

Trajectory Extension Methods for Model Predictive Direct Torque Control

Yashar Zeinaly*, Tobias Geyer† and Bo Egardt‡

*Delft Center for Systems and Control, Delft University of Technology, The Netherlands

Email: y.zeinaly@tudelft.nl

†Department of Electrical and Computer Engineering, The University of Auckland, New Zealand

Email: t.geyer@ieee.org

‡Department of Signals and Systems, Chalmers University of Technology, Sweden

Email: bo.egardt@chalmers.se

Abstract—This paper focuses on model predictive direct torque control (MPDTC), which is a recent control scheme for three-phase ac electric drives combining the notions of model predictive control (MPC) and direct torque control (DTC). Using a dynamic model of the drive, MPDTC predicts several future switch transitions, extends the outputs and chooses the inverter switch positions that minimize the switching frequency or the switching losses. The performance of MPDTC depends on the accuracy of the predictions. However, MPTDC schemes with very accurate predictions are computationally demanding necessitating very fast controller hardware. New methods for extending the output trajectories are proposed that yield fast yet accurate predictions giving rise to a computationally efficient MPDTC scheme. The advantages of the proposed methods are shown in terms of the associated computational complexity and the accuracy of the predictions.

I. INTRODUCTION

The idea underlying MPDTC [1]–[3] is to replace the switching table in conventional DTC [4], [5] with a constrained optimal controller with a receding horizon policy [6]–[9]. It is shown in [1]–[3], [10]–[12] that MPDTC significantly improves the performance of DTC by reducing the switching frequency of the inverter device. The control objectives are to keep the machine’s electromagnetic torque and the stator flux magnitude within predefined hysteresis bounds. With 3-level neutral point clamped inverters [13], it is also necessary to balance the neutral point potential.

The *switching horizon* [10] is composed of *switching events* ‘S’ and *extension events* ‘E’. During the ‘S’ event, all admissible¹ switching transitions are enumerated. For each switching transition, the outputs (i.e. the electromagnetic torque, the stator flux magnitude and the inverter neutral point potential) are predicted using a discrete-time model of the drive sampled with the sampling interval $T_s = 25\mu\text{s}$. To achieve a long *prediction horizon* N_p , the predicted output trajectories are extended during the ‘E’ event. Only the *candidate sequences* are considered, which are the sequences, whose associated

outputs are either *feasible*, or *pointing in the proper direction*² at any intermediate step during the switching horizon. For each candidate sequence, N_p is given by the time duration for which the extended predicted outputs remain within their bounds. The average switching frequency is calculated as the total number of switching transitions in the switching sequence divided by the prediction horizon length. The optimal switching sequence is obtained by minimizing the average switching frequency over all switching sequences. According to the receding horizon policy, only the first input of the optimal sequence is applied to the drive and this procedure is repeated at the next controller sampling time $(k + 1)T_s$ with new measurements.

The most accurate way of extending the output trajectories is to use the internal model of the drive in an open-loop simulation [10]. Even though this requires excessive computational power making this approach impractical, this method is regarded as a benchmark to evaluate the performance of other extension methods against. In [1], [2] *linear* and *quadratic extrapolation* were proposed to extend the output trajectories. The computational burden involved can be handled with the currently available controller hardware. Particularly the linear extrapolation method was successfully implemented on the existing DTC hardware [2]. As will be shown in this paper the accuracy of the predictions can be further increased, without introducing much additional computational effort, by the following proposed methods. (i) prediction with interpolation, (ii) iterative prediction with interpolation and (iii) an analytical approach, where a simplified model of the drive is used to extend the output trajectories analytically.

II. DRIVE MODEL

Throughout the paper a three-level neutral point clamped inverter driving an induction machine is used as an illustrative example for an ac drive system, as shown in Fig. 1. Assuming that the induction motor is three-phase symmetrical and its magnetic core is linear with a negligible core loss, (1a)–(1d)

This work was performed while the first author was at Chalmers University of Technology, Sweden.

¹Not all switching transitions are possible due to physical constraints on the inverter.

²A trajectory points in the proper direction if the magnitude of the bound violation is strictly decreasing over time.

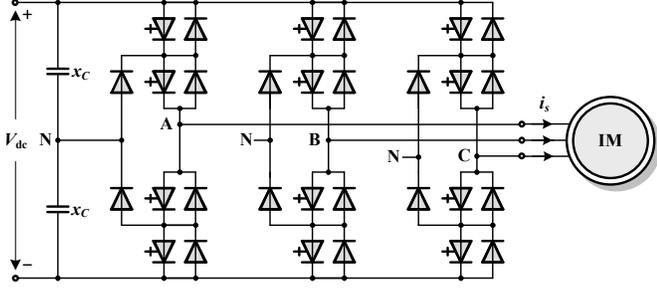


Figure 1: Three-level neutral point clamped VSI inverter driving an induction motor.

describe the dynamical model of the induction machine in the $\alpha\beta$ reference frame. $\psi_{s\alpha}$, $\psi_{s\beta}$, $\psi_{r\alpha}$ and $\psi_{r\beta}$ are the α - and β -components of the stator and the rotor flux linkages, and v_α and v_β are the stator voltages in the stationary reference frame [14]. The model parameters are the electrical angular velocity of the rotor ω_r , the stator and the rotor winding resistances r_s and r_r , and the stator, the rotor and the mutual reactances x_{ls} , x_{lr} and x_m .

$$\dot{\psi}_{s\alpha} = -r_s \frac{x_{rr}}{D} \psi_{s\alpha} + r_s \frac{x_m}{D} \psi_{r\alpha} + v_\alpha \quad (1a)$$

$$\dot{\psi}_{s\beta} = -r_s \frac{x_{rr}}{D} \psi_{s\beta} + r_s \frac{x_m}{D} \psi_{r\beta} + v_\beta \quad (1b)$$

$$\dot{\psi}_{r\alpha} = r_r \frac{x_m}{D} \psi_{s\alpha} - r_r \frac{x_{ss}}{D} \psi_{r\alpha} - \omega_r \psi_{r\beta} \quad (1c)$$

$$\dot{\psi}_{r\beta} = r_r \frac{x_m}{D} \psi_{s\beta} + \omega_r \psi_{r\alpha} - r_r \frac{x_{ss}}{D} \psi_{r\beta} \quad (1d)$$

In the above, we introduced $x_{ss} = x_{ls} + x_m$, $x_{rr} = x_{lr} + x_m$ and $D = x_{rr}x_{ss} - x_m^2$.

The torque is defined as

$$T_e = \frac{x_m}{D} (\psi_{s\beta} \psi_{r\alpha} - \psi_{s\alpha} \psi_{r\beta}) \quad (2)$$

and the length of the stator flux is given by

$$\Psi_s = \sqrt{\psi_{s\alpha}^2 + \psi_{s\beta}^2} \quad (3)$$

Please note that all rotor variables and parameters are referred to the stator side. Throughout this paper, we assume that we are using normalized quantities.

The switch position of the three-level inverter is denoted as $\mathbf{u}_{abc} = [u_a \ u_b \ u_c]^T$, where $u_a, u_b, u_c \in \{-1, 0, 1\}$ represent the switch position of each inverter phase leg. The inverter's neutral point potential v_n is described by [1], [3]

$$\dot{v}_n = \frac{1}{2x_c} |\mathbf{u}_{abc}|^T \mathbf{P}^{-1} \mathbf{i}_{s,\alpha\beta 0}, \quad (4)$$

where

$$\mathbf{P} = \frac{2}{3} \begin{bmatrix} 1 & -\frac{1}{2} & -\frac{1}{2} \\ 0 & \frac{\sqrt{3}}{2} & -\frac{\sqrt{3}}{2} \\ \frac{1}{2} & \frac{1}{2} & \frac{1}{2} \end{bmatrix} \quad (5)$$

is the three phase to $\alpha\beta 0$ transformation matrix. x_c is one of the two dc-link capacitors, and $|\mathbf{u}_{abc}| = [|u_a|, |u_b|, |u_c|]^T$ is the component-wise absolute value of the inverter switch position. $\mathbf{i}_{s,\alpha\beta 0}$ is the stator current in the stationary reference frame. The α - and β -components of $\mathbf{i}_{s,\alpha\beta 0}$ are related to the stator and the rotor flux components according to (6) while the 0 component is always zero [1], [3].

$$\mathbf{i}_{s,\alpha\beta 0} = \frac{1}{D} \begin{bmatrix} x_{rr} \psi_{s\alpha} - x_m \psi_{r\alpha} & x_{rr} \psi_{s\beta} - x_m \psi_{r\beta} & 0 \end{bmatrix}^T \quad (6)$$

The stator voltages in the stationary reference frame, v_α and v_β , are given by

$$\mathbf{v}_{\alpha\beta 0} = \frac{V_{dc}}{2} \mathbf{P} \mathbf{u}_{abc}, \quad (7)$$

where V_{dc} is the dc-link voltage of the inverter.

III. EXTENSION METHODS

For the methods described in this section, the model introduced in Sect. II is discretized with the sampling-time T_s . The resulting discrete-time model can be found in [1], [3]. In accordance with the prediction model introduced in [1], [3], the following notation is defined.

- $\mathbf{x} = [\psi_{s\alpha} \ \psi_{s\beta} \ \psi_{r\alpha} \ \psi_{r\beta} \ v_n]^T$ is the state vector. The i -th element of this vector is denoted by x_i .
- $\mathbf{y} = [T_e \ \Psi_s \ v_n]^T$ is the output vector. The i -th element of this vector is denoted by y_i .
- $\mathbf{u} = \mathbf{u}_{abc}$ denotes the input vector.
- $f(\cdot)$ is the drive's state update equation discretized with the sampling interval T_s .
- $f_d(\cdot)$ is the drive's state update equation discretized with the coarse sampling interval dT_s .
- $g(\cdot)$ is the drive's output equation.

In deriving the formulae for the different extension methods, for the sake of simplicity, we limit ourselves to the switching horizon 'SE', which means a switching action at time-step k is followed by an extension event. It is straightforward to generalize the results to an arbitrary switching horizon.

A. Open-Loop Simulation (OL)

The switch position at time-step k , $\mathbf{u}(k)$, is applied to the discrete-time model of the drive to compute the outputs from time-instant k on. The trajectories are extended until one of the outputs hits a bound, which determines the length of the predicted trajectory, or equivalently, the prediction horizon length N_p . The extended trajectories are described from time-step $k+1$ to $k+N_p$ according to

$$\mathbf{x}(k+n) = f(\mathbf{x}(k+n-1), \mathbf{u}(k)) \quad (8)$$

$$\mathbf{y}(k+n) = g(\mathbf{x}(k+n)),$$

where $0 \leq n \leq N_p$. For an example, see Fig. 2a. More details about this method can be found in [3] and in [10].

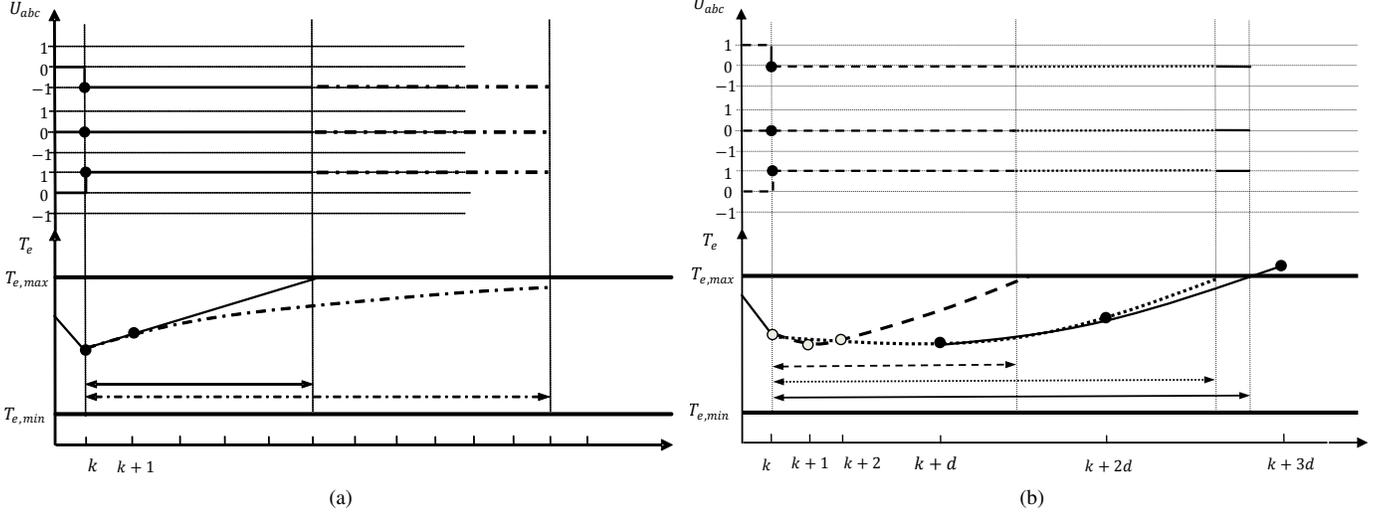


Figure 2: Examples of predicted torque trajectories between their bounds and the associated switch positions for the switching horizon 'SE'. The 'S' event happens at time-step k . (a) Comparison between the OL (dash-dotted line) and LE (solid line) methods and the associated prediction horizon lengths (bi-directional arrows). The points (bold dots) used for the LE method are shown as well. (b) Comparison between the QE (dashed), PQI (dotted) and IPQI (solid) methods and their prediction horizon lengths. The QE method uses the points at time-instants k , $k + 1$ and $k + 2$. The PQI method uses the points at time-instants k , $k + d$ and $k + 2d$. The IPQI method uses the additional point at time-step $k + 3d$.

B. Linear Extrapolation (LE)

The switch position $\mathbf{u}(k)$ is applied to the discrete-time model of the drive once to obtain the output at time-instant $k + 1$ according to

$$\begin{aligned} \mathbf{x}(k+1) &= f(\mathbf{x}(k), \mathbf{u}(k)) \\ \mathbf{y}(k+1) &= g(\mathbf{x}(k+1)). \end{aligned} \quad (9)$$

The output trajectories are then extrapolated from time-step $k + 1$ on linearly using the samples at time-steps k and $k + 1$ until an output variable hits a bound. The extended trajectories from time-step $k + 1$ to $k + N_p$ are given by

$$\mathbf{y}(k+n) = (\mathbf{y}(k+1) - \mathbf{y}(k))n + \mathbf{y}(k), \quad (10)$$

where $0 \leq n \leq N_p$. The time-instant when the i -th output variable crosses a bound is easily calculated and denoted as n_i . The length of the predicted trajectory, N_p , is given by

$$N_p = \min_i \{n_i\} \quad (11)$$

This method is discussed in detail in [1], [3] and was successfully implemented and tested on ABB's ACS6000 medium-voltage drive [2]. See Fig. 2a for an example.

C. Quadratic Extrapolation (QE)

As discussed in [1] and [3], the machine's output trajectories can be better approximated by quadratic curves rather than straight lines. The switch position $\mathbf{u}(k)$ is applied twice to

the drive model to obtain the outputs at time-instants $k + 1$ and $k + 2$ according to

$$\begin{aligned} \mathbf{x}(k+1) &= f(\mathbf{x}(k+n-1), \mathbf{u}(k)), \quad n = 1, 2 \\ \mathbf{y}(k+1) &= g(\mathbf{x}(k+n)), \quad n = 1, 2 \end{aligned} \quad (12)$$

Using the output samples at time-instants k , $k + 1$ and $k + 2$ the output trajectories are extrapolated quadratically from time-step $k + 2$ on until the first of the three output variables leaves a bound. The extended trajectories from time-step k to $k + N_p$ are given by

$$\mathbf{y}(k+n) = \mathbf{a}n^2 + \mathbf{b}n + \mathbf{c}, \quad 0 \leq n \leq N_p \quad (13)$$

where the quadratic coefficient vectors \mathbf{a} , \mathbf{b} and \mathbf{c} are obtained from

$$\mathbf{y}(k) = \mathbf{c} \quad (14a)$$

$$\mathbf{y}(k+1) = \mathbf{a} + \mathbf{b} + \mathbf{c} \quad (14b)$$

$$\mathbf{y}(k+2) = 4\mathbf{a} + 2\mathbf{b} + \mathbf{c}, \quad (14c)$$

which yields

$$\mathbf{a} = \frac{1}{2}\mathbf{y}(k) - \mathbf{y}(k+1) + \frac{1}{2}\mathbf{y}(k+2) \quad (15a)$$

$$\mathbf{b} = -\frac{3}{2}\mathbf{y}(k) + 2\mathbf{y}(k+1) - \frac{1}{2}\mathbf{y}(k+2) \quad (15b)$$

$$\mathbf{c} = \mathbf{y}(k) \quad (15c)$$

It is trivial to calculate where the i -th output variable hits its bounds. The length of the prediction horizon is given by

(11). Fig. 2b provides an example of how the trajectories are extended using this method.

D. Prediction with Quadratic Interpolation (PQI)

Predicting the output samples far ahead in the future and then interpolating between them yields extended trajectories with better accuracy than QE. The algorithm uses the switch position $\mathbf{u}(k)$ to compute the output samples at time-instants $k + d$ and $k + 2d$ according to

$$\begin{aligned} \mathbf{x}(k + nd) &= f_d(\mathbf{x}(k + (n - 1)d), \mathbf{u}(k)), \quad n = 1, 2 \quad (16) \\ \mathbf{y}(k + nd) &= g(\mathbf{x}(k + nd)), \quad n = 1, 2 \end{aligned}$$

Interpolating between outputs at time-instants k , $k + d$ and $k + 2d$ quadratically yields the extended trajectories. The trajectories are extended until the first out of the three output variables becomes infeasible. The extended trajectories from time-step k to $k + N_p$ are given by (13) with

$$\begin{aligned} \begin{bmatrix} \mathbf{a}^T \\ \mathbf{b}^T \end{bmatrix} &= \begin{bmatrix} d^2 & d \\ 4d^2 & 2d \end{bmatrix}^{-1} \begin{bmatrix} \mathbf{y}^T(k + d) - \mathbf{y}^T(k) \\ \mathbf{y}^T(k + 2d) - \mathbf{y}^T(k) \end{bmatrix} \\ \mathbf{c} &= \mathbf{y}(k), \end{aligned}$$

or equivalently by

$$\begin{aligned} \mathbf{a} &= \frac{1}{2d^2} \mathbf{y}(k) - \frac{1}{d^2} \mathbf{y}(k + d) + \frac{1}{2d^2} \mathbf{y}(k + 2d) \quad (17) \\ \mathbf{b} &= -\frac{3}{2d} \mathbf{y}(k) + \frac{2}{d} \mathbf{y}(k + d) - \frac{1}{2d} \mathbf{y}(k + 2d) \\ \mathbf{c} &= \mathbf{y}(k) \end{aligned}$$

Again, the time-step when the i -th output variable hits its bound can be easily calculated. The length of the prediction horizon is given by (11). The performance of this method heavily depends on the choice of d . It should be chosen such that $2d$ covers the prediction horizon with a high probability. See Fig. 2b for more details.

E. Iterative Prediction with Quadratic Interpolation (IPQI)

Particularly for long prediction horizons, the extrapolation (interpolation) based on few predicted output samples can give rise to large errors with respect to the open-loop simulation method. If required, the IPQI method uses additional predicted output samples to refine the extrapolation as the length of the extended trajectories increases. As for the PQI method, a quadratic curve is interpolated between the output samples at time-instants k , $k + d$ and $k + 2d$. If no output bound is violated in the $[k, k + 2d]$ interval, the time is shifted forward by d time-steps and the outputs at time instant $k + 3d$ are predicted using (16). A new quadratic curve is interpolated between the output samples at time-instants $k + d$, $k + 2d$ and $k + 3d$. This procedure is continued until a bound violation is detected. The extended trajectories associated with the IPQI method are composed of m segments. Each segment is characterized by a second order polynomial. The number of segments is calculated from

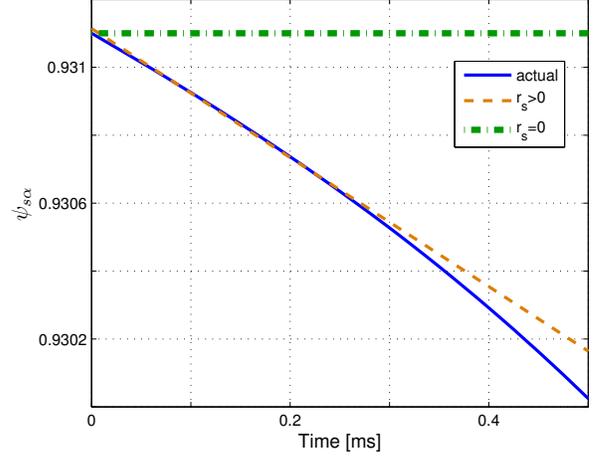


Figure 3: Trajectories of the stator flux α -component for some initial conditions and $u_k = [0\ 0\ 0]^T$. For the zero voltage vectors, the approximated flux trajectory associated with the $r_s = 0$ case (dash-dotted line) is constant while the dashed line ($r_s \neq 0$) approximates the flux components more accurately.

$$m = \begin{cases} \left\lceil \frac{N_p}{d} \right\rceil - 1 & N_p \geq 2d \\ 1 & N_p < 2d \end{cases}, \quad (18)$$

where $\lceil \cdot \rceil$ denotes the ceiling function.

IV. THE ANALYTICAL APPROACH (ANL)

The methods discussed so far are based on (iterative) extrapolation and interpolation. In this section we will introduce an analytical method to extend the output trajectories. We will simplify the continuous-time model of the machine given by (1a)-(1d) enabling us to express the outputs (i.e. the torque, the stator flux and the neutral point's potential) by polynomials of degree three or less and to solve for N_p analytically³.

A. Approximating the Stator Flux

In (1a) and (1b), the $-r_s \frac{x_{rr}}{D}$ terms account for the losses in the stator whereas the $r_s \frac{x_m}{D}$ terms are the positive contributions from the rotor flux to the stator flux. For medium-voltage induction machines, the losses in the stator are almost offset by the positive contributions from the rotor flux. Moreover, as r_s is typically very small, these terms become negligible. We shall consider two cases. (i) neglecting the effect of r_s (i.e. $r_s = 0$) and (ii) taking the effect of r_s into account.

1) *Stator flux trajectories when $r_s = 0$* : In this case the stator flux is only moved by the voltage vectors and the stator flux dynamics given by (1a) and (1b) are reduced to

$$\dot{\psi}_s = \begin{bmatrix} 0 & 0 \\ 0 & 0 \end{bmatrix} \psi_s + \begin{bmatrix} v_\alpha \\ v_\beta \end{bmatrix} \quad (19)$$

It follows that

³Polynomials of higher degree or other nonlinear functions will necessitate the use of a numerical root finding algorithm.

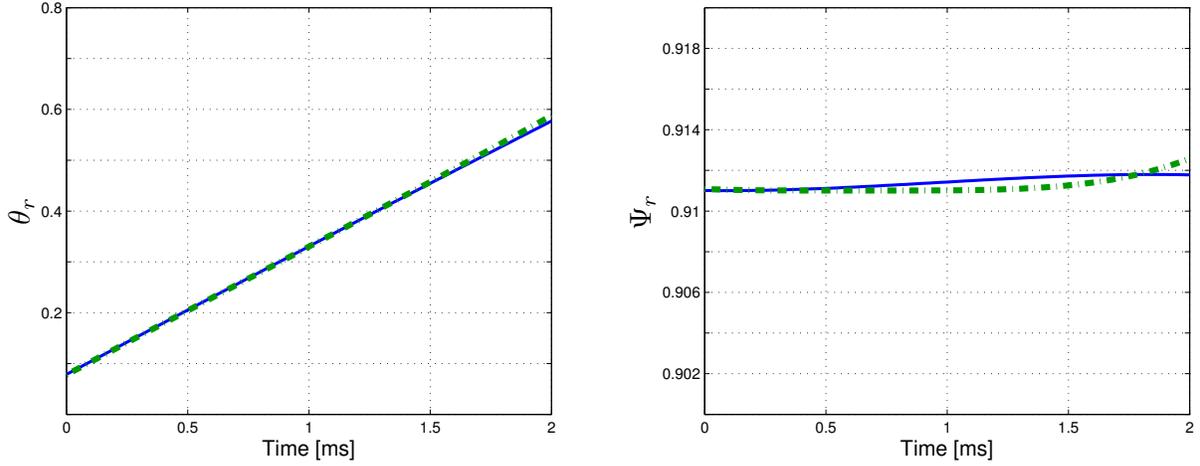


Figure 4: The actual (solid) and approximated (dash-dotted) angle and magnitude of the rotor flux over a typical prediction horizon.

$$\psi_s(t) = \psi_s(0) + t\mathbf{v}, \quad (20)$$

where $\psi_s(0)$ is the initial condition and t is the time.

2) *Taking the effect of r_s into account*: Including the effect of the stator resistance, r_s , allows us to predict the stator flux trajectories more accurately. For this purpose, we shall solve the system of linear differential equations expressed by (1a)-(1d), and consider only the most dominant terms in the solution that account for the stator losses. Then, by using first order Taylor expansions, those terms are approximated about a nominal time t_p yielding

$$\psi_{s\alpha}(t) = \alpha_1 t + \beta_1 \quad (21a)$$

$$\psi_{s\beta}(t) = \alpha_2 t + \beta_2, \quad (21b)$$

where α_i and β_i are constants. For expressions describing α_i and β_i and their derivation please refer to [15]. Fig. 3 depicts the effect of r_s on the approximated stator flux components.

B. Approximating the Rotor Flux

In (1c) and (1d), the ω_r terms correspond to the rotation of the rotor flux vector with constant magnitude, whereas the $-r_r \frac{x_{ss}}{D}$ terms account for the losses due to r_r that reduce the length of the rotor flux. The $r_r \frac{x_m}{D}$ terms are the positive contributions from the stator flux due to the mutual inductance x_m . The losses in the rotor flux are almost offset by the positive contribution from the stator flux. Moreover, r_r and consequently $-r_r \frac{x_{ss}}{D}$ and $r_r \frac{x_m}{D}$ are typically very small compared to ω_r . These observations allow us to omit the $-r_r \frac{x_{ss}}{D}$ and $r_r \frac{x_m}{D}$ terms, which yields

$$\dot{\psi}_r = \begin{bmatrix} 0 & -\omega_r \\ \omega_r & 0 \end{bmatrix} \psi_r \quad (22)$$

(22) is a system rotating with a fixed speed and a constant magnitude. The solution of (22) is

$$\psi_{r\alpha}(t) = \Psi_r \cos(\omega_r t + \Theta) \quad (23a)$$

$$\psi_{r\beta}(t) = \Psi_r \sin(\omega_r t + \Theta), \quad (23b)$$

where $\Psi_r = \sqrt{\psi_{r\alpha}^2(0) + \psi_{r\beta}^2(0)}$ and $\Theta = \arcsin\left(\frac{\psi_{r\beta}(0)}{\Psi_r}\right)$ are the length and the initial angle of the rotor flux, respectively. Fig. 4 illustrates that the rotor flux can be seen as a rotating vector with constant speed and length over the prediction horizon. We use second order Taylor expansions to approximate (23a) and (23b) around a nominal time t_p , to express the rotor flux components in polynomials of degree two.

$$\psi_{r\alpha}(t) = a_1 t^2 + b_1 t + c_1 \quad (24a)$$

$$\psi_{r\beta}(t) = a_2 t^2 + b_2 t + c_2 \quad (24b)$$

Please refer to [15] for explicit expressions of a_i , b_i and c_i .

C. Approximating the Outputs

Using the stator flux and the rotor flux polynomials, we can express the machine's outputs by polynomials of degree three or less, which allows us to solve for N_p analytically.

1) *The length of the stator flux*: Using (21a) and (21b) in (3), and squaring the result, the length of the stator flux (squared) is expressed as a polynomial of degree 2.

$$\begin{aligned} \Psi_s^2(t) &= (\alpha_1 t + \beta_1)^2 + (\alpha_2 t + \beta_2)^2 \\ &= (\alpha_1^2 + \beta_1^2) t^2 + 2(\alpha_1 \beta_1 + \alpha_2 \beta_2) t \\ &\quad + \beta_1^2 + \beta_2^2 \end{aligned} \quad (25)$$

2) *The torque*: The polynomial form of the torque is obtained by substituting (21) and (24) in (2).

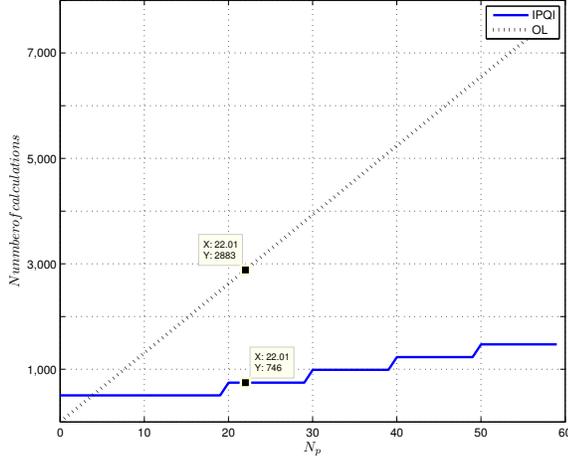


Figure 5: The comparison of the required number of calculations associated with the OL and IPQI methods for the switching horizon 'SE' and $d = 10$. The computational burden for the IPQI method increases stepwise for every d steps. The number of the calculations for $N_p = 22$ is illustrated, which is the most likely trajectory length for the switching horizon 'SE', at $\omega_e = 0.8\text{pu}$ and $T_e = 1\text{pu}$.

$$T_e(t) = \frac{x_m}{D} ((\alpha_2 a_1 - \alpha_1 a_2)t^3 + (\alpha_2 b_1 + \beta_2 a_1 - \alpha_1 b_2 - \beta_1 a_2)t^2 + (\alpha_2 c_1 + \beta_2 b_1 - \alpha_1 c_2 - \beta_1 b_2)t + \beta_2 c_1 - \beta_1 c_2) \quad (26)$$

3) *The inverter's neutral point potential:* The neutral point potential (4) can be expressed as

$$\dot{v}_n = \gamma_1 i_{s,\alpha} + \gamma_2 i_{s,\beta}, \quad (27)$$

where $[\gamma_1 \ \gamma_2 \ \gamma_3] = \frac{1}{2x_c} |\mathbf{u}_{abc}|^T \mathbf{P}^{-1}$ are constant coefficients. Substituting $i_{s,\alpha}$ and $i_{s,\beta}$ with (6), using (21) and (23) and integrating, (27) becomes

$$v_n(t) = v_n(0) - x_m \frac{\gamma_1}{D} \frac{\Psi_r}{\omega_r} (\sin(\omega_r t + \Theta) - \sin(\Theta)) + x_m \frac{\gamma_2}{D} \frac{\Psi_r}{\omega_r} (\cos(\omega_r t + \Theta) - \cos(\Theta)) + x_{rr} \frac{\gamma_1}{D} \left(\frac{\alpha_1}{2} t^2 + \beta_1 t \right) + x_{rr} \frac{\gamma_2}{D} \left(\frac{\alpha_2}{2} t^2 + \beta_2 t \right) \quad (28)$$

Substituting $\Psi_r \cos(\omega_r t + \Theta)$ and $\Psi_r \sin(\omega_r t + \Theta)$ with (24a) and (24b), (28) becomes a polynomial of degree two

$$v_n(t) = z_1 t^2 + z_2 t + z_3, \quad (29)$$

where the coefficients z_i are provided in the Appendix.

V. PERFORMANCE EVALUATION

The computational complexity associated with each method is given in Table I. Fig. 5 shows the required number of

calculations for the OL and IPQI methods as a function of the trajectory length N_p .

Based on simulation runs for 1.5 sec. using a 3-level voltage source inverter with a 2.5 MVA induction machine [10], the accuracy of the different extension methods is compared with the one of the open-loop simulation method. For this purpose, we define the relative trajectory length error as

$$N_p^{rel} = \frac{N_p^{ol} - N_p}{N_p^{ol}}, \quad (30)$$

where N_p and N_p^{ol} are the predicted trajectory lengths of the method under consideration and of the open-loop simulation method, respectively. The relative error histogram associated with each method is depicted in Fig. 6. Table II compares the closed-loop performance of the proposed methods.

The flattest histogram is associated with the LE method, as this approach yields the least accurate predictions. The difference between the histograms in Fig. 6.e and Fig. 6.f shows that the accuracy of the predictions associated with the analytical method can be improved by taking the effect of r_s into account. The QE method and the analytical method ($r_s \neq 0$) are of comparable accuracy. The probability density of the relative error for the IPQI method is highly concentrated about zero highlighting its accuracy.

The closed-loop simulation results suggest that for the short switching horizon 'SE', all methods exhibit performances similar to the one of the OL method. The state vector is needed in the switching step to build the output samples used for extending the output trajectories. For the OL method and the analytical method, the state vector is automatically obtained as a part of the extension step. For other methods, the state trajectories can be extended in the same way as the output trajectories. For the LE method, however, due to its low accuracy, the correspondence between the extended output and state vector is lost making this method not applicable for long switching horizons such as 'SESE'.

As can be seen for the open-loop simulation method, with respect to MPDTC with the switching horizon 'SE', MPDTC with the switching horizon 'SSESE' reduces the switching frequency by about 28%. At the same time, the THDs are reduced. The QE, the PQI and the IPQI methods exhibit almost the same performance as the open-loop simulation method as the switching horizon is increased. For the analytical method, with respect to MPDTC with the switching horizon 'SE', MPDTC with the switching horizon 'SSESE' reduces the switching frequency by almost 25%. This inferior performance improvement as compared to the performance of the PQI and IPQI methods, is partly due to the fact that the output equations are not approximated about the correct value of t_p .

VI. CONCLUSIONS

In this paper, several different methods to extend the output trajectories were proposed, and their performance was compared with respect to their accuracy and the associated computational cost. These methods include open-loop simulation (OL), linear extrapolation (LE), quadratic extrapolation

Table I: The number of basic operations associated with each extension method

extension method	OL	LE	QE	PQI	IPQI	ANL
solution approach	sequential	analytical	analytical	analytical	sequential	analytical
total number of calc.	$131N_p$	149	373	373	$\begin{cases} 242 \left\lfloor \frac{N_p}{d} \right\rfloor - 111 & N_p \geq 2d \\ 373 & N_p < 2d \end{cases}$	246

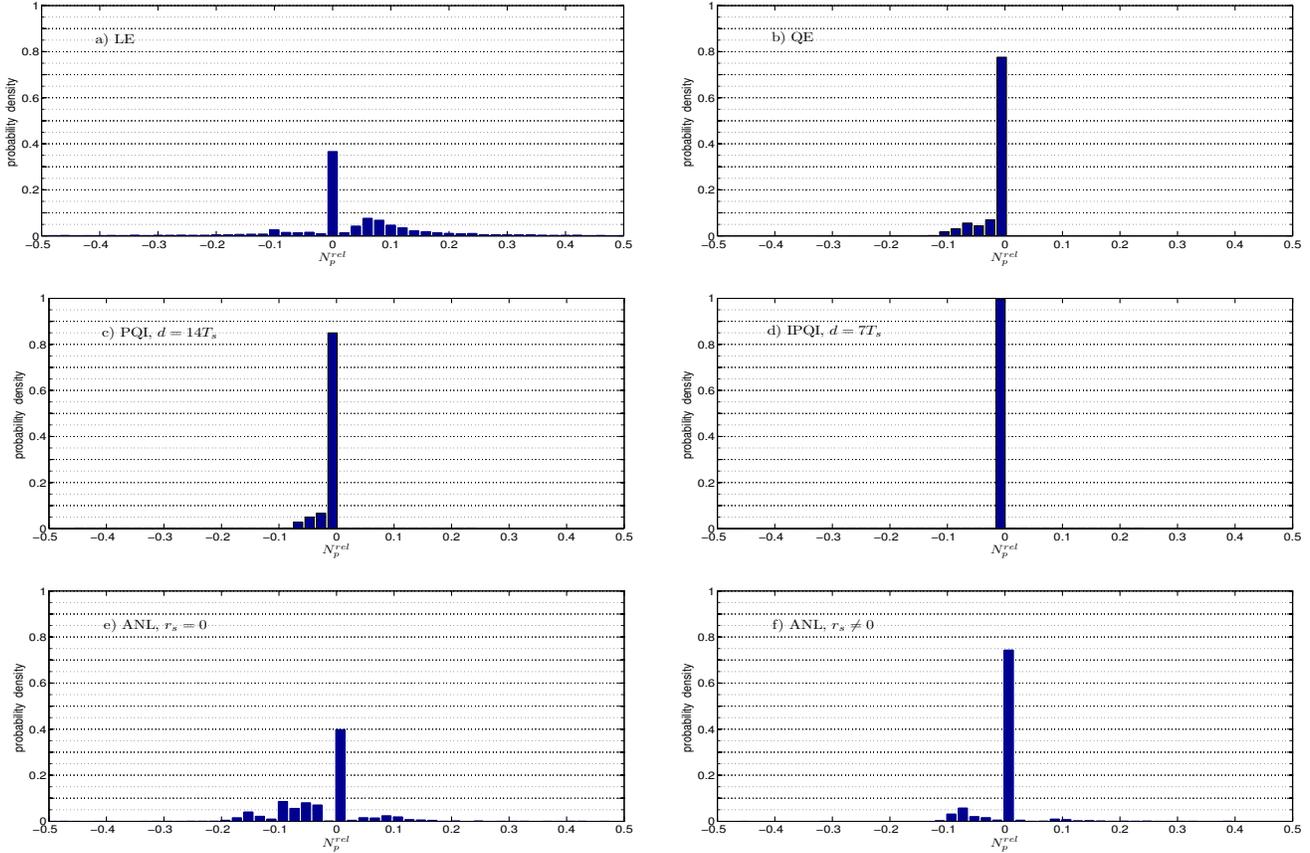


Figure 6: Histograms of the relative trajectory length errors obtained through simulation for the switching horizon 'SE', at $T_e = 1pu$ and $\omega_e = 0.6pu$, where T_e and ω_e are the electromagnetic torque and speed references, respectively.

Table II: Closed-loop simulation results. The table compares the average inverter switching frequency (Hz), the torque's total harmonic distortion (THD) and the stator current THD for different extension methods for the machine running at 100% torque and 60% speed.

Switching Horizon	Extension Method	OL	LE	QE	PQI	IPQI	ANL
					$d = 14$	$d = 7$	$t_p = 7T_s$
'SE'	Freq. (Hz)	199	204	201	199	199	201
	T_e, THD	6.60	6.71	6.76	6.61	6.61	6.61
	I_{THD}	8.15	8.35	8.38	8.17	8.17	8.25
'SESE'	Freq. (Hz)	154	N/A	156	154	153	157
	T_e, THD	6.24	N/A	6.23	6.24	6.20	5.76
	I_{THD}	7.65	N/A	7.65	7.66	7.66	7.24
'SSESE'	Freq. (Hz)	143	N/A	146	143	143	150
	T_e, THD	5.42	N/A	5.89	5.49	5.48	5.65
	I_{THD}	7.01	N/A	7.40	7.09	7.07	7.17

(QE), prediction with quadratic interpolation (PQI), iterative prediction with quadratic interpolation (IPQI) and an analytical approach (ANL).

Linear extrapolation is computationally simple and it performs fairly well for the switching horizon 'SE'. This method has been successfully implemented and tested on ABB's ACS6000 medium-voltage drive [2]. However, it is not applicable to long switching horizons and the predictions are inaccurate especially when the machine operates at high speed. If the machine equations are approximated about the correct time-instant t_p , the proposed analytical approach can achieve a performance comparable to the one of the QE method with slightly less computational effort. However, finding the proper value of t_p is not trivial as it depends on the operating point of the machine and on the switching horizon. With the same computational burden, the PQI method is more accurate than the QE method, and it outperforms the QE method in terms of the switching frequency and the total harmonic distortions for all switching horizons. The IPQI method achieves the best performance with respect to the open-loop simulation method even for very long switching horizons, but the associated computational burden for the average prediction horizon is almost twice as much as for the PQI method, as shown in Table I.

It was shown in [10] that long prediction horizons significantly improve the performance of MPDTC, but standard extrapolation methods such as linear or quadratic extrapolation tend to be too inaccurate to fully exploit the benefits of long horizons such as 'SESESE'. By using the new extension methods proposed in this paper, this issue can be solved. Together with the branch and bound method tailored to MPDTC in [12], which reduces the computational burden of long switching horizons by an order of magnitude, MPDTC is expected to be soon implemented by industry also with very long switching horizons. Moreover, these methods are equally applicable to the current control derivative of MPDTC, model predictive direct current control (MPDCC) [16], [17].

APPENDIX

The coefficients z_i in (29) are given by

$$\begin{aligned} z_1 &= \frac{\gamma_1}{D} \left(\frac{x_{rr}}{2} \alpha_1 - \frac{x_m}{\omega_r} a_2 \right) + \frac{\gamma_2}{D} \left(\frac{x_{rr}}{2} \alpha_2 + \frac{x_m}{\omega_r} a_1 \right) \quad (31) \\ z_2 &= \frac{\gamma_1}{D} \left(x_{rr} \beta_1 - \frac{x_m}{\omega_r} b_2 \right) + \frac{\gamma_2}{D} \left(x_{rr} \beta_2 + \frac{x_m}{\omega_r} b_1 \right) \\ z_3 &= v_n(0) - \frac{\gamma_1 x_m}{D \omega_r} (c_2 - \sin(\Theta)) + \frac{\gamma_2 x_m}{D \omega_r} (c_1 - \cos(\Theta)) \end{aligned}$$

REFERENCES

- [1] T. Geyer, G. Papafotiou, and M. Morari, "Model predictive direct torque control - part I: Concept, algorithm, and analysis," *IEEE Transactions on Industrial Electronics*, vol. 56, no. 6, pp. 1894–1905, Jun. 2009.
- [2] G. Papafotiou, J. Kley, K. Papadopoulos, P. Bohren, and M. Morari, "Model predictive direct torque control - part II: Implementation and experimental evaluation," *IEEE Transactions on Industrial Electronics*, vol. 56, no. 6, pp. 1906–1915, Jun. 2009.
- [3] T. Geyer, "Low complexity model predictive control in power electronics and power systems," Ph.D. dissertation, ETH Zurich, March 2005.

- [4] I. Takahashi and T. Noguchi, "A new quick-response and high-efficiency control strategy of an induction motor," *IEEE Transactions on Industry Applications*, vol. IA-22, no. 5, pp. 820–827, Sept. 1986.
- [5] G. Buja and M. Kazmierkowski, "Direct torque control of PWM inverter-fed AC motors - a survey," *IEEE Transactions on Industrial Electronics*, vol. 51, no. 4, pp. 744–757, Aug. 2004.
- [6] D. Q. Mayne and J. B. Rawlings, "Constrained model predictive control: stability and optimality," *Automatica*, vol. 36, no. 6, pp. 789–814, Jun. 2000.
- [7] C. E. Garcia, D. M. Prett, and M. Morari, "Model predictive control: theory and practice—a survey," *Automatica*, vol. 25, no. 3, pp. 335–348, May 1989.
- [8] S. J. Qin and T. A. Badgwell, "A survey of industrial model predictive control technology," *Control Engineering Practice*, vol. 11, no. 7, pp. 733–764, Jul. 2003.
- [9] D. Q. Mayne and J. B. Rawlings, *Model Predictive Control: Theory and Design*, Nob Hill Publishing, Aug. 2009.
- [10] T. Geyer, "Generalized model predictive direct torque control: Long prediction horizons and minimization of switching losses," in *Proceedings of the 48th IEEE Conference on Decision and Control*, Shanghai, China, Dec. 2009, pp. 6799–6804.
- [11] T. Geyer and G. Papafotiou, "Model predictive direct torque control of a variable speed drive with a five-level inverter," in *Proceedings of the 35th IEEE Conference on Industrial Electronics*, Porto, Portugal, Nov. 2009, pp. 1203–1208.
- [12] T. Geyer, "Computationally efficient model predictive direct torque control," in *Proceedings of the IEEE Energy Conversion Congress and Exposition*, Atlanta, GA, Sept. 2010, pp. 207–214.
- [13] A. Nabae, I. Takarashi and H. Akagi, "A new neutral-point clamped PWM inverter," *IEEE Transactions on Industry Applications*, vol. IA-17, no. 5, pp. 518–523, Sept./Oct. 1981.
- [14] P. Krause, O. Wasynczuk, and S. Sudhoff, *Analysis of Electric Machinery and Drive Systems*, Wiley-IEEE Press, Feb. 2002.
- [15] Y. Zeinaly, "Computationally efficient model predictive direct torque control," Master thesis, Chalmers University of Technology, Nov. 2010.
- [16] T. Geyer, "Model predictive direct current control for multi-level converters," in *Proceedings of the IEEE Energy Conversion Congress and Exposition*, Atlanta, GA, Sept. 2010, pp. 4305–4312.
- [17] J.C. Ramirez Martinez, R.M. Kennel, and T. Geyer, "Model predictive direct current control," in *Proceedings of IEEE International Conference on Industrial Technology*, Vina del Mar, Chile, 2010, pp. 1808–1813.