Nonlinear Efficiency-Optimal Model Predictive Torque Control of Induction Machines

Marius Becker 1, Marius Stender 2, and Oliver Wallscheid 2

1Paderborn University
2Affiliation not available

December 7, 2023

Abstract

Induction machines (IMs) are widely used in many applications, e.g., electric vehicles or industrial automation, which motivates efficiency-optimal operation for the sake of energy and cost savings. However, a loss-minimal IM control leads to reduced flux operation at partial load, i.e., achievable torque dynamics after load steps are physically limited due to the IM’s large rotor time constant. Hence, designing the IM torque control for minimal settling time is of particular importance to allow both an efficient and sufficiently dynamic drive operation. Against this background, a model predictive control (MPC) framework is proposed, which utilizes a precise model of the IM covering magnetic saturation, iron losses, skin effect and thermal influences. As this model is highly nonlinear, it is iteratively linearized along the predicted control trajectory so that a computationally efficient quadratic program can be defined for the MPC problem. To enable sufficiently long prediction horizons, a hierarchical control structure with the stator currents as the actuation variables of the model predictive torque control is utilized. Thanks to this scalable control approach, the proposed framework can be easily extended to multi-machine drive systems covering higher-dimensional problem spaces. Empirical tests validate the feasibility of the proposed approach both for single as well as multi-machine drive applications.
Nonlinear Efficiency-Optimal Model Predictive Torque Control of Induction Machines

Marius Becker*, Marius Stender, Oliver Wallscheid

Power Electronics and Electrical Drives, Paderborn University
Paderborn, Germany
*becker@lea.uni-paderborn.de

Abstract—Induction machines (IMs) are widely used in many applications, e.g., electric vehicles or industrial automation, which motivates efficiency-optimal operation for the sake of energy and cost savings. However, a loss-minimal IM control leads to reduced flux operation at partial load, i.e., achievable torque dynamics after load steps are physically limited due to the IM’s large rotor time constant. Hence, designing the IM torque control for minimal settling time is of particular importance to allow both an efficient and sufficiently dynamic drive operation. Against this background, a model predictive control (MPC) framework is proposed, which utilizes a precise model of the IM covering magnetic saturation, iron losses, skin effect and thermal influences. As this model is highly nonlinear, it is iteratively linearized along the predicted control trajectory so that a computationally efficient quadratic program can be defined for the MPC problem. To enable sufficiently long prediction horizons, a hierarchical control structure with the stator currents as the actuation variables of the model predictive torque control is utilized. Thanks to this scalable control approach, the proposed framework can be easily extended to multi-machine drive systems covering higher-dimensional problem spaces. Empirical tests validate the feasibility of the proposed approach both for single as well as multi-machine drive applications.

Index Terms—induction machine, model predictive torque control, electrical-thermal model, gray-box model, multi-machine drive system, operating strategy

I. INTRODUCTION

There are certain requirements for a torque control of electric machines: On the one hand, the torque has to be controlled as accurately and as fast as possible, e.g., to improve the performance of overlaid speed controllers. On the other hand, the efficiency has to be as high as possible to save energy and costs. These requirements also apply to multi-machine drive systems (MMDS) that are used, e.g., in electric vehicles or in industrial automation applications.

A. State of the Art

In literature, there exist approaches which deal with the steady-state loss optimization of an induction machine (IM) for a given reference torque with steady-state flux linkage maps [1] or based on an equivalent circuit diagram (ECD) with concentrated elements [2]. The basic idea was introduced in [3] where the optimal operating point (OP) is calculated online by approximations of nonlinear equations and reduction of the problem to the solution of a polynomial of fourth order. By iterative solution of the given problem, the desired optimal reference current is obtained. Alternatively, loss-optimal operating points can be pre-calculated offline and stored into look-up tables [4].

In general, for steady-state loss-optimal IM operation, the rotor flux amplitude is adjusted according to the torque reference amplitude. By doing so, the ohmic losses in both the stator and rotor as well as the iron losses are reduced compared to an operation with load-independent rotor flux amplitude at its nominal full-load value. Consequently, the transient torque response is physically limited by the IM’s large rotor time constant when torque reference steps occur from partial to full load. Therefore, a steady-state loss-optimal IM operation scheme motivates the need for a high-performance torque controller such that acceptable torque dynamics can be achieved. This can be realized by fully exploiting the drive’s capabilities in terms of the current and voltage constraints, which leads, among others, to the challenge of choosing a suitable trade-off between the flux- and torque-generating stator current components during transients.

While linear feedback control, such as any variant from the classical PID-style field-oriented approaches, is known for its limited transient control performance, model predictive control (MPC) renders itself the most promising alternative. However, existing predictive torque control approaches very often do not consider the problem of reduced rotor flux at partial load with respect to transient response, i.e., they always operate at the nominal flux level, leading to suboptimal drive efficiency in steady state [5]–[12]. Others use strongly simplified heuristic rules for the \( i_{sq} = i_{eq} \) trade-off during torque transients, e.g., always prioritizing the flux-forming current component [13]–[15], leading to strongly suboptimal transient torque performance. Only few prior publications investigated the transient torque control problem when operating at partial rotor flux: While [16] proposes a model predictive torque control (MPTC) that includes variable rotor flux excitation, it only considers the simplified linear IM model without any non-linear impacts such as magnetic saturation, iron losses, skin effect, or other parameter-varying effects such as thermal influences. In addition, [12] studies a similar problem based on a hierarchical control approach to obtain extended prediction horizons, but also only considers a linear IM model and does not meet the optimal steady-state operating points exactly.

Moreover, the above-discussed control methods from the literature are tailored to single-machine drive system (SMDS) applications, i.e., it remains unclear how they scale to higher-
dimensional MPC problems when considering MMDS with accordingly increased state and action spaces. Very few approaches exist for torque control of MMDS with substantial limitations. E.g., [17] only selects steady-state torque references using highly simplified drive models also neglecting the IM dynamics and [18] considers the dynamic torque references for complex mechanical loads without taking the efficiency issue in steady-state operation into account.

B. Contribution

To overcome the limitations of existing state-of-the-art solutions, this paper deals with fast and efficient torque control of IMs for SMDS as well as MMDS with the help of MPC. The accuracy of this MPTC is guaranteed by the utilization of a highly accurate but nonlinear model of an IM and rotor flux observer published in [19]. The model [19] takes important effects into account like magnetic saturation, iron losses, skin effect and parameter variations due to temperature changes effects into account like magnetic saturation, iron losses, skin effect and parameter variations due to temperature changes.

The proposed control scheme is of a cascaded form and is shown in Fig. 1. As the IM’s stator current dynamics are faster compared to the rotor flux dynamics by at least one order of magnitude, a separate treatment of the current control and the torque control problems in a hierarchical approach is sensible. For the remainder of this contribution, it is therefore assumed that the stator currents reach their references in an infinitesimally short time such that the stator current control will not be considered in the following. For the latter, any suitable solution from the literature can be used, such as [20]–[23]. Consequently, we will focus on the torque control problem by choosing appropriate stator current references to enable both a steady-state loss-minimal operation and a highly dynamic torque response during transients.

In summary, the main innovations of the proposed control solution are as follows:

- Realization of accurate and highly dynamic torque control considering parasitic, nonlinear drive effects,
- Numerically efficient solution of the resulting nonlinear MPC problem using iterative linearization along the predicted trajectories,
- Combination of the steady-state loss-optimal operating point as well as transient current profile calculations in a single real-time capable MPC instance, and
- Scalability from SMDS towards MMDS applications.

The feasibility of the proposed solution is demonstrated on extensive simulative and experimental tests.

II. NONLINEAR MODEL AND NOMENCLATURE

The basis for the following investigation is an accurate but nonlinear model of the IM as introduced in [19], whose ECD is depicted in Fig. 2 in rotor flux-oriented coordinates. Since the considered machine is a squirrel-cage IM, the rotor circuit on the right is short-circuited and the rotor current is only generated by induction. For the sake of brevity, only selected elements of the model from [19], which are particularly important for the considered torque control problem, are discussed in the following. Most importantly, the nonlinear IM model covers the following three effects:

1) The mutual inductance depends on the magnetizing current due to magnetic saturation:

\[ L_m(i_m) = k_1 + \frac{k_1 - k_2}{1 + e^{k_3 ||i_m||^2 - k_4}}. \]  

(1)

This also affects the rotor and stator inductance:

\[ L_{r,s} = L_m(i_m) + L_{	ext{arm}}. \]

2) Iron losses are modeled by a constant resistance leading to a reduced stator current that contributes to the torque and the rotor flux linkage.

3) The rotor and stator resistance depend on the rotor angular frequency \( \omega_r \) and stator electrical angular frequency \( \omega_s \) as well as the rotor \( \vartheta_r \) and stator temperature \( \vartheta_s \):

\[ R_{r,s}(\omega_r, \vartheta_s) = R_{r,s}(1 + h_{r,s}(\omega_r^2) \cdot f_{r,s}(\vartheta_s)). \]  

(2)

Here, the temperature dependency is given by

\[ f_{r,s}(\vartheta_s) = 1 + \alpha_{r,s} \cdot (\vartheta_s - 20 \, \text{°C}) \]  

(3)

and the rotor and stator angular frequency are related by the mechanical rotational speed \( n \) and the pole pair number \( N_p \):

\[ \omega_r = N_p \omega_{\text{mech}} + \omega_r = N_p 2\pi n + \omega_r. \]  

(4)

The parameters \( k_1, k_2, k_3, k_4, \omega_0, \omega_{\text{mech}}, R_{r,s}, h_{r,s} \) are determined based on material data and an offline identification – further information can be obtained in [19].
An arbitrary vector \( \mathbf{x} \) in Fig. 2 consists of a d- and q-coordinate and the matrix \( \mathbf{J} \) is the rotation matrix:

\[
\mathbf{x} = \begin{bmatrix} x_d \\ x_q \end{bmatrix}, \quad \mathbf{J} = \begin{bmatrix} 0 & -1 \\ 1 & 0 \end{bmatrix}.
\]

Moreover, the flux linkage equations can be stated as

\[
\psi_s = L_s i_s + L_m i_r, \quad \psi_r = L_m i_1 + L_r i_r.
\]

### III. PRELIMINARIES OF THE MPC PROBLEM

Starting from the control scheme in Fig. 1, the MPTC uses the reference current components \( i_{s_d}^* \) and \( i_{s_q}^* \) as inputs of the plant system and the rotor flux linkage \( \psi_{rd} \) is the state variable. As already discussed, we will focus on the dominating rotor flux dynamics to describe the torque control state variable. As already discussed, we will focus on the 

\[
\psi \text{ uses the reference current components}
\]

\[
\text{this variable in the cost function, the torque is forced towards}
\]

\[
\text{the control horizon and is an optimization variable. It allows}
\]

\[
\text{Therefore, a slack variable \( \varepsilon \) at time step \( k = k_c - 1 \) are kept constant}
\]

\[
\text{while the parameters and the state are further predicted.}
\]

\[
\text{Considering the mentioned goals, the cost function consists of}
\]

\[
\text{the losses} P_l \text{ and the torque deviation (represented by \( \varepsilon \)):}
\]

\[
J = \sum_{k=0}^{k_p} P_l[k] + q_1 \varepsilon[k] + q_2 \varepsilon^2[k].
\]

Here, \( q_1 \) and \( q_2 \) are the weighting factors of the slack variable and the squared slack variable, respectively. \( \varepsilon \) is needed to penalize small deviations from the reference since \( \varepsilon^2 \) is much smaller in this case. The latter is necessary to obtain a positive semi-definite Hessian \( H \) later in (15). The optimization vector is arranged as follows:

\[
U_{k_c} = [(i_{s_d}^*[0])^\top \cdots (i_{s_d}^*[k_c-1])^\top \varepsilon[0] \cdots \varepsilon[k_c-1]]^\top.
\]

To allow utilizing efficient numerical optimization solvers, the nonlinear MPC problem must be transferred into a QP of the form:

\[
U_{k_c}^{\text{opt}} = \arg \min_{U_{k_c}} \frac{1}{2} U_{k_c}^\top H U_{k_c} + f^\top U_{k_c}, \quad G U_{k_c} \leq e. \quad (15)
\]

Here, \( H \) and \( f \) are the condensed matrices and vectors that are derived from the cost function (13) – see Sec. VI. \( G \) and \( e \) describe the constraints which are derived in detail in Sec. V. In the following, we will show how the given MPTC problem can be transferred into the QP problem (15).

### IV. DERIVATION OF THE ROTOR CIRCUIT DYNAMICS

For the derivation of the discrete-time state equation, the mechanical rotational speed \( n \), the DC-link voltage \( u_{dc} \) and the temperatures \( \vartheta_s, \vartheta_t \) are considered exogenous quantities that are constant within the prediction horizon. The dependence of the parameters \( L_m, R_s, R_t, \omega_r \) on the states and exogenous variables is not explicitly labeled below for reasons of readability and brevity.

The first step of the derivation of the QP problem (15) is the derivation of the discrete-time state equation. Since the rotor flux linkage \( \psi_{rd} \) is the state variable, one has to find its dynamics depending on the input \( i_s \). From (7), the rotor current

\[
i_r(t) = \frac{1}{L_r} (\psi_r(t) - L_m i_1(t))
\]

results. Inserting this into the rotor equation (according to Fig. 2)

\[
0 = \frac{d\psi_r(t)}{dt} + \omega_r \mathbf{J} \psi_r(t) + R_t i_r(t)
\]

leads to the differential equation

\[
\frac{d\psi_r(t)}{dt} = -\left( \frac{1}{\tau_r} \mathbf{I} + \omega_r \mathbf{J} \right) \psi_r(t) + \frac{L_m}{\tau_r} i_1(t)
\]

where \( \mathbf{I} \) is the identity matrix and \( \tau_r = L_r/R_t \) the time constant of the rotor flux. To eliminate the dependency from \( i_1 \), a relation \( i_1(\psi_r, i_r) \) must be found.

Inserting (16) into (6) and applying the time derivative yields

\[
\frac{d\psi_s(t)}{dt} = \frac{L_m}{L_r} \frac{d\psi_r(t)}{dt}.
\]
Here, the assumption $\frac{dx_i}{dt} = 0$ from Sec. III is utilized. The stator equation according to Fig. 2

$$0 = \frac{d\psi_s(t)}{dt} + \omega_a J \psi_s(t) - R_{fe} \cdot (i_s(t) - i_1(t)) \quad (20)$$

can be transferred into a relation between $i_1$ and $i_s$. Substituting the appropriate terms with (6), (16), (18) and (19) results in

$$i_1(t) = G_i i_s(t) + G_{\psi} \psi_1(t), \quad (21)$$

$$G_i = \left[ (1 + \frac{R_L L_m}{R_{fe} L_r}) I + \frac{\omega_a}{R_{fe}} \left( L_n - \frac{L_m^2}{L_r} \right) J \right]^{-1}, \quad (22)$$

$$G_{\psi} = G_i \frac{L_m}{R_{fe} L_r} \left( \frac{1}{\tau_r} \right), \quad (23)$$

If $i_1$ in (18) is replaced by (21), it yields the rotor circuit dynamics. Due to the rotor flux-oriented coordinate system, the model parameters do not drastically change between two sampling steps. The parameters of this differential equation are defined by

$$\text{the rotor flux linkages is given by} \quad \psi_{rd}(t) = a \psi_{rd}(t) + b^T i_s(t) \quad (24)$$

The parameters of this differential equation are defined by

$$a = \frac{1}{\tau_r} (1 - L_m g_{\psi,11}), \quad (25)$$

$$b^T = \frac{L_m}{\tau_r} \begin{bmatrix} g_{\psi,11} \\ g_{\psi,12} \end{bmatrix} \quad (26)$$

where $g_{\psi,ij}$ denotes the element in the $i$-th row and $j$-th column of the matrix $G^{\psi}$. It should be mentioned that $a$ and $b$ are not constant since the parameter trajectories are already known. These are assumed to be constant in each sampling interval within the prediction horizon so the exact discretization can be applied to (24). This leads to

$$\psi_{rd}[k+1] = a_d[k] \psi_{rd}[k] + b_d[k] i_s[k], \quad \psi_{rd}[0] = \psi_{rd,0} \quad (27)$$

with the initial rotor flux linkage $\psi_{rd,0}$, the discrete-time quantities $[27]$

$$a_d[k] = e^{a[k] T_s}, \quad b_d[k] = \left( e^{a[k] T_s} - 1 \right) b^T[k] \quad (28)$$

and the sampling time $T_s$ of the MPC. Here, the exact discretization is applied on a per-sampling-step basis assuming that the model parameters do not drastically change between two sampling steps.

The discrete time steps $k = 0, \ldots, k_c - 1$ of this state equation are condensed in matrix-vector-notation and extended by $k_{\text{ext}}$ time steps. The condensed state vector represented by the rotor flux linkages is given by

$$\Psi_{k_c} = [\psi_{rd}[0] \cdots \psi_{rd}[k_{\text{pe}}]]^T \quad (29)$$

and can be calculated depending on the input signal trajectory (14) with the condensed rotor circuit dynamics

$$\Psi_{k_c} = A_{k_c} \psi_{rd,0} + B_{k_c} U_{k_c}. \quad (30)$$

The matrix $A_{k_c}$ is defined by

$$A_{k_c} = \begin{bmatrix} 1 & a_d[0] & a_d[1] & \cdots & a_d[k_{\text{pe}} - 1] \end{bmatrix}^T \quad (31)$$

and the structure of $B_{k_c}$ is given in appendix A.

V. CONSTRAINTS

A. Current Constraint

For thermal reasons, the stator current must not exceed a certain value. Thus, the magnitude of the stator current vector is limited:

$$i_{sd}^2[k] + i_{sq}^2[k] \leq i_{s,max}^2. \quad (32)$$

Since this is a quadratic constraint with respect to the stator current, it has to be linearized to comply with the form given in (15). This is done by approximating the circle by tangents of the form

$$\cos(\varphi) i_{sd}[k] + \sin(\varphi) i_{sq}[k] \leq i_{s,max} \quad (33)$$

with an arbitrary polar angle $\varphi$ where the tangent is set up. Fig. 3 shows an example of an approximation with four fixed tangents.

The approximation of the circle is a trade-off between high accuracy (many tangents) and low complexity (few tangents) of the resulting problem. Since both the $d$- and $q$-component of $i_s$ can be positive and negative in transients, the entire circle must be considered. Furthermore, the approximation error should not concern one component more than the other so a symmetrical approximation with respect to the $d$- and $q$-axis is necessary. Thus, for the approximation of (32), four tangents with the angles $\varphi = \{0, \frac{\pi}{4}, \pi, \frac{3\pi}{4}\}$ are chosen. Additionally, a dynamic tangent is introduced that depends on the stator current trajectory of the previous prediction $\bar{i}_{s}[k]$ like in [28]. Fig. 3 gives an example for this. The dynamic tangent results in the following inequality [28]:

$$\frac{i_{sd}}{i_s} \leq i_{s,max}. \quad (34)$$

All five inequalities can be represented by

$$\begin{bmatrix} \bar{i}_{s}[k] \\ \bar{i}_{s}[k] \end{bmatrix} \quad \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \quad (35)$$

$$\begin{bmatrix} \omega_{u,1} \end{bmatrix} \quad \begin{bmatrix} 0 & \cdots & 0 & 0 \end{bmatrix} \quad (37)$$

$$\begin{bmatrix} \omega_{u,1} \end{bmatrix} \quad \begin{bmatrix} 0 & \cdots & 0 \end{bmatrix} \quad (38)$$

with

$$\Psi_{k_c} = \Omega_{k_c} \quad (36)$$
where \( u_s \) can be linearly expressed by \( i_s \) and \( \psi_{rd} \) as shown in the following.

For the derivation of \( u_s \), the same assumptions from Sec. IV hold. The stator voltage can be found by setting up the stator mesh equation (compare Fig. 2)

\[
\begin{align*}
R_s i_s(t) + \frac{d\psi_s(t)}{dt} + \omega_r \psi_s(t) &= u_s(t) \\
\psi_s(t) &= \begin{bmatrix} \psi_{d} \\ \psi_{q} \end{bmatrix}
\end{align*}
\]

By utilizing (6), (16), (18), (19) and (21), the stator voltage can be rewritten in such a way that it linearly depends on the stator current and the rotor flux linkage. Therefore, the matrix

\[
R_l = \omega_s \left( L_s - \frac{L_m^2}{L_r} \right) J + \frac{R_r L_m^2}{L_r^2} I.
\]

is introduced so that the stator voltage results in

\[
\begin{align*}
&u_s(t) = V_l i_s(t) + V_{\psi} \psi_r(t) \\
&\text{with the matrices}
\end{align*}
\]

\[
\begin{align*}
V_l &= R_s I + R_s G_l \\
V_{\psi} &= R_t G_{\psi} - \frac{R_r L_m}{L_r^2} I + N_l \omega_{\text{mech}} \frac{L_m}{L_r} J.
\end{align*}
\]

Taking varying parameters into account, equation (45) can be discretized as:

\[
\begin{align*}
u_s[k] &= V_l[i_s[k]] + V_{\psi}[\psi_r[k]].
\end{align*}
\]

Together with (41), (42) and (48), all linearized inequalities can be compactly written as

\[
N_1[k] i_s[k] + N_2[k] \psi_r[k] \leq \omega_v[k].
\]

The definitions of the quantities \( N_1[k] \), \( N_2[k] \) and \( \omega_v[k] \) are given in appendix A. For the extension of the prediction horizon, the stator current \( i_s[k+1] \) is held constant beyond the control horizon. This leads to the condensed inequality

\[
N_1 U_{k_c} + N_2 \Psi_{k_c} \leq \Omega_v
\]

with

\[
\begin{align*}
N_1 &= \begin{bmatrix} N_1[0] & 0 & \ldots & 0 & 0 \\
0 & \ddots & \vdots & \vdots & \vdots \\
\vdots & \ddots & 0 & 0 & N_1[k_c - 1] \\
0 & \ldots & 0 & N_1[k_c] & 0 \\
0 & \ldots & 0 & 0 & N_1[k_p] \end{bmatrix}, \\
N_2 &= \begin{bmatrix} n_2[0] & 0 & \ldots & 0 \\
0 & \ddots & \vdots & \vdots \\
\vdots & \ddots & 0 & n_2[k_p - 1] \\
0 & \ldots & 0 & n_2[k_p] \end{bmatrix},
\end{align*}
\]

\[
\begin{align*}
\Omega_v &= \begin{bmatrix} \omega_v[0] & \ldots & \omega_v[k_p] \end{bmatrix}^T.
\end{align*}
\]

In the end, the constraint is found by inserting (30):

\[
\frac{(N_1 + N_2 B_{k_c}) U_{k_c}}{\mathcal{N}_{u,2} \psi_{rd,0}} \leq \Omega_v - N_2 A_{k_c} \psi_{rd,0}. 
\]

Fig. 3: Exemplary approximation of a circle – with static tangents and a dynamic tangent where \( x \in \{i_s, u_s\} \) represents either the stator current or the stator voltage.
C. Specification of a Minimal Rotor Flux

The IM always requires a certain minimum value of the rotor flux linkage \( \psi_{rd,\text{min}} \) to ensure controllability:

\[
\Psi_{kc} \geq \psi_{rd,\text{min}} \mathbf{1}.
\]  

(55)

Here, \( \mathbf{1} \) represents a column vector of ones. Inserting (30) leads to the condensed constraint

\[
-\mathbf{B}_{kc} U_{kc} \leq \mathbf{A}_{kc} \psi_{rd,0} - \psi_{rd,\text{min}} \mathbf{1} = \Omega_{u,3}.
\]  

(56)

D. Soft-constraint for the Torque Control

The soft-constraint (11) has to be linear to conform with (15). Since the torque (9) in combination with (21) is nonlinear regarding the multiplication of \( \dot{i}_s \) and \( \psi_{rd} \), it is linearized in the OP \( \psi_{rd} \) by the Taylor series

\[
\hat{T} \approx \frac{\partial T}{\partial \psi_{rd}} (\dot{i}_s - \ddot{i}_s) + \frac{\partial T}{\partial \dot{i}_s} (\dot{i}_s - \ddot{i}_s) + \mathbf{T}
\]  

(57)

where \( \mathbf{T} = T(\dot{i}_s, \ddot{i}_s) \) denotes the torque evaluated in the OP. The dependence on the time step \( k \) is neglected here for readability. With the help of the quantities

\[
M_1[k] = \left[ \frac{\partial T}{\partial \psi_{rd}} \right]_{\dot{i}_s = \ddot{i}_s} \quad \text{and} \quad m_2[k] = \left[ \frac{\partial T}{\partial \dot{i}_s} \right]_{\dot{i}_s = \ddot{i}_s}
\]  

(58)

\[
\omega_k = \left[ \begin{bmatrix} T^* - \mathbf{T} \end{bmatrix} \left[ i_s - \ddot{i}_s \right] + \frac{\partial T}{\partial \psi_{rd}} (\psi_{rd} - \ddot{\psi}_{rd}) \right]
\]  

(59)

that all depend on the time step \( k \) the soft-constraint can be compactly written as

\[
M_1[k] \dot{i}_s[k] - \mathbf{1} \mathbf{e}[k] + m_2[k] \dot{\psi}_{rd}[k] \leq \omega_k[k]
\]  

(60)

The condensed matrices \( \mathbf{M}_1 \) and \( \mathbf{M}_2 \) as well as the vector \( \Omega_t \) are built analogously to (51)-(53) but \( \mathbf{M}_1 \) contains the vectors \( -\mathbf{1} \) from (59) instead of the pure zero matrices in the last column. This results in the constraint

\[
(\mathbf{M}_1 + \mathbf{M}_2 \mathbf{B}_{kc}) U_{kc} \leq \Omega_t - \mathbf{M}_2 \mathbf{A}_{kc} \psi_{rd,0} = \Omega_{u,4}.
\]  

(61)

VI. COST FUNCTION AND QP PROBLEM

For the cost function, the machine losses have to be found depending on \( \dot{i}_s \) and \( \psi_{rd} \). Equation (10) describes the calculation of the total machine losses. Utilizing (16), (21), (43), (45) and (46), the machine losses can be stated as

\[
P_l = i_s^3 \mathbf{K}_{ii} i_s + \psi_{rd}^2 \mathbf{K}_{\psi i} i_s + \psi_{rd} \mathbf{K}_{\psi \psi} \psi_r \]

(62)

with the newly introduced matrices

\[
\mathbf{K}_{ii} = \frac{3}{2} \left[ R_i + \frac{R_L I_m^2}{L_e^2} \mathbf{G}_i^T \mathbf{G}_i + \frac{G_i^T R_i^T R_i \mathbf{G}_i}{R_{te}} \right],
\]

(63)

\[
\mathbf{K}_{\psi i} = \frac{3}{2} \left[ \frac{\mathbf{V}_\psi R_i \mathbf{G}_i}{R_{te}} - \frac{R_L I_m}{L_e^2} (\mathbf{I} - L_m \mathbf{G}_\psi)^T R_i \mathbf{G}_i \right],
\]

(64)

\[
\mathbf{K}_{\psi \psi} = \frac{3}{2} \left[ \frac{R_i}{L_e^2} (\mathbf{I} - L_m \mathbf{G}_\psi)^T (\mathbf{I} - L_m \mathbf{G}_\psi) + \frac{\mathbf{V}_\psi^T \mathbf{V}_\psi}{R_{te}} \right].
\]

(65)

It should be noted that these matrices are dependent on the qLPV representation of the IM model and depend on the specific linearization approach based on future state and action trajectories as it will be discussed in Sec. VII. By utilizing (30), the cost function (13) can be compactly written as the QP problem (15) with the quantities

\[
H = 2 \left[ \mathbf{R}_{kc} + \mathbf{B}_{kc}^T \right] \left[ \mathbf{Q}_{kc} \mathbf{B}_{kc} + \mathbf{T}_{kc} \right],
\]

(66)

\[
f^T = \psi_{rd,0} \mathbf{A}_{kc}^T \left[ 2 \mathbf{Q}_{kc} \mathbf{B}_{kc} + \mathbf{T}_{kc} \right] + (\mathbf{V}_{kc})^T,
\]

(67)

\[
G = \begin{bmatrix} \Omega_{u,1} \n_{u,2} \n_{u,3} \n_{u,4} \end{bmatrix}, \quad \epsilon = \begin{bmatrix} \Omega_{u,1} \n_{u,2} \n_{u,3} \n_{u,4} \end{bmatrix}
\]  

(68)

Here, the matrices

\[
\mathbf{Q}_{kc} = \begin{bmatrix} k_{\psi \psi,11} [0] & \cdots & \cdots & \cdots \end{bmatrix},
\]

(69)

\[
(\mathbf{V}_{kc})^T = \begin{bmatrix} 0^T & q_1 \mathbf{1}^T (k_{\text{ext}} + 2)q_1 \end{bmatrix}^T
\]

and \( \mathbf{R}_{kc}, \mathbf{T}_{kc} \) are introduced according to appendix A. \( \mathbf{U}_{h_e,3} \)-independent terms are omitted since they do not affect the optimal solution. The QP problem (15) with the definitions (65)-(67) can be solved numerically with the solver \textit{mpcInteriorPointSolver} from Matlab. This solver, however, is not computationally efficient. To utilize the more efficient Matlab solver \textit{mpcActiveSetSolver}, the matrix \( H \) must be symmetrical which is not the case in (65) – see proof in appendix B. To transform \( H \) into a symmetrical matrix, the coupling term in (61) needs to be linearized with respect to \( \dot{i}_s \) and \( \psi_{rd} \) which yields:

\[
P_l \approx i_s^3 \mathbf{K}_{ii} \dot{i}_s + \psi_{rd}^2 \mathbf{K}_{\psi i} \dot{i}_s + \psi_{rd} \mathbf{K}_{\psi \psi} \dot{\psi}_r - h_i
\]

(70)

\[
\dot{h}_i = \dot{\psi}_r \mathbf{K}_{\psi i}, \quad \dot{h}_i = \dot{\psi}_r \mathbf{K}_{\psi i},
\]

(71)

This leads to the relaxed QP problem defined by (67) and

\[
H = 2 \left[ \mathbf{R}_{kc} + \mathbf{B}_{kc}^T \mathbf{Q}_{kc} \mathbf{B}_{kc} \right],
\]

(72)

\[
f^T = 2 \psi_{rd,0} \mathbf{A}_{kc}^T \mathbf{Q}_{kc} \mathbf{B}_{kc} + \mathbf{V}_{kc}^T + \mathbf{T}_{kc}^T \mathbf{B}_{kc}
\]

(73)

Fig. 4: Simulative comparison of the input currents with the \textit{mpcActiveSetSolver} (ASS) and the \textit{mpcInteriorPointSolver} (IPS) from Matlab for a reference step 0 Nm \( \rightarrow \) 9.5 Nm at \( n = 500 \text{ min}^{-1} \).
with $V_{k_c}$ being defined in the appendix A and
\[
T_{k_c} = [h_{\psi,1}[0] \cdots h_{\psi,1}[k_p]]^T.
\]
$Q_{k_c}$ and $R_{k_c}$ stay the same. Both solvers generate similar results as Fig. 4 illustrates with a simulation example. The `mpcInteriorPointSolver`, however, requires approximately 40 times more time for computation of the QP problem (about 161 ms on the hardware according to Sec. VIII) in comparison to the `mpcActiveSetSolver` (about 4 ms) strongly motivating to use the latter. The stated computation times apply to the steady state in Fig. 4.

VII. OPTIMIZATION PROCESS WITH PARAMETER TRACKING

In the following, the MPTC is carried out in an iterative manner like in [25], [26]. Therefore, two terms are introduced: The solution of the QP problem defined by (67), (72), (73) including parameter prediction and linearization is called subiteration. The optimization process denotes the calculation of the reference current of the succeeding sampling time $i_{sd}^{opt}[k+1]$ depending on the current reference torque $T^* [k]$ and the last estimated rotor flux $\hat{\psi}_{rd}[k]$. This optimization process consists of several subiterations.

The solution of the qLPV MPC problem requires the knowledge of the future parameters $L_m$, $R_s$, $R_r$, $\omega_r$ within the prediction horizon as described in [25], [26]. Fig. 5 illustrates the different steps of the first subiteration of an optimization process including parameter prediction. For reasons of space, only the control horizon is depicted. At the beginning of a subiteration, the parameter trajectories, represented by the vector
\[
p[k] = [L_m[k] \ R_s[k] \ R_r[k] \ \omega_r[k]] \quad 0 \leq k \leq k_c - 1
\]
in Fig. 5, as well as the stator current trajectory $i_s^{opt}$ of the previous subiteration and the estimation of the rotor flux linkage $\hat{\psi}_{rd}$ are given. A subiteration includes three steps:

1) Based on $\hat{\psi}_{rd}$, $p$ and $i_s^{opt}$ from the previous subiteration, the rotor flux linkage $\hat{\psi}_{rd}$ can be predicted for the new control horizon using (27) – compare red arrows in Fig. 5. The current $i_s^{opt}[k_c - 2]$ and, for the sake of computational efficiency, the parameters $p[k_c - 2]$ are reused for the prediction of $\hat{\psi}_{rd}$ for the rest of the prediction horizon.

2) The predicted rotor flux $\hat{\psi}_{rd}$ together with $p$ and $i_s^{opt}$ from the previous subiteration is then used to update the parameters in every time step in the control horizon (compare blue arrows in Fig. 5). Since the new control horizon is shifted compared to the previous one, the old current and parameters are extended constantly for one step. The update of the parameters is carried out with the equations that result from the ECD in Fig. 2.

3) The updated parameters can now be used to set up the QP problem defined by (67), (72), (73) as shown in Fig. 5 with the orange arrows. The OP $\hat{\psi}_{rd}$, $\hat{\psi}_{rd}$ of the linearizations is defined by the predicted rotor flux $i_s^{opt}$.
linkage from step 1) and the current trajectory from the previous subiteration. The dynamic voltage constraint (42) makes use of the voltage trajectory \( \tilde{u}_s \) from the previous subiteration. The solution of (15) delivers the optimal future input trajectory \( \tilde{i}_s^\text{opt} \).

In summary, the nonlinear IM model is successively linearized along the future state and action trajectory predicted by the MPC. To allow convergence of both the linearization and prediction schemes, several subiterations are conducted in every controller update cycle before the optimal reference current \( i_s^\text{opt} \) (Fig. 5 in green) is passed to the current controller. This optimization process is depicted in Fig. 6.

Fig. 7 shows an example for different number of subiterations \( N_{\text{sub}} \) and the resulting trajectories of \( i_s^\text{opt} \). The more subiterations are done, the closer the trajectory is to the optimal one. The predicted trajectories sometimes oscillate around the optimal one with every new subiteration. To avoid this, a damping factor \( \lambda \) is introduced so the adapted current trajectory

\[
\left( i_s^\text{opt}' \right) = i_s^\text{opt,old} + \lambda \cdot \left( i_s^\text{opt} - i_s^\text{opt,old} \right) \quad (76)
\]

is used in the next subiteration instead of \( i_s^\text{opt} \). Here, \( i_s^\text{opt,old} \) is the optimal current trajectory of the previous subiteration. For the whole optimization process (including subiterations), the DC-link voltage \( u_{\text{dc}} \), the mechanical speed \( n \) and the temperatures \( \vartheta_r, \vartheta_s \) are assumed to be constant.

VIII. EXPERIMENTAL VALIDATION

At the test bench, the MPTC is implemented on a rapid control prototyping system (RCPS) that processes the field-oriented control (FOC) for the stator current control as well. The solver mpcActiveSetSolver from Matlab is used to solve the stated QP problem (15). A load machine drive controls the speed of the test machine. Table I and Table II give information on the control hardware.

The proposed MPTC is completed with a PI controller for current control and pulse width modulation. The MPTC is compared with an optimal reference current computation (ORCC) from [2] with a PI controller as the rotor flux controller. The manually tuned MPC parameters and PI controller parameters as well as the control cycle frequencies of the current controller, the ORCC and MPC on the RCPS are shown in Table III. The rotor flux linkage is estimated with the flux observer [19] and filtered with a low-pass filter of first order according to Table III to dampen noise. Both the ORCC and the MPTC utilize only 95% of the DC-link voltage to leave some space for transient overvoltages and voltage drops inside the inverter.

The experimental validation consists of three parts: At first, the dynamic behavior of the MPTC is compared with the

**TABLE I: Drive components, measurement equipment and control hardware**

<table>
<thead>
<tr>
<th>Component</th>
<th>Manufacturer</th>
<th>Product name</th>
</tr>
</thead>
<tbody>
<tr>
<td>Test machine</td>
<td>EMB</td>
<td>Individual fabrication</td>
</tr>
<tr>
<td>Inverter (test machine)</td>
<td>SEMIKRON</td>
<td>Semitech IGBT</td>
</tr>
<tr>
<td>Power analyzer</td>
<td>ZES ZIMMER</td>
<td>LMG 671</td>
</tr>
<tr>
<td>Torque sensor</td>
<td>interfaceforce</td>
<td>FFT2-20NM-B</td>
</tr>
<tr>
<td>RCPS (load machine)</td>
<td>dSPACE</td>
<td>MicroLabBox</td>
</tr>
<tr>
<td>Inverter (load machine)</td>
<td>LTI</td>
<td>ServoOne junior</td>
</tr>
</tbody>
</table>

**TABLE II: Test drive characteristics**

<table>
<thead>
<tr>
<th>Symbol</th>
<th>Description</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>( T_N )</td>
<td>Nominal torque</td>
<td>10.21 Nm</td>
</tr>
<tr>
<td>( I_N )</td>
<td>Nominal phase current (star connection)</td>
<td>3.27 A</td>
</tr>
<tr>
<td>( P_{\text{me},N} )</td>
<td>Nominal mechanical power</td>
<td>1.5 kW</td>
</tr>
<tr>
<td>( n_N )</td>
<td>Nominal speed</td>
<td>1404 min(^{-1})</td>
</tr>
<tr>
<td>( N_p )</td>
<td>Pole pair number</td>
<td>2</td>
</tr>
<tr>
<td>( u_{\text{dc}} )</td>
<td>DC-link voltage</td>
<td>560 V</td>
</tr>
</tbody>
</table>

**TABLE III: MPC parameters, PI controller parameters, control cycle frequencies and low-pass filter cutoff frequencies**

<table>
<thead>
<tr>
<th>Symbol</th>
<th>Description</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>( k_c )</td>
<td>Number of prediction steps</td>
<td>4</td>
</tr>
<tr>
<td>( T_s )</td>
<td>Sampling time of the MPC</td>
<td>17.8 ms</td>
</tr>
<tr>
<td>( q_1 )</td>
<td>Weighting of slack variable</td>
<td>120 W/(Nm)</td>
</tr>
<tr>
<td>( q_2 )</td>
<td>Weighting of squared slack variable</td>
<td>20 W/(Nm)(^2)</td>
</tr>
<tr>
<td>( k_{\text{ext}} )</td>
<td>Number of extended time steps</td>
<td>7</td>
</tr>
<tr>
<td>( N_{\text{sub}} )</td>
<td>Number of subiterations</td>
<td>3</td>
</tr>
<tr>
<td>( \lambda )</td>
<td>Damping factor</td>
<td>0.6</td>
</tr>
</tbody>
</table>

**PI controller parameters**

<table>
<thead>
<tr>
<th>Symbol</th>
<th>Description</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>( K_{p,i} )</td>
<td>Current controller gain</td>
<td>70 V/A</td>
</tr>
<tr>
<td>( T_{r,i} )</td>
<td>Current controller reset time</td>
<td>3.75 ms</td>
</tr>
<tr>
<td>( K_{p,\psi} )</td>
<td>Rotor flux controller gain</td>
<td>5 A/(Vs)</td>
</tr>
<tr>
<td>( T_{r,\psi} )</td>
<td>Rotor flux controller reset time</td>
<td>43 ms</td>
</tr>
</tbody>
</table>

**Control cycle frequencies**

<table>
<thead>
<tr>
<th>Symbol</th>
<th>Description</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>( f_{\text{CC}} )</td>
<td>Current controller and flux observer</td>
<td>10 kHz</td>
</tr>
<tr>
<td>( f_{\text{ORCC}} )</td>
<td>ORCC</td>
<td>2 kHz</td>
</tr>
<tr>
<td>( f_{\text{MPC}} )</td>
<td>MPC</td>
<td>1/(17.8 ms)</td>
</tr>
</tbody>
</table>

**Low-pass filter cutoff frequencies**

<table>
<thead>
<tr>
<th>Symbol</th>
<th>Description</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>( f_{\psi} )</td>
<td>Rotor flux filter</td>
<td>200 Hz</td>
</tr>
<tr>
<td>( f_{T} )</td>
<td>Torque filter</td>
<td>10 Hz</td>
</tr>
</tbody>
</table>
TABLE IV: Settling times of the step responses of ORCC and MPTC according to Fig. 9 and corresponding error bands

<table>
<thead>
<tr>
<th></th>
<th>ORCC</th>
<th>MPTC</th>
<th>Error band</th>
</tr>
</thead>
<tbody>
<tr>
<td>Fig. 9a</td>
<td>217.2 ms</td>
<td>140.1 ms</td>
<td>9.5 Nm ± 0.4 Nm</td>
</tr>
<tr>
<td>Fig. 9b</td>
<td>390.0 ms</td>
<td>52.0 ms</td>
<td>7 Nm ± 0.7 Nm</td>
</tr>
<tr>
<td>Fig. 9c</td>
<td>59.4 ms</td>
<td>35.8 ms</td>
<td>−1 Nm ± 0.1 Nm</td>
</tr>
</tbody>
</table>

ORCC. Afterwards, the MPTC is tested at the power limit with a speed ramp. In the end, the steady-state machine losses of both methods are compared. Since the measured torque includes mechanical oscillations and a delay due to the limited cutoff frequency of the torque sensor, the estimated torque $\hat{T}$ is shown in the graphs which is calculated with the model [19].

Fig. 9 shows the measurements of three different reference torque steps and Table IV the settling times with the corresponding error bands. Fig. 9a contains the case for a large reference torque step. Thereby, the rotor flux has to be built up from a small to a large value and the machine operates at the current limit. The plot shows that the MPTC controls the torque faster than the ORCC despite the delay of one sampling time $T_s = 17.8$ ms after the reference step. The step-like torque trajectory with the MPTC is due to sudden changes of $i_{eq}$ while the slower PT-1 behavior is caused by the rotor flux. In Fig. 9b, a reference step in the flux weakening region is applied. In contrast to the MPTC, the ORCC makes the induced voltage increase too large which can not be compensated for with the given stator voltage limit.

This causes problems with the current control and, hence, the torque decreases before it reaches its reference value. The measurements in Fig. 9c for the case of a small reference torque steps show a negative overshoot with the ORCC. This negative overshoot arises due to the large time constant $\tau_r$ of the rotor flux – compare (18). The MPTC, however, does not show any overshoot and controls the torque approximately in one sampling time.

Furthermore, Fig. 10 illustrates how every subiteration improves the step response, especially when the machine parameters change quickly in time. The settling time and the integrated squared error in Table V support this.

This results of the test at the power limit are shown in Fig. 11. In this test, a speed ramp with constant slope was given whose maximum is at $n = 2500$ min$^{-1}$. During this speed ramp, the MPTC controls a reference torque of $T^* = 9.5$ Nm. The depicted torque is estimated with model [19] and filtered to reduce noise with a low-pass filter of first order according to Table III. The mechanical Power $P_{me}$

![Graph of measured step responses with ORCC and MPTC at different speeds](image)

![Graph of comparison of the measured step responses for different numbers of subiterations](image)

![Graph of experimental evaluation of the MPTC at the power limit](image)
results from the filtered estimated torque and the measured speed. The diagram shows that the torque can be provided below nominal speed (constant torque region). If nominal speed is exceeded, the constant power region is reached where the nominal power can be provided.

The steady-state loss evaluation is done by comparing the MPTC with the ORCC from [2]. For determination of the efficiency, the electrical power is measured by the power analyzer. The mechanical power is indirectly measured by the torque and the speed sensor. Fig. 12 presents the histogram of the efficiency difference \( \Delta \eta = \eta_{\text{MPTC}} - \eta_{\text{ORCC}} \). The efficiency of both methods is very similar in most of the 90 measured OPs. Approximately 94% of all OPs lie within the interval \([-0.544; 0.536]\) pp. The average difference is 0.08 pp. Consequently, the proposed MPTC solution is not only able to achieve fast transient torque dynamics in terms of settling time but also enables optimal efficiency operation in steady state.

IX. APPLICATION TO MMDS

A. Extension of the Single-Machine Equations

The equations of a single machine are reused and can simply be condensed to the equations for MMDS. In the following, the superscript number in brackets indicates the number of the respective machine. E.g., \( s_1 \) is the stator current of machine 1. The number of machines of the MMDS is introduced as \( N_m \). The order of the machines can be chosen arbitrarily.

In contrast to an SMDS, the total reference torque must be equal to the sum of all machine torques. Every single torque needs to be linearized so the torque is

\[
T \approx \sum_{i=1}^{N_m} \frac{\partial T^{(i)}}{\partial (s^{(i)})} (s^{(i)} - \bar{s}^{(i)}) + \frac{\partial T^{(i)}}{\partial \psi_{rd}^{(i)}} (\psi_{rd}^{(i)} - \bar{\psi}_{rd}^{(i)}) + T^{(i)}
\]

which needs to be considered in the soft-constraint. All derivatives are evaluated in the OP \( \bar{s}^{(i)}, \bar{\psi}_{rd}^{(i)} \) which is skipped in the following for readability. The same holds for the losses of the entire system

\[
P_l \approx \sum_{i=1}^{N_m} \left( i^{(i)}_s \right)^T K^{(i)}_{li} i^{(i)}_s + \left( \psi^{(i)}_{rd} \right)^T K^{(i)}_{\psi \psi} \psi^{(i)}_{rd} + \left( \alpha^{(i)}_i \right)^T \psi^{(i)}_{rd} + \left( h^{(i)}_{li} \right)^T \psi^{(i)}_{rd} - h^{(i)}_l
\]

(78)

which are part of the cost function.

The optimization vector is arranged as follows: At first, it contains the stator currents of machine 1 within the prediction horizon. Secondly, those of machine 2 follow and so on until those of the last machine. The slack variables complete the vector in the end. The condensed rotor flux linkage and the initial rotor flux linkage are arranged similarly:

\[
U_{kc} = \begin{bmatrix} i^{(1)}_s \varepsilon[0] \varepsilon[k_c - 1] \\ \vdots \\ \varepsilon[0] \varepsilon[k_c - 1] \end{bmatrix}, \quad \Psi_{kc} = \begin{bmatrix} \Psi^{(1)}_{kc} \\ \vdots \\ \Psi^{(N_m-1)}_{kc} \\ \Psi^{(N_m)}_{kc} \end{bmatrix},
\]

(79)

\[
\psi_{rd,0} = \begin{bmatrix} \psi^{(1)}_{rd,0} \\ \vdots \\ \psi^{(N_m-1)}_{rd,0} \\ \psi^{(N_m)}_{rd,0} \end{bmatrix}^T.
\]

(80)

Hence, the condensed rotor flux linkage can be calculated by the condensation of the rotor circuit dynamics of each machine according to (30) to

\[
\Psi_{kc} = A_{kc} \psi_{rd,0} + B_{kc} U_{kc}
\]

(81)

where the matrices have the following block structure:

\[
A_{kc} = \begin{bmatrix} A^{(1)}_{kc} & 0 & \cdots & 0 \\ 0 & \ddots & \vdots & \vdots \\ \vdots & \ddots & A^{(N_m-1)}_{kc} & 0 \\ 0 & \cdots & 0 & A^{(N_m)}_{kc} \end{bmatrix},
\]

(82)

\[
B_{kc} = \begin{bmatrix} B^{(1)}_{kc} & 0 & \cdots & 0 \\ 0 & \ddots & \vdots & \vdots \\ \vdots & \ddots & B^{(N_m-1)}_{kc} & 0 \\ 0 & \cdots & 0 & B^{(N_m)}_{kc} \end{bmatrix}.
\]

(83)

The submatrices \( A^{(i)}_{kc}, B^{(i)}_{kc}, i \in \{1, \ldots, N_m\} \) are defined like in (31) and (85), respectively, with the corresponding machine parameters. However, the last zero column in (85) does not apply to the submatrices \( B^{(i)}_{kc} \) since it concerns the slack variables.

The matrices and vectors of the constraints and the cost function can be set up in a similar way. Parameter tracking according to Sec. VII is done for every machine separately. Hence, the resulting QP’s problem space scales only linearly depending on the number of machines considered within an MMDS application. Operating strategies, like [1]–[3], that calculate the optimal steady-state reference currents do scale worse than linear, i.e., the extension towards MMDS applications comes at the expense of disproportional computational load.


B. Simulation Results of a Virtual MMDS

The MMDS case is tested with a virtual MMDS in a simulation consisting of two machines that are connected without a gearbox. As an example, the machine parameters $k_1$ and $k_2$ of the second machine are fifty percent larger than those of machine 1, all others are the same. That means the rotor-time constant of the second machine is larger than the one of the first machine. The used MPC parameters from Table VI are manually tuned. The simulated steady-state losses are compared with the optimal losses $P_{1,\text{opt}}$, which are calculated with the ORCC from [2]. The latter is done by considering all possible $T^{(1)} \cdot T^{(2)}$-combinations that fulfill $T^{(1)} + T^{(2)} = T^*$, calculating the optimal OPs for $T^{(1)}$, $T^{(2)}$ with [2] separately and calculating the losses of the most efficient torque combination.

Fig. 13 shows the simulation results for constant temperatures, constant speeds and a varying reference torque. The response to a reference torque step is delayed by the sample time of 30 ms. In steady state, the reference torque is controlled without a significant error in every OP. The steady-state losses of the MPTC in all four OPs are in average about 2.6 % higher than the minimum losses.

The simulation shows one benefit of the proposed MPTC for MMDS over an operating strategy with separately controlled torques of the machines: Even if the small machine does not contribute to the torque in steady state, it can compensate for the time constant of the large machine. At the beginning of a reference torque step, the large machine can only slowly build its rotor flux whereas the small machine can build it fast. This can be seen in Fig. 13, e.g., for $T^* = -15 \text{Nm}$. It results in a more dynamic transient behavior.

X. CONCLUSION AND OUTLOOK

This contribution presents an MPTC that controls the torque high-dynamically and minimizes the losses in steady state with the help of a precise electrical-thermal model of an IM published in [19]. Within the MPTC, the parameters are tracked and the optimization problem is solved in an iterative manner to achieve better results, especially in transients. Moreover, this approach can be easily applied to MMDS by extending the corresponding linearization and MPC schemes. To increase the transient performance in future research, a temporary increase of the current limits could lead to an integrated active thermal management included in the proposed MPTC, i.e., an overload operation [31]. Additionally, the overall system efficiency could be further improved by taking the inverter losses into account as well [32].

REFERENCES


TABLE VI: MPC parameters for the MMDS-case

<table>
<thead>
<tr>
<th>Symbol</th>
<th>Description</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>$k_c$</td>
<td>Number of prediction steps</td>
<td>10</td>
</tr>
<tr>
<td>$T_s$</td>
<td>Sampling time of the MPC</td>
<td>30 ms</td>
</tr>
<tr>
<td>$q_1$</td>
<td>Weighting of slack variable</td>
<td>10 000 W/(Nm)</td>
</tr>
<tr>
<td>$q_2$</td>
<td>Weighting of squared slack variable</td>
<td>10 000 W/(Nm)$^2$</td>
</tr>
<tr>
<td>$k_{opt}$</td>
<td>Number of extended time steps</td>
<td>10</td>
</tr>
<tr>
<td>$N_{s/v}$</td>
<td>Number of subiterations</td>
<td>10</td>
</tr>
<tr>
<td>$\lambda$</td>
<td>Damping factor</td>
<td>0.2</td>
</tr>
</tbody>
</table>

Fig. 13: Simulation results of an MMDS consisting of two machines with $n = 500 \text{min}^{-1}$ and $\vartheta_r = \vartheta_s = 20 ^\circ \text{C}$ for both machines
REFERENCES:


APPENDIX A

STRUCTURE OF SELECTED VECTORS AND MATRICES

The following notation is introduced: The element in the i-th row and j-th column of a matrix V is denoted by v_{ij}. Furthermore, the i-th element of a vector v is denoted by v_i. The matrix B_{k_e} from the rotor-circuit dynamics (27) is defined by (85). The quantities of the voltage constraint (49) are defined in (86) and (87). The matrices R_{k_e} and T_{k_e} are defined according to (88) and (90), respectively. The QP problem’s cost function for exact and approximated losses uses the matrices R_{k_e}, V_{k_e} and T_{k_e} from (88), (89) and (90), respectively.

APPENDIX B

SYMMETRY OF THE MPTC HESSIAN

The Hessian H in (65) is not symmetrical because the matrix M = B_{k_e}^\top T_{k_e} is not symmetrical. This can be proven, e.g., by calculating the element in the first row and third column and the element in the third row and first column that are not equal in general:

\[ m_{13} = b_{d,1}^1 b_{q,11}^1 + a_{d,1}^1 b_{d,1}^1 b_{q,11}^2 \neq m_{31} = 0. \]  

(84)

In contrast, the Hessian H in the relaxed problem formulation (72) is symmetrical because the matrix K_{ij} is symmetrical what can be seen in (62).
\[ B_{k_e} = \begin{bmatrix}
0^\top & 0^\top & \cdots & 0^\top & 0^\top \\
\mathbf{b}_d[0]^\top & \mathbf{b}_d[0]^\top & \cdots & \mathbf{b}_d[0]^\top & \mathbf{b}_d[0]^\top \\
\vdots & \vdots & \ddots & \vdots & \vdots \\
\mathbf{a}_d[1] \mathbf{b}_d[0] & \mathbf{a}_d[1] \mathbf{b}_d[0] & \cdots & \mathbf{a}_d[1] \mathbf{b}_d[0] & \mathbf{a}_d[1] \mathbf{b}_d[0] \\
\mathbf{a}_d[2] \mathbf{a}_d[1] \mathbf{b}_d[0] & \mathbf{a}_d[2] \mathbf{a}_d[1] \mathbf{b}_d[0] & \cdots & \mathbf{a}_d[2] \mathbf{a}_d[1] \mathbf{b}_d[0] & \mathbf{a}_d[2] \mathbf{a}_d[1] \mathbf{b}_d[0] \\
\vdots & \vdots & \ddots & \vdots & \vdots \\
\prod_{i=1}^{k_{e-1}} \mathbf{a}_d[i] \mathbf{b}_d[0] & \prod_{i=2}^{k_{e-1}} \mathbf{a}_d[i] \mathbf{b}_d[1] & \cdots & \mathbf{b}_d[k_e - 1] & \mathbf{0}^\top \\
\prod_{i=1}^{k_e} \mathbf{a}_d[i] \mathbf{b}_d[0] & \prod_{i=2}^{k_e} \mathbf{a}_d[i] \mathbf{b}_d[1] & \cdots & \mathbf{a}_d[k_e] \mathbf{b}_d[k_e - 1] + \mathbf{b}_d[k_e] & \mathbf{0}^\top \\
\vdots & \vdots & \ddots & \vdots & \vdots \\
\prod_{i=1}^{k_{e-1}} \mathbf{a}_d[i] \mathbf{b}_d[0] & \prod_{i=2}^{k_{e-1}} \mathbf{a}_d[i] \mathbf{b}_d[1] & \cdots & \sum_{i=k_{e-1}}^{k_e-2} \prod_{i=k_{e-1}}^{k_e-1} \mathbf{a}_d[j] \mathbf{b}_d[i] + \mathbf{b}_d[k_{e-1} - 1] & \mathbf{0}^\top \\
\end{bmatrix} \]  

(85)

\[
N_1[k] = \begin{bmatrix}
v_{i,11}[k] & v_{i,12}[k] \\
v_{i,21}[k] & v_{i,22}[k] \\
-v_{i,11}[k] & -v_{i,12}[k] \\
-v_{i,21}[k] & -v_{i,22}[k] \\
v_{i,11}[k] + v_{i,21}[k] & v_{i,12}[k] + v_{i,22}[k] \\
v_{i,11}[k] + v_{i,21}[k] & -v_{i,12}[k] + v_{i,22}[k] \\
v_{i,11}[k] - v_{i,21}[k] & -v_{i,12}[k] - v_{i,22}[k] \\
v_{i,11}[k] - v_{i,21}[k] & v_{i,12}[k] - v_{i,22}[k] \\
\pi_{sa}[k] v_{i,11}[k] + \pi_{sa}[k] v_{i,21}[k] & \pi_{sa}[k] v_{i,12}[k] + \pi_{sa}[k] v_{i,22}[k] \\
\end{bmatrix}
\]  

(86)

\[
\mathbf{n}_2[k] = \begin{bmatrix}
v_{\phi,11}[k] \\
v_{\phi,21}[k] \\
-v_{\phi,11}[k] \\
-v_{\phi,21}[k] \\
v_{\phi,11}[k] + v_{\phi,21}[k] \\
v_{\phi,11}[k] + v_{\phi,21}[k] \\
v_{\phi,11}[k] - v_{\phi,21}[k] \\
v_{\phi,11}[k] - v_{\phi,21}[k] \\
\pi_{sa}[k] v_{\phi,11}[k] + \pi_{sa}[k] v_{\phi,21}[k] \\
\end{bmatrix}, \quad \omega_{\phi}[k] = \begin{bmatrix} 1 \\
1 \\
1 \\
\sqrt{2} \\
\sqrt{2} \\
\sqrt{2} \\
\|\pi_{sa}[k]\|_2 \\
\end{bmatrix} \]  

(87)

\[
\mathbf{K}_{\phi}[0] = \begin{bmatrix}
0 & 0 & \cdots & 0 & 0 & 0 \\
0 & 0 & \cdots & 0 & 0 & 0 \\
0 & 0 & \cdots & 0 & 0 & 0 \\
0 & 0 & \cdots & 0 & 0 & 0 \\
0 & \cdots & 0 & 0 & 0 & 0 \\
0 & \cdots & 0 & 0 & 0 & 0 \\
\end{bmatrix} 
\]  

(88)

\[
\mathbf{h}_d[k] = \begin{bmatrix}
\mathbf{h}_d[0] \\
\mathbf{h}_d[1] \\
\vdots \\
\mathbf{h}_d[k_e - 2] \\
\mathbf{h}_d[k_e - 1] \\
\mathbf{h}_d[k_e - 2] \\
\end{bmatrix} + \sum_{k=k_e-1}^{k_e-2} \mathbf{h}_d[k] q_1^\top (k_{ext} + 2) q_1 \\
\]  

(89)

\[
\mathbf{y}_1[k] = \begin{bmatrix}
\mathbf{y}_1[k] \\
\mathbf{y}_1[k] \\
\vdots \\
\mathbf{y}_1[k] \\
\mathbf{y}_1[k] \\
\mathbf{y}_1[k] \\
\end{bmatrix}^\top 
\]  

(90)