Fast Solver for Implicit Continuous Set Model Predictive Control of Electric Drives

This paper was downloaded from TechRxiv (<https://www.techrxiv.org>).

LICENSE

CC BY 4.0

SUBMISSION DATE / POSTED DATE

28-07-2021 / 03-08-2021

CITATION

Torchio, Riccardo; Favato, Andrea; Carlet, Paolo Gherardo; Toso, Francesco; Carli, Ruggero; Bruschetta, Mattia; et al. (2021): Fast Solver for Implicit Continuous Set Model Predictive Control of Electric Drives. TechRxiv. Preprint. <https://doi.org/10.36227/techrxiv.15067401.v1>

DOI

[10.36227/techrxiv.15067401.v1](https://doi.org/10.36227/techrxiv.15067401.v1)

Fast Solver for Implicit Continuous Set Model Predictive Control of Electric Drives

Andrea Favato, Paolo Gherardo Carlet, Francesco Toso, Riccardo Torchio, Mattia Bruschetta, Ruggero Carli, Silverio Bolognani, *IEEE Fellow*, and Jose Rodriguez, *IEEE Fellow*

Abstract—This paper proposes an effective solver for implicit Continuous Set Model Predictive Control for the current control loop of synchronous motor drives with input constraints, allowing for reaching the maximum voltage feasible set. The related quadratic programming problem requires an iterative solver to find the optimal solution. The real-time certification of the algorithm is of paramount importance to move the technology toward industrial-scale applications by the proposed solver. The total number of operations can be computed in the worst-case scenario, thus the maximum computational time is known a priori. The solver is deeply illustrated, showing its feasibility for real-time applications in the microseconds range by means of experimental tests. Promising results are obtained with respect to well known general purpose solvers.

Index Terms—Electric Drives, Model Predictive Control (MPC), Permanent Magnet Synchronous Motor (PMSM), Predictive control, Quadratic Programming (QP).

I. INTRODUCTION

Model Predictive Control (MPC) is an advanced optimization-based control strategy that is gaining more and more popularity in the power electronics framework [1], [2]. Increasing availability of computational power and enhancement of optimization strategies have made MPC suitable for fast dynamics, as electric drives.

In this framework, MPC has been mostly implemented in the Finite Set (FS) form, which applies in each control interval one of the basic inverter voltage vector, selected by the controller [3]–[5]. A second type of MPC refers to the Continuous Set (CS) technique, which requires the modulator for synthesizing the optimal voltage reference. However, it provides fixed switching frequency of the converter and it works efficiently with longer sampling intervals [6]. Several improvements on the computational performances of solvers adopted for handling the optimization problems underlying the MPC led to the establishment of this technology. The MPC problem can be formulated in explicit or implicit form, whose similarities and differences are discussed in [7]. The

explicit form considers the optimization problem as a piecewise linear function of the parameters and solves it offline [8]. However, the computational cost grows significantly with the parameter space [9], thus limiting its applicability for industrial applications. Implementation of explicit MPCs is proposed and analyzed in [10], [11]. The implicit form, instead, requires an online iterative algorithm for solving the QP problem. A significant reduction of the computational burden of CS-MPCs has been achieved by implementing the unconstrained version of the QP problem, where a closed-form expression of the CS-MPC solution exists. However, this approximation suppresses the MPC advantage of including feasible limits in the control law. This has spurred a renewed research interest towards constrained solutions.

The development of efficient QP solvers is now enabling the real-time implementation of CS-MPCs. Gradient methods [17], [21] and active set methods [15] are the two most widespread kind of solvers for the considered application. In Table I, a set of existing QP solvers and some of their principal features are reported. Software packages containing solvers for fast embedded optimization are available too, e.g., *acados* [22]. Moreover, solver-libraries designed for MPC problems have been presented, e.g., *MATMPC* [23].

The constrained CS-MPC for electric motor drive applications has gained interest over the last decade. Table II resumes some of the most relevant papers related to this topic. In [9], an active set solver has been successfully implemented on a Texas Instrument Digital Signal Processor for the CS-MPC torque control of a PMSM, proving that the technology is mature for industrial applications. The work is based on the results of [24], where the same authors provided the exact complexity certification of the Gofar-Ibnani algorithm [25]. Thus, in these works, an upper bound of the computation time in the worst-case scenario has been demonstrated. Moreover, in [20], a nonlinear CS-MPC is presented for the Synchronous Reluctance Motor (SyRM). The work proves the flexibility of the MPC framework in tackling the nonlinear flux-current characteristics of such motors. However, the QP solver adopted for the specific application is a general purpose one, which is effective at the price of an increased complexity. Nevertheless, the computational burden is of paramount importance for embedded applications, and the emerging need from recent research advances is to design purpose-built algorithms.

In this work, we propose an efficient and fast method

Andrea Favato, Paolo Gherardo Carlet, Francesco Toso, Riccardo Torchio and Silverio Bolognani are with the Department of Industrial Engineering, University of Padova, Italy, 35151 code, Italy.

Mattia Bruschetta and Ruggero Carli are with the Department of Information Engineering, University of Padova, 35151, Italy.

Jose Rodriguez is with the Faculty of Engineering, Universidad Andres Bello, Santiago 8370146, Chile

TABLE I
OVERVIEW OF SOME QP SOLVERS. THE ACRONYMS ARE: IP: INTERIOR POINT; AS: ACTIVE SET; ADMM: ALTERNATING METHOD OF MULTIPLIERS; FGM: FAST GRADIENT METHOD. NL: NON LINEAR; LQP: LINEAR QP.

Solver	References	Target	Methods	Licence
FORCES Pro	[12]	NL-MPC, very general QPs	IP, ADMM, FGM	proprietary
HPIPM	[13]	LQP	IP	2-clause BSD
ODYS	[9]	QP, Embedded MPC	AS	proprietary
OSQP	[14]	QP	ADMM	Apache 2.0 License
qpOASES	[15]	QP	AS	2-clause BSD

TABLE II
SURVEY OF MPCs FOR ELECTRIC DRIVE APPLICATIONS. THE ACRONYMS ARE: LTI: LINEAR TIME INVARIANT; LPV = LINEAR PARAMETER VARIANT; E-MPC: EXPLICIT-MPC; IM: INDUCTION MOTOR; IPM: INTERIOR PERMANENT MAGNET; SPM: SURFACE PERMANENT MAGNET;

References	Year	Controlled Variable	Model	Method	Constraints		Tested Motors	Platform
					Input	Output		
[11]	2009	Speed&Currents	LTI	E-MPC	✓	✓	SPM	dSPACE 1004
[16]	2012	Speed&Torque/Flux	LPV	E-MPC	✓	✓	IM-PMSM	Sharc ADSP 21062 + TMS320C6713
[17]	2013	Currents	LPV	FGM	✓		IPM	TMS320F240
[18]	2015	Currents	LTI	AS	✓	✓	SPM	F28335 Delfino
[19]	2019	Currents	LPV	AS	✓		IPM	dSPACE 1006
[9]	2021	Currents	LTI	AS	✓	✓	SPM	F28335 Delfino
[20]	2021	Currents	NL	AS	✓		SyRM	dSPACE 1007

for solving the specific QP problem arising from MPC implementations for applications where a power converter is adopted. In particular, the algorithm is presented for the current control of synchronous motors, where limited computational hardware is usually available. The MPC implementation adopts a linear time-invariant model of the dynamic and considers linear input voltages constraints.

The model is formulated in the dq rotating reference frame, making the voltage feasible set rotating synchronously with the rotor position. The proposed QP solver takes advantage of the specific shape of the feasible set and the cost function to achieve a computationally efficient formulation that allows for exploiting the maximum voltage deliverable by the converter, i.e., a hexagonal region in the stationary reference frame, centered in the origin. The proposed algorithm adopts the choice of $N = 3$ prediction steps and $N_u = 1$ control horizon length, which has been assessed as a good trade-off between accuracy and computational effort [9].

A complete description of the QP solver is presented, allowing an easier replication of the algorithm. Moreover, computational performances in the worst-case scenario are assessed both in terms of number of operations and in experimental conditions. The high accuracy of the algorithm is then assessed comparing with qpOASES.

The paper is organized as follows: Section II presents the adopted model of the motor, which has been proved to be effective for this application [18], [26], [27]. In Section III the MPC problem with input constraints is stated. The novel proposed algorithm is described in Section IV. Experiments and simulations are provided in Section V and Section VI, respectively, proving the real-time feasibility and the effec-

tiveness of the solver on a SyRM.

II. MATHEMATICAL MODEL

The electrical dynamic equations of the synchronous motor is described with respect to the rotating dq -reference frame as follows (for simplicity we omit the time-dependence) [26]:

$$\frac{d}{dt}i_{dq} = A_c i_{dq} + B_c (u_{dq} + w_{dq}), \quad (1)$$

where $i_{dq} = [i_d \ i_q]^T$, $u_{dq} = [u_d \ u_q]^T$, $w_{dq} = [w_d \ w_q]^T$ and

$$A_c = \begin{bmatrix} -\frac{R_s}{L_d} & \omega_e \frac{L_q}{L_d} \\ -\omega_e \frac{L_d}{L_q} & -\frac{R_s}{L_q} \end{bmatrix}, \quad B_c = \begin{bmatrix} \frac{1}{L_d} & 0 \\ 0 & \frac{1}{L_q} \end{bmatrix}, \quad (2)$$

where u_d , u_q , i_d and i_q are the dq -axis voltages and currents, respectively, R_s is the windings resistance, ω_e the electrical speed, whereas L_d and L_q are the dq -axis stator inductances. Assuming perfect knowledge of the parameters and neglecting model inaccuracies, the vector w_{dq} includes only the Back-EMF (B-EMF), i.e., $w_{dq} = [0 \ -\omega_e \lambda_{pm}]^T$, being λ_{pm} the permanent magnet flux linkage.

Discretizing model (1) by Euler integration with sampling time T_s , yields to the following discrete-time model

$$x(k+1) = Ax(k) + Bu(k) + Bw(k), \quad (3)$$

where $B = T_s B_c$, $A = I + T_s A_c$, being I the identity matrix, and where $x = [i_d \ i_q]^T$ denotes the state and the subscript on $u(k)$ and $w(k)$ are neglected for simplicity.

In practice, significant unmodeled disturbances and non-linear dynamics are present, due, e.g., to iron saturation, parasitic effects and harmonic modes introduced by the non-ideal rotor geometry, and to the speed dependence of matrix A_c , which has been obtained by linearizing around the nominal speed. To address this issue in real experiments an observer is implemented to obtain an estimate of both $x(k)$ and $w(k)$ for all k , thus improving the tracking performance of the controller and guaranteeing offset free tracking. More precisely, the state of the model (3) is augmented with the disturbance w , and it is estimated via a Kalman Filter as proposed in [28].

III. MODEL PREDICTIVE CONTROL OF PMSM

The LTI model (3) is used as the prediction model in the MPC problem, whose ultimate goal is to track the desired currents profiles $x^{ref} = [i_d^{ref} \ i_q^{ref}]^T$.

The following quadratic functional cost is adopted

$$J = \sum_{j=0}^{N-1} (\|x^{ref}(k+j+1) - x(k+j+1)\|_Q^2) + \sum_{i=0}^{N_u-1} (\|\Delta u(k+i)\|_R^2) + \|x^{ref}(k+N) - x(k+N)\|_S^2 \quad (4)$$

where Q , S and R are weighting matrices and $\Delta u(k)$ denotes the differential input, i.e., $\Delta u(k) = u(k) - u(k-1)$.

At each time step k , the optimal control move is obtained by solving the following optimal control problem:

$$\begin{aligned} \min_{\Delta u, x} \quad & J \\ \text{s.t.} \quad & x(k+1) = Ax(k) + Bu(k) + Bw \\ & \Delta u(k) = u(k) - u(k-1) \\ & u_{\alpha\beta}(k+j) = T(\theta_e)u(k+j) \in \mathbb{U}_{\alpha\beta}, j = 0, 1, \dots, N_u \end{aligned} \quad (5)$$

where $T(\theta_e)$ is the Park anti-transformation matrix, being θ_e the electric angle, $\mathbb{U}_{\alpha\beta}$ represents the feasible voltage region for $u_{\alpha\beta} = [u_\alpha \ u_\beta]^T$, and where the term w is assumed to be constant in the prediction horizon and equal to the estimate $\hat{w}(k-1)$ provided by the Kalman filter at the previous step.

Observe that $\mathbb{U}_{\alpha\beta}$ is the maximum voltage deliverable by the converter, i.e., a hexagonal region in the stationary reference frame, centered in the origin. Effective methods to manage the transition from sinusoidal output voltage with a linear voltage gain characteristic to six-step operation can be implemented [29]. A feasible solution verifies the following inequalities:

$$\begin{aligned} F\Delta u(k) &\leq f \\ F &= \begin{bmatrix} 1 & 1 & 1 & -1 & -1 & -1 \\ \sqrt{3} & 0 & -\sqrt{3} & -\sqrt{3} & 0 & \sqrt{3} \end{bmatrix}^T T(\theta_e), \\ f &= \frac{2u_{DC}}{\sqrt{3}} \begin{bmatrix} 1 & 0.5 & 1 & 1 & 0.5 & 1 \end{bmatrix}^T - Fu(k-1) \end{aligned} \quad (6)$$

Assuming, as anticipated, that $N = 3$ and $N_u = 1$ (and so it will be hereafter) the condensed problem of (5) results in

$$\begin{aligned} \min_{\Delta u(k)} \quad & J(\Delta u(k)) := \frac{1}{2} \Delta u(k)^T H \Delta u(k) + c^T \Delta u(k) + \text{const}, \\ \text{s.t.} \quad & F\Delta u(k) \leq f \end{aligned} \quad (7)$$

where $H \in \mathbb{R}^{2 \times 2}$, $c \in \mathbb{R}^{2 \times 1}$ and where $F\Delta u(k) \leq f$ defines a hexagon, which is the rotated and translated version of $\mathbb{U}_{\alpha\beta}$ according to the transformation $T(\theta_e)$ and $u(k-1)$. Clearly the region $F\Delta u(k) \leq f$ is composed of six segments lying on six lines that, hereafter, we denote as ℓ_i for $i = 1, \dots, 6$, being ℓ_i associated to the constraint $F(i, :)\Delta u(k) \leq f(i)$. Now, let Δu^* be the optimal solution of (7), then the applied control input at time k is

$$u^*(k) = u^*(k-1) + \Delta u^*(k).$$

In the $\alpha\beta$ -plane the optimal control input is then $u_{\alpha\beta}(k) = T(\theta_e)u(k)$.

We conclude this section by considering a particular scenario that will play an important role in the proposed algorithm, namely the case where only one constraint is imposed, e.g., the constraint associated with the line ℓ_{i^*} . The optimal control problem to be solved is

$$\min_{\Delta u(k)} J(\Delta u(k)) \quad \text{s.t.} \quad F(i^*, :)\Delta u(k) \leq f(i^*)$$

and the solution is computed by solving the almost equivalent unconstrained problem

$$\min_{\Delta u(k)} J(\Delta u(k)) + \lambda_{i^*} (F(i^*, :)\Delta u - f(i^*))^2 \quad (8)$$

for λ_{i^*} sufficiently high to guarantee a negligible distance from the constraint. Basically, in (8) the functional cost J has been augmented with a penalty function forcing the solution to stay on the active constraint. Note that the problem in (8) becomes unconstrained, therefore it admits an analytic solution, as will be seen in the next section.

IV. ALGORITHM DESCRIPTION

In this section, the QP solver algorithm is described. First, the initial assumptions are listed, then, the algorithm is discussed, considering different cases that may occur during the search of the solution. The solver is based on the idea of solving a constrained QP problem with the constraints described in (6), with a finite succession of iterations, each one being analytical solutions of unconstrained problems. In particular, at the beginning, the optimal solution of the unconstrained version of problem (7) is computed in closed form as

$$\Delta u^{uc} = -H^{-1}c, \quad (9)$$

assessing the maximum number of possible active constraints, that depends on the relative position of the hexagon defining the voltage limits with respect to the unconstrained optimal solution. Four different situations can then occur:

- 0) The solution is optimal and feasible, i.e., the optimal unconstrained solution lies within the feasible voltage set, and it is applied as reference for the inverter;

Fig. 1. One constraint violation: the solution is found by adding the penalty function represented by the red line to the cost.

- 1) one constraint is not satisfied (see for example Fig. 1);
- 2) two consecutive constraints are not satisfied (see Fig. 2);
- 3) three consecutive constraints are not satisfied (see, e.g., Fig. 3).

Notice that, in the figures, the unconstrained solution

$$u_{\alpha\beta}^* = T(\theta_e) (u(k-1) + \Delta u^*)$$

and the respective feasible set $\mathbb{U}_{\alpha\beta}$ are represented in the $\alpha\beta$ -plane. As mentioned before, a simple roto-translation allows for mapping the dq -plane into the $\alpha\beta$ one, making the representations equivalent in qualitative terms. In order to better clarify the last three points, we propose a deepening on the shape of the function cost and the relative positioning of the hexagonal constraint.

A. Regions of Violated Constraints

In this section we expressed the functional cost J in terms of $u_{\alpha\beta}$. It turns out that the level set curves are centered in $u_{\alpha\beta}^{uc}$, where the eccentricity depends on the ratio of the dq inductances and the orientation depends on the electric angle. Fig. 4 shows two examples of cost function shapes in the $\alpha\beta$ -plane, where Fig. 4(a) represents the case of an IPM motor, while Fig. 4(b) shows the case of an SPM motor. In both cases, a representative hexagonal feasible voltage set in the $\alpha\beta$ -plane induced by (6) is reported. Notice that, due to the magnetic isotropic typical of an SPM, the level curves become circumferences. Consequently, depending on the type of motor, a priori knowledge of the cost function shape can be obtained.

By analyzing the six inequalities associated with the rows of F and f in (6) and relying on the convexity of the level curves of the cost function, it is possible to relate the relative position of the unconstrained optimal solution with respect to the hexagonal feasible voltage set with the number of violated constraints. In Fig. 5, a graphical representation is shown, where each region is identified by a number from 0 to 3, which indicates the number of violated constraints of an optimal solution located in the corresponding region. In the following, for each of these cases, a tailored solution strategy is described.

Fig. 2. Two constraint violations. Fig. 2(b) represents the case 2c), where the one constraint violation procedure is followed, and the solution is on an intersection.

Fig. 3. Three constraint violations.

B. Algorithm Steps

A detailed description of the steps of the algorithm is proposed below, categorized according to the relative location of the unconstrained solution Δu^{uc} with respect to the cost function (see Fig. 5).

1) *One Constraint Violation*: If Δu^{uc} lies within the triangle adjacent to the segment of the hexagon lying on the line ℓ_i (see Fig. 5), the feasible optimal solution certainly lies on the segment itself. A representative example is shown in Fig. 1, where solutions are plotted, as before, in the $\alpha\beta$ -plane. The feasible solution is then computed according to the following steps:

- 1) The solution of the problem (8) is computed considering as constraint ℓ_i^1 , i.e., with $i^* = i$, namely Δu^{oa} ;
- 2) a feasibility check is then operated, computing

$$h_j := F(j, :)\Delta u^{oa} - f(j) \leq 0 \text{ for } j = i-1, i+1; \quad (10)$$

¹To simplify the description of the algorithm we refer to constraint ℓ_i as the constraint lying on line ℓ_i .

Fig. 4. Cost function contours in $\alpha\beta$ -plane for anisotropic and isotropic motor. The hexagon represents the feasible voltage set, while the center of the contours is the unconstrained solution.

Fig. 5. Portions of $\alpha\beta$ -plane identified by the hexagon lines identified by the number of input constraints that are overcome simultaneously by the unconstrained optimal solution which lies within that region.

- if Δu^{oa} does not violate any constraint, then $\Delta u^* = \Delta u^{oa}$ and lies on the segment;
- if Δu^{oa} violates the next constraint² ℓ_{i+1} , i.e., if $h_{i+1} > 0$, thus Δu^* is the intersection between ℓ_i with ℓ_{i+1} ;
- if $h_{i-1} > 0$, Δu^* is the intersection between ℓ_i with ℓ_{i-1} .

2) *Two Constraint Violations*: The case where the unbounded solution lies in a region that crosses two consecutive boundaries of the feasible set is illustrated in Fig. 2. In this case, Δu^* lies on one of two *consecutive* constraints ℓ_i, ℓ_{i+1} , for some i in the range $[1, 6]$. More precisely, depending on the location of the unconstrained solution and the cost function shape, three different situations can occur:

- the feasible solution lies within one of the hexagon segments that lies either on ℓ_i or on ℓ_{i+1} (see Fig. 2(a));
- the feasible solution lies on the intersection of ℓ_i and ℓ_{i+1} ;
- the feasible solution lies on one of the extremal vertexes of one of the hexagon segments, i.e., those that are not originated by the intersection of ℓ_i and ℓ_{i+1} (see Fig. 2(b)).

²The constraints are cycled through with a counterclockwise direction in a circular buffer fashion, i.e., the subscript $i + n = \text{mod}(i - 1 + n, 6) + 1$ for any integer n where mod is the modulo operator.

The feasible solution is then computed according to the following steps:

- The solution to the problem (8) is computed considering as constraint ℓ_i , i.e., with $i^* = i$, namely Δu^{oa} ;
- a feasibility check is then operated, computing h_j as in (10) for $j = i + 1, i - 1$;
 - if Δu^{oa} does not violate any constraint, then $\Delta u^* = \Delta u^{oa}$;
 - if $h_{i+1} > 0$, thus Δu^* lies on ℓ_{i+1} itself and the *one constraint violation* procedure can be applied (Fig. 2(a));
 - if $h_{i-1} > 0$, Δu^* is intersection between ℓ_i with ℓ_{i-1} (Fig. 2(b)).

3) *Three Constraints Violations*: The last possible condition is reported in Fig. 3, where the unconstrained solution detects three violated boundaries in which the optimal feasible solution can lie. With the same notation adopted in the previous section, the three consecutive constraints are denoted by ℓ_i, ℓ_{i+1} and ℓ_{i+2} for some i in the range $[1, 6]$. Three different situations can then occur:

- the feasible solution lies within one of the hexagon segments that lies or on ℓ_i or on ℓ_{i+1} or on ℓ_{i+2} ;
- the feasible solution lies on the intersections of ℓ_i and ℓ_{i+1} and ℓ_{i+2} ;
- the feasible solution lies on one of the extremal vertexes of one of the hexagon segments, i.e., those that are not originated by the intersection of ℓ_i and ℓ_{i+1} and ℓ_{i+2} .

The feasible solution is then computed according to the following steps:

- The solution to the problem (8) is computed considering as constraint ℓ_{i+i} , i.e., with $i^* = i + 1$ (the central violated constraint), namely Δu^{oa} ;
- A feasibility check is then operated, computing h_j as in (10) for $j = i + 2, i$;
 - if Δu^{oa} does not violate any constraint, then $\Delta u^* = \Delta u^{oa}$ (Fig. 3(a));
 - if $h_{i+2} > 0$ (or $h_i > 0$), Δu^* certainly lies on ℓ_{i+2} (or ℓ_i respectively) and the *one constraint violation* procedure can be applied; Fig. 3(b), Fig. 3(c) and Fig. 3(d) summarized the possible situations.

V. COMPUTATIONAL ANALYSIS

In the previous section we described an algorithm to solve the constrained QP (5) that exploits the cost function shape to provide a solution in a fixed number of operations. The procedure results to be accurate and efficient in terms of computational burden. The computational performance of the algorithm is hereafter compared with those of the open-source tool qpOASES.

A. Comparison with qpOASES

In this section, we show that the proposed method achieves competitive performances and requires a low computational effort. qpOASES is an open source software, it is very robust and suitable for medium-small scale problems, and it can be

TABLE III
OVERVIEW OF THE SYRM MOTOR AND CONTROL PARAMETERS.

Motor Data	Symbol	Value
Pole pairs	p	2
Phase resistance	R	$1\ \Omega$
d-axis inductance	L_d	$0.2\ \text{H}$
q-axis inductance	L_q	$0.06\ \text{H}$
Nominal current	I_N	6 A peak
Nominal d current	$I_{N,d}$	3 A
Nominal q current	$I_{N,q}$	5.2 A
Nominal speed	Ω_N	700 rpm
Prediction Horizon	N	3
Control Horizon	N_u	1
Input Weight d-axis	r_d	0.0001
Input Weight q-axis	r_q	0.0002
Prediction Error Weight	q_s	1

TABLE IV
AVERAGED TIME COMPARISON BETWEEN THE PROPOSED METHOD AND qpOASES FOR DIFFERENT NUMBER OF STEPS.

	Avg Time (μs)	
	Proposed Method	qpOASES
Step 0	1.6	28.2
Step 1	1.65	30.4
Step 1.5	1.67	30.4
Step 2	1.7	30.0
Step 2.5	1.72	30.0

considered as a very good benchmark in terms of solution accuracy. The MPC problem is built considering the motor under test (Table III). The averaged time required by the proposed method is compared with the one of the open source solver qpOASES in MATLAB environment. The solvers were run on a Intel(R) Core(TM) i7-8700 CPU 3.20GHz and they were tested considering all the possible cases presented in Section IV. The problems were run one million times, averaging the time spent for the computation.

The numerical results are presented in Table IV. The following values have been chosen to distinguish among the cases:

- 0 if the unconstrained solution is feasible;
- 1 the algorithm performs the one constraint violation procedure, and the new solution is feasible (step 2 a) in Section IV-B);
- 1.5 if the one constraint violation indicates a solution found in the intersection (step 2 b) or c));
- 2 refers to the case where two of (8) are solved.
- finally, 2.5 corresponds to the worst-case scenario where, in the two and three constraint violations, the solutions is on a vertex after solving (8) twice.

As can be noticed, the proposed solver finds the MPC solution with cheaper computational cost with respect to qpOASES. We highlight that, for the worst-case scenario, where the solution violates three constraints and the optimal one is on a hexagon vertex, the average computation time remains very limited. Thus, the execution time of this case is considered the upper bound for the control algorithm cost.

TABLE V
TOTAL NUMBER OF ALGEBRAIC OPERATION OF THE PROPOSED QP METHOD IN THE WORST-CASE SCENARIO.

PHASE	SUM./SUB.	PRODUCTS	DIVISIONS
1)	7	10	2
2)	15	12	-
3)	15	28	2
4)	15	12	-
5)	15	28	2
6)	15	12	-
7)	-	-	-
TOTAL	82	102	6

Fig. 6. Test-bed layout.

B. Worst-Case Computational Cost

Considering the relevance of real-time feasibility certification for algorithms that aspire to large-scale industrial applications, we propose a worst-case performance analysis. Since the computational performances strongly rely on the specific implementation, we propose a quantification of the computational cost required by the proposed method with the total number of algebraic operations performed to find the solution in the worst-case scenario. The sequence of operations is listed below:

- 1) Compute the unconstrained solution; (9);
- 2) feasibility check (10) $\Rightarrow$ solution unfeasible ;
- 3) find new solution (8);
- 4) feasibility check (10) ; $\Rightarrow$ solution unfeasible;
- 5) find new solution (8) ;
- 6) feasibility check (10); $\Rightarrow$ solution unfeasible;
- 7) find the solution among the vertices.

The corresponding number of operations is reported in Table V. It is worth noting that step 7) does not require any additional algebraic operations. In conclusion, the peculiarities of the proposed method are highlighted: the maximum number of steps are fixed, hence, the total number of operations can be evaluated and the maximum run time can be easily determined depending on the specific hardware.

VI. EXPERIMENTAL RESULTS

The test bench adopted for the experiments is shown in Fig. 6. It consists of a master motor (surface-mounted PMSM) directly connected to the SyRM motor under test.

Fig. 7. Comparison between qpOASES and the proposed method when an unfeasible working point is set. The subscript 1 in Fig. 7(a) refers to qpOASES, while the other one to the proposed method.

The controllers have been implemented in a dSPACE MicroLabBox, a compact development system for laboratory purposes which has dual-core real-time processor at 2 GHz and dedicated electric motor control features, which provides the gate signals for the inverter and performs all the acquisitions from current and position sensors.

A. Solution Accuracy

First, a reduced value of the DC bus was set for the controllers in order to achieve an unfeasible working point, where the solvers are pushed to find solutions along the hexagon edges. During the test qpOASES was used as solver and it is considered as the benchmark. Then, the recorded data were used to build the MPC problem offline and the voltage solutions were computed with the proposed method for each time-step. This experiment is shown Fig. 7. The voltages are plotted in per unit with respect the average value of the DC bus. The solutions found by qpOASES and our solver overlap, as confirmed by the voltage reference in Fig. 7(a). In particular, some test points are plotted in Fig. 7(b) in the $\alpha\beta$ -plane for the selected time snapshots indicated in Fig. 7(a). This confirms that the solutions are found at limits and they are numerically identical.

B. Solver Performance

To evaluate the solver performance in terms of computational cost we propose a current step transient in Fig. 8, where the master motor keeps the motor under test at its nominal speed, and the nominal current value is set as reference. This condition represents one of the most stressing test for evaluating current controller performance. Thus, it is expected that this represents the worst-case scenario for the computational cost of the algorithms. Fig. 8(a) and Fig. 8(c) show the dq currents transients, normalized with respect to the nominal peak current value. The performance are very similar for the two methods, with good tracking and steady-state accuracy. The most interesting part of this test can be observed in Fig. 8(b) and Fig. 8(d), where the turnaround time of the algorithms is measured by dSPACE. The time required by measurements acquisitions and elaborations is about 8 μ s. As an interesting result, it can be observed that, while the proposed solver maintains a good accuracy in tracking the

Fig. 8. Transient test: Comparison between qpOASES and the proposed method at nominal currents and speed. In Fig. 8(e) the flag values are described in Section V-A. The average time for measurements acquisition and elaboration is about 8 μ s.

currents, the good computational performance presented in Table V are replicated in experimental environment. The former shows a peak at the first instant with 14 μ s, while the average time stabilizes at 12 μ s. In Fig. 8(e) is also reported a flag value which indicates at which step the proposed solver finds the optimal solution. Thus, in the worst-case situation, the proposed solver needs 14 μ s to find the optimal voltages. Finally, the voltage solutions are compared in Fig. 8(f), which confirms that solutions are found on the limits during this transient.

As a last confirmation of the averaged time required by the solvers (i.e., the proposed algorithm and qpOASES) to perform all the computations, we report the turnaround time during the test of Fig. 7 in Fig. 9, where an unfeasible working point is considered. We highlight that the time required by qpOASES is inevitably higher, because the generality of the considered solver increases the required time to solve the problem.

VII. CONCLUSIONS

In this work, an implicit model predictive control (MPC) strategy for the current control loop of synchronous motor

Fig. 9. Turnaround time of the QP algorithms considering an unfeasible working point.

drives has been presented, including the input voltage feasible set. A fast and effective method for solving in real-time the quadratic programming problem related to the MPC has been proposed. The number of steps in the worst-case is fixed, thus the number of operations can be assessed in advance. One of the main feature of the proposed method is that real-time certification can be achieved. Experimental results have been performed by using the open source solver qpOASES as benchmark for comparing the optimal solution. The proposed solver has shown very good accuracy and, considering the worst-case scenario, the optimal voltage is found in few microseconds, making it promising for its implementation in large-scale, real-time industrial applications.

REFERENCES

- [1] P. Cortes, M. P. Kazmierkowski, R. M. Kennel, D. E. Quevedo, and J. Rodriguez, "Predictive control in power electronics and drives," *IEEE Trans. on Ind. Electron.*, vol. 55, DOI 10.1109/TIE.2008.2007480, no. 12, pp. 4312–4324, Dec. 2008.
- [2] S. Vazquez, J. Rodriguez, M. Rivera, L. G. Franquelo, and M. Norambuena, "Model predictive control for power converters and drives: Advances and trends," *IEEE Trans. on Ind. Electron.*, vol. 64, DOI 10.1109/TIE.2016.2625238, no. 2, pp. 935–947, 2017.
- [3] S. Kouro, P. Cortes, R. Vargas, U. Ammann, and J. Rodriguez, "Model predictive control—a simple and powerful method to control power converters," *IEEE Trans. on Ind. Electron.*, vol. 56, DOI 10.1109/TIE.2008.2008349, no. 6, pp. 1826–1838, Jun. 2009.
- [4] T. Geyer and D. E. Quevedo, "Performance of multistep finite control set model predictive control for power electronics," *IEEE Trans. Power Electron.*, vol. 30, DOI 10.1109/TPEL.2014.2316173, no. 3, pp. 1633–1644, 2015.
- [5] F. Wang, X. Mei, J. Rodriguez, and R. Kennel, "Model predictive control for electrical drive systems—an overview," *CES Trans. on Electr. Machines and Systems*, vol. 1, DOI 10.23919/TEMS.2017.8086100, no. 3, pp. 219–230, Sep. 2017.
- [6] 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, DOI 10.1109/TII.2017.2758393, no. 4, pp. 1334–1346, 2018.
- [7] M. Kvasnica, "Implicit vs explicit mpc — similarities, differences, and a path towards a unified method," in *2016 Europ. Control Conf. (ECC)*, DOI 10.1109/ECC.2016.7810353, pp. 603–603, Jun. 2016.
- [8] A. Bemporad, M. Morari, V. Dua, and E. N. Pistikopoulos, "The explicit linear quadratic regulator for constrained systems," *Automatica*, vol. 38, DOI [https://doi.org/10.1016/S0005-1098\(01\)00174-1](https://doi.org/10.1016/S0005-1098(01)00174-1), no. 1, pp. 3–20, 2002.
- [9] G. Cimini, D. Bernardini, S. Levijoki, and A. Bemporad, "Embedded model predictive control with certified real-time optimization for synchronous motors," *IEEE Trans. Control Syst. Technol.*, DOI 10.1109/TCST.2020.2977295, pp. 1–8, 2020.
- [10] S. Mariethoz, A. Domahidi, and M. Morari, "Sensorless explicit model predictive control of permanent magnet synchronous motors," in *2009 IEEE Int. Electr. Machines and Drives Conf.*, pp. 1250–1257, May. 2009.
- [11] 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, DOI 10.1109/TIE.2008.2007547, no. 6, pp. 1925–1936, Jun. 2009.
- [12] A. Zanelli, A. Domahidi, J. Jerez, and M. Morari, "Forces nlp: an efficient implementation of interior-point methods for multistage nonlinear nonconvex programs," *International Journal of Control*, vol. 93, DOI 10.1080/00207179.2017.1316017, no. 1, pp. 13–29, 2020.
- [13] G. Frison and M. Diehl, "Hpipm: a high-performance quadratic programming framework for model predictive control," 2020.
- [14] B. Stellato, G. Banjac, P. Goulart, A. Bemporad, and S. Boyd, "OSQP: an operator splitting solver for quadratic programs," *Mathematical Programming Computation*, vol. 12, DOI 10.1007/s12532-020-00179-2, no. 4, pp. 637–672, 2020.
- [15] H. J. Ferreau, C. Kirches, A. Potschka, H. G. Bock, and M. Diehl, "qpOASES: a parametric active-set algorithm for quadratic programming," *Mathematical Programming Computation*, vol. 6, DOI <https://doi.org/10.1007/s12532-014-0071-1>, no. 1, pp. 327–363, 2014.
- [16] S. Mariethoz, A. Domahidi, and M. Morari, "High-bandwidth explicit model predictive control of electrical drives," *IEEE Trans. Ind. Appl.*, vol. 48, DOI 10.1109/TIA.2012.2226198, no. 6, pp. 1980–1992, 2012.
- [17] M. Preindl, S. Bolognani, and C. Danielson, "Model predictive torque control with pwm using fast gradient method," in *2013 Twenty-Eighth Annual IEEE Applied Power Electronics Conf. and Exp. (APEC)*, DOI 10.1109/APEC.2013.6520661, pp. 2590–2597, 2013.
- [18] G. Cimini, D. Bernardini, A. Bemporad, and S. Levijoki, "On-line model predictive torque control for permanent magnet synchronous motors," in *2015 IEEE Int. Conf. on Ind. Tech. (ICIT)*, DOI 10.1109/ICIT.2015.7125438, pp. 2308–2313, 2015.
- [19] S. Hanke, O. Wallscheid, and J. Böcker, "Continuous-control-set model predictive control with integrated modulator in permanent magnet synchronous motor applications," in *2019 IEEE Int. Machines Drives Conf. (IEMDC)*, DOI 10.1109/IEMDC.2019.8785122, pp. 2210–2216, 2019.
- [20] A. Zanelli, J. Kullick, H. M. Eldeeb, G. Frison, C. M. Hackl, and M. Diehl, "Continuous control set nonlinear model predictive control of reluctance synchronous machines," *IEEE Trans. Control Syst. Technol.*, DOI 10.1109/TCST.2020.3043956, pp. 1–12, 2021.
- [21] S. Richter, C. N. Jones, and M. Morari, "Computational complexity certification for real-time mpc with input constraints based on the fast gradient method," *IEEE Trans. Autom. Control*, vol. 57, DOI 10.1109/TAC.2011.2176389, no. 6, pp. 1391–1403, Jun. 2012.
- [22] R. Verschuere, G. Frison, D. Kouzoupis, J. Frey, N. van Duijkeren, A. Zanelli, B. Novoselnik, T. Albin, R. Quirynen, and M. Diehl, "acados: a modular open-source framework for fast embedded optimal control," *arXiv*, 2020.
- [23] Y. Chen, M. Bruschetta, E. Picotti, and A. Beghi, "Matmpc - a matlab based toolbox for real-time nonlinear model predictive control," in *2019 18th Europ. Control Conf. (ECC)*, DOI 10.23919/ECC.2019.8795788, pp. 3365–3370, 2019.
- [24] G. Cimini and A. Bemporad, "Exact complexity certification of active-set methods for quadratic programming," *IEEE Transactions on Automatic Control*, vol. 62, pp. 6094–6109, 2017.
- [25] D. Goldfarb and A. Idnani, "A numerically stable dual method for solving strictly convex quadratic programs," *Math. Program.*, vol. 27, DOI 10.1007/BF02591962, no. 1, p. 1–33, Sep. 1983.
- [26] F. Toso, P. G. Carlet, A. Favato, and S. Bolognani, "On-line continuous control set mpc for pmsm drives current loops at high sampling rate using qpOASES," in *2019 IEEE Energy Conv. Congr. and Exp. (ECCE)*, DOI 10.1109/ECCE.2019.8912838, pp. 6615–6620, 2019.
- [27] X. Sun, M. Wu, G. Lei, Y. Guo, and J. Zhu, "An improved model predictive current control for pmsm drives based on current track circle," *IEEE Trans. on Ind. Electron.*, vol. 68, DOI 10.1109/TIE.2020.2984433, no. 5, pp. 3782–3793, 2021.
- [28] A. Favato, P. G. Carlet, F. Toso, R. Torchio, and S. Bolognani, "Integral model predictive current control for synchronous motor drives," *IEEE Trans. Power Electron.*, DOI 10.1109/TPEL.2021.3081827, pp. 1–1, 2021.
- [29] S. Bolognani and M. Zigliotto, "Novel digital continuous control of svm inverters in the overmodulation range," *IEEE Trans. Ind. Appl.*, vol. 33, DOI 10.1109/28.568019, no. 2, pp. 525–530, 1997.