Online Model Predictive Torque Control for Permanent Magnet Synchronous Motors

Gionata Cimini*, Daniele Bernardini*, Alberto Bemporad*†, Stephen Levijoki‡
*ODYS Srl. Email: {gionata.cimini, daniele.bernardini, alberto.bemporad}@odys.it
†IMT Institute for Advanced Studies Lucca, Italy.
‡General Motors Company. Email: stephen.levijoki@gm.com

***Abstract*—In this paper a Model Predictive Control (MPC) method for torque control of a Permanent Magnet Synchronous Motor (PMSM) is presented. The proposed approach takes into account constraints on voltages and currents, and allows the use of modulation techniques that eliminate the side effects caused by the direct transistor actuation performed by Model Predictive Direct Torque Control (MP-DTC) approaches. The optimization problem resulting from the proposed MPC formulation is solved online, in contrast with what is done in explicit MPC, where the optimal control law is obtained offline by multiparametric optimization. The performance of the proposed control strategy is evaluated in Processor-In-the-Loop (PIL) experiments, carried out on a low cost Digital Signal Processor (DSP) commonly used in motion control. Results show that the proposed approach is able to improve the torque dynamics with respect to a conventional controller, and that an embedded implementation is feasible in terms of required computational power and memory.**

I. Introduction

Permanent Magnet Synchronous Motors (PMSMs) are widely used in applications where high-performance, fast torque response and high power density are required, such as in robotics and machine tool drives. The standard control method for both PMSMs and induction machines is Field Oriented Control (FOC) with a cascade structure, where three linear controllers (usually PI) are employed, one for tracking the reference position/speed and two for the regulation of currents in the direct-quadrature (d, q) reference frame. When Pulse Width Modulation (PWM) is used, this control architecture is often referred to as PI-PWM.

In recent years Model Predictive Control (MPC) has gained considerable attention in the field of power converters and electrical drives [1]–[3]. The main idea of MPC is to obtain the control actions by solving at each sampling time a finite-horizon optimal control problem based on a given prediction model of the controlled process and an estimation of its current state. At the cost of a relatively high on-line computational burden, it provides a coordinated operation of the available actuators to track multiple references and satisfy bounds on inputs, outputs, and states of the process. As such, it is well suited to handle multivariable and constrained control problems. The application of MPC in the power electronics and electrical drives field is mainly motivated by two facts: the mathematical models of these systems, needed by MPC, are well known, and several constraints have to be considered in the design of the controller. In particular, voltage and current constraints must be respected due to the physical limits of the system and to safety reasons. MPC could be applied to other areas of electric motor control such as Electronic Throttle Control (ETC) and Hybrid Electric Motor control in automotive applications. The ability to enforce constraints using MPC could benefit motor control performance and component protection.

To avoid the computational load required by online MPC two main techniques have been widely adopted in recent years: explicit MPC [4] and Finite Control Set MPC (FCS-MPC) [5]. In explicit MPC the optimization problem is pre-solved offline via multiparametric Quadratic Programming (mpQP), and the MPC solution turns out to be an easy-to-implement PieceWise Affine (PWA) function of the parameters, for example via a binary search tree evaluation [6]. The search time, in this case, is logarithmic and the memory occupancy is polynomial in the number of PWA regions, thus the explicit controller is well suited for small problems only. The explicit MPC approach has been successfully applied to motion control, see, e.g., [7]–[11].

The FCS-MPC was initially applied in the field of power converters and drives. It exploits the discrete nature of those systems by manipulating the inverter switch positions directly. The approach results in a combinatorial optimization problem that is solved on line by enumeration and provides a sequence of switch positions over the considered prediction horizon. This technique has received considerable attention in recent years and has been successfully implemented in motion control [12]–[17]. FCS-MPC has been shown to significantly reduce the switching losses and the harmonic distortions when compared to standard methodologies. Unfortunately, such results are only achieved with high control frequency (e.g., 40 kHz in [15]), as this direct actuation couples the switching frequency with the controller sampling time. As a consequence, even though the optimization problem is relatively simple to solve, a powerful digital controller is required to guarantee improvements. Furthermore, the involved computational load scales exponentially with the MPC prediction horizon, due to the combinatorial nature of the problem. A review of the strategies developed to achieve a long prediction horizon while keeping the control problem tractable is presented in [18].

Other MPC approaches that have been proposed for this kind of problem do not model the discrete nature of the system, and can be formulated as convex optimization problems. In this field, such strategies are referred to as Continuous Control Set MPC (CCS-MPC). Several comparisons have been carried out between FCS-MPC and CCS-MPC, and they have been shown to provide similar performances [19]–[21]. However, CCS-MPC has several advantages over FCS-MPC: it works well with longer sampling intervals, it allows for decoupling

978-1-4799-7800-7/15/$31.00 ©2015 IEEE

of the switching frequency from the controller sampling time, and it operates the inverter at a fixed frequency.

In this paper an online MPC method for torque control of a PMSM is proposed, that we refer to as Model Predictive Torque Control (MP-TC). The goal of the control action is to provide an optimal amount of torque that allows to drive the motor at a desired speed profile given the load torque, while satisfying voltage and current constraints. Similar to the standard FOC based on PI, the architecture of the proposed method consists of an outer loop that controls the motor speed, and an inner loop that regulates the (d, q) currents. In the proposed approach, MPC replaces the linear controllers of FOC in the inner loop, while the outer loop is left unchanged. This choice is motivated by two primary reasons: the mechanical dynamics are slower compared to the electrical ones, and the critical constraints on the system concern the electrical variables that are dealt with in the inner loop (voltages and currents). Compared to Model Predictive Direct Torque Control (MP-DTC), where the transistors are actuated directly and the control action is obtained by enumeration [12]–[14], [22], MP-TC guarantees a fixed switching frequency, which is of utmost concern in industrial applications where losses and thermal stress have to be taken into account. Moreover, the modulator decouples the switching frequency from the sampling time, allowing the use of a less powerful Micro Controller Unit.

The proposed MP-TC approach belongs to the class of CCS-MPC, where the control move is obtained by solving an optimization problem at each sampling instant. In particular, the proposed method amounts to solving online a Quadratic Programming (QP) problem, where a quadratic cost function of the variables of interest is minimized, subject to linear constraints. Hence, in order to implement the MP-TC on a digital controller, a QP solver needs to be embedded in it. To the best of the authors' knowledge, online MPC for electrical drives on a low-power platform has been discussed only in [23]. However, in that work no input constraints are taken into account, the prediction horizon is short (one or two prediction steps), and no information about the optimality of the achieved solution is provided. In this paper, the authors have implemented an efficient QP solver and carried out Processor-in-the-Loop (PIL) experiments to prove the feasibility of the proposed approach. The controller and the QP solver have been embedded in a Texas Instruments DSP, and simulated in closed-loop with an accurate model of a PMSM provided by the drive manufacturer, Technosoft SA. As a novel contribution, the authors demonstrate that MPC for constrained torque control can be successfully implemented on low-power DSPs commonly used for motion drive, without sacrificing control features.

The paper is organized as follows. The mathematical model of the considered system is presented in Section II and the design of the proposed online MPC method is described in Section III. PIL results are reported in IV, including a comparison with a standard FOC controller. Finally, conclusions and considerations on future work are reported in Section V.

II. Mathematical Model

With respect to the (d, q) reference frame rotating synchronously with the rotor, the electrical subsystem of a PMSM can be modeled as follows:

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

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

where $t \in \mathbb{R}_+$ is the time index, R is the stator resistance [Ω], L_d and L_q are the stator inductances [H] on the d-axis and the q-axis, respectively, $i_d(t)$ and $i_q(t)$ are the stator currents [A], and $u_d(t)$ and $u_q(t)$ are the stator voltages [V]. The quantities on the d-axis are the direct voltage/current components, whereas those on the q-axis are the quadrature components. The mechanical motion of the motor can be described by the following equations:

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

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

where $\omega(t)$ is the electrical rotor speed [rad/s], B is the coefficient of viscous friction [N·m·s], λ is the motor flux leakage [Wb], J is the inertia coefficient [kg·m^2], $\tau(t)$ is the electrical torque [N·m], $\tau_l(t)$ is the load torque [N·m], and p is the number of pole pairs. Considering an isotropic PMSM where $L_d = L_q = L$, the simplified and complete model can be written as follows:

$$\dot{i}_d(t) = -\frac{R}{L} i_d(t) + \omega(t) i_q(t) + \frac{1}{L} u_d(t) \tag{3a}$$

$$\dot{i}_q(t) = -\frac{R}{L} i_q(t) - \omega(t) i_d(t) + \frac{1}{L} u_q(t) - \frac{\lambda}{L}\omega(t) \tag{3b}$$

$$\dot{\omega}(t) = \frac{B}{J}\omega(t) + \frac{p}{J} K_t i_q(t) - \frac{p}{J}\tau_l(t), \tag{3c}$$

where $\tau(t) = K_t i_q(t) = \frac{3}{2} p \lambda i_q(t)$ and K_t the torque constant. In the following we will assume $p = 1$. This assumption holds true for the PMSM considered in this work. In this case the electrical quantities correspond to the mechanical ones.

III. Control Design

The architecture of a standard FOC based on proportional-integral controllers (PI-FOC) consists of an outer speed loop and an inner current loop (see Figure 1). In order to achieve the desired motor speed, the outer loop provides a torque reference τ_{ref} to the inner loop, that runs at higher frequency and regulates the (d, q) currents in order to track the torque reference. When no flux weakening is considered, the field component i_d is steered to zero and the quadrature component i_q is controlled to track a reference $i_{q,\text{ref}}$, which is obtained by scaling the torque reference τ_{ref} by the constant K_t. Setting the current i_d to zero is a good solution for isotropic machines where maximum current implies maximum torque. In order to take field weakening into account, Maximum Torque per Ampere (MTPA) tracking could be adopted [13], [23], however this is out of the scope of this paper. The rotor speed is estimated from the position information provided by the encoder, and two of the three phase currents are measured to derive the (d, q) currents, since the system is balanced. In PI-FOC, the actuation commands computed by the linear controllers are usually saturated to avoid violations of electrical constraints.

In the proposed MP-TC architecture, the current controllers in the inner loop are replaced by MPC, while the outer loop

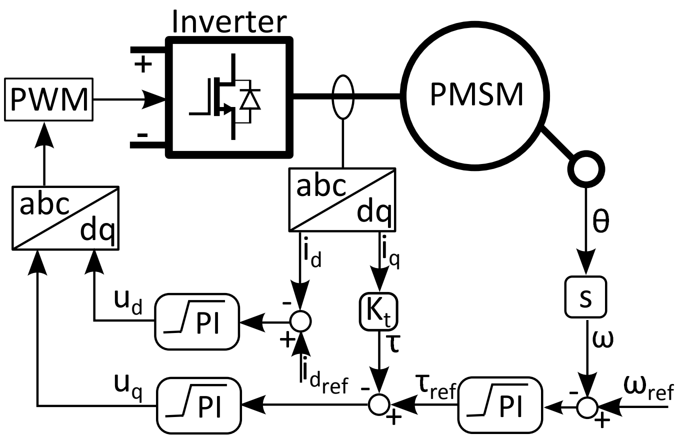


Figure 1. Drive control system with the standard PI-FOC

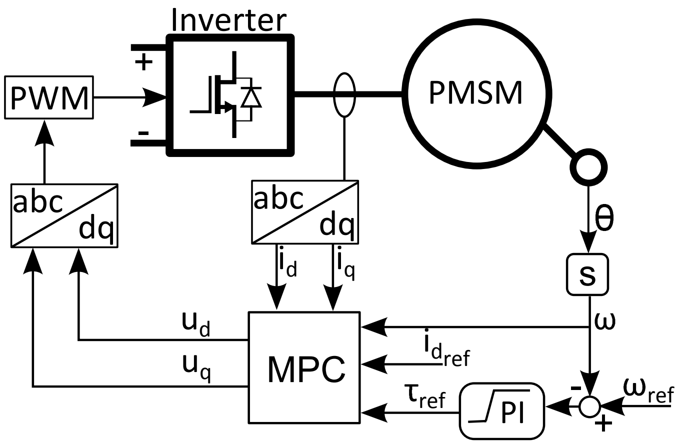


Figure 2. Drive control system with the proposed MP-TC

remains unchanged (see Figure 2). As briefly mentioned above, this hierarchical structure is justified by several reasons. First, the dynamics of the mechanical part of the system are slower than those of the electrical part. Furthermore, since the main constraints in a motor involve stator voltages and currents, MPC is especially useful in the inner loop, where those constraints can be modeled as bounds on input and output variables. Finally, the bilinear model (3) can be decoupled in two linear models that can be used for MPC predictions.

A. MPC formulation

Let us focus on the inner loop to derive the formulation of the proposed MPC problem. The current equations (3a)-(3b) can be linearized by imposing a nominal speed $\omega(t) = \omega_0$ in the bilinear terms $\omega(t)i_q(t)$ and $\omega(t)i_d(t)$. Discretizing the resulting model with the inner loop sampling time T_s yields a discrete-time linear time-invariant (LTI) system of the form

$$x(k+1) = Ax(k) + Bu(k) + Gv(k) \tag{4a}$$
$$y(k) = Cx(k), \tag{4b}$$

where $k \in \mathbb{N}$ is the discrete-time index, $x(k) = [i_d(k),\, i_q(k)]'$ is the state vector, $u(k) = [u_d(k),\, u_q(k)]'$ is the manipulated input vector, $y(k) = [i_d(k),\, \tau(k)]'$ is the output vector, $v(k) = \omega(k)$ is a measured disturbance, and $A = e^{A_c T_s}$, $B = \int_0^{T_s} e^{A_c\tau} d\tau B_c$, $G = \int_0^{T_s} e^{A_c\tau} d\tau G_c$, $C = C_c$, where

$$A_c = \begin{bmatrix} -\frac{R}{L} & \omega_0 \\ -\omega_0 & -\frac{R}{L} \end{bmatrix} \qquad B_c = \begin{bmatrix} \frac{1}{L} & 0 \\ 0 & \frac{1}{L} \end{bmatrix}$$
$$G_c = \begin{bmatrix} 0 \\ -\frac{\lambda}{L} \end{bmatrix} \qquad C_c = \begin{bmatrix} 1 & 0 \\ 0 & K_t \end{bmatrix}$$

are the matrices of the continuous-time system. The LTI model (4) is used as the prediction model in the MPC problem.

At every time step k, the optimal control move for the inner loop is obtained by solving the following MPC control problem:

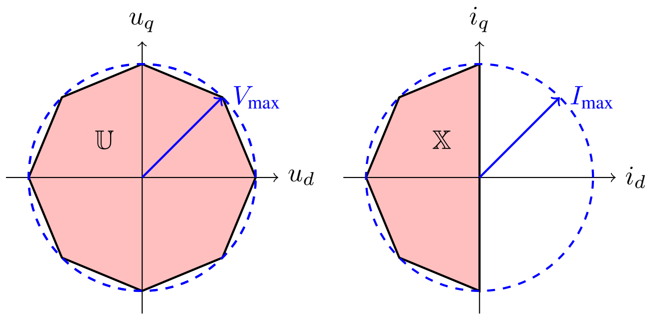
$$\begin{aligned}
\min_{\Delta u} \quad & \sum_{i=0}^{N-1} \|Q(y_{k+i|k} - r(k))\|_2^2 + \sum_{j=0}^{N_u-1} \|R\Delta u_{k+j|k}\|_2^2 + \\
& + \|P(y_{k+N|k} - r(k))\|_2^2 && \text{(5a)}\\
\text{s.t.} \quad & x_{k|k} = x(k), && \text{(5b)}\\
& x_{k+i+1|k} = Ax_{k+i|k} + Bu_{k+i|k} + Gv(k), && \text{(5c)}\\
& y_{k+i+1|k} = Cx_{k+i+1|k}, && \text{(5d)}\\
& \Delta u_{k+N_u+j|k} = 0, && \text{(5e)}\\
& u_{k+i|k} \in \mathbb{U}, && \text{(5f)}\\
& x_{k+i+1|k} \in \mathbb{X}, && \text{(5g)}\\
& i = 0, 1, \ldots, N-1, && \text{(5h)}\\
& j = 0, 1, \ldots, N - N_u - 1, && \text{(5i)}
\end{aligned}$$

where N is the prediction horizon, N_u is the control horizon, Q, R, and P are weight matrices of appropriate dimension, $x_{k+i|k}$ denotes the prediction of the variable x at time $k+i$ based on the information available at time k, and $\Delta u_{k+i|k} = u_{k+i|k} - u_{k+i-1|k}$, $i = 0, 1, \ldots, N-1$, is the vector of the input increments, with $u_{k-1|k} = u(k-1)$.

The goal of minimizing the cost function (5a) is to track the output reference $r(k) = [i_{d,\text{ref}}(k),\, \tau_{\text{ref}}(k)]'$, while penalizing actuation activity. As previously mentioned, in this formulation $i_{d,\text{ref}} = 0$ and τ_{ref} is provided by the outer loop. Constraints (5b) set the initial state as the current state $x(k)$, (5c)-(5d) represent the prediction model, and (5e) force the input to remain constant after the control horizon, to reduce the complexity of the problem. Equations (5f)-(5g) impose constraints on inputs and states. In order to guarantee the feasibility of problem (5) at every time step, the state constraints (5g) are imposed as soft constraints, i.e., small violations of those constraints are allowed with a large penalty in the cost function. The constraints on inputs and states are defined in the following section.

B. Constraints

Since the MPC acts on the current-voltage subsystem, it cannot handle constraints on mechanical quantities such as motor speed. As discussed above, however, the most critical constraints concern the electrical components [24]. The phase-voltage limit is imposed by the maximum DC-bus voltage tolerated by the inverter. The phase voltage is intrinsically related to the modulation scheme and, given a DC-bus voltage V_{DC}, the maximum allowed voltage can be set to $V_{\max} = \frac{V_{\text{DC}}}{\sqrt{3}}$, both



(a) Voltage constraints (b) Current constraints

Figure 3. System's constraints on both voltage and current

for space vector modulation and pulse-width modulation [23]. On the other hand, the maximum current I_{max} is chosen so as to prevent overheating, and peaks larger than this limit are usually allowed for short time intervals. This justifies the of use soft constraints for (5g). In the (d, q) reference frame, those constraints correspond to norm constraints on inputs and states:

$$u \in \tilde{\mathbb{U}} = \{u \in \mathbb{R}^2 : \|u\|_2 \le V_{\max}\}, \tag{6a}$$
$$x \in \tilde{\mathbb{X}} = \{x \in \mathbb{R}^2 : \|x\|_2 \le I_{\max}\}. \tag{6b}$$

Equations (6) would translate into quadratic constraints. In order to retain the linearity of the constraints in (5), polytopic approximations of (6) can be considered [9], [23]. In this paper we approximate the feasible region of (6) with octagons, that resulted in an acceptable trade-off between accuracy of the approximation and number of constraints. Since the direct component of the stator current i_d is almost always very close to zero, except during flux weakening operation when it takes negative values, we consider the constraint $i_d \le 0$ to reduce the number of constraints that have to be imposed in the online MPC problem. The constraints $u \in \mathbb{U}$, $x \in \mathbb{X}$ that are used in (5) are shown in Figure 3.

C. Online Optimization

Problem (5) can be cast in the form of the following convex QP:

$$\min_z \quad q(z) \triangleq \frac{1}{2} z' H z + p(t)' F' z \tag{7a}$$
$$\text{s.t.} \quad Gz \le W + Sp(t), \tag{7b}$$

where z is the vector of optimization variables and p of time-varying affine parameters. With the limited computational power and memory available on embedded platforms, solving such a problem within the available time interval can be a challenging task. The timing constraints in motion control are very restrictive, especially when electric quantities are considered as variables. It is a common choice to require 1kHz sampling frequency for the speed loop and 10kHz sampling frequency for the current loop. In this work we have implemented an efficient QP solver, that is shown to solve the problem (7) within the allowed time limit. Experimental results are discussed in the next section.

Table I. MP-TC PARAMETERS

Prediction horizon N	5
Control horizon N_u	2
Voltage limit $V_{\max}$	$36/\sqrt{3}$ V
Current limit $I_{\max}$	0.8 A
Output weights Q and P	$\begin{bmatrix} 0.1 & 0 \\ 0 & 0.5 \end{bmatrix}$
Input increments weights R	$\begin{bmatrix} 0.05 & 0 \\ 0 & 0.05 \end{bmatrix}$
Sampling Time T_s	0.3ms

IV. Results

The proposed MP-TC has been tested and compared to a standard PI-FOC controller, and the performance in terms of speed tracking and load disturbance rejection have been evaluated. Processor-in-the-loop simulations have been carried out to accurately assess the feasibility of online MPC in terms of required computational power and memory.

The tests have been carried out on a F28335 Delfino DSP by Texas Instruments, which belongs to the TI C2000 series commonly used in motion control. The DSP has a 32-bit, 150 MHz CPU and an IEEE-754 single-precision Floating-Point Unit (FPU) with one hardware multiplier (32x32 bit). The DSP is connected through a serial port to a PC, where a MATLAB/Simulink® model of the PMSM simulates the controlled process and sends the measurements back to the DSP where the control algorithm runs, closing the control loop. The simulation model of the PMSM is provided by Technosoft SA and is related to the MBE.300.E500 PMSM, which is commercially available. The motor has a 4.3Ω line-to-line stator resistance R, a 3.56mH line-to-line stator inductance L, and the flux linkage λ_0 is 0.0245Wb. The model takes into account the quantization error due to the AC/DC acquisition and the encoder resolution. The inverter is not modeled and consequently switching noise is not considered; in experimental setups with a real motor such noise can be handled by a Kalman filter.

The sampling time adopted for the outer speed loop is 1ms for both MP-TC and PI-FOC, while the sampling time of the inner loop is 0.1ms in PI-FOC and 0.3ms in MP-TC. This longer sampling interval is motivated by the computational burden of the optimization problem which must be solved at every iteration in the MP-TC approach. The parameters of the MPC controller are reported in Table I. We verified that longer control or prediction horizons do not improve the performance of the closed-loop system noticeably. The control algorithm was implemented in plain C and involves only floating-point operations, to exploit the potential of the MCU and optimize the execution time. The PI controllers have been tuned considering the speed-to-torque and the current-to-voltage transfer functions. Tests in simulations have been carried out to fine tune the controllers and obtain the best performance in terms of reference tracking.

The rotor speed profile used in simulation is shown in Figure 4: it includes constant speed operations, an acceleration at 0.1s, and an abrupt brake at 0.4s. During the constant speed profile, the load torque τ_l is suddenly changed at time 0.2s and 0.3s. These abrupt changes are considered in order to have the system operating close to the imposed constraints,

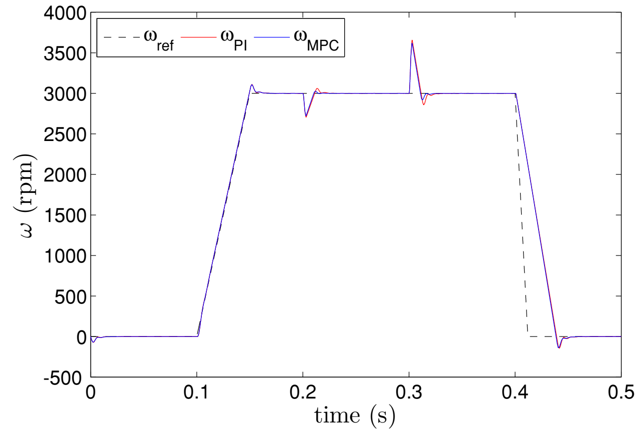


Figure 4. Rotor speed tracking with PI-FOC and MP-TC

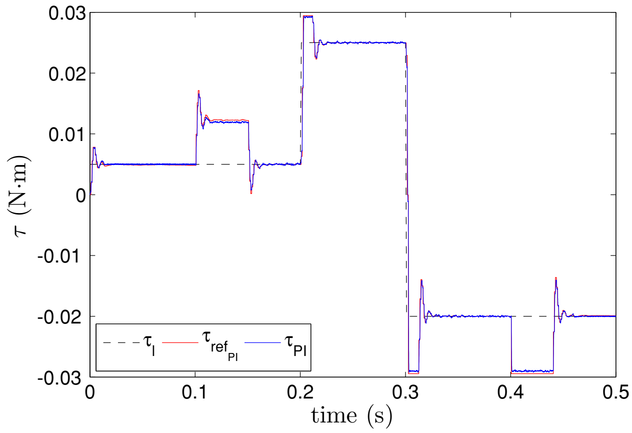


Figure 5. Mechanical and load torque with PI-FOC

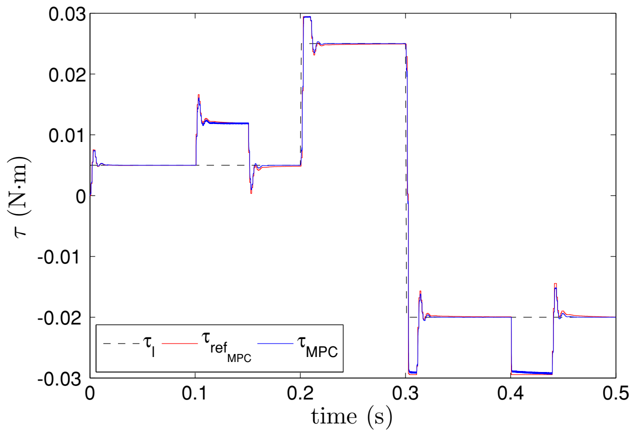


Figure 6. Mechanical and load torque with MP-TC

where the optimal control action is not simply the solution of the unconstrained control problem, so to stress the QP solver and better evaluate its performance. The load-torque profile, the reference torque and the actual mechanical torque with PI-FOC and MP-TC are shown in Figures 5 and 6, respectively.

Simulation results show that MP-TC provides an improved closed-loop performance with respect to PI-FOC, even though the inner-loop sampling time in MP-TC is three times longer than the one in PI-FOC. In particular, MP-TC yields a faster response and smaller overshoots in torque tracking. As expected, improvements in speed tracking are less significant because the speed loop is regulated by a standard PI controller. Performance of MP-TC and PI-FOC are quantitatively compared in terms of integral square error (ISE). The proposed MP-TC grants an improvement of 2.3% and 4.2% in speed and torque tracking, respectively. Figure 7 shows a detail of torque tracking after a set-point change. It is worth noting that both the controllers are able to handle the physical constraints of the system. From the three-phase current plots shown in Figure 8, it is clear that the drive operates close to the output constraints during transient phases. Figure 9 shows the times t_{sol} needed to compute the optimal control action at every sampling instant. These times have been measured on the DSP with an internal clock. The results are very promising: even when the system is controlled near the system constraints, the limit of 0.3ms imposed by the sampling interval is always respected.

The controller is also minimal in terms of memory allocation. The data needed by the MPC are stored in about 2.5KB out of the 34KB of memory provided by the DSP. The memory occupied by the code is not considered in this amount, but it is negligible with respect to the space required by the QP data. Thus, all the control algorithm fits into the volatile SRAM memory, avoiding the access to flash memory at every iteration which could significantly slow down the execution of the control algorithm.

V. Conclusions

In this paper an online model predictive torque control method for a permanent magnet synchronous motor has been proposed. An embedded QP solver is used to solve the MPC problem at each sampling instant. The feasibility of online MPC in terms of required computational power and memory has been demonstrated through Processor-in-the-Loop tests on a low power DSP. PIL experimental results showed that MPC can provide performance improvements with respect to standard controllers. Future work includes extending the proposed strategy to consider the speed loop and to handle flux weakening. Furthermore, the effect of winding resistance increase due to temperature rise will be considered to assess the robustness of the controller. Applications of motor control by

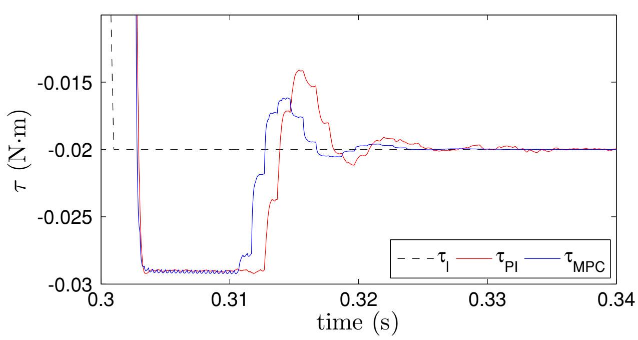


Figure 7. Zoom of the torque response profile with PI-FOC and MP-TC

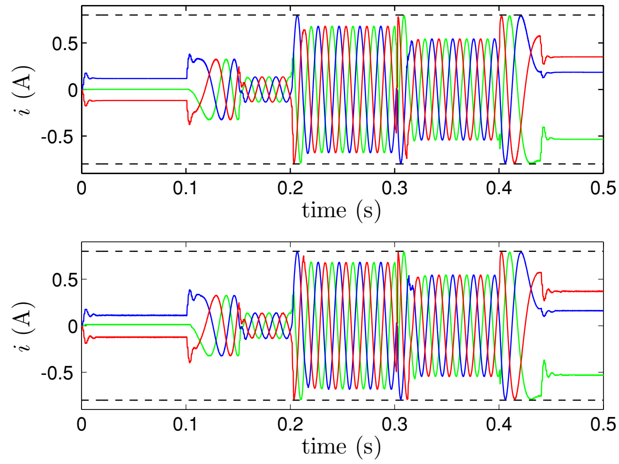


Figure 8. Three-phase currents with MP-TC (top) and PI-FOC (bottom)

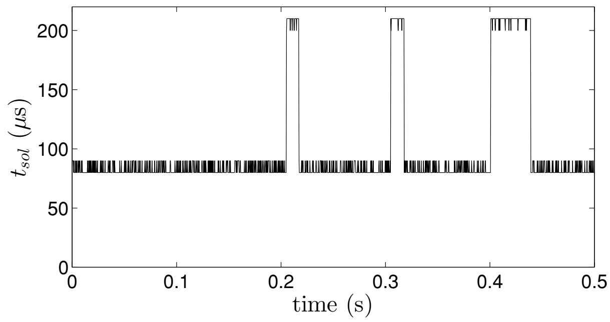


Figure 9. PIL measurement of optimization time on F28335 Delfino DSP by Texas Instruments

online MPC in the automotive domain will also be investigated.

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] J. Rodriguez and P. Cortes, *Predictive Control of Power Converters and Electrical Drives*. John Wiley & Sons, Ltd, May 2012.

[3] S. Vazquez, J. I. Leon, L. G. Franquelo, J. Rodriguez, H. A. Young, A. Marquez, and P. Zanchetta, "Model predictive control: A review of its applications in power electronics," *IEEE Industrial Electronics Magazine*, vol. 8, no. 1, pp. 16–31, Mar. 2014.

[4] 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, 2002.

[5] J. Rodriguez, M. Kazmierkowski, J. Espinoza, P. Zanchetta, H. Abu-Rub, H. Young, and C. Rojas, "State of the art of finite control set model predictive control in power electronics," *IEEE Transactions on Industrial Informatics*, vol. 9, no. 2, pp. 1003–1016, May 2013.

[6] P. Tøndel, T. A. Johansen, and A. Bemporad, "Evaluation of piecewise affine control via binary search tree," *Automatica*, vol. 39, no. 5, pp. 945–950, May 2003.

[7] A. Linder and R. Kennel, "Model predictive control for electrical drives," in *Power Electronics Specialists Conference, 2005. PESC '05. IEEE 36th*, Jun. 2005, pp. 1793–1799.

[8] S. Bolognani, S. Bolognani, L. Peretti, and M. Zigliotto, "Combined speed and current model predictive control with inherent field-weakening features for PMSM drives," in *Electrotechnical Conference, 2008. MELECON 2008. The 14th IEEE Mediterranean*, May 2008, pp. 472–478.

[9] ——, "Design and implementation of model predictive control for electrical motor drives," *IEEE Transactions on Industrial Electronics*, vol. 56, no. 6, pp. 1925–1936, Jun. 2009.

[10] S. Bolognani, R. Kennel, S. Kuehl, and G. Paccagnella, "Speed and current model predictive control of an IPM synchronous motor drive," in *Electric Machines Drives Conference (IEMDC), 2011 IEEE International*, May 2011, pp. 1597–1602.

[11] J. Scoltock, T. Geyer, and U. Madawala, "A comparison of model predictive control schemes for MV induction motor drives," *IEEE Transactions on Industrial Informatics*, vol. 9, no. 2, pp. 909–919, May 2013.

[12] M. Preindl and S. Bolognani, "Model predictive direct torque control with finite control set for PMSM drive systems, part 2: Field weakening operation," *IEEE Transactions on Industrial Informatics*, vol. 9, no. 2, pp. 648–657, May 2013.

[13] ——, "Model predictive direct torque control with finite control set for PMSM drive systems, part 1: Maximum torque per ampere operation," *IEEE Transactions on Industrial Informatics*, vol. 9, no. 4, pp. 1912–1921, Nov. 2013.

[14] ——, "Model predictive direct speed control with finite control set of PMSM drive systems," *IEEE Transactions on Power Electronics*, vol. 28, no. 2, pp. 1007–1015, Feb. 2013.

[15] T. Geyer, G. Papafotiou, and M. Morari, "Model predictive direct torque control - part i: Concept, algorithm, and analysis," *IEEE Transactions on Industrial Electronics*, vol. 56, no. 6, pp. 1894–1905, Jun. 2009.

[16] G. Papafotiou, J. Kley, K. Papadopoulos, P. Bohren, and M. Morari, "Model predictive direct torque control - part II: Implementation and experimental evaluation," *IEEE Transactions on Industrial Electronics*, vol. 56, no. 6, pp. 1906–1915, Jun. 2009.

[17] T. Geyer, "Computationally efficient model predictive direct torque control," *IEEE Transactions on Power Electronics*, vol. 26, no. 10, pp. 2804–2816, Oct. 2011.

[18] P. Karamanakos, T. Geyer, N. Oikonomou, F. Kieferndorf, and S. Manias, "Direct model predictive control: A review of strategies that achieve long prediction intervals for power electronics," *IEEE Industrial Electronics Magazine*, vol. 8, no. 1, pp. 32–43, Mar. 2014.

[19] 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 Transactions on Industrial Electronics*, vol. 56, no. 7, pp. 2715–2728, Jul. 2009.

[20] M. Preindl and S. Bolognani, "Comparison of direct and PWM model predictive control for power electronic and drive systems," in *2013 Twenty-Eighth Annual IEEE Applied Power Electronics Conference and Exposition (APEC)*, Mar. 2013, pp. 2526–2533.

[21] C.-S. Lim, E. Levi, M. Jones, N. 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 Transactions on Industrial Electronics*, vol. 61, no. 8, pp. 3867–3878, Aug. 2014.

[22] M. Preindl and E. Schaltz, "Sensorless model predictive direct current control using novel second-order PLL observer for PMSM drive systems," *IEEE Transactions on Industrial Electronics*, vol. 58, no. 9, pp. 4087–4095, Sep. 2011.

[23] 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 Conference and Exposition (APEC)*, Mar. 2013, pp. 2590–2597.

[24] 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 Transactions on Industrial Informatics*, vol. 10, no. 1, pp. 243–255, Feb. 2014.