Explicit model predictive control of permanent magnet synchronous motors based on multi-point linearization

Qian Guo¹,²,³, Tianhong Pan², Jinfeng Liu³ and Shan Chen¹

Abstract
Permanent magnet synchronous motors (PMSMs) have been broadly applied in servo-drive applications. It is necessary to improve the performance of PMSM. An explicit controller designed for PMSM based on multi-point linearization is proposed to reduce the linearized model error caused by different running status of PMSM. The mathematical model of PMSM system in the synchronous rotating frame and the problem formulation are introduced at first. Then, the preliminaries about explicit model predictive control (MPC) algorithm are presented in this article. Based on this, the multi-point linearization model is created for explicit MPC controller design. Moreover, the block diagram of the proposed method for PMSM system is presented. Finally, the simulation results are provided to demonstrate that the proposed explicit MPC controller based on multi-point linearization achieves better performance than that based on traditional single-point linearization, but requires the same online computation time because of the offline optimization of explicit MPC.

Keywords
Permanent magnet synchronous motors, explicit model predictive control, multi-point linearization

Introduction
In recent decades, permanent magnet synchronous motors (PMSMs) have played a significant role in motion control applications, including electric vehicles such as Toyota Prius, BYD, Nissa Leaf, and so on (Lara et al., 2016; Mwasilu et al., 2016; Sreejith and Singh, 2019; Velagić and Sabić, 2014; Zanma et al., 2020). This is mainly because PMSMs have various advantages like high-power density, high efficiency, high torque-current ratio, and compact structure (Koiwa et al., 2019; Zhang et al., 2019). Traditionally, the proportional integral (PI) (Jung et al., 2014; Yi et al., 2017) control has been applied in PMSM drives to obtain fast four-quadrant operation and smooth starting of the system. With the development of the computing hardware, a number of advanced control algorithms have emerged to obtain reduced torque ripple, improved efficiency and insensitivity to parameter changes but at the expense of high online computational complexity. The popular algorithms are fuzzy control, sliding mode control, adaptive control, neural network inverse method as well as model predictive control (MPC) (Liu et al., 2019; Mani et al., 2018; Sun et al., 2020; Wang et al., 2019; Wu et al., 2020). Among these approaches, MPC is proposed as an alternative control method for PMSM system due to its fast dynamic performance and multivariable constraint control capability (Ganapathy and Santha, 2019). MPC can predict and determine the future voltage vector to achieve the optimization of the cost function. The results are assessed by using the cost function penalizing the deviation from a given set point value to a hard constraint (Zhu et al., 2020). MPC transforms the control problem into an optimization problem, which provides great convenience for dealing with constraints and nonlinear models. The optimal control problem is solved at each sampling moment to obtain the control action sequence. The asymptotic stability and the optimality of constrained MPC are discussed in Mayne et al. (2000). The main outstanding advantage of MPC over other methods is that the security and physical limitations of the motor system can be directly incorporated into the formulation of control problems as control objectives (Jia et al., 2019). Therefore, the future constraint violations are preventable.

¹School of Electrical and Information Engineering, Jiangsu University, China
²School of Electrical Engineering and Automation, Anhui University, China
³Department of Chemical and Materials Engineering, University of Alberta, Canada

Corresponding authors:
Tianhong Pan, School of Electrical Engineering and Automation, Anhui University, #111 Jiulong Road, Hefei, Anhui 230601, China. Email: thpan@ahu.edu.cn
Jinfeng Liu, Department of Chemical and Materials Engineering, University of Alberta, 9211 116 Street, Edmonton, AB T6G 2V4, Canada. Email: jinfeng@ualberta.ca
However, one nonnegligible drawback of MPC is that the real-time implementation of MPC involves high computational complexity resulted by the optimization process at each sampling time. In order to reduce online computational burden, Bemporad et al. (2002) proposed an explicit solution to the optimization problem, which is known as explicit MPC. Explicit MPC can gain the same performance as MPC including the stability of the system. It transfers the optimization problem from online to offline so as to reduce the online computation time. The explicit control law are represented by continuous piecewise affine (PWA) functions. The online computation is only about simple lookup table to confirm the partition of the state variables, which considerably reduces the online computation time. This means that explicit MPC is certainly suitable for fast sampling systems like PMSM system. The design and validation of explicit MPC for PMSM was presented in Cygowski et al. (2005), demonstrating that the resulting controller achieves the same level of performance as conventional MPC but requires much less online computation time. A PMSM control method (Mynar et al., 2016) was developed based on explicit MPC using a novel linearization and constraint handling method, which allowed natural field weakening. It also added an extra compensation voltage calculation to the controller output to remove the inaccuracy introduced by the model linearization process.

Unfortunately, there are limitations on using MPC. It is generally known that the performance of MPC remains heavily dependent on the precision of the model. The linearized model error caused by different running status can lead to poor performance. This indicates that a single linear model is not able to obtain good control performance for this system. Therefore, based on explicit MPC, we propose a multi-point linearization method to linearize the PMSM model at a few operating points to cover the wide operating range. Because of its offline optimization, the online computational burden cannot be increased by using multi-point linearization method. Compared with the single-point linearization method, the dynamic and steady-state performance of the system has been improved when the working condition of the motor changes. The simulation results indicate the effectiveness of the proposed approach.

The structure of this article is arranged as follows. Firstly, the mathematical model of PMSM and the brief account of the existed problem are described. Then, the preliminaries of explicit MPC are reviewed. Subsequently, the proposed algorithm including the multi-point linearization method, model selection and model switching schedule, and the controller design are presented. The next section depicts the simulation results and a comparative method to prove the effectiveness of this approach. Finally, the last section gives conclusions of this proposed method.

**PMSM mathematical model**

PMSM is a complex nonlinear system. In order to better design advanced PMSM control algorithm, it is important to establish an appropriate mathematical model. The $d-q$ model for a PMSM in the rotating reference frame is very suitable for developing the proposed method. The equivalent circuit of a PMSM in rotating $d-q$ reference frame is depicted in Figure 1 (Chaoui and Sicard, 2011). In this figure, $R_s$ is stator resistance, $L_d$ and $L_q$ are frame stator inductances, $p_n$ denotes the number of pole pairs, $\psi_f$ is constant magnetic flux, $u_d$ and $u_q$ are $d-q$ frame stator voltages, $i_d$ and $i_q$ represent $d-q$ frame stator currents, $w_m$ is rotor mechanical velocity. In the $d-q$ reference frame, the electromagnetic moment equation of PMSM can be expressed as:

$$T_c = \frac{3}{2} p_n [(L_d - L_q) i_d i_q + \psi_f i_q] \tag{1}$$

The relation between rotor mechanical $w_m$ and the electromagnetic moment $T_c$ can be described as:

$$T_c = T_L + B_m w_m + \frac{J}{dt} \frac{dw_m}{dt} \tag{2}$$

which is regarded as the motion equation of PMSM. In this equation, $J$ is the moment of inertia, $B_m$ is the viscous friction, and $T_L$ is the load torque. It is worth noting that the electrical angular velocity of the motor $w_e = p_n w_m$. The unit of $w_m$ is rad/s. The mechanical velocity can also be represented by $N_r$, the unit of which is $r/min$(revolutions per minute) and $N_r = \frac{w_e}{p_n}$. From Figure 1, equation (1) and equation (2), the nonlinear differential equation model of a PMSM can be described in rotating $d-q$ reference frame as follows:

$$\begin{align*}
\frac{d i_d}{dt} &= -\frac{R_s}{L_d} i_d + \frac{L_d}{L_d} p_n w_m i_q + \frac{1}{L_d} u_d \\
\frac{d i_q}{dt} &= -\frac{R_s}{L_q} i_q - \frac{L_q}{L_q} p_n w_m i_d + \frac{1}{L_q} u_q - \frac{p_n w_m \psi_f}{L_q} \\
\frac{d w_m}{dt} &= \frac{3}{2} p_n [\psi_f + (L_d - L_q) i_d] i_q - \frac{B_m}{J} w_m - \frac{T_L}{J}
\end{align*} \tag{3}$$
If the state variable is defined as \( x = [i_d, i_q, w_m]^T \), and the control input variable is defined as \( u = [u_d, u_q]^T \), then the above model can be written as follows

\[
\dot{x} = f(x, u)
\]  

(4)

System (4) may be discretized using, for example, Euler or Runge-Kutta methods to obtain a discrete-time description of the system. We denote the discrete-time expression of the model as follows

\[
x(k + 1) = g(x(k), u(k))
\]  

(5)

PMSM is usually controlled by the \( i_d = 0 \) strategy. Then the constraints of the system are primarily reflected in the rotating \( d - q \) axis voltage constraint

\[
|u_{d,q}| \leq \frac{V_{dc}}{\sqrt{3}}
\]  

(6)

where \( V_{dc} \) denotes the DC-link voltage. It can be observed that the presented PMSM model contains nonlinear and cross-coupled terms. In the operation of a PMSM, the track- ing speed and the load torque depend on the requirements of different operating conditions, which may vary significantly. If we directly linearize the PMSM model at only one point, the introduced model error will be too large to neglect. Therefore, it is crucial to design an appropriate linearization model to reduce model errors resulted by different working conditions of the PMSM. The purpose of this work is to create a more accurate approximation of the actual PMSM model based on multi-point linearization. The explicit MPC algorithm is involved in controller design to put the optimization process offline so as to reduce online computational complexity resulted by multiple model.

**Explicit MPC**

Before designing the controller, we review the fundamentals of explicit MPC and present the explicit law of control variables in this section. For a linear time-invariant system with \( n \) state variables, \( m \) inputs, and \( p \) outputs, the discrete-state-space formulation at time instant \( k \) is given by

\[
\begin{cases}
    x(k + 1) = Ax(k) + Bu(k) \\
    y(k) = Cx(k)
\end{cases}
\]  

(7)

where \( x \in \mathbb{R}^n, u \in \mathbb{R}^m, \) and \( y \in \mathbb{R}^p \) represent the system state variable, input, and output, respectively. Suppose that the system is subject to input and output constraints expressed as \( y_{min} \leq y(k) \leq y_{max} \) and \( u_{min} \leq u(k) \leq u_{max} \). In MPC, the control objective is commonly described as a quadratic cost function

\[
\begin{aligned}
    & \min_u V(x, u) = \frac{1}{2} \sum_{l=0}^{N_p-1} [x(k + l)^T Q x(k + l) \\
    & + u(k + l)^T R u(k + l)]
\end{aligned}
\]  

(8)

where \( N_p \) is prediction horizon, \( Q \) and \( R \) are symmetric and non-negative weighting matrices which are used to penalize the state and the control input, respectively.

The prediction model used to compute the future state trajectories is as follows

\[
X(k) = \Lambda x(k) + \Phi U(k)
\]  

(9)

where \( X(k) = [x(k + 1)^T, x(k + 2)^T, \ldots, x(k + N_p)^T]^T \), \( U(k) = [u(k)^T, \ldots, u(k + N_p - 1)^T]^T \), \( \Lambda = \begin{bmatrix} A & A^2 & \cdots & A^N_p \end{bmatrix} \), \( \Phi = \begin{bmatrix} B & 0 & \cdots & 0 \\
AB & B & 0 & \cdots \\
& \ddots & \ddots & \ddots \\
& & \cdots & B \end{bmatrix} \). Then, the formulation of MPC can be described as equation (10)

\[
V(U(k), x(k)) = \frac{1}{2} x^T(k) Y x(k) + \min \left( \frac{1}{2} U(k)^T H U(k) + x^T(k) F U(k) \right)
\]

s.t. \( G U \leq W + S x(k) \)

where \( Y = \Lambda^T Q \Lambda, H = \Phi^T Q \Phi + \tilde{R}, F = \Lambda^T Q \Phi \), and

\[
\hat{Q} = \begin{bmatrix} Q & 0 & 0 \\
0 & \ddots & 0 \\
0 & 0 & \tilde{Q} \end{bmatrix} \text{ and } \hat{R} = \begin{bmatrix} R & 0 \\
0 & \ddots & 0 \\
0 & 0 & R \end{bmatrix}.
\]

The matrices \( G, W \) and \( E \) are decided by the system constraints. A new variable \( z \) needs to be defined such that \( z = U(k) + H^{-1} F^T x(k) \). Then, the MPC optimization problem can be rewritten as

\[
V_z(x) = \min \frac{1}{2} z^T H z
\]

s.t. \( G z \leq W + S x(k) \)

(11)

where \( S = E + G H^{-1} F^T \), \( V_z(x) = V(U(k), x(k)) - \frac{1}{2} x^T(k) (Y - FH^{-1} F^T) x(k) \). \( x(k) \) is the current state vector which can be treated as a vector of parameters. We use \( x \) instead of \( x(k) \) for ease of description in the following part. It is noticed that \( H > 0 \) because \( R > 0 \). The number of inequalities is expressed in terms of \( q \). The number of the free variables is \( n_z = m \times N_p \).

Thus, \( z \in \mathbb{R}^m, H \in \mathbb{R}^{m \times m}, G \in \mathbb{R}^{m \times q}, W \in \mathbb{R}^m, S \in \mathbb{R}^{m \times q} \). The above-mentioned MPC optimization problem can be converted to a multi-parametric quadratic programming(mp-QP) problem.

The mp-QP problem of equation (11) can be dealt with using the first-order Karush-Kuhn-Tucker(KKT) optimality conditions (Bemporad and Morari, 1999)

\[
H z + G^T \lambda = 0, \lambda \in \mathbb{R}^q
\]  

(12)

\[
\lambda_j(G z - W - S x) = 0, j = 1, \ldots, q
\]  

(13)

\[
\lambda \geq 0
\]  

(14)

\[
G z - W - S x = 0
\]  

(15)

where the index \( j \) represents the \( j \) th row and \( \lambda_j \) denotes Lagrange multipliers, \( j = 1, \ldots, q \). From equation (12)
z = -H^{-1}G^T\lambda 

(16)

Substitute the result into equation (13) to achieve the complementary slackness condition, which is \(\lambda (-GH^{-1}G^T\lambda - W - \tilde{S}x) = 0\). The lagrange multipliers corresponding to active and inactive constraints are indicated by \(\tilde{\lambda}\) and \(\lambda\), respectively. For inactive constraints \(\tilde{\lambda} = 0\) and for active constraints, \(-\tilde{G}H^{-1}G^T\lambda - W - \tilde{S}x = 0\). Thus

\[
\tilde{\lambda} = -(\tilde{G}H^{-1}G^T)^{-1}(\tilde{W} + \tilde{S}x)
\]

(17)

where \(\tilde{G}, \tilde{W}, \tilde{S}\) denote the rows of \(G, W, S\) corresponding to the set of active constraints. Therefore, the rows of \(\tilde{G}\) are linearly independent, which indicates that \((\tilde{G}H^{-1}G^T)^{-1}\) exists. To sum up, \(\lambda\) is an affine function of \(x\). Then, substitute equation (17) to equation (16) to get

\[
z = H^{-1}G^T(\tilde{G}H^{-1}G^T)^{-1}(\tilde{W} + \tilde{S}x)
\]

(18)

Obviously, \(z\) is an affine function of \(x\). The variable \(z\) should subject to the constraints of (15)

\[
GH^{-1}G^T(\tilde{G}H^{-1}G^T)^{-1}(\tilde{W} + \tilde{S}x) \leq W + Sx
\]

(19)

Moreover, the Lagrange multipliers in (17) should satisfy (14)

\[
-(\tilde{G}H^{-1}G^T)^{-1}(\tilde{W} + \tilde{S}x) \geq 0
\]

(20)

The inequalities (19) and (20) describe a polyhedron in the state space, which is regarded as the critical region \(CR_0\). Once \(CR_0\) has been created, the rest of the space has to be explored and new critical regions generated. The recursive algorithm for partitioning the state space is demonstrated in Dua and Pistikopoulos (2000). Therefore, online optimizations can be replaced by explicit control law, which is expressed as a piecewise affine (PWA) function. The PWA function is determined by a finite number of regions that partition state variables. Each region of the state variable corresponds to an explicit control input law. In each region, the PWA control function is expressed as

\[
U = z - H^{-1}F^Tx
\]

(21)

Obviously, the controller is a linear continuous PWA function of state vector. Therefore, the online computation is only about searching in a lookup table to find the region where the current state \(x(k)\) belongs to. Then, the corresponding \(u(x)\) can be returned. More detailed explanation about methods for a lookup table are available in TeNdel et al. (2003). In this work, Multi-Parametric Toolbox 3.0 (Herceg et al., 2013) is utilized to design the controller.

**Proposed controller based on multi-point linearization**

In this section, the proposed explicit controller base on multi-point linearization, model selection and model switching schedule, and specific controller design are demonstrated successively.

**Multi-point linearization**

The operating conditions of a PMSM vary significantly. This renders the operating range of a PMSM large. A single linear model is not sufficient to achieve good control performance. We may linearize the model at a few operating points to cover the wide operating range. These different operating conditions mainly contain the changes of tracking speed and load torque. In this work, we consider that the motor has \(M\) different working points, which means that \(M\) steady state points should be included. Therefore, \(M\) linearization models based on these steady state points should be involved, which is described as multi-point linearization.

Multi-point linearization modeling means that the PMSM model is linearized at a few steady state points. In other words, the PMSM model is a linear time-varying model after multi-point linearization. For explicit MPC, this method just increases offline computation time other than online computation time, which means this approach will not increase the online computational burden. The number of the linearization times is determined by the number of the stable operating points required by the PMSM system. Let us consider that there are \(M\) different working points. At the \(i\)-th working point \((x_{s,i}, u_{s,i})\),

\[
x_{s,i} = g(x_{s,i}, u_{s,i}), i = 1 \cdots M
\]

where \(g\) denotes the weight factor of the \(i\)-th working condition of PMSM. \(\alpha_i\) satisfies \(\sum_i \alpha_i = 1\). Define \(\delta x(k) = x(k) - x_{s,i}\) and \(\delta u(k) = u(k) - u_{s,i}\), the model turns to be

\[
\delta x(k + 1) = \sum_i \alpha_i \left( \frac{\partial g}{\partial x} |_{(x_{s,i}, u_{s,i})} \delta x(k) + \frac{\partial g}{\partial u} |_{(x_{s,i}, u_{s,i})} \delta u(k) \right)
\]

(24)

Equation (24) is the linear time-varying model after multi-point linearization. The matrices \(A_i = \frac{\partial g}{\partial x} |_{(x_{s,i}, u_{s,i})}\) and \(B_i = \frac{\partial g}{\partial u} |_{(x_{s,i}, u_{s,i})}\), \(i = 1 \cdots M\) are the input parameters of the MPT3.0 toolbox. The explicit MPC controller is designed based on this equation.

**Model selection and model switching schedule**

The key point of model selection is to choose appropriate working points to linearize the original model. In this work, we assume that the information about the working condition
change is known before the selection of operating points. Three different kinds of working conditions of PMSM are discussed. The selection of the linearization points for these three conditions are shown in Figure 2(a), (b), (c). Here, \( t_1 \) is the time when the working status of the motor changes and \( t_2 \) is the total simulation time. \( P_1 \) and \( P_2 \) are the selected points for two periods. Apparently, \( M = 2 \) in each case. The first working condition is that a sudden load torque is added to the motor system. From Figure 2(a), it is obvious that the steady state point corresponding to the point \( P_1 \) of the system is selected and the resulted linearized model is used from the beginning to time \( t_1 \). When the load torque \( T_{L1} \) is added, the steady state point corresponding to the point \( P_2 \) should be calculated and the second resulted linearized model is used from time \( t_1 \) to time \( t_2 \). The second condition includes a step change of the tracking speed from \( w_{\text{ref1}} \) to \( w_{\text{ref2}} \), which is shown in Figure 2(b). For the second condition, it is a proper behaviour that the steady state points at two different reference speeds corresponding \( P_1 \) and \( P_2 \) in Figure 2(b), are adopted to linearize the original model. The last working condition demonstrated in Figure 2(c) focuses on triangular shape speed reference, which means that the trajectory of the tracking speed goes up continuously and then it goes down gradually. The speed error cannot be entirely cancelled during acceleration due to system constraints (Bolognani et al., 2008). Apart from this, model mismatch is also a significant factor that affects the performance of PMSM. It is crucial to choose suitable linearization points to reduce the speed error as much as possible. According to the results shown in Bolognani et al. (2008), the actual speed is less than the tracking speed during uniform acceleration and the actual speed is greater than the tracking speed during uniform deceleration. Thus, the point at which the speed value is as large as possible is selected as the linearization point in the process of acceleration while the point at which the speed value is as small as possible is selected in the process of deceleration so as to keep the actual speed as close to the reference value as possible. As shown in Figure 2(c), we choose the point \( P_1 \), which is slightly less than the maximum tracking speed, and \( P_2 \), which is slightly greater than the maximum tracking speed to calculate the corresponding steady state points.

How to switch between these \( M \) models is worth to be considered. It is certain that the reference status of the motor at different time is known in advance. The weighting factor \( a_i \) is utilized to determine the working status of the motor. Define that

\[
\alpha_i = 1, i = 1 \cdots M
\]

when the motor is running in the \( i \)-th working condition. This means that the model is switched form one to another by adjusting the weight coefficient \( \alpha_i \).

**Controller design**

We assume that the motor is working in the \( i \)-th working status. For the \( i \)-th (\( i = 1 \cdots M \)) linearized model, \( \alpha_i = 1 \) and the MPC optimization problem is...
\[
\min_{\delta u} V_{/(\delta x, \delta u)} = \frac{1}{2} \sum_{j=0}^{N_p-1} \delta x[(k + h)Q \delta x(k + l)]
+ \delta u[(k + l)R \delta u(k + l)],
\]
\[
s.t. \delta x(k + l) = \frac{\partial g}{\partial x_{(i, l, u)}} \delta x(k) + \frac{\partial g}{\partial u_{(i, u)}} \delta u(k),
- \sqrt{3} - u_{x,l} \leq \delta u(k + l) \leq \sqrt{3} - u_{x,l}, l = 1, 2, \ldots, N_p - 1,
\delta x(0) = x(0) - x_{s,1},
\]
\[
x(0) = [0 \ 0 \ 0]^T
\]
(26)

where \( Q \) and \( R \) are state and control input weight matrices of the \( i \)-th cost function, \( N_p \) is the prediction horizon. From the cost function of the multi-point linearization model, an evident benefit is that the weight matrices can be adjusted with the change of steady state point. The non-cascade explicit MPC controller is applied to the system. The explicit MPC controller is composed of two parts: offline calculation and online searching in a lookup table.

The block diagram of the PMSM system is shown in Figure 3. The offline part is completed by MPT3.0 toolbox, which obtains the partition of the state variables and the corresponding explicit control laws of inputs. The offline calculation process mainly includes initialization and optimization. To be specific, the linear discrete model is regarded as the input of MPT3.0 toolbox at first. Then according to the requirements of the system, the constraints of the model are provided. Next, the system model and the presented constraints are utilized to realize the optimization calculation of MPC. Finally, the partition of the state variables and the PWA functions are generated and saved in a lookup table.

The online sequential search in a lookup table is the controller execution part. At time \( k \), the model switching part is utilized to decide the working condition of the motor so as to determine the points where the model is linearized. For the \( i \)-th working condition, the input value at time \( k \) is \( \delta x(k) \) which is calculated by subtracting \( x(k) \) from \( x_{s,i} \) and \( \delta u(k) \) is obtained according to the partition of \( \delta x(k) \). Thus, \( u(k) \) is acquired by adding \( u_{s,i} \) to \( \delta u(k) \) as the input of PMSM to achieve the state value at the next moment.

![Figure 3. Block diagram of the PMSM system.](image)

| Symbol          | Quantity                     | Value and unit |
|-----------------|------------------------------|----------------|
| \( R_s \)       | Stator resistance            | 0.958\( \Omega \) |
| \( L_d \)       | \( d \)-axis inductance      | 5.23\( \text{mH} \) |
| \( L_q \)       | \( q \)-axis inductances     | 12\( \text{mH} \) |
| \( p_n \)       | Number of poles              | 4              |
| \( J \)         | Moment of inertia           | 0.003\( \text{kg} \cdot \text{m}^2 \) |
| \( B_m \)       | Viscous damping              | 0.008\( \text{N} \cdot \text{m} \cdot \text{s/rad} \) |
| \( \psi_t \)    | Constant flux linkage       | 0.1827\( \text{Wb} \) |
| \( V_{dc} \)    | DC-link voltage             | 311\( \text{V} \) |

### Simulation results

In this section, we apply the proposed approach to a PMSM system to illustrate its applicability and effectiveness. The model of PMSM has been introduced above and the detailed parameters used in this section are displayed in Table 1.

Three different working conditions discussed in the previous section are adopted in the following simulations, including load torque change, tracking speed step change and tracking speed uniformly accelerated change. Case 1 simulates the working status that load torque grows from 0\( \text{N} \cdot \text{m} \) to another constant value. Case 2 implements the simulation of tracking speed step changes. Case 3 simulates the process of accelerating and decelerating. The number of steady state points \( M = 2 \) in each case. The simulation time is set as 0.1s and the tracking value of the PMSM system changes at 0.05s in all cases. Moreover, the simulation results of single-point linearization model is taken for comparison. \( N_p = 5, Q = \text{dig}(10, 20, 1000), R = 0.05I \) are used for single-point linearization of each case. For multi-point linearization, \( N_p = 5, Q_1 = \text{dig}(10, 20, 1000), Q_2 = \text{dig}(10, 20, 1500), R_1 = 0.05I(i = 1, \ldots, M) \) are set in case 1. \( N_p = 5, Q_3 = \text{dig}(10, 20, 1000), R_2 = 0.05I(i = 1, \ldots, M) \) are set in case 2, and \( N_p = 5, Q_4 = \text{dig}(10, 20, 1000), R_3 = 0.05I(i = 1, \ldots, M) \) are set in case 3.

The simulation results of the three cases are shown in Figure 4, 5, and 6, respectively, where speed, \( d - q \) axis currents and \( d - q \) axis voltage dynamic response curves are all...
displayed to show the performance of the motor. In Figure 4 and Figure 5, Figures (a), (c) and (e) are the results of the proposed method. Figures (b), (d) and (f) are the results of the comparison for the two methods. The simulation time from 0.045s to 0.075s is captured to clearly observe the difference between the results obtained from these two methods. The speed maximum fluctuation percentage and the regulation time of the two methods are displayed in Table 2 and Table 3. The stable state error is set to be $2\times10^{-6}$ to calculate the regulation time. ML and SL in the figures and tables represent multi-point linearization and single-point linearization respectively. For the last case, the comparisons of speed response curves and the absolute error of speed are shown in Figure 6. Besides, the mean absolute errors (MAE) shown in Table 4 are calculated to validate the proposed method.

From Figure 4, it is obvious that the proposed method can improve the performance of PMSM compared with single-point linearization model method when load torque changes from 0Nm to 20Nm. The comparison of speed response curves displayed in Figure 4(b), illustrating that the speed response using multi-point linearization can reach steady state faster.

### Table 2. Maximum fluctuation and regulation time of load torque step changes.

| (a) Load torque changes from 0Nm to 10Nm. | Method | Maximum fluctuation | Regulation time |
|----------------------------------------|--------|---------------------|-----------------|
| ML                                     | 2.72%  | 11.2ms              |
| SL                                     | 2.94%  | 12.6ms              |

| (b) Load torque changes from 0Nm to 20Nm. | Method | Maximum fluctuation | Regulation time |
|----------------------------------------|--------|---------------------|-----------------|
| ML                                     | 5.68%  | 8.4ms               |
| SL                                     | 7.63%  | 13.9ms              |

| (c) Load torque changes from 0Nm to 30Nm. | Method | Maximum fluctuation | Regulation time |
|----------------------------------------|--------|---------------------|-----------------|
| ML                                     | 8.29%  | 6.9ms               |
| SL                                     | 13.08% | 14.3ms              |
Figure 5. Results of case 2: Tracking speed changes from 1000 r/min to 1500 r/min.

Table 3. Maximum fluctuation and regulation time of tracking speed step changes.

(a) Tracking speed changes from 1000 r/min to 1500 r/min.

| Method | Maximum fluctuation | Regulation time |
|--------|---------------------|-----------------|
| ML     | 0.16%               | 17.7 ms         |
| SL     | 0.21%               | 18.4 ms         |

(b) Tracking speed changes from 1000 r/min to 800 r/min.

| Method | Maximum fluctuation | Regulation time |
|--------|---------------------|-----------------|
| ML     | 10.43%              | 20.5 ms         |
| SL     | 20.00%              | 22.4 ms         |

Table 4. MAE of speed with acceleration triangular speed reference.

(a) Acceleration speed changes from 10000 r/min/s to −10000 r/min/s.

| Method | MAE (r/min) |
|--------|-------------|
| ML     | 6.56        |
| SL     | 12.97       |

(b) Acceleration speed changes from 20000 r/min/s to −20000 r/min/s.

| Method | MAE (r/min) |
|--------|-------------|
| ML     | 10.60       |
| SL     | 26.31       |
The comparison of current responses as well as control input are presented in Figure 4(d) and Figure 4(f). The data of Table 2 illustrate the maximum fluctuation percentage and regulation time of load torque step changes. When the load torque changes from 0Nm to 10Nm, 20Nm and 30Nm, the corresponding maximum fluctuation percentage of speed response using multi-point linearization method are 2.72%, 5.68% and 8.29% respectively, which are less than that of speed response using single-point linearization method (2.94%, 7.63% and 13.08%). Besides, the regulation speed of presented method is faster than that of single-point linearization method observing the contrast between the three regulation times of these two methods. The results demonstrate that the presented approach can achieve considerable improvement especially when load torque changes significantly.

Furthermore, the results of step change of tracking speed changes are displayed in Figure 5. The proposed controller can track the changes of the reference speed more quickly than the controller using single-point linearization although the current contrast is not that obvious. The data from Table 3(a) clarify that using the proposed approach has 0.16% maximum fluctuation and 17.7ms regulation time, while using single-point linearization method has 0.21% maximum fluctuation and 18.4ms regulation time when tracking speed rises from 1000r/min to 1500r/min. The maximum fluctuation percentage and regulation time of tracking speed changes from 1000r/min to 800r/min are shown in Table 3(b). The results indicate that the proposed algorithm has better performance than single-point linearization whether the tracking speed takes an up step or a down step.

Finally, Figure 6 demonstrates the results of acceleration triangular speed reference. It can be seen from the speed error curve that the speed error using proposed approach is significantly less than that using single-point linearization model. The MAE of these two methods are depicted in Table 4, which provides further evidence that the presented controller has better performance. Besides, the higher the acceleration is, the more the performance is improved compared with the single-point linearization method.

Conclusions
This study focuses on the design and validation of explicit MPC of PMSM based on multi-point linearization. The proposed approach reduces the model error caused by the change of motor operating conditions, demonstrating the potential for enhancing the performance of PMSM system. The simulation results of three given different running conditions proved that the proposed explicit controller based on multi-point linearization can obtain better speed and current performance than that based on single-point linearization without increasing online computational burden. In the future, the differences of mutual inductance between the windings caused by
the change of motor position would be considered because the parameter mismatch will cause the deterioration of the performance of MPC. We intend to develop a new explicit controller based on this parameter-varying PMSM system.

**Declaration of conflicting interests**

The author(s) declared no potential conflicts of interest with respect to the research, authorship, and/or publication of this article.

**Funding**

The author(s) disclosed receipt of the following financial support for the research, authorship, and/or publication of this article: This work was supported by National Natural Science Foundation of China under Grant 61873113, the Postgraduate Research and Practice Innovation Program of Jiangsu Province under Grant SJKY19_2562 and China Scholarship Council.

**ORCID iDs**

Tianhong Pan https://orcid.org/0000-0002-0993-3937

Jinfeng Liu https://orcid.org/0000-0001-8873-847X

**References**

Bemporad A and Morari M (1999) Robust model predictive control: A survey. In: Robustness in Identification and Control, London. Springer, pp. 207–226.

Bemporad A, Morari M, Dua V, et al. (2002) The explicit linear quadratic regulator for constrained systems. *Automatica* 38(1): 3–20.

Bolognani S, Bolognani S, Peretti L, et al. (2008) Design and implementation of model predictive control for electrical motor drives. *IEEE Transactions on Industrial Electronics* 56(6): 1925–1936.

Chaoui H and Sicard P (2011) Adaptive fuzzy logic control of permanent magnet synchronous machines with nonlinear friction. *IEEE Transactions on Industrial Electronics* 59(2): 1123–1133.

Cychowski MT, Nalepa R and O’Mahony T (2005) Explicit model predictive control of a permanent magnet synchronous motor drive. In: 40th Universities Power Engineering Conference, Cork, Ireland, 7–9 September 2005. Cork, Ireland: University College Cork.

Dua V and Pistikopoulos EN (2000) An algorithm for the solution of multiparametric mixed integer linear programming problems. *Annals of Operations Research* 99(1–4): 123–139.

Ganapathy Ram A and Santha KR (2019) Design of an adaptive gain variation sliding mode control algorithm for extended non-singular terminal sliding mode observer based sensorless PMSM drive. *International Journal of Applied Research in Engineering* 14(1): 62–68.

Herczeg M, Kvasnicka M, Jones CN, et al. (2013) Multi-parametric toolbox 3.0. In: 2013 European Control Conference (ECC), Zurich, Switzerland, 17–19 July 2013, pp. 502–510. Piscataway, USA: IEEE.

Jia C, Wang X, Liang Y, et al. (2019) Robust current controller for IPMSM drives based on explicit model predictive control with online disturbance observer. *IEEE Access* 7: 45898–45910.

Jung JW, Leu VQ, Do TD, et al. (2014) Adaptive PID speed control design for permanent magnet synchronous motor drives. *IEEE Transactions on Power Electronics* 30(2): 900–908.

Koiwa K, Kuribayashi T, Zanma T, et al. (2019) Optimal current control for PMSM considering inverter output voltage limit: Model predictive control and pulse-width modulation. *IET Electric Power Applications* 13(12): 2044–2051.

Lara J, Xu J and Chandra A (2016) Effects of rotor position error in the performance of field-oriented-controlled PMSM drives for electric vehicle traction applications. *IEEE Transactions on Industrial Electronics* 63(8): 4738–4751.

Liu S, Mao Y and Liu J (2019) Model-predictive control with generalized zone tracking. *IEEE Transactions on Automatic Control* 64(11): 4698–4704.

Mani P, Rajan R, Shanmugam L, et al. (2018) Adaptive fractional fuzzy integral sliding mode control for PMSM model. *IEEE Transactions on Fuzzy Systems* 27(8): 1674–1686.

Mayne DQ, Rawlings JB, Rao CV, et al. (2020) Constrained model predictive control: Stability and optimality. *Automatica* 36(6): 789–814.

Mwasili F, Nguyen HT, Choi HH, et al. (2016) Finite set model predictive control of interior pm synchronous motor drives with an external disturbance rejection technique. *IEEE/ASME Transactions on Mechatronics* 22(2): 762–773.

Mynar Z, Vesely L and Vaclavek P (2016) PMSM model predictive control with field-weakening implementation. *IEEE Transactions on Industrial Electronics* 63(8): 5156–5166.

Sreejith R and Singh B (2019) Intelligent nonlinear sensorless predictive field oriented control of PMSM drive for three wheeler hybrid solar PV-battery electric vehicle. In: 2019 IEEE Transportation Electrification Conference and Expo (ITEC), Michigan, USA, June 2019, pp. 1–6. New York, USA: IEEE.

Sun Z, Zhang Y, Li S, et al. (2020) A simplified composite current-constrained control for permanent magnet synchronous motor speed-regulation system with time-varying disturbances. *Transactions of the Institute of Measurement and Control* 42(3): 374–385.

TøNdel P, Johansen TA and Bemporad A (2003) An algorithm for multi-parametric quadratic programming and explicit MPC solutions. *Automatica* 39(3): 489–497.

Velagić J and Šabić B (2014) Design, implementation and experimental validation of explicit MPC in programmable logic controller. In: 2014 IEEE 23rd International Symposium on Industrial Electronics (ISIE), Istanbul, Turkey, June 2014, pp. 93–98. Piscataway, USA: IEEE.

Wang Y, Feng Y, Zhang X, et al. (2019) A new reaching law for anti-disturbance sliding-mode control of PMSM speed regulation system. *IEEE Transactions on Power Electronics* 35(4): 4117–4126.

Wu M, Sun X, Zhu J, et al. (2020) An improved model predictive torque control for PMSM drives based on duty cycle optimization. *IEEE Transactions on Magnetics*. DOI: 10.1109/TMAG.2020.3008495.

Yi L., Zhang C and Wang G (2017) Research of self-tuning PID for PMSM vector control based on improved KMTOA. *International Journal of Intelligent Systems and Applications* 9(3): 60–67.

Zanma T, Tozawa S, Takagi Y, et al. (2020) Optimal voltage vector in current control of PMSM considering torque ripple and reduction of the number of switching operations. *IET Power Electronics* 13(5): 1200–1206.

Zhang X, Zhang L and Zhang Y (2019) Model predictive current control for PMSM drives with parameter robustness improvement. *IEEE Transactions on Power Electronics* 34(2): 1645–1657.

Zhu W, Chen D, Du H, et al. (2020) Position control for permanent magnet synchronous motor based on neural network and terminal sliding mode control. *Transactions of the Institute of Measurement and Control* 42(9): 1632–1640.
