IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Date of publication xxxx 00, 0000, date of current version xxxx 00, 0000.

Digital Object Identifier 10.1109/ACCESS.2017.DOI

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

ANDREA FAVATO[1], PAOLO GHERARDO CARLET[1], FRANCESCO TOSO[1], RICCARDO TORCHIO[1](Member, IEEE), LUDOVICO ORTOMBINA[1](Member, IEEE), MATTIA BRUSCHETTA[2], RUGGERO CARLI[2], PIERGIORGIO ALOTTO[1](Senior Member, IEEE), SILVERIO BOLOGNANI[1](Fellow, IEEE), JOSE RODRIGUEZ[3](Fellow, IEEE)

[1]Department of Industrial Engineering, University of Padova, Italy, 35151, Italy.
[2]Department of Information Engineering, University of Padova, 35151, Italy.
[3]Faculty of Engineering, Universidad Andres Bello, Santiago 8370146, Chile.

Corresponding author: Riccardo Torchio (e-mail: riccardo.torchio@unipd.it).

ABSTRACT This paper proposes a fast and accurate 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 control 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. A relevant feature of the proposed solver is that 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. The proposed method outperforms general-purpose algorithms in terms of computation time, while keeping the same accuracy.

INDEX TERMS Electric Drives, Model Predictive Control (MPC), Permanent Magnet Synchronous Motor (PMSM), Synchronous Reluctance Motor (SyRM), 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 [3] and enhancement of optimization strategies have made advanced control schemes, such as MPC and Digital Twins [4], [5], suitable for fast dynamic systems, as electric drives.

In this framework, MPC has been mostly implemented in its Finite Set (FS) form where, in each control interval, the controller applies one of the basic inverter voltage vector [6]–[10]. A second type of MPC refers to the Continuous Set (CS) technique, which requires the modulator for synthesizing the optimal voltage reference. It provides a fixed switching frequency of the converter and it works efficiently with longer sampling intervals [11] with a significant increase in computational load.

Neglecting unconstrained solutions, which nullify the advantage in using the MPC strategy, it is important to mention the attempts to use explicit MPC in the electric drives field [14], [17], [23]. However, they have shown not to allow a substantial reduction in computational cost as the parameter space increases. Instead, we are now observing a growing interest around implementations of implicit CS-MPCs due to the recent development of efficient Quadratic Programming QP solvers, leading to the establishment of this technique also in industrial contexts. Gradient methods [19], [24] and active set methods [16] are the two most widespread kind of solvers for the considered application. In Table 1, 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` [25]. Moreover, solver-libraries designed for MPC problems have been presented, e.g., `MATMPC` [26].

The constrained CS-MPC for electric motor drives has gained interest in electric drives applications, too. Table 2 resumes some of the most relevant papers related to this topic. In [14], an active set solver has been

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2022.3150283, IEEE Access

TABLE 1: 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	[14]	QP, Embedded MPC	AS	proprietary
OSQP	[15]	QP	ADMM	Apache 2.0 License
qpOASES	[16]	QP	AS	2-clause BSD

TABLE 2: 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		
[17]	2009	Speed&Currents	LTI	E-MPC	✓	✓	SPM	dSPACE 1004
[18]	2012	Speed+Torque/Flux	LPV	E-MPC	✓	✓	IM-PMSM	Sharc ADSP 21062 + TMS320C6713
[19]	2013	Currents	LPV	FGM	✓		IPM	TMS320F240
[20]	2015	Currents	LTI	AS	✓	✓	SPM	F28335 Delfino
[21]	2019	Currents	LPV	AS	✓		IPM	dSPACE 1006
[14]	2021	Currents	LTI	AS	✓	✓	SPM	F28335 Delfino
[22]	2021	Currents	NL	AS	✓		SyRM	dSPACE 1007

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 was based on the results of [27], where the same authors provided the exact complexity certification of the Golfarb-Idnani algorithm [28]. In these papers, an upper bound of the computation time in the worst-case scenario has been demonstrated. Moreover, a nonlinear CS-MPC was presented for the Synchronous Reluctance Motor (SyRM) in [22]. The work proved the flexibility of the MPC framework in tackling the nonlinear flux-current characteristics of such motors. The adopted QP solver for the specific application was 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 accurate and fast method 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 (LTI) model of the plant and considers linear input voltages constraints. The motor model is formulated in the dq rotating reference frame, making the feasible voltage 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. It 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 [14].

A complete description of the QP solver is presented, allowing an easier replication of the algorithm. Moreover, the computational burden 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 [16].

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 [20], [29], [30]. 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 effectiveness of the solver on a SyRM.

II. MATHEMATICAL MODEL

The voltage equations of a synchronous motor in the rotating dq-reference frame are described as follows (for simplicity we omit the time-dependence) [29]:

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

where

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

and $i_{dq} = [i_d \; i_q]^{\mathrm{T}}$, $u_{dq} = [u_d \; u_q]^{\mathrm{T}}$, $w_{dq} = [w_d \; w_q]^{\mathrm{T}}$. The 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-Electromotive Force (B-EMF), i.e., $w_{dq} = [0 \; -\omega_e \lambda_{pm}]^{\mathrm{T}}$, being λ_{pm} the permanent magnet flux linkage. In practice, significant unmodeled disturbances and nonlinear dynamics are present [31], due to, e.g., iron saturation, parasitic effects and harmonic modes introduced by the non-ideal rotor geometry. Moreover, the matrix A_c was obtained by linearizing it around the nominal speed, and its speed dependency was lost. This choice permits to build the MPC problem offline, thus simplifying the calculations. If desired, the speed dependence could be included in the motor modeling, at price of more online computations. To account for all the disturbances, in real experiments an observer (e.g. a Kalman filter [32]) is implemented to obtain an estimate of both i_{dq} and w_{dq}, thus improving the tracking performance of the controller and guaranteeing offset free tracking.

Since MPC is a dicreate-time controller, the continuous-time synchronous motor model (1) is discretized by the Euler integration with the sampling period T_s, it 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 $x = i_{dq}$ denotes the system state and the subscripts on $u(k)$ and $w(k)$ are omitted to ease the mathematical notation.

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 $x^{ref} = [i_d^{ref} \; i_q^{ref}]^{\mathrm{T}}$. The MPC optimization problem is solved at each control step k with respect to the dq voltage increment $\Delta u(k) = u(k) - u(k-1)$. The following quadratic functional cost is adopted:

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

where Q, S and R are weighting matrices. At each time step k, the optimal control move is obtained by solving the following optimal control problem:

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

where $T(\theta_e)$ is the Park anti-transformation matrix, being θ_e the electric angle, and 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. $\mathbb{U}_{\alpha\beta}$ represents the feasible voltage region for $u_{\alpha\beta} = [u_\alpha \;\; u_\beta]^T$, namely, it is the maximum voltage deliverable by the converter, i.e., a hexagonal region in the stationary reference frame, centered in the origin.

Hereafter, it is assumed the prediction horizon $N = 3$ and the control horizon $N_u = 1$. The value of N_u is of great importance because it sets the size of the optimization problem (5). A small control horizon allows a very efficient implementation of the proposed solver. By introducing the solution vector $\Delta u = [\Delta u(k), \Delta u(k+1), ...\Delta u(k+N_u-1)]$, the problem (5) can be reformulated as:

$$\begin{aligned} &\min_{\Delta u} \; J(\Delta u) := \frac{1}{2} \Delta u^T H \Delta u + c^T \Delta + \text{const}, \\ &\text{s.t.} \quad F\Delta u \leq f \end{aligned} \quad (6)$$

where the size of the matrices are $H \in \mathbb{R}^{2N_u \times 2N_u}$ and $c \in \mathbb{R}^{2N_u \times 1}$. By choosing $N_u = 1$, i.e., $\Delta u = \Delta u(k)$, the voltage constraints matrices in (6) are (see Appendix A):

$$\begin{aligned} F &= \begin{bmatrix} 1 & 1 & 1 & -1 & -1 & -1 \\ \sqrt{3} & 0 & -\sqrt{3} & -\sqrt{3} & 0 & \sqrt{3} \end{bmatrix}^{\mathrm{T}} T(\theta_e), \\ f &= \frac{2u_{DC}}{\sqrt{3}} \begin{bmatrix} 1 & 0.5 & 1 & 1 & 0.5 & 1 \end{bmatrix}^{\mathrm{T}} - Fu(k-1). \end{aligned} \quad (7)$$

It is worth noting that the above mentioned constraints describe an hexagon, which is the rotated and translated version of $\mathbb{U}_{\alpha\beta}$, according to the transformation $T(\theta_e)$ and $u(k-1)$, respectively. 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 $i = 1, ...6$, being ℓ_i associated to the constraint $F(i,:)\Delta u(k) \leq f(i)$.

Now, let Δu^* be the optimal solution of constrained MPC problem (6), the applied control input at time k is obtained as:

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

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

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2022.3150283, IEEE Access

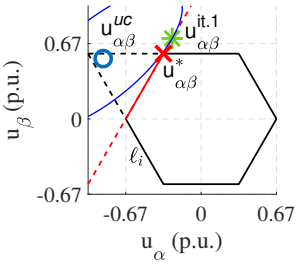


FIGURE 1: One constraint violation: the solution is found by adding the penalty function represented by the red line to the cost function.

IV. ALGORITHM DESCRIPTION

The aim of the proposed method is to solve the constrained QP problem (6) with a finite succession of iterations. First, the optimal solution of problem (6) is computed in closed form, neglecting the voltage constraints:

$$\Delta u^{uc} = -H^{-1}c. \tag{8}$$

Then, the relative position between the unconstrained optimal solution Δu^{uc} and the hexagon region identifies the number of violated constraints. Four different situations can occur:

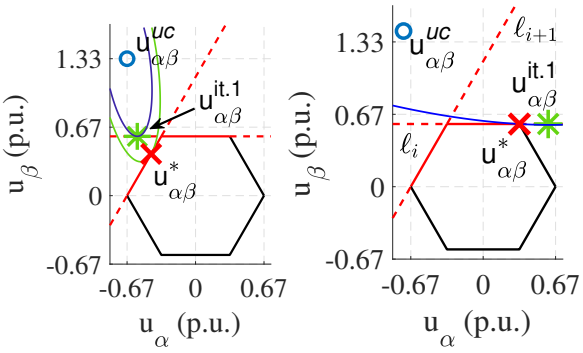
0) The solution is optimal and feasible, i.e., the optimal unconstrained solution lies within the feasible voltage set, and it is applied as voltage reference to the inverter;
1) one constraint is not satisfied (see Fig. 1);
2) two constraints are not satisfied (see Fig. 2);
3) three constraints are not satisfied (see Fig. 3).

Notice that, in the figures, the unconstrained solution $u^{uc}_{\alpha\beta}$ 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, namely when at least one constraint is not respected, we propose a deepening on the shape of the function cost (4) and the relative positioning of the hexagonal constraint.

A. REGIONS OF VIOLATED CONSTRAINTS

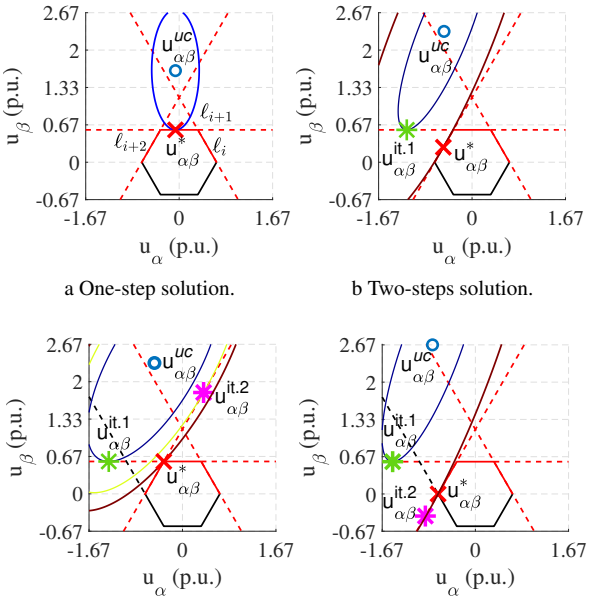
In this section, we express the functional cost J in (4) in terms of $u_{\alpha\beta}$. It turns out that the level set curves are centered in $u^{uc}_{\alpha\beta}$, where the eccentricity depends on the ratio of the dq inductances and the orientation depends on the electric angle. Fig. 4 shows two examples of cost function shapes in the $\alpha\beta$-plane, where Fig. 4a represents the case of an anisotropic motor, while Fig. 4b shows the case of an isotropic one. In both cases, a representative hexagonal feasible voltage set in the $\alpha\beta$-plane described by (7) is reported.

By analyzing the six inequalities defined in (7), it is possible to relate the relative position of the unconstrained optimal solution with respect to the hexagonal feasible voltage set. Thus, the number of violated constraints can be quantified. Fig. 5 shows the partition of the plane according to the number of constraints, numbered from 0 to 3. For each of these cases, in Sec. IV-C a tailored solution strategy is described.



a Two steps solution. b Vertex solution.

FIGURE 2: Two constraint violations. Fig. 2a depicts the cases where the optimal solution lies on an hexagon side, whereas in Fig. 2b the optimal solution lies on an external hexagon vertex.



a One-step solution. b Two-steps solution.

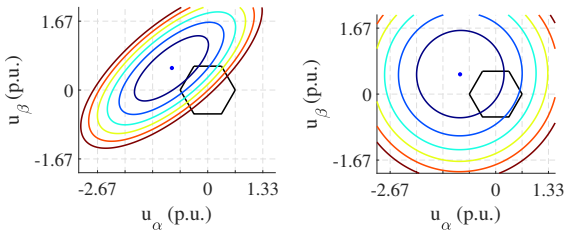
c Vertex solution (first case). d Vertex solution (second case).

FIGURE 3: Three constraint violations. Fig. 3a and Fig. 3b depict the case where the optimal solution lies on the hexagon side and it is found in one or two procedure steps, respectively. In Fig. 3c and Fig. 3d, the optimal solution lies on a hexagon vertex.

B. THE FUNDAMENTAL ALGORITHM STEP

Given an unfeasible unconstrained solution, the proposed method solves the problem (6) by considering

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2022.3150283, IEEE Access



a $L_d \neq L_q$ b $L_d = L_q$

FIGURE 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.

only one violated constraint at a time, named ℓ_{i^*}, regardless of the region in which the unconstrained solution is located. The fundamental routine solves the modified optimal control problem:

$$\min_{\Delta u(k)} \quad J(\Delta u(k)) \qquad \text{s.t. } F(i^*,:)\Delta u(k) \leq f(i^*), \quad (9)$$

where only the ℓ_{i^*} voltage constraint is considered. The solution is computed by relaxing the constraint in (9), solving the almost equivalent unconstrained problem:

$$\min_{\Delta u(k)} \quad J(\Delta u(k)) + \lambda_{i^*}(F(i^*,:)\Delta u - f(i^*))^2, \quad (10)$$

for λ_{i^*} sufficiently high to guarantee a negligible distance from the constraint. Basically, the functional cost J has been augmented with a penalty function in (10), forcing the solution to stay on the active constraint. It is worth noting that the problem (10) is unconstrained with respect to (9), therefore it admits an analytic solution.

C. ALGORITHM STEPS

The proposed algorithm finds the optimal constrained solution of (5) with a two-step procedure. First, it computes the unconstrained solution Δu^{uc}, then, all violated constraints, if present, are taken into account one by one by recalling the *Fundamental Step* described in Sec. IV-B. In the following, all possible cases are described.

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 constrained optimal solution certainly lies on the segment itself. A representative example is shown in Fig. 1, where the solutions are plotted, as before, in the $\alpha\beta$-plane. The optimal solution is then computed according to the following steps:

1) The solution of the problem (10) is computed considering as constraint ℓ_i[1], i.e., with $i^* = i$, namely Δu^{oa};

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

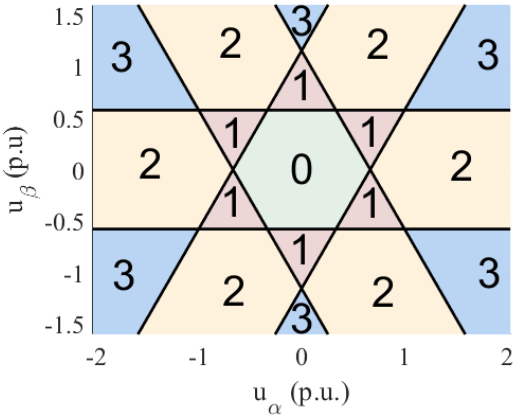


FIGURE 5: Portions of $\alpha\beta$-plane identified by the hexagon lines, highlighting the number of input constraints that are overcome simultaneously.

2) a feasibility check is then operated, computing:

$$h_j := F(j,:)\Delta u^{oa} - f(j) \leq 0 \text{ for } j = i-1, i+1; \quad (11)$$

   a) if Δu^{oa} does not violate any constraint, then $\Delta u^* = \Delta u^{oa}$ and lies on the segment;
   b) if Δu^{oa} violates the next constraint[2] ℓ_{i+1}, i.e., if $h_{i+1} > 0$, thus Δu^* is the intersection between ℓ_i with ℓ_{i+1};
   c) if $h_{i-1} > 0$, Δu^* is the intersection between ℓ_i with ℓ_{i-1}.

It is worth highlighting that the aim of the feasibility check is to bound Δu^{oa} on the feasible hexagon side.

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, the solution 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 optimal solution lies within one of the hexagon segments that lies either on ℓ_i or on ℓ_{i+1} (see Fig. 2a);
- the optimal solution lies on the intersection of ℓ_i and ℓ_{i+1};
- the optimal solution lies on one of the extreme 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. 2b).

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

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

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2022.3150283, IEEE Access

1) The solution to the problem (10) is computed considering as constraint ℓ_i, i.e., with $i^* = i$, namely Δu^{oa};
2) a feasibility check is then operated, computing h_j as in (11) for $j = i+1, i-1$;
   a) if Δu^{oa} does not violate any constraint, then $\Delta u^* = \Delta u^{oa}$;
   b) if $h_{i+1} > 0$, thus Δu^* lies on ℓ_{i+1} itself and the *one constraint violation* procedure can be applied (Fig. 2a);
   c) if $h_{i-1} > 0$, Δu^* is intersection between ℓ_i with ℓ_{i-1} (Fig. 2b).

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 optimal 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 optimal solution lies on the intersections of ℓ_i and ℓ_{i+1} or ℓ_{i+1} and ℓ_{i+2};
- the optimal solution lies on one of the extreme 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 optimal solution is then computed according to the following steps:

1) The solution to the problem (10) is computed considering as constraint ℓ_{i+i}, i.e., with $i^* = i+1$ (the central violated constraint), namely Δu^{oa};
2) A feasibility check is then operated, computing h_j as in (11) for $j = i+2, i$;
   a) if Δu^{oa} does not violate any constraint, then $\Delta u^* = \Delta u^{oa}$ (Fig. 3a);
   b) if $h_{i+2} > 0$ (or $h_i > 0$), Δu^* certainly lies on $\ell_i + 2$ (or ℓ_i respectively) and the *one constraint violation* procedure can be applied; Fig. 3b, Fig. 3c and Fig. 3d summarized the possible situations, respectively.

V. COMPUTATIONAL ANALYSIS

In the previous section we described an algorithm to solve the constrained QP (5) that takes into account the voltage constraints and 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`.

TABLE 3: 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 H
q-axis inductance	L_q	0.06 H
Nominal current	I_N	6 A
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 4: Averaged time comparison between the proposed method and `qpOASES` for different number of steps.

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

A. COMPARISON WITH `qpOASES`

`qpOASES` is an open-source software, it is very robust and suitable for medium-small scale problems, and it can be considered as a very good benchmark in terms of solution accuracy. The MPC problem is built considering a SyRM motor, whose parameters are given in Table 3. 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 algorithms were run one million times, averaging the time spent for the computation. The numerical results are presented in Table 4 where 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 2a);
- 1.5 - if the one constraint violation procedure indicates a solution found on the external vertex (step 2b or 2c);
- 2 - refers to the case where two feasibility checks (10) are evaluated;
- 2.5 - corresponds to the worst-case scenario where, in the two and three constraint violations cases, the optimal solution is on a vertex after solving (10) twice.

As can be noticed, the proposed solver finds the MPC solution with a lower computational burden with respect to

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2022.3150283, IEEE Access

TABLE 5: 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

qpOASES. It is worth highlighting that in the worst case scenario, namely, when the solution violates all three constraints and the optimal one is an hexagon vertex, the average computation time remains very limited. The execution time of this case is considered the upper bound for the control algorithm cost.

B. WORST-CASE COMPUTATIONAL COST

The real-time feasibility certification for algorithms that aspire to large-scale industrial applications plays a crucial role. To this aim, we propose a worst-case performance analysis. Since the computational performances strongly rely on the specific implementation, the total number of algebraic operations performed to find the solution in the worst-case scenario is quantified for the proposed method. The sequence of operations is listed below:

1) compute the unconstrained solution (8);
2) feasibility check (11) $\Rightarrow$ solution unfeasible;
3) find new solution (10);
4) feasibility check (11); $\Rightarrow$ solution unfeasible;
5) find new solution (10);
6) feasibility check (11); $\Rightarrow$ solution unfeasible;
7) find the solution among the vertices.

The corresponding number of operations is reported in Table 5. It is worth noting that step 7) does not require any additional algebraic operations. In conclusion, the peculiarities of the proposed method are twofold: 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. The aforementioned characteristics make the algorithm well suitable for implementation on industrial hardware.

VI. EXPERIMENTAL RESULTS

The test bench adopted for the experiments is shown in Fig. 6. It consists of a master motor directly connected to the SyRM motor under test, whose parameters are listed in Table 3. 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. It provides the gate signals for the inverter and performs current and position acquisitions.

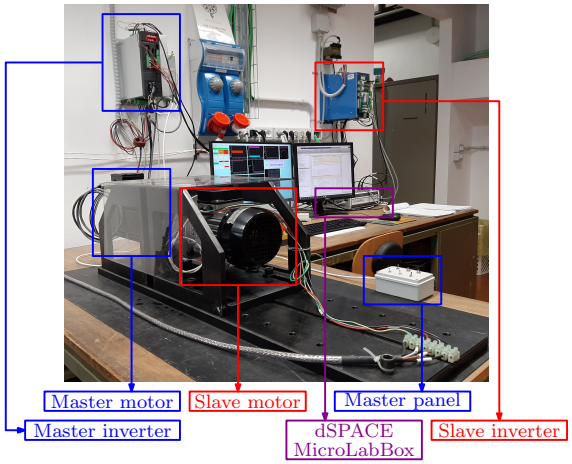
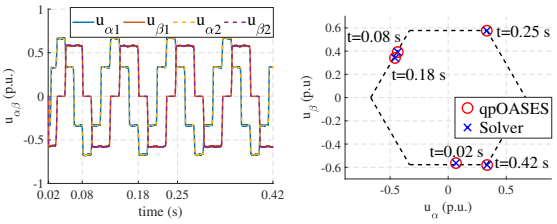


FIGURE 6: Test-bed layout.

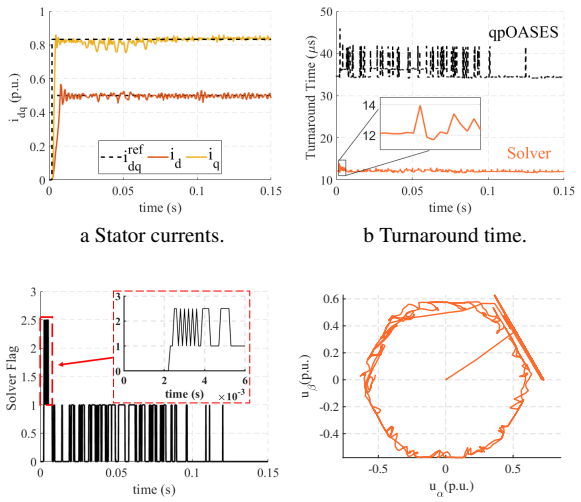


a $\alpha\beta$ Voltages. b Solution Snapshots.

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

A. SOLUTION ACCURACY

To verify the effectiveness of the proposed method, a tailored test was designed. A reduced value of the DC bus was set for the controllers in order to achieve an unfeasible working point, namely, the unconstrained solution is outside the voltage hexagon. In this condition, the solver is pushed to find a solution along the hexagon edges, where voltage constraints are active. During the test, qpOASES was used as solver and it was considered as the benchmark. The recorded data were used to run the MPC problem offline and the voltage solutions were computed with the proposed method for each time-step. The results are shown in Fig. 7. The voltages are plotted in per unit with respect to the average value of the DC bus. The solutions, i.e. the voltage references, found by qpOASES and the presented solver were overlapped, as confirmed by the voltage reference in Fig. 7a. In particular, some test points are plotted in Fig. 7b in the $\alpha\beta$-plane for the selected time snapshots indicated in Fig. 7a. The solutions of both algorithms are identical, proving the correctness of the proposed solver.



a Stator currents. b Turnaround time.

c Solver flag. d Transient Voltage.

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

B. SOLVER PERFORMANCE

To evaluate the solver performance in terms of computational cost, we propose a current step transient (see Fig. 8). The master motor kept the motor under test at its nominal speed, and the nominal current value was set as reference. This condition represents one of the most stressing test for evaluating current controller performance. Thus, it is expected this represents the worst-case scenario for the computational cost of the algorithms. Fig. 8a shows the dq currents transients, normalized with respect to the nominal peak current value. The most interesting part of this test can be observed in Fig. 8b, where the turnaround time of both algorithms is measured by dSPACE. The time required by measurements acquisitions and elaborations is about 8 μs. It can be observed that there is a strong agreement between the computational performance in Table 5 and the experimental-based ones. The proposed solver shows a peak at the first instant with 14 μs, while the average time stabilizes at 12 μs. Fig. 8c shows 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 μs at maximum to find the optimal voltages. The voltage solutions are reported in Fig. 8d, which confirms that solutions are found on the voltage hexagon during this transient.

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, a fast and effective method for solving in real-time the quadratic programming problem related to the MPC has been proposed. The solver was designed for the implicit current model predictive control of a synchronous motor drive. The number of steps in the worst-case scenario is fixed, thus the number of operations can be assessed in advance. One of the main features of the proposed method is that the real-time certification can be achieved. Experimental results have been obtained by using the open source solver qpOASES as benchmark for comparing the optimal solution. The proposed solver has shown an excellent 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.

.

APPENDIX A

The system of inequalities in (7) describes an hexagon centered in the stator reference frame origin. However, the hexagon is described in the rotating reference frame with respect to the optimization variable $\Delta u(k)$, which is the solution of the MPC problem. The expression of the ℓ_i voltage constraint, namely an hexagon edge, is $u_\beta = m_i u_\alpha + q_i$, where m_i, q_i are real numbers. In matrix notation the expression becomes:

$$\begin{bmatrix} -m_i & 1 \end{bmatrix} \begin{bmatrix} u_\alpha \\ u_\beta \end{bmatrix} = q_i. \tag{12}$$

Then, (12) can be expressed with respect to the dq voltage u as:

$$\begin{bmatrix} -m_i & 1 \end{bmatrix} \begin{bmatrix} \cos(\theta_e) & -\sin(\theta_e) \\ \sin(\theta_e) & \cos(\theta_e) \end{bmatrix} \begin{bmatrix} u_d \\ u_q \end{bmatrix} = q_i. \tag{13}$$

Introducing the optimization variable $\Delta u(k)$, it results:

$$\begin{bmatrix} -m_i & 1 \end{bmatrix} \begin{bmatrix} \cos(\theta_e) & -\sin(\theta_e) \\ \sin(\theta_e) & \cos(\theta_e) \end{bmatrix} \begin{bmatrix} \Delta u_d(k) \\ \Delta u_q(k) \end{bmatrix} = q_i - \begin{bmatrix} -m_i & 1 \end{bmatrix} \begin{bmatrix} \cos(\theta_e) & -\sin(\theta_e) \\ \sin(\theta_e) & \cos(\theta_e) \end{bmatrix} \begin{bmatrix} u_d(k-1) \\ u_q(k-1) \end{bmatrix}. \tag{14}$$

Each row of the constraint matrices in (7) contains The coefficients contained in matrix F and f in (7) are the ones of the six lines of the adopted hexagon. For instance, the first border has $m = -\sqrt{3}$ and $q = \frac{2}{\sqrt{3}} u_{DC}$. The expression in (14) is used with the inequality relation for all the lines, which requires that the solution lies within the feasible region identified by the hexagon, including its boundaries.

REFERENCES

[1] 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, no. 2, pp. 935–947, 2017.

[2] F. Wang, S. Li, X. Mei, W. Xie, J. Rodríguez, and R. M. Kennel, "Model-based predictive direct control strategies for electrical drives: An experimental evaluation of ptc and pcc methods," IEEE Trans. on Ind. Informat., vol. 11, no. 3, pp. 671–681, 2015.

[3] W. Tu, G. Luo, Z. Chen, C. Liu, and L. Cui, "Fpga implementation of predictive cascaded speed and current control of pmsm drives with two-time-scale optimization," IEEE Trans. on Ind. Informat., vol. 15, no. 9, pp. 5276–5288, 2019.

[4] F. Toso, R. Torchio, A. Favato, P. G. Carlet, S. Bolognani, and P. Alotto, "Digital twins as electric motor soft-sensors in the automotive industry," in 2021 IEEE International Workshop on Metrology for Automotive (MetroAutomotive), 2021, pp. 13–18.

[5] A. Fuller, Z. Fan, C. Day, and C. Barlow, "Digital twin: Enabling technologies, challenges and open research," IEEE Access, vol. 8, pp. 108 952–108 971, 2020.

[6] M. Siami, D. A. Khaburi, M. Rivera, and J. Rodríguez, "A computationally efficient lookup table based fcs-mpc for pmsm drives fed by matrix converters," IEEE Transactions on Industrial Electronics, vol. 64, no. 10, pp. 7645–7654, 2017.

[7] 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. on Ind. Informat., vol. 14, no. 4, pp. 1334–1346, 2018.

[8] 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, no. 3, pp. 219–230, Sep. 2017.

[9] R. P. Aguilera and D. E. Quevedo, "Predictive control of power converters: Designs with guaranteed performance," IEEE Trans. on Ind. Informat., vol. 11, no. 1, pp. 53–63, 2015.

[10] S. A. Davari, V. Nekoukar, C. Garcia, and J. Rodriguez, "Online weighting factor optimization by simplified simulated annealing for finite set predictive control," IEEE Transactions on Industrial Informatics, vol. 17, no. 1, pp. 31–40, 2021.

[11] 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. on Ind. Informat., vol. 14, no. 4, pp. 1334–1346, 2018.

[12] A. Zanelli, A. Domahidi, J. Jerez, and M. Morari, "Forces nlp: an efficient implementation of interior-point methods for multistage nonlinear nonconvex programs," Intern. Journ. of Contr., vol. 93, 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] G. Cimini, D. Bernardini, S. Levijoki, and A. Bemporad, "Embedded model predictive control with certified real-time optimization for synchronous motors," IEEE Trans. Control System Tech., pp. 1–8, 2020.

[15] B. Stellato, G. Banjac, P. Goulart, A. Bemporad, and S. Boyd, "OSQP: an operator splitting solver for quadratic programs," Mathematical Programming Computation, vol. 12, no. 4, pp. 637–672, 2020.

[16] 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, no. 1, pp. 327–363, 2014.

[21] 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. Elect. Machines Drives Conf. (IEMDC), 2019, pp. 2210–2216.

[17] 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, June 2009.

[18] S. Mariethoz, A. Domahidi, and M. Morari, "High-bandwidth explicit model predictive control of electrical drives," IEEE Trans. Ind. Appl., vol. 48, no. 6, pp. 1980–1992, 2012.

[19] M. Preindl, S. Bolognani, and C. Danielson, "Model predictive torque control with pwm using fast gradient method," in 2013 Twenty-Eighth Annual IEEE Appl. Power Electron. Conf. and Expo. (APEC), 2013, pp. 2590–2597.

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

[22] 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. Contr. Systems Tech., pp. 1–12, 2021.

[23] 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., May 2009, pp. 1250–1257.

[24] 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. Automat. Contr., vol. 57, no. 6, pp. 1391–1403, June 2012.

[25] R. Verschueren, 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.

[26] 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), 2019, pp. 3365–3370.

[27] G. Cimini and A. Bemporad, "Exact complexity certification of active-set methods for quadratic programming," IEEE Trans. on Autom. Contr., vol. 62, pp. 6094–6109, 2017.

[28] D. Goldfarb and A. Idnani, "A numerically stable dual method for solving strictly convex quadratic programs," Math. Program., vol. 27, no. 1, p. 1–33, Sep. 1983.

[29] 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 Expo. (ECCE), 2019, pp. 6615–6620.

[30] 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, no. 5, pp. 3782–3793, 2021.

[31] M. Siami, D. A. Khaburi, and J. Rodríguez, "Torque ripple reduction of predictive torque control for pmsm drives with parameter mismatch," IEEE Trans. on Pow. Electron., vol. 32, no. 9, pp. 7160–7168, 2017.

[32] 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., pp. 1–1, 2021.