Efficient Verification of FPGA-Based FCS-MPCC Control for PMSM Drives Using Simulink/ModelSim Co-Simulation

Van-Quang-Binh Ngo * **Nguyen Khanh Quang** **

* *Faculty of Physics, University of Education, Hue University, Hue, Vietnam (e-mail: nvqbinh@hueuni.edu.vn).*
** *Faculty of Electrical Engineering, University of Science and Technology, The University of Danang, Da Nang, Vietnam (e-mail: nkquang@dut.udn.vn)*

(Corresponding author: Nguyen Khanh Quang, e-mail: nkquang@dut.udn.vn)

Abstract: Field-Programmable Gate Array (FPGA) technology has emerged as a cost-effective solution for implementing predictive controllers. Among these, Finite-Control-Set Model Predictive Current Control (FCS-MPCC) has gained recognition as a powerful strategy for permanent magnet synchronous motor (PMSM) drives due to its fast dynamic response and capability to handle system constraints. However, practical implementation requires rigorous verification prior to hardware deployment.This paper presents a co-simulation-based verification framework that integrates Simulink with ModelSim to validate the FCS-MPCC algorithm for PMSM drives at the Register-Transfer Level (RTL). The proposed approach combines MATLAB/Simulink for system-level modeling and testbench generation with ModelSim for cycle-accurate HDL simulation of the predictive controller. The framework enables early-stage functional verification, facilitates the detection of implementation inconsistencies, and supports improved consistency between control design and FPGA realization. Simulation results demonstrate the feasibility and correctness of the proposed approach, highlighting its effectiveness as a verification stage prior to hardware implementation.

Keywords: FPGA, FCS-MPCC, PMSM Drives, Permanent magnet synchronous motor, Co-Simulation.

1. INTRODUCTION

Permanent magnet synchronous motors (PMSMs) have become the foundation of modern electric drive systems thanks to their high power density, superior efficiency, and excellent dynamic performance. These features make PMSMs particularly suitable for applications such as electric vehicles, renewable energy conversion, and high-performance industrial automation (Quang et al. (2022); Quang and Phuc (2026)). To fully exploit these advantages, advanced control strategies are required to ensure fast transient response, high steady-state accuracy, and robustness against parameter uncertainties. Among the control paradigms that have emerged in the past two decades, Model Predictive Control (MPC)Model Predictive Control (MPC), particularly the finite control set variant (FCS-MPCC), has attracted broad attention in both academia and industry (Gang (2025); Hongfei Yang (2024)).

FCS-MPCC directly predicts motor current trajectories for each feasible inverter switching state and selects the optimal action by minimizing a cost function. Compared with conventional field-oriented control (FOC) and direct torque control (DTC), FCS-MPCC exhibits clear advantages in transient dynamics, multivariable handling, and a conceptually straightforward framework that avoids cascaded PI regulators and modulators. Numerous studies have demonstrated these capabilities, emphasizing its ability to reduce current ripple, suppress torque pulsations, and improve control accuracy under dynamic operating conditions (Petkar et al. (2021); Morel et al. (2009, 2008); Mwasilu et al. (2018); Wang et al. (2015)). Extensions such as multi-vector methods, duty-cycle concepts, and cost-function reformulations have been developed to enhance steady-state quality, improve robustness, and simplify controller design (Petkar et al. (2021); Peng and Yao (2023); Siami et al. (2016); Sandre-Hernandez et al. (2018)).

At the same time, comparative research has systematically evaluated predictive control schemes against classical approaches. For example, comparisons of predictive current control (PCC), two-vector predictive control (2PC), and predictive power control (PPC) reveal their characteristic trade-offs in terms of accuracy, parameter sensitivity, and inverter nonlinearities (Morel et al. (2009, 2008); Wang et al. (2015)). Other studies have contrasted

⋆ This research is funded by Hue University under grant number DHH2025-03-206. This work was partially supported by Hue University under the Core Research Program, Grant No. NCTB.DHH.2025.17.

FCS-MPCC with DTC, showing that predictive methods achieve smaller overshoot and faster settling times while maintaining flexibility in integrating system constraints (Morel et al. (2008); Brosch et al. (2021)). Collectively, these findings strengthen the position of FCS-MPCC as a promising strategy for high-performance PMSM drives.

As a recent example, Gang (2025) embedded a finite-set predictive current model regulator in the PMSM current loop and adopted a matrix converter, showing overshoot-free, 0.02 s settling and smoother torque versus vector control in MATLAB/Simulink. However, the study remained at algorithm-level simulation, without cycle-accurate RTL or FPGA-oriented co-simulation, leaving a gap our work addresses.

Despite these advantages, several inherent limitations remain. A recurring issue is the computational burden of enumerating and evaluating all switching states at each sampling step. This not only raises hardware requirements but also challenges real-time feasibility, particularly at high switching frequencies. Another issue is parameter sensitivity: predictive models strongly depend on machine parameters such as stator resistance and flux linkage, which vary with temperature and operating conditions. Moreover, the variable switching frequency inherent in FCS-MPCC complicates filter design and can reduce efficiency in some applications (Li et al. (2022); Wang et al. (2017)). These challenges have motivated multiple improvements, ranging from multi-vector strategies to reduce ripple (Petkar et al. (2021); Yang et al. (2023)), elimination of weighting factors for simplified tuning (Peng and Yao (2023)), to disturbance observers that enhance robustness against parameter mismatch (Sathyamoorthy (2022)).

Another vibrant research stream focuses on FPGA-based implementation, since field-programmable gate arrays provide the parallel processing capability needed for real-time predictive algorithms. Several works report efficient deployment using automated code generation tools such as MATLAB HDL Coder (Purraji et al. (2025)), which significantly shortens hardware prototyping cycles. Other studies introduce architectural optimizations, such as resource sharing, pipelining, and simplified search strategies, to reduce hardware footprint while sustaining high control frequencies (Yang et al. (2023); Wendel et al. (2017); Saeidi (2015); Ma et al. (2014)). For example, extended MPC schemes with hundreds of vectors have been realized on FPGA using simplified cost-function searches, achieving substantial reductions in current harmonic distortion without compromising transient dynamics (Yang et al. (2023)). More broadly, FPGA implementations confirm that predictive control is feasible in real time, provided that design workflows effectively bridge algorithm simulation and hardware realization.

Nevertheless, these implementation-oriented studies expose a critical gap. Most works rely either on high-level algorithm simulation in Simulink or on post-synthesis verification using HDL simulators, with few addressing their integration. The absence of combined system-level modeling and cycle-accurate Register-Transfer Level (RTL) verification prior to synthesis creates risks: algorithm-level validation alone may overlook hardware timing issues, while RTL verification in isolation lacks system context. As a result, design errors may propagate into FPGA prototypes, leading to extended debugging, increased costs, and longer development cycles.

To address this challenge, the present paper proposes an efficient co-simulation framework integrating MATLAB/Simulink and ModelSim for FPGA-oriented FCS-MPCC controller verification. In this framework, Simulink is used for PMSM system modeling, algorithm design, and automated testbench generation, while ModelSim provides cycle-accurate RTL simulation of the predictive controller. This dual environment enables rigorous cross-verification: the predictive algorithm is validated under realistic machine and inverter dynamics, while the HDL implementation is simultaneously checked for correctness and timing consistency.

Compared with commonly used verification approaches such as hardware-in-the-loop (HIL) and direct FPGA prototyping, the proposed Simulink/ModelSim co-simulation framework is intended for an earlier stage of controller verification. Unlike direct FPGA prototyping, the proposed approach does not require immediate deployment on a physical FPGA platform, which makes it more flexible for early debugging and functional validation. Likewise, unlike HIL-based validation, it is not aimed at evaluating full closed-loop real-time interaction between controller hardware and plant dynamics. Instead, the proposed framework focuses on detecting logic-level mismatches and implementation inconsistencies at the RTL stage before hardware-based validation. Therefore, the proposed method should be viewed as a complementary intermediate verification stage rather than a replacement for HIL or direct FPGA prototyping.

Simulation results demonstrate the feasibility and accuracy of the FCS-MPCC controller and highlight the utility of the co-simulation framework as a verification stage prior to hardware deployment.

In summary, the primary contribution of this paper lies in the proposed co-simulation methodology, which (i) provides a reliable platform for validating FCS-MPCC controllers at the RTL level before FPGA synthesis, (ii) facilitates early-stage detection of implementation mismatches and improves consistency between algorithm and FPGA implementation, and (iii) bridges algorithmic simulation and hardware deployment, supporting a smoother transition toward industrial applications of predictive control.

2. THE CONFIGURATION AND CORRESPONDING DYNAMIC MODEL OF THE SYSTEM

2.1 Mathematical model of the system

The configuration of the PMSM drive system can be depicted in Fig. 1, which employs a two-level voltage source inverter (2L-VSI). The corresponding voltage space vector allocation is detailed in Fig. 2. As shown, the inverter generates eight primary voltage vectors, comprising six active vectors (u_1–u_6) and two zero vectors (u_0 and u_7).

This research utilizes a surface PMSM characterized by equal dq-axis inductance ($L_d = L_q = L_s$), leading to the formulation of the dq-axis voltage expression for the PMSM as follows:

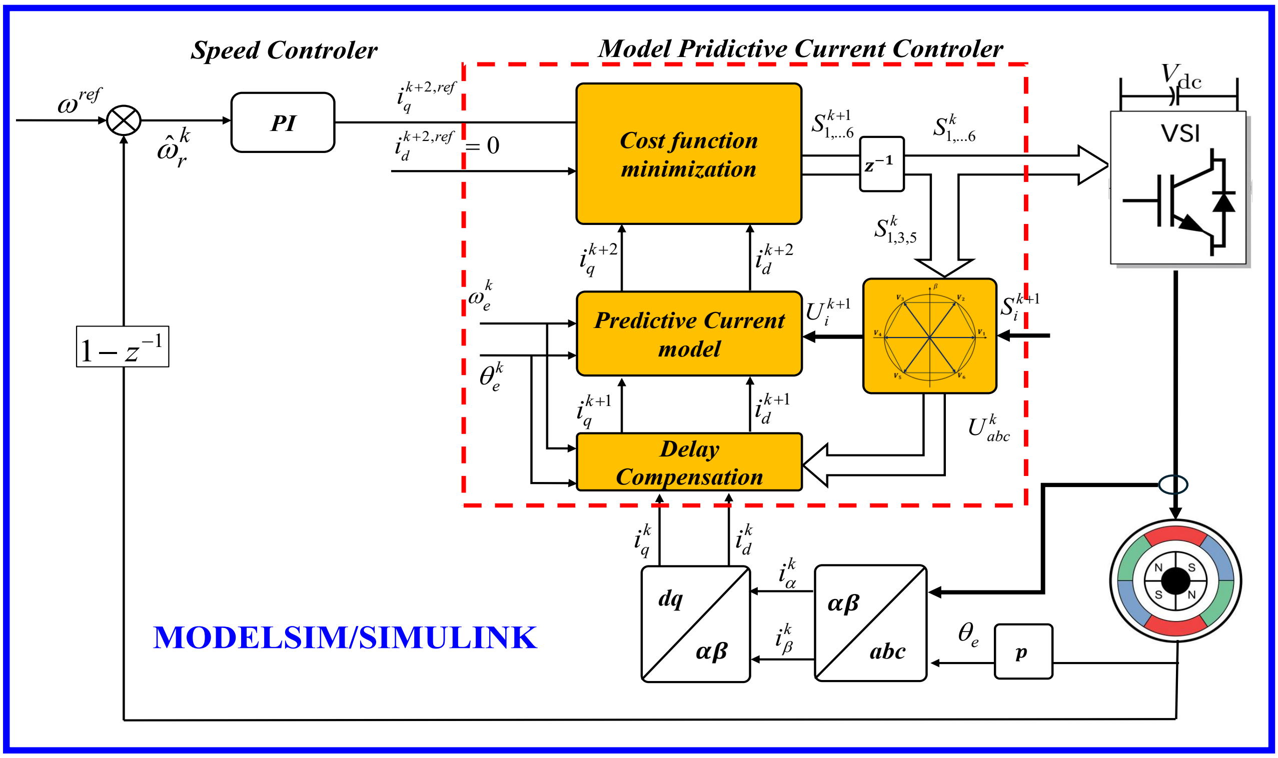

Fig. 1. The architecture of PMSM-based drive systems.

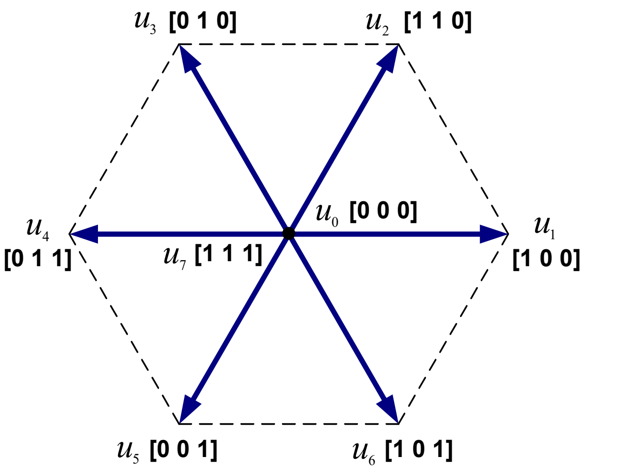

Fig. 2. The spatial configuration of voltage vectors in a 2L-VSI.

$$
\begin{aligned}
u_{sd} &= R_s i_{sd} + L_s \frac{di_{sd}}{dt} - \omega_e L_s i_{sq}, \\
u_{sq} &= R_s i_{sq} + L_s \frac{di_{sq}}{dt} + \omega_e L_s i_{sd} + \omega_e \psi_f,
\end{aligned}
\tag{1}
$$

where u_{sd}, u_{sq}, i_{sd}, and i_{sq} correspond to the voltage and current elements on the dq axis, respectively. ω_e and ψ_f symbolize the motor's electrical angular velocity and flux linkage of the permanent magnet, respectively, while R_s and L_s symbolize the resistance and inductance of the stator.

Adjusting the d-axis stator current facilitates precise regulation of electromagnetic torque:

$$
T_e = \frac{3}{2} n_p \psi_f i_{sq}, \tag{2}
$$

where n_p is the number of pole pairs.

The stator currents are calculated using equation (1), as shown below:

$$
\begin{aligned}
\frac{di_{sd}}{dt} &= \frac{1}{L_s} u_{sd} - \frac{R_s}{L_s} i_{sd} + \omega_e i_{sq}, \\
\frac{di_{sq}}{dt} &= \frac{1}{L_s} u_{sq} - \frac{R_s}{L_s} i_{sq} - \omega_e i_{sd} - \frac{\omega_e}{L_s} \psi_f,
\end{aligned}
\tag{3}
$$

where the stator voltages u_{sd} and u_{sq} in the synchronously rotating frame can be established via the Park's transformation matrix in the following manner:

$$
\begin{aligned}
\begin{bmatrix} u_{sd} \\ u_{sq} \\ u_{s0} \end{bmatrix} &= [P] \begin{bmatrix} u_{sa} \\ u_{sb} \\ u_{sc} \end{bmatrix}, \\
[P] &= \frac{2}{3} \begin{bmatrix} \cos(\theta) & \cos\left(\theta - \frac{2\pi}{3}\right) & \cos\left(\theta + \frac{2\pi}{3}\right) \\ \sin(\theta) & \sin\left(\theta - \frac{2\pi}{3}\right) & \sin\left(\theta + \frac{2\pi}{3}\right) \\ \frac{1}{2} & \frac{1}{2} & \frac{1}{2} \end{bmatrix},
\end{aligned}
\tag{4}
$$

in which θ denotes the the electrical angular displacement that corresponds to the rotor position.

Moreover, the two-level VSI generates the stator-winding voltage (see Fig. 1). The expression is obtained by analyzing the inverter's switching states together with the DC-link

$$
\begin{aligned}
u_{sa} &= \frac{V_{dc}}{3}\left(2S_a - S_b - S_c\right), \\
u_{sb} &= \frac{V_{dc}}{3}\left(2S_b - S_a - S_c\right), \\
u_{sc} &= \frac{V_{dc}}{3}\left(2S_c - S_a - S_b\right).
\end{aligned} \tag{5}
$$

where V_{dc} is the DC-bus voltage. S_a, S_b, S_c represent the switching status of the inverter leg in a three-phase 2L-VSI as:

$$
S_x = \begin{cases} 1 & \text{if upper switch "ON"} \\ 0 & \text{if upper switch "OFF"} \end{cases} \tag{6}
$$

Applying the explicit first-order integration method to discretize equation (3) over the sampling duration T_s yields the discrete-time representation of the pertinent model as given:

$$
\begin{aligned}
i_{sd}(k+1) &= \left(1 - \frac{R_s T_s}{L_s}\right) i_{sd}(k) + T_s \omega_e(k) i_{sq}(k) \\
&\quad + \frac{T_s}{L_s} u_{sd}(k), \\
i_{sq}(k+1) &= \left(1 - \frac{R_s T_s}{L_s}\right) i_{sq}(k) - T_s \omega_e(k) i_{sd}(k) \\
&\quad + \frac{T_s}{L_s} u_{sq}(k) - \frac{T_s}{L_s} \psi_f \omega_e(k).
\end{aligned} \tag{7}
$$

Due to the computational requirements of the model predictive controller, it is unable to implement the best switching state to the PMSM instantaneously following sampling. Consequently, a one-step delay is inevitable. Failure to compensate for this delay will result in a deterioration of the controller's performance. As a result, one-step delay compensation is incorporated into the analysis. Based on the motor's current state and potential control actions, the present measured value is utilized to execute one-step prediction as per equations (4)-(7), yielding the predicted value for time $k+1$, which serves as the estimated value for the subsequent prediction. This allows for the forecasting of the stator current at time $k+2$ in the future, enabling the calculation of the corresponding error. By applying the principle of minimizing the cost objective, the best voltage vector required at time $k+1$ can be identified. The prediction equation post one-step delay compensation is presented as follows:

$$
\begin{aligned}
i_{sd}(k+2) &= \left(1 - \frac{R_s T_s}{L_s}\right) i_{sd}(k+1) \\
&\quad + T_s \omega_e(k+1) i_{sq}(k+1) + \frac{T_s}{L_s} u_{sd}(k+1), \\
i_{sq}(k+2) &= \left(1 - \frac{R_s T_s}{L_s}\right) i_{sq}(k+1) + \frac{T_s}{L_s} u_{sq}(k+1) \\
&\quad - T_s \omega_e(k+1) i_{sd}(k+1) - \frac{T_s}{L_s} \psi_f \omega_e(k+1).
\end{aligned} \tag{8}
$$

This research emphasizes rotor speed regulation with fast dynamic reaction and minimal steady-state error. To meet these objectives, conventional FCS-MPC approaches for PMSMs employ a cascaded structure, consisting of an outer speed loop regulated by a PI controller and an inner stator current loop managed by a predictive controller. The corresponding cost function, incorporating compensation for computational delays, takes the following form:

$$
\begin{aligned}
g(u_{k+1}) &= \left\| i_{sd}^*(k+2) - i_{sd}(k+2) \right\|^2 \\
&\quad + \left\| i_{sq}^*(k+2) - i_{sq}(k+2) \right\|^2,
\end{aligned} \tag{9}
$$

where $i_{sd}^*(k+2)$ and $i_{sq}^*(k+2)$ denote the desired dq-axis stator current components predicted for the sampling step $k+2$. The second-order Lagrange extrapolation technique can be utilized to accomplish these values as given:

$$
\begin{aligned}
i_{sd}^*(k+2) &= 6 i_{sd}^*(k) - 8 i_{sd}^*(k-1) + 3 i_{sd}^*(k-2), \\
i_{sq}^*(k+2) &= 6 i_{sq}^*(k) - 8 i_{sq}^*(k-1) + 3 i_{sq}^*(k-2).
\end{aligned} \tag{10}
$$

The control variables of the PMSM drive u_{sd} and u_{sq} are generated from the array of switching states related to the 2L-VSI $S = [S_a, S_b, S_c]^T$, where x is an element of the set $\{a, b, c\}$. Each switch's state S_x is restricted to a specific subset $\{0, 1\}$. It is imperative to note that the control stator voltage is bounded to a finite set $\{u_0, ..., u_7\}$. Therefore, the realization of the optimal switching status S_{opt} in PMSM systems employing a 2L-VSI is performed by addressing the optimization task:

$$
\begin{aligned}
S_{opt} &= argmin \left\{ g\left(u_{k+1}\right) \right\}, \\
\text{s.t.} \quad & u_{k+1} \in \left\{ u_0, ..., u_7 \right\}, \; u_i \in \{0, 1\}^3 \\
& g(u_{k+1}) = \left\| i_{sd}^*(k+2) - i_{sd}(k+2) \right\|^2 \\
& + \left\| i_{sq}^*(k+2) - i_{sq}(k+2) \right\|^2
\end{aligned} \tag{11}
$$

3. SIMULINK/MODELSIM CO-SIMULATION AND FSM IMPLEMENTATION

3.1 Simulink/Modelsim Co-simulation Method

The simulation of the PMSM drive with MPCC for speed regulation is carried out through the integration of Electronic Design Automation (EDA) tools, establishing a connection between MATLAB/Simulink and the VHDL simulation environment implemented in ModelSim. This co-simulation architecture allows for accurate behavioral validation, closely reflecting the actual hardware operations. In Fig. 1, the speed control model for the PMSM system is presented, while Fig. 3 details the Simulink/ModelSim co-simulation architecture. The PMSM system, IGBT inverter, and speed control signals are simulated in Simulink, while the MPCC controller is described through VHDL code and executed in ModelSim. The modules in ModelSim, from module-1 to module-3, perform essential functions: module-1 implements the PI speed controller, module-2 transforms three-phase current signals (a, b, c) into the $d-q$ coordinate system (direct and quadrature axes), and module-3 executes the Model Predictive Current Control for the PMSM system. These modules are all described in VHDL and co-simulated through the EDA link with ModelSim, enabling precise parameter testing and adjustment. The co-simulation process ensures accurate evaluation of dynamic and control factors, optimizing the simulation parameters to guarantee high efficiency and accuracy in speed control. The PMSM parameters used in the simulation include number of pole pairs of 3, stator resistance of 0.95 Ω, stator inductance of 0.0098 mH, moment of inertia J=0.078 kgm^2. The

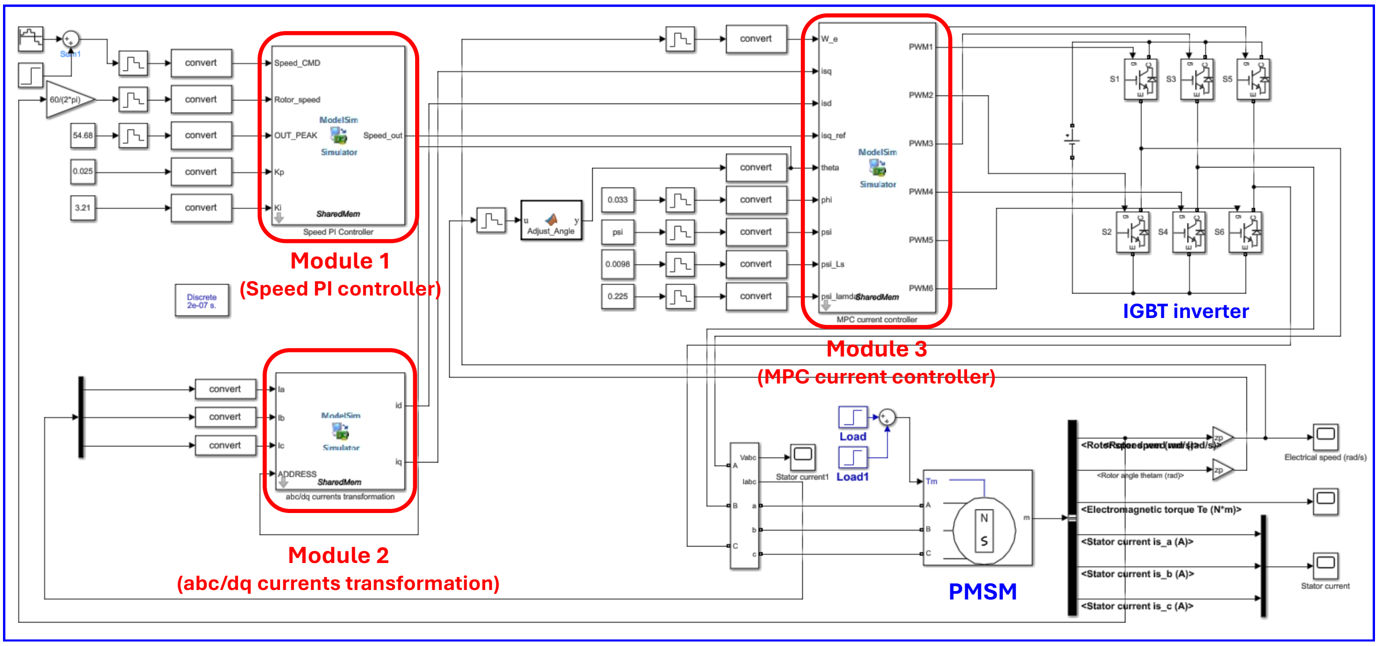

Fig. 3. The Simulink/ModelSim co-simulation architecture for FCS-MPCC control of PMSM drive.

sampling frequencies for current and speed control are set at 16 kHz and 2 kHz, respectively, with clocks at 50 MHz and 12.5 MHz providing synchronized signals for all modules in ModelSim. Before real-world implementation on FPGA, simultaneous simulation in Simulink and ModelSim helps assess and optimize the operational aspects of the system, ensuring high accuracy and efficiency in speed control. This co-simulation architecture not only ensures precise simulations but also optimizes the design process, preparing for efficient and reliable deployment on FPGA hardware.

4. FSM IMPLEMENTATION

The circuit architecture of the MPCC can be depicted in Fig. 4. The system consists of an adder, a multiplier, and a dedicated functional unit designed to predict the current components i_{sd} and i_{sq}, thereby enabling the complete execution of the control algorithm. All numerical processing is implemented in fixed-point arithmetic with 32-bit word length in Q15 format.

The algorithm is implemented as a finite state machine (FSM) and divided into multiple states. The process begins with the initialization of the optimal cost function value g_{opt} and the previous switching state x_{old} (S_0–S_{23}). Next, the stator $d-q$ reference currents are predicted (S_{24}–S_{47}). At states S_{48}–S_{50}, the predicted currents $i_{sd}(k+2)$, $i_{sq}(k+2)$ and the cost function g are calculated. A comparison at S_{51} updates the optimal state if $g < g_{opt}$. This prediction and evaluation procedure is repeated for candidate switching states indexed by $x = 1, ..., 7$ (S_{52}–S_{216}). In each iteration, the predicted currents and the cost function are computed, and the optimal state is updated accordingly (S_{79}–S_{217}). After all candidates are evaluated, the optimal switching state x_{opt} is selected and sent to a look-up table S_{218} to generate the corresponding PWM signals. Finally, the selected state is stored as x_{old} for the next cycle S_{219}. In this way, the FSM ensures that the controller continuously predicts, evaluates, and selects the optimal voltage vector to minimize the cost function.

The current prediction block for i_{sd} and i_{sq} is detailed in Fig. 5: states (S_0–S_{12}) compute the voltage components u_{sd} and u_{sq} according to (4), while states (S_{13}–S_{22}) predict the currents in (7). The speed PI controller as well as the coordinate transformation of the current axes are implemented using look-up tables (LUTs) and arithmetic operations (Quang and Phuc (2026)), requiring an additional 28 steps. Since each computational step takes approximately 80 ns at a clock frequency of 12.5 MHz, the entire algorithm is executed within about 5.92 μs.

5. EVALUATION RESULTS

To assess the performance of the proposed control method, tests are carried out both in steady-state operation at different rotor speeds and in transient conditions that include abrupt variations in speed and load torque.

5.1 Assessment under steady-state conditions

To validate the practicality of the proposed approach, we executed three evaluation cases. In the first case, the current controller operated at 10 kHz, a constant load torque of 9.02 Nm was applied, and the rotor speed was held at 3000 rpm. The corresponding steady-state behavior in Fig. 6 shows that the stator phase current is essentially sinusoidal with an electrical frequency of approximately 150 Hz, consistent with a constant mechanical speed of 3000 rpm. The d-axis current remains effectively zero, while the q-axis current settles near 8.91 A, indicating proper decoupling and torque production. A ripple analysis yields mean absolute errors (MAE) of 0.29 A (i_d) and 0.33 A (i_q); the rotor-speed and electromagnetic-torque MAEs are 1.2 rpm and 0.3 Nm, respectively. Collectively, these results demonstrate that the method suppresses variation in stator-current ripple, speed, and torque. To illustrate

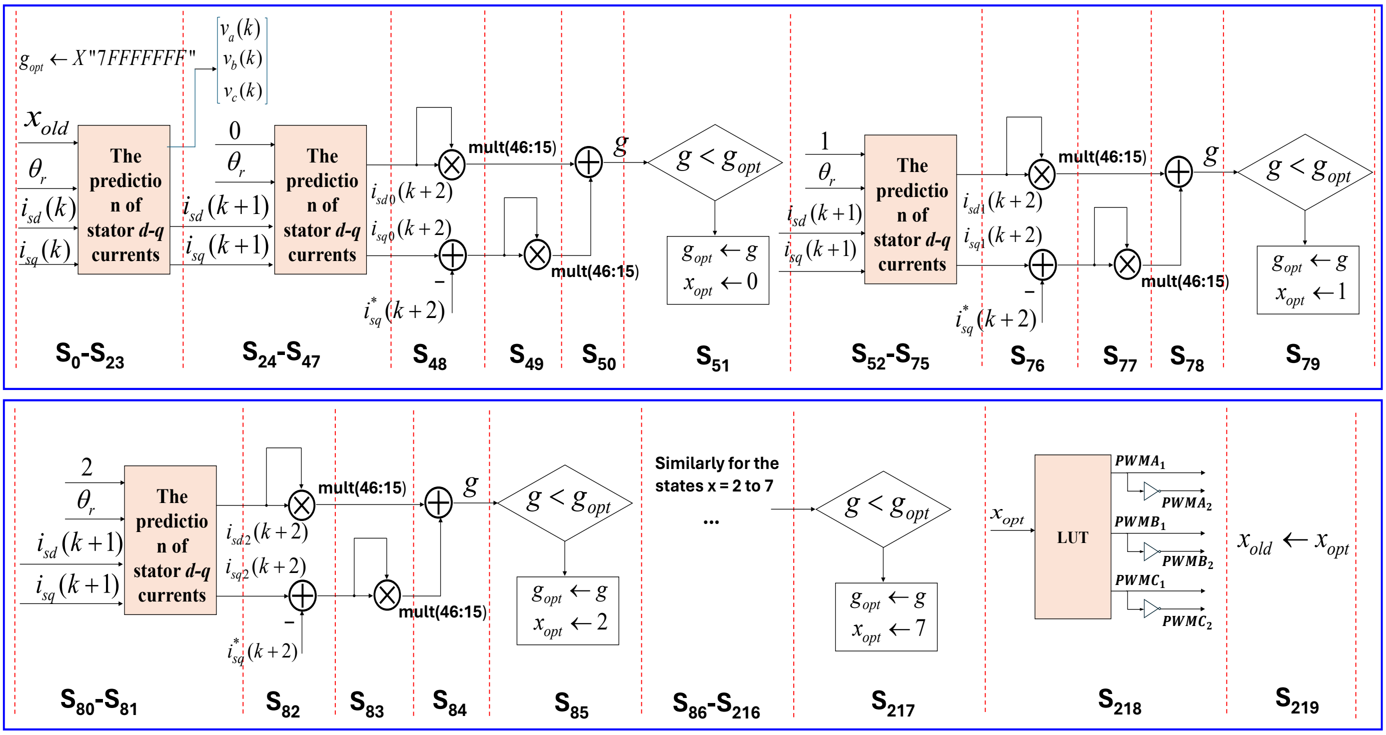

Fig. 4. State machine diagram of the proposed FCS-MPCC algorithm.

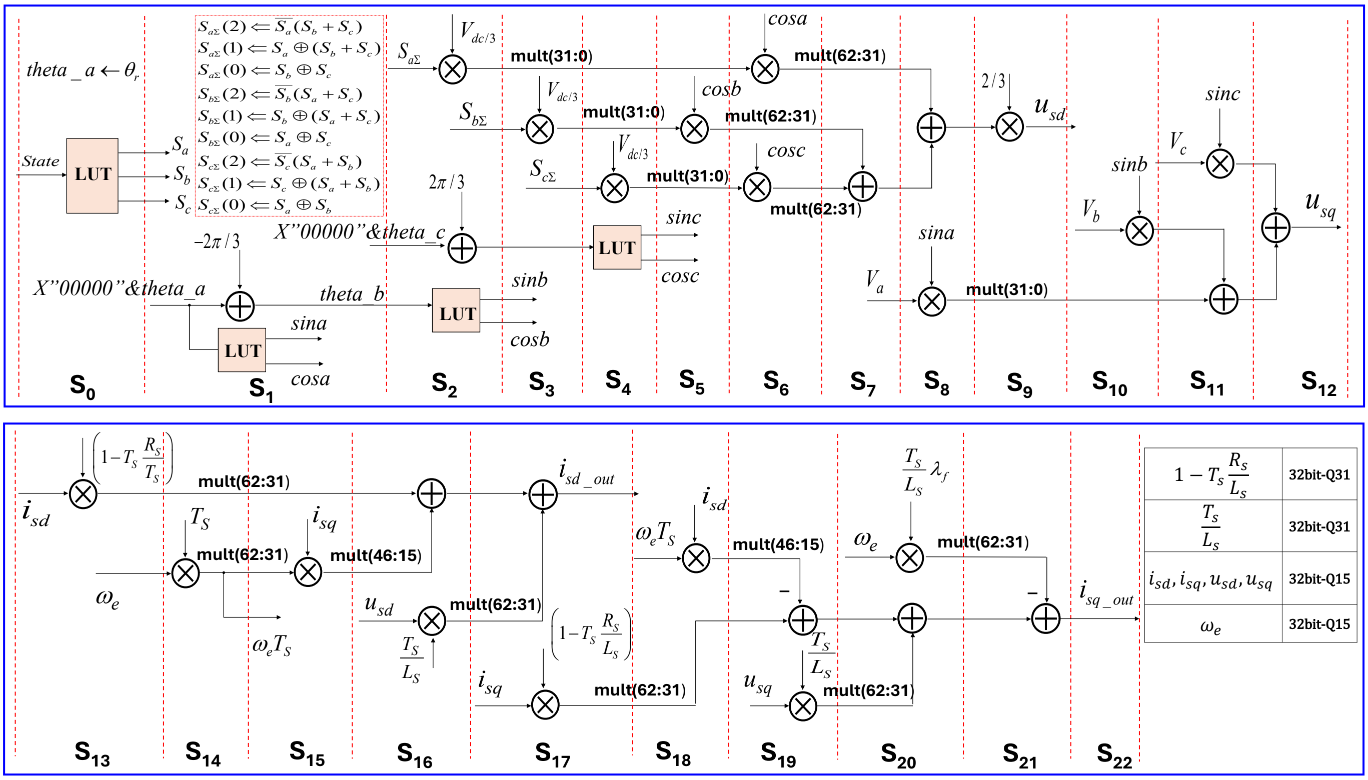

Fig. 5. State machine diagram for the prediction of stator $d-q$ axes currents.

the strategy's broad applicability, we further assessed the stator-current total harmonic distortion (THD) under a constant load across rotor speeds of 500, 1000, 1500, 2000, and 3000 rpm in Fig. 7. For each operating point, steady-state waveforms and their frequency spectra were obtained with the Simulink FFT toolbox, and the phase-A current THD was computed over a single fundamental cycle. At 3000 rpm, the fundamental is 150 Hz; harmonics up to the

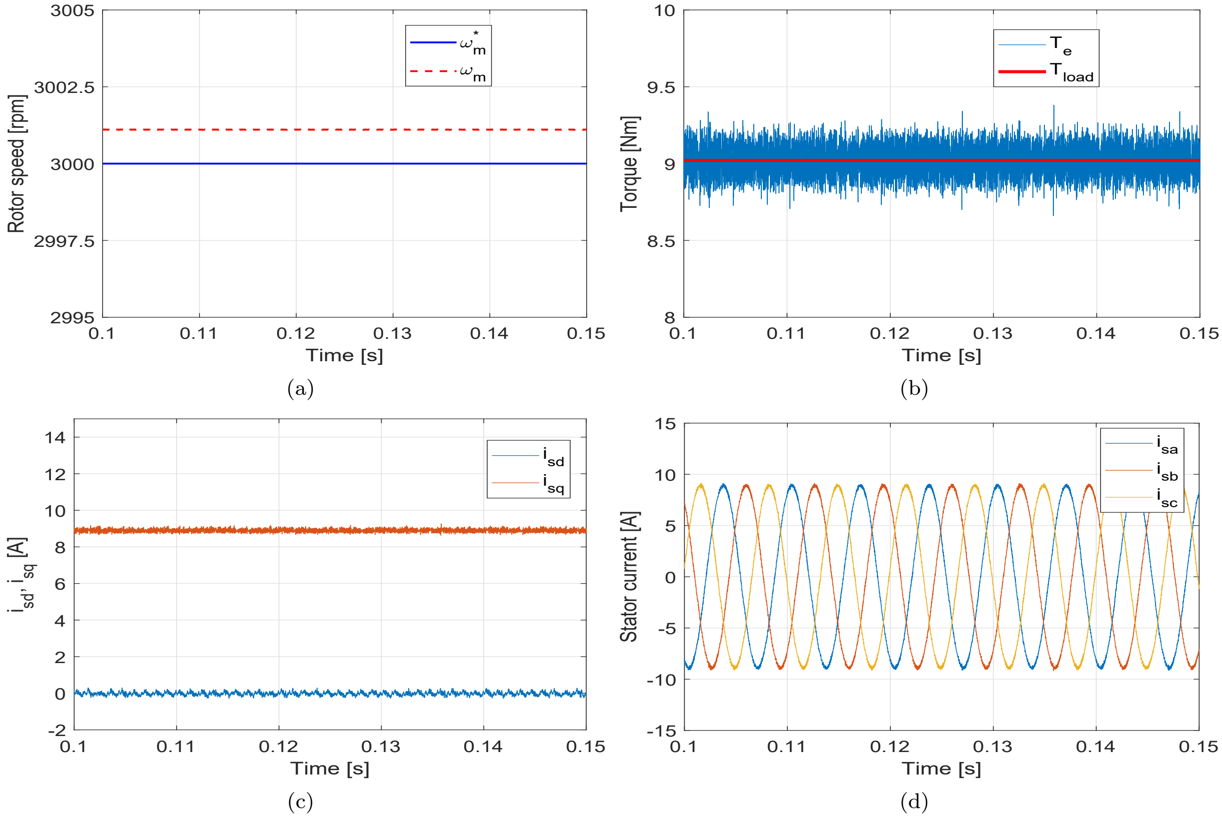

Fig. 6. Assessment of the proposed method under steady-state conditions: (a) Speed response, (b) Torque, (c) *dq*-axis currents and (d) Three-phase stator current.

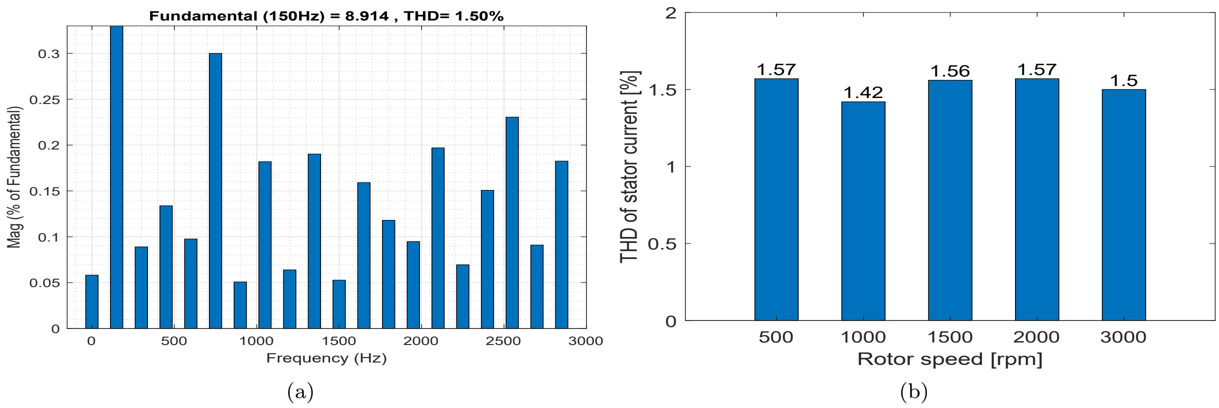

Fig. 7. THD performance analysis over a range of motor speeds: (a) Frequency spectrum of stator current at 3000 rpm and (b) Investigation of THD at different operating speeds.

20th multiple ($\leq$3000 Hz) were included in the analysis. The resulting THD is 1.5%, and across the tested speeds the stator-current THD spans 1.42%–1.57%, which is comfortably below the commonly cited 5% current-distortion benchmark in IEEE 519.

5.2 Transient performance evaluation

In the second scenario, the proposed method is evaluated under step-wise variations of rotor speed. The speed starts at 0 *rpm* and increases to 500 *rpm*, then to 1000 *rpm* at 0.1 *s*, 2000 *rpm* at 0.2 *s*, and 3000 *rpm* at 0.3 *s*. Following this, the speed is reduced to 1500 *rpm* at 0.4 *s*, reversed to –1000 *rpm* at 0.5 *s*, and finally increased back to 500 *rpm* at 0.6 *s*, as illustrated in Fig. 8. Simultaneously, the torque is kept at 9.02 *Nm*. As depicted in Fig. 8b, the PMSM begins its operation from a completely stationary state. It is capable of accelerating to a remarkable speed of 2000 *rpm* within a rapid duration of 23 *s*, before sustaining that speed for approximately 25 *ms*. When subjected to a step reference change, both the *q*-axis and *d*-axis currents exhibit a high degree of accuracy, closely following their designated reference values. This performance is achieved while adhering to a current limit of 50 *A*, as established by the PI controller within the speed regulation loop. The findings clearly indicate that the proposed control strategy significantly enhances the motor's performance in both transient and steady-state cases. The system achieves a

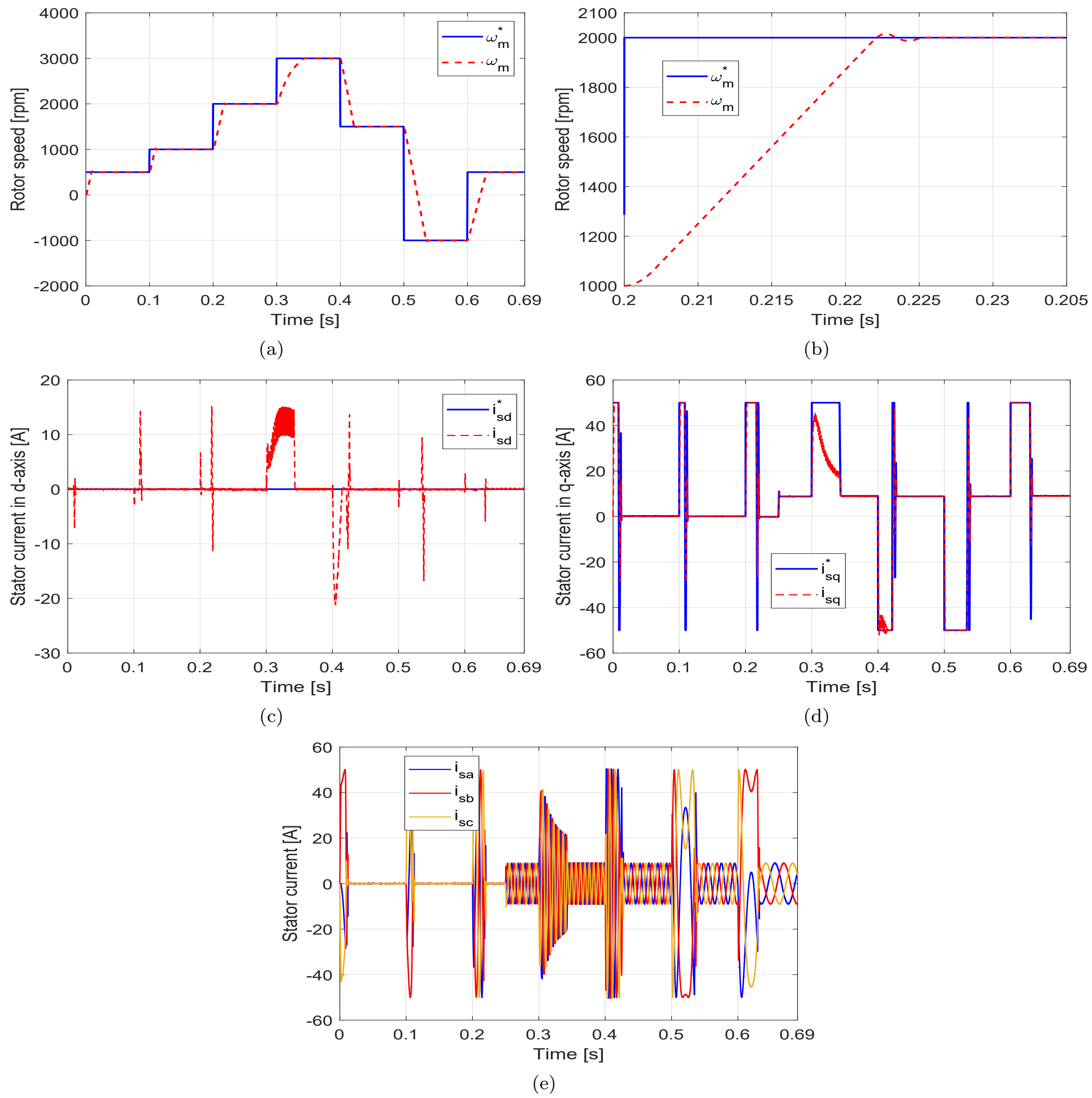

Fig. 8. Validation of the proposed method during transient states: (a) Speed response, (b) Zoom of speed, (c) d-axis current, (d) q-axis current and (e) Three-phase stator currents.

swift dynamic response, characterized by minimal overshoot during acceleration, along with a negligible steady-state error when responding to speed variations. This combination of rapid responsiveness and stability underscores the effectiveness of the proposed approach in managing the PMSM's operational performance.

The final case comprehensively evaluates the system's performance under a sudden torque disturbance. In this scenario, the load torque increases from 4.51 Nm to 9.02 Nm in 0.3 s, then decreases back to 4.51 Nm after 0.5 s, while the rotor speed remains at a constant rate of 3000 rpm. As shown in Fig. 9, the q-axis current reacts quickly and settles well to the torque step, whereas the d-axis current remains close to its 0-A setpoint with only slight oscillations. Following the torque change, the q-axis current takes approximately 0.7 ms to reach the desired reference value of 4.95 A. Notably, the electromagnetic torque generated by the proposed method stabilizes within around 0.4 seconds, demonstrating a quick and effective response to the disturbance. In a different scenario, when the load torque is reduced from a maximum of 9.02 Nm down to 4.51 Nm, the q-axis stator current stabilizes within just 0.5 ms. The motor speed shows a slight deviation of about 0.04% from its initial value, decreasing during the first second and then increasing in the following seconds. After these transient effects, the speed quickly stabilizes and reaches the desired setpoint. This rapid stabilization highlights the system's ability to adapt to varying load conditions, as summarized in Table 1, which presents a summary of the control performance achieved by the proposed strategy.

To evaluate the behavior of the proposed control method under parameter variations, simulation tests were conducted by reducing R_s and L_s by 50% at the initial state

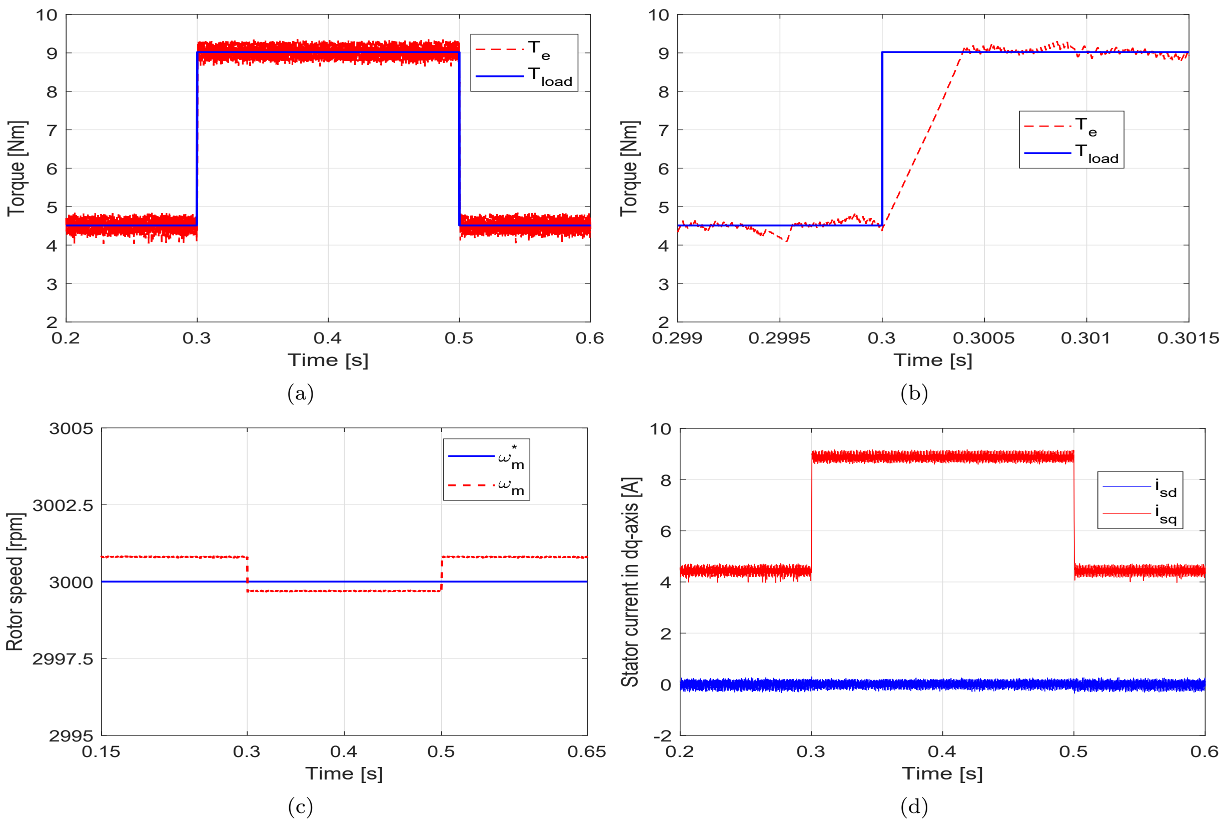

Fig. 9. Evaluation of the proposed method under sudden load changes: (a) Torque, (b) Zoom of torque, (c) Rotor speed and (d) Stator current in dq-axis.

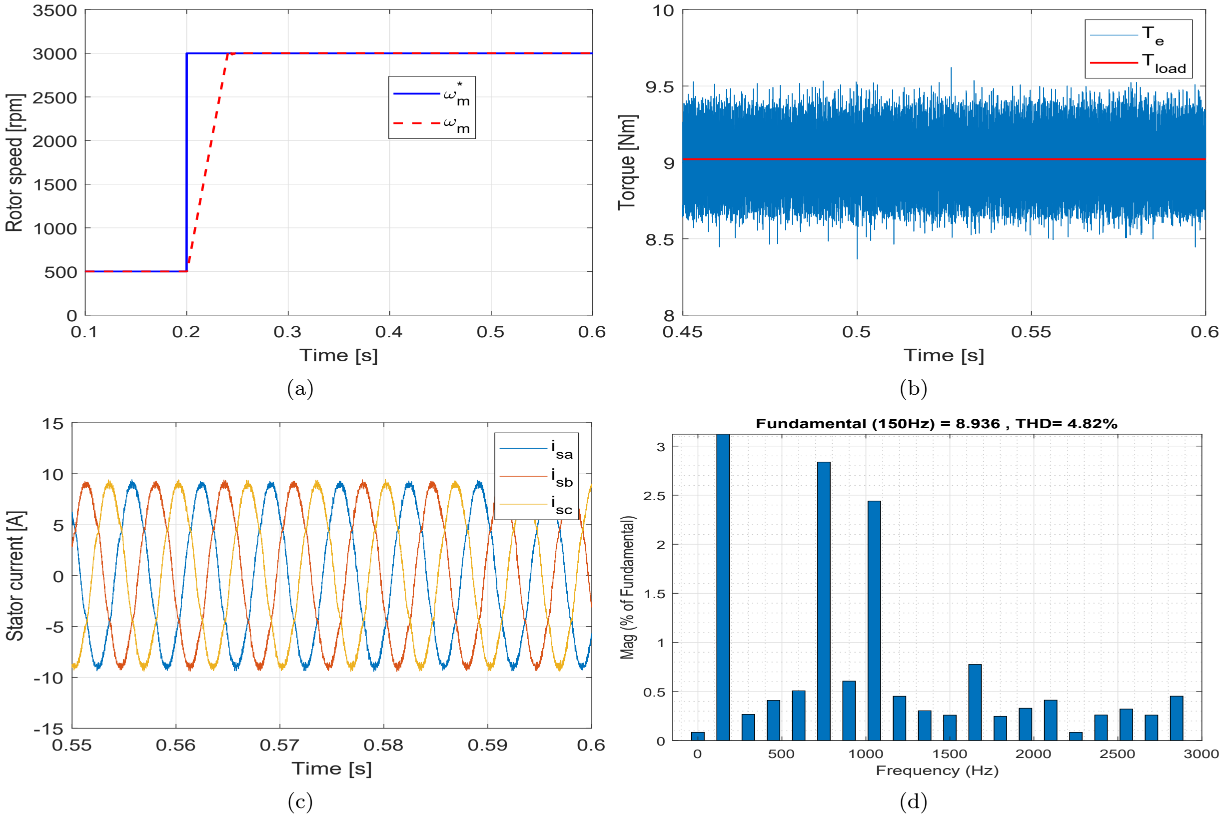

Fig. 10. Performance evaluation of the proposed method under parameter variations: (a) Speed response, (b) Torque, (c) Three-phase stator current and (d) FFT of the stator current.

Table 1. Summary of control performance.

Rotor speed transition	$\omega_m^* = 1000 \rightarrow 2000$ (rpm)
Rise time (ms)	23
Settling time (ms)	25
Overshoot (%)	0.78
MAE of d-axis stator current (A)	0.29
MAE of q-axis stator current (A)	0.33
MAE of rotor speed (rpm)	1.4
MAE of torque (Nm)	0.28
THD of stator current (%)	1.5

and the flux linkage (ψ_f) by 20% at 0.5 s. The system initially operated under no-load conditions, and a load torque of 9.03 Nm was applied at 0.4 s. Meanwhile, the rotor speed was increased from 500 rpm to 3000 rpm at $t = 0.2$ s.The results are presented in Fig. 10. Analysis of Fig. 10 indicates that the motor speed closely follows the reference trajectory, exhibiting a fast dynamic response and effective rejection of disturbances. Moreover, the proposed method maintains a near-sinusoidal current waveform; although the THD increases, it remains within the IEEE standard limit of 5%.

6. CONCLUSION

This paper has presented an FPGA-based co-simulation framework for finite-control-set model predictive current control (FCS-MPCC) applied to PMSM drives, integrating MATLAB/Simulink with ModelSim. By combining system-level modeling, algorithm development, and automated testbench generation in Simulink with cycle-accurate RTL simulation in ModelSim, the proposed framework provides a methodological workflow that bridges the gap between high-level simulation and hardware deployment.

Simulation studies under steady-state and transient conditions confirm the feasibility and correctness of the FCS-MPCC controller, demonstrating fast dynamic response and effective control of current and torque dynamics. The co-simulation framework facilitates early-stage functional verification, detection of implementation mismatches, and consistency checking between the control algorithm and FPGA implementation before hardware deployment.

Overall, the results show that the proposed methodology offers a reliable and practical workflow for developing predictive current controllers in high-performance PMSM drive applications. While this work validates the feasibility of FCS-MPCC for real-time control, future research could focus on quantitative comparisons with alternative verification methods, further enhancing its practical relevance for industrial applications.

REFERENCES

Brosch, A., Wallscheid, O., and Böcker, J. (2021). Model Predictive Control of Permanent Magnet Synchronous Motors in the Overmodulation Region Including Six-Step Operation. *IEEE Open Journal of Industry Applications*, 2, 47–63.

Gang, L. (2025). Pmsm control based on finite set predictive current model regulation. *Journal of Control Engineering and Applied Informatics*, 27(2), 3–13.

Hongfei Yang, Tianyu Gu, Y.Z.D.W. (2024). An improved fcs-mpc strategy for pmsm based on grey prediction and fuzzy dynamic weighting factors. *Journal of Control Engineering and Applied Informatics*, 26(4), 27–39.

Li, T., Sun, X., Lei, G., Guo, Y., Yang, Z., and Zhu, J. (2022). Finite-Control-Set Model Predictive Control of Permanent Magnet Synchronous Motor Drive Systems-An Overview. *IEEE/CAA Journal of Automatica Sinica*, 9(12), 2087–2105.

Ma, Z., Saeidi, S., and Kennel, R. (2014). FPGA Implementation of Model Predictive Control With Constant Switching Frequency for PMSM Drives. *IEEE Transactions on Industrial Informatics*, 10(4), 2055–2063.

Morel, F., Lin-Shi, X., Retif, J.M., Allard, B., and Buttay, C. (2009). A Comparative Study of Predictive Current Control Schemes for a Permanent-Magnet Synchronous Machine Drive. *IEEE Transactions on Industrial Electronics*, 56(7), 2715–2728.

Morel, F., Lin-Shi, X., Rétif, J.M., and Allard, B. (2008). A predictive current control applied to a permanent magnet synchronous machine, comparison with a classical direct torque control. *Electric Power Systems Research*, 78(8), 1437–1447.

Mwasilu, F., Kim, E.K., Rafaq, M.S., and Jung, J.W. (2018). Finite-Set Model Predictive Control Scheme With an Optimal Switching Voltage Vector Technique for High-Performance IPMSM Drive Applications. *IEEE Transactions on Industrial Informatics*, 14(9), 3840–3848.

Peng, J. and Yao, M. (2023). Overview of Predictive Control Technology for Permanent Magnet Synchronous Motor Systems. *Applied Sciences*, 13(10).

Petkar, S.G., Eshwar, K., and Thippiripati, V.K. (2021). A Modified Model Predictive Current Control of Permanent Magnet Synchronous Motor Drive. *IEEE Transactions on Industrial Electronics*, 68(2), 1025–1034.

Purraji, M., Zamiri, E., and de Castro, A. (2025). Easy and Straightforward FPGA Implementation of Model Predictive Control Using HDL Coder. *Electronics*, 14(3).

Quang, N. and Phuc, H. (2026). Fpga-based sensorless pmsm drive with sliding mode observer for velocity tracking. *IEEE Access*, 14, 39687–39701.

Quang, N., Van-Quang-Binh, N., Anh, N., Than, H., Dong, T., and Tho, N. (2022). Neural network pid controller for pmsm drives. 146–149.

Saeidi, S. (2015). *FPGA-Based Nonlinear Model Predictive Control of Electric Drives*. Phd thesis, Universite de Cergy-Pontoise.

Sandre-Hernandez, O., Rangel-Magdaleno, J., and Morales-Caporal, R. (2018). A Comparison on Finite-Set Model Predictive Torque Control Schemes for PMSMs. *IEEE Transactions on Power Electronics*, 33(10), 8838–8847.

Sathyamoorthy, S. (2022). *Finite Control Set-Model Predictive control of Permanent Magnet Synchronous Motor*. Master's thesis, KTH Royal Institute of Technology.

Siami, M., Khaburi, D.A., Abbaszadeh, A., and Rodríguez, J. (2016). Robustness Improvement of Predictive Current Control Using Prediction Error Correction for Permanent-Magnet Synchronous Machines. *IEEE Transactions on Industrial Electronics*, 63(6), 3458–3466.

Wang, F., Li, S., Mei, X., Xie, W., Rodríguez, J., and Kennel, R.M. (2015). Model-Based Predictive Direct Control Strategies for Electrical Drives: An Experimental Evaluation of PTC and PCC Methods. *IEEE Transactions on Industrial Informatics*, 11(3), 671–681.

Wang, F., Mei, X., Rodriguez, J., and Kennel, R. (2017). Model predictive control for electrical drive systems-an overview. *CES Transactions on Electrical Machines and Systems*, 1(3), 219–230.

Wendel, S., Dietz, A., and Kennel, R. (2017). Fpga based finite-set model predictive current control for small PMSM drives with efficient resource streaming. In *2017 IEEE International Symposium on Predictive Control of Electrical Drives and Power Electronics (PRECEDE)*, 66–71.

Yang, C., Liu, K., Hu, M., and Hua, W. (2023). Fpga-Based Extended Control Set Model Predictive Current Control with a Simplified Search Strategy for Permanent Magnet Synchronous Motor. *Electronics*, 12(23).