(Received 17 January 2026, Accepted 10 February 2026, Scheduled 28 February 2026)

A Delay-Compensated Predictive Current Control for PMSM Using a Luenberger Observer

Xuchen Wang* and Chenxuan Zhu

School of Electrical Engineering, Shanghai Dianji University, Shanghai 201306, China

ABSTRACT: This paper proposes a Luenberger-observer-assisted deadbeat predictive current control (LO-DPCC) scheme to compensate the inherent one-sample sampling/computation/PWM delay in embedded PMSM drives. A discrete-time Luenberger observer is designed for the dq-axis current dynamics to provide a one-step-ahead current estimate, which is embedded into a closed-form deadbeat predictive control law under a unified timing configuration. The method is evaluated by MATLAB/Simulink co-simulation using multi-wheel steering actuator profiles (front wheel independent + rear wheel cooperative) and by DSP-based bench experiments at 10 kHz PWM. Compared with a tuned MPC-FOC baseline and an ESO-assisted DPCC benchmark under identical constraints, LO-DPCC consistently improves speed regulation and torque smoothness, indicating that the observer-based one-step prediction is an effective and implementation friendly approach for delay-compensated predictive current control of PMSM drives.

1. INTRODUCTION

Permanent-magnet synchronous motors (PMSMs) are increasingly adopted as traction and actuator machines because they offer high power density, high efficiency, and fast electromagnetic response. In advanced chassis architectures — typified by four-wheel independent steering (4WIS) — the current loop is no longer a "motor-only" detail: current tracking and ripple translate directly into torque smoothness, which then propagates to lateral stability, path tracking, and ride comfort. When commands switch quickly or road friction changes, even a small phase lag or oscillation at the actuator layer can force the upper-level vehicle controller to compensate for the dynamics it did not intend to manage.

Model-predictive-control (MPC) strategies combined with field-oriented control (FOC) are therefore widely used to balance fast transients and low steady-state error by explicitly selecting voltage commands from predicted current/torque evolution [1–5]. Yet in automotive embedded drives, sampling, computation, and pulse width modulation (PWM) update introduce an inherent one-sample latency, so the measured current and applied voltage are temporally misaligned. That mismatch appears as an extra phase delay in the current loop, increasing current/torque ripple and degrading speed regulation; under time-varying loads, it can also leak into vehicle-level tracking metrics [2–4, 6–8]. In practical high-bandwidth designs, the latency is often not a secondary implementation artifact: even with an accurate model, "one beat late" voltage action can shift the reachable current sample by one PWM period, making a nominal deadbeat target effectively unattainable unless timing is explicitly realigned.

One line emphasizes robustness, injecting estimated disturbances or uncertainty terms into the prediction model or control law (e.g., observer-based compensation) to reduce sensitivity to parameter mismatch and load variation. For instance, sliding mode observers (SMOs) [9] and virtual voltage vectors [10] have been employed to enhance the robustness of MPC against disturbances. The limitation is that delay is frequently treated as an external correction rather than being absorbed into the predictive structure itself [6, 8, 11–14]. Another line modifies the prediction horizon or introduces delay-augmented models; performance can improve, but tuning and implementation complexity typically rise. Recently, data-driven and artificial intelligence approaches have also been explored to optimize weighting factors or prediction models [15–17]. However, these methods often impose a heavy computational burden on embedded processors. It leaves a practical question: can delay be handled with a minimal, interpretable mechanism that keeps the closed-form simplicity of deadbeat predictive current control while restoring true timing consistency?

Several gaps remain when the target is actuator-layer deployment for 4WIS-oriented applications. Existing solutions often present observer design, disturbance rejection, and delay treatment as loosely coupled modules, which weakens reproducibility and interpretability in real-time implementation [6, 8–12]. Detailed, implementation-level timing descriptions — how PWM frequency, sampling instant, and computation delay jointly determine whether the predictive loop behaves as intended — are still limited [2, 11, 12, 18]. More importantly, evidence chains that connect current-loop improvements to torque ripple, speed regulation, and vehicle-level tracking in representative 4WIS scenarios remain insufficient [19, 20]. If actuator-layer control is used in safety-relevant chassis systems, it must improve performance and remain explainable under embedded constraints.

* Corresponding author: Xuchen Wang (chen_xu_wang_w@sina.com).

Luenberger observers, with deterministic pole placement and low computational cost, provide a direct route to a one-step-ahead state estimate that can be aligned with the inherent digital delay [21, 22]. Furthermore, recent studies have demonstrated their robustness in sensorless control and parameter estimation applications [23]. Motivated by this, this paper proposes a discrete Luenberger-observer-assisted deadbeat predictive current control (LO-DPCC) for PMSM drives, where the one-sample delay is compensated by embedding a one-step-ahead current prediction into a unified discrete state-space formulation. The approach preserves the closed-form nature of deadbeat control and yields a small set of interpretable tuning parameters, then it is evaluated in MATLAB/Simulink co-simulation and validated on a 10 kHz PWM digital signal processing (DSP) platform, with the intent to show how timing-consistent current control improvements translate into torque smoothness and, ultimately, vehicle-level tracking performance.

2. VEHICLE AND PMSM MODELING

To study how the proposed control method performs under 4WIS-oriented operating conditions, we use models of both the vehicle lateral dynamics and permanent-magnet synchronous motor (PMSM). The vehicle module adopts a compact bicycle model driven by an equivalent steering input δ_{eq} and a steering-allocation layer consistent with the co-simulation platform (front-wheel independent + rear-wheel cooperative). These models form the foundation for the observer design and predictive current-control strategy introduced later.

2.1. Vehicle Dynamics Model

To represent the vehicle's motion, we adopt a simplified "bicycle model". Despite being compact, this model captures the main relationship between steering input and the vehicle's lateral and yaw motions. Let the vehicle-level state vector be

$$x_v = [e_y, e_\omega, \beta]^{\mathrm{T}} \tag{1}$$

where e_y denotes the lateral offset from the reference path, e_ω the heading-angle (yaw-angle) error, and β represents the sideslip angle at the vehicle's center of mass. The control input is chosen as the equivalent steering input δ_{eq} used by the bicycle model. In the co-simulation implementation, the route-tracking controller outputs wheel-level commands with front-wheel independent steering (δ_{fl}, δ_{fr}) and rear-wheel cooperative steering ($\delta_{rl} = \delta_{rr} = \delta_r$). A steering-allocation block then converts these commands into δ_{eq} for the bicycle-model dynamics.

For clarity, we define the front-axle equivalent steering angle as $\delta_f,\, k = (\delta_f l, k + \delta_{fr},\, k)/2$. Under a small-angle kinematic approximation, the equivalent bicycle-model steering input is $\delta_{eq},\, k \approx \delta_f,\, k - \delta_r,\, k$, where $\delta_r,\, k = \delta_{rl},\, k = \delta_{rr},\, k$ for the cooperative rear-steering setting used in this work.

Remark (steering-actuation configuration). The MATLAB/Simulink platform implements a simplified multi-wheel steering configuration: front-wheel independent steering and rear-wheel cooperative steering. This configuration is deliberately chosen to retain the dominant influence of rear steering on lateral-yaw response while avoiding unnecessary kinematic complexity associated with a fully independent 4-wheel steering geometry. Therefore, it provides a consistent and implementation-friendly environment for verifying the proposed actuator-layer control algorithm and its impact on path-tracking metrics.

TABLE 1. Bicycle-model parameters used in co-simulation.

Parameter	Symbol	Value (Unit)
Vehicle mass	m	1200 (kg)
Yaw moment of inertia	I_z	2000 (kg·m^2)
CG to front axle distance	l_f	1.05 (m)
CG to rear axle distance	l_r	1.35 (m)
Front axle cornering stiffness	C_f	60000 (N/rad)
Rear axle cornering stiffness	C_r	65000 (N/rad)
Longitudinal speed (S5 co-simulation)	v_x	12 (m/s)

The measured vehicle quantities include the longitudinal velocity v_x, yaw rate r, and lateral acceleration a_y. Key physical parameters of the bicycle model include the vehicle mass m, yaw moment of inertia I_z, distances from the center of gravity to the front and rear axles l_f and l_r, and the corresponding tire cornering stiffnesses C_f and C_r. The numerical values used in the co-simulation are summarized in Table 1.

The lateral-yaw dynamics can be written in discrete-time form as

$$\begin{aligned} x_{v,k+1} &= f_v\left(x_{v,k}, \delta_{eq,k}, v_{x,k}\right) + w_k, z_k \\ &= h_v\left(x_{v,k}, \delta_{eq,k}, v_{x,k}\right) + v_k \end{aligned} \tag{2}$$

where $f_v(\cdot)$ and $h_v(\cdot)$ denote the nonlinear vehicle dynamics and measurement mappings, respectively. The term w_k represents the process noise, and v_k represents the measurement noise; both are modeled as zero-mean Gaussian disturbances with appropriate covariance matrices.

This simplified yet standard bicycle model captures the principal lateral-yaw behaviour of the 4WIS electric vehicle while it remains compact enough for controller design and analysis. It provides the vehicle-level states used later to assess path-tracking and stability performance. In the numerical implementation, the bicycle model is discretised and coupled with the PMSM drive model of Section 2.2 to form the co-simulation environment shown in Figure 6, where the motor torque and speed generated by the current controller excite the vehicle dynamics.

2.2. Mathematical Model of PMSM

In the rotor-synchronous d-q reference frame, the stator-voltage equations and the electromagnetic-torque equation of a surface-mounted PMSM (SPMSM) can be expressed as:

$$\begin{aligned} u_d &= R_s i_d - \omega_e L_q i_q + L_d \frac{di_d}{dt} \\ u_q &= R_s i_q + \omega_e \psi_f + L_q \frac{di_q}{dt} - \omega_e L_d i_d \\ T_e &= \frac{3}{2} p \psi_f i_q \end{aligned} \tag{3}$$

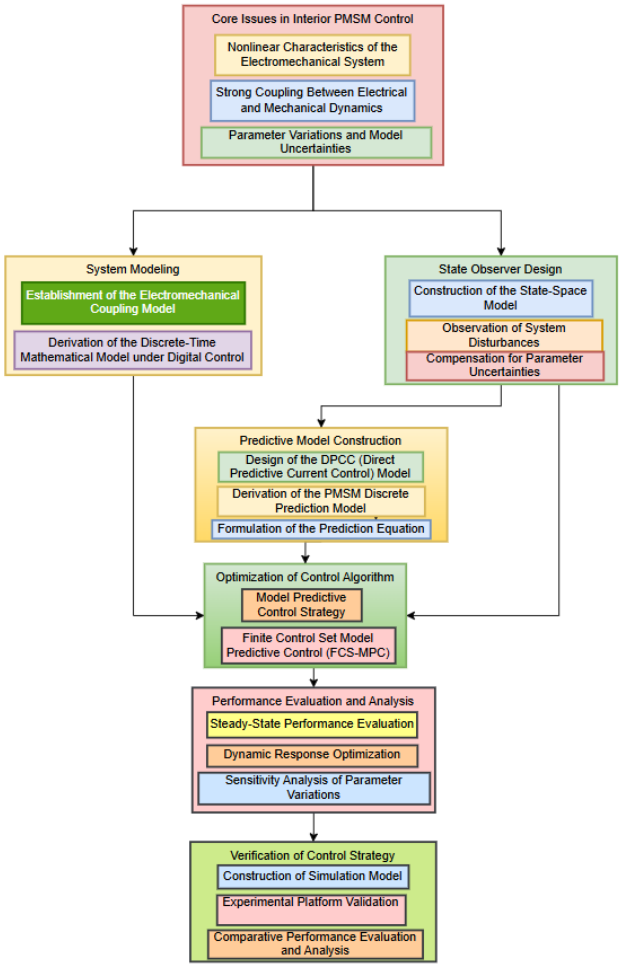

FIGURE 1. Technical roadmap of the proposed method.

where i_d and i_q are the d- and q-axis stator currents; R_s is the stator resistance; L_d and L_q are the d- and q-axis inductances; ω_e is the electrical angular velocity; ψ_f is the permanent-magnet flux linkage (back-EMF constant) of the surface-mounted PMSM (SPMSM); and p is the number of pole pairs.

Although the tested machine is categorized as an SPMSM, and the reluctance-torque contribution is negligible for the torque-current mapping used by the speed-to-current reference generation, we still formulate the electrical dynamics in the dq frame. For the adopted commercial servo motor, the datasheet reports a single-phase inductance, and offline identification indicates that the d/q inductances are approximately equal; therefore, we set $L_d = L_q = 0.59$ mH in Table 4 for both simulation and bench tests. Accordingly, the torque reference generation in Section 4.1 neglects the reluctance-torque term proportional to $(L_d - L_q)idiq$ as a standard SPMSM approximation.

To ensure the accuracy of the adopted PMSM model, all electrical and mechanical parameters — such as pole-pair number and flux linkage — were selected in accordance with standard benchmark PMSM configurations reported in the literature [6, 7] and then matched to the laboratory motor used in the experiments. In particular, a commercial permanent-magnet synchronous servo motor (model 42JSF630AS-1000) was adopted on the test bench, and its nameplate and datasheet constants were first taken as nominal values and subsequently fine-tuned via offline identification tests. The resulting parameter set is listed in Table 4 and is used consistently in both simulations and experiments so that simulated steady-state and dynamic responses remain in close agreement with the measured behaviour, thereby supporting the physical validity of the model employed in this work.

Discretizing the continuous-time dq current dynamics in (3) with a first-order forward-Euler scheme using the current-loop sampling period T_s (10 kHz PWM with center-aligned ADC sampling, $T_s = 100\,\mu$s) yields the predictive current model used in the controller derivation:

$$i\,(k+1) = A_d(\omega_e(k))i\,(k) + B_d u\,(k) + d\,(k) \qquad (4)$$

where $i(k) = [i_d(k), i_q(k)]^{\mathrm{T}}$, $u(k) = [u_d(k), u_q(k)]^{\mathrm{T}}$, $A_d(\omega_e) = I + A(\omega_e)T_s$, and $B_d = BT_s$. For notational consistency with Section 3.3, the Euler implementable matrices are also denoted by $\Phi_a(\omega_e) := A_d(\omega_e)$ and $\Gamma_a := B_d$. The term $d(k)$ collects lumped equivalent electrical disturbances and modelling uncertainties (including those induced by load-torque variations through speed/EMF terms). For readability, the explicit dependence on $\omega_e(k)$ is omitted when it is clear from the context. These discrete-time equations form the foundation for the observer-based predictive-control law developed later and ensure numerical consistency and reproducibility under the unified timing adopted throughout this study.

Figure cross-references. The above models provide the basis for the subsequent controller and observer derivations, see Figure 4 for the overall control block diagram and Figure 5 for the MATLAB/Simulink implementation. The key performance outcomes — rotational-speed response, torque ripple, and step-path tracking — are illustrated in Figures 7–10.

3. LUENBERGER OBSERVER DESIGN AND DISCRETIZATION

3.1. Methodological Workflow

To clarify the design concept and implementation process of the proposed LO-based DPCC strategy with delay compensation, Figure 1 illustrates the complete technical roadmap from problem modeling to simulation verification.

Figure 1 summarizes the design chain of LO-DPCC under a unified timing configuration. First, we formulate the one-sample sampling/computation/PWM delay in the discrete PMSM current loop and clarify why it deteriorates current tracking. In the delay-augmented plant model (Section 4.3), the current $i(k+1)$ depends on the voltage $u(k-1)$ computed one step earlier, which introduces phase lag in a conventional deadbeat controller.

Second, we design a scheduled Luenberger observer that provides a one-step-ahead current estimate $\hat{i}(k+1|k)$. Third, we embed this lead estimate into the deadbeat predictive control law so that the computed voltage $u(k)$, which first affects the current over $(k+1) \to (k+2)$, drives the reachable-step prediction $\hat{i}(k+2|k)$ toward the reference $i^*(k+2)$, thereby com-

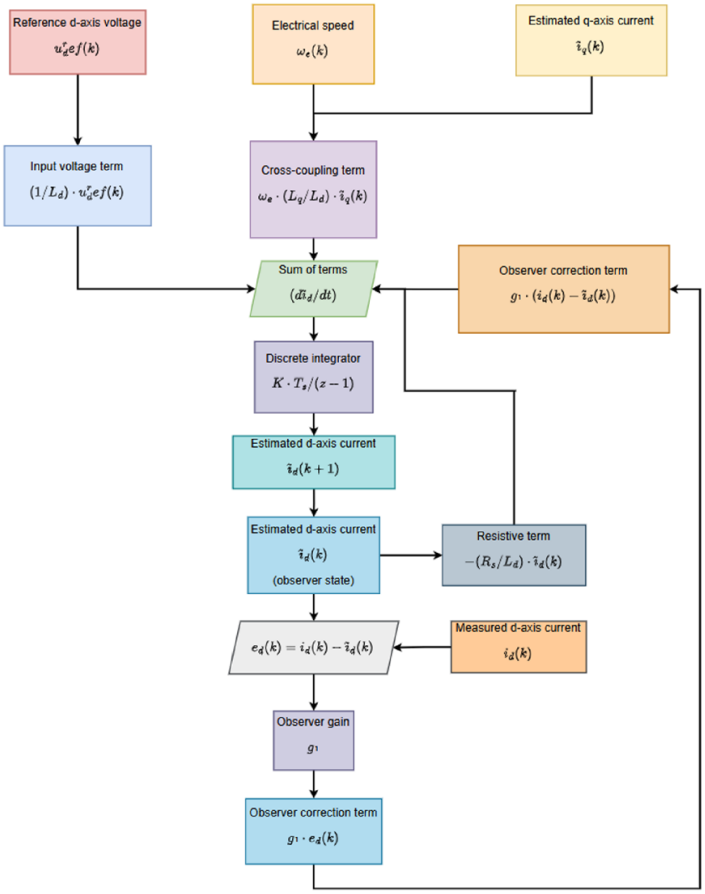

FIGURE 2. Structure of the d-axis current observer branch (q-axis branch analogous and omitted for clarity).

pensating for the inherent one-sample delay and recovering the intended current-loop dynamics.

To provide a timing-consistent state for delay compensation, the discrete-time Luenberger observer is formulated into two distinct steps within each sampling interval T_s. Upon the arrival of the measurement y_k at instant k, a correction step is first performed to obtain the current corrected estimate:

$$\hat{x}_{k|k} = \hat{x}_{k|k-1} + L_k \left(y_k - C\hat{x}_{k|k-1}\right) \tag{5a}$$

where $\hat{x}_{k|k-1}$ is the prior prediction computed in the previous interrupt, and L_k is the observer gain scheduled by the measured electrical speed $\omega_e(k)$. Subsequently, the one-step-ahead prediction $\hat{x}_{k+1|k}$ is computed using the plant model and voltage u_{k-1} applied during the interval $(k-1) \to k$:

$$\hat{x}_{k+1|k} = A_k \hat{x}_{k|k} + B u_{k-1} \tag{5b}$$

This two-step recursive structure explicitly matches the digital observe-predict-execute cycle: the correction in (5a) synchronizes the estimate with the sampled measurement at k, while the prediction in (5b) provides a lead-phase state for reachable-step current prediction under the inherent one-sample compute-to-actuation delay.

Discrete pole placement and scheduled $L(k)$. To make the observer several times faster than the current loop while limiting noise amplification, we specify a discrete-time observer pole via $z_o = \exp(-2\pi f_o T_s)$, where f_o is the desired observer bandwidth in Hz. With $T_s = 100\,\mu s$ and $f_o = 500\,Hz$, $z_o \approx \exp(-0.314) \approx 0.73$. Since the dq currents are directly measured, $C = I_2$, we choose $L(k)$ such that the nominal error map becomes isotropic:

$$A(k) - L(k)C = z_o I_2 \Rightarrow L(k) = A(k) - z_o I_2 \tag{6}$$

This yields $e(k+1) = z_o e(k)$ in the nominal case, i.e., both discrete observer poles are located at $z = z_o$ for all scheduled operating points. The scheduling variable is the measured electrical speed $\omega_e(k) = p\omega_m(k)$, and $A(k)$ (hence $L(k)$) is updated accordingly.

Figure 2 summarizes the signal flow of the d-axis branch of the current observer corresponding to (5)–(6). Similarly, the discrete-time implementation of the q-axis observer branch is illustrated in Figure 3, which shares an analogous structure but utilizes q-axis specific parameters.

3.2. Exact Discretization and Gain Scheduling

With current-loop sampling under a 10 kHz PWM carrier (center-aligned ADC, $T_s = 100\,\mu s$), we derive the exact discrete PMSM current model for analysis and tuning

$$x_i(k+1) = \Phi(\omega_e)\, x_i(k) + \Gamma(\omega_e)\, u(k) \tag{7}$$

where $\Phi(\omega_e) = e^{A(\omega_e)T_s}$ and $\Gamma(\omega_e) = \int_0^{T_s} e^{A(\omega_e)\tau} d\tau B$. For embedded realization, we implement the observer time update using the first-order (forward-Euler) approximations

$$\Phi_a(\omega_e) = I + A(\omega_e)T_s, \quad \Gamma_a = BT_s, \tag{8}$$

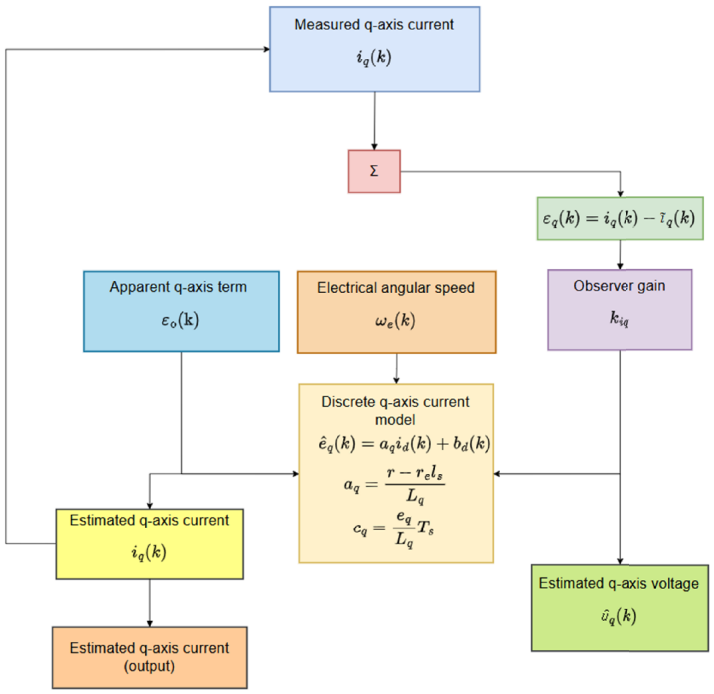

FIGURE 3. Discrete-time implementation of the q-axis current observer branch in MATLAB/Simulink (d-axis branch implemented in the same way).

to reduce arithmetic load. Accordingly, in the observer recursion (5), we set

$$A_d(\omega_e) = \Phi_a(\omega_e), \quad B_d = \Gamma_a \tag{9}$$

so that the implemented observer and controller use the same Euler-discretized current model. Throughout the paper, $\Phi(\omega_e)$ and $\Gamma(\omega_e)$ denote the exact discretization (matrix exponential and its integral) used for analysis and tuning, whereas $\Phi_a(\omega_e)$ and $\Gamma_a(\omega_e)$ denote the implementable approximation used in the digital signal processing (DSP) code. Under bounded speed variation and small T_s, the resulting discretization mismatch is $O(T_s^2)$ and is treated as part of the discretization/scheduling perturbation term in Section 3.4.

3.3. Uniform Exponential Stability under Assumptions A1--A2 and a Common Quadratic Lyapunov Matrix

We provide a verifiable sufficient condition showing that uniform exponential stability (UES) holds when Assumptions A1–A2 are satisfied, and there exists a common quadratic Lyapunov matrix P (independent of the electrical speed) for the scheduled family of error dynamics:

Assumption A1 (bounded operating set and slow variation). The electrical speed satisfies $\omega_e \in [\omega_{\min}, \omega_{\max}]$ with known bounds, and its variation between consecutive sampling instants is bounded.

Assumption A2 (scheduled pole placement). The scheduled gain $L(\omega_e)$ places the eigenvalues of $A_c(\omega_e)$ strictly inside the unit circle with margin $\rho < 1$ for all $\omega_e \in [\omega_{\min}, \omega_{\max}]$.

Stability interpretation under the scheduled pole placement. With $C = I_2$ and scheduled gain selection $L(\omega_e) = A_c(\omega_e) - z_o I_2$, the nominal observer error dynamics are reduced to

$$e(k+1) = z_o I_2 e(k) \tag{10}$$

for all scheduled operating points. Hence, the nominal observer is uniformly exponentially stable provided $|z_o| < 1$. In implementation, discretization mismatch, within-step speed variation, and measurement noise are grouped into an additive bounded perturbation $w(k)$, yielding

$$e(k+1) = z_o e(k) + w(k), \quad \|w(k)\| \le \bar{w} \tag{11}$$

This directly implies an input-to-state stability (ISS) bound

$$\|e(k)\| \le |z_o|^k \|e(0)\| + \frac{1 - |z_o|^k}{1 - |z_o|} \bar{w} \tag{12}$$

which explains the practical tuning rule adopted in this paper: choosing f_o several times larger than the current-loop bandwidth makes $|z_o|$ sufficiently small for fast convergence while maintaining robustness to sensing noise. This stability bound is also consistent with the use of $\hat{x}(k+1)$ as a one-step-ahead estimate in the reachable-step prediction in Section 4.

3.4. Practical Tuning and Timing Consistency

A unified timing configuration is adopted throughout this work: the PWM carrier frequency is 10 kHz; the current loop and the LO are updated every $T_s = 100\,\mu$s; and the speed loop runs at 1 kHz with period $T_{s,\omega} = 1\,\text{ms} = 10T_s$. It yields a fixed one-sample compute-to-actuation latency in the current loop, which is explicitly compensated in Section 4. The observer bandwidth is chosen as 500 Hz, i.e., about 3–5 times the current-loop bandwidth (125 Hz), so as to accelerate error decay without noticeable noise amplification. All controller and observer parame-

ters are kept identical across the simulation scenarios to ensure a fair and reproducible comparison.

Scheduling range of ω_e. The observer gain $L(k)$ is scheduled by the measured electrical speed $\omega_e(k) = p\omega_m(k)$. The schedule is designed over $\omega_e \in [0, \omega_{e,\text{rated}}]$, where $\omega_{e,\text{rated}} = p \cdot 2\pi(\frac{n_{\text{rated}}}{60})$. With $p = 4$ and $n_{\text{rated}} = 3000$ r/min (Table 4), $\omega_{e,\text{rated}} \approx 1257$ rad/s. The reported bench tests in S1–S4 use a speed command of 800 r/min, corresponding to $\omega_e \approx 335$ rad/s, which lies well within the scheduled range.

4. DEADBEAT PREDICTIVE CURRENT CONTROL WITH ONE-SAMPLE DELAY COMPENSATION

This section introduces a deadbeat predictive current control (DPCC) scheme that explicitly compensates the one-sample compute-to-actuation delay inherent in digital PWM drives. The core idea is to retain DPCC's closed-form, "deadbeat-like" voltage solution while recovering the correct current-loop timing by injecting a one-step-ahead current estimate provided by a discrete Luenberger observer (LO) — a practical choice in high-bandwidth PMSM drives, where even a single-sample delay can noticeably erode phase margin and amplify the ripple. Bold symbols denote dq-axis vectors, with $i(k) = [i_d(k)\ \ i_q(k)]^{\mathrm{T}}$ and $u(k) = [u_d(k)\ \ u_q(k)]^{\mathrm{T}}$. The current controller runs at the sampling period T_s, whereas the PWM update imposes a fixed one-sample delay (as formalized in Section 4.5.1): the voltage computed at instant k is applied in the next PWM period and thus affects the current sample at $k+2$. With this timing in mind, the delay-compensated control law is constructed so that the reachable current $i(k+2)$ tracks the reference $i^*(k+2)$, aligning the "target" with what the hardware can actually realize in the subsequent cycle.

4.1. Reference Generation: Speed PI → Torque → Current

The outer speed loop generates the electromagnetic torque reference for the inner current control loop. The rotor-speed tracking error is defined as

$$e_\omega(k) = \omega^*(k) - \omega(k) \tag{13}$$

where $\omega^*(k)$ and $\omega(k)$ denote the reference and measured rotor speed, respectively.

A discrete position-form proportional-integral (PI) controller is employed in the speed loop, obtained by forward-Euler discretization of a continuous-time PI regulator. The discrete integrator state is updated as

$$x_\omega(k) = x_\omega(k-1) + T_{s,\omega} e_\omega(k) \tag{14}$$

and the electromagnetic torque command is given by

$$T_e^*(k) = \text{sat}\left(K_p e_\omega(k) + K_i x_\omega(k)\right) \tag{15}$$

where K_p and K_i are the proportional and integral gains of the speed controller; $T_{s,\omega}$ is the speed-loop sampling period; and sat$(\cdot)$ denotes the torque (or current) saturation imposed by actuator limits.

Anti-windup is implemented by back-calculation, such that the integrator state is corrected according to

$$x_\omega(k) \leftarrow x_\omega(k) + \frac{T_{s,\omega}}{\tau_{a\omega}}\left(T^*_{e,\text{sat}}(k) - T^*_e(k)\right) \tag{16}$$

where τ_{aw} is the anti-windup time constant, and $T^*_{e,\text{sat}}(k)$ is the saturated torque command. For the surface-mounted PMSM (SPMSM) considered in this work, the reluctance torque is negligible, and the torque-current relationship below base speed is

$$T_e^*(k) = \frac{3}{2} p\psi_f i_q^*(k), \quad i_d^*(k) = 0 \tag{17}$$

which yields the q-axis current reference $i_q^*(k)$ subject to the current-circle constraint. The speed-loop PI generates the torque and current references at the slow sampling rate $T_{s,\omega}$. These references are held constant over the fast current-control loop and are fully consistent with the delay-compensated predictive current control formulated in Sections 4.3–4.5.

4.2. Analytical DPCC Law under the Delay-Free Discrete Model

We first summarize the closed-form DPCC voltage computation using the delay-free discrete PMSM current model

$$i(k+1) = Ai(k) + Bu(k) + d(k) \tag{18}$$

where $A := A_d(\omega_e(k))$ and $B := B_d$ are the Euler-discretized system matrices defined in (4), and $d(k)$ collects lumped disturbances and modeling errors.

Strict deadbeat (one-step) objective. If the voltage $u(k)$ could be applied immediately (i.e., without digital delay), the strict deadbeat target would be $i(k+1) = i^*(k+1)$, which yields the one-step closed-form DPCC law.

$$u(k) = B^{-1}\left(i^*(k+1) - Ai(k) - d(k)\right) \tag{19}$$

where B^{-1} denotes a suitable right inverse of B. In practice, $d(k)$ is unknown and is either neglected in the nominal law or compensated by a disturbance estimator (e.g., the extended state observer (ESO) benchmark in later sections).

Pole-placement DPCC (deadbeat-like) generalization. To avoid excessive sensitivity to noise and saturation-induced chattering, DPCC is often formulated with a desired discrete-time error dynamic:

$$e(k+1) = \Lambda e(k), \quad e(k) = i^*(k) - i(k) \tag{20}$$

where $\Lambda = \text{diag}(\lambda_d, and \lambda_q)$ places the closed-loop poles inside the unit circle. Enforcing this on the delay-free model leads to a closed-form law of the same computational form as above, with $\Lambda = 0$ reduced to strict deadbeat.

Importantly, the above derivations describe the ideal timing case. In embedded drives, however, $u(k)$ is not applied immediately; instead, it is applied one PWM period later. Therefore, the quantity that can be made deadbeat in practice is not $i(k+1)$, but the first current sample is actually affected by $u(k)$, namely $i(k+2)$. This motivates the delay-compensated formulation in the next subsection.

Implementation of B^{-1}. With forward-Euler discretization of the dq current dynamics, the input matrix is diagonal,

$B = \text{diag}\left(\frac{T_s}{L_d}, \frac{T_s}{L_q}\right)$, hence B is nonsingular for $L_d, L_q > 0$. In this work, B^{-1} is implemented as the true inverse

$$B^{-1} = \text{diag}\left(\frac{L_d}{T_s}, \frac{L_q}{T_s}\right) \tag{21}$$

which is reduced to a scalar division when $L_d \approx L_q$. Voltage saturation is handled after the closed-form computation: the unconstrained $u(k)$ is first computed by the delay-compensated law, and then clipped to the admissible space vector pulse width modulation (SVPWM) voltage set determined by the DC-bus limit. When saturation occurs, the strict equality at the reachable step cannot be enforced; nevertheless, the proposed delay compensation still preserves correct timing alignment (i.e., $u(k)$ targets the first current sample it can influence), which empirically reduces overshoot and ripple compared with uncompensated designs under the same constraints.

4.3. One-Sample Delay Compensation via LO Prediction

Under the unified observe-compute-update timing, the dq currents are sampled at instant k, and the controller computes $u(k)$ within the same interrupt, but the PWM hardware applies this voltage in the next PWM period. Consequently, the current loop seen by the controller exhibits a one-sample input delay: $i(k+1) = Ai(k) + Bu(k-1) + d(k)$ where $A := A_d(\omega_e(k))$ and $B := B_d$. The crucial implication is: $u(k)$ does not determine $i(k+1)$; $u(k)$ first becomes effective over $(k+1) \to (k+2)$, and therefore determines $i(k+2)$. To make this explicit, propagate one more step:

$$i\,(k+2) = Ai\,(k+1) + Bu\,(k) + d\,(k+1) \tag{22}$$

Observer-based delay compensation. At time k, $i(k+1)$ is not yet measured, but it is predictable because the actually applied voltage during $(k-1) \to k$ is $u(k-1)$. Consistent with the two-step Luenberger observer timing in Section 3.1 — where the measurement update yields a corrected estimate $\hat{x}_{k|k}$ in (5a), and the model-based prediction yields $\hat{x}_{k+1|k}$ in (5b) — the discrete observer provides a filtered current estimate $\hat{i}(k|k)$ after the measurement update at time k and, in particular, the one-step-ahead prediction:

$$\hat{i}\,(k+1|k) = A\hat{i}\,(k|k) + Bu\,(k-1) \tag{23}$$

Using this prediction, the reachable-step current at $k+2$ (i.e., the first sampled current affected by $u(k)$) can be predicted as:

$$\hat{i}\,(k+2|k) = A\hat{i}(k+1|k) + Bu\,(k) \tag{24}$$

Delay-compensated strict deadbeat law (targets $k+2$). Enforcing $\hat{i}(k+2|k) = i^*(k+2)$ yields

$$u\,(k) = B^{-1}\left(i^*\,(k+2) - A\hat{i}\,(k+1|k)\right) \tag{25}$$

This equation makes the timing unambiguous: the voltage computed at k is chosen so that the current at $k+2$ — the first sample affected by $u(k)$ — matches the reference.

Delay-compensated pole-placement DPCC (deadbeat-like, targets $k+2$). Define the predicted tracking error at the reachable instants as

$$\hat{e}\,(k+1|k) \;=\; i^*\,(k+1) - \hat{i}\,(k+1|k)\,,$$

$$\hat{e}\,(k+2|k) \;=\; i^*\,(k+2) - \hat{i}\,(k+2|k) \tag{26}$$

Setting $\Lambda = 0$ recovers the strict deadbeat case. With $0 < |\lambda_d|, |\lambda_q| < 1$, the current response becomes aperiodic and well-damped, improving robustness to measurement noise and saturation.

From an implementation perspective, the proposed LO-DPCC avoids online optimization and multistep iterative prediction. Each interrupt cycle involves only a few matrix-vector multiplications and additions, making it compatible with a $T_s = 100\,\mu\text{s}$ current-loop update on standard DSP platforms.

4.4. Modulation, Saturation, and Implementation

The commanded dq voltages $u(k) = [u_d(k)\ u_q(k)]^{\text{T}}$ are transformed to the stationary frame via the inverse Park transform to obtain (u_α, u_β), which are then applied through an SVPWM module.

Voltage saturation. The voltage vector is constrained by the DC-bus limit and modulation strategy: $\|u(k)\| \le u_{\max}$, where $u_{\max}$ is determined by V_{dc} (e.g., for linear SVPWM, the admissible fundamental voltage magnitude is bounded). When saturation occurs, the strict deadbeat equality at $k+2$ cannot be guaranteed; nevertheless, the delay compensation still provides correct phase alignment and typically reduces overshoot and ripple relative to uncompensated designs.

Current limiting and anti-windup consistency. The current reference $i^*(k)$ is always pre-saturated by the current circle limit, and the speed-loop PI uses backcalculation anti-windup so that the outer loop remains consistent with the inner-loop constraints.

Real-time execution order (one interrupt). A typical implementation follows:

1. sample phase currents and compute $i(k)$ (Clarke/Park);
2. perform LO measurement update to obtain $\hat{i}(k|k)$;
3. compute $\hat{i}(k+1|k)$ using $u(k-1)$;
4. form $i^*(k+2)$ (held constant over the fast loop or scheduled by the outer loop/field weakening);
5. compute $u(k)$ using the boxed delay-compensated law above;
6. apply voltage saturation, inverse Park, and SVPWM update (effective in the next PWM period).

This observe-predict-execute flow is exactly what ensures that $u(k)$ is matched to the correct reachable current $i(k+2)$, thereby compensating for the one-sample compute-to-actuation latency.

4.5. Discrete-Time PMSM Current-Loop Design and Stability

This subsection summarizes the discrete-time delay model, the pole-placement tuning rule, and the closed-loop stability interpretation of the observer-assisted delay-compensated current loop.

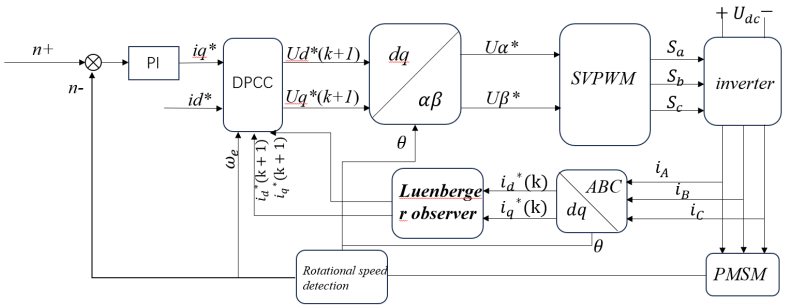

FIGURE 4. Block diagram of the LO-DPCC PMSM drive. The measured phase currents are transformed into dq-axis currents for feedback, and the superscript "*" denotes the reference quantities generated by the speed/field-weakening loop. The discrete Luenberger observer provides a one-step-ahead current estimate for delay compensation in the predictive current-control law.

4.5.1. Discrete-Time Current Dynamics with One-Sample Delay

After discretization in the dq frame with sampling period T_s, the nominal delay-free current dynamics can be written as

$$i\left(k+1\right)=Ai\left(k\right)+Bu\left(k\right)+d\left(k\right) \tag{27}$$

Timing configuration and delay mechanism. In the adopted digital implementation, the voltage vector computed during the kth interrupt is written into PWM compare registers but becomes effective in the next PWM period. This produces an inherent one-sample compute-to-actuation delay, and the input-output relation observed at the sampling instants is

$$i\left(k+1\right)=Ai\left(k\right)+Bu\left(k-1\right)+d\left(k\right) \tag{28}$$

Therefore, the first current sample directly influenced by $u(k)$ is $i(k+2)$, not $i(k+1)$. This is the fundamental reason that a DPCC law derived for the delay-free model must be modified for the embedded use.

4.5.2. Pole Placement and Closed-Form Delay-Compensated DPCC Law

Let $i^*(k)$ be the reference generated by Section 4.1 and define the reachable-step tracking error as:

$$e\left(k+2\right)=i^*\left(k+2\right)-i\left(k+2\right) \tag{29}$$

A first-order discrete-time pole-placement specification is imposed over the reachable step via the predicted tracking error:

$$\hat{e}\left(k+2\right)=\Lambda e\left(k+1\right),\ \Lambda=\mathrm{diag}\left(\lambda_d,\lambda_q\right),\ \left|\lambda_d\right|,\left|\lambda_q\right|<1 \tag{30}$$

A practical bandwidth-matching choice is

$$\lambda_d=\lambda_q=\exp\left(-2\pi f_{bw}T_s\right) \tag{31}$$

where f_{bw} is the desired current-loop bandwidth. This selection yields a monotonic, well-damped response and avoids the noise amplification often associated with strict deadbeat ($\Lambda=0$).

Substituting the LO-based one-step-ahead prediction $\hat{i}(k+1|k)$ into the reachable-step model $\hat{i}(k+2\mid k)=A\hat{i}(k+1|k)+Bu(k)$ leads to the delay-compensated closed-form law

$$u\left(k\right)=B^{-1}\left(i^*\left(k+2\right)-\Lambda\hat{e}\left(k+1|k\right)-A\hat{i}\left(k+1|k\right)\right) \tag{32}$$

which becomes strict deadbeat when $\Lambda=0$. This expression makes the controller's target explicit and temporally consistent: the control voltage computed at time k is designed so that the current at time $k+2$ tracks $i^*(k+2)$.

4.5.3. Stability of the Observer-Assisted Delay-Compensated Current Loop

The discrete Luenberger observer provides a corrected estimate $\hat{i}(k|k)$ and a one-step-ahead prediction $\hat{i}(k+1|k)$. With gain scheduling (w.r.t. electrical speed) and bounded discretization/scheduling perturbations, the observer error dynamics can be made uniformly exponentially stable and input-to-state stable with respect to bounded modeling errors.

When the delay-compensated DPCC law is applied, the reachable-step current error evolves according to the imposed pole-placement map (modulo bounded perturbations due to observer error, parameter mismatch, and saturation). Since the desired poles satisfy $|\lambda_d|,\ |\lambda_q|<1$, the nominal closed-loop error dynamics are contractive. With an ISS-bounded observer error and bounded disturbances, the interconnected observer-controller-plant loop is input-to-state stable, implying that the dq currents converge to a small neighborhood of their references. This stability interpretation is consistent with the observed reduction in delay-induced phase lag, faster transient recovery, and improved torque smoothness reported in the subsequent simulations and experiments.

5. SIMULATION AND EXPERIMENTAL VALIDATION

5.1. Simulation Setup

The simulation study is conducted in MATLAB/Simulink under a unified drive-and-control configuration to strictly validate the control logic before hardware deployment, a standard practice for evaluating motor drive performance [24]. The overall block diagram of the proposed LO-DPCC PMSM drive is shown in Figure 4, and its detailed MATLAB/Simulink implementation is provided in Figure 5. To evaluate the drive behavior under 4WIS-oriented operating conditions, the PMSM drive is further embedded into a route-tracking vehicle environment based on the bicycle dynamics introduced in Section 2.1; the resulting vehicle-drive co-simulation structure is depicted in Fig-

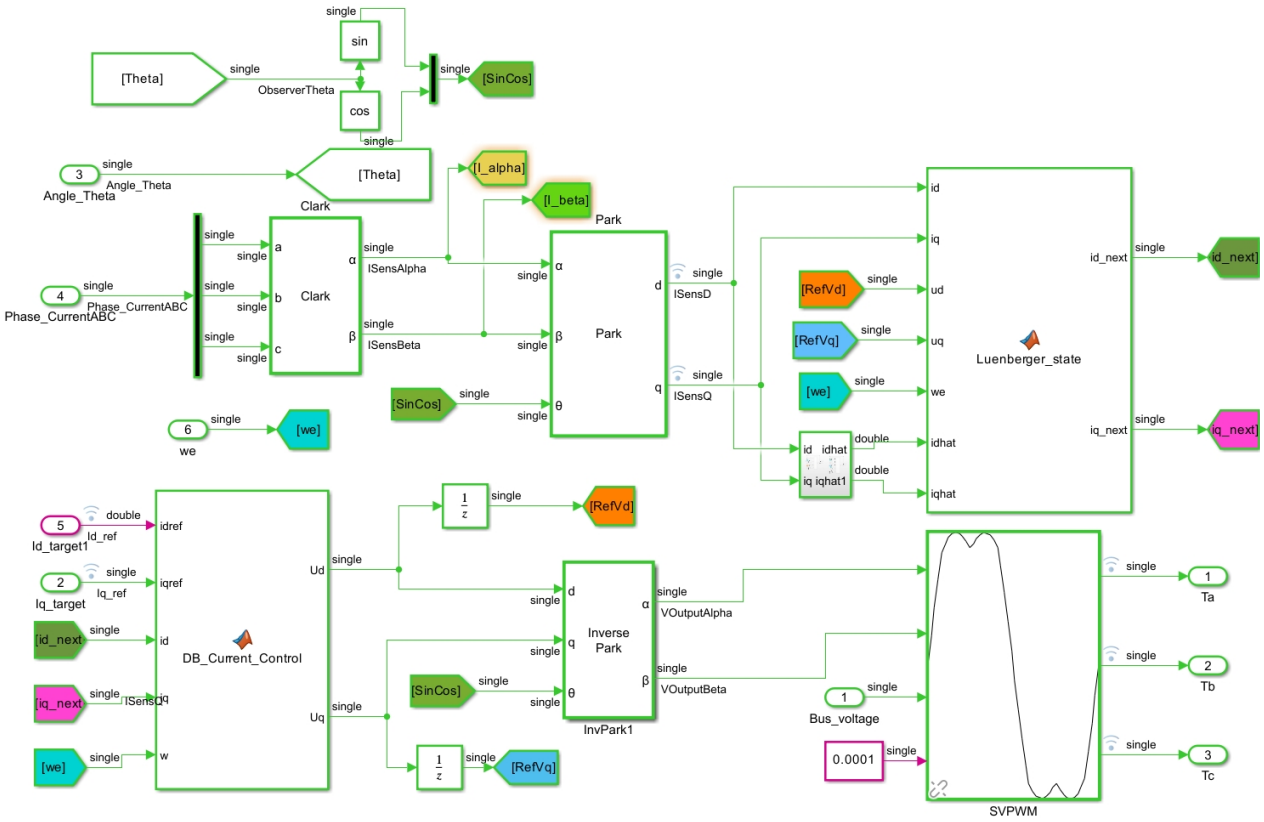

FIGURE 5. MATLAB/Simulink implementation of the LO-DPCC system.

TABLE 2. Parameters of the surface-mounted PMSM.

Nomenclature					
Symbol	Meaning	Unit	Symbol	Meaning	Unit
R_s	Stator resistance	Ω	$u_d u_q$	d/q-axis stator voltages	V
L_d, L_q	d/q-axis inductance	H	i_d, i_q	d/q-axis stator currents	A
ψ_f	PM flux linkage	Wb	ω_e	Electrical angular velocity	rad·s^{-1}
p	Pole pairs	-	f_o	Luenberger observer bandwidth	Hz
T_s	Current-loop/LO sampling period (10 kHz PWM; $T_s = 100$ µs)	s	f_i	Current-loop bandwidth	Hz
$T_{s,\omega}$	Speed-loop sampling period (1 kHz; $T_{s,\omega} = 1$ ms $= 10$ Ts)	s	C_f, C_r	Tire cornering stiffness	N/rad
$v_x v_y$	Vehicle longitudinal speed	m·s^{-1}	$u_d u_q$	d/q-axis stator voltages	V

ure 6 and described in Section 5.1.2. Throughout Section 5, all controllers use the same motor parameters (Table 4). To evaluate the performance of the proposed LO-DPCC scheme, a series of simulations and bench experiments were conducted. The detailed physical parameters of the PMSM and the control specifications of the proposed LO-DPCC system are summarized in Table 2. To verify the effectiveness of the delay-compensation algorithm, the sampling and switching frequency is set to 10 kHz, which is consistent with the discrete-time observer implementation requirements, the same sampling and PWM settings, and the same modulation policy, ensuring that performance differences can be attributed to the control design rather than to inconsistent implementation details.

5.1.1. Speed-Loop PI Parameters (Bandwidth-Matching Design)

The outer speed loop is tuned via bandwidth matching to ensure that its closed-loop dynamics remain compatible with the inner current loop. The continuous-time PI gains are selected as

$$K_p = \frac{2\zeta\omega_n J}{k_t}, \quad K_i = \frac{\omega_n^2 J}{k_t} \tag{33}$$

where J is the lumped inertia of the motor-load system; $k_t = \frac{3}{2}p\psi_f$ is the torque constant of the SPMSM under $i_d \approx 0$; $\zeta \in [0.6, 0.9]$ is the damping ratio (with $\zeta = 0.707$ used by default); and ω_n denotes the desired natural frequency of the speed loop. To avoid interaction with the inner current loop, ω_n is chosen

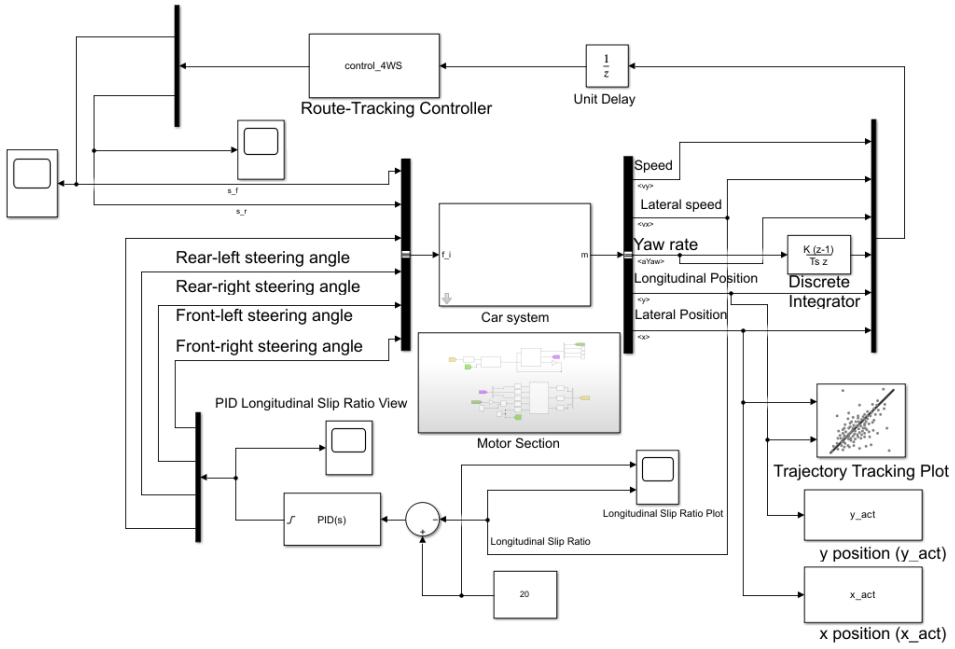

FIGURE 6. MATLAB/Simulink vehicle-drive co-simulation (LO-DPCC PMSM wheel actuator + bicycle-model dynamics + steering allocation).

by

$$\omega_n = \min\left(0.1\omega_{ci}, \frac{4}{T_{\text{settle}}}\right) \tag{34}$$

where ω_{ci} is the current-loop bandwidth (rad·s^{-1}), and T_{settle} is the targeted 2% settling time. Anti-windup is implemented by back-calculation with gain $\beta \in [0.3, 1.0]$. Once computed from the plant parameters in Table 4, the PI gains are fixed and reused in all subsequent scenarios to ensure a fair and reproducible comparison among control methods.

5.1.2. 4WIS Vehicle-Drive Co-Simulation Model

To emulate 4WIS-oriented driving conditions while keeping the focus on drive-level behavior, a vehicle-drive co-simulation model is built in MATLAB/Simulink, as sketched in Figure 6. The PMSM speed loop and LO-DPCC current controller (Figure 4) form the motor section, which outputs the electromagnetic torque and wheel speed for one wheel actuator. This wheel is coupled to a bicycle-model-based vehicle dynamics block representing the longitudinal velocity, lateral velocity, and yaw-rate states in Section 2.1. The route-tracking controller generates wheel-level steering commands with front-wheel independent steering $(\delta_{fl}, \delta_{fr})$ and rear-wheel cooperative steering $(\delta_{rl} = \delta_{rr} = \delta_r)$. A steering-allocation module maps these wheel-level commands to the equivalent steering input δ_{eq} required by the bicycle-model dynamics. The resulting longitudinal and lateral motions are then fed back to update the slip ratio and speed commands of the PMSM drive, thereby closing the vehicle-drive loop. In this way, the proposed strategy is exercised under representative path-tracking conditions with rear-steering assistance, while the remaining wheel actuators are assumed identical and are implicitly reflected in the vehicle parameters.

5.1.3. Test Scenarios

Five representative scenarios are used for evaluation, as summarized in Table 3. These scenarios cover: no-load acceleration (S1), load/torque-demand step excitation (S2), steady-state cornering with periodic torque-demand excitation (S3), dq-axis current tracking under step current-reference changes (S4), and a vehicle-level lateral path-step case (S5). All cases are first implemented in MATLAB/Simulink using the PMSM drive model, and S5 further embeds the drive into the bicycle-model-based lateral dynamics with 4WIS-oriented steering allocation described in Section 2. For hardware validation on the PMSM drive bench, the first four scenarios (S1–S4) are reproduced by applying speed/torque/current commands whose magnitudes and bandwidths resemble those of a single-wheel actuator in a 4WIS electric vehicle. By contrast, the path-step case S5 inherently involves vehicle lateral dynamics and tyre-road interaction; therefore, it is evaluated only in simulation and is treated as vehicle-level numerical evidence rather than as a full 4WIS experimental test.

Key settings. PWM carrier: 10 kHz; current-loop and LO update: $T_s = 100\,\mu$s with center-aligned ADC sampling; speed-loop update: 1 kHz. In simulation, a fixed-step solver of 10 µs is used, and the control signals are held with a Zero-Order Hold (ZOH) at $T_s = 100\,\mu$s. The delay model assumes a one-sample compute-to-actuation latency, i.e., the voltage computed at kT_s is applied during the interval $[(k+1)T_s, (k+2)T_s)$ and therefore first affects the sampled current at $(k+2)T_s$. Metrics — speed 2%-settling time, torque-ripple ratio, dq-axis current-tracking error (S4), and path-tracking overshoot (S5) — are computed as defined in Section 5.2.

The PMSM parameters used in simulation and experiments are taken from the nameplate/datasheet of the adopted commercial servo motor (42JSF630AS-1000) and refined by offline identification so that the simulated steady-state and transient

TABLE 3. Test scenarios and evaluation metrics.

Case	Description	Key parameters	Output metrics
S1	No-load acceleration	Commanded speed 800 r·min^{-1}; $T_L : 0 \rightarrow 0.15$ N·m at 0.2 s; speed command 800 r·min^{-1}	Settling time (s), overshoot (%)
S2	Equivalent torque-demand step (load effect; emulated on bench)		Speed dip (r·min^{-1}), regulation time (s)
S3	Cornering	Evaluation window 2–4 s; $T_{\text{mean}} \approx 0.12$ N·m; sinusoidal equivalent load ± 0.05 N·m with frequency $f_0 = 0.5$ Hz	Torque ripple (%), peak-to-peak torque (N·m), speed ripple (r·min^{-1})
S4	Current tracking	$i_d^* = 0$ A; i_q^* step changes at $t \approx 0.25$ s, 0.65 s, 1.10 s (Figure 15); speed command 800 r·min^{-1}	Peak error (A), steady-state error (A), settling time (s)
S5	Path step	Lateral step 0.2 m; bicycle-model tracker enabled	Overshoot (%), settling time (s)

TABLE 4. PMSM parameters used in simulation.

Parameter	Symbol	Value (Unit)	Parameter	Symbol	Value (Unit)
Motor model	-	42JSF630AS-1000	Back-EMF constant	K_e	4.3V_{rms}/krpm
DC bus voltage	V_{dc}	24 V	Rotor inertia	J	1.85×10^{-5} kg·m^2
Rated power	P_{rated}	62 W	Encoder resolution	-	1000 lines
Rated torque	T_{rated}	0.20 N·m	Pole pairs	p	4
Rated speed	n_{rated}	3000 r·min^{-1}	PWM frequency	f_{pwm}	10 kHz
Rated current	I_{rated}	4 A	Sampling period	T_s	100 µs
Stator resistance	R_s	1.02 Ω	Observer bandwidth	f_{LO}	500 Hz
d-axis inductance	L_d	0.59 mH	Current-loop bandwidth	f_i	125 Hz
q-axis inductance	L_q	0.59 mH	Speed-loop PI gains	K_p, K_i	3.6, 180
PM flux linkage	ψ_f	0.0083 Wb			

responses match the measured behaviour. In addition to the electrical constants (R_s, L_d, L_q, ψ_f, p), Table 4 also reports the rated power of the tested motor to make the operating point and scaling of the reported torque/speed profiles explicit and reproducible.

Choice of the current-loop bandwidth. The current loop is updated at $T_s = 100$ µs (10 kHz PWM), while the embedded implementation exhibits an inherent one-sample compute-to-actuation delay. We therefore select a conservative current-loop bandwidth to retain adequate phase margin under the delay and to avoid noise amplification from current sensing/quantization. Using a first-order discrete-time pole specification $z_i = \exp(-2\pi f_i T_s)$, the nominal choice $f_i = 125$ Hz corresponds to $z_i \approx \exp(-2\pi \cdot 125 \cdot 10^{-4}) \approx 0.924$, which yields a well-damped response while keeping the observer bandwidth (500 Hz) several times faster for reliable prediction. This bandwidth also preserves a practical separation from the outer speed loop (updated at 1 kHz and tuned to a much lower closed-loop bandwidth), thereby reducing loop interaction under saturation and delay.

5.1.4. Control Timing and Compensation Parameters

A unified timing configuration is adopted for all methods to ensure implementation-level consistency. The PWM carrier frequency is set to 10 kHz; the current loop and Luenberger observer (LO) are updated every $T_s = 100$ µs; and the speed loop runs at 1 kHz (see Table 4). Under this configuration, the voltage vector computed during the kth interrupt is written to the PWM update registers and becomes effective in the next PWM period, starting at $(k+1)T_s$. Consequently, the embedded current loop exhibits a fixed one-sample compute-to-actuation latency: the measurement at time k is used to compute $u(k)$, but $u(k)$ influences the reachable current sample at time $k+2$. Therefore, in the delay-augmented plant seen by the controller, the temporally consistent tracking objective is formulated over the reachable step, i.e., the control law is designed so that $i(k+2)$ tracks the reference $i^*(k+2)$.

The proposed LO-DPCC explicitly compensates for this delay by using the one-step-ahead observer prediction $\hat{i}(k+1|k)$ when forming the reachable-step prediction and computing the

voltage command, thereby aligning the digital implementation with the delay-augmented model and preserving the intended deadbeat-like response over the reachable step. All compared methods (MPC-FOC, ESO-DPCC, and LO-DPCC) are implemented under the same sampling/PWM constraints and modulation policy, so that comparisons reflect control-design differences rather than inconsistent timing or actuation.

5.1.5. MPC-FOC Baseline Controller

For benchmarking, a conventional cascaded field-oriented control (FOC) structure is implemented with an outer speed loop and an inner predictive current loop, using exactly the same PMSM model, sampling period, and timing configuration as the proposed LO-DPCC. The outer speed loop uses the same discrete PI regulator as in Section 5.1.1. With the speed error $e_\omega(k) = \omega^*(k) - \omega(k)$, the torque command is

$$T_e^*(k) = K_{p,\omega} e_\omega(k) + K_{i,\omega} \sum_{j=0}^{k} e_\omega(j) \tag{35}$$

with standard back-calculation anti-windup and saturation on T_e^* imposed by the current limit. For the surface-mounted PMSM considered here, below base speed, the electromagnetic torque satisfies

$$T_e(k) = \frac{3}{2} p \psi_f i_q(k) \tag{36}$$

and the q-axis current reference is obtained as

$$i_q^*(k) = \frac{2}{3} \frac{T_e^*(k)}{p\psi_f} \tag{37}$$

while the d-axis reference is set to $i_d^*(k) = 0$. Above base speed, the same field-weakening and voltage-ellipse constraints as in the proposed method are applied to compute (i_d^*, i_q^*) under current and DC-bus voltage limits.

The inner current loop is implemented as a voltage-vector-based model predictive current controller (VV-MPCC) in the rotor-synchronous dq frame. At each sampling instant k, the controller predicts the next-step current using the discrete PMSM model with sampling period T_s and treats the dq-axis voltage vector $u_{dq}(k) = [u_d(k), u_q(k)]^{\mathrm{T}}$ as a continuous decision variable: $\hat{i}_{dq}(k+1|k) = A i_{dq}(k) + B u_{dq}(k)$.

The optimal voltage vector is obtained by minimizing the quadratic cost:

$$J(k) = \left\| i_{dq}^*(k+1) - \hat{i}_{dq}(k+1|k) \right\|_2^2 + \lambda \left\| u_{dq}(k) \right\|_2^2 \tag{38}$$

where $i_{dq}^*(k+1)$ is generated by the speed/field-weakening loop, and $\lambda > 0$ penalizes voltage magnitude. With the one-step model above, the minimizer admits a closed-form expression:

$$u_{dq}^*(k) = \left(B^{\mathrm{T}} B + \lambda I\right)^{-1} B^{\mathrm{T}} \left(i_{dq}^*(k+1) - A i_{dq}(k)\right) \tag{39}$$

After the saturation under the DC-bus voltage constraint, the commanded voltage vector is mapped back to the stationary $\alpha\beta$ frame by the inverse Park transform and applied through the same SVPWM as in Section 4.4. This MPC-FOC baseline does not include explicit one-sample delay compensation: the measured currents at time k are used directly in prediction, while the computed voltage is still updated in the next PWM period; therefore, the inherent compute-to-actuation delay remains uncompensated.

For the extended-state observer deadbeat predictive current control (ESO-DPCC) benchmark in Sections 5 and 6, the same discrete-time DPCC law is combined with an extended-state observer that estimates lumped disturbances and modeling errors in the PMSM current dynamics, and the estimated disturbance is fed back to improve the current-tracking accuracy under parameter mismatch and load-torque variations. In contrast to LO-DPCC, ESO-DPCC does not perform forward state prediction to align the control law with the reachable step under the one-sample delay; thus, the digital delay is only partially mitigated, which explains why its overall performance typically lies between the uncompensated MPC-FOC baseline and the fully delay-compensated LO-DPCC in the subsequent tests.

5.2. Performance Metrics

5.2.1. Torque Computation and Unit Scaling

To avoid ambiguity in torque scaling on the low-power PMSM platform, all torque-related results refer to the electromagnetic torque calculated from the measured dq-axis currents using the SPMSM torque equation in Section 2.2. For the tested motor (Table 4), the rated torque is 0.20 N·m at 4 A, corresponding to a torque constant of approximately 0.050 N·m/A, which serves as a consistency check for the reported torque magnitude.

Torque ripple metric. Over a specified steady-state evaluation window $\mathcal{W} = [t_1, t_2]$ (e.g., 2–4 s in S3), the electromagnetic torque is computed from the measured dq currents using the SPMSM torque equation. The torque ripple ratio reported in this paper is defined as

$$r_T(\%) = \frac{T_{e,\max}(\mathcal{W}) - T_{e,\min}(\mathcal{W})}{\left|\overline{T}_e(\mathcal{W})\right|} \times 100\%,$$
$$\overline{T}_e(\mathcal{W}) = \frac{1}{N} \sum_{k \in \mathcal{W}} T_e(k) \tag{40}$$

where $T_{e,\max}$ and $T_{e,\min}$ are the maximum and minimum torque values within $\mathcal{W}$, and $\overline{T}_e$ is the window-average torque. This definition matches the percentage values reported in Figures 8 and 14 and allows direct comparison across controllers under identical operating conditions.

5.2.2. Comparative Performance Analysis under Dynamic Conditions

To validate the dynamic-performance benefits of the proposed LO-DPCC, comparative simulations were conducted against a conventional MPC-FOC baseline. The evaluation metrics include rotational-speed response, torque-ripple characteristics, and path tracking accuracy, thereby capturing improvements in transient response, steady-state precision, and speed-regulation

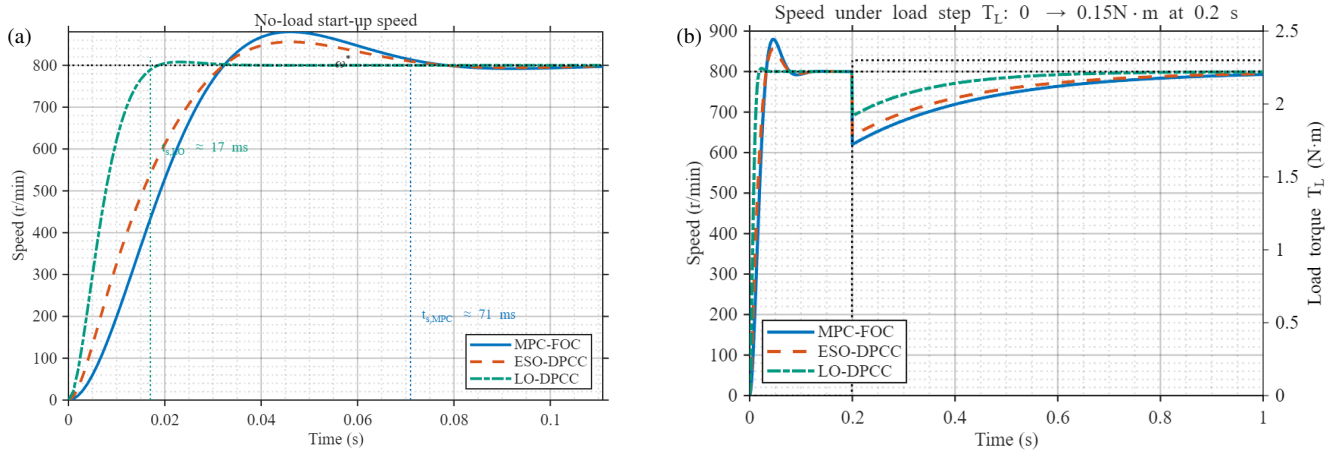

FIGURE 7. Simulated speed responses for the considered drive scenarios: (a) no-load start-up to 800 r·min^{-1}; (b) speed regulation under a 0.15 N·m load-torque step applied at $t = 0.2$ s, comparing MPC-FOC, ESO-DPCC, and LO-DPCC.

behavior under representative torque-demand excitations (including load-torque steps in simulation). These findings align with the recent reports on adaptive feed-forward and delay-compensated predictive current control, which corroborate the effectiveness of delay compensation at high speeds [25].

(1) Rotational-Speed Response (S1 and S2)

Figure 7(a) shows the simulated no-load start-up to 800 r·min^{-1}. The proposed LO-DPCC reaches the 2% settling band fastest ($t_s \approx 17$ ms) with negligible overshoot. The MPC-FOC baseline exhibits the slowest transient ($t_s \approx 71$ ms) and a more pronounced overshoot, while ESO-DPCC provides intermediate performance, indicating that deadbeat current regulation alone improves response speed but does not fully remove the delay-induced phase lag.

Figure 7(b) further illustrates the speed-regulation response to the applied 0.15 N·m load-torque step at $t = 0.2$ s in simulation. LO-DPCC yields a smaller speed dip and a faster return toward the 800 r·min^{-1} reference than MPC-FOC, while ESO-DPCC remains intermediate. This trend is consistent with the use of observer-based one-step-ahead current prediction, which alleviates the effective phase lag introduced by the one-sample sampling/computation/PWM update in the digital current loop.

(2) Torque-Ripple Characteristics (S3)

Torque smoothness in scenario S3 is assessed on the 2–4 s steady-state window under periodic equivalent load excitation. Figure 8 shows representative electromagnetic-torque segments for the three controllers; the normalized torque-ripple levels indicate that MPC-FOC exhibits the largest ripple (25.3%); ESO-DPCC reduces it (22.3%); and LO-DPCC yields the smallest ripple (15.3%). Figure 9 further visualizes the mean-removed torque component and its spectrum for LO-DPCC, where the dominant low-frequency component (∼0.5 Hz) and its harmonics are attenuated under delay-compensated prediction.

(3) Path-Tracking Performance (S5)

For the lateral step-path scenario S5 (Figure 10), all three controllers are embedded in the same bicycle-model-based

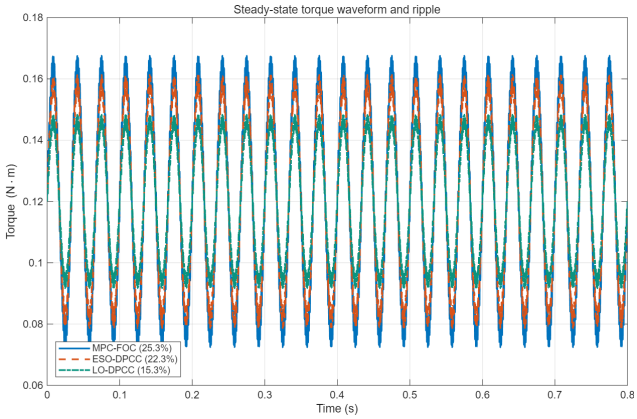

FIGURE 8. Simulated steady-state electromagnetic-torque waveforms in S3: representative segment of the 2–4 s evaluation window for MPC-FOC, ESO-DPCC, and LO-DPCC, with the torque-ripple ratio (%) reported.

tracker. The upper panel of Figure 10 compares the normalized path-tracking response over the first 2 s, while the lower panel shows the corresponding tracking error. MPC-FOC exhibits the largest overshoot (15.2%) and the highest transient error peak. ESO-DPCC improves damping and reduces the overshoot to 10.0%. LO-DPCC achieves the smallest overshoot (5.8%) and the lowest error peak with a faster decay, demonstrating that delay-compensated current prediction can improve vehicle-level tracking when the actuator dynamics are embedded in the path-tracking loop.

Taken together, the simulation results show a consistent hierarchy: ESO-DPCC outperforms MPC-FOC by enforcing deadbeat current tracking, while LO-DPCC further enhances dynamic response, torque smoothness, and path-tracking accuracy by explicitly compensating for the one-sample delay in the current loop through observer-based one-step-ahead prediction.

5.3. Experimental Bench Setup and Dynamic Performance

Figure 11 presents the DSP-based PMSM drive bench used for hardware validation. A TI TMS320F28379D controller and a

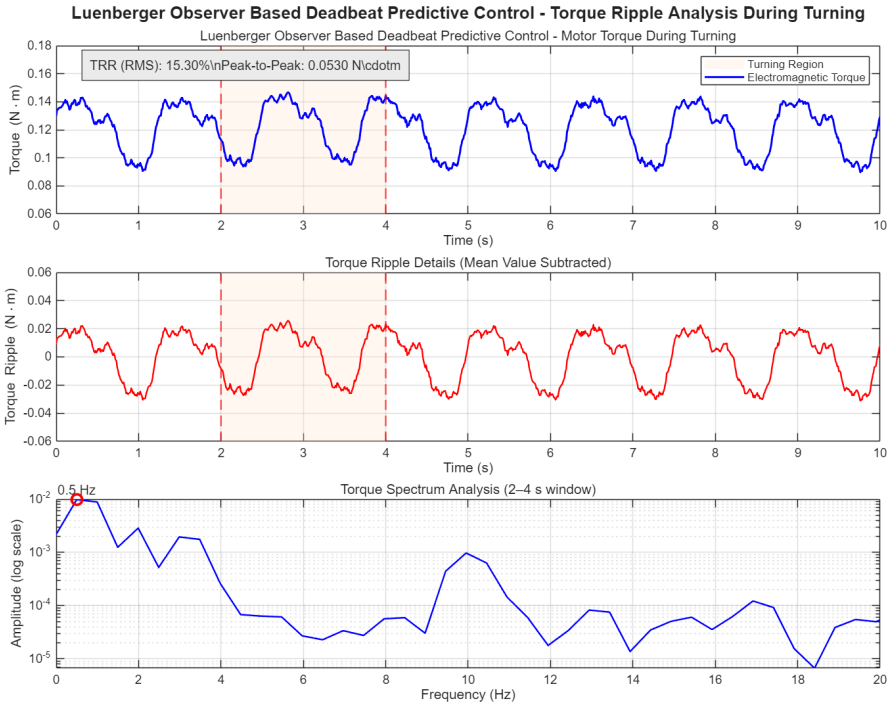

FIGURE 9. LO-DPCC torque-ripple analysis during steady-state cornering (S3): (a) electromagnetic torque with the 2–4 s analysis window highlighted, (b) mean-subtracted torque ripple, and (c) ripple spectrum within the 2–4 s window.

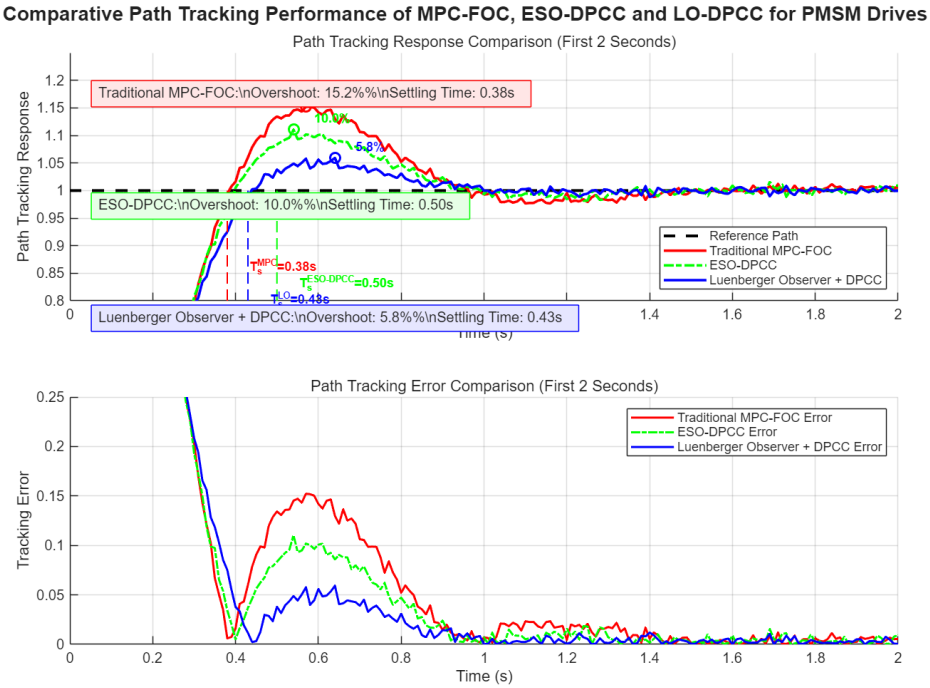

FIGURE 10. Simulated step-path tracking performance for a 0.2 m lateral-step command (S5) using the bicycle-model-based tracker: upper panel — path-tracking response (first 2 s); lower panel — tracking error (first 2 s) for MPC-FOC, ESO-DPCC, and LO-DPCC.

three-phase inverter drive the SPMSM (42JSF630AS-1000) at a 10 kHz PWM frequency; the current loop and observer run at 100 µs and the speed loop at 1 kHz. The three control strategies (MPC-FOC, ESO-DPCC, LO-DPCC) share the same limits and timing so that differences in the measured waveforms can be attributed to the current-control strategy.

To keep the bench tests repeatable, no external dynamometer is used. Hence, the load effect in S2–S3 is implemented as a programmable equivalent torque-demand disturbance under the same PWM/sampling timing for all controllers. In simulation, the disturbance is applied as a mechanical-side load torque $T_L(k)$ in the torque balance against the electromagnetic torque

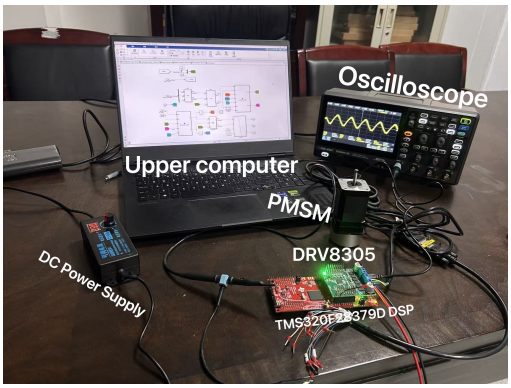

FIGURE 11. TI DSP-based PMSM drive bench for LO-DPCC implementation. A TMS320F28379D control board and a DRV8305EVM inverter drive a commercial PMSM (42JSF630AS-1000) under closed-loop speed and current control.

$T_e(k)$. On the DSP bench, the same profile is emulated by injecting an equivalent disturbance $\tau_d(k)$ in the torque-command path:

$$\tilde{T}^*(k) = \text{sat}\left(T^*(k) + T_d(k)\right) \tag{41}$$

where $T^*(k)$ is the speed-loop torque command, and $\text{sat}(\cdot)$ denotes the actuator saturation used consistently throughout the experiments. This approach provides a repeatable excitation for evaluating drive-layer regulation, but it does not fully replicate a physical mechanical-side load disturbance; therefore, S2–S3 results should be interpreted as the disturbance rejection under an emulated torque-demand profile.

Repeatability and uncertainty. All scenarios were run with identical timing (10 kHz PWM, $T_s = 100\,\mu$s updates and center-aligned current sampling) and identical sensing hardware. Each scenario was repeated $N = 10$ times after warm-up; reported metrics are given as mean $\pm$ standard deviation across runs. Speed is measured by a 1000-line incremental encoder, and torque-related metrics are computed from measured dq currents using the SPMSM torque equation, so sensor noise and ADC/encoder quantization are inherently reflected in the reported statistics.

Five scenarios (Table 3) are considered. S1–S4 are implemented on the bench; S5 is evaluated only in simulation as a vehicle-level step-path case.

S1: no-load acceleration to 800 r·min^{-1};

S2: equivalent 0.15 N·m torque-demand step at $t = 0.2$ s under an 800 r·min^{-1} speed command (implemented on the bench by injecting an equivalent torque-demand disturbance in the torque-command path);

S3: periodic equivalent load excitation around the rated operating point;

S4: current tracking (step current-reference transitions).

S5: Path step (vehicle-level co-simulation only).

5.3.1. No-Load Start-up (S1)

Figure 12 compares the measured no-load start-up to 800 r·min^{-1} (S1). LO-DPCC reaches the reference fastest and exhibits the smoothest torque transient; MPC-FOC is slower, and ESO-DPCC is intermediate.

5.3.2. Equivalent Torque-Demand Step Response (S2)

Figure 13 shows the response to an equivalent 0.15 N·m torque-demand step introduced at $t = 0.2$ s under an 800 r·min^{-1} speed command (S2). On the test bench, this "load effect" is emulated by injecting an equivalent disturbance in the torque-command path prior to the torque-to-current conversion, rather than by applying a physical load through a dynamometer.

5.3.3. Torque Ripple under Periodic Load (S3)

Under periodic equivalent load excitation (S3), Figure 14 compares the measured electromagnetic torque and its spectrum on the steady-state window. LO-DPCC reduces the dominant low-frequency ripple component and its harmonics relative to MPC-FOC and ESO-DPCC, yielding a visibly smoother torque waveform.

5.3.4. Current Tracking (S4)

Figure 15 reports the dq-axis current response in S4, where $i_d^* = 0$ and i_q^* is stepped at approximately $t \approx 0.25$ s, 0.65 s, and 1.10 s under the same timing constraints as the other bench tests (Table 3). The q-axis tracking error $e_q = i_q - i_q^*$ (raw signal used for metrics) is highlighted with zoomed windows around each step, and the settling time is evaluated by the first entry into — and subsequent residence within — the ± 0.10 A error band.

All three controllers regulate i_d close to zero and achieve small steady-state q-axis errors within the ± 0.10 A band; however, their transient recovery differs: LO-DPCC exhibits the fastest error decay and shortest settling after each step; ESO-DPCC is intermediate; and MPC-FOC shows the slowest return, consistent with its uncompensated one-sample compute-to-actuation delay. This direct current-loop evidence supports the speed-regulation observations in S1–S2, because the proposed LO-based one-step-ahead current prediction is explicitly introduced to absorb the one-sample delay and restore the intended timing alignment of the current loop.

5.4. Comprehensive Advantage Analysis

Table 5 summarizes the structural features and representative performance indices of the considered PMSM control strategies, including the MPC-FOC baseline, intermediate ESO-DPCC, representative observer-assisted schemes reported in the literature (e.g., LO-DMPC and LO-ADRC), and the proposed LO-DPCC. For the three in-house methods (MPC-FOC, ESO-DPCC, and LO-DPCC) implemented under the unified timing configuration of this work, the quantitative indices are reported using consistent metric definitions: the speed-response index is taken from the S1–S2 speed-regulation tests; the torque-ripple index is evaluated under the S3 periodic-load/cornering condition (with the normalized ripple metric used for cross-method comparison); and the overshoot index refers to the simulated vehicle-level step-path case S5. The

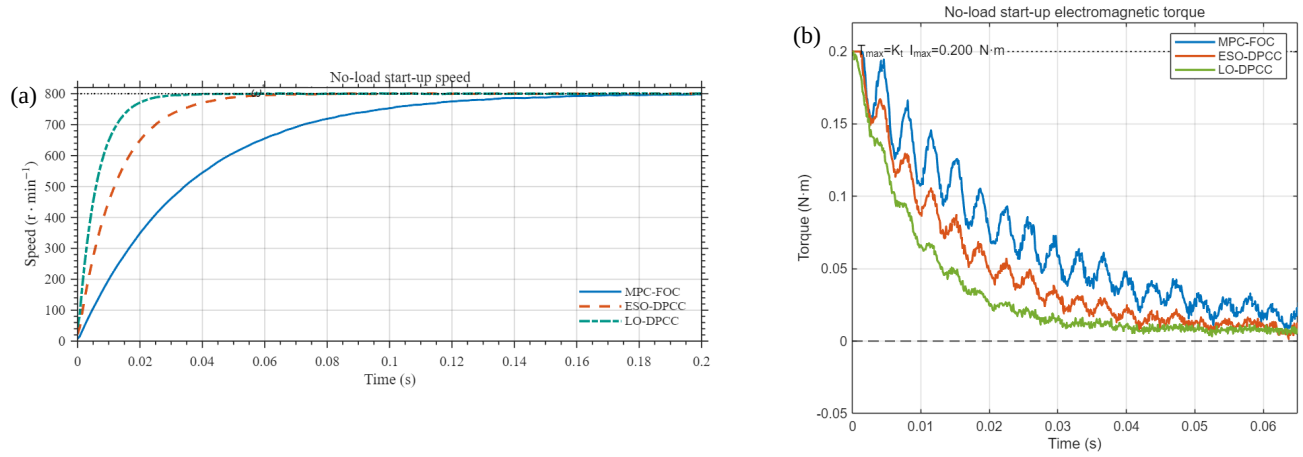

FIGURE 12. Experimental no-load start-up responses (S1) under an 800 r·min^{-1} speed command: (a) measured rotor speed; (b) electromagnetic torque for MPC-FOC, ESO-DPCC, and LO-DPCC.

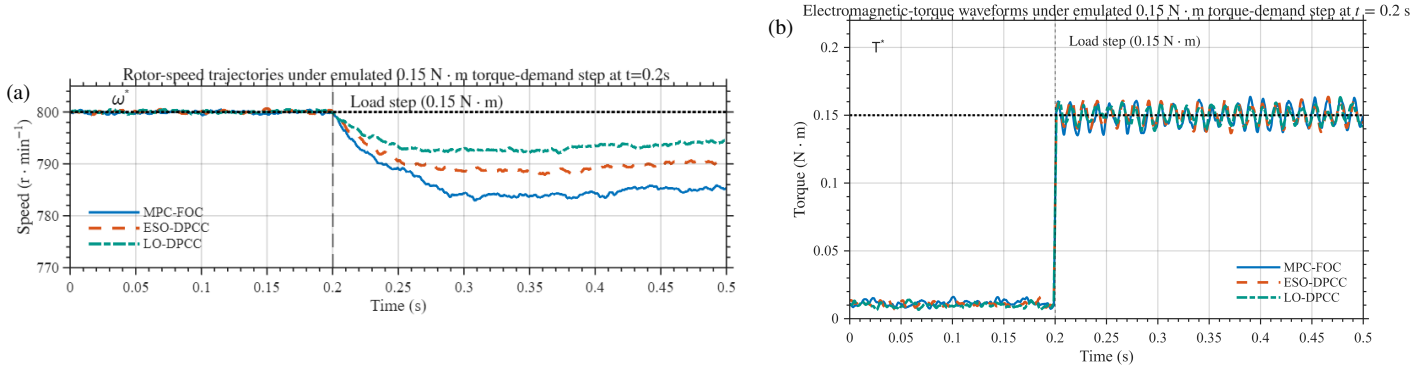

FIGURE 13. Experimental responses to an emulated 0.15 N·m equivalent torque-demand step at 0.2 s (S2) with an 800 r·min^{-1} speed command: (a) rotor-speed trajectories; (b) electromagnetic-torque waveforms for MPC-FOC, ESO-DPCC, and LO-DPCC.

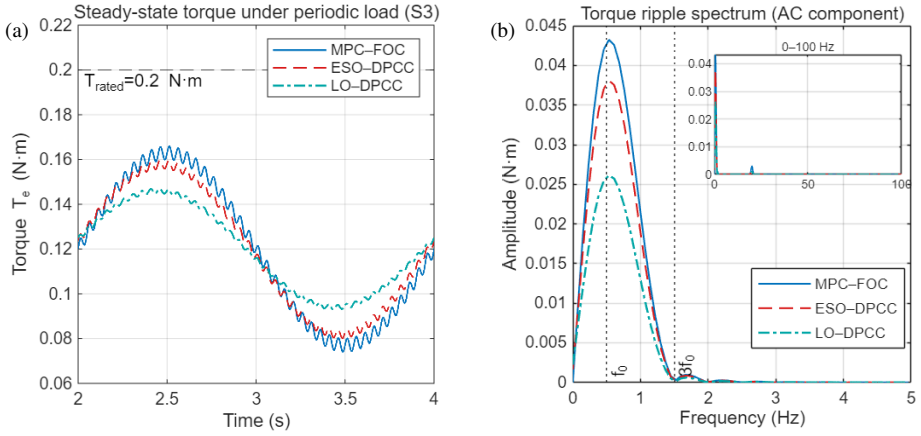

FIGURE 14. Experimental electromagnetic-torque comparison under periodic load excitation (S3): (a) time-domain torque waveforms; (b) frequency-domain spectra for MPC-FOC, ESO-DPCC, and LO-DPCC.

corresponding actuator-layer bench experiments (S1–S4) provide hardware evidence that corroborates the same performance ranking and comparable magnitudes at the same operating point, whereas S5 provides vehicle-level numerical evidence within the co-simulation environment, complementing the bench experiments (S1–S4).

Under the adopted PMSM and timing configuration, the proposed LO-DPCC achieves a fourfold reduction in 2% speed-settling time (from about 71 ms to about 17 ms) relative to the MPC-FOC baseline and a reduction in the torque-ripple ratio from roughly 25.3% to 15.3%. In the vehicle-level step-path scenario S5, the co-simulations additionally predict a decrease

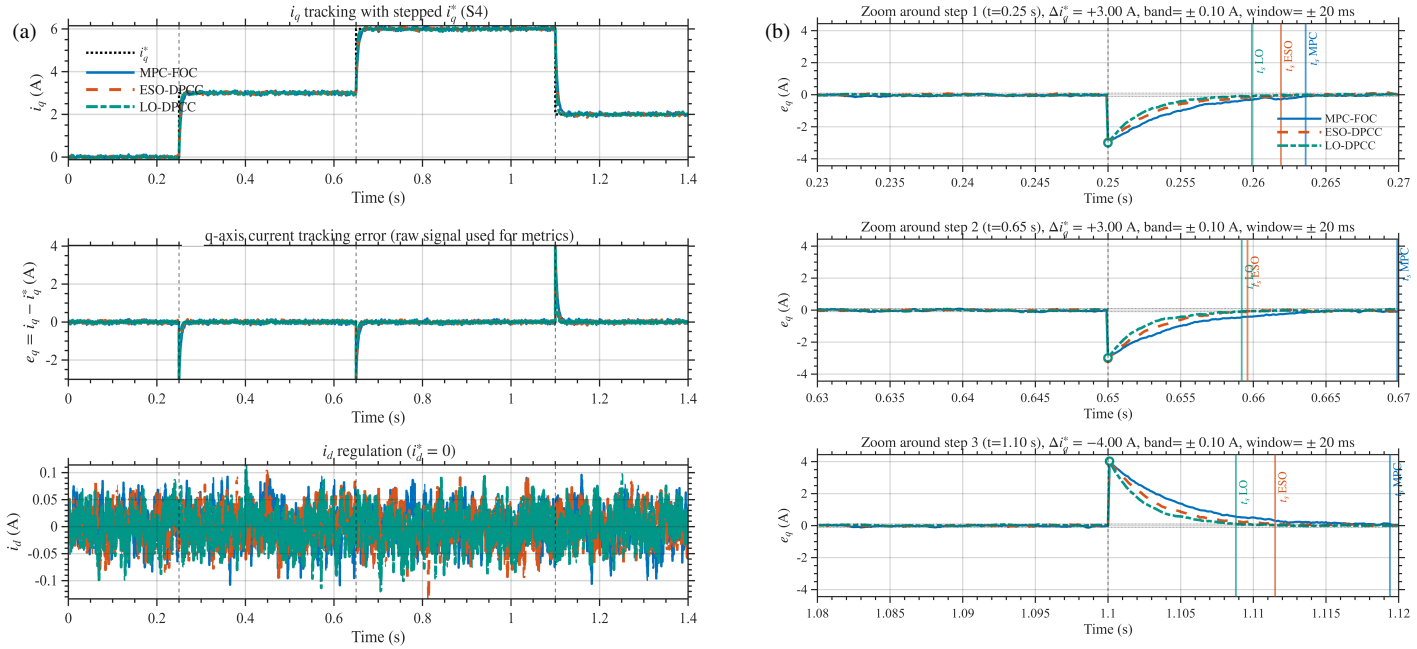

FIGURE 15. Current tracking performance comparison under step current-reference changes.

TABLE 5. Structural comparison of representative PMSM control strategies and quantitative metrics.

Method	Key feature	Speed response (ms)	Torque ripple (%)	Overshoot (%)
MPC-FOC (baseline, this work)	Voltage-vector MPCC + SVPWM inner current loop; no explicit digital-delay compensation in the current loop.	71	25.3	15.2
ESO-DPCC (baseline, this work)	Extended-state-observer-assisted deadbeat predictive current control without explicit digital-delay compensation in the current loop	≈ 40	22.3	1
LO-DMPC [14]	Model-free deadbeat predictive current control using a parameter-free Luenberger disturbance observer	-	-	-
LO-ADRC [8]	Sensorless ADRC vector control with a Luenberger observer for position and disturbance estimation	-	-	-
Proposed LO-DPCC (this work)	Closed-loop observe-predict-execute; single-sample delay compensation in the current loop	17	15.3	5.8

in step-path overshoot from about 15.2% to 5.8% when LO-DPCC is used instead of MPC-FOC. The ESO-DPCC method consistently delivers intermediate performance between MPC-FOC and LO-DPCC in all metrics, suggesting that deadbeat predictive current control alone cannot fully compensate for the digital delay and that the additional observer-based one-step-ahead prediction is important for recovering the intended current-loop dynamics.

Compared with more elaborate observer-assisted predictive control schemes (e.g., SMO-assisted ESO-DPCC, LO-DMPC, and LO-ADRC), the present LO-DPCC formulation performs delay compensation and one-step-ahead current prediction within a single discrete-time state-space framework. It employs pole-placement-based Luenberger gains with a small and intuitively tuned pole set, and preserves a closed-form deadbeat law compatible with standard SVPWM modulation and real-time DSP constraints. The combined simulation and experimental evidence indicates that this implementation-oriented design delivers competitive and, in some metrics, improved dynamic response, torque smoothness, and path-tracking accuracy under realistic embedded-hardware limitations, making LO-DPCC a promising actuator-layer control strategy for PMSM drives in 4WIS electric-vehicle applications.

5.5. Key Findings and Limitations

Across the unified 10 kHz PWM/100 µs current-loop timing, LO-DPCC consistently outperforms the uncompensated MPC-FOC baseline and the ESO-DPCC benchmark in transient speed regulation and torque smoothness. The improvement is attributed to a timing-consistent reachable-step objective: the voltage computed at time k is explicitly designed to shape the first current sample it can influence (approximately $k+2$ under the adopted update latency). A limitation of the present bench validation is that load effects in S2–S3 are emulated by command-side torque-demand injection rather than a mechanical dynamometer, and the vehicle-level case (S5) is evaluated only in co-simulation. Future work will include hardware-in-the-loop or dynamometer tests under mechanical load disturbances and a broader operating envelope with parameter variations (e.g., resistance drift and inductance saturation).

6. CONCLUSIONS

To tackle the response lag and torque ripple caused by the inherent one-sample sampling/computation/PWM delay in embedded PMSM drives, this study proposes a cooperative LO-DPCC scheme that integrates a discrete-time Luenberger observer with deadbeat predictive current control. By forming a closed observe-predict-execute loop, the proposed method replaces the delayed feedback current with a one-step-ahead observed estimate, thereby providing lead-phase compensation for the current loop and enabling accurate voltage-vector computation under realistic timing constraints. The proposed controller is developed and validated at the actuator-drive level on a single PMSM platform, while vehicle-level implications are assessed only through co-simulation using 4WIS-oriented actuation profiles.

Simulations and DSP-based bench experiments indicate that, for the adopted PMSM and unified timing configuration (10 kHz PWM and a 100 µs current/observer interrupt), LO-DPCC achieves approximately fourfold faster 2% speed settling and reduces the mean torque-ripple ratio from 25.3% to 15.3% compared with a tuned MPC-FOC baseline under the tested actuator-drive conditions. In addition, when the same drive-level controllers are embedded into a bicycle-model-based vehicle-drive co-simulation with 4WIS-oriented steering allocation, the step-path case (S5) predicts a reduction of path overshoot from about 15.2% to 5.8%. These results collectively suggest that observer-based one-step-ahead delay compensation can improve torque delivery predictability at the drive layer, with potential benefits to path-tracking performance that are currently supported by co-simulation evidence only.

The present work deliberately confines its quantitative evaluation to a single motor-drive platform and four representative 4WIS maneuvers. Future research will extend the proposed LO-DPCC to a broader range of operating points, investigate robustness against parameter uncertainties and measurement noise, and integrate adaptive delay prediction and nonlinear observer design to further enhance performance in high-dynamic PMSM drive systems [26].

REFERENCES

[1] Peng, J. and M. Yao, "Overview of predictive control technology for permanent magnet synchronous motor systems," *Applied Sciences*, Vol. 13, No. 10, 6255, 2023.

[2] Li, H., C. Zhao, and Y. Xu, "Model predictive current control of permanent magnet synchronous motor based on extended Kalman filter," in *Frontier Academic Forum of Electrical Engineering*, Vol. 1048, 899–912, Springer, Singapore, 2023.

[3] Pan, H., T. Gu, X. Wang, and K. Xiao, "A new model predictive current control strategy for PMSM based on extended control set and dynamic cost function," *Electrical Engineering*, Vol. 107, No. 4, 3947–3959, 2025.

[4] Pan, H., X. Huang, P. Guan, and S. Liu, "Grey-prediction-based double model predictive control strategy for the speed and current control of permanent magnet synchronous motor," *Asian Journal of Control*, Vol. 24, No. 6, 3494–3507, 2022.

[5] Liu, X., J. Wang, X. Gao, W. Tian, L. Zhou, and R. Kennel, "Robust predictive speed control of SPMSM drives with algebraically designed weighting factors," *IEEE Transactions on Power Electronics*, Vol. 37, No. 12, 14 434–14 446, 2022.

[6] Xu, X., L. Wang, Z. Zhang, and Z. Wu, "A novel robust control for permanent magnet synchronous motor integrating deadbeat predictive current control and sliding mode observer," *Alexandria Engineering Journal*, Vol. 130, 57–67, 2025.

[7] Mao, K., A. Dong, Z. Liu, C. Zhou, and S. Zheng, "Deadbeat predictive current control of high-speed PMSM with TLS based multiparameter online identification," *Journal of Electrical Engineering & Technology*, Vol. 21, 465–480, 2026.

[8] Zhang, P., Z. Shi, B. Yu, and H. Qi, "Research on a sensorless ADRC vector control method for a permanent magnet synchronous motor based on the Luenberger observer," *Processes*, Vol. 12, No. 5, 906, 2024.

[9] Shen, W., L. Shao, D. Liu, J. Wang, and C. Ge, "Event-triggered FCS-MPC with sliding mode observer for permanent magnet synchronous motor servo motion systems," *IEEE Transactions on Automation Science and Engineering*, Vol. 22, 2257–2268, 2024.

[10] Yao, M., J. Peng, and X. Sun, "Robust model predictive current control for six-phase PMSMS with virtual voltage vectors," *Journal of Power Electronics*, Vol. 22, No. 11, 1885–1896, 2022.

[11] Li, T., A. Li, L. Hou, and Z. Wang, "An improved deadbeat predictive current control for permanent magnet synchronous motor with sliding mode observer," in *International Conference of Electrical, Electronic and Networked Energy Systems*, Vol. 1330, 141–150, Springer, Singapore, 2025.

[12] Yang, N., S. Zhang, X. Li, and X. Li, "A new model-free deadbeat predictive current control for PMSM using parameter-free Luenberger disturbance observer," *IEEE Journal of Emerging and Selected Topics in Power Electronics*, Vol. 11, No. 1, 407–417, 2023.

[13] Zhao, K., W. Dai, S. Wu, P. Qiu, W. Liu, and G. Huang, "Improved super-twisting-observer-based finite-control-set model-predictive fault-tolerant current control of PMSM considering demagnetization fault," *International Journal of Electrical Power & Energy Systems*, Vol. 142, 108325, 2022.

[14] Luo, P., S. Zhang, A. Shen, X. Luo, and Q. Tang, "Current disturbance compensation of continuous model predictive control scheme for PMSMs," *ISA Transactions*, Vol. 155, 92–103, 2024.

[15] Vazquez, S., D. Marino, E. Zafra, M. D. V. Peña, J. J. Rodríguez-Andina, L. G. Franquelo, and M. Manic, "An artificial intelligence approach for real-time tuning of weighting factors in FCS-MPC for power converters," *IEEE Transactions on Industrial*

Electronics, Vol. 69, No. 12, 11 987–11 998, 2022.

[16] Wang, T., J. Wang, W. Guan, C. Liu, Y. Chen, Z. Chen, and G. Luo, "Data-driven based hybrid predictive model for the PMSM drive system," in *2023 IEEE International Conference on Predictive Control of Electrical Drives and Power Electronics (PRECEDE)*, 1–6, Wuhan, China, 2023.

[17] Li, S., W. Chen, and H. Wan, "Model predictive control method based on data-driven approach for permanent magnet synchronous motor control system," *Journal of Shanghai Jiaotong University (Science)*, Vol. 30, No. 2, 270–279, 2025.

[18] Zhang, H., S. Zhu, J. Jiang, Q. Wang, A. Wang, and D. Jin, "Simplified model prediction current control strategy for permanent magnet synchronous motor," *Journal of Power Electronics*, Vol. 22, No. 11, 1860–1871, 2022.

[19] Jeong, Y. and S. Yim, "Integrated path tracking and lateral stability control with four-wheel independent steering for autonomous electric vehicles on low friction roads," *Machines*, Vol. 10, No. 8, 650, 2022.

[20] Yang, Z., Y. Dai, M. Hu, Y. Bian, Q. Cui, and Y. Li, "Improving path tracking performance of 4WIS vehicles via constraint-oriented consistent coordinated steering," *Chinese Journal of Mechanical Engineering*, Vol. 38, No. 1, 158, 2025.

[21] Gu, D., T. Liu, Y. Yao, L. Ye, and H. Xu, "Sensorless PMSM control system based on Luenberger observer," *Journal of Physics: Conference Series*, Vol. 2497, No. 1, 012009, 2023.

[22] Chebbi, A., M. A. Franchek, and K. Grigoriadis, "Simultaneous state and parameter estimation methods based on kalman filters and Luenberger observers: A tutorial & review," *Sensors*, Vol. 25, No. 22, 7043, 2025.

[23] Zeghlache, A., A. Djerioui, H. Mekki, S. Zeghlache, and M. F. Benkhoris, "Robust sensorless PMSM control with improved back-EMF observer and adaptive parameter estimation," *Electronics*, Vol. 14, No. 7, 1238, 2025.

[24] Makinde, K. A., M. S. Bakare, B. O. Akinloye, A. O. Amole, O. B. Adewuyi, U. O. Zubair, and W. O. Owonikoko, "Simulation based testing and performance investigation of induction motor drives using MATLAB simulink," *SN Applied Sciences*, Vol. 5, No. 3, 73, 2023.

[25] Wang, F., J. Li, Z. Li, D. Ke, J. Du, C. Garcia, and J. Rodriguez, "Design of model predictive control weighting factors for PMSM using Gaussian distribution-based particle swarm optimization," *IEEE Transactions on Industrial Electronics*, Vol. 69, No. 11, 10 935–10 946, 2022.

[26] Hu, L., W. Lei, R. Li, and Y. Gao, "A graphic weighting factor design method for finite control set model predictive control of power converters," *IEEE Journal of Emerging and Selected Topics in Power Electronics*, Vol. 11, No. 2, 1661–1671, 2023.