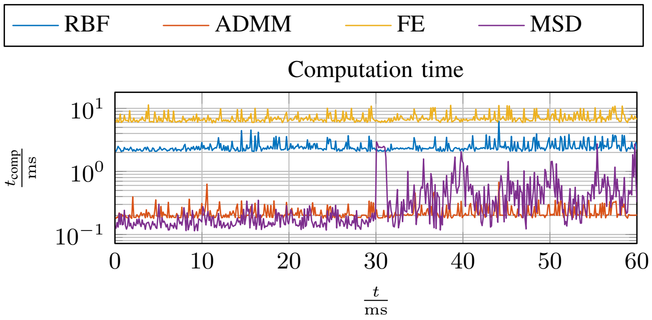

Fig. 6. Measured computation times t_{comp} on a standard PC with an Intel Core i5 CPU (2.4 GHz) and 8 Gb RAM for the different MPDTC algorithms.

The approximate solution of ADMM with ten iterations shows a torque trajectory with small deviations from the optimal solution, suggesting that an approximate solution in closed-loop operation yields a comparable performance. By increasing the iteration limit, the results of ADMM match the optimal solution more closely at the expense of larger computation times. The torque using RBF demonstrates a slightly increased ripple, which is expected, since the integer solution is found by rounding. Also, the cost function of RBF is different close to the current bounds which explains the larger deviations for active current constraints.

By plotting the transformed currents of the first subsystem in the dq-plane, the stationary operation points can be visualized including the circular current constraint and approximate lines of constant torque by neglecting coupling effects. During the partial-load stationary operation, the dq-currents stay close to a point on the 0.13 line where the currents are small. The torque reference step to full-load stationary operation results in a transition to a higher torque level on the current bound. The remaining subsystems show similar results.

The overall control performance of all four algorithms and the constraint satisfaction is very good. In order to be able to compare the algorithms, the computation time is an important measure (Fig. 6). The computation time is measured using the Matlab functions `tic` and `toc` on a standard PC. FE and RBF are very slow. While MSD is extremely fast during unconstrained steady state operation, during transients and with active constraints the worst-case computation time is similar to RBF and in theory even to FE. This can be solved using a projection on the convex hull by solving a QP with relaxed switching constraints [10]. However, this leads to an increased mean computation time. Due to the fixed numer of iterations, ADMM shows constant low computation times.

For longer horizons, FE is not feasible and MSD is only feasible with the slower projection approach or a suboptimal version, while the computation time of RBF and ADMM increases only moderately. However, the accuracy of the suboptimal solutions reduces and the topic requires some further research.

V. Conclusions

With regard to the implementation of an MPDSC algorithm for a multi-phase PMSM in hardware, the computation time seems to pose the biggest challenge and can be considered as the most important criterion for choosing an algorithm. Therefore, ADMM constitutes an appealing alternative to the established methods.

Further research might include the performance in the case of a faulty switch, the comparison of suboptimal MSD strategies compared to ADMM, two-degree of freedom approaches consisting of offline optimized trajectories and MPDSC as well as the hardware implementation.

Appendix

The matrices corresponding to the condensed optimization problem (4) are

$$\mathbf{P}(k) = 2\left(\boldsymbol{\tau}_{\text{s}}{}^{\text{T}}\boldsymbol{\tau}_{\text{s}} + \lambda_{\text{i}}\boldsymbol{\Gamma}_{\text{s}}{}^{\text{T}}\boldsymbol{\Gamma}_{\text{s}} + \lambda_{\text{s}}\boldsymbol{\Upsilon}_{\text{s}}{}^{\text{T}}\boldsymbol{\Upsilon}_{\text{s}}\right), \tag{14}$$

$$\mathbf{q}(k) = 2\left(\boldsymbol{\tau}_{\text{s}}{}^{\text{T}}\boldsymbol{\tau}_{\text{r}} + \lambda_{\text{i}}\boldsymbol{\Gamma}_{\text{s}}{}^{\text{T}}\boldsymbol{\Gamma}_{\text{r}} + \lambda_{\text{s}}\boldsymbol{\Upsilon}_{\text{s}}{}^{\text{T}}\boldsymbol{\Upsilon}_{\text{r}}\right), \tag{15}$$

$$\boldsymbol{\Gamma}_{\text{s}}(k) = \begin{bmatrix} \mathbf{B}_k & \mathbf{0} & \cdots & \mathbf{0} \\ \mathbf{A}\mathbf{B}_k & \mathbf{B}_{k+1} & \cdots & \mathbf{0} \\ \vdots & \vdots & & \vdots \\ \mathbf{A}^{k+n_{\text{h}}-1}\mathbf{B}_k & \mathbf{A}^{k+n_{\text{h}}-2}\mathbf{B}_{k+1} & \cdots & \mathbf{B}_{k+n_{\text{h}}} \end{bmatrix}, \tag{16}$$

$$\boldsymbol{\Gamma}_{\text{r}}(k) = \begin{bmatrix} \mathbf{A}\mathbf{I}_k + \mathbf{E}\boldsymbol{\Psi}_{\text{pm}} \\ \mathbf{A}^2\mathbf{I}_k + (\mathbf{A} + \boldsymbol{I})\,\mathbf{E}\boldsymbol{\Psi}_{\text{pm}} \\ \vdots \\ \mathbf{A}^{n_{\text{h}}}\mathbf{I}_k + \left(\mathbf{A}^{n_{\text{h}}-1} + \cdots + \mathbf{A} + \boldsymbol{I}\right)\mathbf{E}\boldsymbol{\Psi}_{\text{pm}} \end{bmatrix}, \tag{17}$$

$$\boldsymbol{\Upsilon}_{\mathrm{s}} = \begin{bmatrix} \boldsymbol{I} & \mathbf{0} & \cdots & \mathbf{0} \\ -\boldsymbol{I} & \boldsymbol{I} & \cdots & \mathbf{0} \\ \vdots & \ddots & \ddots & \vdots \\ \mathbf{0} & \cdots & -\boldsymbol{I} & \boldsymbol{I} \end{bmatrix}, \qquad \boldsymbol{\Upsilon}_{\mathrm{r}} = \begin{bmatrix} -\mathbf{s}_k \\ 0 \\ \vdots \\ 0 \end{bmatrix}, \tag{18}$$

$$\boldsymbol{\tau}_{\mathrm{s}}(k) = (-\mathbf{M}_{\mathrm{BD}}(\nabla T_k, n_{\mathrm{h}}))\boldsymbol{\Gamma}_{\mathrm{s}}(k)\,, \tag{19}$$

$$\boldsymbol{\tau}_{\mathrm{r}}(k) = \mathbf{t}_k^* - (T_k - \nabla T_k \mathbf{I}_k)\mathbf{1} - \mathbf{M}_{\mathrm{BD}}(\nabla T_k, n_{\mathrm{h}})\boldsymbol{\Gamma}_{\mathrm{r}}\,. \tag{20}$$

$\mathbf{M}_{\mathrm{BD}}(\nabla T_k, n_{\mathrm{h}})$ yields a block diagonal matrix, which can be computed with the Kronecker product $\boldsymbol{I}_{n_{\mathrm{h}}} \otimes \nabla T_k$.

A vector notation for the predicted torque values $\mathbf{t}_k = [T_{k+1}, \ldots, T_{k+n_{\mathrm{h}}}]^{\mathrm{T}}$ and torque reference $\mathbf{t}_k^* = [T_{k+1}^*, \ldots, T_{k+n_{\mathrm{h}}}^*]^{\mathrm{T}}$ is used together with the linearized torque equation

$$T_i \approx T_k + \nabla T_k(\mathbf{I}_i - \mathbf{I}_k)\,, \tag{21a}$$

$$T_k = \tfrac{3}{2} n_{\mathrm{p}} \left(\mathbf{I}_k^{\mathrm{T}} \tfrac{1}{2} \left(\mathbf{Y}\mathbf{L}_{\mathrm{dq}} + (\mathbf{Y}\mathbf{L}_{\mathrm{dq}})^{\mathrm{T}} \right) \mathbf{I}_k + (\mathbf{Y}\boldsymbol{\Psi}_{\mathrm{pm}})^{\mathrm{T}} \mathbf{I}_k \right)\,, \tag{21b}$$

$$\nabla T_k = \tfrac{3}{2} n_{\mathrm{p}} \left((\mathbf{Y}\boldsymbol{\Psi}_{\mathrm{pm}})^{\mathrm{T}} + \mathbf{I}_k^{\mathrm{T}} \left(\mathbf{Y}\mathbf{L}_{\mathrm{dq}} + (\mathbf{Y}\mathbf{L}_{\mathrm{dq}})^{\mathrm{T}} \right) \right)\,. \tag{21c}$$

ACKNOWLEDGMENT

The presented investigations were conducted at Corporate Research of Robert Bosch GmbH, Renningen, Germany.

References

[1] M. J. Duran and F. Barrero. "Recent Advances in the Design, Modeling, and Control of Multiphase Machines - Part II". In: *IEEE Transactions on Industrial Electronics* 63.1 (2016), pp. 459–468.

[2] M. Merzoug, F. Naceri, et al. "Comparison of field-oriented control and direct torque control for permanent magnet synchronous motor (pmsm)". In: *World Academy of Science, Engineering and Technology* 45 (2008), pp. 299–304.

[3] M. Preindl and S. Bolognani. "Model predictive direct torque control with finite control set for PMSM drive systems, Part 1: Maximum torque per ampere operation". In: *IEEE Transactions on Industrial Informatics* 9.4 (2013), pp. 1912–1921.

[4] P. Cortés, M. P. Kazmierkowski, R. M. Kennel, D. E. Quevedo, and J. Rodríguez. "Predictive control in power electronics and drives". In: *IEEE Transactions on Industrial Electronics* 55.12 (2008), pp. 4312–4324.

[5] T. Geyer and D. E. Quevedo. "Performance of Multistep Finite Control Set Model Predictive Control for Power Electronics". In: *IEEE Transactions on Power Electronics* 30.3 (Mar. 2015), pp. 1633–1644.

[6] J. Rodriguez, 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". In: *IEEE Transactions on Industrial Informatics* 9.2 (May 2013), pp. 1003–1016.

[7] T. Geyer and D. E. Quevedo. "Multistep finite control set model predictive control for power electronics". In: *IEEE Transactions on Power Electronics* 29.12 (2014), pp. 6836–6846.

[8] P. Karamanakos, T. Geyer, and R. Kennel. "A Computationally Efficient Model Predictive Control Strategy for Linear Systems With Integer Inputs". In: *IEEE Transactions on Control Systems Technology* 24.4 (July 2016), pp. 1463–1471.

[9] J. Raath, D. T. Mouton, and T. Geyer. "Integration of inverter constraints in geometrical quantification of the optimal solution to an MPC controller". In: *2016 IEEE 17th Workshop on Control and Modeling for Power Electronics (COMPEL)*. June 2016, pp. 1–6.

[10] 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: *IECON 2016 - 42nd Annual Conference of the IEEE Industrial Electronics Society*. Oct. 2016, pp. 5039–5045.

[11] B. Stellato, T. Geyer, and P. J. Goulart. "High-speed finite control set model predictive control for power electronics". In: *IEEE Transactions on Power Electronics* 32.5 (2017), pp. 4007–4020.

[12] T. Geyer and D. E. Quevedo. "Multistep direct model predictive control for power electronics—Part 1: Algorithm". In: *2013 IEEE Energy Conversion Congress and Exposition (ECCE)*. IEEE. 2013, pp. 1154–1161.

[13] C. Feller and C. Ebenbauer. "Relaxed logarithmic barrier function based model predictive control of linear systems". In: *IEEE Transactions on Automatic Control* 62.3 (2017), pp. 1223–1238.

[14] C. Feller and C. Ebenbauer. "A stabilizing iteration scheme for model predictive control based on relaxed barrier functions". In: *Automatica* 80 (2017), pp. 328–339.

[15] S. Sager. *Numerical methods for mixed-integer optimal control problems*. Der andere Verlag Tönning, Lübeck, Marburg, 2005.

[16] S. Boyd, N. Parikh, E. Chu, B. Peleato, and J. Eckstein. "Distributed optimization and statistical learning via the alternating direction method of multipliers". In: *Foundations and Trends® in Machine Learning* 3.1 (2011), pp. 1–122.

[17] R. Takapoui, N. Moehle, S. Boyd, and A. Bemporad. "A simple effective heuristic for embedded mixed-integer quadratic programming". In: *International Journal of Control* (2017), pp. 1–11.

[18] A. U. Raghunathan and S. Di Cairano. "Optimal step-size selection in alternating direction method of multipliers for convex quadratic programs and model predictive control". In: *Proceedings of Symposium on Mathematical Theory of Networks and Systems*. 2014, pp. 807–814.