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 wide 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):

$$u_{sd} = R_s i_{sd} + L_s \frac{d}{dt} i_{sd} - L_s \omega_s i_{sq}$$

(1)

$$u_{sq} = R_s i_{sq} + L_s \frac{d}{dt} i_{sq} + L_s \omega_s i_{sd} + \psi_m \omega_s$$

(2)

where $R_s$, $L_s$, $\psi_m$ are motor parameters (see Table I), $u_{sd}$, $u_{sq}$ are $d$ - $q$ voltages (system inputs), $i_{sd}$, $i_{sq}$ are $d$ - $q$ currents, $\omega_s$ is the electrical rotor speed (mechanical speed $\omega_m = \omega_s / p$; $p$ signifies the number of pole pairs),

$$J \frac{d \omega_s}{dt} = \frac{3}{2} p' \psi_m i_{sd} - B \omega_s - p \tau$$

(3)

where $J$, $B$ are other motor parameters (see Table I), $\theta$ is the electrical rotor position, $\tau$ is a disturbance torque.

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

$$\frac{d}{dt} \begin{bmatrix} i_{sd} \\ i_{sq} \\ \tau_s \end{bmatrix} = \begin{bmatrix} -\frac{R_s}{L_s} & 0 & 0 \\ \frac{1}{L_s} & -\omega_s & -\frac{\psi_m}{L_s} \\ 0 & \frac{1}{L_s} & -\omega_s \end{bmatrix} \begin{bmatrix} i_{sd} \\ i_{sq} \\ \tau_s \end{bmatrix} + \begin{bmatrix} \frac{1}{L_s} \\ 0 \\ 0 \end{bmatrix} \begin{bmatrix} u_{sd} \\ u_{sq} \end{bmatrix}$$

(4)

$$\frac{dx(t)}{dt} = A_x(\omega_s) x(t) + B_x u(t)$$

(5)

where $A_x(\omega_s)$ is a variable state-space matrix relative to $\omega_s$, $B_x$ is a constant input matrix. The two nonlinear terms $\omega_s i_{sd}$ and $\omega_s i_{sq}$ 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.

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 Electromagnetic 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:
\[
x_{k+1} = A x_k + B u_k, \quad y_k = C x_k
\]  
(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:
\[
\begin{align*}
\dot{x}_{k+1} &= A x_k + B u_k \\
\dot{y}_{k+1} &= C x_k + C A x_k + C A^{N-1} B u_k + \cdots + C B u_{k+N-1} \\
\end{align*}
\]  
(7)

which can be expressed in the matrix notation:
\[
\hat{y} = f + G u, \quad f = \begin{bmatrix} CA \\ \vdots \\ CA^N \end{bmatrix} x_k, \quad G = \begin{bmatrix} C & B & \cdots & 0 \\ \vdots \\ 0 & \cdots & \cdots & \cdots \end{bmatrix}
\]  
(8)

The other crucial part of GPC is a quadratic criterion or minimization of quadratic cost function on horizon \(N\):
\[
J_S = \sum_{j=1}^{N_k} \{ ||Q_y (\hat{y}_j - w_j)||^2 + ||Q_u u_{j-1}\|^2 \}
\]  
(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]:
\[
\min_F (u) = \min_u \left\{ \frac{1}{2} u^T (G^T Q_y G + Q_u) u + (f - w)^T G u \right\} + H u^T g
\]  
(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 thereafter. The minimization of (9) can be provided in one shot as a least squares problem solution of algebraic system of equations [15]:
\[
\min_u J = J^* J
\]  
(11)

This way is very computationally effective and mathematically stable solution and is applicable in real-time applications [16], in special situations also under constrains [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):
\[
J_S = \sum_{j=1}^{N_k} \{ ||Q_y (\hat{y}_j - w_j)||^2 + ||Q_u u_{j-1}\|^2 \}
\]  
(13)

Its minimization leads to the following expression for \(u\):
\[
u_k = M (G^T Q_y G + Q_u)^{-1} G^T Q_y (w - f)
\]  
(14)

The selection of the first control actions used for real control can be provided by a matrix \(M\):
\[
u_k = M (G^T Q_y G + Q_u)^{-1} G^T Q_y (w - f)
\]  
(15)

where \(M\) is the \((nu, nu N)\) unit diagonal matrix \((nu\) is a number of inputs, i.e. for PMSM \(nu = 2\), \(u_k = [u_{k1}, u_{k2}]^T\); 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:
\[
u_k = k_x w_k - k_v x_k
\]  
(16)

This control law or gains \(k_x (nu, ny), k_v (nu, nx)\) can be tabulated relative to speed \(\omega\), see state-space model (4).

The other function form is an incremental form of GPC:
\[
J_S = \sum_{j=1}^{N_k} \{ ||Q_y (\hat{y}_j - w_j)||^2 + ||Q_u \Delta \hat{y}_j||^2 + ||Q_u \Delta u_{j-1}\|^2 \}
\]  
(17)

The minimization of (17) leads to the similar consequences as for previous positional form:
\[
\Delta u = (G^T (Q_y + Q_u) G + Q_u)^{-1} G^T
\times ((Q_y w - Q_u \Delta \hat{y}_j) - (Q_u + Q_u) \Delta f)
\]  
(18)

\[
\Delta u_k = M (G^T (Q_y + Q_u) G + Q_u)^{-1} G^T
\times ((Q_y w - Q_u \Delta \hat{y}_j) - (Q_u + Q_u) \Delta f)
\]  
(19)

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

\[
\mathbf{\hat{y}} = \mathbf{\hat{y}} + \mathbf{G} \Delta \mathbf{u}, \quad \mathbf{\hat{y}} = \begin{bmatrix} I & \mathbf{y} + \begin{bmatrix} \mathbf{CA} \\ \vdots \\ \mathbf{CA} + \mathbf{CA}^2 + \cdots + \mathbf{CA}^n \end{bmatrix} \Delta \mathbf{x} \end{bmatrix}
\]

\[
\mathbf{\hat{G}} = \begin{bmatrix} \mathbf{CB} & \cdots & 0 \\ \vdots & \ddots & \vdots \\ \mathbf{CB} + \mathbf{CAB} + \cdots + \mathbf{CA}^{n-1} \mathbf{B} & \cdots & \mathbf{CB} \end{bmatrix}
\]  

(20)

where \( I \) is a partial identity matrix with the same order as dimension or number \( ny \) of output vector \( \mathbf{y} \); \( \mathbf{G} \) and \( \mathbf{f} \) 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:

\[
\Delta \mathbf{u}_i = \mathbf{k}_i (\mathbf{w}_i - \mathbf{y}_i) - \mathbf{k}_i (\mathbf{x}_i - \mathbf{x}_{i-1})
\]

(21)

where the gains \( \mathbf{k}_i (m, ny) \) and \( \mathbf{k}_i (m, nx) \) are matrices of indicated types. This control law has to be further supplemented with a cumulative formula defining real control actions.

\[
\mathbf{u}_i = \mathbf{u}_{i-1} + \Delta \mathbf{u}_i
\]

(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.
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 $\omega_e$ on the left of figures. A dynamical response is demonstrated by triangular desired profiles of $\omega_e$ 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-state 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>
<thead>
<tr>
<th>Symbol</th>
<th>Description</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>$P$</td>
<td>rated power</td>
<td>10.7 kW</td>
</tr>
<tr>
<td>$R_s$</td>
<td>stator resistance</td>
<td>0.28 Ω (Ohm)</td>
</tr>
<tr>
<td>$L_s$</td>
<td>stator inductance</td>
<td>0.003465 H (Henry)</td>
</tr>
<tr>
<td>$\psi_m$</td>
<td>PM rotor magnetic flux</td>
<td>0.1989 Wb (Weber)</td>
</tr>
<tr>
<td>$B$</td>
<td>viscous coef. of load</td>
<td>0 s²</td>
</tr>
<tr>
<td>$p$</td>
<td>number of pole pairs</td>
<td>4</td>
</tr>
<tr>
<td>$J$</td>
<td>moment of inertia</td>
<td>0.04 kg m²</td>
</tr>
<tr>
<td>$\tau_l$</td>
<td>load torque</td>
<td>0 N m</td>
</tr>
</tbody>
</table>

ACKNOWLEDGMENT

The authors appreciate kind support of the Grant Agency of the Czech Republic by the grant No. GI102/11/0437 ‘Control and Parameter Identification of AC Electric Drives under Critical Operating Conditions’.

REFERENCES