MATLAB Function Based Approach to FOC of PMSM Drive

Omer Cihan Kivanc and Salih Baris Ozturk

Department of Electrical and Electronics Engineering
Okan University
34959 Tuzla / Istanbul, TURKEY
E-mail: baris.ozturk@okan.edu.tr

***Abstract* — In this study, modeling and simulation of a speed sensor driven field-oriented control (FOC) of a permanent magnet synchronous motor (PMSM) drive is developed by using MATLAB Function blocks in MATLAB/Simulink. This method allows easier algorithm and software development stages for experimental studies compared to the classical block diagram approach. The superiority of the method over commonly used "Code Generation" tools is also emphasized. First, a MATLAB/Simulink model of the FOC of PMSM drive is developed by using MATLAB programming in MATLAB Functions similar to C coding techniques. The results of the simulation are presented. Then, the MATLAB programming based codes developed in simulation are implemented in a TMS320F28335 floating-point MCU by using C programming language and the experimental results are obtained. Finally, the results of the simulation and experiments are compared.**

Keywords — Permanent magnet synchronous motor; PMSM; AC motor drive; field-oriented control; FOC; speed control; modeling; simulation; Embedded Coder; SimPowerSystems; Digital Motor Control; DMC

I. INTRODUCTION

Recently, AC motor control and driving strategies are attracting more and more interest. Development of embedded systems, observers and control systems are enabling new algorithms in motor control. The complex nature of these algorithms cause difficulties in programming. MATLAB/Simulink® is commonly used for modeling and simulation of electromechanical systems and their control applications before the realization step. It is usually expected that experiments yield results similar to those obtained in the MATLAB/Simulink simulation environment.

Usually, a MATLAB/Simulink simulation model of an AC motor drive is developed in the literature by using classical Simulink® blocks that are available in the standard Simulink Library [1]–[12]. Although, initial developments of the algorithms designed by connecting the library blocks in Simulink Library makes the development stage easy, the future addition or modifications of the system become quite difficult. Moreover, the development of AC motor drive simulation models can also be achieved by using the motor control blocks in MATLAB/Simulink Embedded Coder® [13]–[15]. The MathWorks®, Inc. developed the Embedded Coder toolbox for certain microprocessor families such as the Texas Instruments™ (TI) C2000 microprocessor family which is widely used in motion control applications. The C2000 family offers superior performance in motor control applications. If the experimental motor control system consists of a C2000 microcontroller family, it is an option to build the simulation model of an AC motor drive system by using the blocks provided in Embedded Coder. However, this toolbox is an additional cost to the standard MATLAB/Simulink package and does not allow modifications in its blocks.

The simpler, easily modifiable and more economical choice of MATLAB/Simulink modeling and simulation of an AC motor drive is to use MATLAB Functions. The handicaps given above are overcome by using MATLAB Function blocks in MATLAB/Simulink. Most of the motor control applications nowadays require the C programming language. Therefore using C programming like MATLAB Programming in MATLAB Functions in MATLAB/Simulink produces a similar approach as opposed to the classical block diagram based modeling of an actual system. Using the classical block diagram approach also creates difficulties in understanding and improving the system during future modifications.

In this paper, a proposed MATLAB/Simulink model of a speed sensored field-oriented control (FOC) of a PMSM drive is developed by using MATLAB programming in MATLAB Functions similar to the C programming language. Therefore, the goal of a simple, easily modifiable and economical MATLAB/Simulink modeling method that helps the smooth transition to the experimental stage is achieved.

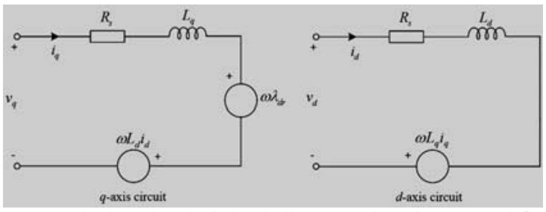


Figure 1. Equivalent electrical circuit diagrams of quadrate q– and d–axes synchronous reference frame that apply to both surface-mount and interior permanent magnet synchronous machines (IPMSM) [16], [17].

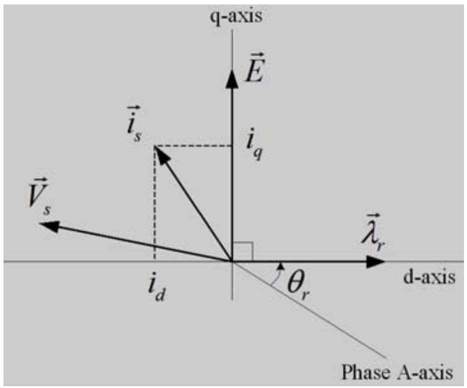


Figure 2. Basic phasor diagram for PMSM [16].

This is especially true for systems that are programmed in the C language. During the application stage, MATLAB

DOI 10.5013/IJSSST.a.17.33.02 ISSN: 1473-804x online, 1473-8031 print

function based drive modules are manually converted to the C language and implemented in a TI's TMS320F28335 Delfino floating-point MCU. Finally, the results of the simulation and experiments are compared.

II. FOC of PMSM Drive

A. Dynamic Mathematical Model of Interior PMSM

The dq model in the rotating synchronous reference frame shown in Fig. 1 is used to analyze the IPMSM for the field-oriented control (FOC).

The stator voltage equations of the IPMSM in the rotating dq reference frame are given by (1) and (2), omitting the influences of magnetic field saturation and magnetic hysteresis as

$$v_q = i_q R_s + L_q \frac{di_q}{dt} + (\omega L_d i_d + \omega \lambda_{dr}) \tag{1}$$

$$v_q = i_d R_s + L_d \frac{di_d}{dt} - \omega L_q i_q \tag{2}$$

where v_d, v_q, i_d, i_q are the stator d– and q–axes voltages and currents in the rotor reference frame, respectively; R_s is the stator winding resistance; L_d and L_q denote the d– and q–axes inductance, respectively; ω is the rotor angular electrical velocity; and λ_{dr} is the flux linkages due to the permanent magnet rotor flux [16]–[18].

B. Vector Control of PMSM

For superior driving performance of PMSM, vector control, or field-oriented control (FOC), is widely used. As shown in Fig. 2, for simplicity and to remove the time-varying quantities, the two-axis theory is used for FOC of sinewave drives. To produce the maximum torque in FOC, q–axis current and the rotor flux linkage aligned with the d–axis are kept in quadrature form, as depicted in Fig. 2. The d– and q–axes current phasor components, i_d and i_q, are generally fixed to the rotor reference frame. To obtain i_d and i_q from the instantaneous phase currents, a reference frame transformation, such as Park transformation is employed [16].

For switching of the inverter, the space vector pulse with modulation (SVPWM) is used in PMSM drives. The space vector form of the stator voltage equation in the stationary reference frame is shown in (3)

$$\vec{v_s} = R_s \vec{i_s} + \frac{d\vec{\lambda_s}}{dt} \tag{3}$$

where R_s, $\vec{v_s}$, $\vec{i_s}$, and $\vec{\lambda_s}$ are the resistance of the stator winding, complex space vectors of the three phase stator voltages, currents, and flux linkages, respectively. These vectors are stated in the stationary reference frame fixed to the stator. The resultant stator reference frame voltage, current, and flux linkage space vectors are shown in (4), (5) and (6), respectively

$$\vec{v_s} = \frac{2}{3}[v_{sa}(t) + a v_{sb}(t) + a^2 v_{sc}(t)] \tag{4}$$

$$\vec{i_s} = \frac{2}{3}[i_{sa}(t) + a i_{sb}(t) + a^2 i_{sc}(t)] \tag{5}$$

$$\vec{\lambda_s} = \frac{2}{3}[\lambda_{sa}(t) + a\lambda_{sb}(t) + a^2 \lambda_{sc}(t)] \tag{6}$$

where $a = e^{j2\pi/3}$, and $a^2 = e^{j4\pi/3}$ are spatial operators for orientation of the stator windings; $v_{sa}(t)$, $v_{sb}(t)$, and $v_{sc}(t)$ are the instantaneous values of stator phase voltages; $i_{sa}(t)$, $i_{sb}(t)$, and $i_{sc}(t)$ are the instantaneous phase currents; $\lambda_{sa}(t)$, $\lambda_{sb}(t)$, and $\lambda_{sc}(t)$ are the stator flux linkages. They are calculated by multiplying instantaneous phase values by the stator winding orientations. The a–phase is chosen as the stator reference axis in the direction of maximum MMF. The b– and c–axes stator reference frames are selected 120° and 240° (electrical degree) ahead of the a–axis, respectively.

When the three-phase windings are fed by an inverter, as shown in Fig. 3, the primary voltages $v_{sa}(t)$, $v_{sb}(t)$, and $v_{sc}(t)$ are determined by the status of the three switches, S_a, S_b, and S_c.

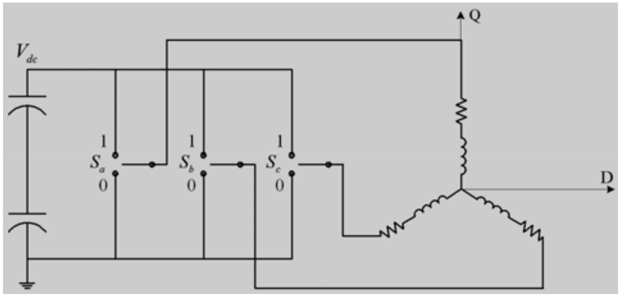


Figure 3. Voltage source inverter (VSI) connected to the R-L load [21].

If the switch is at state 0 that means the phase is connected to the negative and if it is at 1 it means that the phase is connected to the positive leg. The eight basic voltage space vectors defined by the combination of inverter switches are illustrated in Fig. 3 [19].

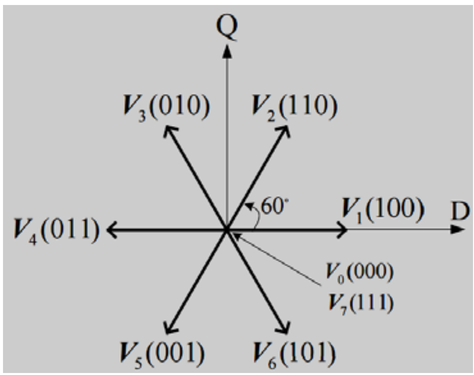


Figure 4. Eight possible voltage space vectors obtained from VSI [22].

For example, phase–a is connected to V_{dc} if S_a is one, otherwise phase–a is connected to zero. This is similar for b–axis and c–axis. There are six nonzero voltage vectors: $V_1(100)$, $V_2(110)$, …, $V_6(101)$ and two zero voltage vectors: $V_7(000)$ and $V_0(111)$. The six nonzero voltage vectors are 60° apart from each other as in Fig. 4 [19].

The stator voltage space vector (expressed in the stationary reference frame) representing the eight voltage vectors can be shown as

$$v_s(S_a, S_b, S_c) = \frac{2}{3} V_{dc}[S_a + aS_b + a^2 S_c] \tag{7}$$

where V_{dc} is the dc-link voltage, S_a, S_b, and S_c are the switching states, and the coefficient of 2/3 is the coefficient comes from the Park transformation [19].

C. Components of FOC of PMSM Drive

The major components of the FOC of PMSM drive is shown in Fig. 5. The system consists of speed, d– and q–axes current PI regulators, Park and Clarke transformations, inverse Park transformation, space vector generation, speed calculation, current and encoder signal conditionings and PWM generator modules. In Fig. 5, an incoming speed command profile goes into a speed PI regulator which outputs q–axis reference current. The d–axis current reference is set to zero for surface-mount PMSM. These current references and their corresponding feedbacks are DC quantities for the PI regulators to track easily. The outputs of the current PI regulators generate stator dq–axes voltage references which are also in DC quantities. To apply sinewave currents to the motor, these DC voltage quantities are then transformed into the instantaneous sinusoidal voltage commands for the individual stator phases using the rotor angle feedback and the inverse reference frame transformation matrix (inverse Park transformation).

The space vector PWM generator converts the stationary reference frame voltage references into abc frame based duty cycle equivalences. These three-phase duty cycles (T_a, T_b, and T_c) are then brought into a PWM generator for the inverter to generate the appropriate three-phase pulsed-voltages that are applied to the motor. To obtain the dq–axes current feedbacks in DC quantities, first a– and b–axes AC currents are transformed into stationary values using Clarke transformation and then the stationary current values acquired from Clarke transformation are set up as inputs to the Park transformation along with the rotor position feedback signal to generate the equivalent DC feedback quantities in dq reference frame.

In PMSM, the rotor windings are already along the d– and q–axes, only the stator windings quantities need transformation from three-phase quantities to the two-phase dq rotor rotating reference frame quantities. Therefore, Park transformation is used to transform the stator quantities of a PMSM onto a dq reference frame that is fixed to the rotor, with the positive d–axis aligned with the magnetic axis of the rotor which has a permanent magnet in PMSM. The dq transformation matrix (Park transformation) used for currents and the inverse Park transformation used for voltages are given respectively by

$$\begin{bmatrix} i_d \\ i_q \end{bmatrix} = \begin{bmatrix} \cos\theta & \sin\theta \\ -\sin\theta & \cos\theta \end{bmatrix} \begin{bmatrix} i_\alpha \\ i_\beta \end{bmatrix} \tag{8}$$

$$\begin{bmatrix} v_\alpha \\ v_\beta \end{bmatrix} = \begin{bmatrix} \cos\theta & \sin\theta \\ -\sin\theta & \cos\theta \end{bmatrix} \begin{bmatrix} v_d \\ v_q \end{bmatrix}. \tag{9}$$

It is possible to separate the motor complex space vectors into stationary real and imaginary parts with the Clarke transformation. By using the Clarke transformation in FOC of AC motor drive, stator currents are transformed from three-phase to two-phase quadrature equivalent values as inputs to the Park transformation. The Clarke transformation matrix for a balanced three-phase system is defined as

$$\begin{bmatrix} i_\alpha \\ i_\beta \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ \frac{1}{\sqrt{3}} & \frac{2}{\sqrt{3}} \end{bmatrix} \begin{bmatrix} i_a \\ i_b \end{bmatrix}. \tag{10}$$

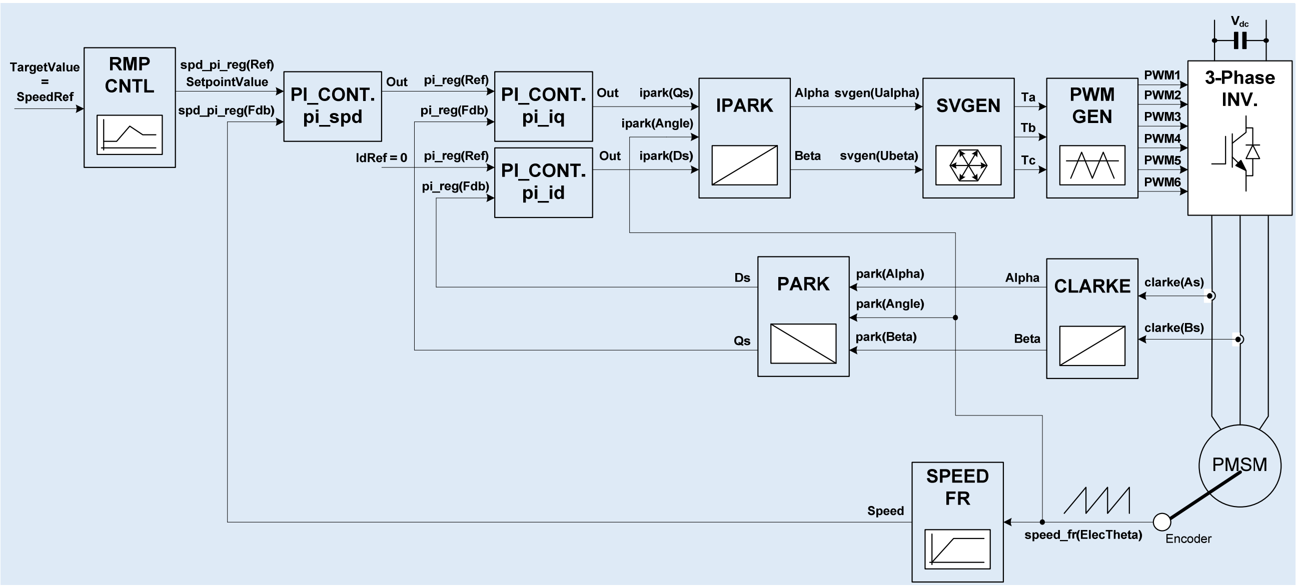
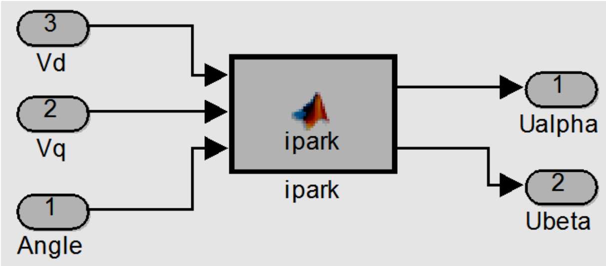


Figure 5. Overall block diagram of the speed sensored FOC of PMSM drive [20].




```
function [Alpha,Beta] = ipark(Ds,Qs,Angle)
Alpha = (Ds*cos(2*pi*Angle) - Qs*sin(2*pi*Angle));
Beta = (Qs*cos(2*pi*Angle) + Ds*sin(2*pi*Angle));
end
```

Figure 6. MATLAB Function based inverse Park transformation.

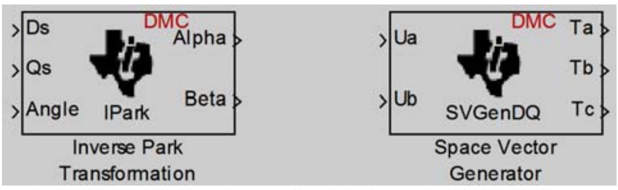


Figure 7. “Code Generation” toolbox blocks (Embedded Coder DMC modules).

III. Proposed FOC of PMSM Drive Model Based on MATLAB Functions in MATLAB/Simulink

C programming like codes written in MATLAB Programming language are developed for the simulation of the FOC of PMSM drive in MATLAB/Simulink using MATLAB Function blocks without using expensive additional toolboxes such as Embedded Coder. Motor control codes developed in C language by Texas Instruments are created in a modular basis [20]. Developers that create algorithms using the same basis can model the system by using MATLAB Function blocks.

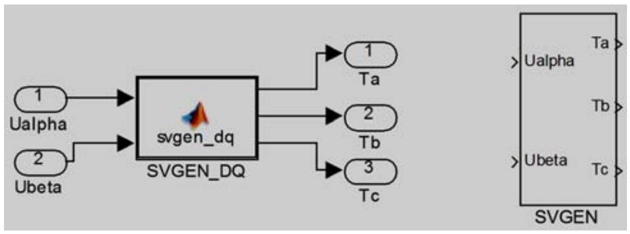


Figure 8. “Code Generation” toolbox blocks (Embedded Coder DMC modules).

As algorithms are being developed in MATLAB Function blocks, other electrical components such as electrical machines and inverters can also be modeled by using MATLAB/Simulink SimPowerSystems^{TM} library components in the same system model. Although, this causes an additional cost to the system, more realistic simulation models are obtained. The cost effective solution is to represent the electrical machine and inverter models mathematically by using classical Simulink and MATLAB Function blocks. MATLAB Function blocks can also be used in Embedded Coder for real-time implementation of FOC of PMSM drive if desired.

The MATLAB Function based inverse Park transformation model given in (10) is shown in Fig. 6. The mathematical representation written in the MATLAB Programming language is provided in the bottom side of the Fig. 6. In a similar fashion, Clarke and Park transformations, PI Controllers, Space Vector PWM Generator, Ramp Control, and Speed Calculation modules are also created by MATLAB Functions. The developed MATLAB Functions work in the same manner as the blocks in Embedded Coder. However, there are limitations in the Embedded Coder since TI's Digital Motor Controller (DMC) blocks in Embedded Coder do not allow modifications. In Fig. 7, inverse Park transformation and Space Vector Generator DMC blocks (modules) are illustrated. MATLAB Function blocks require C like MATLAB programming language. By this method, it is easy to develop and test the algorithms which are suitable for TI C2000 microcontrollers.

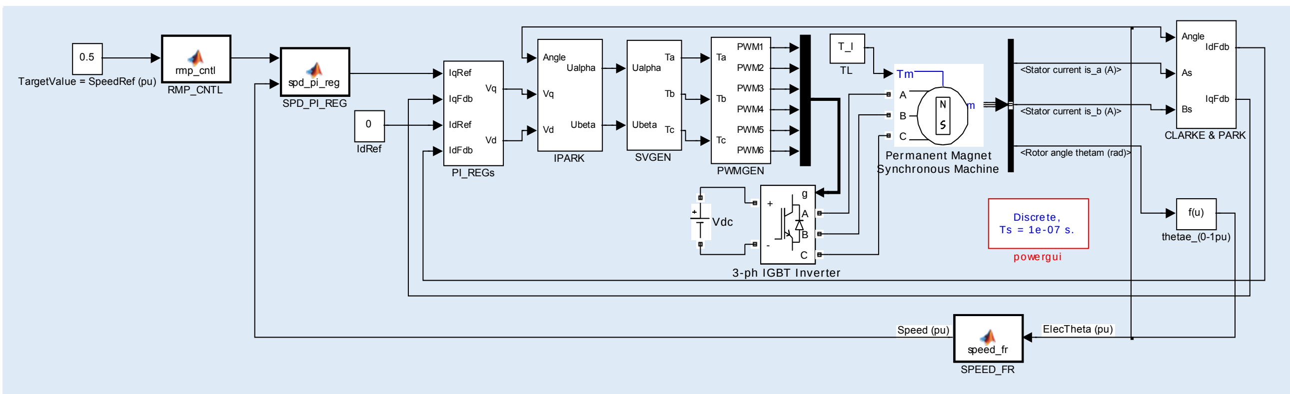


Figure 9. Overall MATLAB/Simulink block diagram of the proposed speed sensored FOC of PMSM drive using MATLAB Function.

Development of the simulation model before the actual implementation is important. It is quite hard to switch from the simulation platform of FOC of PMSM drive to the experimental stage. Moreover, programming microcontrollers is a quite tedious task. The FOC algorithms tested in the MATLAB/Simulink simulation platform by using MATLAB Functions instead of classical Simulink blocks makes switching to the experimental stage a lot easier due to the above mentioned reasons and generates much more realistic results in simulation.

One of the other advantages of using the MATLAB Function based blocks instead of DMC blocks is to enable development of new algorithms instead of the known ones that are provided in the Embedded Coder toolbox. The six-switch version of Space Vector Generator (SVGenDQ) block from DMC module is shown in Fig. 7. In this block, modifications are not allowed. However, Fig. 8 shows the MATLAB Function based SVGEN_DQ block which is easily modifiable.

The inside of the SVGEN_DQ MATLAB Function module is provided in Appendix B. Appendix C represents the C code version of the SVGEN_DQ module available in the TI motor development application library. Appendix D includes flowchart and pseudo code of the SVGEN_DQ module. The similarities are clearly seen between MATLAB Function version and C programming version of the SVGEN_DQ module. The FOC of PMSM drive that is proposed by Texas Instruments is illustrated in Fig. 9. In this figure, the closed loop speed control is performed with a ramp shape speed reference profile. The speed control is achieved by applying a speed PI regulator. Position information is obtained by an optical encoder. The encoder signals are converted to speed information by SPEED_FR block. The rest of the FOC components are also seen in Fig. 9. In Fig. 9, FOC of PMSM drive model developed in MATLAB/Simulink with MATLAB Functions along with electrical components is shown. Inverter and electrical machine parameters are selected to be the same as the experimental setup. The MATLAB Function blocks are then manually converted to the C language based version to be used in the Code Composer Studio™ (CCS) IDE. It is observed that any modifications made either in simulation or in experiment produce the same results in each platform.

IV. Simulation and Experimental Results

D. Simulation Results

The proposed drive system shown in Fig. 9 has been simulated in MATLAB/Simulink using an electrical IGBT inverter and electrical PM motor model from the SimPowerSystems toolbox in order to demonstrate the validity of the speed sensored FOC of a PM synchronous motor drive.

To set the gating signals of the power switches from the output of the SVPWM module easily and represent the real conditions in simulation as close as possible, the proposed drive scheme including the electrical model of the actual PM motor and the inverter with power semiconductor switches considering the snubber circuit and the parameters of the switches are designed in MATLAB/Simulink using the SimPowerSystems blocksets. The dead-time of the inverter and non ideal effects of the PM synchronous machine are neglected in the simulation models. The dc-link voltage $\boldsymbol{V_{dc}}$ is set to 400 V [21], [22].

The switching frequency is chosen as 10 kHz. The sampling interval of the electrical components from SimPowerSystems is selected to be thousand times slower than the sampling time. In the simulations, electrical components and mathematical functions run together. When the switching frequency is 10 kHz, the mathematical functions that run parallel should be sampled faster than 10 kHz. In this case, this is set a thousand times faster than 100 µs sampling rate. If the sampling step is low, the simulation slows down. Similarly, in experiment, the switching update is accomplished in 10 kHz rate as in simulation. However, the microcontroller runs the algorithms in 150 MHz clock frequency and the PWM update is performed in 10 kHz independent of the algorithm sampling as oppose to the simulation. This results in high frequency ripples in the actual current waveforms in experiment compared to the ones in simulation.

In Fig. 10, the speed results under full load start-up condition (2 N·m) is given with a ramp speed reference from zero speed to 0.5 p.u. (450 r/min) in 2 s. As it can be seen in Fig. 10 that the proposed MATLAB Function based speed sensored field-oriented control is able to drive the PM motor without any instability under full load start-up condition. Fig. 11 demonstrates the steady-state phase–$\boldsymbol{a}$ current waveform under nominal full load at start-up. At 7.5 s (steady-state), step-down full load rejection (full load to zero load) is applied and the speed response is provided in Fig. 10. No instability is observed even under full load rejection at steady-state.

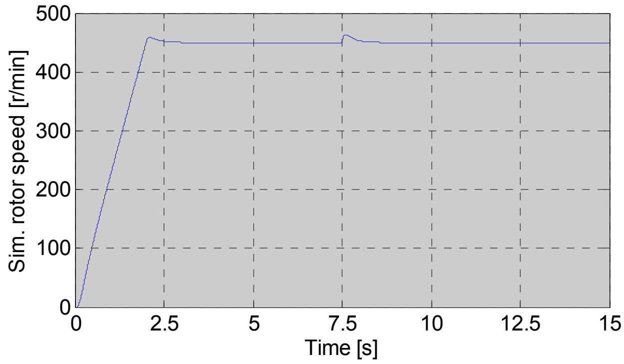


Figure 10.Simulated rotor speed feedback under full load start-up (2 N·m).

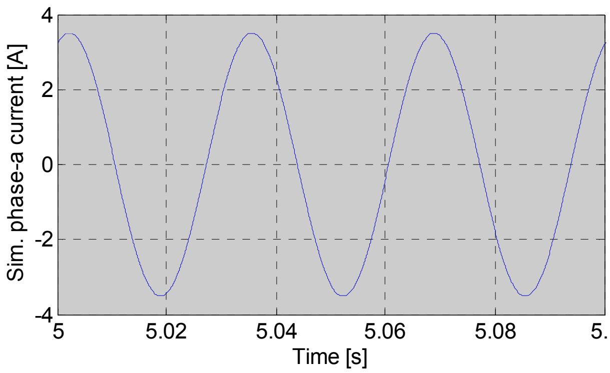


Figure. 11. Simulated phase–*a* current waveform at steady-state (0.5 p.u. = 450 r/min) under full load condition.

E. Experimental Setup

The experimental set-up shown in Figs. 12(a) and (b) consists of a Magtrol AHB-6 model hysteresis dynamometer set, a SEMIKRON Semiteach inverter, a PM synchronous motor, an eZdspTM board with TI's TMS320F28335 MCU chip, an interface board and a signal conditioning card.

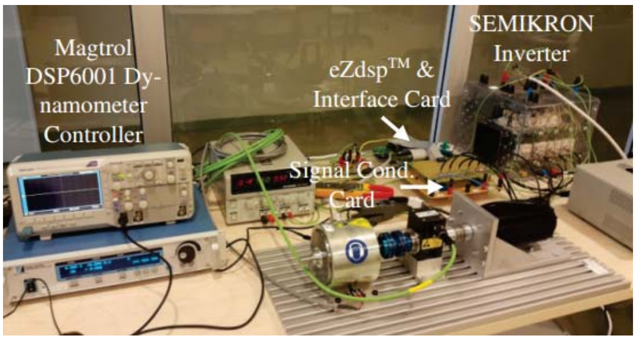
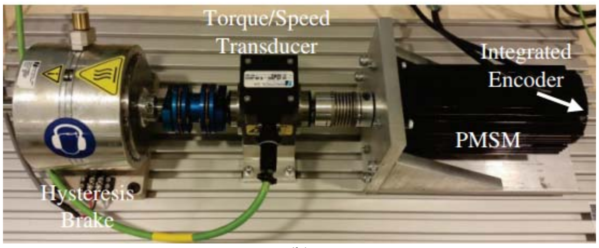


Figure 12(a)



(b)

Figure 12. Experimental test-bed. (a) Dynamometer controller, inverter, DSP control unit, and interface and signal-conditioning cards. (b) PM synchronous motor with integrated incremental position encoder (2500 pulse/rev.) coupled to hysteresis brake through torque/speed transducer.

The Magtrol dynamometer set contains a 6 N·m hysteresis brake, a DSP6001 model programmable DSP torque controller, and a Magtrol TMS306 model torque transducer to monitor the load torque and shaft speed which is installed between the hysteresis brake and the motor. The signal conditioning card which comprises of two LEM LA25NP model current sensors read two motor phase currents and convert the real-world analog current values into equivalent low-voltage values. The interface board that is connected to the signal conditioning card scales the voltage values into proper positive representations for the MCU to sensitize. The SEMIKRON Semiteach PWM voltage-source inverter (VSI) consists of SKM 50 GB 123D model IGBT modules, SKHI 22 model gate drivers with 4.3 µs dead-time, and two 2200 µF caps. The inverter has a maximum dc-link voltage of 750 V and rms current of 30 A.

In addition, an optical incremental encoder integrated to the PM servo motor with 2500 ppr resolution is used to detect the actual position. The speed sensored control scheme is verified using an off-the-shelf 2 N·m surface-mounted PMSM drive which is coupled to the overall system, as shown in Fig. 12(b).

F. *Experimental Results*

The feasibility and practical features of the sensored speed control scheme of a PMSM drive with sinusoidal back-EMF have been evaluated using an experimental test-bed, shown in Figs. 12(a) and (b). The same conditions are applied as in simulation. The control algorithm is digitally implemented using the eZdsp board from Spectrum Digital, Inc. based on a TI's floating-point Delfino MCU (TMS320F28335), as shown in Fig 12(a).

The experimental results are shown in Figs. 13 and 14. Fig. 13 shows experimental waveform of the speed response when zero to 0.5 p.u. ramp speed reference is applied starting at the tenth seconds under nominal load condition as in simulations. It is seen in Fig. 13 that the feedback speed reaches the reference top speed in 2 s as in simulation. The full load rejection is applied at 17.5 s. The drive system under full load rejection is still stable and the speed feedback tracks the reference speed closely. Fig. 14 shows the steady-state current waveform under full load condition.

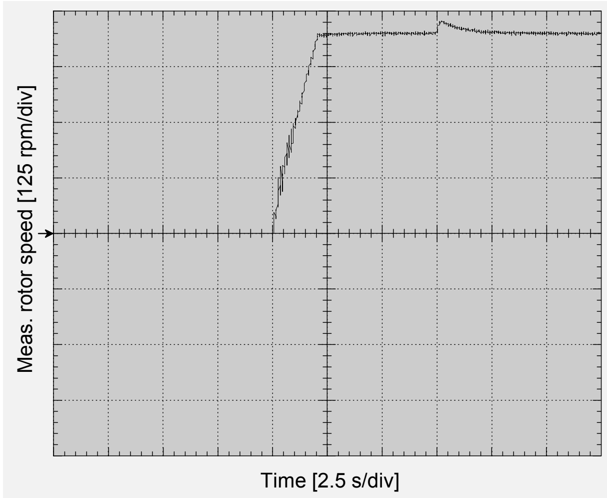


Figure 13. Experimental rotor speed waveform under full load start-up (2 N·m).

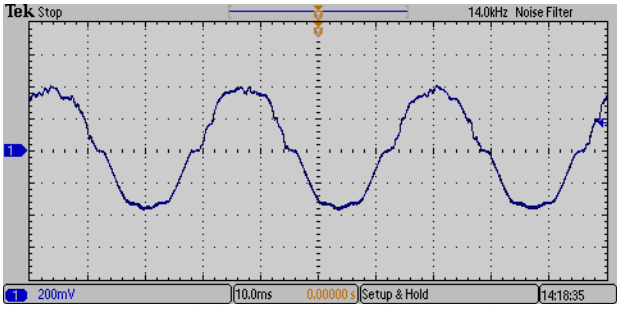


Figure 14. Experimental phase–*a* current waveform at steady-state (0.5 p.u. = 450 r/min) under full load condition (1A/100mV).

To investigate the current wave shape in the experiment, noise filter of 14 kHz is employed in the Tektronix oscilloscope to remove the high frequency ripples in the current waveform. It is seen that the results obtained from experiments are similar to those that are obtained in the simulations. The differences observed in the transient responses between simulation and experiment are because of the dissimilarities in data sampling rate, additional delay due to dynamometer torque controller, nonlinear characteristics of the machine and hysteresis brake, mismatch of moment of inertia, damping and friction of the overall system compared to the one used in simulations.

Due to slight misalignment and mechanical possible slippage in the motor coupling, phase current shows some harmonic signatures on the positive cycle and the rotor speed has oscillatory behavior at start-up as seen in Figs. 13 and 14, respectively. Moreover, because the dead-time effect is not compensated, the phase current waveform exhibits some additional distortion especially at zero crossings and at around positive and negative peaks. The rotor speed data are obtained by using M-TEST 5.0 Motor Testing Software of Magtrol dynamometer at 0.1 s sample rate.

V. Conclusion

In this paper, a simple, easily modifiable and more economical choice of modeling and simulation of a speed sensored field-oriented control (FOC) of a permanent magnet synchronous motor (PMSM) drive is developed by using MATLAB Function blocks in MATLAB/Simulink. This method allows easier algorithm and software development stages for experimental studies compared to the classical block diagram approach. The superiority of the method over commonly used "Code Generation" tools such as MATLAB/Simulink Embedded Coder is also emphasized. The proposed MATLAB/Simulink model of a speed sensored FOC of a PMSM drive scheme is built by using MATLAB programming in MATLAB Functions similar to C programming language. Then, the MATLAB programming based codes developed in simulation are implemented in a TI's TMS320F28335 Delfino floating-point MCU by using C programming. Simulation and experimental results are compared and the results show the effectiveness of the proposed modeling of the FOC of PMSM drive.

Acknowledgments

This work was supported by The Scientific and Technological Research Council of Turkey (TUBITAK) funded project (112E263). The authors would like to thank Gurkan Bozkurt for his help in preparation of the experimental setup and to Dr. Basar Ozkan for his help in reviewing the paper. The authors also thank Dr. Burak Kelleci for insightful technical discussions and his valuable feedback which helped in improving the paper.

References

[1] L. H. Hoang, "Modeling and simulation of electrical drives using MATLAB/Simulink and Power System Blockset," in *Proc. IEEE IECON*, Denver, CO, Nov. 29–Dec. 2, 2001, vol. 3, pp. 1603–1611.

[2] S. B. Ozturk, B. Akin, H. A. Toliyat, and F. Ashrafzadeh, "Low-cost direct torque control of permanent magnet synchronous motor using Hall-effect sensors," in *Proc. IEEE APEC*, Dallas, TX, Mar. 19–23, 2006, pp. 667–673.

[3] H. S. Zhuqiang, H. L. Hess, and K. M. Buck, "The modeling and simulation of a permanent magnet synchronous motor with direct torque control based on Matlab/Simulink," in *Proc. IEEE IEMDC*, San Antonio, TX, May 15–18, 2005, pp. 1150–1156.

[4] L. Ting, Y. Tan, G. Wu, and W. Shumao, "Simulation of PMSM vector control system based on Matlab/Simulink," in *Proc. IEEE ICMTMA*, vol. 2, Zhangjiajie, Hunan, China, Apr. 11–12, 2009, pp. 343–346.

[5] X. Wang, R. Na, and N. Liu, "Simulation of PMSM field-oriented control based on SVPWM," in *Proc. IEEE VPPC*, Dearborn, MI, Sep. 7–10, 2009, pp. 1465–1469.

[6] Z. Zhang and J. Shu, "Matlab-based permanent magnet synchronous motor vector control simulation," in *Proc. IEEE ICCSIT*, Amsterdam, Netherlands, Dec. 10–11, 2010, pp. 539–542.

[7] A. Kamalaselvan and S. L. Prakash, "Modeling simulation and analysis of closed loop speed control of PMSM drive system," in *Proc. IEEE ICCPCT*, Nagercoil, India, Mar. 20–21, 2014, pp. 692–697.

[8] G. Yan, "Simulation for the Vector Control Algorithm of Permanent Magnet Synchronous Motor," in *Proc. IEEE IHMSC*, Aug. 26–27, 2015, vol. 1, pp. 456–459.

[9] H. Qi, Y. Zhang, and N. Gao, "Research and implement of PMSM regenerative braking strategy based on controllable rectification," in *Proc. IEEE PEDS*, Jun. 9–12, 2015, pp. 289–294.

[10] A. A. Laczko, M.V. Zaharia, M.M. Radulescu, and S. Brisset, "Modeling and simulation of a brushless DC permanent-magnet generator-based wind energy conversion system," in *Proc. IEEE EVER*, Mar. 31–Apr. 2, 2015, pp. 1–7.

[11] M. S. Mahammadsoaib, M. P. Sajid, "Vector controlled PMSM drive using SVPWM technique - A MATLAB / simulink implementation," in *Proc. IEEE EESCO*, Jan. 24–25, 2015, pp. 1–5.

[12] D.J. Vaghela, H.N. Chaudhari, "Investigation of design, analysis and performance of hysteresis current control PMSM drive," in *Proc. IEEE EESCO*, Jan. 24–25, 2015, pp. 1–6.

[13] R. Duma, P. Dobra, M. Abrudean, and M. Dobra, "Rapid prototyping of control systems using embedded target for TI C2000 DSP," in *Proc. IEEE MED*, Athens, Greece, Jun. 27–29, 2007, pp. 1–5.

[14] C. Rusu, I. Birou, M. M. Radulecu, and A. Bara, "Developing embedded control system platform for testing PMSM drives," in *Proc. IEEE EPE*, Iasi, Romania, Oct. 16–18, 2014, pp. 677–682.

[15] C. Rusu, M. M. Radulescu, S. Enikö, R. K. Melinda, and Z. L. Jakab, "Embedded motor drive prototype platform for testing control algorithms," in *Proc. IEEE ICATE*, Craiova, Romania, Oct. 23–25, 2014, pp. 1–6.

[16] B. K. Bose. *Power Electronics and Variable Frequency Drives - Technology and Application*. Piscataway, NJ: IEEE Press, 1997.

[17] R. Krishnan. *Permanent Magnet Synchronous and Brushless DC Motor Drives*. Boca Raton, FL: CRC Press, 2009.

[18] P. Pillay and R. Krishnan, "Modeling, simulation and analysis of permanent magnet motor drives, Part I: The permanent magnet synchronous motor drive," *IEEE Trans. Ind. Appl.*, vol. 25, no. 2, pp. 265–273, Mar./Apr. 1989.

[19] S. B. Ozturk, "Modelling, simulation and analysis of low-cost direct torque control of PMSM using hall-effect sensors," M.S. thesis, Texas A&M University, College Station, TX, USA, 2005.

[20] M. Bhardwaj, "Sensored Field Oriented Control of 3-Phase Permanent Magnet Synchronous Motors," Texas Instruments, Dallas, TX, USA, Appl. Rep. SPRABQ2, Jul. 2013.

[21] S. B. Ozturk and H. A. Toliyat, "Direct torque and indirect flux control of brushless dc motor," *IEEE/ASME Trans. Mechatronics*, vol. 16, no. 2, pp. 351–360, Apr. 2011.

[22] S. B. Ozturk, W. C. Alexander, and H. A. Toliyat, "Direct torque control of four-switch brushless dc motor with non-sinusoidal back EMF," *IEEE Trans. Power Electron.*, vol. 25, no. 2, pp. 263–271, Feb. 2010.

Appendix A

Parameters and Specifications of the PMSM

Number of poles	8
Line-to-neutral rms voltage (V)	230
Rated speed (rpm)	3000
Rated rms current (A)	4
Rated torque (N·m)	2
Stator inductance (mH)	0.0033
Stator resistance (Ω)	3.4
Rotor magnetic flux linkage (Wb)	0.095
Moment of inertia (kg.m^2)	0.0075

Appendix B

```
function [Ta,Tb,Tc] = svgen_dq(Ualpha,Ubeta)

  tmp1 = Ubeta;
  tmp2 = Ubeta/2 + sqrt(3)/2 * Ualpha;
  tmp3 = tmp2 - tmp1;

  VecSector = 3;
  if (tmp2 > 0)
    VecSector = VecSector - 1;
  end
  if (tmp3 > 0)
    VecSector = VecSector - 1;
  end
  if (tmp1 < 0)
    VecSector = 7 - VecSector;
```

```
    end

    if(VecSector == 1 || VecSector == 4)
        Ta = tmp2;
        Tb = tmp1 - tmp3;
        Tc = -tmp2;
    elseif(VecSector == 2 || VecSector == 5)
        Ta = tmp3 + tmp2;
        Tb = tmp1;
        Tc = -tmp1;
    else
        Ta = tmp3;
        Tb = -tmp3;
        Tc = -(tmp1 + tmp2);
    end

end
```

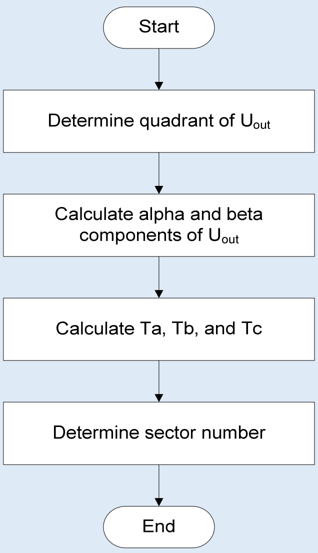
APPENDIX C

```
v.tmp1= v.Ubeta;
v.tmp2= _IQdiv2(v.Ubeta) +(_IQmpy(_IQ(0.866),v.Ualpha));
v.tmp3= v.tmp2 - v.tmp1;

v.VecSector=3;

v.VecSector=(v.tmp2> 0)?( v.VecSector-1):v.VecSector;
v.VecSector=(v.tmp3> 0)?( v.VecSector-1):v.VecSector;
v.VecSector=(v.tmp1< 0)?(7-v.VecSector) :v.VecSector;
   if(v.VecSector==1 || v.VecSector==4)
     { v.Ta= v.tmp2;
      v.Tb= v.tmp1-v.tmp3;
      v.Tc=-v.tmp2;
     }
   else if(v.VecSector==2 || v.VecSector==5)
     { v.Ta= v.tmp3+v.tmp2;
      v.Tb= v.tmp1;
      v.Tc=-v.tmp1;
     }
   else
     { v.Ta= v.tmp3;
      v.Tb=-v.tmp3;
      v.Tc=-(v.tmp1+v.tmp2);
     }
```

APPENDIX D



Pseudo Code:
Start
Determine quadrant of U_{out}
Calculate alpha and beta components of U_{out}
Calculate Ta, Tb, and Tc
Determine sector number
End