FPGA Implementation of Model Predictive Control With Constant Switching Frequency for PMSM Drives

Zhixun Ma, *Student Member, IEEE,* Saeid Saeidi, *Student Member, IEEE,* and Ralph Kennel, *Senior Member, IEEE*

***Abstract*—Field programmable gate array (FPGA) implementation of a model predictive control with constant switching frequency (MPC-CSF) for a permanent magnet synchronous machine (PMSM) is proposed. The basic finite states MPC (FS-MPC) can be combined with a pulsewidth modulation (PWM) modulator because of an effective cost function optimization algorithm in which voltage vectors are dynamically selected and calculated through iteration based on the idea similar to dichotomy. Using model-based design (MBD), MPC-CSF is implemented on an FPGA with parallel and pipeline processing techniques in short execution time. Functionality simulation analysis presents that MPC-CSF is much robust to parameter variations. Experimental results illustrate that MPC-CSF has good dynamic performance for PMSM drives.**

***Index Terms*—Field programmable gate array (FPGA), model predictive control (MPC), optimization algorithm, permanent magnet synchronous machine (PMSM).**

I. Introduction

PERMANENT magnet synchronous machines (PMSMs) are widely used in electrical drives nowadays because of their several advantages: high efficiency, high power density, and high torque-to-weight ratio. High-performance PMSM drives require high dynamic control strategies. Recently, with the rapid development of computing, predictive control becomes attractive in control of PMSM drive systems for high dynamic.

Predictive control can be concluded as any algorithms that predict the future behavior of a system to select the proper control action using a certain optimization criterion. A well-known earlier predictive control is the deadbeat control [1]. Deadbeat control can make the error become zero in the next sampling interval using an optimal control action; therefore, it is considered as one of the fastest current control schemes [2], [3]. However, deadbeat control is highly sensitive to modeling errors and noise. Recently, model predictive control (MPC) is an emerging control strategy for PMSM drives [4]–[8]. The basic idea of MPC is to predict the future behavior of control variables in the time domain based on a model of the control system and to choose the control action according to a minimization of a cost function. The main advantage of MPC is that it is easy to control nonlinearities and to consider constraints of the system.

MPC applied to power electronics and drives can be classified into two categories: 1) continuous MPC; and 2) finite set MPC (FS-MPC) [9]–[11]. In continuous MPC, the output is a continuous voltage vector which can be applied by a PWM modulator. In FS-MPC, the controller directly outputs a switching state to the inverter. One well-known continuous MPC strategy is a generalized predictive control (GPC) which has a high prediction horizon but is unconstrained. GPC belongs to the long-range predictive controllers and is commonly restricted to a linear system model and constant system parameters. The recently developed explicit MPC can be used for both continuous and finite set MPCs, where an offline solution is stored as a lookup table to solve the computation problem [12]. FS-MPC considers only a finite set of possible switching states of the inverter and solves the cost function for each of them and chooses a switching state which minimizes the cost function. It can be easily implemented when the prediction horizon is set to one step. However, for the control of PMSM drives, basic FS-MPC usually causes large current and torque ripples. The main reasons are: one voltage vector will be applied to the inverter in the whole sampling interval; the inductance of a PMSM is relatively small. To overcome this major drawback, a variable switching time method has been introduced. One approach [13] is based on hysteresis bounds, which keep a fixed torque ripple using a variable switching frequency. Another approach is from the idea of direct mean torque control. In [14], one zero switching state follows one active switching state in every sampling period. In [15], the minimization of torque ripple is the object of the switching time calculation. Another direction is combing an FS-MPC with external modulator. In [16], to obtain constant switching frequency and low sampling frequency, MPC is combined with discrete space vector modulation by adding virtual state vectors. In [17], voltage vectors are dynamically selected and calculated by an optimization algorithm.

Field programmable gate arrays (FPGAs) are one of the best candidates for the implementation of MPC [18]–[20]. Traditionally, FPGAs are used as hardware interface logic circuits. Nowadays, the price of the reprogrammable logic

Manuscript received January 08, 2014; revised May 29, 2014; accepted July 11, 2014. Date of publication July 30, 2014; date of current version November 04, 2014. Paper no. TII-14-0020.

The authors are with the Institute for Electrical Drive Systems and Power Electronics, Technical University of Munich, Munich 80333, Germany (e-mail: zhixun.ma@tum.de; saeid.saeidi@tum.de; ralph.kennel@tum.de).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TII.2014.2344432

has reduced dramatically, and the density of logic elements on a single FPGA chip has increased tremendously [21]. Moreover, as hardware, FPGAs have several advantages surpassing microcontrollers or DSPs for drive control: parallel processing capabilities can shorten the computational time to a large extent, which results in a lower control delay and better dynamic performance. All dedicated interfaces and control algorithms can be implemented in a single chip, which is flexible and friendly to PCB design. At present, model-based design (MBD) for FPGA implementation of complex control systems becomes attractive because of the time-saving and great flexibility in simulation and debugging [22], [23].

This paper presents an FPGA implementation method for an MPC with constant switching frequency (MPC-CSF) for PMSM drives. The basic FS-MPC can be combined with a PWM modulator because of an effective cost function optimization algorithm in which voltage vectors are dynamically selected and calculated through loops. Using an MBD method, MPC-CSF is implemented with a pipelined parallel structure in a low-cost single FPGA chip. Experimental results present that MPC-CSF has good dynamic performance as FS-MPC. Furthermore, the switching frequency is constant. Therefore, it is proper for industrial application.

A previous version of this work is proposed in [17]. A detailed study is then made. Indeed, the cost function optimization algorithm is elaborated in a new section. An in-depth description of FPGA implementation is added. New experimental results are also provided.

II. Machine Model

The stator voltage equation of a PMSM in cartesian stator coordinates (α, β) is given as

$$\boldsymbol{u}_s^s = R_s \boldsymbol{i}_s^s + \frac{d\boldsymbol{\psi}_s^s}{dt} \tag{1}$$

where $\boldsymbol{u}_s^s = [u_\alpha, u_\beta]^T$ and $\boldsymbol{i}_s^s = [i_\alpha, i_\beta]^T$ are the stator voltage and current, R_s is the stator resistance, and $\boldsymbol{\psi}_s^s$ is the stator flux linkage.

The flux linkage in the stator windings is assumed to be caused by the stator current and the rotor magnetization, i.e.,

$$\boldsymbol{\psi}_s^s = \mathbf{L}_s^s \boldsymbol{i}_s^s + \boldsymbol{\psi}_{\mathrm{pm}}^s. \tag{2}$$

The rotor flux linkage $\boldsymbol{\psi}_{\mathrm{pm}}^s$ can be transformed from its constant rotor coordinate value $\boldsymbol{\psi}_{\mathrm{pm}}^r$

$$\boldsymbol{\psi}_{\mathrm{pm}}^s = \mathbf{C}\boldsymbol{\psi}_{\mathrm{pm}}^r = \begin{bmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{bmatrix} \begin{bmatrix} \Psi_{\mathrm{pm}} \\ 0 \end{bmatrix} \tag{3}$$

where θ is the rotor angle, $\mathbf{C}$ is the transformation matrix, and Ψ_{pm} is the magnitude of the flux generated by the rotor magnet. In this work, the control algorithm is tested on a surface-mounted PMSM (SMPMSM), so the inductance tensor $\mathbf{L}_s^s$ can be considered as a constant value

$$\mathbf{L}_s^s = \begin{bmatrix} L_s & 0 \\ 0 & L_s \end{bmatrix} \tag{4}$$

where L_s is the stator inductance.

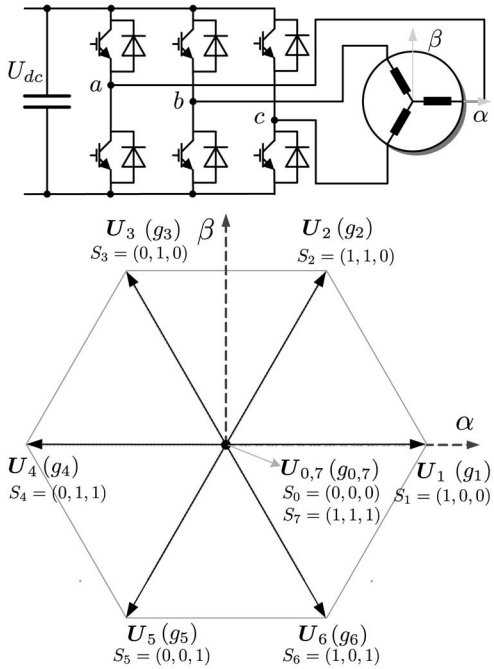

Fig. 1. Three-phase two-level converter-fed PMSM drive system and its finite voltage vector set.

The torque produced by the machine can be expressed mathematically by the vector product of stator current and stator flux linkage

$$T_m = \frac{3}{2} p \boldsymbol{i}_s^s \cdot j\boldsymbol{\psi}_s^s = \frac{3}{2} p(\psi_\alpha i_\beta - \psi_\beta i_\alpha) \tag{5}$$

where p is the number of pole pairs of the machine and $\boldsymbol{\psi}_s^s = [\psi_\alpha, \psi_\beta]^T$ is the stator flux linkage.

The equation used to model the dynamics is given by (6), where J_m is the system inertia and B_m is the system friction coefficient. Equation (7) presents that the electrical speed ω_r (in rad/s) is related to the mechanical speed ω_m by the number of pole pairs p

$$J_m \frac{d\omega_m}{dt} = T_m - T_l - B_m \omega_m \tag{6}$$

$$\omega_r = p\omega_m. \tag{7}$$

III. FS-MPC for PMSM Drives

Considering the discrete nature of power converters, it is possible to simplify the optimization of MPC without using PWM modules. Since the possible switching states of power converters is finite, the optimization problem is reduced to evaluate all possible states and to select the one which minimizes the given cost function. For a three-phase two-level inverter-fed PMSM drive system, there are eight possible switch states and eight corresponding cost function values in every prediction step, as shown in Fig. 1. The number of calculations of all feasible switching states will increase exponentially by increasing the prediction horizon. However, only one-step prediction satisfies the requirements in most cases [24].

In FS-MPC, all predictions are evaluated by the cost function. The switching state which makes the cost function minim

is selected as the future control actions. In this paper, the prediction is based on a discrete model of the machine which is obtained by approximating the derivation by Euler's formula. Equations (1)–(3) are used to predict the stator current at the next sampling instant $\boldsymbol{i}_s^s(k+1)$ based on a given stator voltage $\boldsymbol{u}_s^s(k)$ and the present sampling current $\boldsymbol{i}_s^s(k)$. Equation (8) presents the current prediction

$$\boldsymbol{i}_s^s(k+1) = \boldsymbol{i}_s^s(k) + \frac{T_s}{L_s}[\boldsymbol{u}_s^s(k) - R_s\boldsymbol{i}_s^s(k) - \mathbf{C}(\frac{\pi}{2})\omega_r(k)\boldsymbol{\psi}_{\mathrm{pm}}^s] \tag{8}$$

where T_s is the sampling interval and L_s is the inductance. Equation (1) can be used for the flux linkage prediction as well. The torque prediction presented in (10) is from (5)

$$\boldsymbol{\psi}_s^s(k+1) = \boldsymbol{\psi}_s^s(k) + T_s[\boldsymbol{u}_s^s(k) - R_s\boldsymbol{i}_s^s(k)] \tag{9}$$

$$T_m(k+1) = \frac{3}{2}p\boldsymbol{i}_s^s(k+1)\cdot j\boldsymbol{\psi}_s^s(k+1). \tag{10}$$

The cost function is selected for torque and flux control. This FS-MPC is also called predictive torque control (PTC) [1]

$$g = |T_m^* - T_m^{k+1}| + \lambda||\psi_s^*| - |\psi_s^{k+1}|| \tag{11}$$

where T_m^* and ψ_s^* are the reference values of torque and flux, respectively, and T_m^{k+1} and ψ_s^{k+1} are the predictions of the torque and flux in the next sampling step $k+1$. λ is the weighting factor. In FS-MPC, all eight possible switching are considered at every sampling instant. For basic FS-MPC, it applies single switching state for the entire sampling interval which results large current and torque ripples.

IV. MPC-CSF for PMSM Drives

A. MPC-CSF for PMSM Drive System

The structure of MPC-CSF for PMSM drives is shown in Fig. 2. In the MPC-CSF scheme, the prediction process and the cost function selection are the same as FS-MPC. One difference is that the output voltage vector of MPC-CSF is continuous, and the PWM module can be applied to the control system. This is the reason for an effective cost function solution algorithm in MPC-CSF. For every iteration, a set of voltage vectors is considered and every voltage vector depends on previous suboptimal vector $\boldsymbol{u}_s^{k+1}$. The initial set of voltage vectors is the same for starting of the algorithm every time.

The basic part of the cost function selected in MPC-CSF is the same as that of FS-MPC. In this case, the over current protection is also added

$$G = |T_m^* - T_m^{k+1}| + \lambda||\psi_s^*| - |\psi_s^{k+1}|| + \lambda_I \tag{12}$$

$$\lambda_I = \begin{cases} 0, & \text{if } i_s < I_{s.\max} \\ \infty, & \text{else} \end{cases}$$

where λ is the weighting factor. There is no standard to select it. In this paper, λ is selected to be T_n/ψ_n to give torque and flux the same weight, where T_n and ψ_n are the rated values for torque and stator flux, respectively.

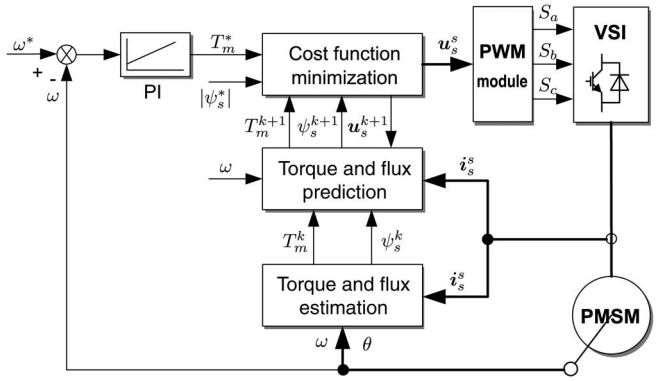

Fig. 2. MPC-CSF for PMSM drive system.

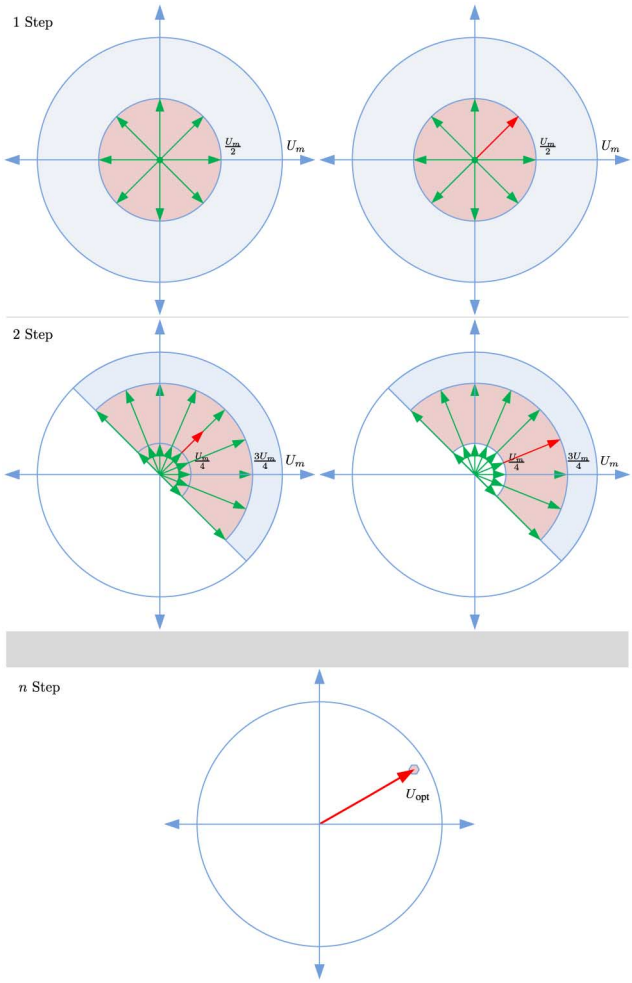

Fig. 3. Cost function optimization algorithm.

B. Cost Function Optimization Algorithm

The basic idea of the cost function optimization algorithm is dichotomy which is an important idea for computer systems. As shown in Fig. 3, all voltage vectors are inside the circular plane with a radius of U_m (the maximum available voltage). In every switching interval, the initialization of the

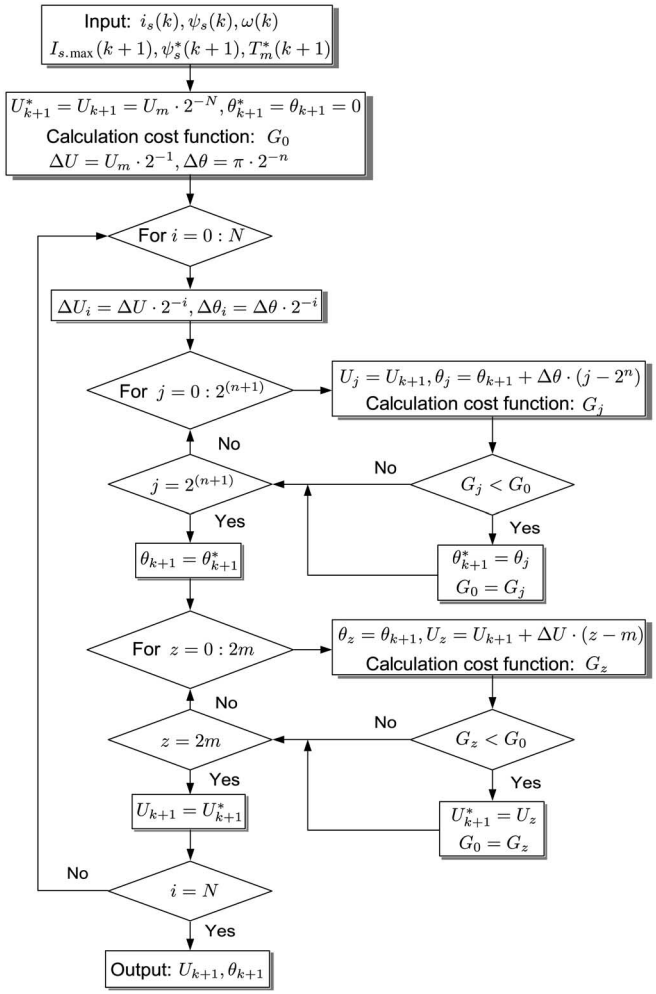

Fig. 4. Flowchart of the cost function optimization algorithm for MPC-CSF.

algorithm is always the same. Here, for example, there are eight voltage vectors symmetrically located inside the circular plane with a radius of $\frac{U_m}{2}$, i.e., with the amplitude of $\frac{U_m}{2}$ and $\frac{\pi}{4}$ phase difference with each other. In the first step, one of the eight vectors is selected as the optimal vector (the red color vector or bold color vector for black and white) based on the cost function. In the next step, based on the optimal vector selected in the last step, 18 new vectors, whose amplitude and phase angle are added or subtracted by half of last step optimal vector, are selected to solve the cost function. Therefore, the search area decreases to half of the last step. After 12–16 times iteration, the optimal vector is selected as the reference vector for the PWM modulator.

C. Flowchart of the Optimization Algorithm

Fig. 4 shows the flowchart of the proposed cost function solution algorithm. N defines the resolution of the controlled voltage signal. For industrial application, it is proper to choose N between 12 and 16. This means that the resolution of the voltage signal is 12–16 bit. n defines the search resolution of the phase of the voltage vector in every iteration. In this case, $n \in \{1, 2, 3\}$. Correspondingly, $2m$ defines the search resolution of the amplitude of the voltage vector in every iteration. It is enough to set $m \in \{1, 2, 3\}$.

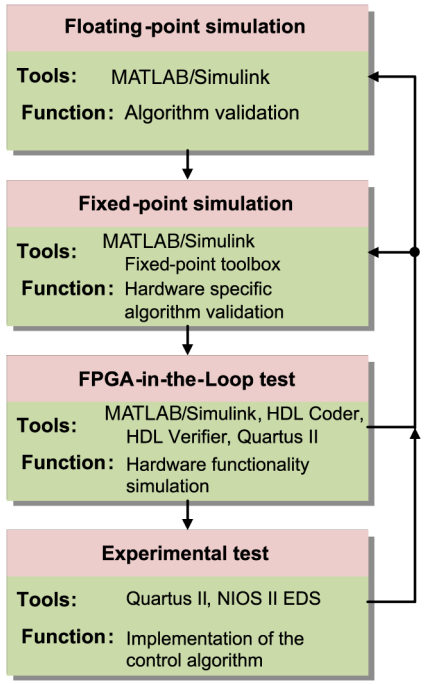

Fig. 5. MBD flow for FPGA-based controllers.

D. Phase Calculation Loop

In the phase calculation loop, the voltage vector rotates in two directions, each by 2^n steps, around its axes. Totally, $2^{n+1} + 1$ voltage vectors, whose phases vary $\Delta\theta \cdot 2^{-i}$ sequentially, are calculated. The phase of the voltage vector which minimizes the cost function is selected as the output of the phase calculation loop.

E. Amplitude Calculation Loop

After updating the phase of the voltage vector, the amplitude is modified in a separate loop. The amplitude can be increased or decreased by $\Delta U \cdot 2^{-i}$ in each iteration.

V. Model-Based Design for FPGA-Based Controllers

Fig. 5 presents the MBD process for FPGA-based controllers. First, a floating-point model is constructed in MATLAB/Simulink. The control algorithm is validated in this phase. Second, the floating-point model is converted to a fixed-point data type model which can be implemented on an FPGA-based controllers. Both fixed-point Simulink models and MATLAB function blocks are used to describe the algorithm. The structure of the whole algorithm is usually required to be revised for implementation on FPGA-based controller. Pipeline and resource sharing are the two approaches usually used for optimization. The dedicated code that is similar to the corresponding HDL code can be used in the MATLAB function blocks to realize logic functions. The idea behind these two code types is the same, only the description is different. Therefore, it is flexible and convenient to describe the hardware structure using both Simulink and MATLAB function blocks. Third, the dedicated fixed-point model is

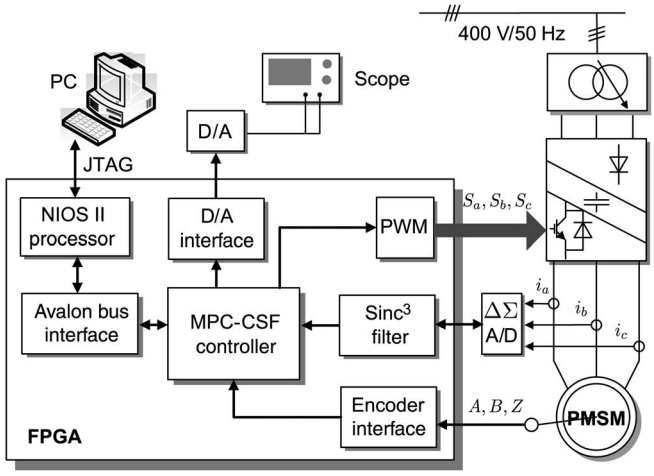

Fig. 6. Block diagram of the designed FPGA-based controller for a PMSM drive.

tested by FPGA-in-the-Loop simulation. There are three steps in this phase.

1) The control model is converted to VHDL code using HDL Coder.
2) Using HDL verifier, MATLAB calls Quartus II to compile the generated VHDL files and constructs the Ethernet commutation interface between Simulink and the FPGA development board (Altera DE2-115 development board).
3) The compiled VHDL program is downloaded to the FPGA board, and FPGA-in-the-Loop test is ready.

When the test is started, the PMSM mathematical model works in MATLAB/Simulink and the control algorithm works on the FPGA board synchronously with Ethernet communication. The process can be seen as a functionality simulation. Last, the control algorithm is tested on the real test bench. The VHDL files are recompiled by Quartus II, and NIOS II EDS is responsible for sending and receiving test signals and parameters. The target FPGA control board is based on a low-cost Cyclone III FPGA chip (EP3C40F484C7).

Fig. 6 shows the block structure of the designed FPGA-based controller for a PMSM drive. For realizing the MPC-CSF control algorithm, the blocks such as PWM, Sinc^3 filter, encoder interface, D/A interface, and MPC-CSF controller are designed by MATLAB using the MBD method. The NIOS II processor and Avalon bus interface are designed in Quartus II. NIOS II EDS software is used for programming the NIOS II processor. The NIOS II EDS software and the NIOS II processor communicate with each other by JTAG.

VI. FPGA Implementation of MPC-CSF for a PMSM

The FPGA-based implementation architecture of the presented MPC-CSF algorithm is shown in Fig. 7. By MATLAB/Simulink and HDL Coder, MBD method is used for realizing the control algorithm. For the calculation of the trigonometric and quadratic functions, the CORDIC (COordinate Rotation Digital Computer) algorithm is used. The calculation algorithm is realized in pipeline-parallel architecture. The entire number of pipeline steps is 40. Every time when input data go through all of these pipeline stages, the cost function for one voltage vector is evaluated. It takes 40 clock events which mean 0.5 μs for 80 MHz clock frequency. In every iteration, 18 voltage vectors need to be calculated for this implementation. It takes 18 clock events which mean 0.225 μs. For the calculation of the optimal voltage vector, the same procedure must be repeated, e.g., 15 times. Therefore, the whole time delay is 10.875 μs. At every clock event new input data can be inserted because of the pipeline structure. The output data will also be ready in the same way after 40 clock events. Therefore, the total computational delay in (13) is significantly reduced

$$T_D = (N_p + N_v)/F_c \cdot N_i \tag{13}$$

where T_D is the total computational delay, N_p is the number of pipeline steps, N_v is the number of voltage vectors to be evaluated, F_c is the clock frequency, and N_i is the number of iterations.

Every pipeline stage consists of several parallel combinatorial logic (CL) blocks. The CL blocks have different time delays. Consequently, parallel tasks in every stage are synchronized with the clock both at the input and the output. For every CL block, the maximum delay is considered to satisfy the timing constraint of the clock frequency.

The area performance of the MPC-CSF-based control system for PMSM drives is shown in Table I. The implementation contains not only MPC-CSF itself but also all interfaces and debugging hardware such as A/D interface and the NIOS II processor. This implementation result presents that it is feasible for industrial applications.

VII. Sensitivity of Model Parameters

For MPC-CSF, the prediction and the cost function optimization algorithm are all based on the machine model; therefore, it is necessary to evaluate the sensitivity of parameter variation in the model. In order to avoid other interference and to clarify the effect of parameter variation, MATLAB/Simulink environment is used for the analysis. The fixed-point data-type Simulink model used in the simulation can be directly used for VHDL code generation. Hence, the operation can be seen as functionality simulation.

The parameters of the PMSM in the simulation are presented in Table II, the same as the PMSM used for the experiment. In the simulation, the PWM switching frequency is 10 kHz for MPC-CSF; PMSM is started with a 500 rpm speed step, then a 5 Nm torque is given at 0.06 s. The control performance based on MPC-CSF is shown in Figs. 8 and 9 with different parameters. As a result, the proposed algorithm for PMSM drives is very robust against parametric uncertainties. In Fig. 8, the value of the inductance in the control algorithm is set from 3 to 0.2 times of the real inductance. In this range inductance variation, MPC-CSF can work well except unexpected current and torque ripples. It is interesting to see that the performance of MPC-CSF with 3 times inductance variation is similar with the performance of a basic FS-MPC. For the resistance variation, Fig. 9 shows that it has limited effect on this control algorithm. The difference

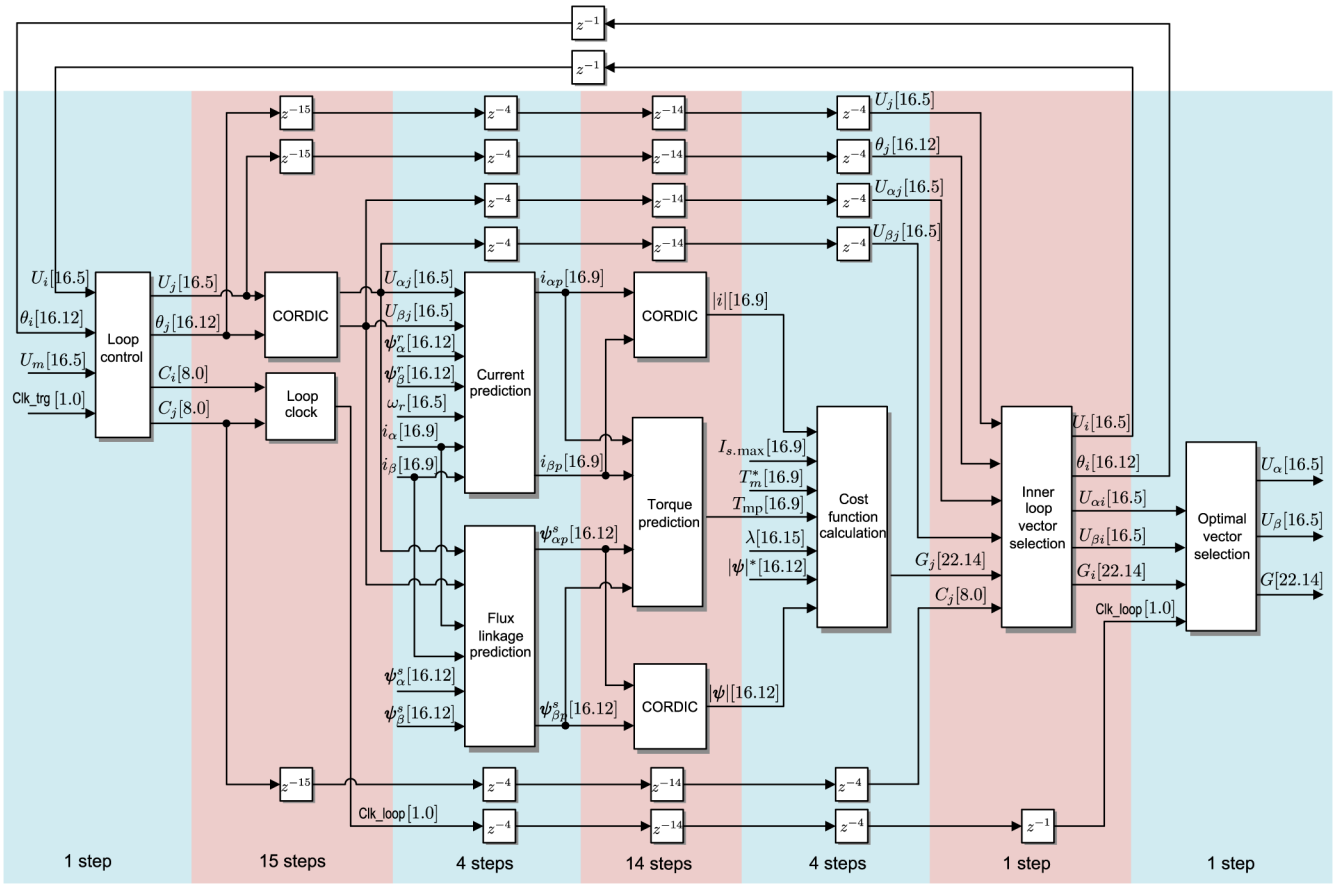

Fig. 7. FPGA implementation architecture of MPC-CSF.

TABLE I
FPGA AREA UTILIZATION OF MPC-CSF IMPLEMENTATION

EP3C40F484	Resources	Design usage
Logic elements	39 600	15 470 (39%)
Registers	39 600	6111 (15%)
Embedded 9-bit multiplier	252	38 (15%)

TABLE II
NOMINAL PARAMETERS OF SMPMSM

Rated power P_N	2.7 kW
Rated torque τ_{MN}	8.5 Nm
Rated current (eff.)	5 A
Rated speed ω_{MN}	3000 rpm
Pole pairs n_p	3
Rated voltage U_N (eff.)	400 V
Stator inductance L_s	9 mH
Stator resistance R	1.3 Ω
PM flux ψ_{PM}	0.41 Vs

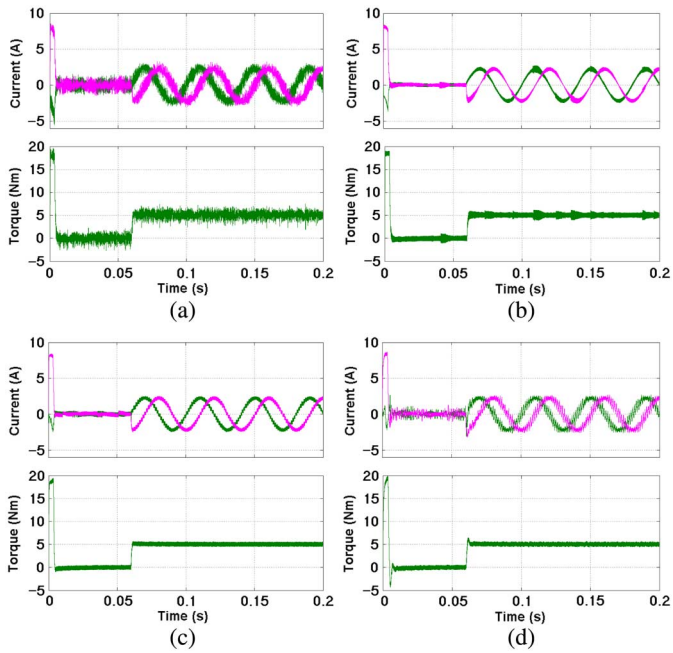

Fig. 8. PMSM drive performance using MPC-CSF with different inductance variations (simulations). (a) $\hat{L}_s = 3L_s$. (b) $\hat{L}_s = 2L_s$. (c) $\hat{L}_s = 0.5L_s$. (d) $\hat{L}_s = 0.2L_s$.

is only the maximum current and torque in the dynamic start period of PMSM with $\hat{R}_s = (0 \sim 5)R_s$.

The effect of current offset on the torque and flux linkage is also tested and analyzed. Fig. 10(a) and (b) shows the torque and flux linkage performance with 1 A current offset of Phase A and all three phases, respectively. With one-phase current offset, the torque is fluctuant because of the unbalanced current. Three phase with the same current offset has no effect on the torque and flux linkages since there is no offset after Clarke transformation.

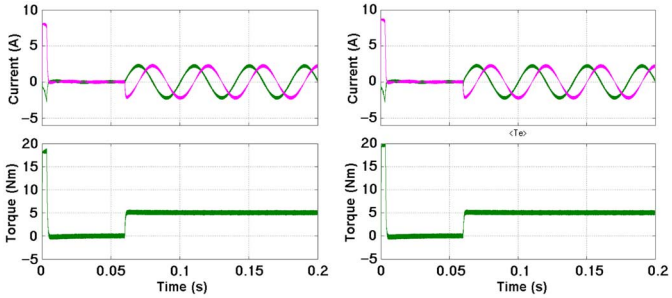

Fig. 9. PMSM drive performance using MPC-CSF with different resistance variations (simulations). (a) $\hat{R}_s = 0$. (b) $\hat{R}_s = 5R_s$.

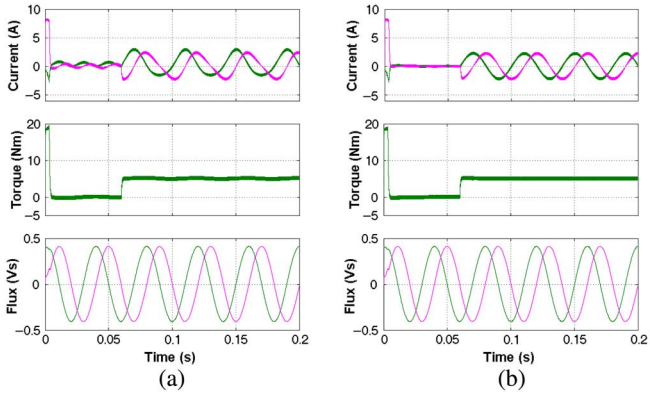

Fig. 10. PMSM drive performance using MPC-CSF with current offset (simulation). (a) 1 A current offset of Phase A. (b) 1 A current offset of all three phases.

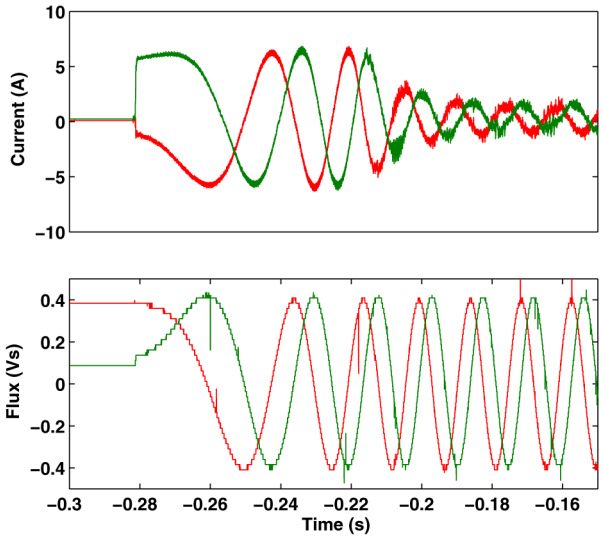

Fig. 11. Flux Linkage in the dynamic process.

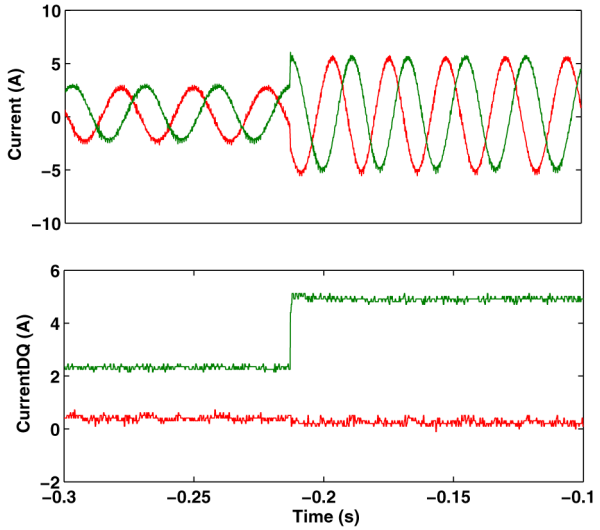

Fig. 12. MPC-CSF for PMSM drives: $\alpha\beta$-axis and dq-axis currents performance with a torque step from 4 to 8 Nm.

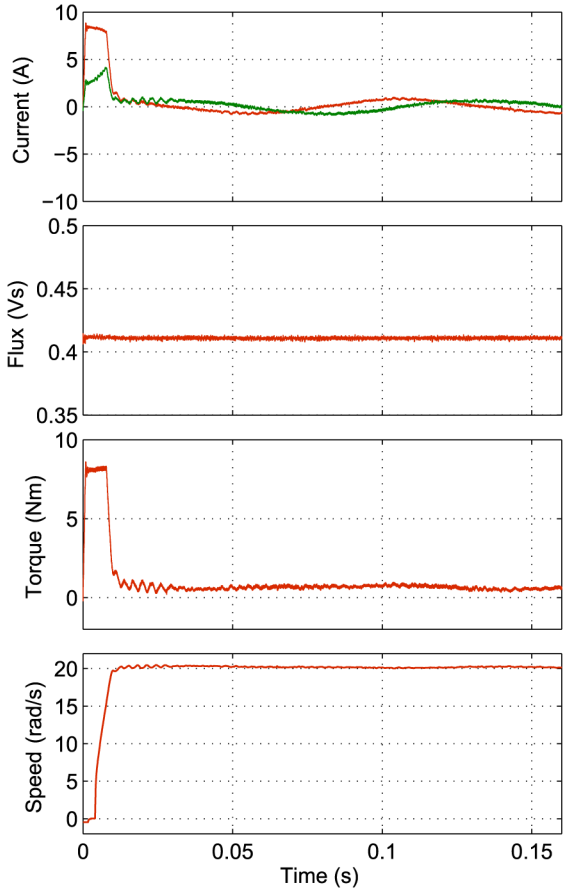

Fig. 13. MPC-CSF for PMSM drives: $\alpha\beta$-axis current, the flux linkage, the torque, and the speed performance with a speed step from 0 to 20 rad/s.

VIII. Experimental Results

The presented MPC-CSF control algorithm was experimentally tested on a commercial 3-pole-pair SMPMSM servo motor made by Kollmorgen Seidel Corporation, fed by a two-level voltage source inverter (VSI), with a 10 kHz PWM switching frequency. The motor parameters are given in Table II. The dc-link voltage was 300 V and the rated output voltage of the inverter was 230 V. The shaft of the SMPMSM was mechanically connected to an induction motor load. An incremental encoder was fixed in the shaft to obtain the actual rotor speed and position. In this work, there is no voltage sensor in the test bench. The dc-linkage voltage is seen as a constant since there are proper dc-link capacitors. In FPGA implementation of the cost function optimization algorithm,

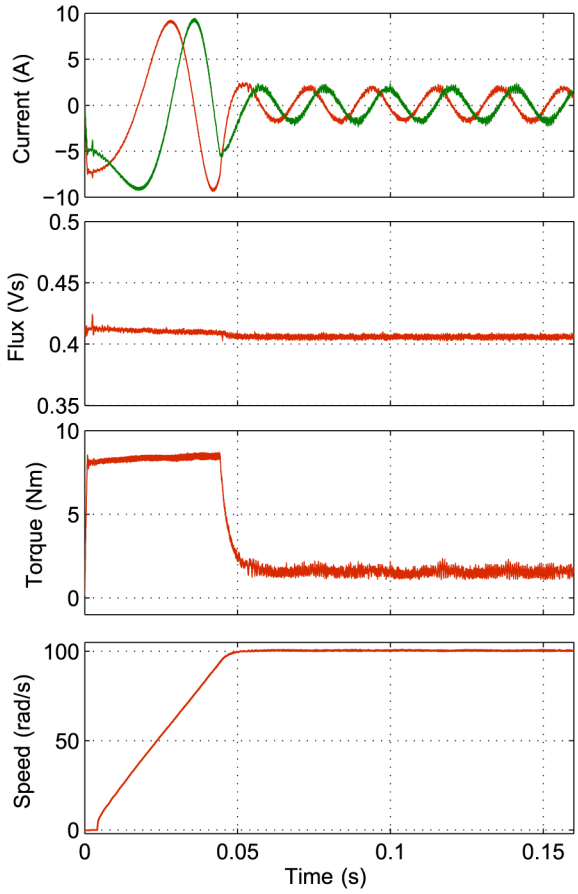

Fig. 14. MPC-CSF for PMSM drives: $\alpha\beta$-axis current, the flux linkage, the torque, and the speed performance with a speed step from 0 to 100 rad/s.

we can use CORDIC algorithm to calculate u_α and u_β in the stationary frame when the magnitude of dc-link voltage (the maximum available voltage) and rotor position are both obtained. After calculation, as shown in Fig. 7, the optimal vector is selected as the voltage reference for the PWM modulator.

Fig. 11 shows the $\alpha\beta$-axis current and flux linkage performance in the dynamic start process with a 150 rad/s speed. The flux was stable when the current dynamically changed. Fig. 12 shows the $\alpha\beta$-axis current and dq-axis current dynamic performance when the torque is changed from 4 to 8 Nm. The current can step to its reference value in one PWM switching interval and there is no coupling in the dq-axis current. The prerequisite of decoupling in the dq-axis current is that there is no constraint over control input that is mainly maximum voltage. Otherwise, the controller may not be able to cancel the coupling term due to limit of control voltage vectors. Decoupling of independent control parameters is an important issue for control of multi-input multi-output (MIMO) systems. In the proposed control scheme, a multiparametric optimization algorithm ensures the decoupling of torque and flux control loops. It makes the controller suitable for field weakening and minimization of losses. It must be, however, mentioned that decoupling can be maintained while no constraint is active, otherwise the cost function allows the controller to find the best compromise between all control objectives with respect to weighting factors. In Figs. 13 and 14, the speed and torque dynamic response, current, and flux linkage performance of MPC-CSF for PMSM drives are presented with a speed step from 0 to 20 rad/s and from 0 to 100 rad/s, respectively. It can be seen that the proposed MPC-CSF control algorithm has good dynamic performance, meanwhile, can keep the current ripple in a small range.

IX. Conclusion

FPGA implementation of an MPC scheme, MPC-CSF, for a PMSM is proposed. Due to the effective cost function optimization algorithm based on the idea of dichotomy, the optimization output voltage vector in the control region can be dynamically selected and calculated in a short time. This makes FS-MPC to easily combine with a PWM modulator. Therefore, MPC-CSF preserves the same dynamic performance as FS-MPC with constant switching frequency. At the same time, the current and torque ripples are keep in a small range because the proper voltage vector is selected in every 100 μs switching interval. Furthermore, functionality simulation analysis presents that MPC-CSF is much robust to parameter variations. Using MBD, MPC-CSF is fully implemented on a low-cost FPGA chip with pipeline and parallel structures to achieve a short execution time. Experimental results present that MPC-CSF has good dynamic performance for PMSM drives and is proper for industrial application.

References

[1] P. Cortes, M. P. Kazmierkowski, R. M. Kennel, D. E. Quevedo, and J. Rodriguez, "Predictive control in power electronics and drives," *IEEE Trans. Ind. Electron.*, vol. 55, pp. 4312–4324, Dec. 2008.

[2] L. Malesani, P. Mattavelli, and S. Buso, "Robust dead-beat current control for PWM rectifiers and active filters," *IEEE Trans. Ind. Appl.*, vol. 35, pp. 613–620, May/Jun. 1999.

[3] H. Moon, H. Kim, and M. Youn, "A discrete-time predictive current control for PMSM," *IEEE Trans. Power Electron.*, vol. 18, pp. 464–472, Jan. 2003.

[4] F. Morel, X. Lin-Shi, J. M. Retif, B. Allard, and C. Buttay, "A comparative study of predictive current control schemes for a permanent-magnet synchronous machine drive," *IEEE Trans. Ind. Electron.*, vol. 56, pp. 2715–2728, Jul. 2009.

[5] J. Stumper, A. Dotlinger, J. Jung, and R. Kennel, "Predictive control of a permanent magnet synchronous machine based on real-time dynamic optimization," in *Proc. 2011 14th Eur. Conf. Power Electron. Appl. (EPE'11)*, 2011, pp. 1–8.

[6] S. Bolognani, S. Bolognani, L. Peretti, and M. Zigliotto, "Design and implementation of model predictive control for electrical motor drives," *IEEE Trans. Ind. Electron.*, vol. 56, pp. 1925–1936, Jun. 2009.

[7] H. Zhu, X. Xiao, and Y. Li, "Torque ripple reduction of the torque predictive control scheme for permanent-magnet synchronous motors," *IEEE Trans. Ind. Electron.*, vol. 59, pp. 871–877, Feb. 2012.

[8] M. Preindl and S. Bolognani, "Model predictive direct speed control with finite control set of PMSM drive systems," *IEEE Trans. Power Electron.*, vol. 28, pp. 1007–1015, Feb. 2013.

[9] S. Kouro, P. Cortes, R. Vargas, U. Ammann, and J. Rodriguez, "Model predictive control—A simple and powerful method to control power converters," *IEEE Trans. Ind. Electron.*, vol. 56, pp. 1826–1838, Jun. 2009.

[10] T. Geyer, "Computationally efficient model predictive direct torque control," *IEEE Trans. Power Electron.*, vol. 26, pp. 2804–2816, Oct. 2011.

[11] R. Kennel, A. Linder, and M. Linke, "Generalized predictive control (GPC)-ready for use in drive applications?" in *Proc. IEEE 32nd Annu. Power Electron. Spec. Conf. (PESC)*, 2001, vol. 4, pp. 1839–1844.

[12] S. Bolognani, R. Kennel, S. Kuehl, and G. Paccagnella, "Speed and current model predictive control of an IPM synchronous motor drive," in *Proc. IEEE Int. Elect. Mach. Drives Conf. (IEMDC)*, 2011, pp. 1597–1602.

[13] T. Geyer, G. Papafotiou, and M. Morari, "Model predictive direct torque control—part I: Concept, algorithm, and analysis," *IEEE Trans. Power Electron.*, vol. 56, pp. 1894–1905, 2009.
[14] M. Pacas and J. Weber, "Predictive direct torque control for the PM synchronous machine," *IEEE Trans. Ind. Electron.*, vol. 52, pp. 1350–1356, 2005.
[15] P. Landsmann, P. Stolze, and R. Kennel, "Optimal switching time calculation in predictive torque control," in *Proc. IEEE IEEE 8th Int. Conf. Power Electron. Energy Convers. Congr. Expo Asia (ICPE & ECCE)*, 2011, pp. 923–930.
[16] S. Vazquez *et al.*, "Model predictive control with constant switching frequency using a discrete space vector modulation with virtual state vectors," in *Proc. IEEE Int. Conf. Ind. Technol. (ICIT'09)*, 2009, pp. 1–6.
[17] Z. Ma, S. Saeidi, and R. Kennel, "Continuous set nonlinear model predictive control for PMSM drives," in *Proc. 15th Eur. Conf. Power Electron. Appl. (EPE)*, 2013, pp. 1–10.
[18] X. Lin-Shi, F. Morel, A. M. Llor, B. Allard, and J. M. Retif, "Implementation of hybrid control for motor drives," *IEEE Trans. Ind. Electron.*, vol. 54, pp. 1946–1952, 2007.
[19] T. Vyncke, S. Thielemans, and J. Melkebeek, "Finite-set model based predictive control for flying capacitor converters: Cost function design and efficient FPGA implementation," *IEEE Trans. Ind. Informat.*, vol. 9, pp. 1113–1121, 2013.
[20] P. M. Sanchez, O. Machado, E. Bueno, F. J. Rodriguez, and F. J. Meca, "FPGA-based implementation of a predictive current controller for power converters," *IEEE Trans. Ind. Informat.*, vol. 9, pp. 1312–1321, 2013.
[21] E. Monmasson *et al.*, "FPGAs in industrial control applications," *IEEE Trans. Ind. Informat.*, vol. 7, pp. 224–243, 2011.
[22] Z. Ma, J. Gao, and R. Kennel, "FPGA implementation of a hybrid sensorless control of SMPMSM in the whole speed range," *IEEE Trans. Ind. Informat.*, vol. 9, pp. 1253–1261, 2013.
[23] E. Monmasson, L. Idkhajine, and M. W. Naouar, "FPGA-based controllers," *IEEE Ind. Electron. Mag.*, vol. 5, pp. 14–26, 2011.
[24] J. Rodriguez *et al.*, "State of the art of finite control set model predictive control in power electronics," *IEEE Trans. Ind. Informat.*, vol. 9, pp. 1003–1016, 2013.

Zhixun Ma (S'13) was born in Jiangsu, China. He received the B.S. and M.Sc. degrees in electrical engineering in 2006 and 2009, respectively, from the China University of Mining and Technology, Xuzhou, China. He received the Dr.-Ing. (Ph.D.) degree in electrical engineering from the Institute for Electric Drive Systems and Power Electronics, Technical University of Munich, Munich, Germany, in 2014.

He is currently with the Institute for Electric Drive Systems and Power Electronics, Technical University of Munich. His research interests include control of permanent magnet synchronous machines and switched reluctance machines, predictive control and sensorless control of electrical drives, and field programmable gate array (FPGA)-based digital control of power electronics and drive systems.

Saeid Saeidi (S'13) was born in Iran, in 1981. He received the B.Sc. degree in electrical engineering from Bu-Ali Sina University, Hamedan, Iran, and received the M.Sc. degree in electrical engineering from Tomsk Polytechnic University, Tomsk, Russia, in 2009. He received the Ph.D. degree in electrical engineering from the Technical University of Munich, Munich, Germany, in 2013.

From 2009 to 2013, he worked toward the Ph.D. degree at the Technical University of Munich, Munich, Germany.

In 2014, he joined the E-Drive Group, dSPACE GmbH, Paderborn, Germany, where he is responsible for developing field programmable gate array (FPGA)-based model and control algorithms for electrical drives and hardware emulators. His research interests include nonlinear model predictive control and FPGA-based hardware-software codesign.

Ralph Kennel (M'89–SM'96) received the Dr.-Ing. (Ph.D.) degree in electrical engineering from the University of Kaiserslautern, Kaiserslautern, Germany, in 1984.

From 1983 to 1999, he was holding several positions with Robert Bosch GmbH, Erbach, Germany. Until 1997, he was engaged in the development of servo drives. Between 1997 and 1999, he was responsible for advanced and product development of fractional horsepower motors in automotive applications. From 1994 to 1999, he was appointed as a Visiting Professor at the University of Newcastle upon Tyne, Newcastle upon Tyne, U.K. Since 1999, he has been a Professor of Electrical Machines and Drives with Wuppertal University, Wuppertal, Germany. Since 2008, he is a Professor of Electrical Drive Systems and Power Electronics with the Technische Universität München, Munich, Germany. His research interests include sensorless control of ac drives, predictive control of power electronics, and hardware-in-the-loop systems.

Prof. Kennel is a Fellow of the Institution of Electrical Engineers and a Chartered Engineer in the U.K.