

Combined Speed and Current Model Predictive Control for energy-efficient PMSM Drives

S.Bolognani**, *Student Member, IEEE*, S.Bolognani*, *Member, IEEE*,
L.Peretti**, *Member, IEEE*, M.Zigliotto**, *Member, IEEE*

* Department of Electrical Engineering, University of Padova, Italy

** Department of Technique and Management of Industrial Systems, University of Padova, Italy

Abstract—The paper deals with the Model Predictive Control (MPC) algorithm applied to control a permanent magnet synchronous motor, which is, at present, among the motors with the highest power efficiency and then very attractive for energy-saving applications. Absolute novelty of the proposed MPC is its feature of inherently managing the flux weakening control above base speed. Speed and current controller are combined together, including all the state variables of the system, instead of keeping the conventional cascade structure. This way it is possible to enforce in the controller the current and voltage limits. Simulation and experimental results point out the validity of the design procedure and the powerful capabilities of the MPC in the electrical drives field.

I. INTRODUCTION

Model Predictive Control (MPC) is based on a rather old approach whose first ideas have been published more than 20 years ago. Its strategies are characterized by an explicit and identifiable model of the controlled system. This model is used to precalculate the behavior of the plant and to choose an optimal value of the control variables [1], [2], [3], [4], [5].

MPC generally requires a significant computational effort. As the performance of the available computing hardware has rapidly increased and new faster algorithms have been developed, it is now possible to implement MPC to command fast systems with shorter time steps, as electrical drives.

Electric drives are of particular interest for the application of MPC for at least two reasons:

- 1) they fit in the class of systems for which a quite good linear model can be obtained both by analytical means and by identification techniques;
- 2) bounds on drive variables play a key role in the dynamics of the system; indeed, two main approaches are available to deal with systems constraints: anti-windup techniques, widely used in the classical PI controllers, and MPC. The presence of the constraint is one of the main reasons why, for example, state space controllers have limited application in electrical drives.

In spite of these advantages, MPC applications to electrical drives are still largely unexplored and only few research laboratories are involved on them. For example Generalized Predictive Control (GPC) – a special case of MPC – has been applied to induction motors for the only current regulation [6] and later for the speed and current control [7]. In [8], the more general MPC solution has been adopted for the design of the current controller in the same drive.

The main contribution of this paper is the design and the implementation of a MPC for a Permanent Magnet Synchronous

Motor (PMSM), which is, at present, among the motors with the highest power efficiency and then very attractive for energy-saving applications. Absolute novelty of the proposed MPC is its feature of inherently managing the flux weakening control above base speed. Speed and current controller are combined in a single MPC, instead of keeping the cascade structure of a speed controller and a current controller. In this way it is possible to enforce in the controller the constraints due to current and voltage limits.

The proposed approach allow the main advantages of MPC to be proved, i.e. its capability of systematically coping with hard constraints on inputs and states, its suitability to directly address multi-variable systems, and to perform an inherent FW management in AC drives.

The paper is organized as follows: in Section II an overview of the basic of MPC is provided. In Section III the model for the drive is presented. In Section IV the model predictive controller is designed. The algorithm that implements the controller is presented in Section V with its simulation. Experimental results are presented in Section VI.

II. BASICS OF MODEL PREDICTIVE CONTROL

MPC is a discrete-time algorithm in which the input sequence is chosen on the basis of the prediction of the future behaviour of the system state. More precisely, the controller chooses the input signal that minimize a given cost function of the state, generally an \mathcal{L}_2 norm.

As the controller has to predict the future system behaviour, the core of MPC is the model of the system. Consider then a discrete-time state-space model in the form

$$\mathbf{x}(k+1) = \mathbf{A}\mathbf{x}(k) + \mathbf{B}\mathbf{u}(k) \quad (1)$$

$$\mathbf{y}(k) = \mathbf{C}\mathbf{x}(k) + \mathbf{D}\mathbf{u}(k) \quad (2)$$

where the system variables \mathbf{x} , \mathbf{u} and \mathbf{y} satisfy the constraints

$$\mathbf{x} \in \mathbf{X} \subset \mathbb{R}^n, \mathbf{y} \in \mathbf{Y} \subset \mathbb{R}^p, \mathbf{u} \in \mathbf{U} \subset \mathbb{R}^m.$$

The cost function in its general form is

$$J_{N_p} = \mathbf{x}(k+N_p)^T \mathbf{P}\mathbf{x}(k+N_p) + \sum_{j=k}^{k+N_p-1} [\mathbf{x}(j)^T \mathbf{Q}\mathbf{x}(j) + \mathbf{u}(j)^T \mathbf{R}\mathbf{u}(j)] \quad (3)$$

where $\mathbf{Q} = \mathbf{Q}^T \geq 0$ weighs the state vector, $\mathbf{R} = \mathbf{R}^T > 0$ penalizes the control action and $\mathbf{P} = \mathbf{P}^T \geq 0$ weights the state value at the end of the prediction period N_p .

The problem of finding the best control input to be applied to the system can be transformed in a problem of *quadratic programming*, once some manipulations are applied to (3).

Iterative solvers are available in the technical literature on nonlinear programming and optimization methods (see [9], [10] and references therein) for this quadratic programming problem. Because of the constraints, an analytic solution is not available. However, the computational effort of solving this optimization problem at each time step can be greatly reduced, as explained later in Section V.

Once the optimal input sequence has been computed, only the first sample is applied to the plant, according to the *receding horizon policy*. The starting point of the optimal control scheme is periodically updated through feedback and the prediction horizon accordingly shifted in time, so that the control scheme sees a predicted behaviour which is naturally updated to account for the measured evolution of the system.

III. DRIVE MODEL

In order to design the MPC for the speed and current control of a PMSM, the drive has to be appropriately modelled. A synchronous reference frame (d, q), with d -axis fixed to the stator PM flux linkage vector is considered, as it allows all the electrical variables to be constant under steady state operation. The electrical dynamics can be thus described by the following stator equations, where the time dependency of the variables is understood, and iron saturation is assumed negligible:

$$\begin{aligned} \frac{di_d}{dt} &= \frac{1}{L} (u_d - Ri_d + \omega_{me}Li_q) \\ \frac{di_q}{dt} &= \frac{1}{L} (u_q - Ri_q - \omega_{me}Li_d - \omega_{me}\Lambda_{mg}) \end{aligned} \quad (4)$$

The mechanical dynamics is described by the following equation derived from the torque balance

$$\frac{d\omega_{me}}{dt} = \frac{p}{J} \left(k_t i_q - \frac{B}{p} \omega_{me} - \tau_L \right) \quad (5)$$

In (4) and (5), k_t is the motor torque constant given by $3p\Lambda_{mg}/2$ and ω_{me} is the mechanical speed in el.rad/s given by $p\omega_m$, where Λ_{mg} is the PM flux linkage, p is the number of pole pairs, and ω_m is the mechanical speed. In addition τ_L is the disturbance torque, J and B moment of inertia and viscous coefficient of the load and the other symbols are well known.

From (4) and (5), one can realize that the PMSM drive system is described by a non linear set of equations, even if electrical and mechanical parameters are constant, because of the motional coupling terms in each axis equations (4), involving the speed and the current of the other axis. Input voltages are limited by the inverter capabilities that produce a hexagonal feasible voltage region rotating in the $d - q$ voltage plane with rotational speed ω_{me} . For the sake of simplicity, a voltage limit approximated by a circle with radius $U_{dc}/\sqrt{3}$, where U_{dc} is the inverter DC bus voltage, can be assumed or, when more convenient, by a polygonal region fixed in the $d - q$ reference frame.

A further limit occurs on the current level, mainly for thermal reasons of the motor and inverter. Since the quadrature current prevails with respect to direct current, constant limits are here imposed for both the amplitude (absolute value) of i_d and i_q current components.

IV. MPC DESIGN

A. MPC state variables and discrete time model

The first step consists in determining a discrete time model for the system. If the model is a linear model with constraints, most of the optimization process can be moved off-line (see Section V). For this reason, an appropriate choice of the state variables has to be done to achieve linearity of the plant description.

As pointed out in Section III, the most important non-linearity of an PMSM drive is given by the coupling terms involving the speed and the currents i_d and i_q . In particular the effect of $\omega_{me}i_q$ on the state equation for i_d cannot be ignored. For delivering a linear model to the MPC, the $\omega_{me}i_q$ has been considered as a measured disturbances and therefore included in the model as an additional state variable.

The state variables of the system would then be the two components of the current in the d - and q - plane and the angular speed ω_{me} , together with the measured disturbance $\widehat{\omega_{me}i_d}$ and $\widehat{\omega_{me}i_q}$

$$\mathbf{x} = \left[i_d \quad i_q \quad \widehat{\omega_{me}i_d} \quad \widehat{\omega_{me}i_q} \quad \omega_{me} \right]^T$$

The dynamics of the measured disturbance $\widehat{\omega_{me}i_d}$ has been linearized around a fixed speed value equal to the base one while the dynamics of $\widehat{\omega_{me}i_q}$ has been linearized around $i_q = 0$ and $\omega_{me} = 0$, therefore having

$$\dot{\widehat{\omega_{me}i_q}} = 0$$

The effect of the cross coupling term $\omega_{me}i_d$ on the dynamic equation for i_q has instead been managed in a slightly different way. To keep the model linear, the term has been linearized around a constant speed value, which has been chosen equal to the base speed ω_B of the motor. In this way its effect on the dynamics of i_q has been roughly approssimated and, more important, flux weakening can be obtained in an automatic way. In fact, the value of i_d will be imposed by the MPC controller to satisfy the voltage limit even when the drive is working above the base speed.

The input signal is the two dimensional vector of the projections of the voltage \mathbf{u} on the d - and q - axes.

$$\mathbf{u} = [u_d \quad u_q]^T$$

According to this choice, the resulting discrete time model (obtained with forward Eulero discretization) is the following

$$\begin{aligned} \mathbf{x}(k+1) &= \\ &= T \begin{bmatrix} -\frac{R}{L} & 0 & 1 & 0 \\ -\omega_B & -\frac{R}{L} & 0 & -\frac{\Lambda_{mq}}{L} \\ 0 & 0 & 0 & 0 \\ 0 & \frac{pk_t}{J} & 0 & -\frac{B}{J} \end{bmatrix} \mathbf{x}(k) + \frac{T}{L} \begin{bmatrix} 1 \\ 1 \\ 0 \\ 0 \end{bmatrix} \mathbf{u}(k) \end{aligned} \quad (6)$$

where T is the discretization time step, which has to be chosen short enough to model correctly the fastest dynamic of the system (the currents one).

Load torque disturbance is ignored in writing (6), managing it as an unpredictable disturbance. Its rejection is achieved by the integral action that will be introduced later to achieve offset-free tracking.

B. State augmentation for tracking

As the objective of the design is tracking and not driving the system to zero, some modifications are needed to the state variables. First of all, to be able to weigh the speed error in the cost function, the reference ω_{me}^{ref} has to appear as a state of the system, assuming it is constant during the prediction horizon.

Moreover, as the cost function has to penalize the control signal, we may want to weigh its variation at each time step instead of its value, otherwise we would not be able to achieve output-free tracking. Transforming the input description in its differential form $\Delta \mathbf{u}$, where $\Delta \mathbf{u}$ is the new input, it's particularly straightforward once it has been discretized.

$$\begin{aligned} u_d(k) &= u_d(k-1) + \Delta u_d(k) \\ u_q(k) &= u_q(k-1) + \Delta u_q(k) \end{aligned}$$

Two additional state variables are then needed, which are the d - and q - components of the voltage at time step $k-1$, that is u_d^{k-1} and u_q^{k-1} . The new augmented state vector results to be

$$\mathbf{x}_{TR}(k) = \left[i_d \quad i_q \quad \widehat{\omega_{me} i_q} \quad \omega_{me} \quad \omega_{me}^{ref} \quad u_d^{k-1} \quad u_q^{k-1} \right]^T$$

and the new input vector is

$$\Delta \mathbf{u}(k) = [\Delta u_d \quad \Delta u_q]^T$$

Another minor modification is actually needed to have a more accurate model for the system. As the control input $\Delta \mathbf{u}$ won't drive the inverter before the next time step, a 1-step delay should be included in the model. It can be obtained removing the dependency of $\mathbf{x}_{TR}(k+1)$ on the input $\Delta \mathbf{u}(k)$, therefore having

$$\mathbf{x}_{TR}(k+1) = \begin{bmatrix} A & 0 & B \\ 0 & 1 & 0 \\ 0 & 0 & I \end{bmatrix} \mathbf{x}_{TR}(k) + \begin{bmatrix} 0 \\ 0 \\ I \end{bmatrix} \Delta \mathbf{u}(k) \quad (7)$$

where A and B are the matrix and the column vector that appear in (6).

C. Cost function and control law parameters

Once that a discrete time model for the system is available, the next important step is the definition of the cost function that has to be minimized by an appropriate choice of the control inputs at each time step. As pointed out in Section II, the cost function is a quadratic expression of the state in the form (3). In the specific setup of this problem, the main quantity that has to be weighted is the speed error $\omega_{me}^{ref} - \omega_{me}$; as i_d has to be zero when it is not needed for flux weakening, its value will be included in the cost function as well. Moreover, as the cost matrix for the system input signals has to be positive definite, a non-zero weights has to be chosen for the inputs Δu_d and Δu_q . There is an additional cost that has to be chosen, which is the weight for i_q . There is no real reason to penalize a large value of i_q , especially if the goal is to achieve reference tracking in presence of possible load disturbances. However, choosing a small, non-zero value for the weight of i_q plays a key role in the stability of the whole closed loop system [11]. For a very similar reason, a small weight has to be applied to the voltage u_d as well.

The matrices \mathbf{Q} and \mathbf{R} that appear in (3) are then

$$\mathbf{Q} = \begin{bmatrix} \gamma_{i_d} & 0 & 0 & 0 & 0 & 0 \\ 0 & \gamma_{i_q} & 0 & 0 & 0 & 0 \\ 0 & 0 & \gamma_{w_{me}} & -\gamma_{w_{me}} & 0 & 0 \\ 0 & 0 & -\gamma_{w_{me}} & \gamma_{w_{me}} & 0 & 0 \\ 0 & 0 & 0 & 0 & \gamma_{u_d} & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \quad (8)$$

$$\mathbf{R} = \begin{bmatrix} \gamma_{\Delta u_d} & 0 \\ 0 & \gamma_{\Delta u_q} \end{bmatrix} \quad (9)$$

where the γ s are the cost coefficients of the different quantities. The degree of freedom of tuning the matrix \mathbf{P} has not been exploited in this work, therefore considering it equal to zero.

These parameters have been set to the following values.

$$\begin{aligned} \gamma_{i_d} &= 1, \quad \gamma_{i_q} = 3, \quad \gamma_{w_{me}} = 600, \\ \gamma_{u_d} &= 8 \times 10^{-4} \gamma_{\Delta u_d} = 8 \times 10^{-7} \gamma_{\Delta u_q} = 8 \times 10^{-8} \end{aligned}$$

Another important parameter of the cost function is the length N_p of the prediction horizon, which has to be long enough to include the most relevant part of the system dynamics. On the other hand, a larger horizon always means a longer computational time and a more complex control algorithm. In the implementation presented in this paper, it has been chosen $N_p = 5$.

Together with these parameters related to the cost function, there is another parameter related instead to the control law. It is the control horizon N_u , that is the number of steps after which the input signal is assumed constant, mainly to have a faster optimization and a simpler algorithm. A quite common choice to obtain simple but usually effective controllers is to choose $N_u = 1$. That is, choosing the only next control input to apply to the plant. This option has been adopted in the paper.

D. State variable bounds

One of the most important features of MPC is its ability in automatically dealing with bounds on the state variables. To exploit this capability an accurate set of bounds have to be included in the model of the system. The two most important bounds in a PMSM drive are the current limit and the voltage limit. They both are two upper limits on the magnitude of \mathbf{i} and \mathbf{u} , that correspond to a circular limit in the d - q plane. However, as linear limits are strongly preferable for implementation purposes, two polygonal approximations of those limits have been adopted.

For the current limit a rectangular region has been enforced, which corresponds to a couple of independent limits on i_d and i_q , resulting in the two constraints

$$i_d \in [-I_{dN} \quad 0], \quad i_q \in [-I_{qN} \quad I_{qN}]$$

where I_{dN} and I_{qN} are the nominal (limit) stator currents of the drive. The upper limit for i_d has been set equal to zero as only negative direct current is required for flux weakening.

For the voltage limit, instead, a finer approximation has been used (because both u_d and u_q can be far from zero). The circular region $|\mathbf{u}| \leq U_N$ has been converted in an octagon, therefore enforcing 8 linear constraints.

$$\mathbf{u} \in \mathcal{P}_8 \subset \{|\mathbf{u}| \leq U_N\}$$

E. Integral action for offset-free tracking

A typical problem of MPC is the way it deals with unmeasured disturbances and parameter mismatch in the system model. In fact, as the control input that is applied to the system depends univocally on the current state and not on the past history of the system, there is no guarantee of steady-state zero offset in case of inexact modeling of the plant or of disturbances acting on it.

Therefore, an integrator of the angular speed error $\omega_{me}^{ref} - \omega_{me}$ has been added outside of the controller and its output has been used to move the reference ω_{me}^{ref} before it drives the MPC controller. In other words, instead of feeding the controller with the speed reference provided by the user, it is fed by

$$\bar{\omega}_{me}^{ref} = \omega_{me}^{ref} + K_{INT} \int (\omega_{me}^{ref} - \omega_{me}) dt$$

Integrator wind up can be avoided with the conventional tools that are commonly used in PI controllers.

V. IMPLEMENTATION AND ALGORITHM

The way MPC is implemented is crucial when it has to be applied to fast systems such as electrical drives. The approach presented in [8] consists in moving most of the optimization problem offline, keeping the complexity of the online part of the algorithm extremely low. This is possible computing an explicit solution of the control law [9].

In the case of a linear system with constraints and a quadratic cost function of the state, the optimization problem of finding the control values that minimize the given cost function is a *multi-parametric quadratic program*. The control law to be applied results to be a piecewise linear and continuous feedback of the state defined on a finite number of contiguous regions. More precisely, in each region of the state-space it is defined an affine control law in the form

$$\mathbf{u}_i(k) = \mathbf{F}_i \mathbf{x}(k) + \mathbf{G}_i \quad (10)$$

where i is the index of the *active region*, that is the convex region in which the state is at the sampling time k .

Applying this explicit control law to the system means to search the region in which the system state is located and to evaluate the affine feedback law in the form (10) to obtain the input to be applied to the system. The online part of the algorithm reduces then to be a region search and a matrix multiplication and sum, as one can see in Figure 1.

Another part of the optimization problem, namely the search of the active region, can be moved offline with the use of a binary search tree to find the active region [12]. The *Multi-Parametric Toolbox* (MPT) [10] for Matlab provides both a solver for the optimization problem and some routines to generate the region partition and the binary search tree for its simulation and implementation.

The developed MPC algorithm has been widely simulated in different operating conditions, before its actual implementation. The Multi-Parametric Toolbox allows both the simulation in Matlab and Simulink, and the implementation in C code. One of the simulation results is reported in Fig. 2, which shows the response of the drive to a pulse speed reference from 500 rpm to 1000 rpm. The parameters of the simulated drive are those of the test bed that has later been used for experimental validation,

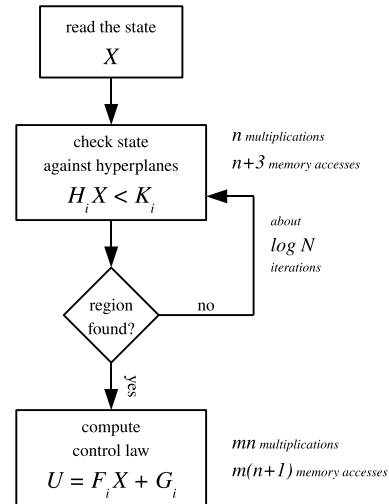


Fig. 1. Flow chart of the control algorithms with a rough estimation of the computational complexity of each step. N is the number of regions, n is the dimension of the state and m is the dimension of the input vector.

and they are reported in Section VI. The MPC has been designed to enforce a quadrature current limit of 6 A and a voltage limit of 173 V (DC bus at 300 V).

One can note the good speed response with limited overshoot and with no steady state error. Acceleration and deceleration are limited by the current limit that affects i_q , while i_d is well kept close to zero as flux weakening is not required. Voltage limits are not active in this operating conditions.

Figure 3 shows a more challenging operation of the drive. Starting from an initial speed lower than the base one, a sequence of three steps is applied. After the first step the speed of the motor is still below base speed, therefore no flux weakening occurs. This can be easily noted looking at the zero value of the direct current i_d . The next level of the speed is above base speed, thus a negative i_d is impressed to the motor to limit the voltage within the nominal value. The controller is then able to drive the motor with no steady state error even in the flux weakening region. The last level of the required speed cannot be achieved by the drive, as the flux weakening negative direct current reached its maximum value. A steady state speed error is then unavoidable.

The figure shows that the MPC controller is able to adjust all the state variables, including the flux, to drive the motor in all its feasible regions.

VI. EXPERIMENTAL RESULTS

The designed MPC for a PMSM drive has been experimentally validated on a laboratory testbench, sketched in Fig. 4. The testbench uses an SPM motor supplied by a three-phase inverter, the latter being controlled by a dSpace 1104 Fast Control Prototyping (FCP) board. Stator inductance of the motor is large enough to allow the motor to be operated above the base speed, in the flux weakening region. A wider flux weakening operating region can be obtained by additional, external inductances. A vector-controlled induction motor drive is used as mechanical load. The whole digital control of the SPM motor drive is implemented in the FCP board from the PWM generation to the speed/current model predictive control. The PWM and sampling frequencies are both 12 kHz. Main drive data are in Table I.

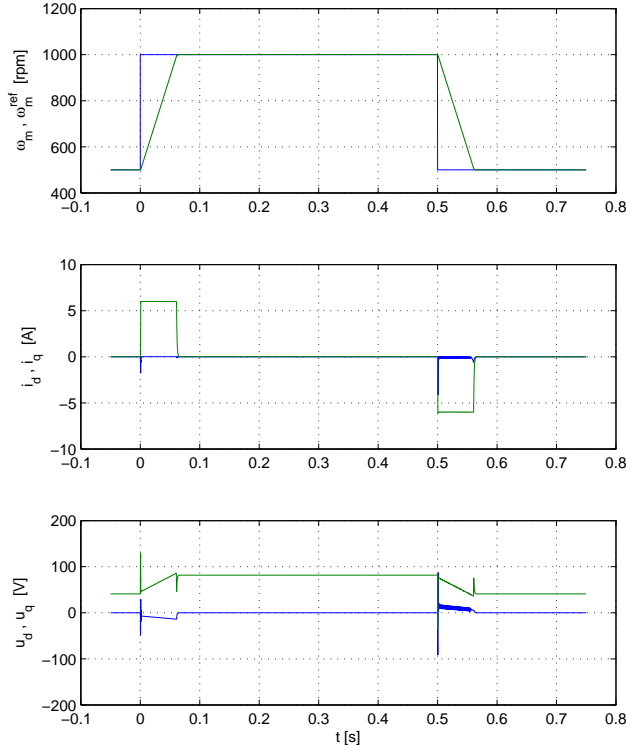


Fig. 2. Simulated drive response to a 500-to-1000 rpm reference speed pulse. From top to bottom: speed reference and response, d - and q - currents, and d - and q - voltages.

TABLE I
MAIN DRIVE DATA

Nominal torque	13.8	Nm
Nominal current	8.5	A rms
Base speed	2100	rpm
Phase resistance	0.8	Ω
Phase inductance	$6.5 \cdot 10^{-3}$	H
Pole pairs	3	
Total moment of inertia	$8.2 \cdot 10^{-3}$	kgm^2
DC bus voltage	300	V

Some experimental results are hereafter reported. At first the drive has been tested in the same conditions of the simulation reported in Fig. 2. The experimental results are reported in Fig. 5. As in the simulation, an i_q limit of 50% of the nominal current (the latter being $8.5\sqrt{2} = 12$ A) has been enforced to highlight its effects on the dynamics. More than one feedback law is used during the transient, as one can see from the indexes of the active region that are returned by the MPC controller and plotted at the bottom of the figure.

One can note the good overall performance of the drive and the tight correspondance between the experimental and simulated results. This confirms the reliability of simulation in predicting MPC behavior, which is the main tool for its best design. The same correspondance will be shown for operation in the flux weakening region, by some additional experimental results reported in the final paper.

In Fig. 6 a constant speed reference of 800 rpm is commanded,

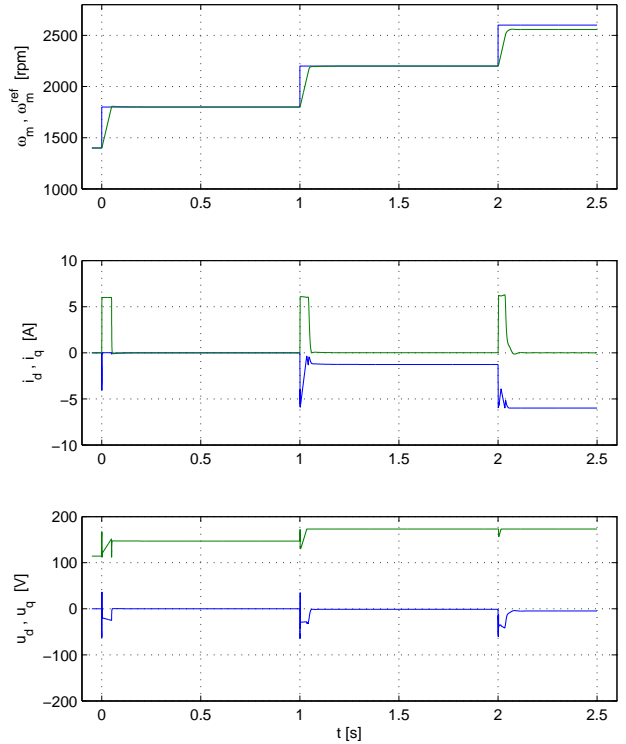


Fig. 3. Simulated drive response to a sequence of reference speed steps. From top to bottom: speed reference and response, d - and q - currents, and d - and q - voltages.

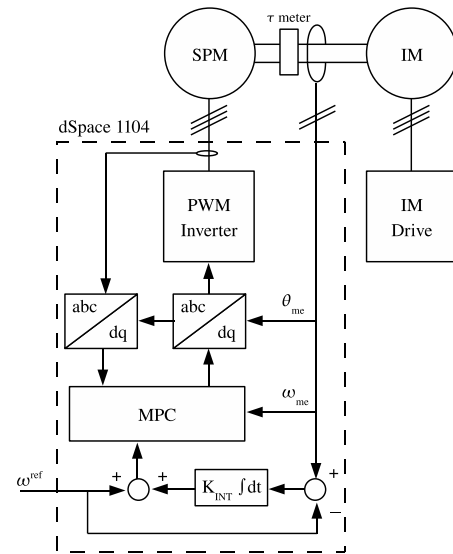


Fig. 4. Schematic of the laboratory test bench.

while the torque load goes from 20% to 40% and then 20% again of the nominal torque (13.8 Nm). The torque load is obtained by commanding the torque control of the induction motor drive connected to the drive under test. The torque measured by the torquemeter shows a slightly larger value due to additional friction torque at this speed. The figure shows a good capability of the MPC to limit the torque disturbance effects and to achieve

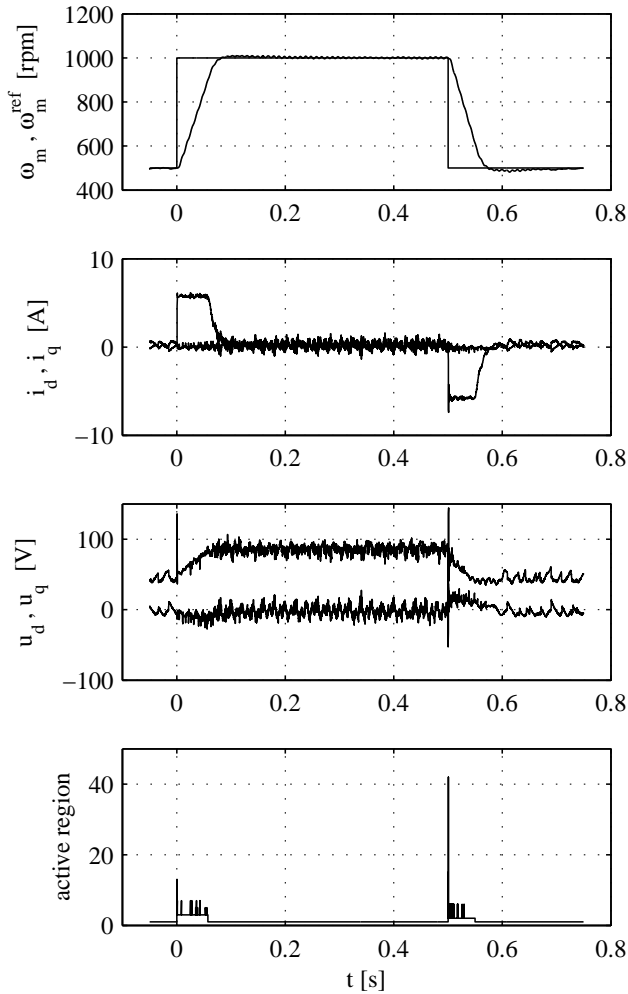


Fig. 5. Experimental drive response to a 500-to-1000 rpm reference speed pulse. From top to bottom: speed reference and response, d - and q - currents, d - and q - voltages, and the active region index.

zero steady state speed error. Max transient speed error is within 1.5% of the nominal drive speed.

VII. CONCLUSIONS

Permanent magnet synchronous motor drive are a very attractive solution for high-efficient motion control in industrial / commercial / residential applications. The paper gives a contribution for the best exploitation of this type of drive. Indeed, the combined current and speed model predictive controller proposed in the paper, is able to command the motor into both below and above the base speed, by an effective and inherent flux weakening. Details of the design procedure and of the controller implementation have been presented. Simulation and experimental results prove the promising performance of the proposed controller.

REFERENCES

- [1] D. W. Clarke, C. Mohtadi, and P. S. Tuffs, "Generalized predictive control – part I. the basic algorithm," *Automatica*, vol. 23, pp. 137–148, 1987.
- [2] —, "Generalized predictive control – part II. extensions and interpretations," *Automatica*, vol. 23, pp. 149–160, 1987.
- [3] D. W. Clarke and C. Mohtadi, "Properties of generalized predictive control," *Automatica*, vol. 25, pp. 859–875, 1989.
- [4] C. E. Garcia, D. M. Prett, and M. Morari, "Model Predictive Control: Theory and practice – a survey," *Automatica*, vol. 25, pp. 335–348, 1989.

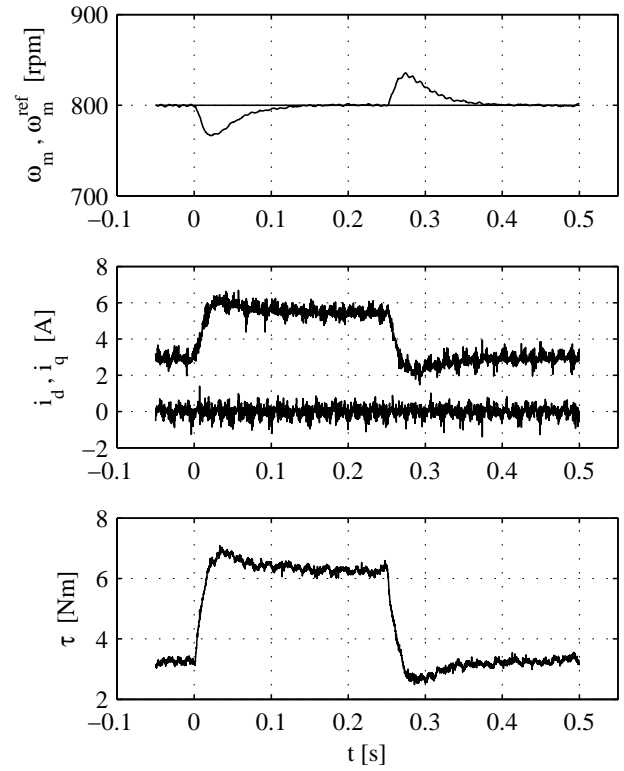


Fig. 6. Experimental drive response at 800 rpm to a 20% to 40% load torque. From top to bottom: speed reference and response, d - and q - current, and measured torque.

- [5] M. Morari and J. H. Lee, "Model Predictive Control: Past, present, and future," *Computers and Chemical Engineering*, vol. 23, no. 4–5, pp. 667–682, 1999.
- [6] L. Zhang, R. Norman, and W. Shepherd, "Long-Range Predictive Control of current regulated PWM for induction motor drives using the synchronous reference frame," *IEEE Trans. Control Syst. Technol.*, vol. 5, no. 1, pp. 119–126, 1997.
- [7] R. Kennel, A. Linder, and M. Linke, "Generalized predictive control (GPC) – ready for use in drive applications?" in *Proc. of the IEEE Power Electronics Specialists Conference PESC '01*, 2001, pp. 1839–1844.
- [8] A. Linder and R. Kennel, "Model predictive control for electrical drives," in *Proc. of the IEEE Power Electronics Specialists Conference PESC '05*, 2005, pp. 1793–1799.
- [9] A. Bemporad, M. Morari, V. Dua, and E. N. Pistikopoulos, "The explicit solution of model predictive control via multiparametric quadratic programming," in *Proc. of the American Control Conference*, Chicago, IL, 2000, pp. 872–876.
- [10] M. Kvasnica, P. Grieder, and M. Baotić, "Multi-Parametric Toolbox (MPT)," 2004. [Online]. Available: <http://control.ee.ethz.ch/mpt/>
- [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.*, submitted for acceptance.
- [12] P. Tøndel, T. A. Johansen, and A. Bemporad, "Evaluation of piecewise affine control via binary search tree," *Automatica*, vol. 39, pp. 945–950, 2003.