Embedded Model Predictive Control With Certified Real-Time Optimization for Synchronous Motors

Gionata Cimini, Daniele Bernardini, Stephen Levijoki, and Alberto Bemporad, *Fellow, IEEE*

***Abstract*— Model predictive control (MPC) is a very attractive candidate to replace standard field-oriented control algorithms for electrical motors. We demonstrate that it is possible to implement an MPC algorithm for continuous control set (CCS-MPC), with both inputs and states constraints, in which the associated quadratic programming (QP) problem is solved online, even on the computationally limited platforms used in control of electrical motors. We detail the implementation of an active-set algorithm to solve efficiently the associated QP problem. Moreover, by exploiting recent results on active-set solver certification, we are able to assess the computational complexity of the online optimization algorithm, providing the exact worst-case solution time. The controller is experimentally tested on an embedded control unit for the torque regulation of a permanent magnet synchronous motor and benchmarked against explicit MPC. Computational feasibility, low-memory occupancy, and worst-case certification are achieved, fulfilling all the requirements of embedded control.**

***Index Terms*— Complexity certification, electrical motors, embedded control, model predictive control (MPC), quadratic programming (QP), real-time optimization, synchronous drives, torque control.**

I. Introduction

IMPROVING the performance of electrical systems is strategic in many applications; therefore, model predictive control (MPC) is a very active topic for power converters and drives control [1]–[4]. As far as transistor-based systems, the literature splits into continuous control set (CCS)-MPC, which takes actions into a continuous set, corresponding to the modulator duty-cycle [5], and finite control set (FCS)-MPC, which instead manipulates directly the transistors [6], [7]. Recent research has compared the two strategies [8]–[10], with attempts to benefit from both [11]. FCS-MPC provides a faster response and possibly reduces the switching losses. However, the variable frequency increases component stress [12]–[14] and introduces a tradeoff between tracking error, harmonics, and energy efficiency [15]–[17].

CCS-MPC cancels the drawbacks associated with variable switching frequency and provides decoupling between sampling and switching times [7], [18]. It is also the preferred choice when dealing with precompensated systems [19]. Unfortunately, the online optimization problem associated with constrained CCS-MPC, and nontrivial prediction horizon, is typically considered unmanageable on low-cost platforms [20].

The computational burden has been an issue for FCS-MPC too, but recently, the combinatorial complexity has been severely reduced [21], [22], with a consequent gain in its popularity. The goal of this brief is to demonstrate that recent advances in convex optimization make CCS-MPC feasible on cheap hardware as well, providing in addition an analytical bound to its complexity [23]. The alternative to online optimization would be explicit MPC (EMPC), which presolves the optimization problem offline [24]. However, the logarithmic search time and polynomial memory occupancy make EMPC feasible only for small problems, with few inputs and a short horizon. Successful applications of EMPC for electrical motors hold only for simplified formulations, one-step ahead prediction and/or approximated constraints [25], [26].

This brief presents a CCS-MPC algorithm based on online optimization for the torque tracking of a permanent magnet synchronous motor (PMSM), which we refer to as model predictive torque control (MP-TC). We demonstrate that it is possible to efficiently solve the quadratic programming (QP) problem arising in linear MPC on a platform with scarce computational capabilities. To the best of the authors' knowledge, online CCS-MPC for motor control and embedded in a low-cost platform has been investigated only in [27] and [28]. However, in [27], the input constraints are not present, and the quality of the obtained solution is not discussed. The authors have considered input constraints in [28], but the control performance and computational feasibility have been assessed only with hardware-in-the-loop, and the worst-case run time for the solver is unknown. This brief considers an experimental setup, where the optimal control sequence is computed by an embedded active-set solver. Such optimization methods achieve very high accuracy in a fixed amount of iterations, even under single-precision arithmetic.

As a further contribution, we analytically certify the worst-case execution time of the optimization. With embedded MPC becoming increasingly popular in industry, see for instance the mass production of General Motors [29], [30], complexity certification is of paramount importance. Namely, one has to prove that the inputs will be always computed in a time interval that is smaller than the sampling one [31]. Active-set methods have suffered for a while from the lack of such theoretical bounds, despite the experimental evidence of an

Manuscript received September 15, 2018; revised July 15, 2019; accepted January 28, 2020. Manuscript received in final form February 24, 2020. Recommended by Associate Editor L. Wang. *(Corresponding author: Gionata Cimini.)*

Gionata Cimini and Daniele Bernardini are with ODYS Srl, 20159 Milan, Italy (e-mail: gionata.cimini@odys.it; daniele.bernardini@odys.it).

Stephen Levijoki is with General Motors Company, Detroit, MI 48265 USA (e-mail: stephen.levijoki@gm.com).

Alberto Bemporad is with the IMT School for Advanced Studies Lucca, 55100 Lucca, Italy, and also with ODYS Srl, 20159 Milan, Italy (e-mail: alberto.bemporad@imtlucca.it).

Color versions of one or more of the figures in this article are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TCST.2020.2977295

average polynomial complexity. The authors have solved this issue in [23] by demonstrating that the iterations taken by a dual active-set method when solving a parametric QP problem depend on a piecewise affine (PWA) fashion on the vector of parameters. Therefore, we are able to compute exactly the number of floating-point operations (flops) needed in the worst case to solve the optimization problem. A block principal pivoting method has been proven to enjoy similar properties [32] while being faster; however, the need to impose constraints on the motor currents restricts the choice to algorithms that can handle general inequality constraints, i.e., [23].

The proposed MP-TC has been tested on a commercially available PMSM, controlled by a Texas Instruments DSP, commonly used in power electronics and electrical drives. The comparison with EMPC is also shown, being the only valuable alternative to meet embedded requirements for a CCS-MPC algorithm. The results show that the proposed MP-TC highly reduces the memory occupancy, and, more interestingly, it requires a lower worst-case number of flops when compared to EMPC, increasing the throughput.

This brief is organized as follows. Section II derives the mathematical model of the PMSM, and the MP-TC algorithm is presented in Section III. The embedded solver design and certification are detailed in Section IV, and the experimental results are reported in Section V. Finally, Section VI concludes this brief.

II. Mathematical Model

The electrical subsystem of a PMSM is modeled with respect to the (d, q) reference, frame rotating synchronously with the rotor

$$\dot{i}_d(t) = -\frac{R}{L_d} i_d(t) + \frac{L_q}{L_d}\omega(t) i_q(t) + \frac{1}{L_d} u_d(t) \tag{1a}$$

$$\dot{i}_q(t) = -\frac{R}{L_q} i_q(t) - \left(\frac{L_d}{L_q} i_d(t) + \frac{\lambda}{L_q}\right)\omega(t) + \frac{1}{L_q} u_q(t) \tag{1b}$$

where $t \in \mathbb{R}_+$ is the time index, d and q are the subscripts for direct and quadrature quantities, L, R, i, and u are the stator inductance [H], resistance [Ω], current [A], and voltage [V]. The mechanical dynamics of a PMSM are as follows:

$$\dot{\omega}(t) = \frac{B}{J}\omega(t) + \frac{p}{J}\tau(t) - \frac{p}{J}\tau_l(t) \tag{2a}$$

$$\tau(t) = \frac{3}{2} p\left(\lambda i_q(t) + (L_d - L_q) i_d(t) i_q(t)\right) \tag{2b}$$

where $\omega(t)$ is the electrical rotor speed [rad/s], $\tau(t)$ is the electrical torque [Nm], J is the inertia coefficient [kg·m^2], λ is the motor flux leakage [Wb], $\tau_l(t)$ is the load torque [Nm], and p is the number of pole pairs. Let us assume that the PMSM is isotropic with $L_d \equiv L_q \equiv L$ and nominal speed ω_0, and let $K_t = 3/2p\lambda$ be the torque constant. Define the states $x(t) = [i_d(t), i_q(t)]'$, the manipulated inputs $u(t) = [u_d(t), u_q(t)]'$, the measured outputs $y(t) = [i_d(t), \tau(t)]'$, and the measured disturbances $v(t) = \omega(t)$. Then, the following linear-time-invariant (LTI) model defines an approximation of the electrical subsystem (1):

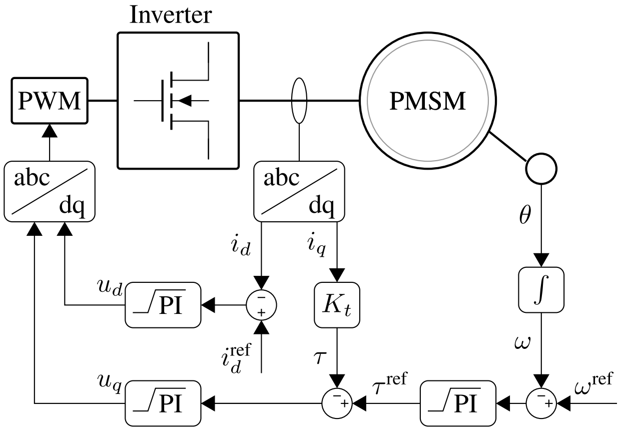

Fig. 1. Standard field-oriented control scheme for a PMSM. Both the speed loop and the two stator currents loops are controlled by linear regulators.

$$\dot{x}(t) = \underbrace{\begin{bmatrix} -\frac{R}{L} & \omega_0 \\ -\omega_0 & -\frac{R}{L} \end{bmatrix}}_{A_c} x(t) + \underbrace{\begin{bmatrix} \frac{1}{L} & 0 \\ 0 & \frac{1}{L} \end{bmatrix}}_{B_c} u(t) + \underbrace{\begin{bmatrix} 0 \\ -\frac{\lambda}{L} \end{bmatrix}}_{G_c} v(t)$$

$$y(t) = \underbrace{\begin{bmatrix} 1 & 0 \\ 0 & K_t \end{bmatrix}}_{C_c} x(t). \tag{3}$$

Such model is obtained by imposing a nominal speed in the terms $\omega(t) i_q(t)$ and $\omega(t) i_d(t)$, while allowing the flexibility of having time-varying values in the affine term $v(t)$. This improves the accuracy of the model with respect to more conservative approaches where the effect of the speed is completely linearized around a steady-state value [27].

III. MPC Design

Standard PI-FOC consists of a cascaded architecture, where the outer loop regulates either the rotor speed or position and the inner loop controls the (d, q) stator currents (see Fig. 1). MP-TC replaces the controller of the electrical subsystem, keeping the external loop unchanged (see Fig. 2). The focus on the inner loop is motivated by the faster dynamics and the necessity to impose voltage and current constraints. In a not-weakening flux operation, the direct component i_d has to be stabilized at zero, while the quadrature component i_q has to track the set-point i_q^{ref}, obtained by scaling the torque reference τ^{ref} by the constant K_t. With isotropic machines, this control algorithm is effective, as maximum current implies maximum torque. However, in order to tackle field weakening, maximum torque per ampere (MTPA) could be considered [7], [27] but that is out of the scope of this brief.

A. MPC Formulation

Linear MPC solves a finite-horizon, constrained, optimal control problem based on a linear prediction model obtained by discretizing (3) with sampling time T_s, such that

$$\begin{aligned} x(k+1) &= Ax(k) + Bu(k) + Gv(k) \\ y(k) &= Cx(k) \end{aligned} \tag{4}$$

where $k \in \mathbb{N}$ is the discrete-time index, $A = e^{A_c T_s}$, $B = \int_0^{T_s} e^{A_c \tau} d\tau B_c$, $G = \int_0^{T_s} e^{A_c \tau} d\tau G_c$, and $C = C_c$. The goal

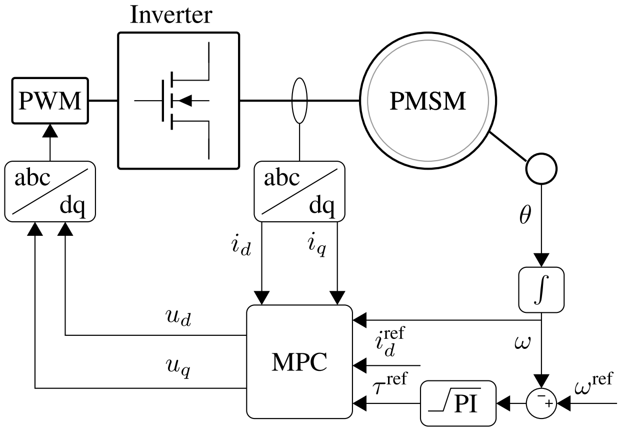

Fig. 2. Proposed control scheme for PMSM. The speed loop is controlled by a standard regulator, and the inner loop is implemented the MP-TC here introduced.

is to track the reference $\tau^{\rm ref}$ while constraining voltages and currents. Therefore, the following optimization problem is solved at every time step k:

$$\min_{\Delta\bar{u}} \ \sum_{i=1}^{N_p} \|W_y(y_{k+i|k} - r(k))\|_2^2 + \sum_{j=0}^{N_u-1} \|W_{\Delta u}\Delta u_{k+j|k}\|_2^2 \tag{5a}$$

$$\text{s.t.} \quad x_{k|k} = x(k) \tag{5b}$$

$$x_{k+i+1|k} = Ax_{k+i|k} + Bu_{k+i|k} + Gv(k) \tag{5c}$$

$$y_{k+i+1|k} = Cx_{k+i+1|k} \tag{5d}$$

$$\Delta u_{k+N_u+j|k} = 0, \quad j = 0, \dots, N_p - n_u - 1 \tag{5e}$$

$$u_{k+i|k} \in \mathbb{U} \tag{5f}$$

$$x_{k+i+1|k} \in \mathbb{X} \tag{5g}$$

$$i = 0, 1, \dots, N_p - 1 \tag{5h}$$

where $\Delta\bar{u} = \{\Delta u_{k|k}, \dots, \Delta u_{N_u-1|k}\}$ is the sequence of input increments, and $\Delta u_{k+i|k} = u_{k+i|k} - u_{k+i-1|k}$, $i = 0, 1, \dots, N-1$, with $u_{k-1|k} = u(k-1)$. In (5), N_p and N_u are the prediction and control horizons, and $r(k) = [0, \tau^{\rm ref}(k)]'$, $x_{k+i|k}$ denotes the prediction of the variable x at time $k+i$ based on the information available at time k, and W_y and $W_{\Delta u}$ are the weight matrices of appropriate dimensions. Inputs and state constraints are imposed by (5f) and (5g), respectively, where $\mathbb{U}$ and $\mathbb{X}$ are polyhedral sets defined in Section III-B.

B. Constraints

We need to impose constraints on stator voltages and currents [33]. Specifically, the inverter imposes a phase voltage limit linked to the modulation scheme. Given a dc-bus voltage $V_{\rm dc}$, such limit is set to $V_{\max} = V_{\rm dc}/\sqrt{3}$ for both space-vector and pulsewidth modulations [27]. The maximum stator current $I_{\max}$ is imposed to prevent overheating, and their violation for short intervals due to constraints' softening of (5g) is usually permitted. Such bounds get transformed into norm constraints in the (d, q) reference frame, such as

$$u \in \tilde{\mathbb{U}} = \{u \in \mathbb{R}^2 : |u\|_2 \ \leq \ V_{\max}\} \tag{6a}$$

$$x \in \tilde{\mathbb{X}} = \{x \in \mathbb{R}^2 : |x\|_2 \ \leq \ I_{\max}\}. \tag{6b}$$

A polytopic approximation of the quadratic constraints (6) helps reducing the complexity of the optimization problem [25], [27], and in this brief, we approximate the feasible regions as in Fig. 3, which results into an acceptable tradeoff between the accuracy and number of inequality constraints.

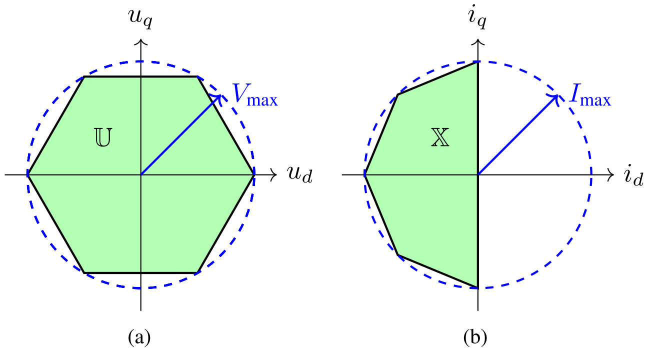

Fig. 3. (a) Inputs (voltage) and (b) outputs (current) constraints imposed by the MPC controller. The blue circles represent the original norm constraints of (6). The green regions represent their approximations and are the polyhedra described by the sets $\mathbb{U}$ and $\mathbb{X}$, respectively.

C. State Estimation and Integral Action

Although all states are measured for this application, the controller takes advantage of an observer to reduce the impact of measurement noise, e.g., due to transistor switching. A Kalman filter is employed to mitigate such noise, using the state estimation $\hat{x}_{k|k}$ instead of the measured state $x(k)$ in (5b). We introduce a delay of one time step in the input channel to account for the time spent in computing the optimal control move. Let L be the Kalman filter gain, the one-step delayed implementation is achieved by computing the state estimate as follows:

$$\hat{x}_{k+1|k} = (A - LC)\hat{x}_{k|k-1} + Bu_{k-1} + G_v v_k + Ly_k. \tag{7}$$

Integral action is added to the torque reference to minimize tracking error, thus dealing with model uncertainties and noise. This allows us to reject constant disturbances without affecting the optimization problem size. The modified torque reference is denoted by $\tilde{y}^{\rm ref} = [\tilde{i}_d^{\rm ref}, \tilde{\tau}^{\rm ref}]'$ and computed as follows:

$$\tilde{i}_d^{\rm ref}(k) = \tilde{i}_d^{\rm ref}(k-1) + k_1(i_d^{\rm ref}(k) - i_d(k)) \tag{8a}$$

$$\tilde{\tau}^{\rm ref}(k) = \tilde{\tau}^{\rm ref}(k-1) + k_2(\tau^{\rm ref}(k) - \tau(k)) \tag{8b}$$

where k_1 and k_2 are the scalar parameters. Fig. 4 shows the complete real-time MPC controller scheme.

IV. Online Optimization and Complexity Certification

Let us rewrite (5) as a the following parametric QP problem:

$$\begin{aligned} &\min_z \ \frac{1}{2}z'Hz + \theta_k{}'F'z \\ &\text{s.t. } Gz \leq S\theta_k + s \end{aligned} \tag{9}$$

where $z = \begin{bmatrix}\Delta\bar{u}' \ \rho'\end{bmatrix}' \in \mathbb{R}^{n_z}$ is the vector of optimization variables, collecting the sequence of input increments $\Delta\bar{u}$ and the vector ρ of slack variables used for softening state constraints, $\theta_k \in \Theta \in \mathbb{R}^{n_\theta}$ is the vector of parameters defined as $\theta_k = [u'_{k-1}, \ \hat{x}'_{k+1|k}, \ r'_k, \ v'_k]'$, with Θ a set of interest, $H \in \mathbb{R}^{n_z \times n_z}$ is a symmetric and positive definite matrix, $F \in \mathbb{R}^{n_z \times n_\theta}$, $G \in \mathbb{R}^{m_z \times n_z}$, and $S \in \mathbb{R}^{m_z \times n_\theta}$ and $s \in \mathbb{R}^{m_z}$.

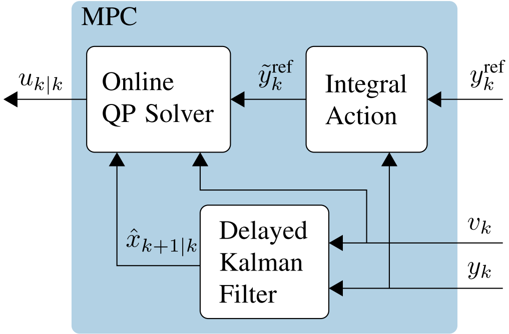

Fig. 4. Block scheme of the MPC controller, with delayed Kalman filter for state estimation, integral action for offset-free tracking, and the online solver to obtain the inputs sequence.

Solving (9) accurately and with a throughput in the millisecond scale is challenging on embedded hardware. One option could be EMPC, in which the QP is solved offline for all z, and the optimal control law is obtained via multiparametric programming [24]. However, EMPC is less attractive when the number of regions of the resulting PWA controller grows, as it could require a significant amount of memory. Indeed, in Section V, we show that the memory occupancy of EMPC would be too high for this application.

In this brief, we implemented an efficient QP solver based on an active-set method that will be shown to solve problem (9) within the allowed time limit, and with an exact assessment of the worst-case execution time.

A. Active-Set Solver

Definition 1 (Active Set): Given a QP problem as in (9)*, a constraint is said to be active at* $\bar{z}$ *if it is satisfied as equality constraint, i.e.,* $G_i\bar{z} = S_i\theta_k + s_i$*, where* A_i *represents the ith row of a matrix* A *and* a_i *represents the ith element of a vector* a*. Otherwise, it is inactive. We define the two sets*

$$\mathcal{A}(\bar{z}) = \{i \in \mathcal{K} \mid G_i\bar{z} = S_i\theta_k + s_i\} \quad (10a)$$

$$\mathcal{I}(\bar{z}) = \{i \in \mathcal{K} \mid G_i\bar{z} < S_i\theta_k + s_i\} \quad (10b)$$

where $\mathcal{K} = \{1, \ldots, m\}$ *collects the constraint indexes,* $\mathcal{A}$ *is the active set, and* $\mathcal{I}$ *is the inactive set, such that* $\mathcal{I} = \mathcal{K} \setminus \mathcal{A}$*.*

Given $\mathcal{A}^*$ the set of constraints active at the optimal solution z^*, the idea behind active-set methods is to iteratively update the optimal constraints guess $\mathcal{A}^j$, known as working-set, until $\mathcal{A}^j \equiv \mathcal{A}^*$ is verified. Let $\mathcal{B}^j$ indicate the set of blocking constraints, namely, those that prevent a full step in the primal space without violating dual feasibility. Then, at each iteration j, a constraint $v^j \in \mathcal{V}^j \subseteq \mathcal{I}$ is selected, with $\mathcal{V}^j$ being the set of all the violated constraints, such that $\mathcal{A}^j = \{\mathcal{A}^{j-1} \cup v^j\} \setminus \mathcal{B}^j$. A step toward the optimal solution is obtained by solving the reduced equality-constrained QP problem

$$\begin{aligned} z^j = \arg\min_z & \frac{1}{2}z'Hz + \theta_k'F'z \\ \text{s.t. } & G_{\mathcal{A}^j}z = S_{\mathcal{A}^j}\theta_k + s_{\mathcal{A}^j}. \end{aligned} \quad (11)$$

Problem (11) amounts to find a primal-dual pair $(z^j, \pi_{\mathcal{A}^j}^j)$ that solves the Karush–Kuhn–Tucker (KKT) system

$$\text{KKT}(\mathcal{A}^j) \qquad \begin{bmatrix} H & G_{\mathcal{A}^j}' \\ G_{\mathcal{A}^j} & 0 \end{bmatrix} \begin{bmatrix} z^j \\ \pi^j \end{bmatrix} = \begin{bmatrix} -F\theta_k \\ S_{\mathcal{A}^j}\theta_k + s_{\mathcal{A}^j} \end{bmatrix} \quad (12)$$

with $\pi = \left[\pi_{\mathcal{A}^j}' \; \pi_{\mathcal{I}^j}'\right]'$ and $\pi_{\mathcal{I}^j} = \mathbf{0}$. The set $\mathcal{B}^j \subseteq \mathcal{A}^{j-1}$ is composed by all those constraints that would prevent to solve the system KKT($\{\mathcal{A}^{j-1} \cup v^j\}$) without incurring in dual infeasibility and that have to be dropped from the current working set. Algorithm 1 describes the pseudocode of the implemented dual active-set solver, which starts from the unconstrained solution $H^{-1}F\theta_k$ and iterates toward primal feasibility by adding the most violated constraint at each iteration, preserving dual feasibility. If z^j is primal feasible, that is, $G_iz^j \leq S_i\theta_k + s_i \, \forall i \in \{1, \ldots, m\}$, then $z^j \equiv z^*$ holds [34]. Dual active-set solvers are preferable to primal ones because at the price of infeasible subiterates, they do not require Phase I to find a feasible initial guess and usually find the optimizer with less iterations [35].

Algorithm 1 Dual Active-Set Solver

Input: Matrices H, F, G, S, s of problem (9), vector of parameters θ_k and set $\mathcal{K}$

```
$z^0 \leftarrow -H^{-1}F\theta_k$, $\pi^0 \leftarrow 0$, $j \leftarrow 0$, $\mathcal{A}^{j-1} \leftarrow \emptyset$;
$\mathcal{V}^0 \leftarrow \{i \in \mathcal{K} \mid G_iz^0 > S_i\theta_k + s_i\}$;
while $\mathcal{V}^j \neq \emptyset$ do
    $v^j \leftarrow \arg\max\{G_iz^j - S_i\theta_k - s_i, \forall i \in \mathcal{V}^j\}$;
    $j \leftarrow j + 1$;
    Find blocking constraints $\mathcal{B}^j$;
    $\mathcal{A}^j \leftarrow \{\mathcal{A}^{j-1} \cup v^j\} \setminus \mathcal{B}^j$;
    Solve KKT($\mathcal{A}^j$) with respect to $(z^j, \pi^j)$;
    $\mathcal{V}^j \leftarrow \{i \in \mathcal{K} \mid G_iz^j > S_i\theta_k + s_i\}$;
end while
```

Output: Optimal solution $z^* = z^j$ and active set $\mathcal{A}^* = \mathcal{A}^j$

B. Complexity Certification of the Dual Active-Set Algorithm

Knowing the computational complexity of the controller is of the highest importance in embedded real-time systems, in order to ensure that the control routine is always performed within the sampling time. The interest in certifying QP solvers is a very recent matter, as a consequence of the interest in embedded applications for MPC [23], [31]. Cimini and Bemporad [23] have introduced the theory and methods to compute exactly the worst-case execution time of a dual active-set solver, such as Algorithm 1, when solving a QP problem with the linear term of the cost function and right-hand side of the constraints depending linearly on a vector of parameters as in (9). The objective of the certification algorithm is to exactly compute the maximum number of flops $\mathcal{F}^{\max}$ performed by Algorithm 1 when solving problem (9) for any $\theta_k \in \Theta$, in order to guarantee that $t(\mathcal{F}^{\max}) < T_s$, where $t(\mathcal{F})$ is the time needed by the selected processor to perform $\mathcal{F}$ flops. The core result behind the certification algorithm is summarized by the following lemma.

Lemma 1: Let $\Theta \subseteq \mathbb{R}^{n_\theta}$ *be a polyhedron and* z^0, π^0 *be two affine functions of* $\theta \in \Theta$*, then for each iteration* $j \in \mathbb{N}$ *of Algorithm 1, the primal-dual pair* (z^j, π^j) *obtained by solving KKT(*$\mathcal{A}^j$*) is PWA, and* $N : \Theta \rightarrow \{0, \ldots, N^{\max}\}$ *is an integer piecewise constant function, with* $N^{\max} = \max_{\theta \in \Theta} N(\theta)$ *the maximum number of iterations performed for all* $\theta \in \Theta$*.*

This article has been accepted for inclusion in a future issue of this journal. Content is final as presented, with the exception of pagination.

Proof: The reader is referred to [23, Th. IV.4]. □

Given the result of Lemma 1, it is possible to certify Algorithm 1 by iteratively splitting Θ into polyhedral subregions according to the possible constraints added and dropped at each iteration. The polyhedra generated in this way at a given iteration j represent regions of the parameter space where the solution can be either optimal, infeasible, or requiring further iterates. Let us define the PWA primal dual pair as follows:

$$z^j(\theta_k) = A_z^j \theta_k + b_z^j \tag{13a}$$

$$\pi^j(\theta_k) = A_\pi^j \theta_k + b_\pi^j \tag{13b}$$

with $A_z^0 = -H^{-1}F$, $b_z = \mathbf{0}^{n_z}$, $A_\pi^0 = \mathbf{0}^{m_z \times n_\theta}$, and $b_\pi^0 = \mathbf{0}^{m_z}$. Then, z^0 and π^0 are affine functions of θ and the certification algorithm builds a set $\mathbb{T}_{\text{opt}}$ of optimal tuples and a set $\mathbb{T}_{\text{inf}}$ of infeasible ones, where the generic tuple T^i is uniquely defined by the following parameters:

$$T^i = (\Theta^i, \mathcal{F}^i, j, \mathcal{A}^i, A_z^i, b_z^i, A_\pi^i, b_\pi^i). \tag{14}$$

It is, therefore, possible to certify the worst-case complexity by computing $\mathcal{F}^{\max} = \max(\mathcal{F}_{\text{opt}}^{\max}, \mathcal{F}_{\text{inf}}^{\max})$, where $\mathcal{F}_{\text{opt}}^{\max}$ and $\mathcal{F}_{\text{inf}}^{\max}$ are the number of flops needed in the worst case to reach the optimum and to detect infeasibility, computed as follows:

$$\mathcal{F}_{\text{opt}}^{\max} = \max_{T \in \mathbb{T}_{\text{opt}}} \{\mathcal{F}(T)\} \tag{15a}$$

$$\mathcal{F}_{\text{inf}}^{\max} = \max_{T \in \mathbb{T}_{\text{inf}}} \{\mathcal{F}(T)\} \tag{15b}$$

where $\mathcal{F}(T)$ denotes the value $\mathcal{F}$ associated with the tuple T. The certification is also useful to identify if there is any region in the parameters' space where the QP algorithm would fail due to problem infeasibility. Moreover, the explicit solution can be easily obtained as a by-product of the certification algorithm [23], given that for each tuple, we know the active constraints and the piecewise functions (13). Such feature will be exploited in Section V to compare the embedded QP solver with respect to its corresponding explicit version.

V. Experimental Results

The proposed MP-TC algorithm with real-time optimization has been tested on the commercially available MBE.300.E500 PMSM, by Technosoft SA. The control unit is an F28335 Delfino DSP by Texas Instruments, which belongs to the C2000 series. It has a 32-bit, 150-MHz CPU (6.67-ns cycle time) and an IEEE-754 single-precision Floating-Point Unit, with an integrated multiplier (32×32 bit). This hardware has been chosen to demonstrate the viability of implementing an online QP solver in a low-cost platform like one of the C2000 family, which is commonly used in motion control.

The motor specifications are listed in Table I. The control scheme of Fig. 4 and the QP solver have been implemented in *C* code to maximize the efficiency. Two interrupt levels manage the controller scheduling: 1) a fast loop for current control with 0.3-ms sampling time and 2) a slow loop for speed control with 1.2-ms sampling time. The main design parameters of the MP-TC are summarized in Table II. In the next, the so designed controller will be referred to as implicit MP-TC, and later compared to explicit MP-TC for which the online optimization is replaced by EMPC.

The certification algorithm described in Section IV-B has been applied to the MPC setup of Table II, in order to compute the worst-case behavior of the solver and, thus, validate offline the controller's complexity feasibility. The set Θ has been derived in accordance with the physical constraints collected in Table I. Fig. 5 shows a 2-D projection in the parameter space of the optimal polyhedra corresponding to the tuples $T^i \subseteq \mathbb{T}_{\text{opt}}\, \forall i = 1, \ldots, \mathbf{card}(\mathbb{T}_{\text{opt}})$. The same color means the same number of iterations performed by the solver, but with possibly different number of flops due to a different

TABLE I
Technosoft MBE.300.E500 PMSM Parameters

Parameter	Units	Value
Phase-phase resistance	ohm	8.61
Phase-phase inductance	mH	07.13
Back-EMF constant	V/1000 rpm	3.86
Torque constant	mNm/A	36.8
Pole pairs	–	1
Rated voltage	V	36
Max. voltage	V	58
No-load current	mA	73.2
No-load speed	rpm	9170
Max. cont. current (at 5000 rpm)	mA	913
Max. cont. torque (at 5000 rpm)	mNm	30
Max. permissible speed	rpm	15000
Peak torque (stall)	mNm	154
Rotor inertia	$\text{kgm}^2 \cdot 10^{-7}$	11
Mechanical time constant	ms	7

TABLE II
Design Parameters of MP-TC

Prediction horizon N_p	3
Control horizon N_u	1
Voltage limit $V_{\max}$	$24/\sqrt{3}$ V
Current limit $I_{\max}$	1 A
Output weights W_y and P	$\begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$
Input increments weights $W_{\Delta u}$	$\begin{bmatrix} 0.01 & 0 \\ 0 & 0.01 \end{bmatrix}$
Sampling time T_s	0.3 ms

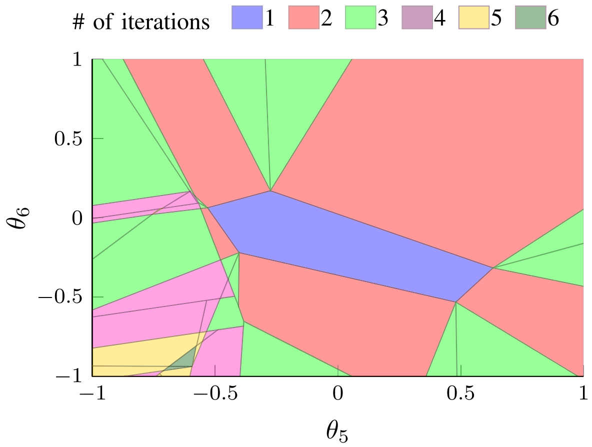

Fig. 5. 2-D projection of the certification algorithm result over the plane defined by QP parameters $p_5 \equiv i_d^{\text{ref}}$ and $p_6 \equiv i_q^{\text{ref}}$. Polyhedral regions with the same color share the same number of iterations but possibly different flops to reach the optimum.

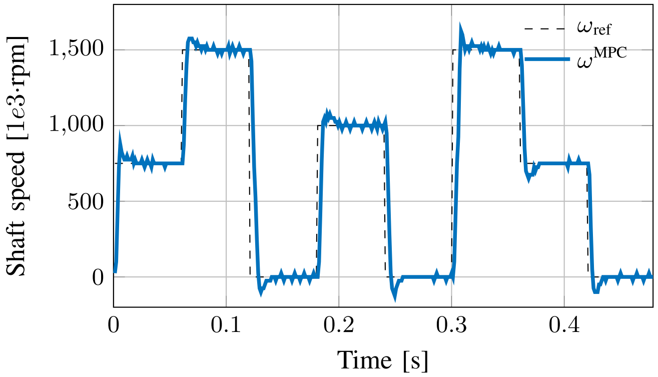

Fig. 6. Speed-tracking performance for MP-TC controller experimentally tested on MBE.300.E500 motor. Abrupt step changes in the reference shaft speed are applied to test the QP solver when constraints are active.

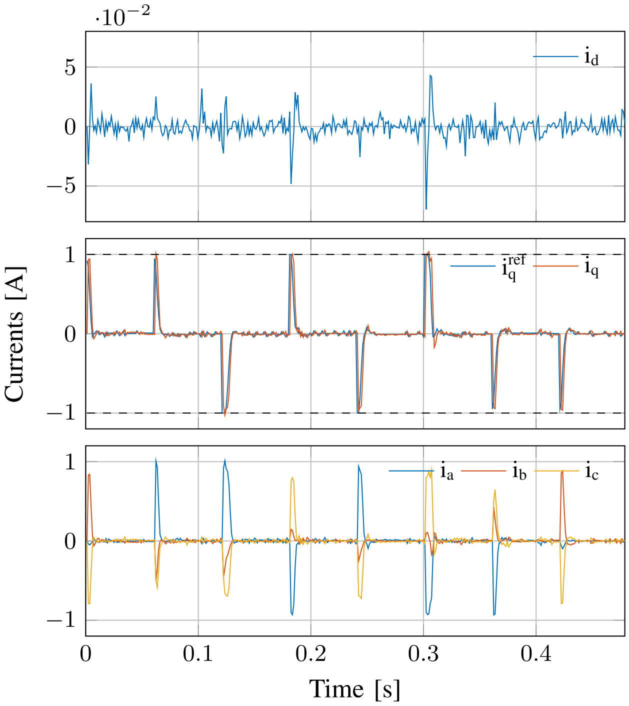

Fig. 7. Stator currents for the experimental test of MP-TC controller. From top to bottom: 1) stator current in the direct frame; 2) stator current in the quadrature frame (red) and the reference to track (blue); and 3) three-phase stator currents.

sequence of constraints added or removed from the active set. The algorithm certifies that $\mathbb{T}_{\text{inf}} = \emptyset$ which guarantees that the solver is never infeasible on the set Θ, and that the worst-case number of flops required by the implicit MP-TC controller is $\mathcal{F}_I^{\max} = 2431$, among which ten are square roots. The memory occupation m_I of both code and data is computed to be 12.7 kB which is well below the single-access RAM block of 34 kB provided by the F28335. The desired rotor speed ω^{ref} and the achieved speed ω are shown in Fig. 6. Abrupt changes of the speed are requested in order to operate the system close to the constraints, thus stressing the QP solver for speed evaluation. Fig. 7 shows the stator currents in the (d, q) reference frame and the corresponding three-phase currents. Fig. 8 presents the (d, q) stator voltages together with the corresponding three-phase variables. As shown, the system constraints are correctly handled. Fig. 9 shows the task times obtained by means of a high-precision internal clock and the number of iterations needed by the solver to obtain the optimal control sequence. The solution is always computed within the time limit of 0.3 ms. This time includes ADC sampling, state estimation, and real-time optimization.

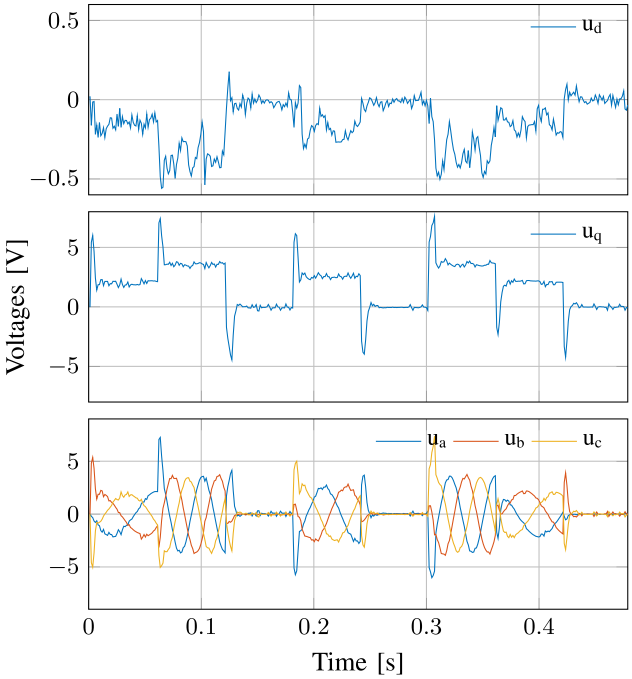

Fig. 8. Stator voltages for the experimental test of MP-TC controller. From top to bottom: 1) stator voltage in the direct frame; 2) stator voltage in the quadrature frame; and 3) three-phase stator voltages.

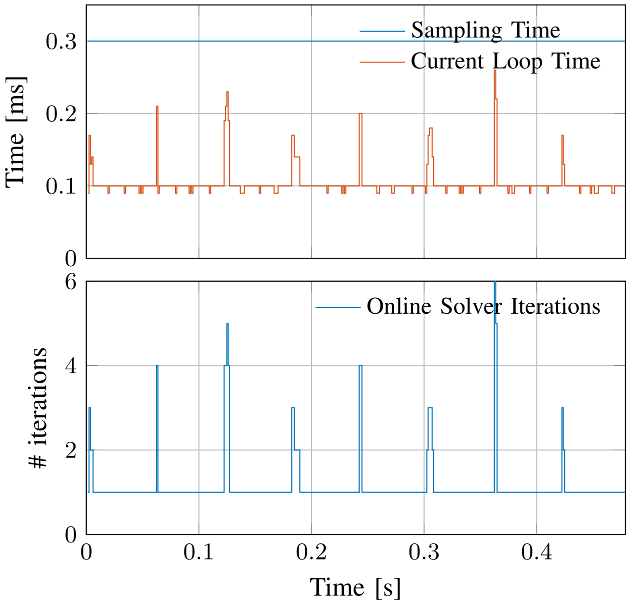

Fig. 9. Timing performance of the F28335 DSP for the experimental test. From top to bottom: 1) the time acquisitions of the control routine (red), compared to the time limit allowed (blue) and 2) the number of iterations performed by the QP algorithm.

The proposed implicit MP-TC is compared to explicit MP-TC. For a more comprehensive benchmark, two configurations have been proposed. Setup 1 is the one discussed above, whereas Setup 2 denotes a simplified version with the polyhedral constraints on the currents replaced by box

TABLE III

PERFORMANCE BENCHMARK OF IMPLICIT AND EXPLICIT MP-TCS WITH DIFFERENT PROBLEM SETUPS AND CONTROL PARAMETERS

	Setup 1			Setup 2		
N_p	2	3	4	2	3	4
			Implicit MP-TC			
m_I 32-bit [kB]	12.4	12.7	12.9	12.1	12.2	12.3
$\mathcal{F}_I^{\max}$ $(\pm, *, \div)$|sqrt	1932|10	2421|10	2904|10	944|7	1436|10	1598|10
$N^{\max}$	6	6	6	5	6	6
			Explicit MP-TC			
m_E 32-bit [kB]	31.1	50.9	62.8	21.8	36.6	45.6
$\mathcal{F}_E^{\max}$ $(\pm, *)$	2333	3868	4806	1740	2737	3812
N_R	82	133	163	54	95	119
t_{CE} [s]	71.7	182.5	281.7	4.3	14.7	24.7

constraints in the (d, q) frame, similar to [25], such that

$$i_d \in [-\epsilon I_{\max}, \epsilon I_{\max}] \tag{16a}$$

$$i_q \in [I_{\max}, I_{\max}] \tag{16b}$$

with $\epsilon > 0$. The flops and memory occupancy of explicit MP-TC, respectively, identified by $\mathcal{F}_E^{\max}$ and m_E, are obtained by considering the implementation described in [36, Algorithm 4]. For each of the two controller setups, different prediction horizons are also considered, and the complete benchmark is summarized in Table III, where N_R indicates the number of explicit regions and t_{CE} is the time required by the certification algorithm on a 2.2-GHz Intel Core i7-8750H. The results show that the so implemented implicit MP-TC is always faster and less eager in memory than EMPC, with the performance scaling more favorably for implicit MP-TC when increasing the horizon. Interestingly, explicit MP-TC is feasible only in two out of six configurations, namely, the ones where the horizon is shortened with respect to the original setup, exceeding for the others both time and memory constraints. Implicit MP-TC is always feasible besides the single case of Setup 1 with increased horizon. This not only proves the feasibility of CCS-MPC embedded on cheap hardware for motor control but also that implicit MPC can be more efficient than EMPC even on a small problem.

VI. CONCLUSION

This brief has proposed an MP-TC method with online optimization for PMSMs. The optimal control inputs are obtained by solving a QP problem with an efficient, active-set, embedded solver. The computational complexity of the solver has been exactly assessed, fulfilling the mandatory requirement of worst-case time estimation for embedded control. We have shown that the proposed MP-TC with online solver is feasible on a control unit with scarce computational resources. The algorithm has been experimentally tested and compared to EMPC which is considered so far one of the few options to implement MPC in motion control. The results shows that the proposed QP solver not only saves considerable memory with respect to EMPC but also certifies to be faster in the worst case, even if EMPC is implemented with an approximated formulation.

REFERENCES

[1] P. Cortes, M. Kazmierkowski, R. Kennel, D. Quevedo, and J. Rodriguez, “Predictive control in power electronics and drives,” *IEEE Transactions on Industrial Electronics*, vol. 55, no. 12, pp. 4312–4324, Dec. 2008.

[2] H. Guzman, M. J. Duran, F. Barrero, B. Bogado, and S. Toral, “Speed control of five-phase induction motors with integrated open-phase fault operation using model-based predictive current control techniques,” *IEEE Trans. Ind. Electron.*, vol. 61, no. 9, pp. 4474–4484, Sep. 2014.

[3] Y. Zhang, D. Xu, J. Liu, S. Gao, and W. Xu, “Performance improvement of model-predictive current control of permanent magnet synchronous motor drives,” *IEEE Trans. Ind. Appl.*, vol. 53, no. 4, pp. 3683–3695, Jul. 2017.

[4] S. Kouro, M. A. Perez, J. Rodriguez, A. M. Llor, and H. A. Young, “Model predictive control: MPC’s role in the evolution of power electronics,” *IEEE Ind. Electron. Mag.*, vol. 9, no. 4, pp. 8–21 Dec. 2015.

[5] Z. Mynar, L. Vesely, and P. Vaclavek, “PMSM model predictive control with field-weakening implementation,” *IEEE Trans. Ind. Electron.*, vol. 63, no. 8, pp. 5156–5166, Aug. 2016.

[6] J. Rodriguez *et al.*, “State of the art of finite control set model predictive control in power electronics,” *IEEE Trans Ind. Informat.*, vol. 9, no. 2, pp. 1003–1016, May 2013.

[7] M. Preindl and S. Bolognani, “Model predictive direct torque control with finite control set for PMSM drive systems, part 2: Field weakening operation,” *IEEE Trans Ind. Informat.*, vol. 9, no. 2, pp. 648–657, May 2013.

[8] A. A. Ahmed, B. K. Koh, and Y. I. Lee, “A comparison of finite control set and continuous control set model predictive control schemes for speed control of induction motors,” *IEEE Trans Ind. Informat.*, vol. 14, no. 4, pp. 1334–1346, Apr. 2018.

[9] C.-S. Lim, E. Levi, M. Jones, N. A. Rahim, and W.-P. Hew, “A comparative study of synchronous current control schemes based on FCS-MPC and PI-PWM for a two-motor three-phase drive,” *IEEE Trans. Ind. Electron.*, vol. 61, no. 8, pp. 3867–3878, Aug. 2014.

[10] F. Morel, X. Lin-Shi, J.-M. Retif, B. Allard, and C. Buttay, “A comparative study of predictive current control schemes for a permanent-magnet synchronous machine drive,” *IEEE Trans. Ind. Electron.*, vol. 56, no. 7, pp. 2715–2728, Jul. 2009.

[11] H. T. Nguyen and J.-W. Jung, “Disturbance-Rejection-Based model predictive control: Flexible-mode design with a modulator for three-phase inverters,” *IEEE Trans. Ind. Electron.*, vol. 65, no. 4, pp. 2893–2903, Apr. 2018.

[12] T. Geyer, G. Papafotiou, and M. Morari, “Model predictive direct torque control—Part I: Concept, algorithm, and analysis,” *IEEE Trans. Ind. Electron.*, vol. 56, no. 6, pp. 1894–1905, Jun. 2009.

[13] F. Mwasilu, E.-K. Kim, M. S. Rafaq, and J.-W. Jung, “Finite-set model predictive control scheme with an optimal switching voltage vector technique for high-performance IPMSM drive applications,” *IEEE Trans Ind. Informat.*, vol. 14, no. 9, pp. 3840–3848, Sep. 2018.

[14] J. I. Metri, H. Vahedi, H. Y. Kanaan, and K. Al-Haddad, “Real-time implementation of model-predictive control on seven-level packed U-cell inverter,” *IEEE Trans. Ind. Electron.*, vol. 63, no. 7, pp. 4180–4186, Jul. 2016.

[15] M. R. Arahal, F. Barrero, M. J. Durán, M. G. Ortega, and C. Martín, “Trade-offs analysis in predictive current control of multi-phase induction machines,” *Control Eng. Pract.*, vol. 81, pp. 105–113, Dec. 2018.

[16] M. R. Arahal, C. Martin, F. Barrero, and M. J. Duran, “Assessing variable sampling time controllers for five-phase induction motor drives,” *IEEE Trans. Ind. Electron.*, vol. 67, no. 4, pp. 2523–2531, Apr. 2020.

[17] A. D. Alexandrou, N. Adamopoulos, and A. Kladas, “Development of a constant switching frequency deadbeat predictive control technique for field-oriented synchronous permanent-magnet motor drive,” *IEEE Trans. Ind. Electron.*, vol. 63, no. 8, pp. 5167–5175, Aug. 2016.

[18] M. Preindl and S. Bolognani, “Model predictive direct speed control with finite control set of PMSM drive systems,” *IEEE Trans. Power Electron.*, vol. 28, no. 2, pp. 1007–1015, Feb. 2013.

[19] L. Cavanini, G. Cimini, and G. Ippoliti, “Model predictive control for pre-compensated power converters: Application to current mode control,” *J. Franklin Inst.*, vol. 356, no. 4, pp. 2015–2030, Mar. 2019.

[20] S. Rubino, R. Bojoi, S. A. Odhano, and P. Zanchetta, “Model predictive direct flux vector control of multi-three-Phase induction motor drives,” *IEEE Trans. Ind. Appl.*, vol. 54, no. 5, pp. 4394–4404, Sep. 2018.

[21] D. E. Quevedo, R. P. Aguilera, and T. Geyer, *Model Predictive Control for Power Electronics Applications*. Cham, Switzerland: Springer, 2019, pp. 551–580.

[22] P. Karamanakos, T. Geyer, and R. P. Aguilera, "Long-horizon direct model predictive control: Modified sphere decoding for transient operation," *IEEE Trans. Ind. Appl.*, vol. 54, no. 6, pp. 6060–6070, Nov. 2018.

[23] G. Cimini and A. Bemporad, "Exact complexity certification of active-set methods for quadratic programming," *IEEE Trans. Autom. Control*, vol. 62, no. 12, pp. 6094–6109, Dec. 2017.

[24] A. Bemporad, M. Morari, V. Dua, and E. N. Pistikopoulos, "The explicit linear quadratic regulator for constrained systems," *Automatica*, vol. 38, no. 1, pp. 3–20, Jan. 2002.

[25] S. Bolognani, S. Bolognani, L. Peretti, and M. Zigliotto, "Design and implementation of model predictive control for electrical motor drives," *IEEE Trans. Ind. Electron.*, vol. 56, no. 6, pp. 1925–1936, Jun. 2009.

[26] J. Scoltock, T. Geyer, and U. K. Madawala, "A comparison of model predictive control schemes for MV induction motor drives," *IEEE Trans Ind. Informat.*, vol. 9, no. 2, pp. 909–919, May 2013.

[27] M. Preindl, S. Bolognani, and C. Danielson, "Model predictive torque control with PWM using fast gradient method," in *Proc. Twenty-Eighth Annu. IEEE Appl. Power Electron. Conf. Exposit. (APEC)*, Mar. 2013, pp. 2590–2597.

[28] G. Cimini, D. Bernardini, A. Bemporad, and S. Levijoki, "Online model predictive torque control for permanent magnet synchronous motors," in *Proc. IEEE Int. Conf. Ind. Technol. (ICIT)*, Mar. 2015, pp. 2308–2313.

[29] A. Bemporad, D. Bernardini, R. Long, and J. Verdejo, "Model predictive control of turbocharged gasoline engines for mass production," SAE Tech. Paper 2018-01-0875, Apr. 2018.

[30] A. Bemporad, D. Bernardini, M. Livshiz, and B. Pattipati, "Supervisory model predictive control of a powertrain with a continuously variable transmission," SAE Tech. Paper 2018-01-0860, Apr. 2018.

[31] I. Necoara, "Computational complexity certification for dual gradient method: Application to embedded MPC," *Syst. Control Lett.*, vol. 81, pp. 49–56, Jul. 2015.

[32] G. Cimini and A. Bemporad, "Complexity and convergence certification of a block principal pivoting method for box-constrained quadratic programs," *Automatica*, vol. 100, pp. 29–37, Feb. 2019.

[33] A. Damiano, G. Gatto, I. Marongiu, A. Perfetto, and A. Serpi, "Operating constraints management of a surface-mounted PM synchronous machine by means of an FPGA-based model predictive control algorithm," *IEEE Trans Ind. Informat.*, vol. 10, no. 1, pp. 243–255, Feb. 2014.

[34] A. Forsgren, P. E. Gill, and E. Wong, "Primal and dual active-set methods for convex quadratic programming," *Math. Program.*, vol. 159, nos. 1–2, pp. 469–508, Dec. 2015.

[35] H. J. Ferreau, H. G. Bock, and M. Diehl, "An online active set strategy to overcome the limitations of explicit MPC," *Int. J. Robust Nonlinear Control*, vol. 18, no. 8, pp. 816–830, 2008.

[36] M. Baotić, F. Borrelli, A. Bemporad, and M. Morari, "Efficient on-line computation of constrained optimal control," *SIAM J. Control Optim.*, vol. 47, no. 5, pp. 2470–2489, Jan. 2008.