Speed Control of PMSM Drives by Generalized Predictive Algorithms

Květoslav Belda
Department of Adaptive Systems
Institute of Information Theory and Automation of ASCR
Pod Vodárenskou věží 4, 182 08 Prague 8, Czech Republic
E-mail: belda@utia.cas.cz

David Vošmik
Regional Innovation Centre for Electrical Engineering (RICE)
University of West Bohemia (UWB)
Univerzitní 26, 306 14 Pilsen, Czech Republic
E-mail: vosmik1@rice.zcu.cz

***Abstract*–The paper deals with generalized predictive algorithms applied to the industrial drives employing Permanent Magnet Synchronous Motors (PMSM). The Generalized Predictive Control (GPC) belongs to the multistep model-based control design. The presented GPC algorithms are arranged in the form suitable for direct use in a real PMSM drive application. Here, the usual form of GPC action calculating formula is specifically decomposed in a summation of products of tabulated GPC gains with real values of PMSM topical outputs, state, future required reference and possibly with values of previous control actions. The formulated GPC algorithms are experimentally compared with a standard cascade vector PI control. A speed control task is considered for the comparison. The experiments are documented by the time histories in oscillogram screenshots and appropriate figures of applied GPC gains.**

I. Introduction

Development of new electrical drives is constantly raising area due to steady-stay interest and demands from industrial production or end users as well [8]. Advanced type of electrical drives is based on three phase Permanent Magnet Synchronous Motors (PMSM). They are favored for limited mechanical elements leading to long operation life with minor demands on maintenance together with a relative wide range of operation use. However, these positive properties are balanced by a necessity to control simultaneously amplitude and frequency of all three terminal Alternate Currents (AC) with Pulse-Width-Modulation (PWM). The control has key role for use of PMSM drives in industrial applications.

From general control point of view, PMSM represents very high dynamic system with very short response. Usual solution is based on PI controllers coupled in cascade loops [4], [5]. The cascade configuration proves convenient behavior for vide range of operating points [8]. However, for the best behavior, it can require different setting, which provides control actions closer to requirements of considered PMSM drive. The setting depends frequently on empirical rules only.

Another way of solution, considered in this paper, is a solution based on more detailed mathematical physical PMSM analysis involved in the control design. From operation point of view, the model-based approach can optimize the control process for whole operation range within some defined time interval [12]. It can naturally generate control actions relative to topical operational points. One promising model-based approach is Generalized Predictive Control (GPC) [1], [2]. The GPC represents a multistep optimal control with an optimization in a certain time-finite receding horizon.

The aim of this paper is to explain and to demonstrate a full-valued simple arrangement of Generalized Predictive Control and its implementation in the real speed control of Permanent Magnet Synchronous Motor drive (Fig. 1). There is a brief summarization of substantial GPC features, which may be significant or interesting for applications driven by PMSM drives. The GPC features are discussed from practical point of view within the frame of experiments. Thus, in contrast of the mentioned approaches above, the paper follows standard GPC control design and shows way of GPC adaptation for control demands of modern PMSM drives.

The proposed arrangement consists in a reconfiguration/decomposition of the standard GPC action computation formula into a summation of the products of tabulated GPC gains with real values of PMSM topical outputs, state, future required reference and possibly with values of previous control actions. This expression of summation of products gives also analogy with control laws of Linear-Quadratic (LQ) Control well known from control theory [3].

The paper is organized as follows. Section II defines a suitable mathematical physical model for model-based control design. Section III makes an overview of the standard vector PI cascade control. Section IV deals with a derivation of Generalized Predictive algorithms. Section V describes algorithm implementation issues. Finally, Section VI illustrates described theoretical results by oscillogram screenshots and appropriate figures showing considered tabulated GPC. The oscillograms in this section show data from real experiments both for PI cascade control and for GPC.

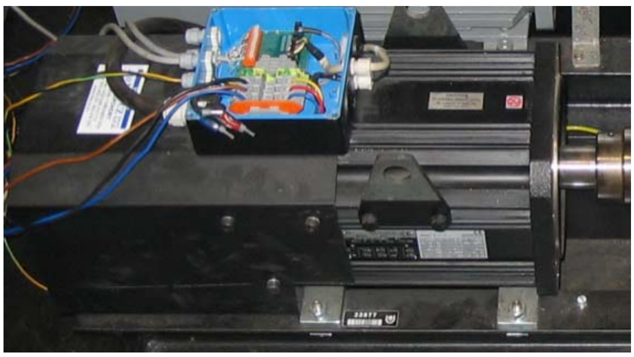

Fig. 1. Testing Permanent Magnet Synchronous Motor drive.

II. Model Definition for Control Design

The usual mathematical model follows from the voltage distribution in individual phases of three AC phase system and from torque equilibrium equation. For control, the model of PMSM (considering Clarke and Parke transformations) is defined by the following set of equations (1) - (3) in d - q rotating field coordinate system (rotating reference frame):

$$u_{Sd} = R_S\, i_{Sd} + L_S \frac{d}{dt} i_{Sd} - L_S\, \omega_e\, i_{Sq} \tag{1}$$

$$u_{Sq} = R_S\, i_{Sq} + L_S \frac{d}{dt} i_{Sq} + L_S\, \omega_e\, i_{Sd} + \psi_M\, \omega_e \tag{2}$$

where R_S, L_S, ψ_M are motor parameters (see Table I), u_{Sd}, u_{Sq} are d - q voltages (system inputs), i_{Sd}, i_{Sq} are d - q currents, ω_e is the electrical rotor speed (mechanical speed $\omega_m = \omega_e / p$; p signifies the number of pole pairs),

$$J\,\ddot{\vartheta}_e = \frac{3}{2} p^2 \psi_M\, i_{Sq} - B\omega_e - p\tau_L \tag{3}$$

where J, B are other motor parameters (see Table I), ϑ_e is the electrical rotor position, τ_L is a disturbance torque.

The model (1) - (3) can be rearranged in a state-space like form:

$$\frac{d}{dt}\begin{bmatrix} i_{Sd} \\ i_{Sq} \\ \omega_e \\ \tau_L \end{bmatrix} = \begin{bmatrix} -\frac{R_S}{L_S} & \omega_e & 0 & 0 \\ -\omega_e & -\frac{R_S}{L_S} & -\frac{\psi_M}{L_S} & 0 \\ 0 & \frac{3}{2}\frac{p^2}{J}\psi_M & -\frac{B}{J} & -\frac{p}{J} \\ 0 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} i_{Sd} \\ i_{Sq} \\ \omega_e \\ \tau_L \end{bmatrix} + \begin{bmatrix} \frac{1}{L_S} & 0 \\ 0 & \frac{1}{L_S} \\ 0 & 0 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} u_{Sd} \\ u_{Sq} \end{bmatrix} \tag{4}$$

$$\frac{d}{dt}\mathbf{x}(t) = \mathbf{A}_C(\omega_e)\, \mathbf{x}(t) + \mathbf{B}_C\, \mathbf{u}(t) \tag{5}$$

where $\mathbf{A}_C(\omega_e)$ is a variable state-space matrix relative to ω_e, $\mathbf{B}_C$ is a constant input matrix. The two nonlinear terms $\omega_e\, i_{Sq}$ and $\omega_e\, i_{Sd}$ in (1) and (2) are decomposed in (4) according to the idea of a specific linearizing decomposition described in [13].

The state-space model (4) or (5) represents as simple as possible description for model-based control.

III. Standard Vector Control

The standard vector control follows directly from the model definition described in the previous section. After measurement of individual phase currents and measurement or estimation of a rotor position and rotor speed, the currents are transformed stepwise by forward Clarke transformation and by forward Park transformation into d - q coordinate system. In it, the main control operation is executed. Generated values of control actions (required voltages) are transformed to the form used in PWM modulator, which generates appropriate individual voltage curves for individual A-B-C phases. The described way is illustrated by the scheme in Fig. 3.

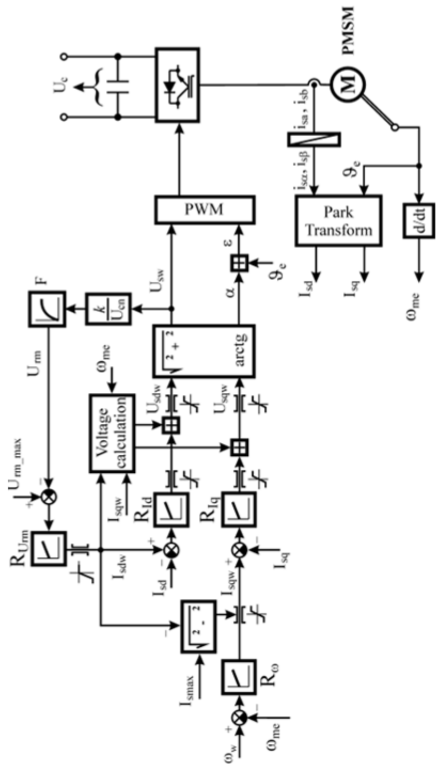

Fig. 3. Speed control of PMSM by vector control (cascade control) [8].

Speed control of PMSM (Fig. 3) consists of the four interconnected loops. Current d and q components are controlled separately. They correspond to torque and flux respectively. Block of voltage calculation is used for faster dynamical response and its outputs are added to current controllers' outputs [5], [8]. The speed loop is the master loop in the speed control process and the speed PI controller provides required values of q - component of current vector.

In some cases, it is important to use field weakening to reach high speed region, due to increasing Electro Magnetic Field (EMF) voltage and finite supply voltage. Field weakening is done by current d - component, which produces magnetic flux opposite to flux of permanent magnets. Required d - component of current vector is provided by PI controller of the modulation index. That index is calculated directly from the required voltage vector magnitude. It is important to mention that the output of the current controller (current component in q axis) must be limited according to rising current component in d axis to respect maximal allowed value of current magnitude.

IV. Predictive Control Algorithms

The model-based control uses in general the knowledge of the dynamic model and that way it globally optimizes control actions in the view of the controlled system behavior and required outputs. Generalized Predictive Control (GPC) is a one of model-based control. It represents a multistep control strategy based on equations of predictions and the local repetitive minimization of quadratic cost function [1]. For the considered speed control task, the GPC replaces the conventional cascade form by one numerical calculation.

The GPC is usually implemented as discrete (digital) control. Therefore, the model (5), due to its linearized form, can be discretized by standard exponential discretization procedure to the form:

$$\mathbf{x}_{k+1} = \mathbf{A}_k\ \mathbf{x}_k + \mathbf{B}\ \mathbf{u}_k\,, \qquad \mathbf{y}_k = \mathbf{C}\ \mathbf{x}_k \tag{6}$$

The equations of predictions serve for the expression of feed-forward within a horizon of predictions *N*. On their basis, the dominant part of the control actions is determined.

Using discrete state-space form (6), the equations are given in the basic form:

$$\begin{aligned}
\hat{\mathbf{x}}_{k+1} &= \mathbf{A}\ \mathbf{x}_k + \mathbf{B}\mathbf{u}_k \\
\hat{\mathbf{y}}_{k+1} &= \mathbf{C}\ \mathbf{A}\ \mathbf{x}_k + \mathbf{C}\ \mathbf{B}\mathbf{u}_k \\
&\vdots \\
\hat{\mathbf{x}}_{k+N} &= \mathbf{A}^N\ \mathbf{x}_k + \mathbf{A}^{N-1}\ \mathbf{B}\mathbf{u}_k + \cdots + \mathbf{B}\mathbf{u}_{k+N-1} \\
\hat{\mathbf{y}}_{k+N} &= \mathbf{C}\mathbf{A}^N\ \mathbf{x}_k + \mathbf{C}\mathbf{A}^{N-1}\ \mathbf{B}\mathbf{u}_k + \cdots + \mathbf{C}\mathbf{B}\mathbf{u}_{k+N-1}
\end{aligned} \tag{7}$$

which can be expressed in the matrix notation:

$$\hat{\mathbf{y}} = \mathbf{f} + \mathbf{G}\mathbf{u}\,, \quad \mathbf{f} = \begin{bmatrix} \mathbf{CA} \\ \vdots \\ \mathbf{CA}^N \end{bmatrix} \mathbf{x}_k\,, \quad \mathbf{G} = \begin{bmatrix} \mathbf{C} & \mathbf{B}\cdots & \mathbf{0} \\ \vdots & \ddots & \vdots \\ \mathbf{C}\mathbf{A}^{N-1} & \mathbf{B}\cdots & \mathbf{CB} \end{bmatrix} \tag{8}$$

The other crucial part of GPC is a quadratic criterion or minimization of quadratic cost function on horizon *N*:

$$J_k = \sum_{j=k+1}^{k+N} \{ \| \boldsymbol{Q}_{\mathrm{y}} (\hat{\mathbf{y}}_j - \mathbf{w}_j) \|^2 + \| \boldsymbol{Q}_{\mathbf{u}} \mathbf{u}_{j-1} \|^2 \} \tag{9}$$

The cost function (9) can be minimized by several ways. The most related optimization way at GPC is a quadratic programming, i.e. optimization of the objective function by algorithms of the quadratic programming [14]:

$$\begin{aligned}
\min_{\mathbf{u}} \mathbf{F}(\mathbf{u}) &= \min_{\mathbf{u}} \{ \tfrac{1}{2} \mathbf{u}^T \underbrace{(\mathbf{G}^T \mathbf{Q}_{\mathrm{y}} \mathbf{G} + \mathbf{Q}_{\mathbf{u}})}_{\mathbf{H}} \mathbf{u} + \underbrace{(\mathbf{f} - \mathbf{w})^T \mathbf{G}}_{\mathbf{g}^T} \mathbf{u} \} \\
&= \min_{\mathbf{u}} \{ \tfrac{1}{2} \mathbf{u}^T \mathbf{H}\ \mathbf{u} + \mathbf{g}^T\ \mathbf{u} \}, \quad \mathbf{A}\mathbf{u} \le \mathbf{b}
\end{aligned} \tag{10}$$

The quadratic programming can solve simultaneously equality and inequality constrains as it is indicated in (10). However, for PMSM drives, it is a quite time-consuming way apart from pre-computed offline implementations [6], [7].

Very powerful way is a square-root optimization approach. The quadratic cost function is minimized via its square-root only as indicated thereinafter. The minimization of (9) can be provided in one shot as a least squares problem solution of algebraic system of equations [15]:

$$J_k = \mathbf{J}^T \times \mathbf{J} \tag{11}$$

$$\min_{\mathbf{u}} \mathbf{J} = \begin{bmatrix} \mathbf{Q}\mathrm{y} & \mathbf{0} \\ \mathbf{0} & \mathbf{Q}\mathrm{u} \end{bmatrix} \begin{bmatrix} \hat{\mathbf{y}} - \mathbf{w} \\ \mathbf{u} \end{bmatrix} = \begin{bmatrix} \mathbf{Q}\mathrm{y}\ \mathbf{G} \\ \mathbf{Q}\mathrm{u} \end{bmatrix} \mathbf{u} - \begin{bmatrix} \mathbf{Q}\mathrm{y}\ (\mathbf{w} - \mathbf{f}) \\ \mathbf{0} \end{bmatrix} \tag{12}$$

This way is very computationally effective and mathematically stable solution and is applicable in real-time applications [16], in special situations also under constraints [17]. Nevertheless, it still requires some powerful digital signal processor, which cannot be considered for usual broad use.

The last, the standard way is based on the searching for a local minimum of the cost function [2]. It solves the optimization tasks without constrains, but it can be especially formulated in a form feasible on usual available signal processors. This way will be considered as a basis for proposed specific GPC algorithms designed as low-end solution.

In general, the cost function (9) itself can have different forms according to requirements on controlled system behaviour without any respect of selection of optimization way. Here, two forms, considered for experiments, are expressed.

One *cost function form* is a *positional (absolute) form* of GPC algorithm, the same with (9):

$$J_k = \sum_{j=k+1}^{k+N} \{ \| \boldsymbol{Q}_{\mathrm{yw}} (\hat{\mathbf{y}}_j - \mathbf{w}_j) \|^2 + \| \boldsymbol{Q}_{\mathbf{u}} \mathbf{u}_{j-1} \|^2 \} \tag{13}$$

Its minimization leads to the following expression for **u**:

$$\mathbf{u} = (\mathbf{G}^T \mathbf{Q}_{\mathrm{yw}} \mathbf{G} + \mathbf{Q}_{\mathbf{u}})^{-1} \mathbf{G}^T \mathbf{Q}_{\mathrm{yw}} (\mathbf{w} - \mathbf{f}) \tag{14}$$

The selection of the first control actions used for real control can be provided by a matrix **M**:

$$\mathbf{u}_k = \mathbf{M}\ (\mathbf{G}^T \mathbf{Q}_{\mathrm{yw}} \mathbf{G} + \mathbf{Q}_{\mathbf{u}})^{-1} \mathbf{G}^T \mathbf{Q}_{\mathrm{yw}} (\mathbf{w} - \mathbf{f}) \tag{15}$$

where **M** is the $(nu,\ nu\,N)$ unit diagonal matrix (nu is a number of inputs, i.e. for PMSM $nu = 2$, $\mathbf{u}_k = [u_{Sd}, u_{Sq}]^T$); matrix **G** and vector **f** with an appropriate type and dimension are defined by (8). After several modifications of (15), the suitable formula for control actions (control law) is:

$$\mathbf{u}_k = \mathbf{k}_{\mathrm{w}} \mathbf{w}_k - \mathbf{k}_{\mathrm{x}} \mathbf{x}_k \tag{16}$$

This control law or gains $\mathbf{k_w}$ (nu, ny), $\mathbf{k_x}$ (nu, nx) can be tabulated relative to speed ω_e, see state-space model (4).

The other function form is an *incremental form* of GPC:

$$J_k = \sum_{j=k+1}^{k+N} \{ \| \boldsymbol{Q}_{\mathrm{yw}} (\hat{\mathbf{y}}_j - \mathbf{w}_j) \|^2 + \| \boldsymbol{Q}_{\Delta\mathrm{y}} \Delta\hat{\mathbf{y}}_j \|^2 + \| \boldsymbol{Q}_{\Delta\mathbf{u}} \Delta\mathbf{u}_{j-1} \|^2 \} \tag{17}$$

The minimization of (17) leads to the similar consequences as for previous positional form:

$$\begin{aligned}
\Delta\mathbf{u} &= (\widetilde{\mathbf{G}}^T (\mathbf{Q}_{\mathrm{yw}} + \mathbf{Q}_{\Delta\mathrm{y}})\, \widetilde{\mathbf{G}} + \mathbf{Q}_{\mathbf{u}})^{-1}\, \widetilde{\mathbf{G}}^T \\
&\quad \times (\, \mathbf{Q}_{\mathrm{yw}} (\mathbf{w} - \mathbf{I}\mathbf{y}_k) - (\mathbf{Q}_{\mathrm{yw}} + \mathbf{Q}_{\Delta\mathrm{y}})\, \widetilde{\mathbf{f}}\,)
\end{aligned} \tag{18}$$

$$\begin{aligned}
\Delta\mathbf{u}_k &= \mathbf{M}\ (\widetilde{\mathbf{G}}^T (\mathbf{Q}_{\mathrm{yw}} + \mathbf{Q}_{\Delta\mathrm{y}})\, \widetilde{\mathbf{G}} + \mathbf{Q}_{\mathbf{u}})^{-1}\, \widetilde{\mathbf{G}}^T \\
&\quad \times (\, \mathbf{Q}_{\mathrm{yw}} (\mathbf{w} - \mathbf{I}\mathbf{y}_k) - (\mathbf{Q}_{\mathrm{yw}} + \mathbf{Q}_{\Delta\mathrm{y}})\, \widetilde{\mathbf{f}}\,)
\end{aligned} \tag{19}$$

where **M** is $(nu,\ nu\,N)$ unit diagonal matrix again. However, the structures of matrix $\widetilde{\mathbf{G}}$ and vector $\widetilde{\mathbf{f}}$ differ from **G** and **f** of the first algorithm due to incremental properties.

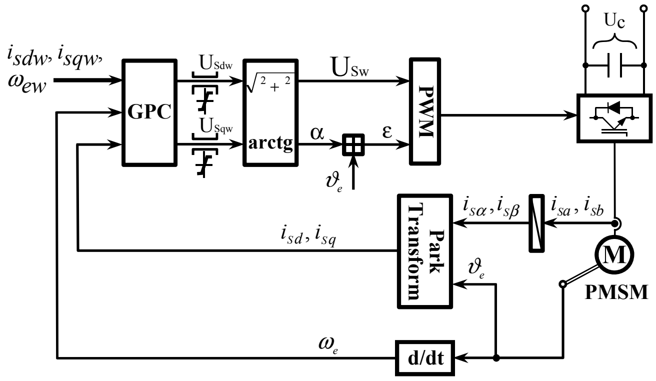

Fig. 4. Speed control of PMSM by Generalized Predictive Control.

The $\widetilde{\mathbf{G}}$ and $\widetilde{\mathbf{f}}$ are given as:

$$\hat{\mathbf{y}} = \widetilde{\mathbf{f}} + \widetilde{\mathbf{G}}\,\Delta\mathbf{u}\,,\quad \widetilde{\mathbf{f}} = \begin{bmatrix} \mathbf{I} \\ \vdots \\ \mathbf{I} \end{bmatrix} \mathbf{y}_k + \begin{bmatrix} \mathbf{CA} \\ \vdots \\ \mathbf{CA} + \mathbf{CA}^2 + \cdots + \mathbf{CA}^N \end{bmatrix} \Delta\mathbf{x}_k$$

$$\widetilde{\mathbf{G}} = \begin{bmatrix} \mathbf{CB} & \cdots & \mathbf{0} \\ \vdots & \ddots & \\ \mathbf{CB} + \mathbf{CAB} + \cdots + \mathbf{CA}^{N-1}\mathbf{B} & \cdots & \mathbf{CB} \end{bmatrix} \tag{20}$$

where $\mathbf{I}$ is a partial identity matrix with the same order as dimension or number ny of output vector $\mathbf{y}_k$; $\widetilde{\mathbf{G}}$ and $\widetilde{\mathbf{f}}$ have similar origin as $\mathbf{G}$ and $\mathbf{f}$ except for cumulative element character.

Final form of control law for incremental form of GPC is given as follows:

$$\Delta\mathbf{u}_k = \mathbf{k}_e(\mathbf{w}_k - \mathbf{y}_k) - \mathbf{k}_{\Delta x}(\mathbf{x}_k - \mathbf{x}_{k-1}) \tag{21}$$

where the gains $\mathbf{k_e}$ (nu, ny) and $\mathbf{k_{\Delta x}}$ (nu, nx) are matrices of indicated types. This control law has to be further supplemented with a cumulative formula defining real control actions.

$$\mathbf{u}_k = \mathbf{u}_{k-1} + \Delta\mathbf{u}_k \tag{22}$$

The Fig. 4 in analogy to Fig. 3 shows scheme for speed control of PMSM by GPC algorithms. The proposed solution aggregates standard cascade control into one GPC block.

V. Hardware Implementation of Algorithms

The algorithms were implemented in UWB control system with DSP TMS320f28335. This DSP works with floating point arithmetic and single precision format. The control system is connected to the laboratory PMSM drive of rated power 10.7 kW. The drive parameters are listed in the Table I. The testing stand with PMSM drive is shown in Fig. 7.

The standard PI controllers were tuned experimentally on the PMSM drive. Appropriate parameters of PI controllers are listed in Table II.

The GPC control law (21) (GPC incremental algorithm) with individual penalizations of outputs, output increments and input increments (see the cost function (17)) were programmed to the same DSP as the PI controllers in the cascade configuration. The individual GPC gains were set either as constants or were interpolated for boundary and zero values of the speed ω_e. The calculation of the interpolation was realized as linear or parabolic according to the appropriate gain element profile (Fig. 5 and Fig. 6).

The gains were set up by a simulation with the described GPC algorithms using mathematical model given by (1) - (3). The GPC algorithms were simulated against UWB simulator. The UWB simulator was used as an accurate mathematical model of reality. The simulator is based on the model (1) - (3), but is complemented relative to real components of testing drive equipment (dead time, delays, voltage drops etc.) The gains correspond to GPC control parameters, which are listed in Table III. The gain dependences on the speed ω_e are shown in Fig. 5 and Fig. 6.

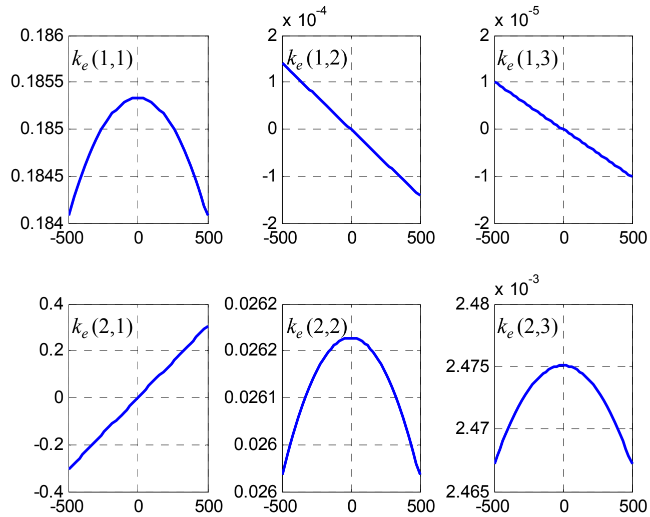

Fig. 5. GPC gain $\mathbf{k}_e = f(\omega_e)$, $\omega_e \in \langle -500, 500 \rangle\,[rad_e\, s^{-1}]$.

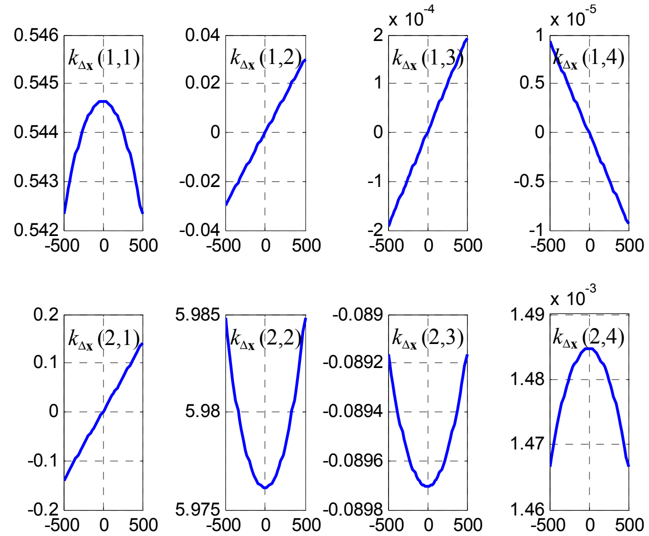

Fig. 6. GPC gain $\mathbf{k}_{\Delta x} = f(\omega_e)$, $\omega_e \in \langle -500, 500 \rangle\,[rad_e\, s^{-1}]$.

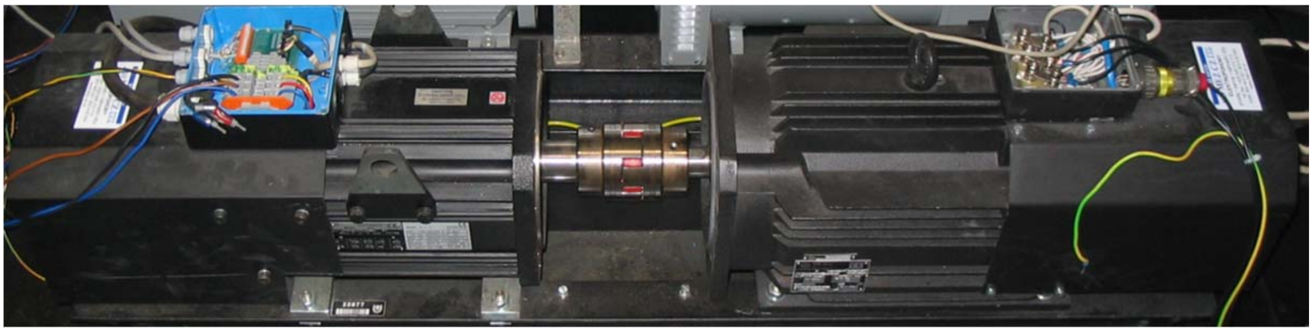

Fig. 7. Testing stand with coupled PMSM drives.

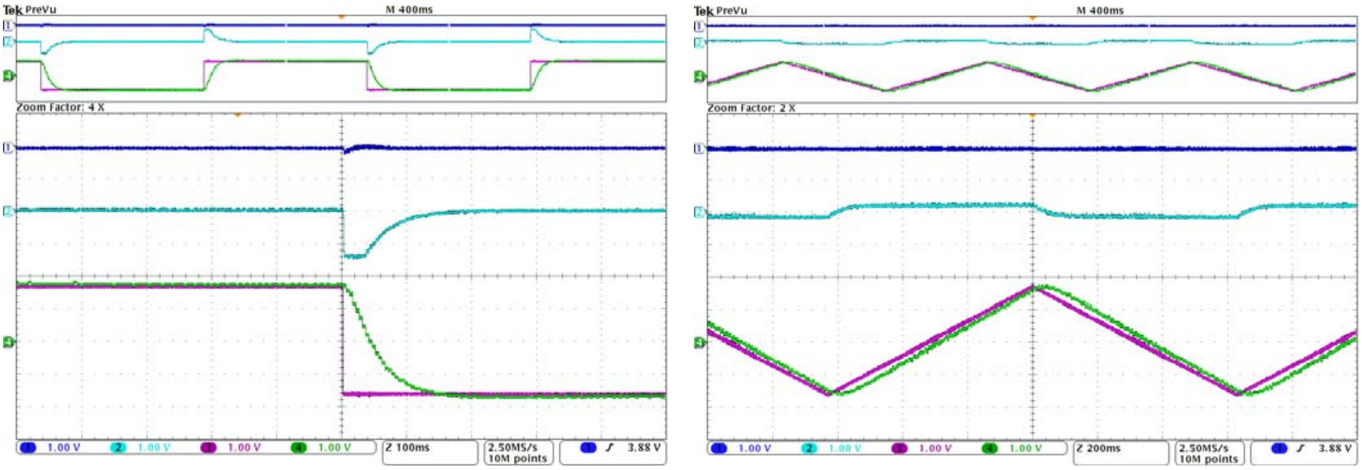

Fig. 8. PI vectror control: time histories for rectangular (left) and triangular (right) desired profiles; commanded fme=+-25Hz, ch1: i_{Sd} current (25A/div), ch2: i_{Sq} current (25A/div), ch3: commanded el. rotor speed (15Hz/div), ch4: measured el. rotor speed (15Hz/div).

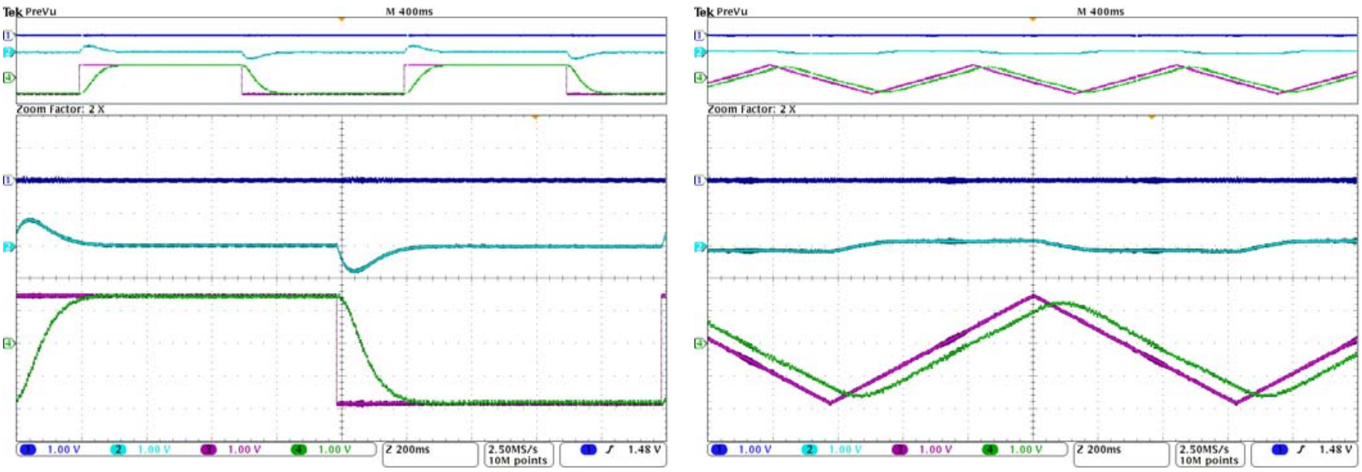

Fig. 9. Incr. GPC algorithm: time histories for rectangular (left) and triangular (right) desired profiles; commanded fme=+-25Hz, ch1: i_{Sd} current (25A/div), ch2: i_{Sq} current (25A/div), ch3: commanded el. rotor speed (15Hz/div), ch4: measured el. rotor speed (15Hz/div).

VI. Real Experiments

Real experiments were realized for two type standard testing signals: rectangular and triangular profiles. The appropriate time histories are shown in Fig. 8 for standard PI vector control and in Fig. 9 for incremental GPC algorithm.

A transient response and a response in steady-stays are demonstrated by rectangular desired profiles of ω_e on the left of figures. A dynamical response is demonstrated by traingular desired profiles of ω_e. on the right of figures.

The behavior of incremental GPC algorithm is comparable with PI vector control. However, the individual profiles are a little bit different after all. It is caused by different way of control action construction.

From GPC algorithms (positional (absolute) and incremental) derived, explained and indicated in Section IV., the incremental algorithm is shown in figures only. However, the both algorithms and furthermore their several other variants were tested. From control point of view, the both algorithms positional and incremental prove similar features in dynamical processes. The positional algorithm in these processes is more manageable due to direct penalization of the whole control actions in contrast of incremental algorithm, where increments of control actions are penalized only.

In steady-state cases, the positional algorithm cannot remove steady-stay error from its positional principle. In these cases, the incremental algorithm is more capable. It accumulates increments of the control actions permanently as long as the steady-state error is zero (tends zero). This process is asymptotical due to discrete realization. When a slight deviation from the required outputs (nonzero control error) appears, the incremental algorithm starts to level out that deviation to zero as a standard I component in PI controller. In this relation, the oscillation or chattering of both control actions and system outputs has to be considered and solved. The proposed GPC incremental algorithm copes with this problem via two measures. The first is an adequate setting of penalization of increments of control actions and the second consists in additional penalization of increments of the system outputs. The both measures provide smoother profiles both control actions and system outputs.

However, the tuning techniques of GPC control parameters (horizons, weighting/penalization matrices) is a challenge not only in the drive area, but for varied types of industrial and home applications. The mentioned challenge is a subject of continual permanent investigation [18].

VII. CONCLUSION

The paper deals with the speed control of PMSM drives. Here, there is a real experimental comparison of standard PI vector control with developed incremental GPC algorithm including penalization of output signal increments. The PI vector control or its appropriate loops contains usual hard limits. The incremental GPC algorithm has only limits on its outputs, i.e. on generated input voltages – control actions. The limiting behavior of GPC algorithm is reached via specific tuning of penalization matrices only. The presented research focused on low-end solutions feasible in broad drive applications.

TABLE I
PARAMETERS OF THE USED PMSM DRIVE

Symbol	Description	Value
P	rated power	10.7 kW
R_s	stator resistance	0.28 Ω (Ohm)
L_s	stator inductance	0.003465 H (Henry)
ψ_M	PM rotor magnetic flux	0.1989 Wb (Weber)
B	viscous coef. of load	0 s^{-2}
p	number of pole pairs	4
J	moment of inertia	0.04 kg m^2
τ_L	load torque	0 N m

TABLE II
PARAMETERS OF THE PI VECTOR CONTROL

Symbol	Description	Value
k_ω	speed gain	0.075
T_ω	speed time constant	0.1
k_i	current gain	5
T_i	current time constant	0.01
k_{Urm}	modulation index gain	75
T_{Urm}	mod. index time const.	0.1
Ts	sampling period	0.000125 s

TABLE III
PARAMETERS OF THE INCREMENTAL GPC

Symbol	Description	Value
N	horizon of prediction	4
$\boldsymbol{Q}_{\mathbf{yw}}$	output penalization	diag(5.5, 0.5, 1) 10^3
$\boldsymbol{Q}_{\mathbf{\Delta y}}$	out. incr. penalization	diag(1, 17.5, 1) 10^3
$\boldsymbol{Q}_{\mathbf{\Delta u}}$	input incr. penalization	diag(7.5, 1.5) 10^3
Ts	sampling period	0.000125 s

ACKNOWLEDGMENT

The authors appreciate kind support of the Grant Agency of the Czech Republic by the grant No. GP102/11/0437 'Control and Parameter Identification of AC Electric Drives under Critical Operating Conditions'.

REFERENCES

[1] A. Ordys and D. Clarke, "A state - space description for GPC controllers", *Int. J. Systems SCI.* 1993, 24, (9), pp. 1727-1744.
[2] L. Wang, *Model Predictive Control System Design and Implementation Using MATLAB®*, Springer, 2009.
[3] Ch. L. Phillips and H. T. Nagle, *Digital Control System Analysis and Design*, Prentice-Hall, Inc., 1995.
[4] P. Vas, *Sensorless Vector and Direct Torque Control*, Oxford Univesrity Press, New York, USA, 1998.
[5] W. Leonhard, *Control of Electrical Drives*, Springer, Berlin 2001.
[6] A. Bemporad, M. Morari, V. Dua and E. N. Pistikopoulos, "The explicit linear quadratic regulator for constrained systems", *Automatica 38*, 2002, pp. 3-20.
[7] M. Kvasnica, J. Löfberg, M. Herceg, L. Cirka, and M. Fikar, "Low complexity polynomial approximation of explicit MPC via linear programming," *American Control Conference* 2010, pp. 4713-4718.
[8] Z. Peroutka, K. Zeman, F. Krus, F. and Kosta, "New generation of full low-floor trams: Control of wheel drives with permanent magnet synchronous motors", *Energy Conversion Congress and Exposition*, IEEE, 2009, pp.1833-1840.
[9] J. Maciejowski, *Predictive Control with Constrains*, Prentice Hall, 2002.
[10] E. Souza, E. Bim and W. Caradori, "A Predictive Algorithm for Controlling Speed and Rotor Flux of Induction Motor", *IEEE Transactions on industrial electronics*, Vol. 55, No. 12, 2008, pp. 4398-4407.
[11] S. Bolognani, L. Peretti and M. Zigliotto, "Design and Implementation of Model Predictive Control for Electrical Motor Drives", *IEEE Trans. on industrial electronics*, 2009, Vol. 56, No. 6, pp. 1925-1936.
[12] M. Preindl, and S. Bolognani, "Model predictive direct speed control with finite control set of PMSM-VSI drive systems", *PRECEDE 2011*, pp. 17-23.
[13] M. Valášek and P. Steinbauer, "Nonlinear Control of Multibody Systems", *Euromech*, Lisabon, 1999, pp. 437-444.
[14] P.E. Gill, and W. Murray, "Numerically stable methods for quadratic programming", *Mathematical Programming 14*, North-Holland Publishing Comp. 1978, pp. 349-372.
[15] Ch. Lawson, R. Hanson, "Solving Least Squares Problems", Siam, Prentice-Hall, Inc., New York, 1995.
[16] K. Belda, J. Böhm, M. Valášek, "State-Space GPC for Redundant Parallel Robots", *Mechanics Based Design of Structures and Machines*, Vol.31, No.3, Marcel Dekker, Inc., New York, 2003, pp. 413-432.
[17] J. Böhm, K.. Belda and M. Valášek, "The Antibacklash Task in the Path Control of Redundant Parallel Robots", *Proc. of the 13th Int. Conf. on Process Control*, Slovak University of Technology, 2001, 6 pp.
[18] K. Belda, "On-line Parameter Tuning of Model Predictive Control", *Preprints of the 18th IFAC World Congress*, Italy 2011, pp. 5489-5494.