Continuous Control Set Nonlinear Model Predictive Control of Reluctance Synchronous Machines

Andrea Zanelli, Julian Kullick, Hisham Eldeeb, Gianluca Frison, Christoph Hackl, Moritz Diehl

Abstract—In this paper we describe the design and implementation of a current controller for a reluctance synchronous machine based on continuous set nonlinear model predictive control. A simplified experimentally identified grey box model of the flux linkage map is employed in a tracking formulation which is implemented using the high-performance framework acados. The resulting controller is validated in simulation and deployed on a dSPACE real-time system connected with a physical reluctance synchronous machine. Experimental results are presented where the proposed implementation can reach sampling times in the range typical for electrical drives and outperforms state-of-the-art classical control strategies.

Index Terms—predictive control, electric motors, nonlinear systems.

I. INTRODUCTION

In recent years, reluctance synchronous machines (RSMs) have emerged as a competitive alternative to classical synchronous machines (SMs) with permanent magnet (PMSM) or direct current excitation. In addition to the favourable properties of SMs in general, e.g. high efficiency, reliability and compact design, RSMs are easy to manufacture and comparably cheap due to the absence of magnets. Moreover, their anisotropic magnetic path in the rotor, makes them particularly suitable for saliency-based encoderless control [29, 30]. However, a major drawback of the RSM concerning control is its characteristic nonlinearity of the flux linkage, caused by magnetic saturation and cross-coupling effects in the rotor. As a consequence, the machines’ inductances vary significantly with the stator currents. Additional coupling between the stator $d$- and $q$-currents is imposed by the cross-coupling inductances and the coupling of the nonlinear back electro-motive force in the synchronous reference frame, which requires further measurements to be carried out online.

Regarding the control of RSMs, two main concepts have been pursued in the past: (i) Direct Torque Control (DTC) [4, 28] and (ii) field-oriented control (FOC) [3, 31, 46]. While DTC is known for its robustness and fast dynamics [5], it produces a high current distortion leading to torque ripples [6]. In contrast, vector control improves the torque response [42] and the efficiency of the system [26], but good knowledge of the system parameters is required for implementation. In [20], a completely parameter-free adaptive PI controller is proposed which guarantees tracking with prescribed transient accuracy. The controller is applied to current control of (reluctance) synchronous machines, but measurement results are not provided. In [42] and [47], the inductances are tracked online in order to adjust the current references thus achieving a higher control accuracy. In [23], a FOC control scheme is proposed, where the PI control parameters are continuously adapted to the actual system state, which improves the overall current dynamics.

An alternative to classical control approaches is the use of optimization-based control techniques such as model predictive control (MPC). When using MPC, a parametric optimization problem is formulated that exploits a model of the plant to be controlled and enforces constraints while minimizing a certain objective function. Although MPC can in principle improve the control performance and ease the controller design [18], meeting the required sampling times is in general a challenging task due to the high computational burden associated with the solution of the underlying optimization problems.

In order to circumvent this difficulty, several algorithmic strategies have been proposed over the past decade that use different approaches and (potentially) different formulations of the optimal control problems to be solved. Among the possible classifications of methods present in the literature, in the fields of electrical drives and power electronics, a fundamental distinction can be made between what is sometimes referred to as finite (FS-) and continuous control set (CS-) MPC [37, 17].

In FS-MPC, the switch positions of the power converter are regarded as optimization variables leading to mixed-integer programs. In this way, the need for an external modulator is eliminated and the switching sequences are directly determined by the solution to the optimal control problem (hence the name “direct” MPC used in some of the literature on MPC for electrical drives and power converters [17]).

When using CS-MPC instead, we delegate the determination of switching sequences to an external modulator in order to obtain a continuous optimization problem. For this reason, CS-MPC is sometimes referred to as “indirect” MPC [17].
Although the computation times associated with this latter approach scale favourably with prediction horizon length and number of control variables (typically complexity $O(N(n_u + n_x)^3)$ can be achieved, where $N, n_u$ and $n_x$ represent horizon length, number of inputs and states, respectively), for short horizons, strategies based e.g. on sphere decoding algorithms applied to FS-MPC formulations can achieve sufficiently short computation times. On the contrary, CS-MPC is generally regarded as more computationally expensive and it is still, arguably for this reason, largely unexplored [17]. Among the experimental results in the literature obtained with CS-MPC, in [2] a DC-excited synchronous motor is controlled using the real-time iteration method. In [14], a fixed-point iteration scheme is used to control a permanent magnet synchronous machine (PMSM). Among applications leveraging linear-quadratic CS-MPC we mention the work in [12] in which permanent magnet synchronous machines and induction machines are controlled using explicit model predictive control.

A. Contribution

In this paper, we describe the design and implementation details together with experimental results of a nonlinear CS-MPC controller (CS-NMPC) for an RSM. The contributions of the present work are:

- We describe the design and implementation details of a tracking CS-NMPC formulation that relies on the software package acados, which is capable of achieving timings in the microsecond time scale necessary to control the electrical drive.
- We propose the use of a simple grey box model for the flux maps of RSMs that can be used for online applications where computation times are of key importance.
- Finally, we present simulation and experimental results that confirm the validity of the proposed control formulation and its implementation and its superior performance in comparison with state-of-the-art methods from the field of classical control. This is, to the best of the authors’ knowledge, one of the earliest experimentally validated applications of CS-NMPC to an RSM.

II. BACKGROUND ON RSMs AND NMPC

In order to facilitate the discussion of the design and implementation of the proposed controller, in the following, mathematical models of RSMs and voltage source inverters (VSI) will be derived and numerical methods for NMPC will be introduced. Note that the argument $(t)$, used to denote dependence on time, is dropped for the sake of readability.

A. Generic model of the RSM

The machine model in the synchronously rotating $(d,q)$-reference frame is given by [21 Chap. 14]

\[
\begin{align*}
    u_s &= R_s i_s + \omega \begin{pmatrix}
    -1 & 0 \\
    0 & 1
    \end{pmatrix} \psi_s(i_s) + \frac{d}{dt}\psi_s(i_s), \\
    \frac{d}{dt}\omega &= \frac{n_p}{\Theta} \left[ m_m(i_s) - m_1 \right],
\end{align*}
\]

where $u_s := (u_s^d, u_s^q)\top$ are the applied stator voltages, $R_s$ is the stator resistance, $i_s := (i_s^d, i_s^q)\top$ are the stator currents and $\psi_s := (\psi_s^d, \psi_s^q)\top$ are the stator flux linkages (functions of $i_s$). The $(d,q)$-reference frame rotates with electrical angular frequency $\omega = n_p \omega_m$ of the rotor where $n_p$ is the number of pole pairs and $\omega_m$ denotes the mechanical angular frequency of the machine. Furthermore, $\Theta$ is the total moment of inertia, $m_m(i_s) := \frac{1}{2} n_p (i_s)\top J \psi_s(i_s)$ is the electro-magnetic machine torque, and $m_1$ represents an external (time-varying) bounded load torque.

In order to formulate an optimal control problem, the flux dynamics can be described, based on [1], with the following differential algebraic equation (DAE):

\[
\begin{align*}
    \frac{d}{dt}\psi_s &= u_s - R_s i_s - \omega J \psi_s + v, \\
    0 &= \psi_s - \Psi_s(i_s),
\end{align*}
\]

where $\Psi_s := (\Psi_s^d, \Psi_s^q)\top : \mathbb{R}^2 \rightarrow \mathbb{R}^2$ represent the identified flux maps and $v := (v^d, v^q)\top$ are additive disturbances which will be used in an offset-free NMPC formulation (see Section II-F).

Based on the available flux maps computed through finite element method (FEM), we obtained a continuously differentiable model by fitting a simple grey box model. Due to
their low-dimensionality and simple structure, we propose the following parametrization of the flux maps:

$$\Psi_s^d(i_{s,k}, \theta_d) = \frac{c_s^d}{\sqrt{2\pi\sigma_d}} \exp\left(\frac{-i_{s,k}^d}{2\sigma_d^2}\right) \tan(c_s^d i_{s,k}) + c_s^d i_{s,k},$$

$$\Psi_s^q(i_{s,k}, \theta_q) = \frac{c_s^q}{\sqrt{2\pi\sigma_d}} \exp\left(\frac{-i_{s,k}^q}{2\sigma_d^2}\right) \tan(c_s^q i_{s,k}) + c_s^q i_{s,k},$$

with unknown coefficients $\theta_d := (c_s^d, c_s^q, \sigma_d)$ and $\theta_q := (c_s^d, c_s^q, \sigma_q)$. The numerical values of the coefficients can be computed by solving the following (decoupled) nonlinear least-squares problems:

$$\min_{\theta_d} \sum_{j=1}^{m} \sum_{k=1}^{n} \left(\Psi_s^d(i_{s,j,k}, \theta_d) - \hat{\Psi}_s^d(i_{s,j,k}, \theta_d)\right)^2$$

$$\min_{\theta_q} \sum_{j=1}^{m} \sum_{k=1}^{n} \left(\Psi_s^q(i_{s,j,k}, \theta_q) - \hat{\Psi}_s^q(i_{s,j,k}, \theta_q)\right)^2$$

where $\vec{i}_{s,j,k}^d$, $\vec{i}_{s,j,k}^q$, respectively, are the $j$-th and $k$-th data point of the current base vectors from the FEM data $\vec{\Psi}_s^d$ and $\vec{\Psi}_s^q$. The fitting problems have been solved with the MATLAB Curve Fitting Toolbox and the resulting fits are shown in Figure 4.

### B. Model of the two-level VSI

The machine is supplied by a two-level voltage source inverter (VSI), which – on average over one switching period $T_s$ – translates a given voltage reference

$$u_{s,\text{ref}} := (u_{s,\text{ref},a}, u_{s,\text{ref},b}, u_{s,\text{ref},c})$$

in the stationary $s = (\alpha, \beta)$-reference frame into the inverter output voltage $u_s$, i.e.

$$u_s^k(T_s) \approx u_{s,\text{ref}}^k((k-1)T_s), \quad k \in \mathbb{N}.$$  

Since a two-level voltage source inverter may produce a total of eight unique switching vectors, i.e. $s_{a}^{abc} := (s_{a}^a, s_{b}^a, s_{c}^a) \in \{000, 001, 010, 100, 011, 101, 110, 111\}$, the typical voltage hexagon in the $\alpha\beta$-plane is obtained (see Figure 2), where

$$u_s^a = \kappa u_{dc}^e \begin{bmatrix} \frac{1}{\sqrt{3}} & 0 & -\frac{1}{\sqrt{3}} \\ 0 & 1 & -1 \end{bmatrix} s_{a}^{abc}$$

depends on the switching vector $s_{a}^{abc}$ and the Clarke-factor $\kappa \in \{2/3, \sqrt{2}/3\} \quad [21]$ Chap. 14. Using space-vector modulation (SVM) to generate the switching vector, any voltage reference within the circle of radius $u_{dc}/\sqrt{3}$ can be realized, with $u_{dc}$ denoting the (assumed constant) DC link voltage. Finally, the inverter output voltage is transformed into the rotating $(d,q)$-reference frame using the inverse Park transformation, i.e.

$$u_s = \begin{bmatrix} u_s^d \\ u_s^q \end{bmatrix} = \begin{bmatrix} \cos(\phi) & -\sin(\phi) \\ \sin(\phi) & \cos(\phi) \end{bmatrix} u_s^e, \quad \phi = \frac{2}{\sqrt{3}}.$$

From now on, we will only refer to currents, fluxes and voltages applied to the stator and in the $(d,q)$-frame, we will simplify the notation by dropping the associated subscript such that, for example, $i = (i^d, i^q)$ denotes the stator currents in the $(d,q)$-frame.

### C. Nonlinear model predictive control

NMPC is an optimization-based control strategy that allows one to tackle control problems involving potentially nonlinear dynamics, constraints and objectives by solving online a series of parametric nonlinear programs (NLP). Due to the computational challenge of solving NLPs within the required sampling times, NMPC has initially found application in the chemical industry and in the field of process control in general [43], where relatively slow dynamics allow for sufficiently long sampling times. In more recent times, due to the development of increasingly efficient numerical methods and software implementation and due to the growing computational power of embedded control units, NMPC has gradually become a viable approach for applications with much shorter computation times. Among other recent works that reported on the successful application of MPC to control systems with sampling times in the range of milli- and microsecond we mention [11] [48].

In this paper, we will regard the following standard tracking formulation, where the squared deviation of fluxes $\psi$ and voltages $u_s$ from properly defined steady-state references are penalized:

$$\min_{\psi_0, \ldots, \psi_N} \frac{1}{2} \sum_{i=0}^{N-1} \left\| \psi_i - \bar{\psi} \right\|^2 + \frac{1}{2} \left\| \psi_N - \bar{\psi} \right\|^2 W_N$$

s.t.

$$\psi_0 = \psi_c = 0,$$

$$g(\psi_i, u_i, \omega_c, v_c) - \psi_{i+1} = 0, \quad i = 0, \ldots, N - 1,$$

$$u_i^+ \leq \left( \frac{u_{dc}}{\sqrt{3}} \right)^2, \quad i = 0, \ldots, N - 1,$$

$$Cu_i \leq c, \quad i = 0, \ldots, N - 1,$$

where $g$ describes the discretized dynamics obtained by integrating the differential-algebraic model in [4] using the Gauss-Legendre collocation method of order 2 [24] assuming
constant angular velocity $\omega_c$ and disturbances $v_d$. The variables $\vec{u}$ and $\vec{u}$ denote the steady-state references computed for a given desired torque using a maximum-torque-per-Ampere (MTPA) criterion [13]. Given the flux maps obtained from FEM data in Figure 1, it is possible to compute off-line lookup tables (LUTs) that contain the MTPA reference fluxes and voltages for a finite number of values of the target torque in a specified range. The LUTs are then interpolated online in order to compute approximate values of $\vec{u}$ and $\vec{u}$ associated with the specified target torque $\bar{m}_f$. We use the notation $\|x\|^2_P = x^T P x$, for some positive definite matrix $P$, to denote the squared $P$-weighted norm of the vector $x$. Finally, $C$ and $c$ define polytopic constraints (“safety” constraints later) that are meant to be always inactive at any local solution of (9), but can mitigate constraint violation of intermediate SQP iterates.

**Remark 1.** Notice that the actual dynamics of the system involve a coupling of mechanical ($\omega$) and electrical states ($\psi$). It is however common, given the large difference between associated time constants, to assume a constant angular velocity $\omega$ when designing controllers. In our case, it will allow to use much shorter prediction horizons since we do not require the OCP in (9) to steer the speed of the motor to the desired reference, but only fluxes which directly map to currents and, for a given speed, to torques.

Problem (9) is used to define an implicit feedback policy that requires the solution of an instance of the parametric NLP at every sampling time, where the value of the parameter $x$ is given by the current estimate of the system’s state. The resulting solutions are feasible with respect to the constraints and minimize (at least locally) the cost function. Nominal and resulting solutions are feasible with respect to the constraints of the problem in order to better capture control design requirements. However, for the application discussed in this paper, the nonlinear least-squares problem described in (9) is general enough.

**D. Numerical methods and software for NMPC**

In order to be able to solve problem (9) within the available computation time, the use of efficient numerical methods is fundamental. First, since (9) is obtained through a multiple-shooting discretization strategy, an efficient way of computing evaluations of the discretized dynamics $f$ and of its first (and eventually second order) needs to be available. This is commonly achieved by means of numerical integration of the ordinary differential equation (ODE) or algebraic differential equations (DAE) describing the dynamics of the system. Although for ODEs explicit and implicit integration methods can be used, for DAEs, as for the system under consideration (see Section II-A), implicit schemes are generally necessary that involve the solution of nonlinear root-finding problems via Newton-type iterations. In the context of NMPC, efficient strategies and tailored implementations are available that rely on, for example, exploiting the structure of the model [16] [38], on reuse of Jacobian factorizations and efficient sensitivity generation [41] on lifting-based formulations [39] and on inexact iterations [40].

Once the linearization is carried out, one generally needs to solve structured linear systems that can be used to compute the solution to a quadratic program (QP) as in sequential quadratic programming (SQP), compute the update defined by an interior-point method or the one used by other various strategies, e.g. first-order methods. Since the description of the details of the different available approaches to solve (9) goes well beyond the scope of this work, we will focus, in the following, on the SQP strategy, which constitutes the basis for the real-time iteration (RTI) method used in this application.

When using SQP, a sequence of structured QPs of the following form needs to be solved:

$$\min_{\bar{u}_0, \ldots, \bar{u}_{N-1}} \frac{1}{2} \sum_{i=0}^{N-1} \left[ \begin{array}{c} s_i \\ N \end{array} \right]^{\top} H \left[ \begin{array}{c} s_i \\ N \end{array} \right] + \frac{1}{2} \left[ \begin{array}{c} s_n \\ 1 \end{array} \right]^{\top} H_N \left[ \begin{array}{c} s_n \\ 1 \end{array} \right].$$

s.t.

$$s_0 - x = 0,$$

$$s_{i+1} = A x_i + B u_i + c, \quad i = 0, \ldots, N - 1,$$

$$C u_i + D x_i + e \leq 0, \quad i = 0, \ldots, N - 1,$$

$$D_N x_N + e_N \leq 0,$$

where $A \in \mathbb{R}^{n_x \times n_x}$, $B \in \mathbb{R}^{n_x \times n_u}$ and $c \in \mathbb{R}^{r}$ define the linearized dynamics obtained through numerical integration and where $C \in \mathbb{R}^{n_u \times n_u}$, $D \in \mathbb{R}^{n_u \times n_x}$, $e \in \mathbb{R}^{r_x}$ and $C_N \in \mathbb{R}^{n_x \times n_u}$, $e_N \in \mathbb{R}^{r_x}$ define the linearized constraints. Finally the matrices

$$H = \left[ \begin{array}{cc} Q & S \\ S^{\top} & R \end{array} \right]$$

$$q$$

$$\pi$$

$$r$$

$$q^{\top}$$

$$r^{\top}$$

$$0$$

and

$$H_N = \left[ \begin{array}{c} [Q_N^{\top} \\ q_N] \\ [0] \end{array} \right],$$

with $Q \in \mathbb{R}^{n_x \times n_x}$, $S \in \mathbb{R}^{n_x \times n_u}$, $R \in \mathbb{R}^{n_u \times n_u}$, $r \in \mathbb{R}^{n_u}$, $q \in \mathbb{R}^{r_x}$ obtained through linearization of the cost, define the cost of the QP. The matrices and vectors defining the QP (10) are computed based on the linearization associated with the given current primal-dual iterate $z^k = (s^k, u^k, \lambda^k, \mu^k)$ (where $\lambda$ and $\mu$ represent the Lagrange multipliers associated with the equality and inequality in (9), respectively) and, after solving (10), the iterate is updated:

$$z^{k+1} \leftarrow z^k + \alpha (z_{QP}^k - z^k),$$

where $z_{QP}^k$ represents the primal-dual solution of the QP associated with the linearization point $z^k$ and $\alpha > 0$ is the step size, which can be adjusted to achieve convergence. Under standard assumptions [34], the iterates in (12) converge to a local minimum of (9).

Due to the computational burden associated with the solution of QPs and re-linearization of the original NLP in (9), several approximate strategies can be used that can significantly reduce computation times (e.g. [42], [19], [15]). In this work, we will use the RTI strategy [8] [10], which relies on a single SQP iteration in order to provide an approximate feedback
law. The RTI has been implemented in the software packages MUSCOD-II [11], ACADO [25] and in its successor acados [44], which is shortly described in the next subsection.

E. The acados framework

The high-performance software package acados [44] provides a modular framework for NMPC and moving horizon estimation (MHE). It consists of a C library that implements building blocks needed to solve NLPs arising from NMPC and MHE formulations. It relies on the high-performance linear algebra package BLASFEO and on the quadratic program (QP) solver HPIPM and contains efficient implementations of explicit and implicit integration methods. Moreover, it interfaces a number of QP solvers such as qpOASES, qpDUNES and OSQP and it provides high-level Python and MATLAB interfaces. Through these interfaces, one can conveniently specify optimal control problems and code-generate a self-contained C library that implements the desired solver and can be easily deployed onto embedded control units such as dSPACE using the automatically generated C wrapper and S-Function. The code-generation takes place through templated C code which is rendered by the Tera templating engine written in Rust. In this way, human-readable C code can be generated that facilitates the deployment on the target unit.

F. NMPC offset-free tracking formulation

In order to achieve offset-free regulation, we adopt the standard strategies discussed, for example, in [36]. In particular, we use the following augmented dynamics to design an extended Kalman filter (EKF):

\[
\begin{align*}
\frac{d}{dt} \psi & = u - Ri - \omega J \psi + v, \\
\frac{d}{dt} v & = 0, \\
0 & = \psi - \Psi(i),
\end{align*}
\]

where the disturbance state \( v \) is introduced and we assume that pseudo-measurements \( y \) are available through the interpolated FEM flux maps:

\[
y = \psi_m = \Psi(i_m),
\]

while current measurements \( i_m \) are physically carried out on the machine. An EKF is designed using (13) and (14) which uses flux measurements to estimate fluxes \( \psi_c \) and disturbances \( v_c \). Notice that the angular velocity \( \omega_c \) is estimated externally and is considered as a constant-over-time parameter that is updated at every sampling time. In [35, Theorem 14] a streamlined version of the results from [32, 33, 36] is presented, where under the assumptions, among others, of observability of the augmented dynamics [13] and asymptotically constant disturbances \( v \), the steady-state of the closed-loop system is offset-free.

III. IMPLEMENTATION AND SIMULATION RESULTS

An RTI strategy [9], where a single QP of an SQP algorithm is carried out per sampling time, is used to solve (9). In particular, the generalized Gauss-Newton Hessian approximation proposed in [44] is used. In this way, the (positive) curvature contribution coming from the convex spherical constraints on voltages can be exploited in order to improve the Hessian approximation used in the QP subproblems. Although in our experience this improves a lot the convergence of the RTI iterates on this specific problem, the approximate feedback law can, from time to time, be largely infeasible with respect to the nonlinear spherical constraints (recall that the intermediate full-step SQP iterates are feasible only with respect to linear constraints). Since a-posteriori projection of the control actions onto the feasible set can deteriorate control performance, we add extra polytopic “safety” constraints (defined by \( C \) and \( c \) in [9]) around the spherical ones in order to ensure that constraint violation will be bounded at any successfully computed iterate.

In order to be able to meet the short sampling times required to control the electrical drive, we use a prediction horizon of \( T_h = 3.2 \) ms obtained with 2 shooting nodes \( N = 2 \) and we use the QP solver qpOASES, which is particularly suited for problems with short horizons [27]. For both simulation and experimental results the controller is run at 4 kHz.

In the following Sections, we will be discussing simulation and experimental results obtained using the above described RTI strategy to solve (9) with acados.

In order to validate, first in simulation and then experimentally, the proposed approach, we regard a setting where the RSM is connected to a permanent synchronous machine (PMSM) which can be used to simulate different load conditions. The CS-NMPC controller has been implemented in acados using its Python interface and integrated in a Simulink model that makes use of a high-fidelity model of the system to be controlled including a model of the PMSM and of the two-level VSI described in Section II. Moreover,
we have implemented an EKF based on the augmented model \(^\text{(13)}\) using the implicit integrators available in \textit{acados}.

We set the PMSM such that it maintains a constant rotational speed and we change the torque reference fed to the RSM’s controller in order to assess the tracking performance of the proposed controller. We compare the closed-loop trajectories obtained with the ones achieved when using instead the gain-scheduled PI controller with anti-windup presented in \cite{22}. The current trajectories obtained with the CS-NMPC and PI controller are reported in Figure \textbf{6} (similarly for input trajectories in Figure \textbf{5}). It can be clearly seen from the snapshots of the trajectories reported in Figure \textbf{7} that the transient can be drastically improved when operating near the boundaries of the feasible control set.

\section{IV. EXPERIMENTAL RESULTS}

The presented NMPC scheme has been implemented and verified experimentally on a custom-built 9.6 kW RSM (Courtesy of Prof. Maarten Kamper, Stellenbosch University, South Africa) with the parameters

\begin{align}
R_s &= 0.4 \, \Omega, \quad \omega_{m,\text{nom}} = 157.07 \, \text{rad} \, \text{s}^{-1}, \\
m_{m,\text{nom}} &= 61 \, \text{Nm}, \quad \dot{i}_{s,\text{max}} = 29.7 \, \text{A}, \\
\hat{u}_{s,\text{max}} &= 580 \, \text{V},
\end{align}

and the nonlinear flux linkage maps as depicted in Figure \textbf{1} (maps were obtained from FEM). The overall laboratory setup is depicted in Figure \textbf{3} and comprises the \textit{dSPACE} real-time system with processor board DS1007 and various extensions and I/O boards, two 22 kW SEW inverters in back-to-back configuration sharing a common DC-link, the HOST-PC running MATLAB/Simulink with RCPHIL R2017 and \textit{dSPACE ControlDesk} 6.1p4 for rapid-prototyping, data acquisition and evaluation, the custom-built 9.6 kW RSM as device under test and a 14.5 kW SEW PMSM as load machine. The DR2212 torque sensor allows to measure the mechanical torque too, but it was not used. The controller based on the formulation described in Section \textbf{III} and implemented using the \textit{acados} framework has been deployed on the \textit{dSPACE} unit connected to the physical RSM.

Two different experiments have been carried out. In the first case we used the PMSM to maintain the nominal rotational speed of the rotor \((157 \, \text{rad} \, \text{s}^{-1})\) and different torque references have been fed to the RSM controllers under analysis. In the second case, on the contrary, a fixed torque reference was fed to the CS-NMPC and PI controller and the speed of the shaft was changed by the PMSM.

The closed-loop trajectories for the conducted experiments are reported in Figure \textbf{8}-\textbf{9}. Similarly to the results obtained in simulation, when tracking torque steps at a fixed speed (in Figure \textbf{8}), the proposed CS-NMPC controller achieves better tracking performance than the gain-scheduled PI controller. Notice that there is a substantial discrepancy between simulation and experimental results right after the third torque step, at \(t = 0.75 \, \text{s}\), due to a drop in the DC-link voltage. In fact, in the presence of a sudden change in the torque reference, the voltage of the DC-link capacitor can drop if the recharging rate is slower the discharging rate (behavior not modelled in simulation). In the second experiment, in Figure \textbf{9} similar conclusions can be drawn since the desired currents can be more closely tracked when using CS-NMPC.

\section{V. CONCLUSIONS}

In this paper we present simulation and experimental results obtained with a CS-NMPC torque controller for RSMs. As
Figure 6: Current steps at 157 rad/s: simulation results obtained using the CS-NMPC (left) and gain-scheduled PI controller (right).

Figure 7: Current steps at 157 rad/s: simulation (left) and experimental (right) results obtained using CS-NMPC and the gain scheduled PI controller: actual (thin) and filtered (thick). Using the proposed CS-NMPC controller, the settling time can be substantially reduced.

opposed to most successful implementations present in the literature, that use instead FS-MPC/NMPC, we show the effectiveness and real-time feasibility of the continuous control set approach. We show that, using the software implementation of the real-time iteration method for NMPC available in the software package acados, it is possible to deploy the proposed controller on embedded hardware and to meet the challenging sampling times typically required to control electrical drives. We discuss implementation details and report on simulation as well as experimental results which show that the proposed approach can outperform state-of-art control methods.

Future research will involve the investigation of novel numerical methods, e.g. the real-time first-order methods proposed in [50], to speed up the computation times, which are currently still rather long and do not easily allow for extensions of the optimal control formulations (e.g longer horizons, state or input spaces of higher dimension, etc).

REFERENCES

[1] T. Albin, D. Ritter, N. Liberda, R. Quirynen, and M. Diehl. In-vehicle realization of nonlinear MPC for gasoline two-stage turbocharging airpath control. *IEEE Transactions on Control Systems Technology*, pages 1–13, 2017.

[2] Thomas J. Besselmann, Sture Van de moortel, Stefan Almér, Pieder Jörg, and Hans Joachim Ferreau. Model predictive control in the multi-megawatt range. *IEEE Transactions on Industrial Electronics*, 63(7):4641–4648, 2015.

[3] R.E. Betz, R. Lagerquist, M. Jovanovic, T.J.E. Miller, and R.H. Middleton. Control of synchronous reluctance machines. *IEEE Transactions on Industry Applications*, 29(6):1110–1122, 1993.

[4] I Boldea, ZX Fu, and SA Nasar. Torque vector control (TVC) of axially-laminated anisotropic (ALA) rotor reluctance synchronous motors. *Electric machines and power systems*, 19(3):381–398, 1991.

[5] S. Bolognani, L. Peretti, and M. Zigliotto. Online
MTPA control strategy for DTC synchronous-reluctance-motor drives. *IEEE Transactions on Power Electronics*, 26(1):20–28, 2011.

[6] Abdesselam Chikhi, Mohamed Djarallah, and Khaled Chikhi. A comparative study of field-oriented control and direct-torque control of induction motors using an adaptive flux observer. *Serbian Journal of Electrical Engineering*, 7(1):41–55, 2010.

[7] P. Cortes, M.P. Kazmierkowski, R.M. Kennel, D.E. Quevedo, and J. Rodriguez. Predictive control in power electronics and drives. *IEEE Transactions on Industrial Electronics*, 55(12):4312–4324, 2008.

[8] M. Diehl. *Real-Time Optimization for Large Scale Nonlinear Processes*. PhD thesis, University of Heidelberg, 2001.

[9] M. Diehl, H.G. Bock, and J.P. Schlöder. Newton-type methods for the approximate solution of nonlinear programming problems in real-time. In G. Di Pillo and A. Murli, editors, *High Performance Algorithms and Software for Nonlinear Optimization*, pages 177–200. Kluwer Academic Publishers B.V., 2002.

[10] M. Diehl, R. Findeisen, and F. Allgöwer. A stabilizing real-time implementation of nonlinear model predictive control. In L. Biegler, O. Ghatts, M. Heinkenschloss,
D. Keyes, and B. van Bloemen Waanders, editors. Real-Time and Online PDE-Constrained Optimization, pages 23–52. SIAM, 2007.

[11] M. Diehl, D.B. Leineweber, and A.A.S. Schäfer. MUSCOD-II Users’ Manual. IWR-Preprint 2001-25, University of Heidelberg, 2001.

[12] A. Domahidi, S. Mariethoz, and M. Morari. High-bandwidth explicit model predictive control of electrical drives. IEEE Transactions on Industry Applications, 48(6):1980–1992, November 2012.

[13] Hisham Eldeeb, Christoph M. Hackl, Lorenz Horlbeck, and Julian Kullick. A unified theory for optimal feed-forward torque control of anisotropic synchronous machines. International Journal of Control, 91(10):2273–2302, 2017.

[14] T. Englert and K. Graichen. A fixed-point iteration scheme for model predictive control of pmsms. In Proceedings of the 6th IFAC Conference on Nonlinear Model Predictive Control, volume 51, pages 568–573, Madison, Wisconsin, USA, August 2018.

[15] C. Feller and C. Ebenbauer. Relaxed logarithmic barrier function based model predictive control of linear systems. IEEE Transactions on Automatic Control, 62(3):1223–1238, March 2017.

[16] Jonathan Frey, Rien Quirynen, Dimitris Kouzoupis, Gi-anluca Frison, Jens Geisler, Axel Schild, and Moritz Diehl. Detecting and exploiting Generalized Nonlinear Static Feedback structures in DAE systems for MPC. In Proceedings of the European Control Conference (ECC), 2019.

[17] Tobias Geyer. Model Predictive Control of High Power Converters and Industrial Drives. John Wiley & Sons, 2016.

[18] Tobias Geyer, Georgios Papafotiou, and Manfred Morari. Model predictive direct torque control – Part I: Concept, algorithm, and analysis. IEEE Transaction on Industrial Electronics, 56(6):1894–1905, 2009.

[19] K. Graichen and A. Kugi. Stability and incremental improvement of suboptimal mpc without terminal constraints. IEEE Transactions on Automatic Control, 55(11):2576–2580, 2010.

[20] Christoph M. Hackl. Current PI-funnel control with anti-windup for synchronous machines. In Proceedings of the 54th IEEE Conference on Decision and Control, pages 1997–2004, 2015.

[21] Christoph M. Hackl. Non-identifier Based Adaptive Control in Mechatronics. Springer International Publishing, 2017.

[22] Christoph M. Hackl, Maarten J. Kamper, Julian Kullick, and Joshua Mitchell. Nonlinear PI current control of reluctance synchronous machines. arxiv.org/pdf/1512.09301v1, 2015.

[23] C.M. Hackl, M.J. Kamper, J. Kullick, and J. Mitchell. Current control of reluctance synchronous machines with online adjustment of the controller parameters. In 2016 IEEE 25th International Symposium on Industrial Electronics (ISIE). IEEE, jun 2016.

[24] Ernst Hairer and Gerhard Wanner. Solving Ordinary Differential Equations II: Stiff and Differential–Algebraic Problems. Springer Series in Computational Mathematics. Springer-Verlag, Berlin, 2nd revised edition, 2004.

[25] B. Houska, H. J. Ferreau, and M. Diehl. ACADO toolkit – an open source framework for automatic control and dynamic optimization. Optimal Control Applications and Methods, 32(3):298–312, 2011.

[26] Marten J. Kamper, F.S. van der Merwe, and S. Williamson. Direct finite element design optimisation of the cageless reluctance synchronous machine. IEEE Transactions on Power Conversion, 11(3):547–555, 1996.

[27] D. Kouzoupis, G. Frison, A. Zanelli, and M. Diehl. Recent advances in quadratic programming algorithms for nonlinear model predictive control. Vietnam Journal of Mathematics, 46(4):863–882, 2018.

[28] R. Lagerquist, I. Boldea, and T.J.E. Miller. Sensorless control of the synchronous reluctance motor. IEEE Transactions on Industry Applications, 30(3):673–682, 1994.

[29] Peter Landsmann, Ralph Kennel, Hugo W. de Kock, and Marten J. Kamper. Fundamental saliency based encoderless control for reluctance synchronous machines. In Proceedings of the XIX International Conference on Electrical Machines (ICEM), pages 1–7, 2010.

[30] Peter Landsmann, Dirk Paulus, Peter Stolze, and Ralph Kennel. Reducing the parameter dependency of encoderless predictive torque control for reluctance machines. In Proceedings of the IEEE International Symposium on Sensorless Control for Electrical Drives (SLED), pages 93–99, 2010.

[31] T. Matsuo and T.A. Lipo. Field oriented control of synchronous reluctance machine. In Power Electronics Specialists Conference, 1993. PESC ’93 Record., 24th Annual IEEE, pages 425–431, Jun 1993.

[32] Manfred Morari and Urban Maeder. Nonlinear offset-free model predictive control. Automatica, 48(9):2059–2067, 2012.

[33] K. R. Muske and T. A. Badgwell. Disturbance modeling for offset-free linear model predictive control. Journal of Process Control, 12:617–632, 2002.

[34] J. Nocedal and S. J. Wright. Numerical Optimization. Springer Series in Operations Research and Financial Engineering. Springer, 2 edition, 2006.

[35] G. Panuncioa, M. Gabiccini, and A. Artoni. Offset-free MPC explained: novelites, subtleties, and applications. In Proceedings of the 5th IFAC Conference on Nonlinear Model Predictive Control, volume 48, pages 342–251, 2015.

[36] G. Panuncioa and J.B. Rawlings. Disturbance Models for Offset-Free Model-Predictive Control. AIChE Journal, 49:426–437, 2003.

[37] D. E. Quevedo, R. Aguiler, and T. Geyer. Model predictive control for power electronics applications. In S. V. Rakovic and W. S. Levine, editors, Handbook of Model Predictive Control, pages 551–580. Birkhäuser, Cham, 2019.

[38] R. Quirynen, S. Gros, and M. Diehl. Efficient NMPC for
nonlinear models with linear subsystems. In Proceedings of the IEEE Conference on Decision and Control (CDC), pages 5101–5106, 2013.

[39] R. Quirynen, S. Gros, and M. Diehl. Lifted implicit integrators for direct optimal control. In Proceedings of the IEEE Conference on Decision and Control (CDC), pages 3212–3217, 2015.

[40] R. Quirynen, S. Gros, and M. Diehl. Inexact Newton-type optimization with iterated sensitivities. SIAM Journal on Optimization, 28(1):74–95, 2018.

[41] R. Quirynen, M. Vukov, and M. Diehl. Auto generation of implicit integrators for embedded NMPC with microsecond sampling times. In Mircea Lazar and Frank Allgöwer, editors, Proceedings of the 4th IFAC Nonlinear Model Predictive Control Conference, pages 175–180, 2012.

[42] E.M. Rashad, T.S. Radwan, and M.A Rahman. A maximum torque per ampere vector control strategy for synchronous reluctance motors considering saturation and iron losses. In Conference Record of the 2004 IEEE Industry Applications Conference (39th IAS Annual Meeting), volume 4, pages 2411–2417, 2004.

[43] J. B. Rawlings, D. Q. Mayne, and M. M. Diehl. Model Predictive Control: Theory, Computation, and Design. Nob Hill, 2nd edition edition, 2017.

[44] Robin Verschueren, Gianluca Frison, Dimitris Kouzoupis, Niels van Duijkeren, Andrea Zanelli, Rien Quirynen, and Moritz Diehl. Towards a modular software package for embedded optimization. In Proceedings of the IFAC Conference on Nonlinear Model Predictive Control (NMPC), 2018.

[45] Robin Verschueren, Niels van Duijkeren, Rien Quirynen, and Moritz Diehl. Exploiting convexity in direct optimal control: a sequential convex quadratic programming method. In Proceedings of the IEEE Conference on Decision and Control (CDC), 2016.

[46] Longya Xu, Xingyi Xu, Thomas A. Lipo, and Donald W. Novotny. Vector control of a synchronous reluctance motor including saturation and iron losses. IEEE Transactions on Industrial Applications, 27(5):977–985, 1991.

[47] S. Yamamoto, J.B. Adawey, and T. Ara. Maximum efficiency drives of synchronous reluctance motors by a novel loss minimization controller considering cross-magnetic saturation. In Proceedings of the 2009 IEEE Energy Conversion Congress and Exposition, pages 288–293, 2009.

[48] A. Zanelli, G. Horn, G. Frison, and Moritz Diehl. Nonlinear model predictive control of a human-sized quadrotor. In Proceedings of the European Control Conference (ECC), pages 1542–1547, 2018.

[49] A. Zanelli, R. Quirynen, and M. Diehl. Efficient zero-order nmpc with feasibility and stability guarantees. In Proceedings of the European Control Conference (ECC), Naples, Italy, June 2019.

[50] A. Zanelli, Q. Tran-Dinh, and M. Diehl. Contraction estimates for abstract real-time algorithms for nmpc. In Proceedings of the IEEE Conference on Decision and Control (CDC), Nice, France, Dec 2019. (accepted).