Model Predictive Control of a Permanent Magnet Synchronous Motor with Experimental Validation

Shan Chai[a,c], Liuping Wang[a], Eric Rogers[b]

[a] *School of Electrical and Computer Engineering, RMIT University, Melbourne, Victoria, Australia*
[b] *Electronics and Computer Science, University of Southampton, Southampton, SO17 1BJ, UK*
[c] *shan.chai@student.rmit.edu.au*

Abstract

This paper uses model predictive control to regulate the speed of a permanent magnet synchronous motor where the design is based on a linearized state space representation of the nonlinear model that describes the dynamics. An integrator is embedded into the design and hence zero steady-state error and the ability to reject load torque disturbance are guaranteed. The analysis includes operational constraints where a quadratic programming procedure is employed to solve the resulting online optimization problem. Experimental results obtained from application to an industry-size motor are also given with supporting discussion.
Keywords: embedded integrator, disturbance rejection, constraints, quadratic programming.

1. Introduction

Permanent Magnet Synchronous Motors (PMSMs) have found extensive use in industrial applications due to their high efficiency and dynamic performance. The standard controller for a PMSM is a vector control based cascade arrangement that uses Proportional plus Integral (PI) action as shown in Fig. 1. The two inner loop PI controllers are used to regulate the d-q axis current and the outer loop PI speed controller produces the q-axis current command for the inner loop q-axis current controller. This current command must be limited due to the constraints imposed by the motor characteristic and to protect the drive. A controller with constraints, is required since otherwise the closed-loop performance will deteriorate when the q-axis current command is saturated Shin (1998). In such cases, an anti-windup compensation algorithm, such as those discussed in, for example, Peng et al. (1996) and much recent work has been employed to improve controller performance.

An alternative to the cascade PI controller for electrical drives is Model Predictive Control (MPC), which is an optimization-based approach that computes current control action by minimizing difference between the predicted output of a system and its desired response. The many algorithms for MPC design can, in general terms, can be categorized as: i) transfer-function based, such as Generalized Predictive Control (GPC) Clarke et al. (1987), ii) step response model based, such as Dynamic Matrix Control (DMC) Cutler and Ramaker (1979), and iii) state-space model based Maciejowski (2002); Wang (2009) where the optimization can be completed by solving a Quadratic Programming (QP) problem. In absence of constraints, it is possible to obtain the optimal solution in an analytic form and hence the feedback gain can

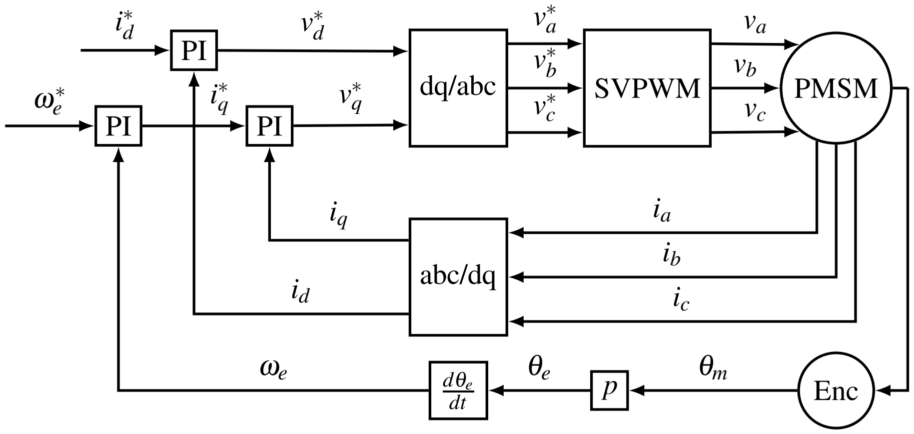


Figure 1: Schematic of the cascade PI controller for a PMSM.

be pre-calculated off-line. However, MPC can also be applied where the QP problem is solved on-line under different system operational constraints, as required in many industrial applications.

Some problems in, for example, the process industries can allow a relatively long sample time to solve the QP problem on-line and this is the major reason why MPC is well established in some areas. In contrast, MPC has been less commonly used in electric drive and power electronics applications due to the fast sampling requirement and the computational load imposed by the QP. More recently the development of faster micro-controllers and advances in research have made MPC a feasible control strategy for application to electric drives, see, for example, de Santana et al. (2008) for induction motors and Bolognani et al. (2009) Ishida et al. (2010) PMSMs.

The main reasons for applying MPC to electric drives are: i) the model for the electrical parts of a PMSM or an induction motor can be linearized as a Multi-Input and Multi-Output (MIMO) system, ii) the phase currents and voltages of motor drives are usually limited by electrical constraints and iii) MPC can be extended to include periodic disturbances Mariethoz et al. (2009) and the completion of repetitive tasks Wang et al. (2010).

The MPC design in Bolognani et al. (2009) for electric motor drives has to have an extra integrator implemented outside the controller to eliminate the steady-state error that arises under a load torque disturbance. In the new design developed in this paper, an integrator is directly embedded into the design of MPC and hence zero steady-state error is guaranteed in the presence of a load torque disturbance. This new approach is based on the MPC design of Wang (2009), with implementation using Hildreth's Quadratic Programming Hildret (1957), for AC motor control. The development of the new design is supported by experimental results from application to an industry-size motor, including description of implementation including the encoder, the electrical tests for parameter identification and model validation.

The next section gives the model of a PMSM motor and describes how the linearized model for MPC design is constructed. Section 3 gives the necessary background on MPC and the control law and Section 5 the parameter identification and model validation. Section 6 details the experiment setup using the MATLAB

XPC host-target environment and Section 7 the experimental results obtained. Section 8 concludes the paper.

2. Electrical Model of a PMSM

The most commonly considered model of a PMSM takes the following form in the d-q rotating reference frame Pillay and Krishnan (1988)

$$
\begin{aligned}
&\tfrac{di_d}{dt} = \tfrac{1}{L_d}(v_d - Ri_d + \omega_e L_q i_q)\\
&\tfrac{di_q}{dt} = \tfrac{1}{L_q}(v_q - Ri_q - \omega_e L_d i_d - \omega_e \phi_{mg})\\
&\tfrac{d\omega_e}{dt} = \tfrac{p}{J}(T_e - \tfrac{B_v}{p}\omega_e - T_L)\\
&T_e = \tfrac{3}{2}p[\phi_{mg} i_q + (L_d - L_q) i_d i_q]
\end{aligned}
\tag{1}
$$

where ω_e is the electrical speed, related to the rotor speed by $\omega_e = p\omega_m$ where p denotes the number of pole pairs, v_d and v_q represent the stator voltages in the d-q frame and i_d and i_q the stator currents in this frame, respectively. The symbol T_L denotes the load torque that is treated as a slowly varying disturbance to be rejected. Moreover, electromagnetic torque, denoted by T_e, consists of two parts: that produced by the flux of the permanent magnet ϕ_{mg} and that by i_d and i_q, respectively. In the control of a PMSM, the set-point signal for the d-axis current is set to zero for energy efficiency Pillay and Krishnan (1988) and in the steady-state the output signal $i_d = 0$. For a surface mounted PMSM, the d-q axis inductance satisfies $L_d = L_q$. In either case, the electromagnetic torque T_e and i_q are linearly related through

$$T_e = \frac{3}{2} p \phi_{mg} i_q \tag{2}$$

Maintaining $i_d = 0$ allows the electromagnet torque T_e to be controlled through i_q directly, which is also the underlying idea of field oriented control. The cross-coupling terms $\omega_e i_q$ and $\omega_e i_d$ in (1) are the nonlinear elements and therefore have to be linearized in order to use linear MPC. The most common approach is to linearize these elements around an operating point and in this paper the operating points of the state variables i_d, i_q and ω_e in (1) are denoted by i_{d0}, i_{q0} and ω_{e0}, respectively. The bilinear terms are approximated at the operating points using first order Taylor series expansion as,

$$\omega_e(t) i_q(t) \approx \omega_{e0} i_{q0} + i_{q0}(\omega_e(t) - \omega_{e0}) + \omega_{e0}(i_q(t) - i_{q0}) \tag{3}$$

$$\omega_e(t) i_d(t) \approx \omega_{e0} i_{d0} + i_{d0}(\omega_e(t) - \omega_{e0}) + \omega_{e0}(i_d(t) - i_{d0}) \tag{4}$$

Substituting these approximations into (1), gives the linearized model

$$
\begin{aligned}
\dot{x}_m &= \mathbf{A_m} x_m + \mathbf{B_m} u + \varepsilon\\
y &= \mathbf{C_m} x_m
\end{aligned}
\tag{5}
$$

with

$$\mathbf{A_m} = \begin{bmatrix} -\frac{R}{L_d} & \frac{L_q}{L_d}\omega_{e0} & \frac{L_q}{L_d}i_{q0} \\ -\frac{L_d}{L_q}\omega_{e0} & -\frac{R}{L_q} & -(\frac{L_d}{L_q}i_{d0}+\frac{\phi_{mg}}{L_q}) \\ 0 & \frac{3p^2\phi_{mg}}{2J} & -\frac{B_v}{J} \end{bmatrix}, \mathbf{B_m} = \begin{bmatrix} \frac{1}{L_d} & 0 \\ 0 & \frac{1}{L_q} \\ 0 & 0 \end{bmatrix}, \mathbf{C_m} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

$$\varepsilon = \begin{bmatrix} -\frac{L_q}{L_d}\omega_{e0}i_{q0} \\ \frac{L_d}{L_q}\omega_{e0}i_{d0} \\ -\frac{pT_L}{J} \end{bmatrix}$$

where $\varepsilon \in \mathbb{R}^3$ denotes the slowly varying disturbance vector that is assumed to have constant entries with respect to the operating point where linearization is applied, $x_m = \begin{bmatrix} i_d & i_q & \omega_e \end{bmatrix}^T \in \mathbb{R}^3$ is the state vector, $y = \begin{bmatrix} i_d & \omega_e \end{bmatrix}^T \in \mathbb{R}^2$ is the output vector and $u = \begin{bmatrix} v_d & v_q \end{bmatrix}^T$ is the input vector. The model parameters and their numerical values used to obtain the experimental results included in this paper are given in Table 1. The control objective is to achieve the desired speed set-point whilst maintaining the i_d component at zero and the i_q component within operational limits.

Symbol	Description	Value
p	no. of pole pairs	2
J	total moment of inertia	2.35 ($Kg \cdot cm^2$)
B_v	viscous coefficients	1.1e-4
L_d, L_q	d-q axis inductance	7.0 (mH)
R	Stator resistance	2.98 (Ohm)
ϕ_{mg}	rotor flux	0.125 (Weber)

Table 1: PMSM parameters.

To digitally implement MPC, the continuous-time model (5) is discretized with a time step T_s using the zero-order hold method, resulting in

$$\begin{aligned} x_m(k+1) &= \mathbf{A_d}x_m(k) + \mathbf{B_d}u(k) + d \\ y(k) &= \mathbf{C_d}x_m(k) \end{aligned} \tag{6}$$

where d is a constant vector resulting from the discretization of ε and

$$\mathbf{A_d} = e^{\mathbf{A_m}T_s}, \ \ \mathbf{B_d} = (\int_0^{T_s} e^{A_m\tau}d\tau)\mathbf{B_m}, \ \ \mathbf{C_d} = \mathbf{C}_m$$

3. MPC Design

The MPC design used in this paper is from Wang (2009) and this section summarizes the main steps involved for an m-input, p-output and n-state model where in the experimental results section the case of speed regulation for a PMSM is considered, for which $m = 2$, $p = 2$, and $n = 3$.

In common with the PI controller, an integrator is usually embedded into an MPC design, where in the MIMO application considered the embedded integrator in each channel is also used to eliminate the constant vector d in (6) or ε in (5), which contain motor parameters that have a certain degree of uncertainty associated with them in applications. Once the integrators have been added to the model, the optimization analysis is applied to the model expressed in terms of incremental, rather than the actual variables. One advantage of using the incremental model is that the steady-sate values of the state variables are not required in the implementation. Moreover, use of the incremental variables removes the constant terms (represented by the vector d) in the linearized model state equations from subsequent analysis.

Taking the difference between the current and next sample instances of the state-space model (6), gives incremental state dynamics as

$$\Delta x_m(k+1) = \mathbf{A_d}\Delta x_m(k) + \mathbf{B_d}\Delta u(k) \tag{7}$$

where $\Delta x_m(k) = x_m(k) - x_m(k-1)$, $\Delta u(k) = u(k) - u(k-1)$ and $d(k) - d(k-1) = 0$. Applying the same procedure to the output vector gives

$$\begin{aligned} y(k+1) - y(k) &= \mathbf{C_d}(x_m(k+1) - x_m(k)) \\ &= \mathbf{C_d}\Delta x_m(k+1) \\ &= \mathbf{C_d}\mathbf{A_d}\Delta x_m(k) + \mathbf{C_d}\mathbf{B_d}\Delta u(k) \end{aligned}$$

or

$$y(k+1) = \mathbf{C_d}\mathbf{A_d}\Delta x_m(k) + y(k) + \mathbf{C_d}\mathbf{B_d}\Delta u(k) \tag{8}$$

Hence the augmented state-space model with the integrators embedded is

$$\begin{aligned} x(k+1) &= \mathbf{A}x(k) + \mathbf{B}\Delta u(k) \\ y(k) &= \mathbf{C}x(k) \end{aligned} \tag{9}$$

where

$$x(k) = \begin{bmatrix} \Delta x_m(k) \\ y(k) \end{bmatrix}$$

and

$$\mathbf{A} = \begin{bmatrix} \mathbf{A_d} & 0 \\ \mathbf{C_d}\mathbf{A_d} & I \end{bmatrix}, \mathbf{B} = \begin{bmatrix} \mathbf{B_d} \\ \mathbf{C_d}\mathbf{B_d} \end{bmatrix}, \mathbf{C} = \begin{bmatrix} 0 & I \end{bmatrix}$$

and for the remainder of this paper 0 and I, respectively, denote the zero and identity matrices of compatible dimensions.

Control law design requires the predicted future output based on the model and current state of system.

The state vectors from one to N_p instants in the future are given by

$$\begin{cases} x(k+1|k) = \mathbf{A}x(k) + \mathbf{B}\Delta u(k) \\ x(k+2|k) = \mathbf{A}^2 x(k) + \mathbf{AB}\Delta u(k) + \mathbf{B}\Delta u(k+1) \\ \qquad \vdots \\ x(k+N_p|k) = \mathbf{A}^{N_p} x(k) + \mathbf{A}^{N_p-1}\mathbf{B}\Delta u(k) \\ \qquad + \cdots + \mathbf{A}^{N_p-N_c}\mathbf{B}\Delta u(k+N_c-1) \end{cases} \tag{10}$$

where N_p and N_c are termed the prediction and control horizons, respectively. Also the choice $N_c \leq N_p$ assumes that the incremental control Δu has reached the steady-state after N_c instants. Therefore, the predicted outputs for the next N_p instants can be written as

$$Y = Fx(k) + \Phi\Delta U \tag{11}$$

where

$$\begin{aligned} Y &= \begin{bmatrix} y^T(k+1) & y^T(k+2) & \dots & y^T(k+N_p) \end{bmatrix}^T \\ \Delta U &= \begin{bmatrix} \Delta u^T(k) & \Delta u^T(k+1) & \dots & \Delta u^T(k+N_c-1) \end{bmatrix}^T \\ F &= \begin{bmatrix} (\mathbf{CA})^T & (\mathbf{CA}^2)^T & \dots & (\mathbf{CA}^{N_p})^T \end{bmatrix}^T \\ \Phi &= \begin{bmatrix} \mathbf{CB} & 0 & \cdots & 0 \\ \mathbf{CAB} & \mathbf{CB} & \cdots & 0 \\ \vdots & \vdots & \ddots & 0 \\ \mathbf{CA}^{N_p-1}\mathbf{B} & \mathbf{CA}^{N_p-2}\mathbf{B} & \cdots & \mathbf{CA}^{N_p-N_c}\mathbf{B} \end{bmatrix} \end{aligned}$$

The general form of the cost function is

$$J = (R_s - Y)^T Q (R_s - Y) + \Delta U^T \bar{R} \Delta U \tag{12}$$

where R_s is a reference vector whose elements are assumed to be constant within the prediction horizon, Q is a $(p \times N_p) \times (p \times N_p)$ symmetric positive semi-definite matrix, and $\bar{R}$ is a $(m \times N_c) \times (m \times N_c)$ symmetric positive definite matrix. For the PMSM model (6), Q is given by a block diagonal matrix with diagonal block entries Q_{id} and Q_{ω} of the form

$$Q = \begin{bmatrix} Q_{id} & 0 & 0 & \cdots & \cdots & 0 \\ 0 & Q_{\omega} & 0 & \ddots & \cdots & \vdots \\ 0 & 0 & \ddots & \ddots & \ddots & \vdots \\ \vdots & \ddots & \ddots & \ddots & \ddots & 0 \\ \vdots & \vdots & \ddots & \ddots & Q_{id} & 0 \\ 0 & \cdots & \cdots & 0 & 0 & Q_{\omega} \end{bmatrix}$$

In terms of speed regulation, Δu_d and Δu_q are treated as equally important and, with r_w denoting the weighting, R has the form

$$\bar{R} = \begin{bmatrix} r_w & 0 & 0 & \cdots & \cdots & 0 \\ 0 & r_w & 0 & \ddots & \cdots & \vdots \\ 0 & 0 & \ddots & \ddots & \ddots & \vdots \\ \vdots & \ddots & \ddots & \ddots & \ddots & 0 \\ \vdots & \vdots & \ddots & \ddots & r_w & 0 \\ 0 & \cdots & \cdots & 0 & 0 & r_w \end{bmatrix}$$

The receding horizon principle states that a N_c control sequence is obtained by optimizing the cost function (12) but only the first control move is applied to the plant. In the absence of constraints the global optimal solution is given by

$$\frac{\partial J}{\partial \Delta U} = 0$$

and solving this equation leads to global optimal control sequence

$$\Delta U = (\Phi^T Q \Phi + \overline{R})^{-1} \Phi^T Q (\overline{R_s} r(k) - F x(k))$$

Moreover, the first control move is

$$\Delta u(k) = \overbrace{\begin{bmatrix} I & 0 & \cdots & 0 \end{bmatrix}}^{N_c \times m} \Delta U$$

$$\Delta u(k) = K_y r(k) - K_x x(k) \tag{13}$$

where K_y is formed by taking the first m rows of the matrix

$$(\Phi^T Q \Phi + \overline{R})^{-1} \Phi^T Q \overline{R_s}$$

and K_x by taking the first m rows of the matrix

$$(\Phi^T Q \Phi + \bar{R})^{-1} \Phi^T Q F$$

The matrix $\bar{R}_s = \overbrace{[I\, I\, \ldots I]}^{p \times N_p}{}^T$ occurs in last p columns of the matrix F due to the special structure of the augmented state-space model matrices $\mathbf{C}$ and $\mathbf{A}$. Consequently, K_y is given by the last p columns of the matrix K_x and, by using the relationship between K_x and K_y, the control law (13) can be reduced to

$$\begin{aligned} \Delta u(k) &= K_y r(k) - K_x \begin{bmatrix} \Delta x_m(k) \\ y(k) \end{bmatrix} \\ &= -K_x \begin{bmatrix} \Delta x_m(k) \\ y(k) - r(k) \end{bmatrix} \end{aligned} \tag{14}$$

and the control law applied is

$$u(k) = u(k-1) + \Delta u(k) \tag{15}$$

In the absence of constraints, the optimal feedback gain K_x can be pre-calculated off-line in MATLAB, and the optimal control $u(k)$ obtained online using (14) and (15).

4. MPC with constraints

Many different types of system operational constraints often arise in the application of control algorithms to physical systems. In the case of a PMSM, the two input voltages to the motor, v_d and v_q, respectively, are limited by the DC bus voltage. Moreover, the maximum voltage that can be modulated by the Pulse Width Modulator (PWM) is $V_{dc}/\sqrt{3}$. In the PI control structure, an anti-windup scheme, such as those discussed in, for example, Peng et al. (1996) and many more recent publications, is usually employed to compensate for constraints, but the success of MPC with constraints in, for example, process control applications makes it a natural alternative to be considered.

The cost function of the previous section was optimized with respect to ΔU. Hence any constraints on, for example, the control input and state variables have to be translated into constraints on the incremental control variable Δu. One relevant set for the PMSM is given by (16) and (17), respectively. These constraints are the amplitudes of the d-axis and q-axis voltages and their incremental changes and are caused by the physical limits of the motor but also provide protection to the electronic hardware.

$$C_1(u_{min} - u(k_i - 1)) \leq C_2 \Delta U \leq C_1(u_{max} - u(k_i - 1)) \tag{16}$$

$$C_1 \Delta u_{min} \leq \Delta U \leq C_1 \Delta u_{max} \tag{17}$$

$$\text{where} \quad C_2 = \left.\overbrace{\begin{bmatrix} I & 0 & \dots & 0 \\ I & I & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ I & I & \dots & I \end{bmatrix}}^{N_c \times m}\right\} N_s \times m$$

$$\text{and } C_1 = \overbrace{\begin{bmatrix} I & I & \dots & I \end{bmatrix}^T}^{N_s \times m}$$

Let N_s denote the number of the control moves with these constraints imposed over the control horizon N_c. Then by the receding horizon principle, these constraints can only be realized on the first control move, which is the current time. Although future constraints on the control moves will affect the solution of optimal control parameter vector ΔU, if they become active, this effect is often small in applications Wang (2009). Conversely, if all future control moves were constrained by choosing $N_s = N_c$, the computational load would increase and could become expensive. Hence it is preferable in practical applications that the parameter N_s is treated as a tuning parameter depending on the step-time of the micro-controller.

In the MPC literature, the constrained optimization problem is termed QP, see for example, Boyd and Vandenberghe (2004), Luenberger (1984), which is essentially to minimize the cost function (12) subject to a set of linear inequality constraints: $M\Delta U \leq \gamma$. One of the methods available is based on an on-line search of the active constraints using Hildreth's QP algorithm Hildret (1957); Wang (2009). In this approach, the objective function (12) is replaced by one with the Lagrange multipliers as the decision variables and the constrained optimization problem is solved subject to the Lagrange multiplier vector $\lambda \geq 0$:

$$\min_{\lambda \geq 0} (\frac{1}{2}\lambda^T H \lambda + \lambda^T K + \frac{1}{2}\gamma^T E^{-1}\gamma) \tag{18}$$

where $H = M(\Phi^T Q \Phi + \overline{R})^{-1} M^T$ and $K = \gamma + M(\Phi^T Q \Phi + \overline{R})^{-1} F$. Once the positive Lagrange multipliers are found, the optimal ΔU vector is given in closed from by Wang (2009)

$$\begin{aligned} \Delta U = & (\Phi^T Q \Phi + \overline{R})^{-1} \Phi^T Q (\overline{R_s} r(k) - F x(k)) \\ & - (\Phi^T Q \Phi + \overline{R})^{-1} M_{act}^T \lambda_{act} \end{aligned} \tag{19}$$

where M_{act} and λ_{act} are the active linear constraint matrix and Lagrange multiplier vector, respectively.

5. Parameter Identification and Model Validation

As MPC is a model based control strategy, an adequate model reflecting the motor characteristics is necessary for controller design to achieve, or get acceptably close to, desired closed-loop performance. The dynamic models for a PMSM are derived for both electrical and mechanical components Vas (1992).

The mechanical parameters J and B in Table 1 were obtained from the constant acceleration and velocity regions of an open-loop test, respectively and next how the electrical parameters in this table were obtained is detailed.

The inductance can be measured by conducting the block rotor test with a 50Hz 3-phase voltage applied to the motor through a step-down transformer. As the rotor is blocked, the back-EMF voltage induced by the rotating rotor does not exist. Hence the phase inductance L_s can be calculated Vas (1993) using

$$|\mathbf{V_s}| = |R + j\omega_e L_s||\mathbf{I_s}|$$

where $|\mathbf{V_s}|$ and $|\mathbf{I_s}|$ are the amplitudes of the phase voltage and current vectors, respectively. The amplitude of the phase voltages are related to the measured line-to-line voltage (rms value) by,

$$|\mathbf{V_s}| = \frac{\sqrt{2}}{\sqrt{3}} V_{l-l(rms.)}$$

and the resistance R of stator can be measured by a DC test. For a non-salient (surface mounted) machine considered in the remainder of this paper, the d-q axis inductance equals the phase inductance and hence $L_d = L_q = L_s$.

The flux linkage can be measured by coupling the PMSM motor to another motor running at a constant speed ω_m. In this arrangement the second PMSM motor works as a generator, and the flux linkage can be computed by measuring the line-to-line voltage and rotation speed using

$$|\mathbf{V_s}| = \omega_e \phi_{mg} = p\ \omega_m \phi_{mg}$$

The identified parameters for the motor used in this research, with further details in the next section, have been validated by comparing the open-loop responses of a simulation against those experimentally measured, where the former have been developed in MATLAB/SIMULINK using the mathematical model (1). In the experimental validation, the desired input voltage, denoted by v_d^* and v_q^*, are implemented by dq/abc transformation and Space Vector Pulse Width Modulated (SVPWM) control of a two level voltage source inverter, see Figure 2. In the linear modulation range of SVPWM with sufficiently high switching frequency, it is assumed that $v_d^* \approx v_d$ and $v_q^* \approx v_q$. Figure 3 and Figure 4 give representative results from the model validation tests for the dq-axis current and the velocity, showing a high degree of agreement between simulated and predicted values in each case.

6. Control Experiment setup

The experiments were conducted using the MALTAB realtime workshop and XPC host-target environment. The host PC has the MATLAB real-time workshop compiler installed and the target PC with the XPC

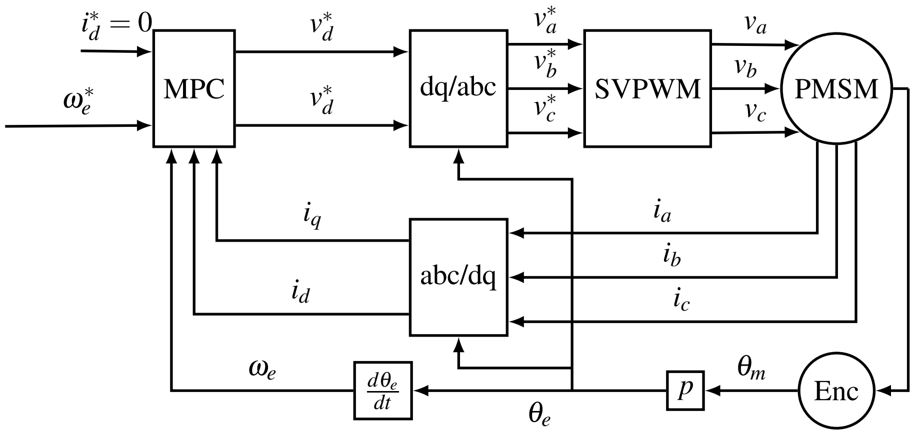
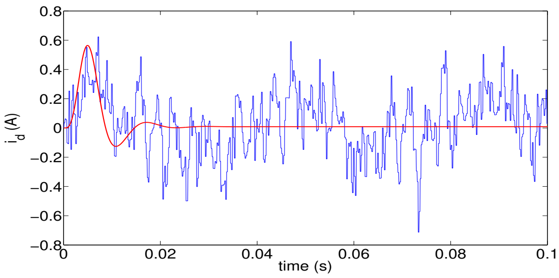
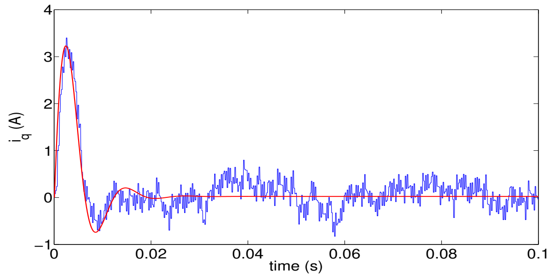


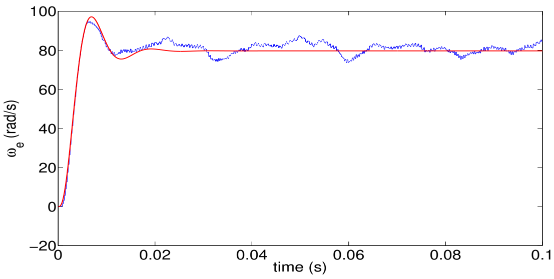
Figure 2: MPC speed control of the PMSM.



(a) i_d

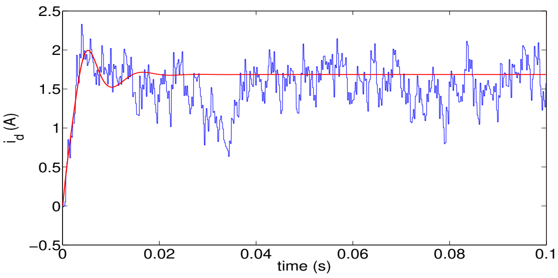


(b) i_q

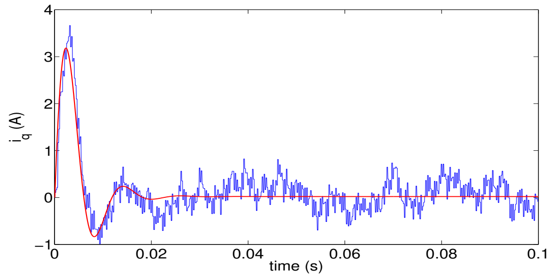


(c) ω_e

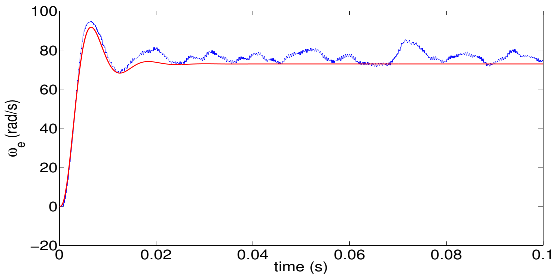
Figure 3: Model Validation
($v_d = 0$ (v) and $v_q = 20$ (v))



(a) i_d



(b) i_q



(c) ω_e

Figure 4: Model Validation
($v_d = 5$ (v) and $v_q = 20$ (v)).

target kernel and the quadrature encoder card. The model is developed in MATLAB/SIMULINK and downloaded to the XPC target. This setup allows rapid prototyping and hardware-in-loop simulation functionality using PC hardware.

To test the performance of the MPC design under a change of load disturbance, the second PMSM is coupled to the controlled PMSM through a rigid coupler, as shown in Figure 5. The generated phase voltages from the second PMSM are fed to resistor loads through an uncontrolled three-phase rectifier. With this setup, the equivalent load toque delivered by the electrical load is proportional to the speed of the motor shaft.

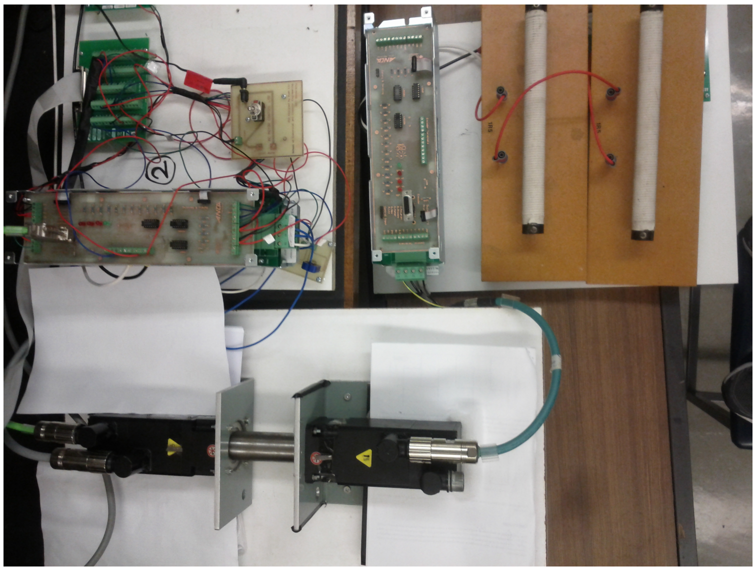


Figure 5: PMSM and loads.

Rotor position information is important in order to acquire the feedback speed and commutation angle for both the abc/dq and dq/abc transformations. Hence a sin-cosine optical encoder with 512 cycles/rev was equipped to acquire high resolution position information. As output, the encoder produces sine and cosine signals that are converted to channel A and B pulses, respectively, as shown in Figure 6. A quadrature counter card was employed to count both the rising and falling edges of a square wave and the position obtained using Staebler (2000)

$$\theta = \frac{2\pi}{N}(cnt >> 2 + \frac{\phi}{2\pi}) \tag{20}$$

where N is number of cycles per revolution, cnt denotes the number of counts obtained by the quadrature counter card and ϕ is the phase angle determined from sampled values of both the sine, denoted by A and cosine, denoted by B, signals using

$$\phi = \begin{cases} \arctan(\frac{A}{B}), & A \geq 0 \\ \arctan(\frac{A}{B}) + \pi, & A < 0 \end{cases}$$

The sampling time of the sine-cosine signal may not be synchronized with the counter card and hence

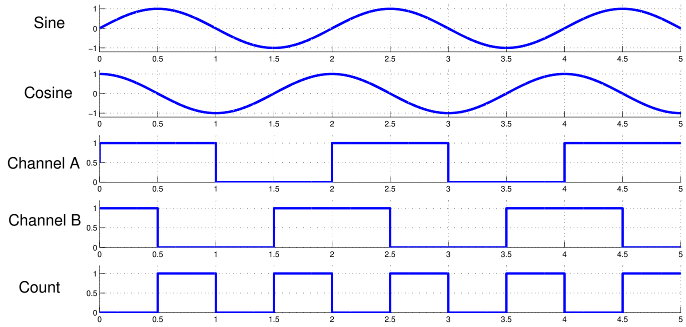


Figure 6: Encoder signal.

sudden variation in the position calculated by (20) can arise. There are two scenarios Staebler (2000): a) $0 \leq \phi \leq \frac{\pi}{2}$ and $cnt\%4 = 3$ and b) $\frac{3}{2}\pi \leq \phi \leq 2\pi$ and $cnt\%4 = 0$. Case a) arises because sampling ϕ leads to the counter increasing and it should be increased by 1 to compensate for the mismatch. Conversely, in case b) sampling of ϕ lags the counter increasing and the counter value should be decreased by 1.

A common method to obtain velocity is to calculate the derivative of a position measurement and one discrete implementation is by backward difference as

$$\omega_m(k) = \frac{\theta(k) - \theta(k-1)}{T_s}.$$

but this would produce significant variation in the value obtained due to quantization error of the position measurements. An alternative approach is to use the filter derivative of position measurement by passing $\theta(t)$ through the filter with transfer-function

$$F(s) = \frac{s}{\alpha s + 1} \tag{21}$$

to obtain $\omega_{mf}(t)$, with discrete implementation by the backward Euler method as

$$\omega_{mf}(k) = \frac{\theta(k) - \theta(k-1) + \alpha\omega_{mf}(k-1)}{\alpha + T_s} \tag{22}$$

Moreover, (22) can be directly implemented to obtain the velocity for use by the MPC controller. The choice of α is a trade-off between filtering noise and the lags introduced. Another approach is to include filtered velocity as a state variable Bolognani et al. (2009) by passing $\omega_m(t)$ through the filter with transfer-function

$$G(s) = \frac{1/\alpha}{s + 1/\alpha} \tag{23}$$

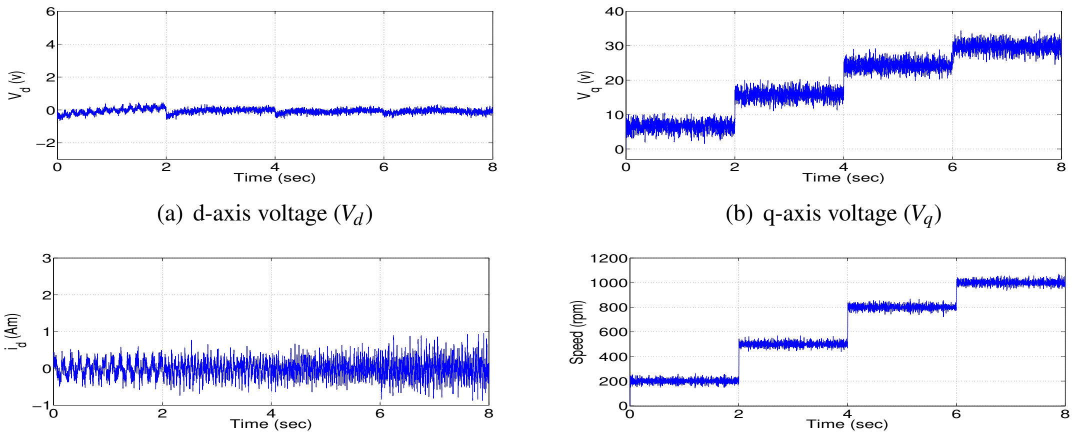
to obtain $\omega_{mf}(t)$ and hence the effects of filter can be embedded into the MPC design.

7. Experiment Results and Discussion

In this section, the experimental results for three different cases are given and discussed. In particular, the performance of the MPC controller is analyzed in terms of the constraints, tuning of controller feedback gain and load disturbance rejection.

7.1. Response to a staircase reference

Figure 7 shows the performance of MPC controller in response to a staircase reference speed. The control objective is to regulate the motor to a different reference speed with i_d maintained at zero. The controller gain for this case was selected to obtain a fast response whilst minimizing the noise level. Figure 7(c) shows that the steady-state d-axis current is zero but corrupted by the noise and high frequency ripples. The high frequency ripples appear as pulsing torque Jahns and Soong (1996) in the closed-loop system due to various reasons, such as current measurement errors Chung and Sul (1998), flux harmonics and cogging torques Xu et al. (2004). It is also seen from Figure 7(c) that the frequency of oscillation increases as the synchronous speed increases.



(a) d-axis voltage (V_d) (b) q-axis voltage (V_q)

(c) d-axis current (i_d) (d) motor speed

Figure 7: Performance of the MPC controller in response to a staircase reference speed.

7.2. Constraints

The maximum voltages in the servo drive are determined by the DC bus voltage (V_{dc}) and the pulse width modulation techniques adopted to prevent over-modulation Hava et al. (1998) occurring. Particular to the SVPWM technique employed in this work, the linear modulation range limits amplitudes of maximum achievable voltage by

$$\sqrt{v_d^2 + v_q^2} \leq \frac{V_{dc}}{\sqrt{3}}. \tag{24}$$

and this limitation can be imposed in favor of V_q priority as

$$|v_q| \leq \varepsilon \frac{V_{dc}}{\sqrt{3}}, \quad |v_d| \leq \sqrt{(1-\varepsilon^2)}\frac{V_{dc}}{\sqrt{3}}.$$

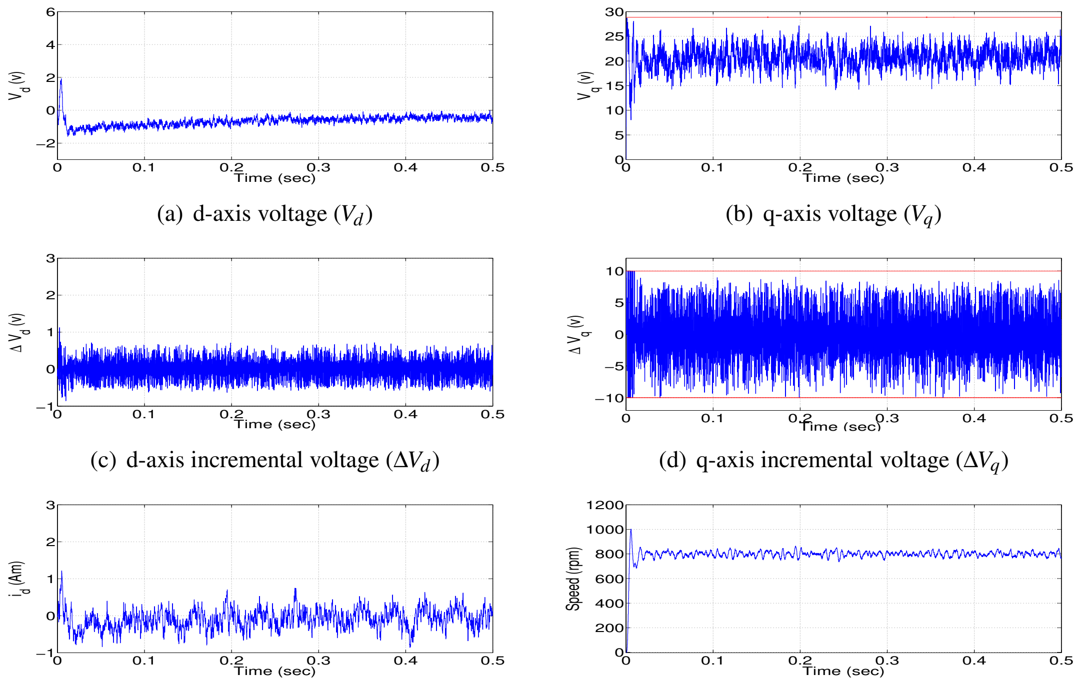
where $0 \le \varepsilon \le 1$. The performance of the controller relies on the tuning of the weighting terms r_w, Q_{id} and Q_ω. Figure 8 shows the case resulting from both V_q and ΔV_q hitting the constraints given in (25) at startup when $Q_\omega = 1$.

$$\begin{aligned} &-28.87 \le V_q \le 28.87 \\ &-10 \le \Delta V_q \le 10 \end{aligned} \tag{25}$$

The MPC and constraints parameters used in the experiments are given in Table 2.

N_c	N_p	Q_{id}	Q_w	r_w	ε	V_{dc}
5	20	1	1	100	0.5	100

Table 2: Design parameters for MPC



(a) d-axis voltage (V_d) (b) q-axis voltage (V_q)

(c) d-axis incremental voltage (ΔV_d) (d) q-axis incremental voltage (ΔV_q)

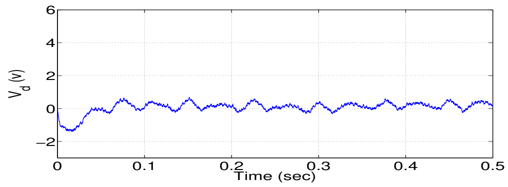
(e) d-axis current (i_d) (f) motor speed

Figure 8: Step responses for $Q_\omega = 1$.

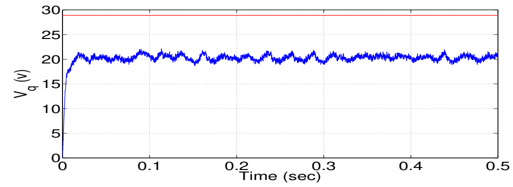
7.3. MPC controller tuning

The performance achieved with the MPC design is dependent on the weighting matrix Q in the cost function (12) where, due to the diagonal form of this matrix, there are two parameters to be selected in the performance specification, Q_{id} the weighting parameter on the d- axis current and Q_ω the weighting parameter on the velocity ω. If theses take relatively large values then this produces high gains in the closed-loop control that, in turn, amplify the noise and high frequency ripples in the drive. Hence the selection of these weights is a trade-off between response time and the attenuation of noise and high frequency ripples.

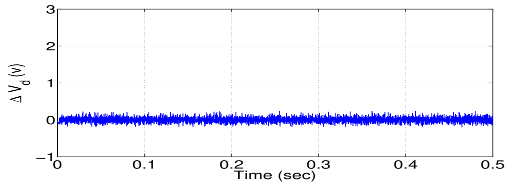
Figure 9 shows the case when $Q_\omega = 0.01$ and $Q_{id} = 1$, which significantly reduces the magnitude of the noise in the steady-state comparing with Figure 8, but at the cost of slow dynamics.



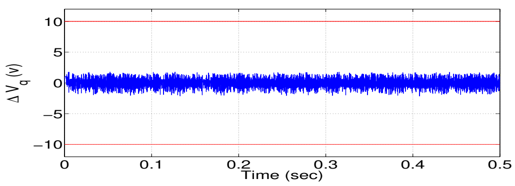
(a) d-axis voltage (V_d)



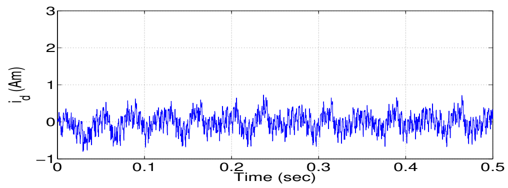
(b) q-axis voltage (V_q)



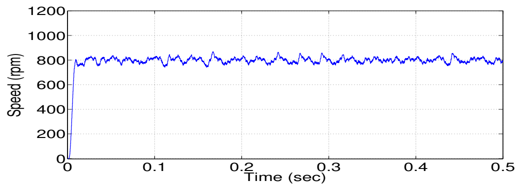
(c) d-axis incremental voltage (ΔV_d)



(d) q-axis incremental voltage (ΔV_q)



(e) d-axis current (i_d)



(f) motor speed

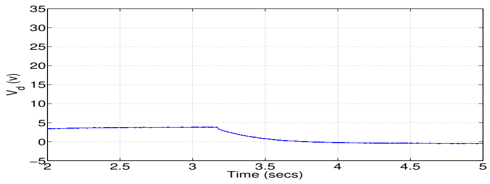
Figure 9: Step responses for $Q_\omega = 0.01$.

7.4. Response to load disturbances

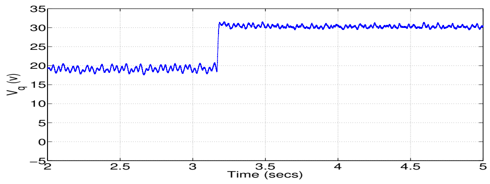
The load disturbance is represented by T_L in (1), which can be modeled as a slowly varying signal to represent, for example, the external friction forces in the grinding process. Under the assumption that the load disturbance is mainly composed of low frequency components, MPC design with the embedded integrator can reject the impact of external load disturbances. In the experiments an external load disturbance was introduced by suddenly switching on the resistor loads when the coupled motors are running at the constant speed using the setup shown in Figure 5. This process yields a equivalent step load disturbance (T_L) on the motor shaft. Figure 10 demonstrates that the controller immediately brings the motor speed back to the steady-state using the design parameters in Table 2, the d-axis current i_d and motor speed ω_m are well controlled at their set-points and the q-axis current i_q and voltage V_q step up to compensate for the load increase. In this case, the constraint on the q-axis voltage is relaxed by choosing $\varepsilon = 0.9$, that is,

$$-51.96 \leq V_q \leq 51.96, \; -25.17 \leq V_d \leq 25.17$$

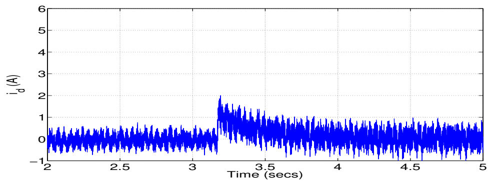
in order to return the speed to the set-point when the load torque is added. Otherwise, the constraints would not allow enough q-axis voltage to bring the speed back to the set-point, resulting in a steady-state error.



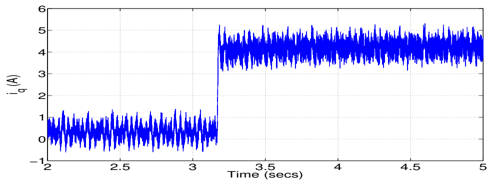
(a) d-axis voltage (V_d)



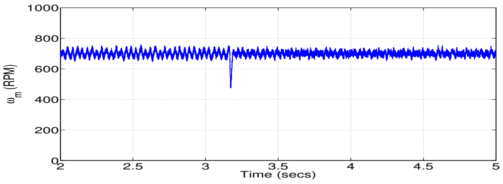
(b) q-axis voltage (V_q)



(c) d-axis current (i_d)



(d) q-axis current (i_q)



(e) motor speed (ω_m)

Figure 10: Rejection of a step load disturbance.

8. Conclusions

This paper has developed constrained MPC for a PMSM with the resulting quadratic programming problem solved on-line using Hildreth's procedure. In comparison to the classical cascade PI control structure, this MPC approach is a model-based single controller. Another major advantage of MPC is the ability to impose constraints, as confirmed by the experimental results given in this paper with constraints imposed on the next input move. In addition, the choice of tuning parameters and their effects on velocity response and noise attenuation have been discussed and supported by experimental results.

Bolognani, S., Peretti, L., Zigliotto, M., 2009. Design and implementation of model predictive control for electrical motor drives. IEEE Transactions on Industrial Electronics 56 (6), 1925 –1936.

Boyd, S., Vandenberghe, L., 2004. Convex Optimization. Cambridge University Press.

Chung, D.-W., Sul, S.-K., 1998. Analysis and compensation of current measurement error in vector-controlled ac motor drives. IEEE Transactions on Industrial Applications 34 (2), 340 –345.

Clarke, D. W., Mohtadi, C., Tuffs, P. S., 1987. Generalized predictive control.part 1: The basic algorithm. part 2: Extensions and interpretations. Automatica Vol.23, 137–160.

Cutler, C. R., Ramaker, B. L., 1979. Dynamic matrix control-a computer control algorithm. Presented at the Meeting of the American Institute of Chemical Engineers, Houston, Texas.

de Santana, E., Bim, E., do Amaral, W., 2008. A predictive algorithm for controlling speed and rotor flux of induction motor. IEEE Transactions on Industrial Electronics 55 (12), 4398 –4407.

Hava, A., Kerkman, R., Lipo, T., 1998. Carrier-based PWM-VSI overmodulation strategies: analysis, comparison, and design. IEEE Transactions on Power Electronics 13 (4), 674 –689.

Hildret, C., March 1957. A quadratic programming procedure. Naval Research Logistics Quarterly 4 4 (1), 79–85.

Ishida, J., Doki, S., Okuma, S., 2010. Fast torque control system of pmsm based on model predictive control considering overmodulation region. In: International Power Electronics Conference (IPEC). pp. 1403 –1406.

Jahns, T., Soong, W., 1996. Pulsating torque minimization techniques for permanent magnet ac motor drives-a review. IEEE Transactions on Industrial Electronics 43 (2), 321 –330.

Luenberger, D. G., 1984. Linear and Nonlinear Programming. Second edition, Addison-Wesley Publishing Company.

Maciejowski, J. M., 2002. Predictive Control with Constraints. Prentice Hall, Essex, England.

Mariethoz, S., Domahidi, A., Morari, M., 2009. A model predictive control scheme with torque ripple mitigation for permanent magnet motors. In: In Proc. IEEE IECON. pp. 2943 –2948.

Peng, Y., Vrancic, D., Hanus, R., 1996. Anti-windup, bumpless, and conditioned transfer techniques for pid controllers. IEEE Control Systems Magazine 16 (4), 48 –57.

Pillay, P., Krishnan, R., 1988. Modeling of permanent magnet motor drives. IEEE Transactions on Industrial Electronics, 35 (4), 537 –541.

Shin, H.-B., 1998. New antiwindup pi controller for variable-speed motor drives. IEEE Transactions on Industrial Electronics 45 (3), 445 –450.

Staebler, M., 2000. TMS320F240 DSP-solution for high- resolution position with sin/cos- encoders. Tech. rep., Texas Instruments.

Vas, P., 1992. Electrical Machines and Drives- A Space-Vector Theory Approach. Oxford University Press.

Vas, P., 1993. Parameter Estimation, Condition Monitoring, and Diagnosis of Electrical Machines. Oxford University Press, New York, USA.

Wang, L., 2009. Model Predictive Control System Design and Implementation Using MATLAB, 1st Edition. Springer London.

Wang, L., Chai, S., Rogers, E., July 2010. Predictive repetitive control based on frequency decomposition. In: American Control Conference (ACC), 2010. pp. 4277 –4282.

Xu, J.-X., Panda, S., Pan, Y.-J., Lee, T. H., Lam, B., 2004. A modular control scheme for pmsm speed control with pulsating torque minimization. IEEE Transactions on Industrial Electronics 51 (3), 526 – 536.