PMSM Sliding Mode FPGA-Based Control for Torque Ripple Reduction

Karel Jezernik, *Senior Member, IEEE*, Jože Korelič, and Robert Horvat

***Abstract*—This paper presents a torque ripple reduction approach to the direct torque control of a permanent magnet synchronous motor, using a sliding mode control technique. A distinctive feature of this approach is that, by appropriately parameterizing and implementing the sliding mode controller, the discontinuous nature of the voltage source inverter may be directly incorporated into the design process. The key idea is to incorporate the benefits of the variable structure systems control design and the event-driven sequential control structures in order to raise the system's performance and control efficiency. A predictive sliding-mode controller has been developed, designed as finite-state automata, and implemented using a field-programmable gate array (FPGA). This new FPGA logic regarding torque and speed control has been developed, analyzed, and experimentally verified.**

***Index Terms*—AC motor drives, field-programmable gate array (FPGA), predictive control, torque control, variable structure systems (VSS).**

I. Introduction

THE vector control of electrical drives using two main commercially implemented approaches, field-oriented control (FOC) and direct torque control (DTC) [1], has generated much discussion when comparing the advantages and disadvantages of each scheme [2], [3]. FOC appears to have a better performance than DTC over a wider range of speed and load conditions [4]. The performance of FOC implementation critically depends on very accurately coordinated transformations, and flux angle estimation. This complexity leads to significantly more computations that those of the simpler alternative, the equivalent DTC implementation that normally only requires the setting up of a look-up table that specifies the actuations for each given torque and flux condition.

Conventional DTC has several drawbacks: it exhibits large torque, flux, and current ripple, produces annoying acoustical noise, operates with a nonzero steady-state torque error, has difficulties in controlling the flux at low speeds, and the switching frequency is variable and lower than the sampling frequency [5]–[8].

Four classes of modified DTC schemes that deal with these problems and attempts to improve DTC behavior have evolved:

1) schemes that use improved comparators and switching tables, while the original topology is unchanged [9];
2) solutions that implement the DTC concepts by means of space vector modulation (SVM) [10]–[12];
3) DTC solutions based on fuzzy logic and/or neural networks' schemes [13], [14];
4) torque and flux control systems that explicitly use the variable structure systems (VSS) control approach [15]–[17].

The VSS is an efficient, high-frequency switching control strategy for nonlinear systems with uncertainties. It features good robustness in the face of parameter uncertainties and other disturbances and has fast-response control, but the controlled quantities exhibit chattering [18]. Since the voltage source inverter (VSI) is, by its nature, a switching device, it is natural to regard the drive control from a VSS perspective. In fact, the classic DTC is a VSS system, remarkably well designed to take advantage of VSI's discrete nature.

This paper proposes a predictive sliding mode-control technique which, besides the sign of control error $\boldsymbol{e}$, such as in classical DTC, takes into account the sign of control error derivative $\dot{\boldsymbol{e}}$. The predictive VSS control under consideration is a permanent magnet synchronous motor (PMSM) combined with a two-level three-phase VSI. Based on the use of a simple state-table-based model including the motor and the inverter, the proposed control algorithm calculates the directional evaluation of the current vector for all the possible 2^3 switching states of the inverter [19], [20].

The field-programmable gate array (FPGA) implementation of the proposed predictive VSS control algorithm is performed using the finite-state machine (FSM) approach. The controller synthesis procedure also yields an explicit characterization of those stability regions associated with each mode. Finally, it derives a set of switching rules that orchestrates the transition between the constituent modes and their respective controllers. This is done in such a way that ensures asymptotic stability and reference-input tracking within the overall constrained switched closed-loop system. The proposed methodology was successfully applied to a novel means of design, effective robust torque, and speed control, for a PMSM motor within an FSM software environment.

II. FSM Control Design

A. Basic Controller Design

This section presents the basic controller VSS strategy. Consider the error model as given by

$$\dot{\boldsymbol{e}} = \mathrm{f}(\boldsymbol{x}, t) - \dot{\boldsymbol{x}}^d + \mathrm{b}(\boldsymbol{x}, t)\boldsymbol{u}. \tag{1}$$

Manuscript received June 20, 2012; revised August 1, 2012 and September 10, 2012; accepted September 20, 2012. Date of current version December 24, 2012. This paper was presented at the EPE-PEMC 2012, Sep. 4–6, 2012, Novi Sad, Serbia. Recommended for publication by Associate Editor A. Muetze.

The authors are with the Faculty of Electrical Engineering and Computer Science, University of Maribor, SI-2000 Maribor, Slovenia (e-mail: karel.jezernik@uni-mb.si; joze.korelic@uni-mb.si; robert.horvat@uni-mb.si).

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/TPEL.2012.2222675

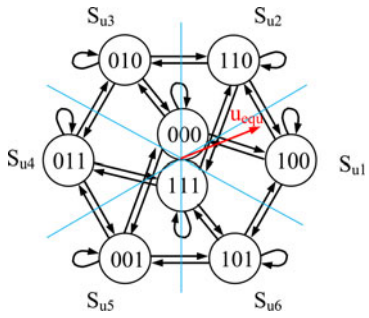


Fig. 1. Stator voltage $\boldsymbol{u}_s(\boldsymbol{V}_k)$ of a three-phase inverter.

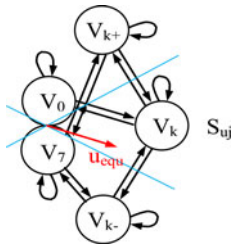


Fig. 2. Stator voltage $\boldsymbol{u}_s(\boldsymbol{V}_k)$ in S_{uj} voltage sector, $j = 1, \ldots, 6$.

Here, $\dot{\boldsymbol{e}}$ is the derivative of control error, $\boldsymbol{x}$ is the state vector, $\boldsymbol{u}$ is the control input, $\mathrm{f}(\boldsymbol{x}, t)$ and $\mathrm{b}(\boldsymbol{x}, t)$ are generally nonlinear functions of time and states, and $\dot{\boldsymbol{x}}^d$ is the derivative of reference state vector.

This method proceeds to solve the feedback control $\boldsymbol{u}$ from (1). The feedback controller is referred to as the "equivalent control"

$$\boldsymbol{u}_{\mathrm{equ}} = -\mathrm{b}^{-1}(\boldsymbol{x}, t)\left(\mathrm{f}(\boldsymbol{x}, t) - \dot{\boldsymbol{x}}^d\right). \tag{2}$$

The controller, as obtained, represents a virtual feedback control action that, in the absence of perturbations and modeling errors, ideally maintains the system's responses by enrolling them in a manifold represented by the condition $\dot{\boldsymbol{e}} = 0$.

Nevertheless, for the class of systems studied here, the controller is allowed to take values within a discrete set $\boldsymbol{V} = \{\boldsymbol{V}_0, \boldsymbol{V}_1, \ldots, \boldsymbol{V}_7\}$. So efforts will be made in order to obtain a discontinuous feedback control solution of the tracking problem proposed on the original system.

Let us propose the Lyapunov function $L = \boldsymbol{e}^2/2$. Its time derivative along the trajectories of (1) can be expressed in terms of $\boldsymbol{u}_{\mathrm{equ}}$ as

$$\dot{L} = \boldsymbol{e}^T \dot{\boldsymbol{e}} = \boldsymbol{e}^T \mathrm{b}(\boldsymbol{x}, t)\left(\boldsymbol{u} - \boldsymbol{u}_{\mathrm{equ}}\right). \tag{3}$$

So now the problem is translated into the selection of a vector $\boldsymbol{u} \in \boldsymbol{V}$ in such a way that $\dot{L}$ is definitely render negative. However, the controller outputs $\boldsymbol{S}_k$, which can be applied after a given input, are limited for each equivalent voltage sector (see Fig. 1) by a switch state approach, as shown in Fig. 2.

Such a graph has several benefits. Firstly, the switching frequency is limited since only one switch state change is permitted per switching period. The application of two opposite voltage vectors is avoided leading to a lower number of predictions to be computed, limited to six on three active voltages and two zero vectors.

Then the control law can be expressed as follows:

1) IF sign $(\boldsymbol{e}(t)\ \dot{\boldsymbol{e}}(t)) < 0$ THEN the control action leads to convergence and should be maintained;
2) IF sign $(\boldsymbol{e}(t)\ \dot{\boldsymbol{e}}(t)) > 0$ THEN the control action leads to divergence and should be altered.

Once the state vector $[\boldsymbol{e}(t),\ \dot{\boldsymbol{e}}(t)]^T$ is formed within the semiplane $\sigma(\boldsymbol{x}, t) = \boldsymbol{e}(t)\ \dot{\boldsymbol{e}}(t) < 0$, it gradually approaches the null vector $[\boldsymbol{0},\ \boldsymbol{0}]^T$. Thus the goal is to discover a control law $\boldsymbol{u}$ that will be able to keep the state vector in the semiplane $\sigma(\boldsymbol{x}, t) = \boldsymbol{e}(t)\ \dot{\boldsymbol{e}}(t, \boldsymbol{u}) < 0$.

B. FSM Design Basis

FSM, also known as finite-state automata (FSA), at their simplest, are models of a system's behaviors, with a limited number of defined conditions or modes, where mode transitions change according to circumstances. FSM is typically used as a type of control system, where knowledge is represented by the states, and actions are constrained by rules. An FSM consists of four main elements: 1) states that define behavior and may produce actions; 2) state transitions that are movements from one state to another; 3) rules or conditions that must be met to allow a state transition; and 4) trigger events that are either externally or internally generated, and may possibly trigger state transitions.

The semantics of event-condition-action (ECA) rules are: when an event E occurs, evaluate condition C, and if the condition is satisfied, then execute action A. The applicability of the ECA rules paradigm has been extensively explored within the context of database systems, and has recently triggered ECA-related activities within other research communities, like real-time systems, workflow, and cooperative problem solving. An occurrence of the event (E) triggers the rule that will start a query for checking the condition (C), which determines whether the system is in a particulate state. The action (A) will fire if the conditions are satisfied. ECA rules allow functionalities to be errors whenever the changes are made, thus enhancing the performance of the system's diagnostic properties.

III. Machine Dynamics and Problem Statement

A. PMSM Model

The model of the control object, the nonsalient PMSM within the stationary stator frame, is expressed by complex notation as follows:

$$\boldsymbol{u}_s = R_s\, \boldsymbol{i}_s + d\boldsymbol{\Psi}_s/dt \tag{4}$$

$$\boldsymbol{\Psi}_s = \boldsymbol{\Psi}_r + L_s\, \boldsymbol{i}_s \tag{5}$$

$$\boldsymbol{\Psi}_r = \Psi_{\mathrm{PM}}\, e^{j\Theta} \tag{6}$$

$$J\, d\omega/dt = T_e - T_l \tag{7}$$

$$d\Theta/dt = p\,\omega \tag{8}$$

$$T_e = (3p/2)(\boldsymbol{\Psi}_r \otimes \boldsymbol{i}_s) = (3p/2)(\Psi_{ra} i_{sb} - \Psi_{rb} i_{sa}). \tag{9}$$

Complex variables $\boldsymbol{u}_s = u_{sa} + j u_{sb}, \boldsymbol{i}_s = i_{sa} + j i_{sb}, \boldsymbol{\Psi}_s = \Psi_{sa} + j\Psi_{sb}, \boldsymbol{\Psi}_r = \Psi_{ra} + j\Psi_{rb}$ are stator voltage, stator current, stator flux, and rotor flux, respectively, Ψ_{PM} is the rotor permanent magnet flux linkage, ω is the mechanical rotor-angle

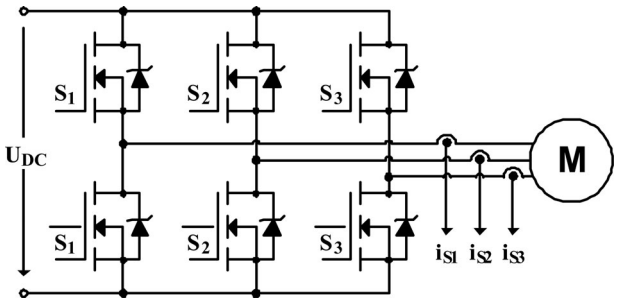


Fig. 3. Basic circuit of a voltage source inverter.

speed, T_e is the motor torque, T_l is the load torque, J is the inertia of the rotor with load, Θ is the electrical angle and p is the number of pole pairs. Motor torque T_e is expressed as a product between the rotor flux and the stator current.

B. Functional Description of VSI

The energy flow between the input and output sides of the three-phase inverter is controlled by a switching matrix. By introducing the binary variables S_i that are "1" if a particular switch S_i is HIGH, and "0" if switch S_i is LOW ($i = 1, 2, 3$), the set $\boldsymbol{S}(S_1, S_2, S_3)$ defines eight possible voltage vectors $\boldsymbol{V}_k$ as $V_0 = [0,0,0]$, $V_1 = [1,0,0]$, $V_2 = [1,1,0]$, $V_3 = [0,1,0]$, $V_4 = [0,1,1]$, $V_5 = [0,0,1]$, $V_6 = [1,0,1]$, and $V_7 = [1,1,1]$. The inverter circuit, depicted in Fig. 3, shows that this particular switching matrix is able to generate three independent control actions, denoted as the components S_1, S_2, and S_3 of the control vector

$$\begin{aligned}\boldsymbol{u}_s(\boldsymbol{V}_k) &= \frac{2}{3}U_{\mathrm{DC}}[S_1 e^{j0} + S_2 e^{j2\pi/3} + S_3 e^{j4\pi/3}]^T \\ &= [\, u_{sa} \quad u_{sb}\,]. \end{aligned} \tag{10}$$

The components, i.e., switch position of the inverter, are generated by the look-up table of the FPGA controller.

The basic circuit of a VSI feeding a Y-connected three-phase load is presented in Fig. 3, where the load has been modeled by phase resistance R_s, inductance L_s, and induced voltages e_{si}. The voltage equation of the Y-connected three-phase load is

$$u_{si}(\boldsymbol{V}_k) = R_s\, i_{si} + L_s\, (d\, i_{si})/(dt) + e_{si} \qquad i = 1,2,3,\ k = 0,1,\ldots,7. \tag{11}$$

Variables u_{si} and i_{si} are motor phase voltages and currents, respectively. Index i has the meaning of three motors' windings, and index k means the discrete switching states of transients in the VSI. The phase currents satisfy the linear condition

$$i_1 + i_2 + i_3 = 0. \tag{12}$$

The considered control problem is the tracking of a three-phase current reference signal. After the current control error is defined $\Delta \boldsymbol{i}_s = \boldsymbol{i}_s^d - \boldsymbol{i}_s$, (11) should be rewritten in error form

$$L_s\, (d\Delta \boldsymbol{i}_s)/(dt) + R_s\, \Delta \boldsymbol{i}_s = \boldsymbol{u}_s(\boldsymbol{V}_k) - \boldsymbol{e}_s \tag{13}$$

containing all the disturbance (exogenous and endogenous) actions on the system.

The basic principle of the current control is to manipulate the input voltage vectors $\boldsymbol{u}_s(\boldsymbol{V}_k)$ so that the desired current is produced by the inverter. This is achieved by choosing an inverter switch combination that applies the appropriate inverter voltages $\boldsymbol{u}_s(\boldsymbol{V}_k)$ to the PMSM windings (11). The three-phase inverter can produce 2^3 voltage vector combinations, from which two are zero vectors; see Fig. 1 [21].

C. Current-Control Design

The switching control (sometimes referred to as the bang–bang control) is a simple strategy that only switches the actuation between full on/off, based on the observation of the system's states relative to a reduced-order switching surface. The benefits of this controller (simplicity, efficiency, performance) make it well suited for driving electrically powered electronic systems [22].

For a first-order system, switching control implies turning the actuation to full positive if the error is negative, or full negative if the error is positive. In general, switching control performs well, even for nonlinear systems with uncertainty. However, it suffers from chattering as the actuation is continually reversed. Most systems do not support such an operation well, with the exception of electrical systems where vibrations do not fatigue the system. In the presented application, the motor winding inductance will filter out this chattering behavior before it reaches the output variable.

Continual switching will induce some ripple in the inductor current. Using a simplified linear model as a lumped inductor and resistor, the current response is given by (13). By ignoring the stator resistance drop in (4), then the stator voltage expression can be represented as

$$d\,\Delta \boldsymbol{i}_s/dt = (\boldsymbol{u}_s(\boldsymbol{V}_k) - \boldsymbol{e}_s)\,/L_s. \tag{14}$$

For a short-time interval of $\Delta\, t$, the stator current is

$$\Delta \boldsymbol{i}_s = (\boldsymbol{u}_s(\boldsymbol{V}_k) - \boldsymbol{e}_s)\ \Delta t/L_s. \tag{15}$$

Thus, the stator current space vector moves by $\Delta\, \boldsymbol{i}_s$ in the direction of the stator voltage space vector at a speed proportional to the difference between the induced voltage $\boldsymbol{e}_s$ and the realizable control $\boldsymbol{u}_s(\boldsymbol{V}_k)$. By selecting the appropriate stator voltage vector $\boldsymbol{u}_s(\boldsymbol{V}_k)$ step by step, it is then possible to change the stator current in the required direction. Decoupled control of the torque and stator flux is achieved by acting on the stator current vector in the locus. These two components are directly proportional to the components of the stator voltage vector in the same direction. With constant motion of the PMSM rotor flux linkage space vector, and if a forward active voltage vector $\boldsymbol{V}_{1,\ldots,6}$ is applied, it causes speed increase of current $\boldsymbol{i}_s$ and the torque increases. On the other hand, when a zero-voltage vector is used, $\boldsymbol{V}_0, \boldsymbol{V}_7$, the current $\boldsymbol{i}_s$ becomes stationary and torque will decrease. If the time duration of a zero-voltage-space vector is sufficiently long, then the rotor flux linkage space vector exceeds the stator current vector, and the torque will change its direction. Thus, it is possible to change the speed of current vector $\boldsymbol{i}_s$ by changing the ratio between the zero and active voltage vectors of the VSI.

The VSS predictive switching control strategy is based on the fact that only a finite number of possible switching states can be generated by a VSI, and that a model of the system can be used to predict the behaviors of the variables for each switching state. The selection criteria must be defined for the selection of an appropriate switching state to be applied. When considering the control error model, as given by (13), the proposed method proceeds to solve the feedback control input function $\boldsymbol{u}_s(\boldsymbol{V}_k)$.

The predictive sliding mode condition on the semiplane $\sigma(\boldsymbol{x}, t)$ can be calculated by applying the following procedure:

1) for the selected quadratic Lyapunov function as $L = \Delta \boldsymbol{i}_s^T \ \Delta \boldsymbol{i}_s/2$;
2) the predictive control should be calculated, so that the time derivative of the Lyapunov function has the form

$$\dot{L} = \Delta \boldsymbol{i}_s^T \ (d\Delta \boldsymbol{i}_s)/(dt) < 0. \tag{16}$$

The sliding mode feedback control is referred to as the equivalent control

$$\boldsymbol{u}_{\text{equ}} = \boldsymbol{e}_s + R_s \, \boldsymbol{i}_s + L_s (d \, \boldsymbol{i}_s^d)/(dt). \tag{17}$$

The convergence condition $\boldsymbol{e}^T(t) \, \dot{\boldsymbol{e}}(\boldsymbol{u}, t) < 0$ for the motor control with $\boldsymbol{e} = \Delta \, \boldsymbol{i}_s$ can now be expressed with $\boldsymbol{u}_{\text{equ}}$ as follows:

$$\sigma(t) = -\Delta \boldsymbol{i}_s^T \ (\boldsymbol{u}_s - \boldsymbol{u}_{\text{equ}}) / L_s \ < 0. \tag{18}$$

IV. FSM Predictive Sliding Mode Controller

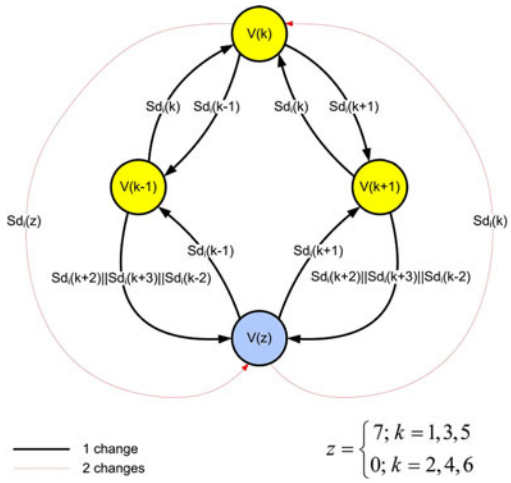
Let us now look at the design of the pulse width modulation (PWM) switching pattern for the three-phase inverter, in more detail [23]. For a particular combination of control errors $[\Delta \, i_{sa}, \ \Delta \, i_{sb}]$ all permissible vectors $\boldsymbol{V}_k$ satisfying the sliding mode's existing conditions should be determined as follows:

$$\text{sign}\,(\boldsymbol{u}_s(\boldsymbol{V}_k) - \boldsymbol{u}_{\text{equ}}) = -\text{sign}\,(\Delta \, \boldsymbol{i}_s)\,; \quad k = 0, 1, \ldots, 7. \tag{19}$$

The control problem can be interpreted as follows: find the sequence and the duration of the ON and OFF states for the inverter switches, so that the given dynamical specifications of the closed-loop control are satisfied. This formulation embraces the control of the closed-loop dynamics and the PWM pattern selection, as one problem. The PWM pattern selection is a result of the sliding mode's existence within a closed-loop control of the inverter [24]. The vector $\boldsymbol{V}_k$ is treated as the output of the control system. This vector changes whenever the control error $\boldsymbol{e}$ exceeds the prescribed value, by taking into account all the dynamical changes and external disturbances enclosed by the equivalent control $\boldsymbol{u}_{\text{equ}}$.

The FPGA implementation of the predictive switching strategy could be accomplished using three independent two-level hysteresis currents' error $\text{sign}\,(\Delta \, \boldsymbol{i}_s)$ comparators. However, this might introduce limited cycles, as only two motor currents are independent. Therefore, the control law (19) can be implemented using FSM-based ECA rules:

1) $\text{sign}\,(\Delta \, \boldsymbol{i}_s)$ current error defines event *Sdi*;
2) $\text{sign}\,(\boldsymbol{u}_{\text{equ}})$ change of equivalent control defines condition (Su_x);
3) $\boldsymbol{V}_k$ voltage vector defines the output action of the PMSM current controller.



(a)

signDi \ secU$_x$		Su1	Su2	Su3	Su4	Su5	Su6
		100	110	010	011	001	101
events	Sdi_0 000	V_z	V_z	V_z	V_z	V_z	V_z
	Sdi_1 100	V_1	V_1	V_z	V_z	V_z	V_1
	Sdi_2 110	V_2	V_2	V_2	V_z	V_z	V_z
	Sdi_3 010	V_z	V_3	V_3	V_3	V_z	V_z
	Sdi_4 011	V_z	V_z	V_4	$V4$	V_4	V_z
	Sdi_5 001	V_z	V_z	V_z	V_5	V_5	V_5
	Sdi_6 101	V_6	V_z	V_z	V_z	V_6	V_6
	Sdi_7 111	V_z	V_z	V_z	V_z	V_z	V_z

$V_Z(k) = \begin{cases} V_0; V(k\text{-}1) = V_0 \vee V_1 \vee V_3 \vee V_5 \\ V_7; V(k\text{-}1) = V_7 \vee V_2 \vee V_4 \vee V_6 \end{cases}$

(b)

Fig. 4. Interface for selecting the voltage vector implemented with (a) FSM and (b) state transition table.

The FSM in Fig. 4(a) uses four main states. Depending on the voltage sector, the states of the FSM have three different active voltage vectors, $\boldsymbol{V}_{k-1}, \boldsymbol{V}_k, \boldsymbol{V}_{k+1}$, and the zero vectors, $\boldsymbol{V}_z$, that can change their value from "000" to "111" Fig. 1. The top state of the FSM represents an active voltage vector, the states in the middle represent the adjacent active voltage vectors, and the bottom state represents a zero-voltage vector inside the voltage sector. Only transitions between adjacent vectors are allowed. Only when transition from $\boldsymbol{V}_0$ to an even vector ($\boldsymbol{V}_2$, $\boldsymbol{V}_4$, $\boldsymbol{V}_6$) or from $\boldsymbol{V}_7$ to an odd vector ($\boldsymbol{V}_1$, $\boldsymbol{V}_3$, $\boldsymbol{V}_5$) is requested, two changes are necessary.

The conditions of the FSM are voltage sectors and the current state. Any transition between switching states can be done when the defined event is recognized. Another method of implementing FSM is to use a state-transition table, as shown in Fig. 4(b). The number of rows corresponds to the number of events, and each row's event contains the actions and next switching states for the defined conditions along the columns. In the table are highlighted the allowed transitions. For the actual sector Su_x and required *Sdi* event only vector $\boldsymbol{V}_k$ or adjacent vectors are activated, in all other cases the zero vector $\boldsymbol{V}_0$ or $\boldsymbol{V}_7$ is activated. This is an efficient way of encoding Boolean logic functions,

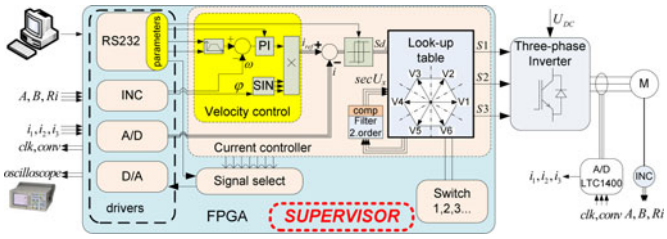


Fig. 5. FPGA controller of a PMSM motor.

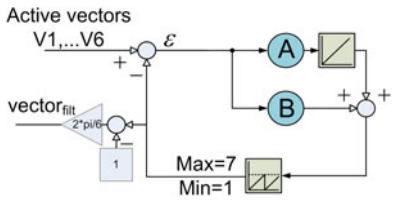


Fig. 6. Second-order PLL filter.

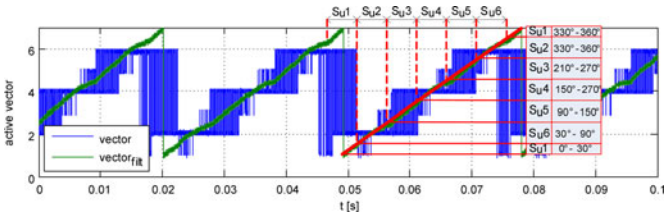


Fig. 7. Active voltage vectors filtering.

and FPGAs are, in fact, the key components of a modern digital controller.

Most transitions at low voltage/low speed are between the adjacent voltage vectors and the zero-voltage vector. Consequently, a zero-voltage vector inside the voltage sector is selected, according to the adjacent active voltage vectors. For example, in voltage sector Su_x, the adjacent active voltage vectors are vectors $\boldsymbol{V}_{k-1}$ and $\boldsymbol{V}_{k+1}$. Direct transition between vectors $\boldsymbol{V}_{k-1}$ and $\boldsymbol{V}_{k+1}$ is not permitted. The application of opposite voltage vectors is avoided, thus leading to lower current ripple. The switching frequency of the PWM inverter is limited since only one switch state change is permitted per conduction interval. This method provides most of the transitions, using only one inverter leg.

The implementation software of the state-transition table has a simple solution for FPGA. Only one constant and two short processes are necessary for implementation. The first process synchronizes the input signals that define the voltage sector S_{ux}. The second process initiates the action of choosing the voltage vector from the table.

V. Implementation

The proposed approach is based on fast parallel processing and is suitable for FPGA implementation. In such an implementation it would be possible to reproduce a near-ideal switching mode process. However, with FPGA implementation, the designer has a difficult task when characterizing and describing the hardware architecture that corresponds to the chosen control algorithm. FPGA designers must follow an efficient design methodology in order to benefit from the advantages of the FPGAs and their powerful CAD tools. From the software point of view, the HDL modeling system is based on using variables that require logical values.

Fig. 5 presents the general structure of the different elementary modules. As shown in Fig. 5, Xilinx Spartan XC3S1200E is used to implement the PMSM motor controller. This motor control, built in our laboratory, is divided into three parts. The first part is the driver's part with ADC and DAC management modules, an incremental module for speed and position measurement, and an RS 232 module for connecting the host PC equipment within the MATLAB/Simulink program environment. The second part includes the PI-speed motor controller. The output of the speed controller is the desired torque value of the PMSM motor that multiplies the three-phase reference currents of the machine. The motor phase currents depend on rotor position. The third part includes a look-up table with a hysteresis current controller, and the average output voltage of the PMSM motor that is used for synchronization–voltage sector selection.

The sign of the stator voltage is determined by the second-order PLL filter, shown in more detail in Fig. 6, that filters the active voltage vectors, $\boldsymbol{V}_1, \ldots, \boldsymbol{V}_6$. The filter's output, $\text{vector}_{\text{filt}}$, represents the position of the stator voltage angle [25]. For more details regarding FPGA controller and hardware setup see [26].

This signal, $\text{vector}_{\text{filt}}$ is used for voltage sector selection, secU_x in the state transition table [see Fig. 4(b)]. The PLL filter's output represents a 360° electric angle of stator voltage (see Fig. 7). The electric angle is divided into six parts, determined voltage sectors, secU_x in Fig. 4(b).

Based on this simplified PLL closed loop, the transfer function is expressed as follows:

$$\frac{U_s}{(U_s(\boldsymbol{V}_k))} \cong \frac{sB + A}{s^2 + sB + A}. \tag{20}$$

This transfer function corresponds to a second-order system. The observation dynamic is then set using A as the square of the filter frequency and $B = 2D\sqrt{A}$, where D is the damping coefficient of the second-order low pass filter (20). The damping coefficient D is set to 1; the criteria for choosing the filter frequency A are dependent on the mechanical time constant of the PMSM.

The FPGA controller includes diagnosis features necessary for drive installation, and tests for problem detection and elimination. The control algorithm execution interval was 2.5 μs, and the switching frequency of the inverter was set by the tolerance band of the hysteresis current control. Control of both the speed and torque was possible; thus, hardware within the loop operation could also be performed.

The FPGA-based PMSM system (see Fig. 5) is meant to be operated according to certain established rules and principles, which are presented as FSM. Fig. 8 shows, on the right-hand side, normal machine operation including start/shut/up/down operations. The left-hand side represents operation if a warning or error occurs.

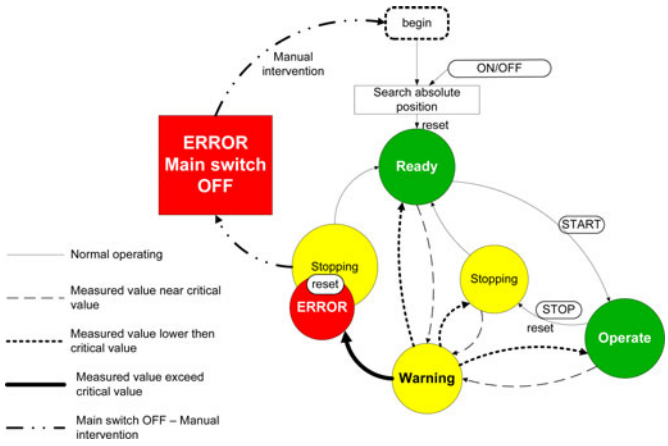


Fig. 8. State CAD diagram.

VHDL code was created within Xilinx ISE software. This software scheme is divided into individual blocks for a better review. It is composed of a data path and a control unit coded in VHDL. The data path is composed of adders, multipliers, multiplexers, and registers. The data transfer between these operators is managed by a control unit, which is synchronized with the clock signal (CLK). The control unit of the developed modules is always activated via a Start pulse signal. When the computation process is finished, an end pulse signal indicates that the data outputs of the module are ready.

Such an operational scheme occupies 35% of the "4 input LUTs FPGA circuit," and 67% of 18-bit multipliers of the XC3S1200E.

VI. Results

In order to test the proposed Xilinx Spartan 3 FPGA-based controller, experiments were carried out on an in-house built FPGA prototyping platform. The PMSM motor EC32 from Maxon was used for all simulations and experimental results. The parameters were $R_s = 0,665\,\Omega$, $L_s = 0.113$ mH, $J = 2 \times 10^{-6}$ kg·m^2, $\Psi_{\rm PM} = 0.0167$ V·s, $t_M = 6.6$ ms, and additional series inductance $L_f = 1$ mH.

The simulation results with the proposed speed and current PMSM control are presented next. The sampling time $T_s = 20$ μs and bandwidth $\pm 1e-16$ for all hysteresis controllers were chosen. The results of the start-up characteristic under change of load and reference speed are shown in Fig. 9—reference and motor speed, reference torque, estimated machine torque, load torque, and stator current errors. Fig. 10 shows a comparison of three control methods. The methods chosen are hysteresis torque control and the proposed control method. The figures are from top: motor speed, hysteresis control, proposed control, and proposed control with additional restrictions. Additional restrictions prohibit state change between opposite vectors in the state transition table. Step change of reference torque from +100% to –100% at low speed is depicted. It can be clearly seen that not only torque ripple is reduced (from 7.8% to 4.4% and 4.1%, respectively) but also the number of state changes is reduced. Additional restrictions in the state transition table have

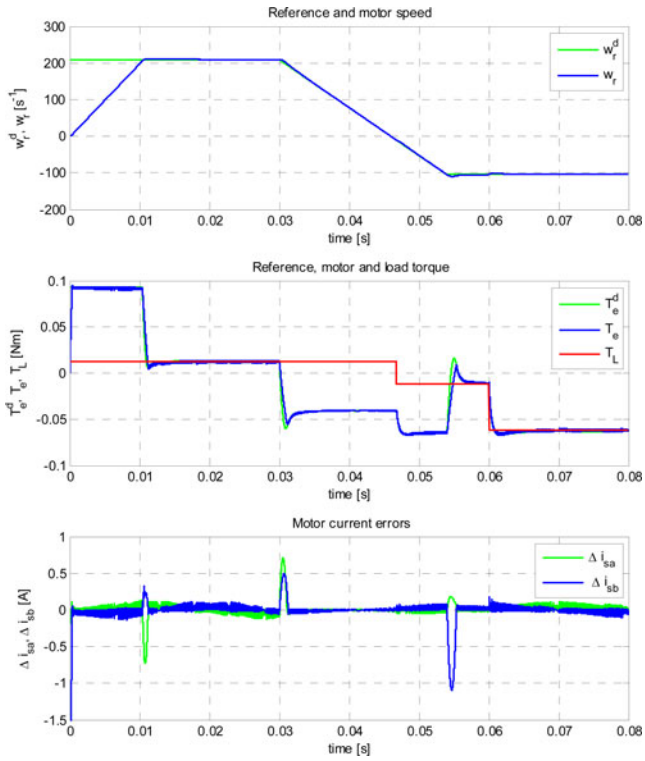


Fig. 9. Motor current errors Δi_{sa}, Δi_{sb}, reference motor torque T_e^d, motor torque T_e, load torque T_L, reference ω_r^d, and actual rotor speed ω_r.

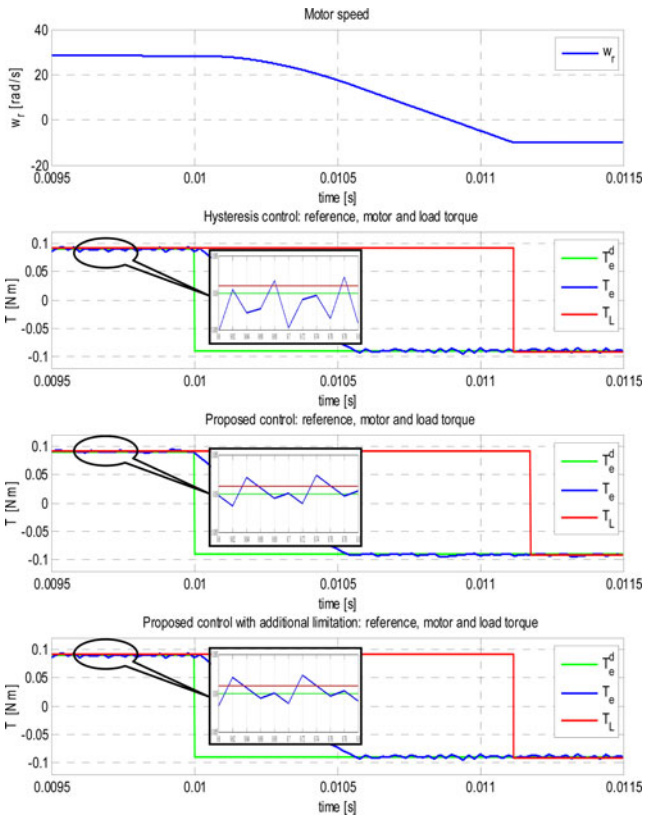


Fig. 10. Comparison of three torque control methods. Step change of reference torque from +100% to –100% at low speed. From top: motor speed, torque response with hysteresis control, torque response with proposed control [see Fig. 4(b)], and torque response with proposed control and additional restrictions [see Fig. 4(a)].

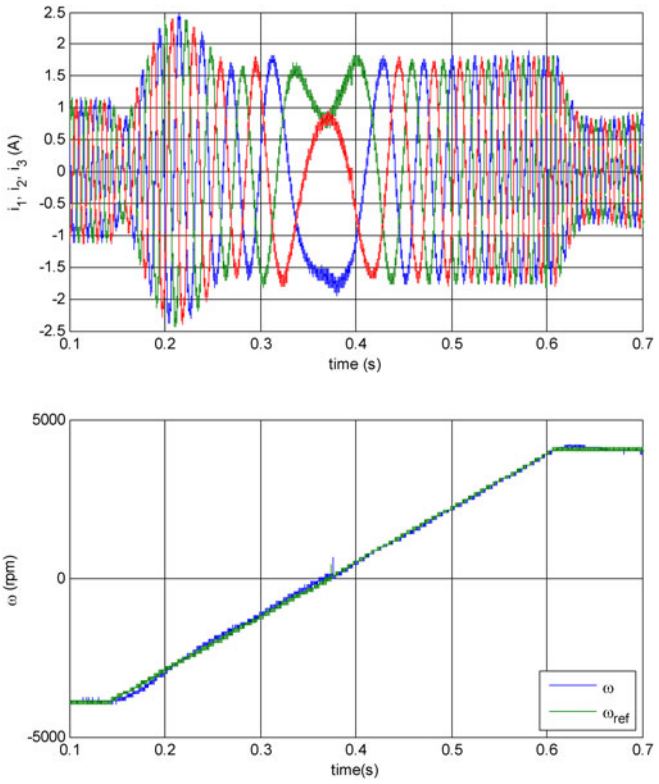


Fig. 11. Three-phase stator currents i_{s1}, i_{s2}, i_{s3}, reference speed $\omega_{\rm ref}$, and actual speed ω.

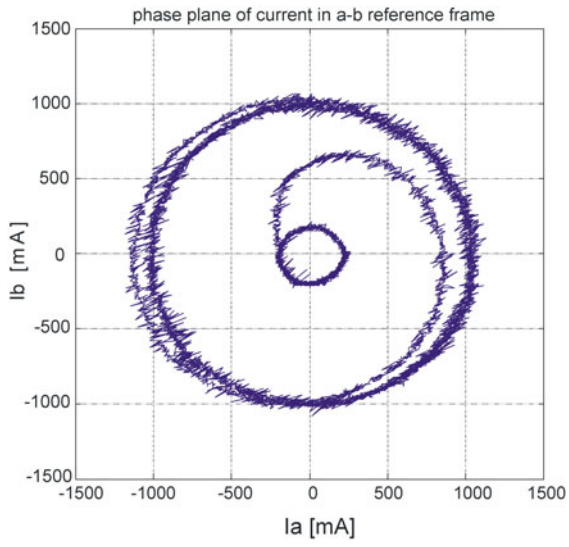


Fig. 12. Current response to step load change of the motor in a stator fixed reference frame.

a benefit to further lower the number of state changes, although the impact is minor and is load and task dependent.

The experimental results are illustrated in Fig. 11 through to Fig. 12. Fig. 11 shows the three-phase currents, and reference ($\omega_{\rm ref}$) and actual speed (ω). Reference speed changed the value from $\omega_{\rm ref} = –4000$ [r/min] to $\omega_{\rm ref} = +4000$ [r/min].

A current response to load step change is illustrated in Fig. 12. The results were recorded using a digital storage oscilloscope and plotted within MATLAB.

VII. Conclusion

The predictive VSS control presented here is a general approach from any system constituted of a continuous process and a finite number of discrete control states. In the context of ac machine control by means of an inverter, the VSS control realizes torque/current regulation by dealing directly with the inverter switches. Very good results with the proposed predictive VSS control were obtained for the PMSM drive, and also for asynchronous machines not presented in this paper. Superior transient performance and torque ripple reduction of the proposed control algorithms, such as vector control, were demonstrated.

High-frequency flexible control of power electronics can be achieved using an FSM controller. Its increased operating frequency provides higher power densities for power converters while its software desired control algorithm allows for flexibility when system designing. Adaptive and predictive control techniques can be implemented within the FSM controller, increased system performance, faster load-change response, and more stable operation over traditional DTC motor control can be achieved.

References

[1] D. Casadei, F. Profumo, G. Serra, and A. Tani, "FOC and DTC: Two variable schemes for induction motors torque control," *IEEE Trans. Power Electron.*, vol. 17, no. 5, pp. 779–787, Sep. 2002.

[2] I. Takahashi and T. Noguchi, "A new quick response and high efficiency control strategy of an induction motor," *IEEE Trans. Ind. Appl.*, vol. IA-22, no. 5, pp. 820–827, Sep./Oct. 1986.

[3] T. Geyer, "Computationally efficient model predictive direct torque control," *IEEE Trans. Power Electron.*, vol. 26, no. 10, pp. 2804–2816, Oct. 2011.

[4] M. Depenbrock, "Direct self control of inverter-fed induction machines," *IEEE Trans. Power Electron.*, vol. 3, no. 4, pp. 420–429, Oct. 1988.

[5] G. S. Buja and M. P. Kazmierkowski, "Direct torque control of pwm inverter-fed ac motors—A survey," *IEEE Trans. Ind. Electron.*, vol. 51, no. 4, pp. 744–757, Aug. 2004.

[6] F. Fang, X. Zhou, and G. Liu, "Instantaneous torque control of small inductance brushless dc motor," *IEEE Trans. Power Electron.*, vol. 27, no. 12, pp. 4952–4964, Dec. 2012.

[7] J. A. Restrepo, J. M. Aller, J. C. Viola, A. Bueno, and T. G. Habetler, "Optimum space vector computation technique for direct power control," *IEEE Trans. Power Electron.*, vol. 24, no. 6, pp. 1637–1645, Jun. 2009.

[8] Y. Zhang and J. Zhu, "Direct torque control of permanent magnet synchronous motor with reduced torque ripple and commutation frequency," *IEEE Trans. Power Electron.*, vol. 26, no. 1, pp. 235–248, Jan. 2011.

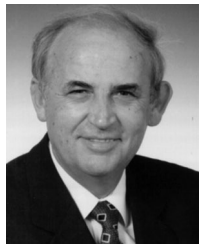
[9] D. Casadei, G. Serra, and A. Tani, "Implementation of a direct torque control algorithm for induction motors based on discrete space vector modulation," *IEEE Trans. Power Electron.*, vol. 15, no. 4, pp. 769–777, Jul. 2000.

[10] B. H. Kenny and R. D. Lorenz, "Stator- and rotor-flux-based deadbeat direct torque control of induction machines," *IEEE Trans. Ind. Appl.*, vol. 39, no. 4, pp. 1093–1101, Jul./Aug. 2003.

[11] C. Lascu, I. Boldea, and F. Blaabjerg, "A modified direct torque control for induction motor sensorless drive," *IEEE Trans. Ind. Appl.*, vol. 36, no. 1, pp. 122–130, Jan./Feb. 2000.

[12] Y. Zhang and J. Zhu, "A novel duty cycle control strategyto reduce both torque and flux ripples for DTC of permanent magnet synchronous motor drives with switching frequency reduction," *IEEE Trans. Power Electron.*, vol. 26, no. 10, pp. 3055–3067, Oct. 2011.

[13] P. Z. Grabowski, M. P. Kazmierkowski, B. K. Bose, and F. Blaabjerg, "A simple direct-torque neuro-fuzzy control of PWM-inverter-fed induction motor drive," *IEEE Trans. Ind. Electron.*, vol. 47, no. 4, pp. 863–870, Aug. 2000.
[14] V. Q. Leu, H. H. Choi, and J. W. Jung, "Fuzzy sliding mode speed controller for PM synchronous motors with a load torque observer," *IEEE Trans. Power Electron.*, vol. 27, no. 3, pp. 1530–1539, Mar. 2012.
[15] Z. Yan, C. Jin, and V. I. Utkin, "Sensorless sliding-mode control of induction motors," *IEEE Trans. Ind. Electron.*, vol. 47, no. 6, pp. 1286–1297, Dec. 2000.
[16] Z. Sorchini and P. T. Krein, "Formal derivation of direct torque control for induction machines," *IEEE Trans. Power Electron.*, vol. 21, no. 5, pp. 1428–1436, Sep. 2006.
[17] G. H. B. Foo and M. F. Rahman, "Direct torque control of an ipm-synchronous motor drive at very low speed using a sliding-mode stator flux observer," *IEEE Trans. Power Electron.*, vol. 25, no. 4, pp. 933–942, Apr. 2010.
[18] V. I. Utkin, *Sliding Modes in Control and Optimization*. Berlin, Germany: Springer-Verlag, 1992.
[19] J. Rodríguez, J. Pontt, C. A. Silva, P. Correa, P. Lezana, P. Cortés, and U. Ammann, "Predictive current control of a voltage source inverter," *IEEE Trans. Ind. Electron.*, vol. 54, no. 1, pp. 495–503, Feb. 2007.
[20] P. Landsmann and R. Kennel, "Saliency-based sensorless predictive torque control with reduced torque ripple," *IEEE Trans. Power Electron.*, vol. 27, no. 10, pp. 4311–4320, Oct. 2012.
[21] A. Sathyan, N. Milivojevic, Y.-J. Lee, M. Krishnamurthy, and A. Emadi, "An FPGA-based novel digital PWM control scheme for BLDC motor drives," *IEEE Trans. Ind. Electron.*, vol. 56, no. 8, pp. 3040–3049, Aug. 2009.
[22] N. Oikonomou and J. Holtz, "Closed-loop control of medium-voltage drives operated with synchronous optimal pulsewidth modulation," *IEEE Trans. Ind. Appl.*, vol. 44, no. 1, pp. 115–123, Jan./Feb. 2008.
[23] 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, no. 6, pp. 1925–1936, Jun. 2009.
[24] M. Preindl, E. Schaltz, and P. Thogersen, "Switching frequency reduction using model predictive direct current control for high power voltage source inverters," *IEEE Trans. Ind. Electron.*, vol. 58, no. 7, pp. 2826–2835, Jul. 2011.
[25] C. H. da Silva, R. R. Pereira, L. E. B. da Silva, G. Lambert-Torres, B. K. Bose, and S. U. Ahn, "A digital PLL scheme for three-phase system using modified synchronous reference frame," *IEEE Trans. Ind. Electron.*, vol. 57, no. 11, pp. 3814–3821, Nov. 2010.
[26] R. Horvat and K. Jezernik, "Implementation of discrete event control for brushless ac motor," *IET Power Electron.*, vol. 4, no. 7, pp. 767–775, 2011.



Karel Jezernik (M'77–SM'04) received the B.Sc., M.Sc., and Dr. Eng. degrees from the University of Ljubljana, Ljubljana, Slovenia, in 1968, 1974, and 1976, respectively, all in electrical engineering.

In 1972, he joined the University of Maribor, Maribor, Slovenia, and in 1985 he became a Full Professor and Head of the Institute of Robotics (till 2010). His research and teaching interests include automatic control, robotics, power electronics, and electrical drives. Current projects in these areas are high-precision tracking control in machine tools, DD robots, and robust torque control in EVs. He consults on industrial servo control systems and other control applications.

Dr. Jezernik is a member of the Electrotechnical Association of Slovenia and the Automation and Robotization Society of Slovenia.

Jože Korelič received the B.Sc. and M.Sc. degrees in electrical engineering from the University of Maribor, Maribor, Slovenia, in 1979 and 2010, respectively, where he is currently working toward the Ph.D. degree in electrical engineering.

Since 1980, he has been with the Institute of Robotics, University of Maribor, where he currently serves as a Researcher. His research interests include control of electrical drives, power electronics, and robotics.

Robert Horvat received the B.Sc. degree from the Faculty of Electrical Engineering and Computer Science, University of Maribor, Maribor, Slovenia, in 2007, where he is currently working toward the Ph.D. degree in electrical engineering.

His research interests include control of electrical drives, driving, and measurement interfaces in programmable logic devices.