The Journal of Engineering

The 9th International Conference on Power Electronics, Machines and Drives (PEMD 2018)

Non-linear MPC for winding loss optimised torque control of anisotropic PMSM

eISSN 2051-3305
Received on 25th October 2018
Accepted on 25th October 2018
doi: 10.1049/joe.2018.8143
www.ietdl.org

Christoph Schnurr[1] ✉, *Sören Hohmann*[1], *Johannes Kolb*[2]
[1]*Institute of Control Systems, Karlsruhe Institute of Technology, Karlsruhe, Germany*
[2]*SHARE am KIT, Schaeffler Technologies AG & Co. KG, Karlsruhe, Germany*
✉ *E-mail: c.schnurr@kit.edu*

Abstract: For a non-linear anisotropic permanent magnet synchronous machine (PMSM), a prediction model for model predictive control (MPC) considering effects like cross-coupling and saturation is developed in a straight forward procedure. The objective of the designed MPC is either tracking of reference currents or torque tracking. Both approaches use the projected fast gradient method (PFGM) as optimisation algorithm. The latter approach makes look-up-tables for current references obsolete and additionally minimises winding losses. This two approaches are compared in a simulation study with a state of the art PI controller.

1 Introduction

The scope of this paper is torque control of anisotropic permanent magnet synchronous machines (PMSM) in the rotor-oriented reference frame (*dq*-frame). The state of the art method for this task is PI control of the correspondent current components. For model predictive control (MPC), often many simplifications in the modelling of the PMSM and the voltage constraints are done, which deteriorates the performance compared to PI control. In general, MPC for drives can be classified in continuous control set (CCS) MPC with modulator and finite control set (FCS) MPC, which directly controls the hardware switches. FCS is advantageous, if constant current quality rather than constant switching frequency is important [1]. An advantage of CCS MPC is the higher switching frequency with the same sample time due to the modulator, which is relevant for computational intensive MPC approaches. In [2], a CCS was used, but saturation effects are neglected and voltage constraints are simplified. Also [3] utilises a CCS but does neither consider saturation nor inherently fulfils constraints, since it uses a closed-form solution rather than online optimisation. Infinite *d*- and *q*-current pairs can generate a specific torque value as long as the machine is not operating at the voltage limitation. Most MPC methods get the reference values for the current controller from a static mapping, transferring a given torque reference to a current reference in the *dq*-frame. The mapping is carefully calculated offline using maximum torque per ampere (MTPA) and maximum torque per voltage (MTPV) methods to select loss minimisation for steady state. See e.g. [1, 4] for a torque control or [5] for a speed control with CCS MPC approaches. This kind of MPC method will be denoted as i-MPC in the following. For loss optimised operation of an electrical drive, all machine and power electronic losses must be considered. The biggest impacts have the copper losses in the resistance of the machine windings, so they will be considered here. In general, one could incorporate any other losses, but the objective function may not be convex, what is important for the convergence of the optimisation algorithm. The control objective is stated as a weighted combination of torque tracking error and winding losses. Minimising losses as an additional goal in the objective function of a CCS MPC for PMSM control is done in [6, 7]. However, a simple linear machine model and simplified voltage constraints are used. The projected fast gradient method (PFMG) [8] was already used as online optimisation algorithm within the MPC for drive applications. This attests PFGM to be capable for real-time computation. The usage of this method for solving a MPC problem similar to this approach was presented in [1]. There, the direct current and the torque of a PMSM were controlled. However, the components of the flux vector in the *dq*-frame were used as states and no losses were considered in the cost function. Also [9] has shown that MPC with online optimisation with PFGM is capable for current control of a AC/DC transmitter.

This contribution presents a novel CCS MPC approach, which uses no reference values for the currents at all, since in the objective function the torque control error is penalised. The MPC will decide which combination of *d*- and *q*-voltage needs to be set to reach the control objective formulated in the cost function. This makes the generation and usage of look-up-tables for the references obsolete which simplifies and speeds up controller setup. Furthermore, the MPC allows online adaption of the model to varying machine parameters, which makes the regeneration of tables or changing the controller's parameters obsolete. Hence, this approach requires the losses to be considered in the objective function to cope with the freedom in choice of the currents in the *dq*-frame. The approach uses two dimensional look-up-tables for mapping fluxes to currents and thus iron saturation and cross-coupling effects are considered. Hence, the method is well suited for high performance interior PMSMs with non-linear magnetics. The new presented approach is also compared to the state of the art PI approach as well as to an i-MPC with current references.

2 Modelling of the PMSM

Since the MPC utilises a modulator, the complete modelling and control scheme will be shown in the *dq*-frame. First, the machine model from [10] will be presented in Section 2.1 with the properties mentioned above. In addition, all equations and the modelling will be shown in normalised variables. Hence, the system is described as per unit system.

2.1 Machine equations

As motivated before, the non-linear magnetics have to be considered. Hence, the functions describing the relation between the magnetic flux and the current

$$\psi_d(i_d, i_q), \tag{1a}$$

$$\psi_q(i_d, i_q), \tag{1b}$$

in the *dq*-frame have to be known. Typically (see e.g. [10]), they are considered in the form of look-up-tables depending on the currents in the *dq*-frame. Time dependencies of the currents i_d and i_q, the voltages u_d and u_q, the electrical angular velocity ω_{el} and the fluxes ψ_d and ψ_q are suppressed for better readability. This leads to the voltage equations

$$u_d = r_s i_d + \dot{\psi}_d - \omega_{\text{el}} \psi_q, \tag{2a}$$

$$u_q = r_s i_q + \dot{\psi}_q - \omega_{\text{el}} \psi_d, \tag{2b}$$

describing the dynamics of the PMSM with the normalised stator resistance r_s. To handle the dynamics of the plant, it is assumed that for every single optimisation step, the electrical angular velocity ω_{el} is constant. This may be appropriate in many applications, where the mechanical time constant is much bigger than the electrical one. The number of pole pairs and the constant 2/3 resulting from the amplitude invariant Park Transformation are included in the base of the normalisation constant, so the torque can be calculated as

$$m = \psi_d i_q - \psi_q i_d\,. \tag{3}$$

2.2 Affine state space model

Starting from the machine equations, the model is customised to allow MPC operation. To receive a convex optimisation function, we deduce an affine state space model. Applying the total differential to ψ_d and ψ_q given by (1) yields

$$\dot{\psi}_d = \frac{\partial \psi_d}{\partial i_d}\dot{i}_d + \frac{\partial \psi_d}{\partial i_q}\dot{i}_q, \tag{4a}$$

$$\dot{\psi}_q = \frac{\partial \psi_q}{\partial i_d}\dot{i}_d + \frac{\partial \psi_q}{\partial i_q}\dot{i}_q\,. \tag{4b}$$

Now (4) and (2) can be combined to a non-linear system of differential equations. After some calculus, one gets two coupled differential equations for the d- and q-current. Using the currents as states $\boldsymbol{x}^{\text{T}} = [i_d \quad i_q] \in \mathbb{R}^2$, the voltages as inputs $\boldsymbol{u}^{\text{T}} = [u_d \quad u_q] \in \mathbb{R}^2$, and the torque as output $m \in \mathbb{R}$, an affine state space description depending on look-up-tables of the flux is derived. Using the flux as state would lead to simpler dynamics (only integrator dynamics) and a time-invariant model [4]. Nevertheless, this is not appropriate for considering the winding losses in the objective function since they depend on the currents. We assume that in the neighbourhood of the current state, which is reachable within the prediction horizon of the MPC, it is sufficient to use a linearisation of the system dynamics. This can be validated by simulations as explained in Section 3. Then, a first-order Taylor approximation for the equations of the derivatives of i_d and i_q can be done. The system matrices of the affine system description depend on the current operating point $z^{\text{T}} = [\boldsymbol{x}^{\text{T}} \quad \boldsymbol{u}^{\text{T}}] \in \mathbb{R}^4$.The equations for the matrices describing this affine PMSM model are the basis for the discretisation step. As discussed later, simulations showed that discretisation with Euler approximation is not sufficient, which is in accordance with [11, 12]. The system was discretised with a third-order Picard Iteration [13] to provide remedy to that problem. This equals a third-order Taylor approximation of the transition matrix and leads to the affine discrete state space system

$$\boldsymbol{x}^+ = \boldsymbol{A}(z)\boldsymbol{x} + \boldsymbol{B}(z)\boldsymbol{u} + \boldsymbol{G}(z) \tag{5a}$$

$$m = \boldsymbol{C}(z)\boldsymbol{x} + F(z)\,. \tag{5b}$$

The equations for the matrix elements are rather complex but can be determined as explicit equations in advance. Online they are evaluated with the current operating point. From now on, the state dependency of the matrices will be left out for better readability.

2.3 MPC design

Getting a torque set point and the measurement of the currents in the dq-frame and the angular frequency ω_{el}, the MPC will provide a voltage vector in the dq-frame. In the following, the MPC controller synthesis is explained in detail, but first the model presented in the former chapter is included in the objective function of the MPC.

2.3.1 Model adaption for MPC: First, the model from (5) is expanded over the prediction horizon n_p. This method is known as stacking [8]. Therefore, the new vectors $\boldsymbol{x}_s(k) \in \mathbb{R}^{n_p \times 2}$, $\boldsymbol{u}_s(k) \in \mathbb{R}^{n_p \times 2}$ and $\boldsymbol{m}_s(k) \in \mathbb{R}^{n_p}$ encapsulate the discrete time values as shown for the states:

$$\boldsymbol{x}_s(k)^{\text{T}} = [\boldsymbol{x}(k) \cdots \boldsymbol{x}(k + n_p - 1)]\,. \tag{6}$$

So far, the dependencies from instant of time k were omitted, but to make things clear they are used temporarily now. From now on, all variables indexed with an s are determined by the above scheme. This leads to a new stacked model

$$\boldsymbol{x}_s(k+1) = \boldsymbol{A}_s x(k) + \boldsymbol{B}_s \boldsymbol{u}_s(k) + \boldsymbol{G}_s \tag{7a}$$

$$\boldsymbol{m}_s(k) = \boldsymbol{C}_s \boldsymbol{x}_s(k) + \boldsymbol{F}_s\,. \tag{7b}$$

Note that in (7a), $\boldsymbol{x}(k)$ is the current state rather than a stacked variable. The matrices are calculated by repeatedly applying (5) over the prediction horizon n_p.

2.3.2 Optimisation with constraints: First and most important, we want the torque m of the PMSM to track the reference $\hat{m}$. Hence, in the objective function, it is natural to penalise the control deviation $\Delta m = m - \hat{m}$. This is different to i-MPC, which will penalise $\boldsymbol{\Delta x} = \boldsymbol{x} - \hat{\boldsymbol{x}}$.The subsidiary control objective for the first approach is to minimise the function $P(\boldsymbol{x}) = \boldsymbol{x}^{\text{T}} \cdot \mathbf{I}\boldsymbol{x}$, where $\mathbf{I}$ is the identity matrix, hence it is already in quadratic form, which is standard for the cost function in MPC. This represents the winding losses. The winding resistance has not to be taken into account, since in the objective function the losses will be multiplied by a weighting factor anyway. The stacked variables can be used to formulate the objective functions in a single matrix equation

$$J_1(k, \boldsymbol{x}_s(k), \boldsymbol{m}_s(k), \hat{\boldsymbol{m}}_s(k)) = \| \boldsymbol{\Delta m}_s(k) \|_2^2 + \lambda P(\boldsymbol{x}_s(k))\,. \tag{8}$$

The weighting factor λ is introduced to adjust the dynamic behaviour. For comparison, also an i-MPC is implemented using the objective function

$$J_2(k, \boldsymbol{x}_s(k), \hat{\boldsymbol{x}}_s(k)) = \| \boldsymbol{\Delta x}_s(k) \|_2^2\,. \tag{9}$$

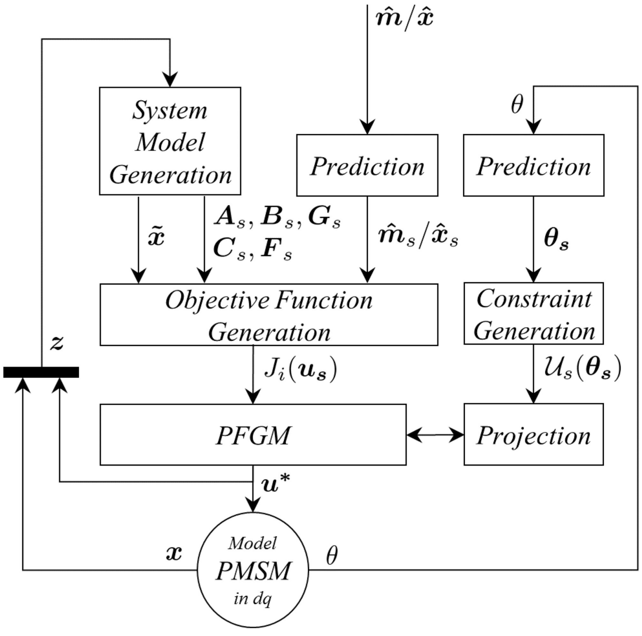
From now on, the time step dependency will be omitted again for better readability. Using the stacked model (7), (8) and (9) can be expressed solely depending on the voltage vector $\boldsymbol{u}_s$, since the current state and the stacked reference are constant during an optimisation cycle. This yields in

$$J_i(\boldsymbol{u}_s) = \boldsymbol{u}_s^T \boldsymbol{H}_i \boldsymbol{u}_s + \boldsymbol{h}_i^T \boldsymbol{u}_s, i \in \{1, 2\} \tag{10}$$

Note that here the matrices $\mathbf{H}_i$ and $\mathbf{h}_i$ depend on the operating point z of the previous iterate $k - 1$ and the stacked reference. Constant terms are omitted since they do not affect the optimisation result. This procedure is known as substitution or condensing [8], which is a crucial step in modifying the objective function in order to use projective algorithms like the PFGM. Due to thermal constraints, it is desirable to constraint the currents i.e. the states to a maximum rating. The substitution approach comes with the drawback that no hard state constraints are possible, since the objective does not depend on $\boldsymbol{x}_s$ anymore. Since hard state constraints are not desirable anyway – because of feasibility problems [8] – this may not be a disadvantage. By accepting small and short violations of the current limits during transients and the assumption on steady feasible references, we drop the state constraint in accordance with [1]. However, maximum current could be penalised by soft

```
Initializations:
$k = 1$
$\boldsymbol{u}_k = \boldsymbol{u}_{K,old} \in \mathcal{U}_s(\boldsymbol{\theta}_s)$ {warmstarting}
$\boldsymbol{u}_{k-1} = \boldsymbol{u}_k$ {first step is gradient step}
$\beta_k = 0$
$\alpha_k = \left\{\alpha_0 | 0 < \sqrt{\frac{\mu}{L}} \leq \alpha_0 \leq 1\right\}$
repeat
    $\hat{\boldsymbol{u}}_k = \boldsymbol{u}_k + \beta_k(\boldsymbol{u}_k - \boldsymbol{u}_{k-1})$
    $\hat{\boldsymbol{u}}_k = \hat{\boldsymbol{u}}_k - \frac{1}{L}\nabla J_i(\hat{\boldsymbol{u}}_k)$
    $\boldsymbol{u}_{\boldsymbol{k+1}} = \boldsymbol{\pi}\left(\hat{\boldsymbol{u}}_k\right)$
    $\alpha_{k+1} = \left\{\alpha_{k+1} | \alpha_{k+1}^2 = (1 - \alpha_{k+1})\alpha_k^2 + \frac{\mu\alpha_{k+1}}{L}\right\}$
    with $\alpha_{k+1} \in (0, 1)$
    $\beta_{k+1} = \dfrac{\alpha_k(1-\alpha_k)}{\alpha_k^2 + \alpha_{k+1}}$
    $k = k + 1$
until $k > K_{max} \vee \|\boldsymbol{u}_{k+1} - \boldsymbol{u}_k\| < \epsilon$
Result: $\boldsymbol{u}_K$ approximating $\boldsymbol{u}^*$,$K \leq K_{max} \in \mathbb{Z}$
```

Fig. 1 *Projected fast gradient method (PFGM) [8]*



$\hat{\boldsymbol{m}}/\hat{\boldsymbol{x}}$
System Model Generation
Prediction
θ
Prediction
$\tilde{\boldsymbol{x}}$
$\boldsymbol{A}_s, \boldsymbol{B}_s, \boldsymbol{G}_s$
$\boldsymbol{C}_s, \boldsymbol{F}_s$
$\hat{\boldsymbol{m}}_s/\hat{\boldsymbol{x}}_s$
$\boldsymbol{\theta_s}$
Objective Function Generation
Constraint Generation
$\boldsymbol{z}$
$J_i(\boldsymbol{u_s})$
$\mathcal{U}_s(\boldsymbol{\theta_s})$
PFGM
Projection
$\boldsymbol{u}^*$
$\boldsymbol{x}$
Model PMSM in dq
θ

Fig. 2 *Simulation setup and controller overview*

constraints also transformed to the voltage space, which will be ellipsoids in each subspace of one prediction step. The voltage constraint of a standard two-level three-phase voltage source inverter (VSI) in the voltage plane is a hexagon in the *dq*-frame rotating with ω_{el}. Hence, the orientation depends on the rotor angle θ. It will be described by the constraint $\boldsymbol{u} \in U(\theta)$. $\boldsymbol{U}_s(\boldsymbol{\theta}_s)$ is the expansion of $U(\theta)$ over the prediction horizon using $\boldsymbol{\theta}_s$. The resulting optimisation problem

$$\boldsymbol{u}_s^* = \arg \min_{\boldsymbol{u}_s \in \boldsymbol{U}_s(\boldsymbol{\theta}_s)} (J_i(\boldsymbol{u}_s)) \tag{11}$$

with the shown objectives is solved by the optimisation algorithm. The PFGM algorithm in Fig. 1 can now be used to efficiently solve (11) for both objectives J_1 and J_2.

It can be seen as a modified gradient search for the minimum of the functions J_i. In the PFGM algorithm (Fig. 1), the index *s* of the stacked voltage vector $\boldsymbol{u}_s$ is omitted to give space for the iteration index *k*. The parameter μ is the convexity parameter of J_i and *L* is the Lipschitz constant of ∇J_i. Whereas μ and *L* for a quadratic function can be defined as the smallest and the greatest eigenvalue of the Hessian, respectively [8]. Since the objectives J_i in form of (10) are quadratic functions μ and *L* are determined by calculating the eigenvalue of $2\boldsymbol{H}_i$. Since $\boldsymbol{H}_i$ is changing during operation, μ and *L* are determined online. Also indicated in the PFGM algorithm (Fig. 1) is the utilisation of the optimisation result from the previous optimisation result as starting point $\boldsymbol{u}_1$. This is known as warmstarting [8]. Two termination criteria are implemented. First, the optimisation is stopped, if the Euclidean norm of the deviation of the voltage vector from one step to another is smaller than the design parameter ϵ. Second, to speed up the computation, the iteration also terminates if a maximum number of $K_{\max}$ iterations is reached. The latter criterion is also called early termination. It leads to suboptimal optimisation results. Together with the warmstarting in steady states, very precise results are reached.

The projection operator $\boldsymbol{\pi}(\cdot)$ in the PFGM algorithm (Fig. 1) is

$$\boldsymbol{\pi}(\boldsymbol{u}) = \begin{cases} \arg \min\limits_{\boldsymbol{w} \in \boldsymbol{U}_s(\boldsymbol{\theta}_s)} \| \boldsymbol{u} - \boldsymbol{w} \|, & \boldsymbol{u} \notin \boldsymbol{U}_s(\boldsymbol{\theta}_s) \\ \boldsymbol{u}, & \boldsymbol{u} \in \boldsymbol{U}_s(\boldsymbol{\theta}_s) \end{cases}. \tag{12}$$

This projection on the symmetrical polytopic constraint set $\boldsymbol{U}_s(\boldsymbol{\theta}_s)$ can be calculated explicitly without further optimisation. This is the reason for the efficient deployment of the PFGM in the MPC. The projection ensures that every iteration *k* ends with feasible $\boldsymbol{u}_{\text{k}}$, that lies within the voltage constraint of the VSI.

2.3.3 Overall scheme: In Fig. 2, an overview of the resulting controller structure is given. Applications may not provide knowledge of future reference values, hence $\hat{\boldsymbol{m}}_s$ is solely the n_p times repetition of $\hat{m}$ and is used for the calculation of $\Delta\boldsymbol{m}_s$. The same holds for $\hat{\boldsymbol{x}}_s$ if i-MPC is used. In the right *Prediction* block, the calculation of $\boldsymbol{\theta}_s$ is straight forward, since the sample time and ω_{el} are known and assumed to be constant. The steps to calculate the matrices of the operating point-dependent affine state space model from Section 2.2 are encapsulated in the block *Matrix Generation*. In this block, also a prediction of the state one sample time step ahead $\tilde{\boldsymbol{x}}$ is calculated with the generated model to compensate the delay time of one sample due to the computation. The procedure presented in Section 2.3.2 is contained in the blocks *Objective Function*, *Constraint Generation,* and *PFGM*. In the *Projection* block, the function $\boldsymbol{\pi}(\boldsymbol{u}_s)$ from (12) is implemented. Note that as typical for MPC, the receding horizon strategy is applied. Hence, the *PFGM* block in Fig. 2 only applies the first entry of $\boldsymbol{u}_s$* to the PMSM and the others are discarded.

3 Simulation study

The PMSM model used for the simulation is experimentally validated, it was also used in [10]. This machine model is implemented in MATLAB Simscape and simulated with a variable-step solver using a relative tolerance of 0.2%. The MPC is implemented in MATLAB Simulink and runs with a constant sampling time of 125 µs which corresponds to a switching frequency of the VSI of 8 kHz. The VSI is assumed to be a two-level inverter with a DC link voltage of 400 V and is modelled as an ideal voltage source. All simulations are done in the *dq*-frame, thus the set voltage from the controller is directly applied as input to the machine model. The PI controller with a static mapping from reference torque to reference currents (classical MTPA and MTPV strategy) used for comparison is the same as in [14].

The model of the PMSM, which is designed for electric vehicles, has parameters as seen in Table 1. The machine has interior permanent magnets and hence shows a reluctance torque. The flux depends on currents as shown in Figs. 3 and 4. Therefore, cross-coupling and saturation effects are present. The very same tables are used in (1) for the MPC model generation.

3.1 Simulation results

Simulations shown here are carried out at constant angular velocity of the drive i.e. 5000 1/min. This imitates an experimental setup with a much stronger machine as load. Before the controller is tested, simulations are used to verify the affine state space model,

Table 1 Machine properties

nom. power	P_n	90.32 kW
nom. speed	Ω_n	5000 1/min
max. torque at Ω_n	$M_{max,N}$	170 Nm
number of pole pairs	p	3
resistance	R_s	0.0284 Ω
PM flux linkage	Ψ_{PM}	0.1019 Wb

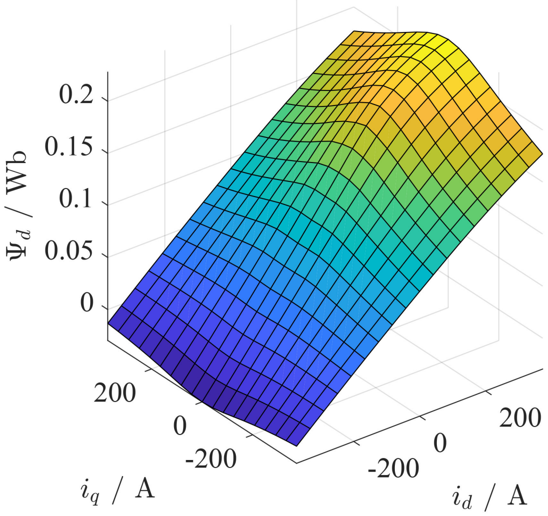


Fig. 3 *Look-up-table used for Ψ_d*

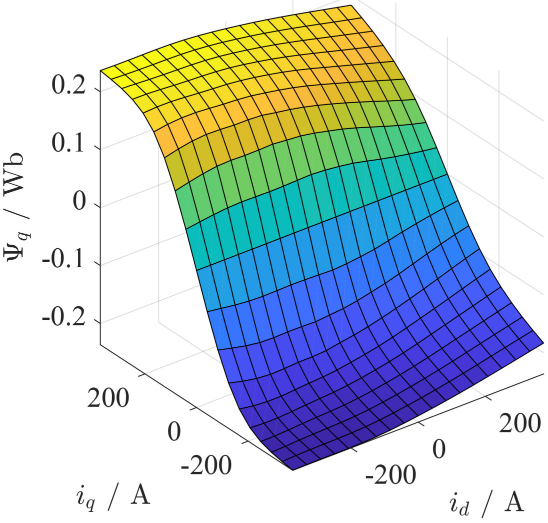


Fig. 4 *Look-up-table used for Ψ_q*

which will be used for the MPC. As mentioned before, the discretisation with Euler approximation is not sufficient. As a result, the simulations showed that a third-order Picard iteration is a good choice. At lower orders, the model becomes instable at higher speed but a further increase does not lead to a significant increase in model accuracy.

The receding linearisation technique proposed in [15] can only marginally improve the model accuracy. It uses multiple operating points over the prediction horizon determined by applying the predicted voltage vector from the optimisation of the previous instant of time to (7). This shows that the linearisation approach with a constant operating point during one prediction horizon is well suited for the matter in hand. The receding linearisation technique was also used for model generation during the presented controller simulations.

Fig. 5 shows the comparison of the response to a torque reference step of three controllers. The presented i-MPC is shown, using the same look-up-tables as the PI controller for the mapping

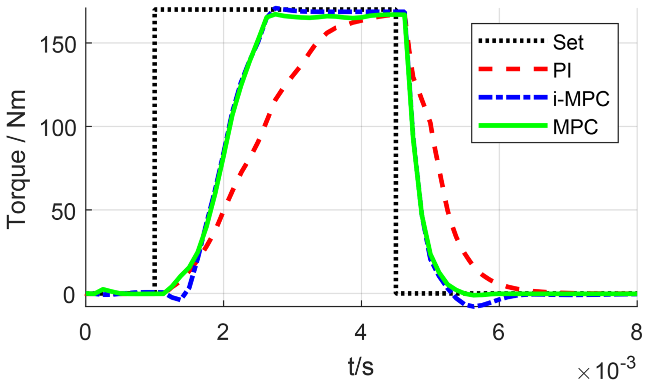


Fig. 5 *Torque reference step to maximum torque*

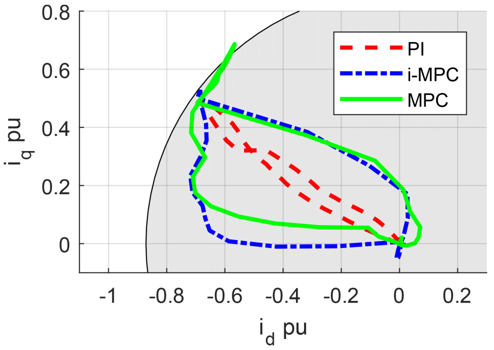


Fig. 6 *Current trajectory for reference step in Fig. 5*

from torque reference to current references. It can be seen that the proposed MPC reaches very similar performance compared to the i-MPC. Both of them show superior performance compared to the PI controller. The current plane in the *dq*-frame in Fig. 6 shows also close relation of the MPC approaches and their difference to PI control. Note that the maximum current circle (grey) has a smaller radius than one, since for the basis of the normalisation is chosen as the maximum current of the simulation model. The MPC spreads in transients the voltage vector over *d*- and *q*-axes differently compared to PI control. This results in faster *q*-current dynamics but slower *d*-current dynamics in the very beginning, which can also be seen in Fig. 7. This setup has the aim to reach short computation times while preserving good performance. Even in the case of early termination after $K_{\max} = 6$ iterations, the PFGM algorithm leads to a good control performance. The termination bound can be chosen to $\varepsilon = 1.9 \times 10^{-3}$, which corresponds to 0.5 V. To reach minimal computation times, the prediction horizon is determined of course to a minimum, which reaches still good performance. For the shown simulation, it was set to $n_p = 3$. In general, the choice of the loss factor λ affects the dynamics. Higher values will increase the importance of the losses, which will slow down the dynamic behaviour. For the proposed MPC, it has an additional effect. In Fig. 7, it can also be seen that the currents oscillate much more compared to i-MPC approach. However, they oscillate exactly along the torque hyperbola, which can be seen in Fig. 6. This leads to the fact that the torque settles as fast as the i-MPC, while the currents still move towards their steady state. The oscillation will settle much quicker as λ is increased, but this again will slow down dynamics. For the shown simulations, a good trade off was found for $\lambda = 5 \times 10^{-3}$. The computation time of the optimisation with a desktop PC (Intel Core i5 CPU @ 2.5 GHz and 8 GB RAM @ 800 MHz) is within 20 ms.

4 Conclusions

The proposed MPC method that directly controls the torque makes the usage of look-up-tables for current references obsolete. Therefore, this novel method follows first and foremost the minimisation of the torque tracking error during a transient. When the torque tracking error is settled at the end of the transient, the

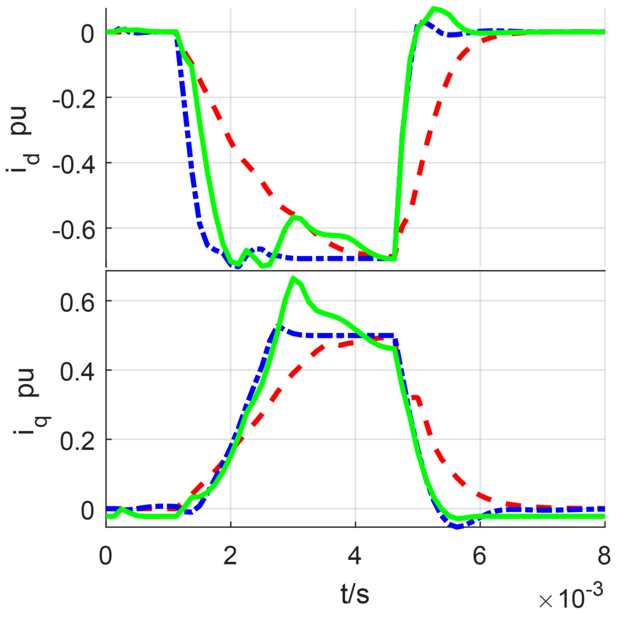


Fig. 7 *Time evolution of the currents for the reference step from Fig. 5 (legend ibidem)*

loss minimisation increases its influence and hence the currents ensue their winding loss optimal steady state. However, compared to the presented i-MPC approach, the tuning is more complex since additionally to the parameters of PFGM the loss weighting factor has to be chosen carefully to restrict the oscillations in the currents. Both MPC approaches show good performance even at steps to maximum torque compared to state of the art PI control. This confirms that the presented linearisation and discretisation approach is well suited for the generation of prediction models for MPC controlling anisotropic PMSM in saturation. The PI controller could also be tuned to be faster, but this will result in overshoots, which is not observed for the MPC approaches.

5 References

[1] Preindl, M., Bolognani, S., Danielson, C.: ‘Model predictive torque control with PWM using fast gradient method’. IEEE Applied Power Electronics Conf. and Exposition (APEC), Long Beach, CA, USA, 2013, vol. **28**, pp. 2590–2597

[2] Cimini, G., Bernardini, D., Bemporad, A., *et al.*: ‘Online model predictive torque control for permanent magnet synchronous motors’. IEEE Int. Conf. on Industrial Technology (ICIT), Seville, Spain, 2015, pp. 2308–2313

[3] Errouissi, R., Al-Durra, A., Muyeen, S., *et al.*: ‘Continuous-time model predictive control of a permanent magnet synchronous motor drive with disturbance decoupling’, *IET Electr. Power Appl.*, 2017, **11**, (5), pp. 697–706

[4] Preindl, M.: ‘Robust control invariant sets and lyapunov-based MPC for IPM synchronous motor drives’, *IEEE Trans. Ind. Electron.*, 2016, **63**, pp. 3925–3933

[5] Carpiuc, S., Lazar, C.: ‘Energy-efficient model predictive speed control of permanent magnet synchronous machine based automotive traction drives’. IEEE Vehicle Power and Propulsion Conf. (VPPC), Coimbra, Portugal, 2014, pp. 1–6

[6] Stumper, J.-F., Dötlinger, A., Jung, J., *et al.*: ‘Predictive control of a permanent magnet synchronous machine based on real-time dynamic optimization’. European Conf. on Power Electronics and Applications, Birmingham, UK, 2011

[7] Stumper, J.-F., Dötlinger, A., Kennel, R.: ‘Classical model predictive control of a permanent magnet synchronous motor’, *Eur. Power Electron. Drives J.*, 2012, **22**, (3), pp. 24–31

[8] Borrelli, F., Bemporad, A., Morari, M.: ‘*Predictive control for linear and hybrid systems*’ (Cambridge University Press, Cambridge, UK, 2017)

[9] Richter, S., Mariéthoz, S., Morari, M.: ‘High-speed online MPC based on a fast gradient method applied to power converter control’. American Control Conf., Baltimore, MD, USA, 2010, pp. 4737–4743

[10] Richter, J., Gemaßmer, T., Doppelbauer, M.: ‘Predictive current control of saturated cross-coupled permanent magnet synchronous machines’. Int. Symp. on Power Electronics, Electrical Drives, Automation and Motion, Ischia, Italy, 2014, pp. 830–835

[11] Rojas, C. A., Yuz, J. I., Aguirre, M., *et al.*: ‘A comparison of discrete-time models for model predictive control of induction motor drives’. IEEE Int. Conf. on Industrial Technology, Seville, Spain, 2015, pp. 568–573

[12] Graf, M., Otava, L., Buchta, L.: ‘Simple linearization approach for MPC design for small PMSM with field weakening performance’. Conf. on Programmable Devices and Embedded Systems (PDES), Krakow, Poland, 2015, vol. **13**, pp. 159–164

[13] Rodríguez-Millán, J., Patete, A., González, C.: ‘Picard discretization of nonlinear systems: symbolic or numeric implementation?’, *Lect. Notes Comput. Sci.*, 2007, **4739**, pp. 121–129

[14] Gemaßmer, T., Schnarrenberger, M., Späth, H.: ‘Simple strategy of overmodulation in control of interior permanent magnet synchronous machines for improving efficiency in automotive applications’. PCIM Europe, Nuremberg, Germany, 2013, pp. 231–238

[15] Rahideh, A.B., Shaheed, M.: ‘Constrained output feed-back model predictive control for nonlinear systems’, *Control Eng. Pract.*, 2012, **20**, (4), pp. 431–443