JTE HCM-UTE
ISSN: 2615-9740

JOURNAL OF TECHNICAL EDUCATION SCIENCE
Ho Chi Minh City University of Technology and Engineering
Website: https://jte.edu.vn
Email: jte@hcmute.edu.vn

MPC Based Trajectory Planning for Permanent Magnet Synchronous Motor

Nhat Nam Dang, Duc Anh Le, Vinh Hao Nguyen*
Ho Chi Minh City University of Technology, VNU-HCM, Vietnam

*Corresponding author. Email: vinhhao@hcmut.edu.vn

ARTICLE INFO

Received: 17/01/2025
Revised: 10/02/2025
Accepted: 17/04/2025
Online First: 10/11/2025
Published: 28/11/2026

KEYWORDS

PMSM;
Model Predictive Control;
Load Torque Observer;
Trajectory Planning;
Space-Vector Pulse Width Modulation.

ABSTRACT

Permanent Magnet Synchronous Motor (PMSM) is increasingly used in applications requiring precise control, particularly in robotic joint control systems. This study aims to assess the feasibility of applying Model Predictive Control (MPC) for position and velocity control of PMSM. The research methodology involves using simulations to verify the effectiveness of MPC, along with the integration of an Kalman Filter (KF) to estimate external loads, thereby improving response time and control over motor parameters. Simulation results demonstrate that MPC achieves high precision in controlling PMSM under basic trajectory planning modes, while the KF enhances load estimation accuracy, allowing the system to stabilize quickly. Additionally, the algorithm has been tested to operate stably at low frequencies, confirming the applicability of MPC in real-time systems. The study concludes that the combination of MPC and KF is a feasible and effective solution for precise PMSM control, with broad potential applications in modern control systems.

DOI: https://doi.org/10.54644/jte.2025.1795

Copyright © JTE. This is an open access article distributed under the terms and conditions of the Creative Commons Attribution-NonCommercial 4.0 International License which permits unrestricted use, distribution, and reproduction in any medium for non-commercial purpose, provided the original work is properly cited.

1. Introdution

Nowadays, permanent magnet synchronous motors (PMSM) are widely used in industrial settings due to their high efficiency and outstanding dynamic performance [1]. For this reason, developing optimal control algorithms for precise position control applications has been a focus for many research groups. Since it was first proposed in the early 1970s, field-oriented control (FOC) has become highly popular [2]-[4].

Many velocity and position control algorithms have been introduced to replace the traditional PID controllers for velocity control, such as adaptive PID controllers [5], [6], and sliding mode controllers [7], [8], among others. However, these control models share a common limitation: they do not fully meet practical requirements, such as handling external load disturbances, addressing system current limitations, ensuring real-time position and velocity trajectory tracking, and, most importantly, controlling only a single parameter - either position or velocity.

In recent years, an alternative control strategy for PMSM has been gaining prominence: Model Predictive Control (MPC) [9], particularly discrete-time MPC with constraints. The discrete-time MPC with constraints utilizes a system model to predict future behavior over a defined time frame. At each prediction step, MPC solves an optimization problem to find the optimal control sequence, ensuring that state and input constraints are satisfied. Only the first control signal in the sequence is implemented, and the process is then repeated with updated states. This makes MPC highly flexible and effective in handling systems with complex and uncertain constraints.

Discrete-time MPC with constraints have been widely utilized in numerous papers and studies, primarily as a replacement for the PI controllers managing the I_d and I_q current [10]-[11] or find optimal switching pattern for inverter [12]. A smaller number of studies have implemented integrated MPC for both electrical and mechanical systems [13]. However, current controllers need to operate at high to very high frequencies to match the electrical time constants of the motor. As a result, their feasibility is limited due to the computational constraints of commonly used microcontrollers.

In this paper, the research team implements a discrete-time MPC controller with constraints for the mechanical model of a PMSM. Unlike previous studies, the MPC controller is combined with an KF to estimate the value of the external load. Analog-to-digital conversion (ADC) noise and encoder noise during velocity readings are also considered in this simulation. Additionally, the controller's virtual model is augmented with a 30% model error to accurately simulate the controller's behavior and performance in real-world scenarios. Simulation results are conducted using a real-world identification model of a PMSM, with a feasible control frequency (5 kHz) and external loads that vary in step, ramp, and sinusoidal patterns to demonstrate the validity of the control and observation algorithms. The principles of operation and design procedures are presented in Section 2. The motor model and control parameter selection are described in Section 3. Simulation results, along with analysis and evaluation, are provided in Section 4.

2. Theoretical Foundation

The structure of the discrete-time MPC controller with constraints is illustrated in Figure 1. The diagram comprises three main components: the MPC controller block, responsible for trajectory planning and computing the optimal control value $i_{q\,ref}$; the FOC block, which includes two PI controllers for regulating the currents i_d and i_q; and the Space Vector Pulse Width Modulation (SVPWM) block, which calculates the switching cycles for the two-level inverter controlling the PMSM.

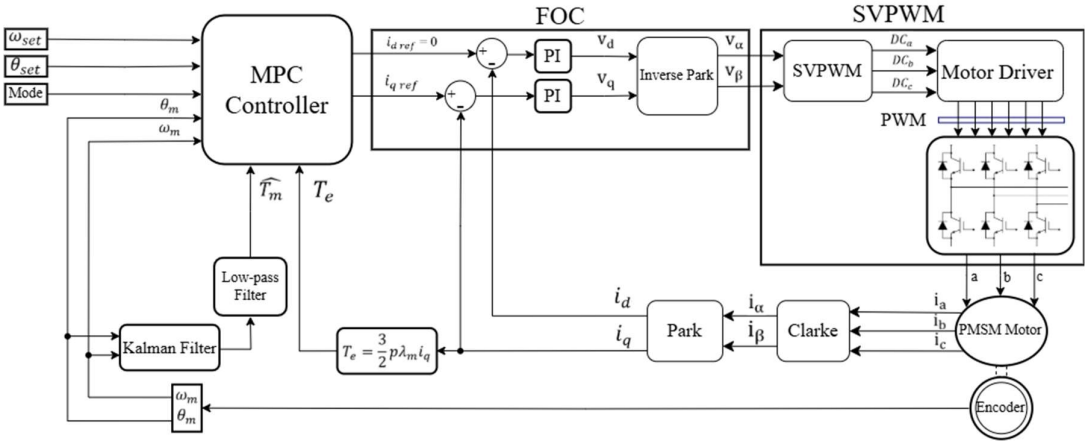


Figure 1. *The structure of MPC.*

2.1. Mathematical model of PMSM

The discrete-time MPC with constraints requires a system model to perform its control tasks. The research team decided to select the mechanical model of the PMSM to design the controller due to its computational simplicity, making it suitable for practical applications. The mechanical state equations are referenced in [14] as follows

$$\begin{cases} \dot{\omega_m} = -\dfrac{B_m}{J}\omega_m + \dfrac{1}{J}(T_e - T_m) \\ \dot{\theta_m} = \omega_m \end{cases} \tag{1}$$

Where T_e is motor torque, T_m is load torque, ω_m is the motor angular velocity and θ_m is the motor angular position.

The system of differential equations is discretized using the simple Euler method with a sampling time T_s.

$$x(k+1) = x(k) + T_s f(x,u) \tag{2}$$

Where $x(k)$ is the state of the system at time step k; $x(k+1)$ is the state of the system at time step $k+1$; $f(x,u)$, which is the system's dynamics function, describes the rate of change of the state over time. It is typically given by a differential equation of the form: $\dot{x} = f(x,u)$.

Define $u(k) = T_e(k) - T_m(k)$; $X(k) = \begin{bmatrix} \omega_m(k) \\ \theta_m(k) \\ u(k) \end{bmatrix}$, $Y(k) = \begin{bmatrix} \omega_m(k) \\ \theta_m(k) \end{bmatrix}$, the extended discrete-time state-space representation of the system can then be expressed in matrix form as follows

$$\begin{cases} X(k+1) = \underbrace{\begin{bmatrix} 1 - T_s\frac{B_m}{J} & 0 & \frac{T_s}{J} \\ T_s & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}}_{A_e} \mathrm{X(k)} + \underbrace{\begin{bmatrix} \frac{T_s}{J} \\ 0 \\ 1 \end{bmatrix}}_{B_e} \Delta u(k) \\ Y(k) = \underbrace{\begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix}}_{C_e} X(k) \end{cases} \quad (3)$$

Define A_e as $\begin{bmatrix} 1 - T_s\frac{B_m}{J} & 0 & \frac{T_s}{J} \\ T_s & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$; B_e as $\begin{bmatrix} \frac{T_s}{J} \\ 0 \\ 1 \end{bmatrix}$ and C_e as $\begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix}$

2.2. Estimate load torque T_m using KF

From Equation (1), it can be observed that the control signal is a combination of two components: the electromagnetic torque T_e and the load torque T_m. While T_e can be calculated, T_m represents a random external disturbance to the system. Therefore, the research team decided to employ an KF to estimate T_m. The KF-based estimation method is introduced in references [1], [15]. Additionally, the KF is tasked with filtering noise from the motor's angular position and velocity measurements. From the system of equations (1), we discretize the model and define $Y_{KF}(k) = X_{KF}(k) = \begin{bmatrix} \omega_m(k) \\ \theta_m(k) \\ T_m(k) \end{bmatrix}$, we obtain the following discrete-time state-space model

$$\begin{cases} X_{KF}(k) = \begin{bmatrix} 1 - T_s\frac{B_m}{J} & 0 & 0 \\ T_s & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} X_{KF}(k-1) + \begin{bmatrix} \frac{T_s}{J} \\ 0 \\ 0 \end{bmatrix} T_e(k-1) - \frac{T_s}{J} T_m(k-1) \\ Y_{KF}(k) = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix} X_{KF}(k) \end{cases} \quad (4)$$

Where $-\frac{T_s}{J} T_m(k-1)$ is considered as the disturbance component of the model.

Let Q and R denote the covariance matrices of the system noise and measurement noise. The KF loop consists of two main stages: the prediction stage and the update stage. At the first stage, the state and covariance at time *k* are predicted based on the information from time $k-1$.

Step 1: State prediction matrix equation

$$\hat{x}_{k|k-1} = f\left(\hat{x}_{k-1|k-1}, u_{k-1}\right) = Ax_{k-1|k-1} + Bu_{k-1} \quad (5)$$

Step 2: Covariance prediction matrix equation

$$P_{k|k-1} = AP_{k-1|k-1}A^T + Q \quad (6)$$

At the update state, after obtaining the measurement data at time k, the state and covariance matrices are updated as follows

Step 1: Calculate the measurement covariance matrix

$$S_k = CP_{k|k-1}C^T + R \tag{7}$$

Step 2: Calculate the Kalman gain

$$K_k = P_{k|k-1}C^T S_k^{-1} \tag{8}$$

Step 3: Update the state matrix

$$\hat{x}_{k|k} = \hat{x}_{k|k-1} + K_k(y_k - \hat{y}_{k|k-1}) \tag{9}$$

Step 4: Update the covariance matrix

$$P_{k|k} = (I - K_k C)P_{k|k-1} \tag{10}$$

The estimated value $\widehat{T_m}$ is contained in the state matrix $\hat{x}_{k|k}$. Once $\widehat{T_m}$ is obtained, it is passed through a low-pass filter to stabilize the estimation and minimize the impact of noise from angular velocity and position measurements on the load estimation.

2.3. Control signal constraints and predicted output matrix

To constrain the control signal within the limits defined by the motor's maximum electromagnetic torque $T_{e\,max}$, the following inequality must be satisfied $-T_{e\,max} \le \Delta u(k) + u(k) \le T_{e\,max}$. Define the control difference matrix ΔU as $\begin{bmatrix} \Delta u(k) \\ \Delta u(k+1) \\ \dots \\ \Delta u(k+N_c-1) \end{bmatrix}$, the control signal constraints can then be expressed as follows

$$M\Delta U \le \mathrm{N} \tag{11}$$

Where the matrix M = $\begin{bmatrix} 1 & 0 & 0 & 0 & \dots & 0 \\ 1 & 1 & 0 & 0 & \dots & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ 1 & 1 & 1 & 1 & \dots & 1 \\ -1 & 0 & 0 & 0 & \dots & 0 \\ -1 & -1 & 0 & 0 & \dots & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ -1 & -1 & -1 & -1 & -1 & -1 \end{bmatrix}$, N = $\begin{bmatrix} T_{e\,max} - u(k) \\ T_{e\,max} - u(k+1) \\ \dots \\ T_{e\,max} - u(k+N_c-1) \\ T_{e\,max} + u(k) \\ T_{e\,max} + u(k+1) \\ \dots \\ T_{e\,max} + u(k+N_c-1) \end{bmatrix}$

With the prediction step N_p and control step N_c, we have the following

$$\begin{cases} X(k_i+1|k_i) = A_e X(k_i) + B_e \Delta u(k_i) \\ X(k_i+2|k_i) = X(k_i) + A_e B_e \Delta u(k_i) + B_e \Delta u(k_i+1) \\ \vdots \\ X(k_i+N_p|k_i) = {A_e}^{N_p} X(k_i) + \dots + {A_e}^{N_p-N_c} B_e \Delta u(k_i+N_c-1) \end{cases} \tag{12}$$

$$\begin{cases} \tilde{Y}(k_i+1|k_i) = C_e(A_e X(k_i) + B_e \Delta u(k_i)) \\ \tilde{Y}(k_i+2|k_i) = C_e(A_e^2 X(k_i) + A_e B_e \Delta u(k_i) + B_e \Delta u(k_i+1)) \\ \vdots \\ \tilde{Y}(k_i+N_p|k_i) = C_e {A_e}^{N_p} X(k_i) + \dots + C_e {A_e}^{N_p-N_c} B_e \Delta u(k_i+N_c-1) \end{cases} \tag{13}$$

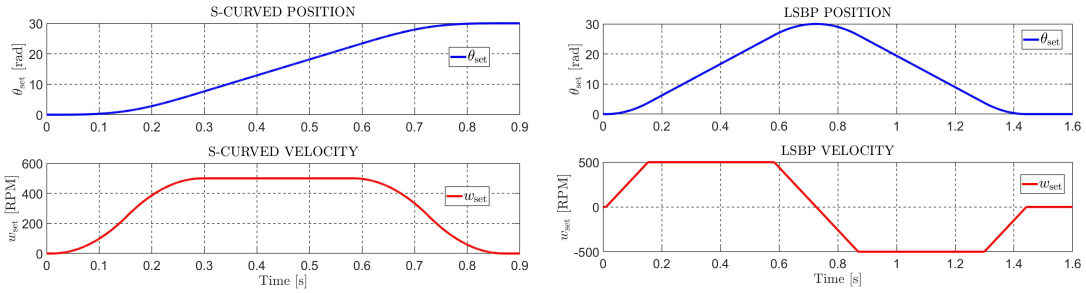
Define $\tilde{Y} = \begin{bmatrix} \tilde{Y}(k_i + 1|k_i) \\ \tilde{Y}(k_i + 2|k_i) \\ \cdots \\ \tilde{Y}(k_i + N_p|k_i) \end{bmatrix}$ as the prediction sequence from the model, which results in

$$\tilde{Y} = FX(k_i) + G\Delta U \tag{14}$$

Where the matrix $F = \begin{bmatrix} C_e A_e \\ C_e A_e^2 \\ \cdots \\ C_e {A_e}^{N_p} \end{bmatrix}$, $G = \begin{bmatrix} C_e A_e^0 B_e & \cdots & 0 \\ \vdots & \ddots & \vdots \\ C_e {A_e}^{N_p-1} B_e & \cdots & C_e {A_e}^{N_p-N_c} B_e \end{bmatrix}$

2.4. Trajectory planning

Let R_s denote the matrix of instantaneous trajectory points that the MPC controller needs to follow. For each trajectory planning mode, there will be different sequences of reference points ω_{ref}, θ_{ref}. These trajectory point sequences are calculated from basic velocity and position kinematic equations. A typical use case of the MPC controller is to control the smooth motion of the joints of a robotic arm in robotics applications. In this paper, the research team establishes two commonly used trajectory forms for position control: the S-Curved trajectory and the Linear Segment with Parabol Blend (LSPB) velocity trajectory which are presented in Figure 2.



(a) *(b)*

Figure 2. *Trajectory Angular Position and Velocity.*
(a): The S-Curved position trajectory; (b): The LSPB velocity trajectory.

With the velocity, position, and time trajectory equations, the MPC controller calculates the trajectory point sequence R_s. After the reference points $\omega_{ref}, \theta_{ref}$ are obtained, the matrix R_s is defined as follows

$$R_s = \begin{bmatrix} r(k_i + 1|k_i) \\ r(k_i + 2|k_i) \\ \cdots \\ r(k_i + N_p|k_i) \end{bmatrix}$$

Where $r(k_i + j|k_i) = \begin{bmatrix} \omega_{ref}(k_i + \text{j}) \\ \theta_{ref}(k_i + \text{j}) \end{bmatrix}$ with $j = 0, 1, 2, ..., N_p - 1$.

2.5. The cost function and optimize the cost function using Hildreth's quadratic programming (QP) procedure

The cost function is formulated as follows:

$$J = \left(\tilde{Y} - R_s\right)^T Q_y \left(\tilde{Y} - R_s\right) + \Delta U^T R_u \Delta U \tag{15}$$

Where the first term $\left(\tilde{Y} - R_s\right)^T Q_y \left(\tilde{Y} - R_s\right)$ is the tracking cost, used to evaluate the deviation between the current state and the desired state. The second term ($\Delta U^T R_u \Delta U$) is the control effort cost, used to assess the control signal applied in order to prevent excessively large or sudden control actions. The matrices Q_y and R_u are the weighting matrices for the state and control signals, respectively. These two matrices are defined as follows

$$Q_y = \begin{bmatrix} Q & 0 & 0 \\ 0 & \ddots & 0 \\ 0 & 0 & 2xQ \end{bmatrix} with\ Q = \begin{bmatrix} Q_\omega & 0 \\ 0 & Q_\theta \end{bmatrix};\ R_u = \begin{bmatrix} R & 0 & 0 \\ 0 & \ddots & 0 \\ 0 & 0 & R \end{bmatrix}$$

By substituting the predicted output matrix $\tilde{Y}$ into the cost function J, which results in

$$\begin{aligned} J &= (FX + G\Delta U - R_s)^T Q_y (FX + G\Delta U - R_s) + \Delta U^T R_u \Delta U \\ &= ((FX - R_s)^T + \Delta U^T G^T)\left(Q_y (FX - R_s) + Q_y G \Delta U\right) + \Delta U^T R_u \Delta U \\ &= \Delta U^T \underbrace{\left(G^T Q_y G + R_u\right)}_{H} \Delta U + 2 \underbrace{(G^T Q_y (FX - R_s))^T}_{f} \Delta U + \underbrace{(FX - R_s)^T Q_y (FX - R_s)}_{const} \end{aligned} \tag{16}$$

To optimize the cost function, we solve the minimization problem J_{min}. Because the term $(FX - R_s)^T Q_y (FX - R_s)$ is fixed, the cost function that needs to be optimized:

$$J_{min} = \frac{1}{2} \Delta U^T \underbrace{\left(G^T Q_y G + R_u\right)}_{H} \Delta U + \underbrace{\left(G^T Q_y (FX - R_s)\right)^T}_{f} \Delta U \tag{17}$$

From (11) and (17), we derive the quadratic programming (QP) problem with constraints as follows

$$J_{min} = \frac{1}{2} \Delta U^T H \Delta U + f^T \Delta U \text{ with } M\Delta U \le N \tag{18}$$

Where H is a semi-definite matrix ($H \ge 0$) to ensure the existence of an optimal solution.

Applying the Primal-Dual Method, where the Lagrange multipliers are treated as dual variables [16], the problem becomes

$$\max_{\lambda \ge 0} \min_{\Delta U} \left[\frac{1}{2} \Delta U^T H \Delta U + \Delta U^T f + \lambda^T (M\Delta U - N)\right] \tag{19}$$

Minimizing the dual function with respect to ΔU, which is unconstrained, is derived from [16] as follows

$$\Delta U = -H^{-1}(f + M^T \lambda) \tag{20}$$

Substituting (20) into (19), the problem reduces to

$$\min_{\lambda \ge 0} \left(\frac{1}{2} \lambda^T M H^{-1} M^T \lambda + \lambda^T (N + M H^{-1} f) + \frac{1}{2} \lambda^T H^{-1} \lambda\right) \tag{21}$$

Starting from the original optimization problem with complex constraints, using duality transforms it into the optimization of $J_{min}(\lambda)$ with the condition $\lambda \ge 0$.

Now, the problem is formulated as finding ΔU that optimizes the cost function. To solve this problem, we use Hildreth's quadratic programming procedure. This procedure is used to find the optimal Lagrange multipliers for the dual function by updating each Lagrange multiplier in every iteration until they converge at a point. When the set of Lagrange multipliers converges, the following properties hold

$$\text{If } \lambda_i > 0 \text{ then } M\Delta U < N$$

$$\text{If } \lambda_i = 0 \text{ then } M\Delta U = N$$

The Hildreth's quadratic programming procedure is implemented as follows

Step 1: Calculate the unconstrained solution

$$\Delta U = -H^{-1}f \tag{22}$$

Step 2: Check the constraints $M\Delta U \leq N$. If ΔU does not violate any constraints, ΔU will be the global optimal solution. Otherwise, proceed to step 3 to adjust the solution.

Step 3: Compute the Lagrange multiplier

$$\lambda_i^{m+1} = max\left(0; -\frac{1}{h_{ij}}\left[k_i + \sum_{j=1}^{i-1} h_{ij}\lambda_j^{m+1} + \sum_{j=i+1}^{n} h_{ij}\lambda_j^{m}\right]\right) \tag{23}$$

Where h_{ij} is the element at row i and column j of the matrix $MH^{-1}M^T$, and k_i is the element at row i of the matrix $N + MH^{-1}f$.

Step 4: Check the convergence of the Lagrange multiplier.

$$(\lambda - \lambda_p)^T(\lambda - \lambda_p) \tag{24}$$

If the convergence is less than the desired threshold, return to step 3.

Step 5: The local optimal solution for ΔU is achieved

$$\Delta U = -H^{-1}(f + M^T\lambda^*) \tag{25}$$

Where $\lambda^* = \lambda_i^{m+1}$ is the optimal Lagrange multiplier

Once ΔU is computed, update the electromagnetic torque $T_e(k)$ as follows

$$T_e(k+1) = T_e(k) + \Delta u(k) \tag{26}$$

The reference current $i_{q\,ref}$ is calculated using the formula for electromagnetic torque T_e

$$T_e = \frac{3}{2}p\lambda_m i_q \tag{27}$$

Where λ_m is the maximum flux linkage and p is the number of pole pairs of the PMSM.

To focus the entire torque on the motor's rotation, the reference current in the d-axis $i_{d\,ref}$ is set to zero.

3. Results and Discussion

To evaluate the performance of the MPC controller, the research team conducted simulations to analyze the system response and error under different control modes and various external load types by using Matlab Simulink. The MPC controller and the KF were implemented using MATLAB function blocks, ensuring efficient integration and computational effectiveness. The L_2 norm was selected as the criterion for assessing the tracking quality of the system's responses (e.g., velocity and load estimation), while the L_∞ norm was used to evaluate the maximum error for responses requiring high precision (e.g., position). Additionally, the team separately assessed the predictive control capability and the response to external loads to provide a more comprehensive view of the control system's performance. The motor selected for model identification and simulation was the MJ5208 from MJBOTS, with specifications provided in Table 1.

JTE
HCM-UTE
ISSN: 2615-9740

JOURNAL OF TECHNICAL EDUCATION SCIENCE
Ho Chi Minh City University of Technology and Engineering
Website: https://jte.edu.vn
Email: jte@hcmute.edu.vn

Table 1. *Motor's specifications.*

Parameter	Identification Value	Units
Bus voltage V_{DC}	24	V
Stator resistance R_s	0.1	Ω
Stator inductance L_s	0.064333	mH
Flux linkage λ_m	0.0026382	V.s
Pole pairs p	7	
Viscous damping B_m	3.1599 x10^{-6}	Nm.s
Inertia J	6.7464 x10^{-6}	kg.m^2
Max electromagnetic torque $T_{e\,max}$	1.25	Nm
Max current I_{max}	22	A

The simulation parameters is selected as follows: $f_{MPC} = 5\ kHz; N_p = N_c = 6; Q = \begin{bmatrix} 3 & 0 \\ 0 & 10 \end{bmatrix}; R = [0{,}01];\ f_{KF} = 5\ kHz; Q_{KF} = \begin{bmatrix} 0{,}01 & 0 & 0 \\ 0 & 0{,}01 & 0 \\ 0 & 0 & 10 \end{bmatrix}; R_{KF} = [50]; f_{LPF} = 500\ Hz;\ f_{PI} = 50\ kHz; K_p = 0.067367; K_i = 104.72; f_{SVPWM} = 30\ kHz.$

The virtual model of the MPC controller and the KF was intentionally modified by adding a 30% error to the parameters B_m, J and λ_m. This simulates the modeling errors that may occur during real-world model identification. Additionally, noise in reading three-phase current values, as well as encoder noise during velocity and position measurement, was also taken into account.

3.1. Assessment of Predictive Control Performance of MPC and Response to Varying Load Torque

To demonstrate the predictive control capability, the research team conducted a velocity control simulation using a step function trajectory combined with an external load torque.

$$T_m = \begin{cases} 0\ (t < 0\ or\ t > 0{,}45) \\ 0{,}3\ Nm\ (0{,}15 \le\ t \le\ 0{,}45) \end{cases}$$

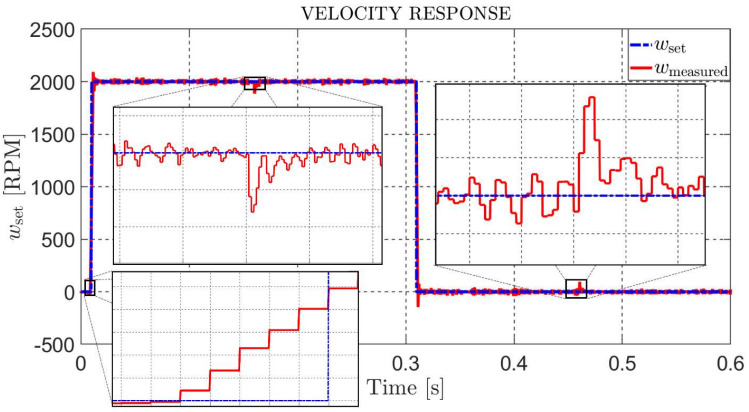


Figure 3. *Velocity Response in the Case of Step Function Velocity Planning.*

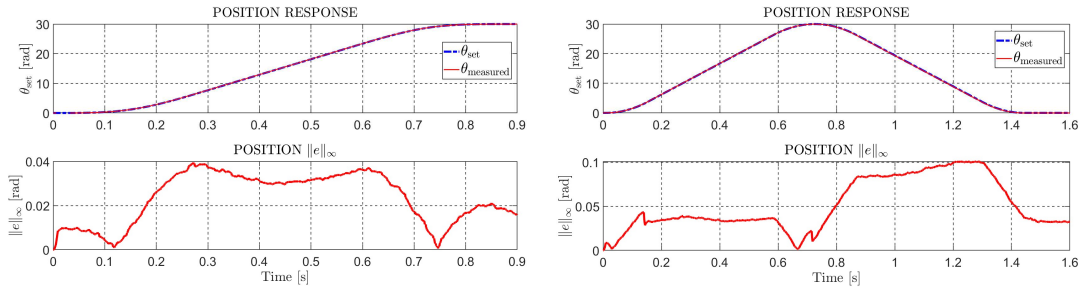
In Figure 3, it can be observed that before the set velocity changes, the controller adjusts the velocity in advance by up to $\le N_p$ control steps (6 steps in this simulation). This demonstrates the superiority of the MPC controller, highlighting its ability to foresee N_c steps and control the system N_p steps ahead. By generating control signals earlier instead of reacting only after the setpoint changes, as traditional controllers do, the system achieves faster response and reduces settling time.

With the KF estimating the external load, the system can quickly return to the desired set velocity, demonstrating its dynamic response capability in the presence of unknown load disturbances. At t = 0.15 second, an external load of 0.3 Nm is applied to the motor, causing a reduction in its speed. However, the Kalman Filter promptly estimates this external load and transmits to the MPC controller. The MPC controller compensates for the external load within the virtual model and increases $i_{q\,ref}$ to regulate the system. A similar process occurs at t = 0.45 second, when the external load is removed, ensuring that the system maintains the planned velocity and position. This feature is crucial in precise control systems subjected to external influences, particularly in robotics, where external loads constantly vary with the position of the robot's joints.

3.2. Evaluation of Tracking Control Quality

The most typical application of the MPC controller is in achieving smooth motion control for robotic arm joints in robotics applications. Therefore, the research team conducted simulations and evaluated the overall performance of the controller in two trajectory planning scenarios: S-Curved and Linear Segment with Parabolic Blend (LSPB). These scenarios were combined with external load estimations in the form of sinusoidal loads and mixed loads (a combination of step and ramp functions).

3.2.1. Evaluation of Angular Position Control Quality

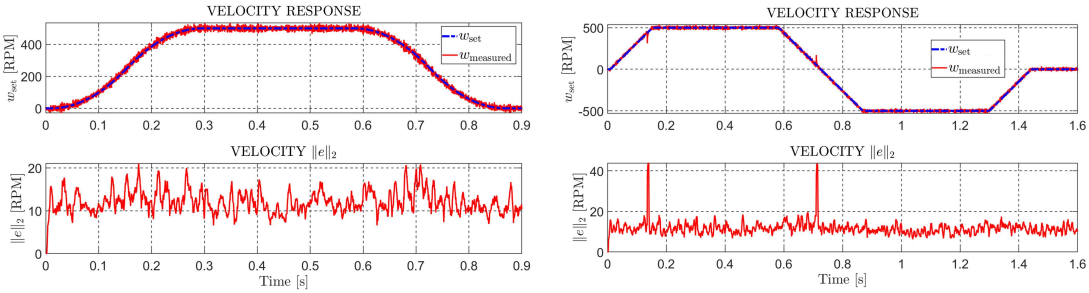


(a) *(b)*

Figure 4. *Position Response and Position Response Error*
(a): The S-Curved trajectory with sinusoidal load; (b): The LSPB velocity trajectory with mixed load.

As shown in Figure 4, from the L_∞ error plots of the two control modes, it can be observed that the controller provides excellent position tracking performance along the predefined trajectory. Additionally, the response is unaffected by external loads, and the system's response time is nearly zero. This indicates that the controller excels in prioritizing position control along the predefined trajectory, making it highly suitable for precision control applications.

3.2.2. Evaluation of Angular Velocity Control Quality

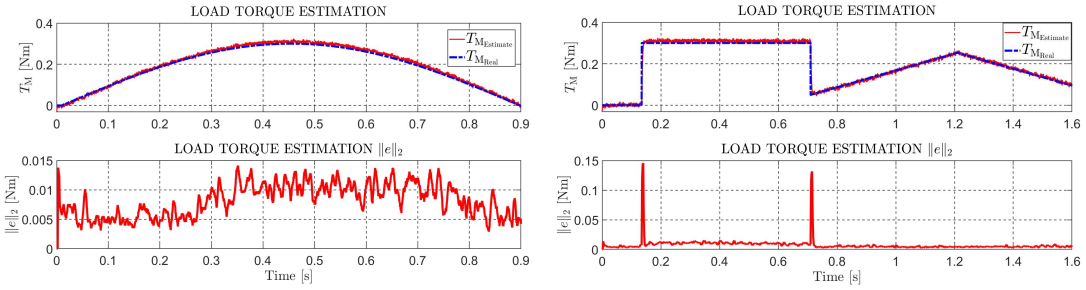


(a) *(b)*

Figure 5. *Velocity Response and Velocity Response Error*
(a) The S-Curved trajectory with sinusoidal load; (b) The LSPB velocity trajectory with mixed load.

In Figure 5, the velocity still tends to increase ahead of the setpoint due to the ability of the MPC controller. Meanwhile, the tracking error in both scenarios remains within ±4% of the set velocity. This demonstrates the controller's ability to effectively track velocity along the predefined trajectory. For the LSPB trajectory with mixed external loads, it can be observed that the velocity response increases and fluctuates abruptly due to the step change in the external load. However, the controller quickly stabilizes the velocity by adjusting the control signals, leveraging the KF to estimate the external load.

3.2.3. Evaluation of Load Torque Estimation



(a) *(b)*

Figure 6. *Estimation of Torque Load and Estimation of Torque Load Error*
(a) The S-Curved trajectory with sinusoidal load; (b) The LSPB velocity trajectory with mixed load.

In Figure 6, it can be observed that the estimation error is relatively high when the load spikes suddenly. This occurs because the KF cannot immediately estimate the abrupt changes in external load, and the low-pass filter causes a delay in the estimation compared to the actual value. However, this trade-off improves the accuracy of the load estimation, reducing large fluctuations under the continuous changes in motor velocity. The delay is brief (7 ms), after which the filter estimates the external load with reasonable accuracy (3% error) and compensates for it in the MPC controller.

For sinusoidal load changes (simulating a gradual increase in external load, as seen in robotic arm applications), the KF performs very well, with an estimation error of only 3.3% relative to the reference signal and almost no delay in the estimation. This shows that the combination of the KF for external load estimation and the MPC delivers excellent tracking performance, even when external loads intervene in the system.

4. Conclusion

In this paper, the research team successfully simulated a discrete MPC with constrains combined with an KF to estimate external loads for velocity and position trajectory planning of a PMSM. Comprehensive comparative experiments were conducted in a simulation environment to evaluate the performance of this control method. The results show that the control system achieved excellent tracking across various trajectory planning modes and types of external loads, with very low position tracking errors and external load estimation errors. These outcomes demonstrate that the controller offers high accuracy, practical feasibility, and strong applicability in precision control systems.

Acknowledgement

We acknowledge the support of time and facilities from Ho Chi Minh City University of Technology (HCMUT), VNU-HCM for this study.

Conflict of Interest

The authors declare no conflict of interest.

REFERENCES

[1] Z. Zedong, M. Fadel, and Y. Li, "A high-performance control system of PMSM based on load torque observer," in *Proc. IEEE Power Electron. Spec. Conf. (PESC)*, Orlando, FL, USA, 2007, pp. 587–592, doi: 10.1109/PESC.2007.4342053.
[2] R. Cai, *Current Control with Self-Learning Ability for PMSM Drives*, Aalborg Univ., 2022.

[3] E. Yesilbag and L. T. Ergene, "Field oriented control of permanent magnet synchronous motors used in washers," in *Proc. Int. Power Electron. Motion Control Conf. (PEMC)*, Antalya, Turkey, 2014, pp. 1259–1264, doi: 10.1109/EPEPEMC.2014.6980685.
[4] H. Wang, S. Xu, and H. Hu, "PID controller for PMSM speed control based on improved quantum genetic algorithm optimization," *IEEE Access*, vol. 11, pp. 61091–61102, 2023, doi: 10.1109/ACCESS.2023.3284971.
[5] J. W. Jung, V. Q. Leu, T. D. Do, E. K. Kim, and H. H. Choi, "Adaptive PID speed control design for permanent magnet synchronous motor drives," *IEEE Trans. Power Electron.*, vol. 30, no. 2, pp. 900–908, Feb. 2015, doi: 10.1109/TPEL.2014.2311462.
[6] R. C. García, W. I. Suemitsu, and J. O. P. Pinto, "Precise position control of a PMSM based on new adaptive PID controllers," in *Proc. IEEE Annu. Conf. Ind. Electron. Soc. (IECON)*, Melbourne, VIC, Australia, 2011, pp. 1983–1988, doi: 10.1109/IECON.2011.6119443.
[7] K. Li, J. Ding, X. Sun, and X. Tian, "Overview of sliding mode control technology for permanent magnet synchronous motor system," *IEEE Access*, vol. 12, pp. 71685–71704, 2024, doi: 10.1109/ACCESS.2024.3402983.
[8] L. Weijie, L. Dongliang, W. Qiuxuan, and Z. Xiaodan, "On sliding mode control of permanent magnet synchronous motor," in *Proc. Chinese Control Decis. Conf. (CCDC)*, Changsha, China, 2014, pp. 4555–4559, doi: 10.1109/CCDC.2014.6852985.
[9] A. Kiselev, A. Kuznietsov, and R. Leidhold, "Position control of a permanent magnet synchronous motor using generalized predictive control algorithm," in *Proc. Int. Symp. Power Electron., Electr. Drives, Autom. Motion (SPEEDAM)*, Capri, Italy, 2016, pp. 110–115, doi: 10.1109/SPEEDAM.2016.7525840.
[10] X. Huang, H. Pan, and K. Yuan, "Speed and current control of PMSM based on double MPC," in *Proc. Int. Forum Electr. Eng. Autom. (IFEEA)*, Hefei, China, 2020, pp. 300–304, doi: 10.1109/IFEEA51475.2020.00070.
[11] S. Bolognani, L. Peretti, and M. Zigliotto, "Combined speed and current model predictive control with inherent field-weakening features for PMSM drives," in *Proc. IEEE Mediterranean Electrotech. Conf. (MELECON)*, Ajaccio, France, 2008, pp. 472–478, doi: 10.1109/MELCON.2008.4618480.
[12] Č. Lukáš, *Algorithms for Advanced Motion Control Using Permanent Magnet Synchronous Motors and Brushless DC Motors*, 2020, ch. 3, sec. 3.5, pp. 35–36.
[13] S. Chai, L. Wang, and E. Rogers, "Model predictive control of a permanent magnet synchronous motor," in *Proc. IEEE Annu. Conf. Ind. Electron. Soc. (IECON)*, Melbourne, VIC, Australia, 2011.
[14] S. V. Zadeh, "Machine modeling," in *Control of Permanent Magnet Synchronous Motors*, Oxford, U.K.: Oxford Univ. Press, 2018, ch. 2, sec. 2.3, pp. 43–75.
[15] Z. Zheng, Y. Li, X. Xiao, and M. Fadel, "Load torque observer of permanent magnet synchronous motor," *Trans. China Electrotech. Soc.*, vol. 25, no. 2, pp. 30–36, 2010. ⟨hal-03549440⟩
[16] L. Wang, *Model Predictive Control System Design and Implementation Using MATLAB*, Springer, RMIT Univ., Australia, 2009, ch. 2, sec. 2.4, pp. 43–67.

Nhat Nam Dang is pursuing a B.S. degree in Control Engineering & Automation at Ho Chi Minh City University of Technology, Vietnam. His research interest focuses on control algorithms in embedded systems.
Email: nam.dang932037@hcmut.edu.vn. ORCID: https://orcid.org/0009-0001-4917-8948

Duc Anh Le is pursuing a B.S. degree in Control Engineering & Automation at Ho Chi Minh City University of Technology, Vietnam. His research interest focuses on control algorithm.
Email: anh.anhle44@hcmut.edu.vn. ORCID: https://orcid.org/0009-0001-7335-8367

Vinh Hao Nguyen is presently a lecturer of electrical and electronics engineering at Ho Chi Minh City University of Technology, Vietnam. He received B.S. and M.S. degrees in electrical and electronics engineering from Ho Chi Minh City University of Technology, Vietnam, in 2001 and 2003, respectively. He received Ph.D. degree in Electrical Engineering from the University of Ulsan, Korea, in 2009. His research interests are INS/GPS positioning system, autonomous robot and adaptive control system.
Email: vinhhao@hcmut.edu.vn. ORCID: https://orcid.org/0000-0003-2868-5534