Using Power PC-MPC for Real-Time VHFIM Sensorless Drive Control

Branislav DOBRUCKY

Peter DRGONA

Pavol SPANIK

and

Viktor BOBEK

Department of mechatronics and electronics, University of Zilina
Zilina, 01026, Slovakia

ABSTRACT

The paper deals with real-time control of PM synchronous motor for sensorless position using Power PC-MPC processor. Real parametric-and real-time model could be ideal tools for both control and simulation of the PMSM (permanent magnet synchronous motor) position drive. Unfortunately that idealized model doesn’t exist so far. Such a model has to face following serious problems: it must be excited and synchronized by real input including non-linearities of power supply converter (switch voltage drops, dead-times,...); it should also provide on-line identification of the motor parameters including load changes. Combine real-time control system proposed in the paper fulfills most of those requirements: it is supplied by real data from the motor stator voltages, so it is exactly synchronized with real system, and it works with off-line identified motor parameters.

Keywords: VHFIM, sensorless control, PowerPC, model, motor

1. INTRODUCTION

Unlike from original 'true' high frequency (HF) injection method [6], the virtual HF signal is not added to basic supply voltage of inverter, but it is subtracted from it, so acting stator voltage of model [5] as follows:

$$V_{act} = V_x - V_{inj}$$  \hspace{1cm} (1)

The injection- and acting model voltages are virtual, obviously synchronized by real stator voltage. The acting voltage obtained from the model of PMSM causes the virtual acting stator current as response on it. From this response it is possible to extract the information about rotor position even under the rotating motor, by the same way as classical injection method (using filters).

DSP with its computing capabilities are dedicated for high level control, but its peripherals are not designed for motor control. Powerful MPC offers solutions with very good computing performance (floating point), also with high speed peripherals – especially eTPU. This type of devices can be classified as intersection between FPGA and DSP. Field Oriented Control structure has been used to control PMSM drive. Typical cascade FOC control structure is shown in Fig. 2.
2. VHFIM CONTROL SYSTEM FOR PMSM DRIVE

It is possible to express HF component of actual modeled stator current $i_{\text{act}}$.

$$i_{\text{sinj}} = i_{\text{act}} - i_{\text{act}}$$

This current can be expressed as follows:

$$i_{\text{sinj}}^{' \alpha, \beta} = i_{\text{injpos}} \exp(j(\omega_t \theta + \frac{\pi}{2})) + i_{\text{injneg}} \exp(j(2\theta_t - \omega_t \theta + \frac{\pi}{2}))$$

As it can be seen the overall current consists of two components: positive sequence component and negative sequence component. The negative sequence component is proportional to the differential stator transient inductance and contains useful spatial information, and can be extracted by filtering [6] and [7].

Resulting PMSM rotor position is calculated as:

$$\theta_{\text{calc}} = \arctan\left(\frac{i_{\text{injneg}}}{i'}\right)$$

(4)

The estimated rotor position is available in time $t_{\text{calc}}$. Consequently, the real rotor position then will be:

$$\theta_{\text{real}} = \theta_{\text{calc}} + \omega_r t_{\text{calc}}$$

(5)

The time relations between measured, calculated and filtered quantities are shown in Fig. 3. From it is clear, that measured quantities have to be memorized during PMSM model calculation $t_{\text{model}}$. After carried-out the current injection response it’s possible to filter that signal and to extract it for the rotor position [8].

3. SIMULATION USING EXPERIMENTAL INPUT DATA

The model based methods require accurate knowledge of phase voltages and current for proper functionality. At low speeds, phase voltage reference and measured phase current are low, making it very difficult to separate from noise. Also distortion by inverter nonlinearities and model parameter deviation becomes significant with decreasing speed [5]. The differential equation system of the PMSM model (4th order) has been solved by forward difference numerical approximation. Equations of the PMSM are given as difference equations of the first order:

$$L_d \frac{di_q}{dt} = v_q - R_d i_q + P_p L_q \omega_j i_q$$

$$L_q \frac{di_q}{dt} = v_q - R_q i_q - P_p \omega_r (\Psi_{pm} + L_q i_q)$$

$$J_m \frac{d\omega_r}{dt} = \frac{3}{2} P_p [\Psi_{pm} i_q + (L_d - L_q) i_q - T]$$

$$\theta_r = \int \omega_r dt$$

(6)-(9)

At the low velocity or the dead-time voltage distortion becomes serious. The voltage measurement has been done as a reconstruction from measured DC-bus voltage $V_{dc}$, switching frequency $f_{\text{sw}}$ and dead-time $t_d$. Dead-time deviation voltage $V_{\text{e}}$ is calculated as is shown in (10), where $V_{dc}$ is a DC bus voltage, $t_d$ is a dead-time and $f_{\text{sw}}$ a inverter switching frequency.

$$V_{\text{e}} = f_{\text{sw}} t_d V_{dc}$$

(10)

The measured stator voltages serve as an excitation of the real and virtual model. They can be seen on Fig. 4 - 5.

![Fig. 3. Time relations between input and output PMSM quantities [2]](image-url)
The major differences between measured-real waveforms and simulated waveforms come from the idealized simulation motor model, transistor parasitic components, current shunt resistors, AD converter error and numerical error.

Used motor has small value of the stator resistance, see Tab. I. Three phase inverter has one shunt resistor in DC bus for the DC bus current measurement. Another shunt resistor is placed in every phase of the inverter.

**Table I**

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Rs</td>
<td>165 mΩ</td>
</tr>
<tr>
<td>Ld</td>
<td>150 µH</td>
</tr>
<tr>
<td>Lq</td>
<td>150 µH</td>
</tr>
<tr>
<td>Ψpm</td>
<td>8.55·10⁻³</td>
</tr>
<tr>
<td>Jm</td>
<td>8.10⁻⁸kgm²</td>
</tr>
<tr>
<td>F</td>
<td>0 Nms/rad</td>
</tr>
</tbody>
</table>

**Offline Identification Procedure**

Because of the idealized motor model and parasitic components the Simulink Parameter Estimation has been used in order to estimate the motor drive parameters. Basically we have looked for estimated parameters, with those estimated parameters waveforms are approaching to measured waveforms. Another reason to estimate the model parameters with higher accuracy is future usage of a rotor flux observer. Observer has an ability to work properly with the small parameters changes that is why parameter estimation is needed. Simulink Parameter Estimation software is a Simulink-based product for estimating and calibrating model parameters from experimental data. It provides the functions and operations used to:

1) Set up the problem.
2) Specify which model parameters to estimate.
3) Import and prepare the experimental data for parameter estimation (or preprocess).
4) View the estimation progress.
5) Validate the estimation results based on plots of measured versus simulated data and residuals.

Simulink Parameter Estimation software compares empirical data with data generated by a Simulink model. Using optimization techniques, the software estimates the parameter and (optionally) initial conditions of states such that a user selected cost function is minimized. The cost function typically calculates a least-square error between the empirical and model data signals. The stator resistance Rs, stator inductances in the rotating frame and PM flux Ψpm were measured.

The measured values are in Tab. II. The motor drive inertia Jm and friction factor F were estimated by Simulink Parameter Estimation software. The estimation of those parameters was profitable, because of the small values and calibration of the simulated data with measured ones. PMSM simulation model velocity ωr in Fig. 5 and rotor position θel in Fig. 6 is more accurate with estimated parameters Jm and F.

**Table II**

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Rs</td>
<td>17 mΩ</td>
</tr>
<tr>
<td>Ld</td>
<td>120 µH</td>
</tr>
<tr>
<td>Lq</td>
<td>120 µH</td>
</tr>
<tr>
<td>Ψpm</td>
<td>8.55·10⁻³</td>
</tr>
</tbody>
</table>

As it can be seen in Fig. 5 - 6 error between measured waveform and simulated waveform with estimated parameters is almost eliminated. Rotor position θel with estimated parameters has the constant error 0.024 rad/s. This error can be added to the calculated rotor position and the error is eliminated. Unlike the simulated electrical position θr depicted on Fig. 9 estimated parameter has been validated by different motor start-up with the same parameters as in Tab. II and Tab. III. The validation confirmed accuracy of θel and ωr with measured and estimated parameters.

**Table III**

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Jm</td>
<td>26.534·10⁻⁶ kgm²</td>
</tr>
<tr>
<td>F</td>
<td>562.23·10⁻⁶</td>
</tr>
</tbody>
</table>

**Improvements of virtual high frequency injection method**

Because of the slightly inaccuracy of the current equations in spite of the estimated and the measured parameters we have decided to complete VHFIM
structure including a back EMF observer. High accuracy of the current values is needed because of the first BP filter elimination.

Extended ElectroMotive Force (EEMF) based Sliding Mode Observer (SMO) of the salient PMSM can be expressed as (11).

$$\frac{d}{dt}\begin{bmatrix} \hat{i}_a \\ \hat{i}_b \end{bmatrix} = \begin{bmatrix} A & B \\ -k \end{bmatrix} \begin{bmatrix} \hat{\nu}_a \\ \hat{\nu}_b \end{bmatrix} - kH \begin{bmatrix} \hat{i}_a \\ \hat{i}_b \end{bmatrix}$$  \hspace{1cm} (11)

The coefficient $k$ is the feedback coefficient, the SMO parameters are expressed by (12). EEMF SMO has been used without PLL for VHFIM, because the estimation of the rotor position and the velocity is done through the simplified synchronous filter [4]. The outputs of EEMF SMO are the extended EMF $e\hat{e}_\alpha$ and $e\hat{e}_\beta$. Next block EEMF PMSM Model computes the acting current $i_{act}$, from the acting voltage $v_{act}$, estimated velocity $\omega_{el}$ and extended EMF $e\hat{e}$.

$$\frac{d}{dt}\begin{bmatrix} \hat{i}_{act} \\ \hat{i}_{beta} \end{bmatrix} = \begin{bmatrix} A & B \\ -k \end{bmatrix} \begin{bmatrix} \hat{\nu}_{act} \\ \hat{\nu}_{beta} \end{bmatrix} - \begin{bmatrix} \hat{e}e_{\alpha} \\ \hat{e}e_{\beta} \end{bmatrix}$$  \hspace{1cm} (12)

The current harmonic dependent on the rotor position $\theta_{el}$ is extracted from the acting current $i_{act}$ through the simplified synchronous filter.

The parameter estimation is very strong technique to simulate PMSM model accurately and fast with real voltages. Despite that fact the estimated parameters do not provide to reach requested performance without $\omega_{r}$ and $\theta_{el}$ errors.

4. IMPLEMENTATION IN POWER PC-MPC

All above given equations needed to rotor position determination have to be calculated in real time to provide sampling of FOC (or other) control of PMSM. System represented by MPC5567 with e200z6 single precision floating point core, uses IEEE754 compatible operands. Although the core is single precision, MPC5567 supports double floating point operands with software library. However, using of software floating point library dramatically increase the computing time. Therefore, for computing of differential PMSM equations, only single precision with two 32bit operands is used. With 132MHz system clock the e200z6 core needs 4 cycles for multiplication of 2 operands and 24 cycles for dividing of 2 single precision floating point operands.

The equations for PMSM model can be implemented as a classic C language or intrinsic functions. Intrinsic functions offers better performance, but C language is easy to use. It depends on user, which approach will choose. Fig.8 shows accuracy of mechanical velocity calculation in Power PC compared to simulation in Matlab.
Fig. 8. Mechanical velocity or Power PC-MPC - Matlab

Fig. 8a. Mechanical velocity error $\Delta \omega_r$ of Power PC-MPC in detail

Calculation of mechanical rotor speed during transient from 100rad/sec to 300rad/sec at steady/state is presented in Fig. 9.

Calculation of mechanical rotor position during transient from 100 rad/sec to 300 rad/sec at steady/state is presented in Fig. 10.

Fig. 9. Mechanical speed during acceleration of the unloaded PMSM 100 - 300 rad/s

Fig. 10. Mechanical position during acceleration of the unloaded PMSM 100 - 300 rad/s

Carried-out results are practically the same as [9] with Matlab simulation but calculation time of rotor position with Power PC-PCM reached 29$\mu$sec (by the Program Counter) - 31$\mu$sec (by the Timer). So it’s possible to use it for main control loop of FOC of the drive (50-100 $\mu$sec). It has been used 32 bit single-precision operands and FLASH program memory during calculation. Using floating point math functions, the calculation time could be rather shorter.

5. CONCLUSIONS

The PMSM adaptive model was used to eliminate the current computation inaccuracy despite the SMPM parameters estimation. Essential part of the adaptive model is EEMF sliding mode observer. EEMF SMO is easy to design and it’s advantage is the robustness. Drawback of the sliding mode observers is lower accuracy at low speeds as it’s depicted on Fig. 10. The steady-state mechanical velocity error is less than 5 rad/s and the steady-state mechanical position error is less than 1.89 deg.

It has been shown, that speed computing speed of Power Pc-MPC 5567 is about 10 times higher than the DSC 56F8013. However, DSC with its peripherals is dedicated for vector control of PMSM, in spite its lower computing capabilities. MPC offers solutions with very good computing performance (floating point), also with high speed peripherals – especially eTPU.

Acknowledgments

The authors wish to thank for the financial support to VEGA Agency for the project No. 1/0470/09 and R&D operational program Centre of excellence of power electronics systems and materials for their components...
REFERENCES


