2016 European Control Conference
June 29 - July 1, 2016. Aalborg, Denmark

# High-Speed Direct Model Predictive Control for Power Electronics

Bartolomeo Stellato and Paul J. Goulart

*Abstract*— **Common approaches for direct model predictive control (MPC) for current reference tracking in power electronics suffer from the high computational complexity encountered when solving integer optimal control problems over long prediction horizons. We propose an efficient alternative method based on approximate dynamic programming, greatly reducing the computational burden and enabling sampling times under $25\,\mu$s. Our approach is based on the offline minimization of an infinite horizon cost function estimate which is then applied to the tail cost of the MPC problem. This allows us to reduce the controller horizon to a very small number of stages improving overall controller performance. Our proposed algorithm is validated on a variable speed drive system with a three-level voltage source converter.**

## I. Introduction

Among the control strategies adopted in power electronics, model predictive control (MPC) [1] has recently gained popularity due to its various advantages [2]. MPC has been shown to outperform traditional control methods because of its ease in handling time-domain constraint specifications and its applicability to general power systems topologies and operating conditions.

In power electronics, many conventional control strategies are based on proportional-plus-integral (PI) controllers providing continuous input signals to a modulator, which manages conversion to discrete switching positions. Direct MPC [3] instead combines the current control and modulation into a single computational problem, providing a powerful alternative to conventional PI controllers. With direct MPC, the manipulated variables are the switch positions, which lie in a discrete and finite set, giving rise to a switched system. Therefore, this approach does not require a modulator and is often referred to as *finite control set* MPC.

Since the manipulated variables are restricted to be integers, the optimization problem underlying direct MPC is provably $\mathcal{NP}$-hard [4]. Consequently, direct MPC rapidly becomes intractable for real-time applications as the horizon length is increased. Despite attempts to overcome the computational burden of these methods [5], the problem remains open to allow implementation of these algorithms on embedded systems.

A recent technique introduced in [6] and benchmarked in [7], reduces the computational burden of direct MPC when increasing the prediction horizon. In that work the optimization problem was formulated as an integer least-squares (ILS) problem and solved using a tailored branch-and-bound algorithm, described as *sphere decoding* [8]. Although this approach appears promising relative to previous work, the computation time required to perform the sphere decoding algorithm for long horizons (i.e. $N = 10$), is still far slower than the sampling time typically required, i.e. $T_s = 25\,\mu$s.

This paper introduces a different method to deal with the direct MPC problem. In contrast to common formulations [9] where the switching frequency is controlled indirectly via penalization of the input switches over the controller horizon, in this work the system dynamics are augmented to directly estimate the switching frequency and to penalize its deviations from the desired value. To address the computational issues of long prediction horizons, we formulate the tracking problem as a regulation one by augmenting the state dynamics and cast it in the framework of approximate dynamic programming (ADP) [10]. The infinite horizon tail cost is approximated using the approach in [11] and [12] by solving a semidefinite program (SDP) [13] offline. This enables us to shorten the controller horizon by applying the estimated tail cost to the last stage and to keep good control performance.

Simulation results show that with our method, even very short prediction horizons exhibit better performance than the approach in [7] and [6] with much longer planning horizons, while drastically reducing the computational burden. This approach has also been implemented in [14] on a small size FPGA running comfortably within $25\,\mu$s sampling time.

As a case study, our proposed approach is applied to the variable-speed drive system in [6] consisting of a three-level neutral point clamped voltage source inverter connected to a medium-voltage induction machine. The plant is modeled as a linear system with a switched three-phase input with equal switching steps for all phases.

The remainder of the paper is organized as follows. In Section II we describe the drive system case study and sketch the physical model. In Section III the direct MPC problem is derived by augmenting the state dynamics and approximating the infinite horizon tail cost using ADP. Finally, we show the simulation results on the derived model in Section IV and provide conclusions in Section V.

## II. Drive System Case Study

In this work we consider a variable speed drive system consisting of a three-level neutral point clamped (NPC) voltage source inverter driving a medium-voltage (MV) induction machine shown in Figure 1. The total dc-link voltage $V_{dc}$ is assumed to be constant and the neutral point potential $N$ fixed.

The input variables are chosen to be the switch positions in the three-phase legs of the inverter, i.e. $\boldsymbol{u}_{sw} = \begin{bmatrix} u_a & u_b & u_c \end{bmatrix}^\top$ with $u_a, u_b, u_c \in \{-1, 0, 1\}$. We define the model of the induction motor and the drive in terms of the stator currents $\boldsymbol{i}_s$ and the rotor fluxes $\boldsymbol{\psi}_r$ in the $\alpha\beta$ plane. The dynamics can be expressed as the following discrete-time linear time

This work was supported by the People Programme (Marie Curie Actions) of the European Unions Seventh Framework Programme (FP7/2007-2013) under REA grant agreement no 607957 (TEMPO).

B. Stellato and P. J. Goulart are with the University of Oxford, Parks Road, Oxford, OX1 3PJ, U.K. {bartolomeo.stellato, paul.goulart}@eng.ox.ac.uk

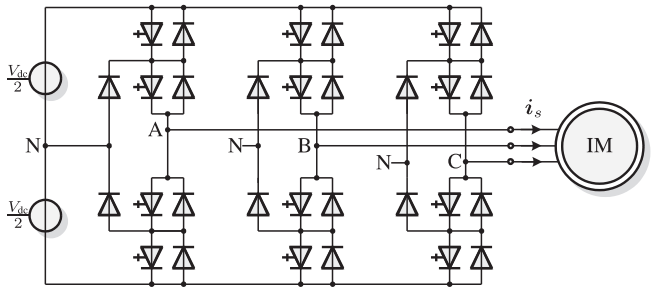



Fig. 1. Three-level three-phase neutral point clamped (NPC) voltage source inverter driving an induction motor with a fixed neutral point potential. Image taken from [6].

invariant (LTI) system

$$\begin{aligned} \boldsymbol{x}_{ph}(k+1) &= \boldsymbol{A}_{ph}\boldsymbol{x}_{ph}(k) + \boldsymbol{B}_{ph}\boldsymbol{u}_{sw}(k) \\ \boldsymbol{y}_{ph}(k) &= \boldsymbol{C}_{ph}\boldsymbol{x}_{ph}(k), \end{aligned} \tag{1}$$

where the state vector is $\boldsymbol{x}_{ph} = \begin{bmatrix} i_{s,\alpha} & i_{s,\beta} & \psi_{r,\alpha} & \psi_{r,\beta} \end{bmatrix}$ and the output vector corresponds to the stator currents, i.e. $\boldsymbol{y}_{ph} = \boldsymbol{i}_s$. The sampling time is $T_s = 25\,\mu$s. See [6] for a detailed derivation.

## III. Model Predictive Current Control

### A. Problem Description

Our control scheme must address two conflicting objectives simultaneously. On one hand, the distortion of the stator currents $\boldsymbol{i}_s$ corresponds to ripples in the torque of the motor that are the main source of mechanical stress on the load and the bearings. In order to reduce damage to the machine and the load, the distortion of stator currents must be kept as low as possible. On the other hand, high frequency switching of the inputs $\boldsymbol{u}_{sw}$ produces high power losses and stress on the physical devices. To reduce the energy needed and to preserve the lifespan of the components, we therefore should minimize the switching frequency of the integer inputs. In power converters, there is an unavoidable tradeoff between these two criteria.

In contrast to the common approaches in direct MPC where the switching frequency is minimized, in this work we penalize its difference from the desired frequency, denoted $f^*_{sw}$. This is motivated by the fact that drives are usually designed to run at a specific nominal switching frequency and the control algorithms are typically tuned to match this requirement.

The current distortion is measured via the total harmonic distortion (THD). Given an infinitely long time-domain current signal $i$ and its fundamental component $i^*$ of magnitude 1, we define the THD as the ratio between the root mean square (RMS) of their difference and the RMS of $i^*$. Since $i^*$ is sinusoidal, its RMS value is $1/\sqrt{2}$. Hence, we can write the THD as

$$THD \coloneqq \sqrt{2}\mathrm{RMS}(i - i^*). \tag{2}$$

It is of course not possible to calculate the THD in real time within our controller computations because of finite storage constraints.

The switching frequency of the inverter can be identified by computing the average frequency of each active semiconductor device. As displayed in Figure 1, the total number of switches for all three phases is 12, and for every transition of each phase $u_a, u_b, u_c$ between different adjacent switch positions one physical device is turned on. Given $2M \in \mathbb{N}$ time steps, it is possible to estimate the switching frequency of every device by counting the number of *on* transitions over the time interval and dividing the sum by the interval's length. We can then average over all the physical switches by dividing the computed fraction by 12. At time $k$, the switching frequency estimate can be written as

$$f_{sw,M}(k) \coloneqq \frac{1}{12 \cdot 2M \cdot T_s} \sum_{i=-M}^{M} \|\boldsymbol{u}_{sw}(k+i) - \boldsymbol{u}_{sw}(k+i-1)\|_1, \tag{3}$$

that corresponds to a non-causal finite impulse response (FIR) filter of order $2M$. The true average switching frequency is the limit of this quantity as the window length goes to infinity

$$f_{sw} \coloneqq \lim_{M\to\infty} f_{sw,M}(k), \tag{4}$$

and does not depend on time $k$.

The $f_{sw}$ computation brings issues similar to those of the THD. In addition to finite storage constraints, the part of the sum regarding the future signals produces a non-causal filter that is impossible to implement in a real time control scheme.

These issues in computing THD and $f_{sw}$ will be addressed in the following two sections via augmentation of our state space model to include suitable approximation schemes for both quantities.

### B. Total Harmonic Distortion

By definition, the THD can be minimized by reducing the effect of the ripples in the produced currents.

The three reference phase sinusoids $\boldsymbol{i}^*_s$ are defined in the normalized per unit (pu) time scale and have shifted phases (in seconds, these signals oscillate at frequency $\omega_b$)

$$\boldsymbol{i}^*_s(k) = \begin{bmatrix} \sin(k) & \sin\left(k - \frac{2}{3}\pi\right) & \sin\left(k - \frac{4}{3}\pi\right) \end{bmatrix}^\top.$$

It is straightforward to show (see [15]) that the corresponding reference currents in the $\alpha\beta$ reference frame are

$$\boldsymbol{i}^*_{s,\alpha\beta}(k) = \begin{bmatrix} \sin(k) & -\cos(k) \end{bmatrix}^\top. \tag{5}$$

From the stator currents in $\alpha\beta$ frame, the ripples can be obtained by subtracting the perfect sinusoids that need to be tracked, producing the error signal $\boldsymbol{e_i}(k) \coloneqq \boldsymbol{i}_{s,\alpha\beta}(k) - \boldsymbol{i}^*_{s,\alpha\beta}(k)$. In order to minimize a reasonable approximation of the THD defined in (2) for both dimensions $\alpha\beta$, in our controller we minimize the squared 2-norm of $\boldsymbol{e_i}$ for all future time instants. We also introduce a discount factor $\gamma \in (0, 1)$ to normalize the summation preventing it from going to infinity in case of persistent error. The cost function related to THD minimization is, therefore,

$$\sum_{k=0}^{\infty} \gamma^k \left\|\boldsymbol{e_i}(k)\right\|_2^2. \tag{6}$$

In order to construct a regulation problem, we will include the oscillating currents from (5) as two additional uncontrollable states $\boldsymbol{x}_{osc} = \boldsymbol{i}^*_{s,\alpha\beta}$ within our model of the system

dynamics. The ripple signal $\boldsymbol{e_i}(k)$ is then modeled as an output defined by the difference between two pairs of system states.

### C. Switching Frequency

To overcome the difficulty of dealing with the filter in (3), we consider only the past input sequence, with negative time shift giving a causal FIR filter estimating $f_{sw}$. This filter is approximated with an infinite impulse response (IIR) one whose dynamics can be modeled as a linear time invariant (LTI) system. Note that future input sequences in (3) will be taken into account inside the controller prediction.

Let us define three binary phase inputs denoting whether each phase switching position changed at time $k$ or not,

$$\boldsymbol{p}(k) = \begin{bmatrix} p_a(k) & p_b(k) & p_c(k) \end{bmatrix}^\top \in \{0,1\}^3, \tag{7}$$

with $p_s(k) = \|u_s(k) - u_s(k-1)\|_1$, $s \in \{a,b,c\}$. It is straightforward to show that the following second order IIR filter will approximate the one-sided version of the FIR filter in (3) [16]:

$$\begin{aligned} \boldsymbol{x}_{flt}(k+1) &= \begin{bmatrix} a_1 & 0 \\ 1-a_1 & a_2 \end{bmatrix} \boldsymbol{x}_{flt}(k) + \frac{1-a_2}{12T_s} \begin{bmatrix} 1 & 1 & 1 \\ 0 & 0 & 0 \end{bmatrix} \boldsymbol{p}(k) \\ \hat{f}_{sw}(k) &= \begin{bmatrix} 0 & 1 \end{bmatrix} \boldsymbol{x}_{flt}(k), \end{aligned}$$

where $\hat{f}_{sw}(k)$ is the estimated switching frequency. The two poles in $a_1 = 1 - 1/r_1$ and $a_2 = 1 - 1/r_2$ with $r_1, r_2 >> 0$ can be tuned to shape the behavior of the filter. By increasing $a_1, a_2$, the estimate becomes smoother while decreasing $a_1, a_2$ gives a faster estimation with more noisy values. We denote the difference between the approximation $\hat{f}_{sw}(k)$ and the target frequency $f^*_{sw}$ by $e_{sw}(k) := \hat{f}_{sw}(k) - f^*(k)$.

Therefore, the quantity to be minimized in order to bring the switching frequency estimate as close to the target as possible is

$$\sum_{k=0}^{\infty} \delta \cdot \gamma^k \left\| e_{sw}(k) \right\|_2^2, \tag{8}$$

where $\delta \in \mathbb{R}_+$ is a design parameter included to reflect the relative importance of this part of the cost relative to the THD component.

Finally, we can augment the state space to include the filter dynamics and the target frequency by adding the states $\begin{bmatrix} \boldsymbol{x}_{flt}^\top & f^*_{sw} \end{bmatrix}^\top$ so that the control inputs try to drive the difference between two states to zero.

### D. MPC Problem Formulation

Let us define the complete augmented state as

$$\boldsymbol{x}(k) := \begin{bmatrix} \boldsymbol{x}_{ph}(k)^\top & \boldsymbol{x}_{osc}(k)^\top & \boldsymbol{x}_{sw}(k)^\top & \boldsymbol{u}_{sw}(k-1)^\top \end{bmatrix}^\top, \tag{9}$$

with $\boldsymbol{x}(k) \in \mathbb{R}^9 \times \{-1,0,1\}^3$ and total state dimension $n_x = 12$. Vector $\boldsymbol{x}_{ph}$ represents the physical system from Section II, $\boldsymbol{x}_{osc}$ defines the oscillating states of the sinusoids to track introduced in Section III-B, $\boldsymbol{u}_{sw}(k-1)$ are additional states used to keep track of the physical switching positions at the previous stage and $\boldsymbol{x}_{sw}$ the states related to the switching filter from Section III-C.

The system inputs are defined as

$$\boldsymbol{u}(k) := \begin{bmatrix} \boldsymbol{u}_{sw}(k)^\top & \boldsymbol{p}(k)^\top \end{bmatrix}^\top \in \mathbb{R}^{n_u},$$

where $\boldsymbol{u}_{sw}$ are the physical switches positions and $\boldsymbol{p}$ are the three binary inputs entering in the frequency filter from Section III-C. The input dimension is $n_u = 6$. To simplify the notation, let us define the matrices $\boldsymbol{G}$ and $\boldsymbol{T}$ to obtain $\boldsymbol{u}_{sw}(k)$ and $\boldsymbol{p}(k)$ from $\boldsymbol{u}(k)$ respectively: i.e. $\boldsymbol{u}_{sw}(k) = \boldsymbol{G}\boldsymbol{u}(k)$ and $\boldsymbol{p}(k) = \boldsymbol{T}\boldsymbol{u}(k)$. Similarly, to obtain $\boldsymbol{u}_{sw}(k-1)$ from $\boldsymbol{x}(k)$ we define matrix $\boldsymbol{W}$ so that $\boldsymbol{u}_{sw}(k-1) = \boldsymbol{W}\boldsymbol{x}(k)$.

The MPC problem with horizon $N \in \mathbb{N}$ can be written as

$$\begin{aligned} \underset{\boldsymbol{u}(k)}{\text{minimize}} \quad & \sum_{k=0}^{N-1} \gamma^k \ell(\boldsymbol{x}(k)) + \gamma^N V(\boldsymbol{x}(N)) && \text{(10a)} \\ \text{subject to} \quad & \boldsymbol{x}(k+1) = \boldsymbol{A}\boldsymbol{x}(k) + \boldsymbol{B}\boldsymbol{u}(k) && \text{(10b)} \\ & \boldsymbol{x}(0) = \boldsymbol{x}_0 && \text{(10c)} \\ & \boldsymbol{x}(k) \in \mathcal{X},\ \boldsymbol{u}(k) \in \mathcal{U}(\boldsymbol{x}_0), && \text{(10d)} \end{aligned}$$

where the stage cost is defined combining the THD and the switching frequency penalties in (6) and (8) respectively

$$\ell(\boldsymbol{x}(k)) = \left\| \boldsymbol{C}\boldsymbol{x}(k) \right\|_2^2 = \left\| \boldsymbol{e}(k) \right\|_2^2 + \delta \left\| e_{sw}(k) \right\|_2^2.$$

The tail cost $V(\boldsymbol{x}(N))$ is an approximation of the infinite horizon tail that we will compute in the next section using approximate dynamic programming (ADP). The matrices $\boldsymbol{A}$, $\boldsymbol{B}$ and $\boldsymbol{C}$ define the extended system dynamics and the output vector; they can be derived directly from the physical model (1) and from the considerations in Sections III-C and III-B.

The input constraint set is denoted as

$$\begin{aligned} \mathcal{U}(\boldsymbol{x}_0) = \{ & -\boldsymbol{T}\boldsymbol{u}(k) \le \boldsymbol{u}(k) - \boldsymbol{W}\boldsymbol{x}(k) \le \boldsymbol{T}\boldsymbol{u}(k), && \text{(11a)} \\ & \left\| \boldsymbol{T}\boldsymbol{u}(k) \right\|_\infty \le 1, && \text{(11b)} \\ & \boldsymbol{G}\boldsymbol{u}(k) \in \{-1,0,1\}^3 \}, && \text{(11c)} \end{aligned}$$

where constraint (11a) defines the relationship between $\boldsymbol{u}_{sw}$ and $\boldsymbol{p}$ from (7). Constraint (11b) together with (11a) defines the switching constraints $\left\| \boldsymbol{u}_{sw}(k) - \boldsymbol{u}_{sw}(k-1) \right\|_\infty \le 1$ imposed to avoid a shoot through in the inverter positions that could damage the components. Finally, (11c) enforces integrality of the switching positions.

Observe that the controller tuning parameters are $\delta$, which defines the relative importance of the THD and $f_{sw}$ components in the cost function, and $r_1, r_2$ that shape the switching frequency estimator.

Following a receding horizon control strategy, at each stage $k$ problem (10) is solved, obtaining the optimal sequence of $\boldsymbol{u}(k)$ from which only the first input $\boldsymbol{u}(0)$ is applied to the switches. At the next stage $k+1$, given new information on $\boldsymbol{x}_0$, a new optimization problem is then solved providing an updated optimal switching sequence, and so on.

By exploiting the highly parallelizable structure of the integer problem (10), this formulation has been solved on a low size FPGA for short horizons of $N = 1$, $N = 2$ while keeping the execution time within the sampling time $T_s = 25\,\mu$s; see [14].

### E. Approximate Dynamic Programming

The goal of this section is to compute a value function approximation $V^{adp}$ for an infinite horizon version of (10). The function $V^{adp}$ is used as a tail cost in (10).

Let $V^*(\boldsymbol{z})$ be the value function evaluated at $\boldsymbol{z}$, i.e. the optimal value of the objective of our infinite horizon control

problem starting at state $\boldsymbol{x} = \boldsymbol{z}$ subject to the system dynamics (10b) and initial state constraint $\boldsymbol{x}(0) = \boldsymbol{z}$. The main idea behind dynamic programming is that the function $V^*$ is the unique solution of the equation

$$V^*(\boldsymbol{z}) = \min_{\boldsymbol{u}\in\mathcal{U}(\boldsymbol{z})} \{l(\boldsymbol{z},\boldsymbol{u}) + \gamma V^*(\boldsymbol{A}\boldsymbol{z} + \boldsymbol{B}\boldsymbol{u})\} \quad \forall \boldsymbol{z},$$

known as the Bellman equation. The right-hand side can be written as monotonic operator on $V^*$, usually referred to as the Bellman operator: $V^* = \mathcal{T}V^*$. Unfortunately, solutions to the Bellman equation can only be solved analytically in a limited number of special cases; e.g. when the state and inputs have small dimensions or when the system is linear, unconstrained and the cost function is quadratic [17]. For more complicated problems, dynamic programming is limited by the so-called curse of dimensionality; storage and computation requirements tend to grow exponentially with the problem dimensions. Because of the integer switches in the power converter analyzed in this work, it is intractable to compute the optimal infinite horizon cost and policy and, hence, systematic methods for approximating the optimal value function offline are needed.

Approximate dynamic programming [10] consists of various techniques for estimating $V^*$. The approach developed in [11] and [12] relaxes the Bellman equation into an inequality

$$V^{adp}(\boldsymbol{z}) \leq \min_{\boldsymbol{u}\in\mathcal{U}(\boldsymbol{z})} \left\{l(\boldsymbol{z},\boldsymbol{u}) + \gamma V^{adp}(\boldsymbol{A}\boldsymbol{z} + \boldsymbol{B}\boldsymbol{u})\right\}, \quad \forall \boldsymbol{z}, \tag{12}$$

or, equivalently, using the Bellman operator: $V^{adp} \leq \mathcal{T}V^{adp}$.

The set of functions $V^{adp}$ that satisfy the Bellman inequality are underestimators of the optimal value function $V^*$; see [12]. The Bellman inequality is therefore a sufficient condition for underestimation of $V^*$. In [12] the authors show that by iterating inequality (12), the conservatism of the approximation can be reduced. The iterated Bellman inequality is defined as $V^{adp} \leq \mathcal{T}^M V^{adp}$, where $M > 1$ is an integer defining the number of iterations, or equivalently, from [12], as $V_{i-1}^{adp} \leq \mathcal{T}V_i^{adp}, \quad i = 1,\dots,M$, where $V_i^{adp}$ are the iterates of the value function.

To make the problem tractable, we restrict the iterates to the finite-dimensional subspace spanned by the basis functions $V^{(j)}$ defined in [11], [12]

$$V_i^{adp} = \sum_{j=1}^{K} \alpha_{ij} V^{(j)}, \quad i = 0,\dots,M-1. \tag{13}$$

The coefficients $\alpha_i$ are computed by solving a Semidefinite Program (SDP) [13]. The rewritten iterated Bellman inequality suggests the following optimization problem for finding the best underestimator for the value function $V^*$

$$\text{maximize} \quad \int_{\mathcal{X}} V^{adp}(\boldsymbol{z}) c(\mathrm{d}\boldsymbol{z}) \tag{14a}$$

$$\text{subject to } V_{i-1}^{adp}(\boldsymbol{z}) \leq \min_{\boldsymbol{u}\in\mathcal{U}(\boldsymbol{z})} \left\{l(\boldsymbol{z},\boldsymbol{u}) + \gamma V_i^{adp}(\boldsymbol{A}\boldsymbol{z} + \boldsymbol{B}\boldsymbol{u})\right\} \tag{14b}$$

$$\forall \boldsymbol{z} \in \mathbb{R}^6 \times \{-1,0,1\}, \quad i = 1,\dots,M, \tag{14c}$$

$$V_0^{adp} = V_M^{adp} = V^{adp}, \tag{14d}$$

where $c(\cdot)$ is a non-negative measure over the state space that can be viewed as a distribution giving more importance to regions where we would like a better approximation. On the chosen subspace (13), the inequality (14b) is convex in the coefficients $\alpha_{ij}$ [12].

Following the approach in [12], we make use of quadratic candidate functions of the form

$$V_i^{adp}(\boldsymbol{z}) = \boldsymbol{z}^\top \boldsymbol{P}_i \boldsymbol{z} + 2\boldsymbol{q}_i^\top \boldsymbol{z} + r_i, \quad i = 0,\dots,M, \tag{15}$$

where $\boldsymbol{P}_i \in \mathbb{S}^{n_x}$, $\boldsymbol{q}_i \in \mathbb{R}^{n_x}$, $r_i \in \mathbb{R}$, $i = 0,\dots,M$.

If we denote $\boldsymbol{\mu}_c \in \mathbb{R}^{n_x}$ and $\boldsymbol{\Sigma}_c \in \mathbb{S}_+^{n_x}$ as the mean and the covariance matrix of measure $c(\cdot)$ respectively, by using candidate functions as in (15) the cost function of problem (14) becomes

$$\int_{\mathcal{X}} V^{adp}(\boldsymbol{z}) c(\mathrm{d}\boldsymbol{z}) = \mathrm{Tr}\left(\boldsymbol{P}_0 \boldsymbol{\Sigma}_c\right) + 2\boldsymbol{q}_0^\top \boldsymbol{\mu}_c + r_0.$$

We now focus on rewriting the constraint (14b) as a Linear Matrix Inequality (LMI) [18]. We first remove the $\min$ in the right-hand side by imposing the constraint for every admissible $\boldsymbol{u} \in \mathcal{U}(\boldsymbol{x}_0)$ and obtain

$$\begin{aligned} &V_{i-1}^{adp}(\boldsymbol{z}) \leq l(\boldsymbol{z},\boldsymbol{u}) + \gamma V_i^{adp}(\boldsymbol{A}\boldsymbol{z} + \boldsymbol{B}\boldsymbol{u}), \\ &\forall \boldsymbol{z} \in \mathbb{R}^6 \times \{-1,0,1\},\ \forall \boldsymbol{u} \in \mathcal{U}(\boldsymbol{z}),\ i = 1,\dots,M. \end{aligned} \tag{16}$$

From [12], we can rewrite (16) as a quadratic form

$$\begin{aligned} &\begin{bmatrix} \boldsymbol{z} \\ 1 \end{bmatrix}^\top \boldsymbol{M}_i(\boldsymbol{u}) \begin{bmatrix} \boldsymbol{z} \\ 1 \end{bmatrix} \geq 0,\ \forall \boldsymbol{z} \in \mathbb{R}^6 \times \{-1,0,1\}, \\ &\forall \boldsymbol{u} \in \mathcal{U}(\boldsymbol{z}),\ i = 1,\dots,M, \end{aligned} \tag{17}$$

where $\boldsymbol{M}_i(\boldsymbol{u}) = \boldsymbol{L} + \gamma \boldsymbol{G}_i(\boldsymbol{u}) - \boldsymbol{S}_{i-1} \in \mathbb{S}^{n_x}$. The matrices $\boldsymbol{S}_{i-1}, \boldsymbol{L}$ and $\boldsymbol{G}_i(\boldsymbol{u})$ are derived in [14] according to [12].

By noting that the state vector $\boldsymbol{z}$ includes two parts which can take only a finite set of values — the desired frequency $f_{sw}^*$ and the previous physical input $\boldsymbol{u}_{sw}(k-1) \in \{-1,0,1\}$ — we can explicitly enumerate part of the state-space and rewrite the quadratic form (17) more compactly as

$$\begin{bmatrix} \tilde{\boldsymbol{z}} \\ 1 \end{bmatrix}^\top \tilde{\boldsymbol{M}}_i(\boldsymbol{m}) \begin{bmatrix} \tilde{\boldsymbol{z}} \\ 1 \end{bmatrix} \geq 0,\ \forall \tilde{\boldsymbol{z}} \in \mathbb{R}^8,\ \forall \boldsymbol{m} \in \mathcal{M},\ i = 1,\dots,M, \tag{18}$$

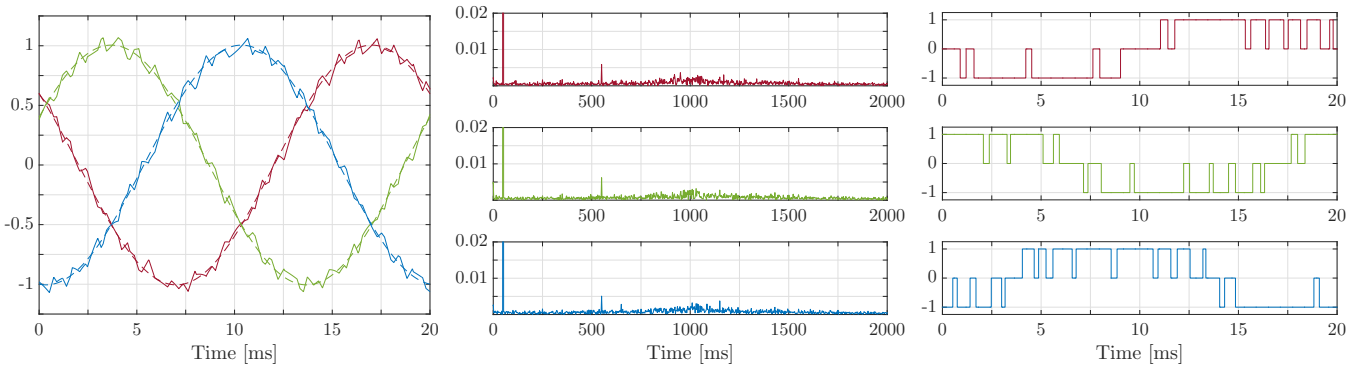
where $\tilde{\boldsymbol{z}}$ is the state vector without the desired frequency and $\boldsymbol{u}_{sw}(k-1)$. Moreover, $\boldsymbol{m} := (\boldsymbol{u}_{sw}, \boldsymbol{u}_{sw,pr}) \in \mathcal{M}$ are all the possible combinations of current and previous physical inputs satisfying the switching and integrality constraints (11). The detailed derivation of $\tilde{\boldsymbol{M}}(\boldsymbol{m}) \in \mathbb{S}^9$ can be found in [14].

Using the non-negativity condition of quadratic forms [13], it is easy to see that (18) holds if and only if $\tilde{\boldsymbol{M}}_i(\boldsymbol{m})$ is positive semidefinite. Hence, problem (14) can finally be rewritten as the following SDP

$$\begin{aligned} \text{maximize} \quad & \mathrm{Tr}\left(\boldsymbol{P}_0 \boldsymbol{\Sigma}_c\right) + 2\boldsymbol{q}_0^\top \boldsymbol{\mu}_c + r_0 \\ \text{subject to} \quad & \tilde{\boldsymbol{M}}_i(\boldsymbol{m}) \succeq 0, \quad \forall \boldsymbol{m} \in \mathcal{M}, \quad i = 1,\dots,M \\ & V_0^{adp} = V_M^{adp} \\ & \boldsymbol{P}_i \in \mathbb{S}^{n_x},\ \boldsymbol{q}_i \in \mathbb{R}^{n_x},\ r_i \in \mathbb{R},\ i = 0,\dots,M, \end{aligned} \tag{19}$$

which can be solved efficiently using a standard SDP solver, e.g. [19]. Once we obtain the solution to (19), we can define the infinite horizon tail cost to be used in problem (10) as

$$V^{adp}(\boldsymbol{z}) = \boldsymbol{z}^\top \boldsymbol{P}_0 \boldsymbol{z} + 2\boldsymbol{q}_0^\top \boldsymbol{z} + r_0. \tag{20}$$




(a) Three-phase stator currents (solid lines) with their references (dashed lines).

(b) Stator current spectrum.

(c) Three-phase switching position inputs.

Fig. 2. Simulated waveforms produced by the direct model predictive controller at steady state operation, at full speed and rated torque. Horizon of $N = 1$ is used. The switching frequency is approximately 300 Hz and the current THD is 5.24 %.

## IV. Simulation Results

We run simulations on the model from [6] of a neutral point clamped voltage source inverter connected with a medium-voltage induction machine and a constant mechanical load. The simulations are run in MATLAB, using an idealized model with the semiconductors switching instantaneously. This is motivated by the fact that, using a similar model, previous simulations [20] showed a very close match with the experimental results in [21]. The infinite horizon estimation SDP (19) is formulated in MATLAB using YALMIP [22] and solved offline using MOSEK [19]. The resulting integer optimization problems are solved using Gurobi Optimizer [23].

### A. Steady State Performance

The steady-state performance of the system with the proposed method is shown in Figure 2. The discount factor is chosen as $\gamma = 0.95$, the switching frequency filter parameters as $r_1 = r_2 = 800$ in order to get a smooth estimate. The desired switching frequency is set to 300 Hz and the weighting $\delta$ is chosen accordingly. The simulation results with $N = 1$ are shown in Figure 2 in the (pu) system.

For comparison, we simulate the drive system also with the controller discussed in [6] denoted as DMPC tuned in order to have the same switching frequency by adjusting the weighting parameter $\lambda_u$.

TABLE I
Simulation Results with Approximate Dynamic Programming Approach and with DMPC from [7]

| | ADP | | DMPC [7] | |
|---|---|---|---|---|
| | $\delta$ | THD [%] | $\lambda_u$ | THD [%] |
| $N = 1$ | 4 | 5.24 | 0.00235 | 5.44 |
| $N = 2$ | 5.1 | 5.13 | 0.00690 | 5.43 |
| $N = 3$ | 5.5 | 5.10 | 0.01350 | 5.39 |
| $N = 10$ | 10 | 4.80 | 0.10200 | 5.29 |

Numerical results with both approaches are presented in Table I. Our method, with a horizon of $N = 1$ provides at the same time a THD improvement over the direct MPC formulation in [6] with $N = 10$ and a drastically better numerical speed. Moreover, we also perform a comparison with longer horizons $N = 2$, $N = 3$ and $N = 10$. The results show that, if our method were applied on the drive system with horizon $N = 3$, it would show a clear improvement over the DMPC technique leading to a THD of 5.10 %. Horizon $N = 10$ would give an even greater reduction in THD until 4.80 %. Furthermore, it is important to underline that the method we propose is the *only* method available exhibiting this performance that can produce integer optimal solutions to this problem in 25 $\mu$s sampling time.

### B. Performance During Transients

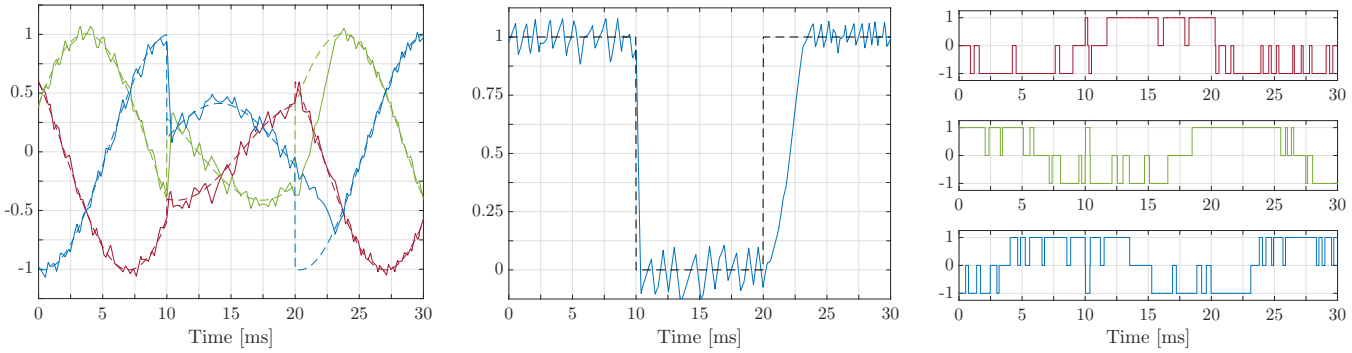
One of the main advantages of direct MPC is the fast transient response [6]. We simulated the system with the same tuning parameters as in the steady state benchmarks. At nominal speed, reference torque steps are imposed, see Figure 3b. These steps are translated into different current references to track, as shown in Figure 3a, while the computed inputs are shown in 3c.

The torque step from 1 to 0 presents an extremely short settling time of 0.35 ms similar to deadbeat control approaches [24]. This is achieved by inverting the voltage applied to the load. Since we prohibited switchings between $-1$ and 1 in (11a) and (11b), the voltage inversion is performed in $2T_s$ via an intermediate zero switching position.

Switching from 0 to 1 torque produces much slower response time of approximately 3.5 ms. This is due to the limited available voltage in the three-phase admissible switching positions. As shown in Figure 3c, during the second switching at time 20 ms, the last two phases inputs saturate to values $+1$ and $-1$ respectively for the majority of the transient providing the maximum available voltage that could steer the currents to the desired direction.

We simulate the transient responses also for horizon $N = 10$ obtaining nearly identical settling times. This is because the benefit of longer prediction is reduced by the saturation of the inputs during the transients.

These transient results match the simulations of DMPC method from [6] in terms of settling time. This shows that our method does not slow down the fast dynamical behavior during transients typical of direct current MPC.




(a) Three-phase stator currents (solid lines) with their references (dashed lines).

(b) Simulated (solid line) and reference (dashed line) torque.

(c) Three-phase switching position inputs.

Fig. 3. Reference torque steps produced by the direct model predictive controller with horizon $N = 1$.

Finally, the proposed approach shows far superior performance than offline methods for open-loop switch sequence computations like Optimized Pulse Patterns (OPPs) [25].

## V. Conclusion and Future Work

This work proposes a new computationally efficient direct model predictive control (MPC) scheme for current reference tracking in power converters. We extended problem formulation in [6] and [7] in order to include a direct switching frequency estimator in the system state. To reduce the horizon length and decrease the computational burden while preserving good control performances, we estimated the infinite horizon tail cost of the MPC problem formulation using approximate dynamic programming (ADP).

Steady state simulation results show that with our method requiring short horizons and extremely inexpensive computations, it is possible to obtain better performance than the direct MPC formulation in [6] with long horizons. This is due to the predictive behavior of the tail cost function obtained with ADP. We also performed transient simulations where the proposed approach exhibits the very fast dynamic response typical of the direct MPC [6].

This method has also been implemented in [14] on a small size FPGA running comfortably within $25\,\mu$s.

## Acknowledgment

The authors would like to thank Tobias Geyer of ABB Corporate Research, Baden-Dättwil, Switzerland, for his continuous advice and for the fruitful suggestions.

## References

[1] J. B. Rawlings and D. Q. Mayne, *Model Predictive Control: Theory and Design* . Nob Hill Publishing, LLC, 2014.

[2] P. Cortes, M. P. Kazmierkowski, R. M. Kennel, D. E. Quevedo, and J. Rodriguez, "Predictive Control in Power Electronics and Drives," *IEEE Transactions on Industrial Electronics*, vol. 55, no. 12, pp. 4312–4324, Nov. 2008.

[3] T. Geyer, "Low Complexity Model Predictive Control in Power Electronics and Power Systems," Ph.D. dissertation, ETH Zürich, 2005.

[4] D. Bertsimas and R. Weismantel, *Optimization over integers*. Belmont, Massachussetts: Dynamic Ideas, 2005.

[5] P. Karamanakos, T. Geyer, N. Oikonomou, F. D. Kieferndorf, and S. Manias, "Direct Model Predictive Control: A Review of Strategies That Achieve Long Prediction Intervals for Power Electronics," *IEEE Industrial Electronics Magazine*, vol. 8, no. 1, pp. 32–43, Mar. 2014.

[6] T. Geyer and D. E. Quevedo, "Multistep Finite Control Set Model Predictive Control for Power Electronics," *IEEE Transactions on Power Electronics*, vol. 29, no. 12, pp. 6836–6846, 2014.

[7] ——, "Performance of Multistep Finite Control Set Model Predictive Control for Power Electronics," *IEEE Transactions on Power Electronics*, vol. 30, no. 3, pp. 1633–1644, 2015.

[8] B. Hassibi and H. Vikalo, "On the sphere-decoding algorithm I. Expected complexity," *IEEE Transactions on Signal Processing*, vol. 53, no. 8, pp. 2806–2818, Jul. 2005.

[9] S. Kouro, P. Cortes, R. Vargas, U. Ammann, and J. Rodriguez, "Model Predictive Control - A Simple and Powerful Method to Control Power Converters," *IEEE Transactions on Industrial Electronics*, vol. 56, no. 6, pp. 1826–1838, May 2009.

[10] D. P. Bertsekas, *Dynamic programming and optimal control*. Athena Scientific Belmont, Massachusetts, 1996.

[11] D. P. de Farias and B. Van Roy, "The linear programming approach to approximate dynamic programming," *Operations Research*, 2003.

[12] Y. Wang, B. O'Donoghue, and S. Boyd, "Approximate dynamic programming via iterated Bellman inequalities," *International Journal of Robust and Nonlinear Control*, 2014.

[13] L. Vandenberghe and S. Boyd, "Semidefinite Programming," *SIAM Review*, vol. 38, no. 1, pp. 49–95, 1996.

[14] B. Stellato and P. J. Goulart, "High-Speed Finite Control Set Model Predictive Control for Power Electronics," *arXiv.org*, Oct. 2015.

[15] W. C. Duesterhoeft, M. W. Schulz, and E. Clarke, "Determination of Instantaneous Currents and Voltages by Means of Alpha, Beta, and Zero Components," *American Institute of Electrical Engineers, Transactions of the*, vol. 70, no. 2, pp. 1248–1255, Jul. 1951.

[16] A. V. Oppenheim and A. S. Willsky, *Signals and Systems*, 2nd ed. Upper Saddle River, NJ, USA: Prentice-Hall, Inc., 1997.

[17] R. E. Kalman, "When Is a Linear Control System Optimal?" *Journal of Basic Engineering*, vol. 86, no. 1, pp. 51–60, 1964.

[18] S. Boyd, L. El Ghaoui, E. Feron, and V. Balakrishnan, *Linear Matrix Inequalities in System and Control Theory*. Philadelphia: Society for industrial and applied mathematics, 1994, vol. 15.

[19] MOSEK ApS, *The MOSEK optimization toolbox for MATLAB manual. Version 7.1 (Revision 35).*, 2015.

[20] 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, May 2009.

[21] G. Papafotiou, J. Kley, K. G. 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, May 2009.

[22] J. Löfberg, "YALMIP : A Toolbox for Modeling and Optimization in MATLAB," in *Proceedings of the CACSD Conference*, Taipei, Taiwan, 2004.

[23] Gurobi Optimization, Inc., "Gurobi Optimizer Reference Manual," Tech. Rep., 2015.

[24] J. Rodriguez and P. Cortes, *Predictive control of power converters and electrical drives*. John Wiley & Sons, 2012, vol. 40.

[25] J. Holtz and B. Beyer, "Fast current trajectory tracking control based on synchronous optimal pulsewidth modulation," *IEEE Transactions on Industry Applications*, vol. 31, no. 5, pp. 1110–1120, 1995.