Online MTPA Control of IPMSM Based on Robust Numerical Optimization Technique

Hyeon-Sik Kim, Member, IEEE, Younggi Lee, Member, IEEE, Seung-Ki Sul, Fellow, IEEE, Jayeong Yu, and Jaeyoon Oh

Abstract—This paper presents an online maximum torque per ampere (MTPA) control that considers both magnetic saturation and cross-magnetization effects of interior permanent-magnet synchronous machines (IPMSMs). For IPMSM drives, especially in torque-controlled applications, torque accuracy and high-efficiency operations are important issues. They can be dealt as a constrained optimization problem to satisfy both torque reference tracking and loss-minimizing operation. In this paper, nonlinear simultaneous equations are derived from Lagrange multiplier method, which could be solved by numerical algorithms. Among them, Levenberg–Marquardt algorithm (LMA) is employed to guarantee a robust calculation of optimal current references. It is optimized to alleviate calculation burden while maintaining the stability of the proposed algorithm. In addition, a torque reference limiter is implemented to satisfy a current limit in real time. The feasibility of the proposed method is verified under various operating conditions by simulation and experimental results. Through the proposed algorithms, accurate MTPA control is achieved under not only unsaturated but also highly saturated operating conditions.

Index Terms—Interior permanent-magnet synchronous machines (IPMSM), maximum torque per ampere (MTPA), Levenberg–Marquardt algorithm (LMA), loss minimization, robust numerical algorithm, torque control.

I. INTRODUCTION

INTERIOR permanent-magnet synchronous machines (IPMSMs) have received much attention in a variety of applications, thanks to their excellent features, such as high efficiency, high power density, and extended speed range. For IPMSM drives, accurate torque control and high-efficiency operation are always emphasized for improved and cost-effective performance. Thus, maximum torque per ampere (MTPA) operation should be applied below the base speed to minimize the current magnitude while satisfying torque accuracy. However, it is difficult to find optimal current commands due to the influence of magnetic saturation, cross-magnetization effects, and their variation according to the operating conditions [1], [2].

There have been lots of researches on MTPA operation and torque accuracy [3]–[19]. Generally, look-up table (LUT) methods are adopted for IPMSM torque control to avoid complex calculation in real time [3], [4]. However, the construction of LUTs requires costly and time-consuming process. In addition, the LUT methods require a large memory usage and an accurate interpolation algorithm. To overcome these constraints, online MTPA searching methods have been studied to reflect operating conditions without premade LUTs [5]–[19].

MTPA searching algorithms can be classified into two groups: perturbation searching methods and calculation methods based on a mathematical model. Perturbation searching methods trace the fluctuation of output torque after injecting current angle perturbation, which has been improved to high-frequency current injection methods [5]–[8]. It is capable of keeping MTPA condition regardless of machine parameter variations. However, dynamic performance is hard to satisfy the specification of industrial and automotive applications, i.e., too slow to trace rapid changes of a torque reference.

Besides, virtual signal injection methods were proposed where the current signal is virtually injected to a signal processing block instead of the real machine, i.e., IPMSM [9]–[11]. It can eliminate additional power losses and torque variation provoked by perturbation and observation method. However, it did not consider the machine parameter variations with respect to currents, which would result in MTPA tracking error due to nonlinear characteristics of IPMSM [11]. Moreover, these methods have been designed for speed-control applications, which cannot guarantee torque accuracy of torque-control applications such as a traction motor of the electric vehicle.

Otherwise, the calculation methods have been developed based on the mathematical model of IPMSM, where Lagrange method is usually adopted to derive optimal current references [12]–[19]. It can offer satisfactory dynamic performance and torque accuracy by reflecting accurate parameters in real time. Generally, a fourth-order polynomial was derived where static inductances and magnet flux linkage model, i.e., \( L_{ds}, L_{qs}, \) and \( \lambda_f \), are utilized [12]. It can be solved by approximating the fourth-order polynomial with a second-order one or utilizing...
Ferrari’s method [13]–[15]. However, these methods can provoke approximation error or high computational burden due to complex calculations.

In addition, the fourth-order polynomial did not consider the inductance variations with respect to currents, e.g., \( \frac{\partial L_{dq}}{\partial i_d} \) and \( \frac{\partial L_{qq}}{\partial i_q} \). Therefore, MTPA operation would not be achieved for IPMSM with extremely nonlinear parameters. In [16], a correction algorithm was proposed to remove the discrepancy between the calculated and real MTPA point. In [17] and [18], a curve-fitting method was adopted to model the inductance and magnet flux linkage. However, these methods require high calculation burden or parameter curve fittings. Furthermore, it is well known that the simultaneous estimation of both static inductances and permanent magnet (PM) flux linkage with reasonable accuracy is difficult [20].

In [19], Newton’s method was utilized to solve equations based on the flux linkages, which requires the flux linkages and dynamic inductances. It has many advantages compared to the other methods. First, Newton’s method has small calculation burden compared to analytical approaches, e.g., Ferrari’s method. Second, the discrepancy in MTPA operation does not appear because the flux variations with respect to currents are considered by applying the concept of dynamic inductances. In addition, the flux linkages and dynamic inductances are relatively easy to estimate as compared to the static inductances and PM flux linkage. Thus, the online calculation could reflect the flux linkage and dynamic inductance variations in real time by applying online parameter estimation algorithms at once [21]–[23].

However, the convergence of Newton’s method cannot be guaranteed under harsh conditions, e.g., short-period overload operation [24]. Consequently, there is a risk of divergence under a highly saturated operating condition of IPMSM when Newton’s method is applied. In addition, the cross-magnetization also cannot be ignored especially under the saturated operation, i.e., under a heavy-load condition, due to the increase of cross-coupling inductance ratio [25]. Therefore, the robust numerical algorithm should be applied considering the cross-coupling inductances to extract the optimal current references under any operating conditions. Besides, a torque command was not limited in the process of the calculation, which would incur large current references. However, the current should be limited by motor and inverter current rating or active thermal management strategy [26].

In this paper, an online MTPA control based on the numerical optimization technique is proposed where the magnetic saturation and cross-magnetization effects are fully considered. To alleviate ill-convergence risks under the saturated operating condition, Levenberg–Marquardt algorithm (LMA) is employed, which guarantees a robust calculation of current references under high-torque references. In addition, a torque reference limiter is implemented to satisfy a current limit, which could be modified in real time to accommodate active thermal management of inverter and/or motor. Through simulation and experimental results, the validity of the proposed methods is verified for various operating conditions, especially under the highly saturated condition of IPMSM. The contents of this research were previously presented in [27], which was revised for better understanding. In addition, the concepts of a condition number and the torque reference limiter are introduced to clarify the effectiveness of the proposed algorithm.

### II. IPMSM Modeling

In IPMSM, the flux linkages, \( \lambda_{dq} \equiv (\lambda_d, \lambda_q) \), vary depending on \( d \)- and \( q \)-axis currents, \( i_{dq} \equiv (i_d, i_q) \). It is well known that flux variations are highly nonlinear under the load condition due to the magnetic saturation. In addition, the magnetic saturation induces the cross magnetization in IPMSM, which means a mutual influence between \( d \)- and \( q \)-axis flux linkages excited by PM and currents [28]. This mutual correlation is usually referred to as the cross-coupling effect. Thus, the flux linkages are hard to express as a linear function and rather easy to use a nonlinear flux model as they are. The stator voltage equations of IPMSM in the rotor reference frame can be represented as

\[
\begin{align*}
\dot{v}_d &= R_s i_d + s \lambda_d - \omega_r \lambda_q \\
\dot{v}_q &= R_s i_q + s \lambda_q + \omega_r \lambda_d
\end{align*}
\]

(1)

where \( R_s \) and \( \omega_r \) denote a stator resistance and electrical rotor speed; \( \dot{v}_d \) and \( \dot{v}_q \) are \( d \)- and \( q \)-axis stator voltages, respectively.

The electromagnetic torque \( T_e \) can be expressed by the cross product of \( \lambda_{dq} \) and \( i_{dq} \) as

\[
T_e = \frac{3P}{2} (\lambda_d i_q - \lambda_q i_d)
\]

(2)

where \( P \) is the number of pole pairs.

When the magnetic saturation and cross-coupling effects are taken into account, the flux linkage model of IPMSM and its time derivative in the rotor reference frame can be expressed as (3) and (4) near an operating point, where \( L_{dd}, L_{dq}, L_{qd}, \) and \( L_{qq} \) denote the dynamic inductances; and \( \lambda_d \) and \( \lambda_q \) are \( d \)- and \( q \)-axis flux linkages at the operating point \( (i_{d0}, i_{q0}) \)

\[
\begin{align*}
\dot{\lambda}_d (i_d, i_q) &= L_{dd} \left( i_d - i_{d0} \right) + L_{dq} \left( i_q - i_{q0} \right) + \lambda_d (i_{d0}, i_{q0}) \\
\dot{\lambda}_q (i_d, i_q) &= L_{qd} \left( i_d - i_{d0} \right) + L_{qq} \left( i_q - i_{q0} \right) + \lambda_q (i_{d0}, i_{q0})
\end{align*}
\]

(3)

\[
\begin{align*}
\frac{d\lambda_d}{dt} &= L_{dd} \frac{di_d}{dt} + L_{dq} \frac{di_q}{dt} \\
\frac{d\lambda_q}{dt} &= L_{qd} \frac{di_d}{dt} + L_{qq} \frac{di_q}{dt}
\end{align*}
\]

(4)

Eq. (3) could be comprehended as the linearized flux linkage model near the operating point, i.e., small-signal model, which depends on both \( i_d \) and \( i_q \). The dynamic inductance \( L_{dq} \) can be defined as the rate of change for \( \lambda_{dq} \) with respect to \( i_{dq} \) as [29]

\[
L_{dq} = \begin{bmatrix}
\frac{\partial \lambda_d}{\partial i_d} & \frac{\partial \lambda_d}{\partial i_q} \\
\frac{\partial \lambda_q}{\partial i_d} & \frac{\partial \lambda_q}{\partial i_q}
\end{bmatrix} = \begin{bmatrix}
L_{dd} & L_{dq} \\
L_{qd} & L_{qq}
\end{bmatrix}
\]

\[
\begin{bmatrix}
\lambda_d (i_{d0} + \Delta i_d, i_{q0} + \Delta i_q) - \lambda_d (i_{d0}, i_{q0}) \\
\lambda_q (i_{d0} + \Delta i_d, i_{q0} + \Delta i_q) - \lambda_q (i_{d0}, i_{q0})
\end{bmatrix}
\]

\[
\begin{bmatrix}
L_{dd} \\
L_{qd}
\end{bmatrix}
\]

(5)

The flux linkages with \( d \)- and \( q \)-axis currents of IPMSM for automotive application are illustrated in Fig. 1. It is obtained by using an FEA software where the parameters are presented.
on a per-unit (p.u.) basis. The base current is defined by a peak current where a maximum torque is produced for a short period of time. Fig. 1 shows that \( \lambda_d \) varies with not only \( i_d \) but also \( i_q \), and \( \lambda_q \) does similarly. It reveals that the cross-coupling effect as well as saturation cannot be negligible under the heavy-load condition. The dynamic inductances can be extracted from flux linkage maps, which are defined as partial derivatives of the flux linkages as (5). The cross-coupling effect can be modeled as cross-coupling inductances, \( L_{dq} \) and \( L_{qd} \), which are also utilized for the online MTPA control along with \( L_{dd} \) and \( L_{qq} \).

Fig. 2 shows the electromagnetic torque on \( d \)- and \( q \)-axis current where the maximum torque is defined as 1 p.u. The MTPA curve is indicated as a red dashed line where the locus of constant current magnitude is tangential to the constant torque locus. It reveals that the current angle at maximum MTPA point is \( 45^\circ \), where the saturation and cross-coupling effect are very severe. Thus, the piecewise linear flux-linkage model, i.e., the flux linkages and dynamic inductances, could be a practical solution to follow the MTPA curve under the heavily saturated operating conditions.

III. ONLINE MTPA CONTROL

A. Derivation of MTPA Equations

The efficient operation can be achieved by minimizing the losses of motor, the sum of copper loss and iron loss. The copper loss \( P_{Cu} \) is dominant than the iron loss \( P_{Fe} \), especially in low-speed area, where the iron loss is almost negligible. Therefore, MTPA control by online calculation could be adopted as suboptimal solution especially when operating under the base speed. Furthermore, because the extraction of iron loss requires much computational power in operation, minimizing the copper loss could be regarded as a practical suboptimal solution.

In the MTPA region, the loss minimization is formulated as a nonlinear constrained optimization problem defined as follows:

\[
\min i_d^2 + i_q^2, \quad \text{subject to} \quad T_e^* = \frac{3P}{2} (\lambda_d i_q - \lambda_q i_d) \quad (6)
\]

where \( T_e^* \) denotes the torque reference.

It can be solved by Lagrange multiplier method, where a Lagrangian function is defined as (7) and \( \mu \) is a Lagrange multiplier [19]

\[
\mathcal{L}(i_d, i_q, \mu) = i_d^2 + i_q^2 + \mu \left( \frac{3P}{2} (\lambda_d i_q - \lambda_q i_d) - T_e^* \right). \quad (7)
\]

Necessary conditions for optimization can be extracted as

\[
\begin{align*}
0 &= \frac{\partial \mathcal{L}}{\partial i_d} = \frac{3P}{4} (\lambda_d i_q - \lambda_q i_d) - T_e^* \\
0 &= \frac{\partial \mathcal{L}}{\partial i_q} = 2i_d + \mu \frac{3P}{2} (L_{dd} i_q - \lambda_q - L_{qd} i_d) \\
0 &= \frac{\partial \mathcal{L}}{\partial \mu} = 2i_q + \mu \frac{3P}{2} (L_{dq} i_q - L_{qq} i_d)
\end{align*}
\]

where the partial derivatives of \( \lambda_d \) and \( \lambda_q \) are defined as \( L_{dq} \).

It can be simplified to a pair of equations with two unknowns, \( i_d \) and \( i_q \), by eliminating \( \mu \) because the flux linkages and dynamic inductances are defined as the functions of \( i_d \) and \( i_q \). As a result, constant torque curve is defined as (9) and MTPA curve is (10). They are represented by \( f(i_d, i_q) \) and \( g(i_d, i_q) \), respectively

\[
\begin{align*}
0 &= (\lambda_d i_q - \lambda_q i_d) - \frac{2T_e^*}{3P} = f(i_d, i_q) \quad (9) \\
0 &= (\lambda_d L_{qq} i_d + L_{dq} i_q) i_d + (\lambda_q + L_{qd} i_d - L_{dd} i_q) i_q = g(i_d, i_q). \quad (10)
\end{align*}
\]

Fig. 3(a) shows \( f(i_d, i_q) \) on \( d \)- and \( q \)-axis current plane when \( T_e^* = 1 \) p.u., which shows that the torque reference can be tracked by solving \( f(i_d, i_q) = 0 \). Similarly, Fig. 3(b) shows
A convergence speed is linear and often very slow near \( \mathbf{x}^* \) because \( \nabla E(\mathbf{x}_k) \) is close to 0 near \( \mathbf{x}^* \) with fixed \( \alpha_k \). The rate of convergence could be increased by using curvature information of \( E(\mathbf{x}) \). However, the curvature of \( E(\mathbf{x}) \) is not utilized in this method. This makes it hard to set \( \alpha_k \), which should be modified at each iteration to increase the speed.

Gauss–Newton algorithm (GNA) was proposed where the curvature as well as gradient information is used to improve the convergence speed. It is a modification of Newton’s method, which derives same iteration function as

\[
\mathbf{x}_{k+1} = \mathbf{x}_k - \mathbf{H}(\mathbf{x}_k)^{-1}\nabla E(\mathbf{x}_k) = \mathbf{x}_k - \mathbf{J}(\mathbf{x}_k)^{T} F(\mathbf{x}_k) \tag{15}
\]

where \( \mathbf{H}(\mathbf{x}_k) \) is the Hessian matrix of \( F(\mathbf{x}_k) \) defined as

\[
\mathbf{J}(\mathbf{x}_k) = \begin{bmatrix}
\frac{\partial f(i_d,i_q)}{\partial i_d} & \frac{\partial f(i_d,i_q)}{\partial i_q} \\
\frac{\partial g(i_d,i_q)}{\partial i_d} & \frac{\partial g(i_d,i_q)}{\partial i_q}
\end{bmatrix}.
\tag{14}
\]

A convergence speed is linear and often very slow near \( \mathbf{x}^* \) because \( \nabla E(\mathbf{x}_k) \) is close to 0 near \( \mathbf{x}^* \) with fixed \( \alpha_k \). The rate of convergence could be increased by using curvature information of \( E(\mathbf{x}) \). However, the curvature of \( E(\mathbf{x}) \) is not utilized in this method. This makes it hard to set \( \alpha_k \), which should be modified at each iteration to increase the speed.

B. Calculation of Current References

Solving simultaneous equations can be redefined as an unconstrained optimization problem to minimize the objective function \( E(\mathbf{x}) \), which means the sum of squared error. \( E(\mathbf{x}) \) is defined as

\[
E(\mathbf{x}) = \mathbf{F}(\mathbf{x})^T \mathbf{F}(\mathbf{x}) = f(\mathbf{x})^2 + g(\mathbf{x})^2
\tag{11}
\]

where \( \mathbf{x} \) is a current vector called parameters, which is defined as \( \mathbf{i}_{dq} = [i_d, i_q]^T \); and \( \mathbf{F}(\mathbf{x}) = [f(i_d,i_q), g(i_d,i_q)]^T \).

To find the minimum point of \( E(\mathbf{x}) \), i.e., \( \mathbf{x}^* \), numerical optimization methods for solving nonlinear least square problems can be applied. It is an iterative process that starts from the operating point \( \mathbf{x}_0 \) and converges to \( \mathbf{x}^* \).

Fig. 4 shows \( E(i_d, i_q) \) on \( d \)- and \( q \)-axis current plane when \( T_s^* = 1 \) p.u.. The intersection of \( f(\mathbf{x}) = 0 \) and \( g(\mathbf{x}) = 0 \) is presented by the point at \( E(\mathbf{x}) = 0 \), which is the minimum value of \( E(\mathbf{x}) \). There is only one minimum point in the objective function, i.e., no local minima. Thus, the numerical algorithms to find the local minimum can be applied without difficulty. It is implemented based on the local parameters at the operating point, i.e., \( \lambda_{dq} \) and \( L_{dq} \).

There have been various techniques proposed to find the minimum of the objective function [30]–[32]. Gradient descent is the simplest and intuitive technique where the parameters are updated by subtracting a scaled gradient at each step as follows:

\[
\mathbf{x}_{k+1} = \mathbf{x}_k - \alpha_k \cdot \nabla E(\mathbf{x}_k) = \mathbf{x}_k - \alpha_k \cdot \mathbf{J}(\mathbf{x}_k)^{T} \mathbf{F}(\mathbf{x}_k) \tag{12}
\]

where \( \alpha_k \) is a step length. \( \nabla E(\mathbf{x}_k) \) means the gradient of \( E(\mathbf{x}_k) \) and \( \mathbf{J}(\mathbf{x}_k) \) is the Jacobian matrix of \( \mathbf{F}(\mathbf{x}_k) \), which are, respectively, defined as

\[
\nabla E(\mathbf{x}_k) = \mathbf{J}(\mathbf{x}_k)^{T} \mathbf{F}(\mathbf{x}_k)
\tag{13}
\]

\[
\mathbf{J}(\mathbf{x}_k) = \begin{bmatrix}
\frac{\partial f(i_d,i_q)}{\partial i_d} & \frac{\partial f(i_d,i_q)}{\partial i_q} \\
\frac{\partial g(i_d,i_q)}{\partial i_d} & \frac{\partial g(i_d,i_q)}{\partial i_q}
\end{bmatrix}.
\tag{14}
\]

A convergence speed is linear and often very slow near \( \mathbf{x}^* \) because \( \nabla E(\mathbf{x}_k) \) is close to 0 near \( \mathbf{x}^* \) with fixed \( \alpha_k \). The rate of convergence could be increased by using curvature information of \( E(\mathbf{x}) \). However, the curvature of \( E(\mathbf{x}) \) is not utilized in this method. This makes it hard to set \( \alpha_k \), which should be modified at each iteration to increase the speed.

There have been various techniques proposed to find the minimum of the objective function [30]–[32]. Gradient descent is the simplest and intuitive technique where the parameters are updated by subtracting a scaled gradient at each step as follows:

\[
\mathbf{x}_{k+1} = \mathbf{x}_k - \alpha_k \cdot \nabla E(\mathbf{x}_k) = \mathbf{x}_k - \alpha_k \cdot \mathbf{J}(\mathbf{x}_k)^{T} \mathbf{F}(\mathbf{x}_k) \tag{12}
\]

where \( \alpha_k \) is a step length. \( \nabla E(\mathbf{x}_k) \) means the gradient of \( E(\mathbf{x}_k) \) and \( \mathbf{J}(\mathbf{x}_k) \) is the Jacobian matrix of \( \mathbf{F}(\mathbf{x}_k) \), which are, respectively, defined as

\[
\nabla E(\mathbf{x}_k) = \mathbf{J}(\mathbf{x}_k)^{T} \mathbf{F}(\mathbf{x}_k)
\tag{13}
\]

\[
\mathbf{J}(\mathbf{x}_k) = \begin{bmatrix}
\frac{\partial f(i_d,i_q)}{\partial i_d} & \frac{\partial f(i_d,i_q)}{\partial i_q} \\
\frac{\partial g(i_d,i_q)}{\partial i_d} & \frac{\partial g(i_d,i_q)}{\partial i_q}
\end{bmatrix}.
\tag{14}
\]

A convergence speed is linear and often very slow near \( \mathbf{x}^* \) because \( \nabla E(\mathbf{x}_k) \) is close to 0 near \( \mathbf{x}^* \) with fixed \( \alpha_k \). The rate of convergence could be increased by using curvature information of \( E(\mathbf{x}) \). However, the curvature of \( E(\mathbf{x}) \) is not utilized in this method. This makes it hard to set \( \alpha_k \), which should be modified at each iteration to increase the speed.

Gauss–Newton algorithm (GNA) was proposed where the curvature as well as gradient information is used to improve the convergence speed. It is a modification of Newton’s method, which derives same iteration function as

\[
\mathbf{x}_{k+1} = \mathbf{x}_k - \mathbf{H}(\mathbf{x}_k)^{-1}\nabla E(\mathbf{x}_k) = \mathbf{x}_k - \mathbf{J}(\mathbf{x}_k)^{T} \mathbf{J}(\mathbf{x}_k) \tag{15}
\]

where \( \mathbf{H}(\mathbf{x}_k) \) is the Hessian matrix of \( \mathbf{F}(\mathbf{x}_k) \) defined as

\[
\mathbf{J}(\mathbf{x}_k) = \begin{bmatrix}
\frac{\partial f(i_d,i_q)}{\partial i_d} & \frac{\partial f(i_d,i_q)}{\partial i_q} \\
\frac{\partial g(i_d,i_q)}{\partial i_d} & \frac{\partial g(i_d,i_q)}{\partial i_q}
\end{bmatrix}.
\tag{14}
\]

A convergence speed is linear and often very slow near \( \mathbf{x}^* \) because \( \nabla E(\mathbf{x}_k) \) is close to 0 near \( \mathbf{x}^* \) with fixed \( \alpha_k \). The rate of convergence could be increased by using curvature information of \( E(\mathbf{x}) \). However, the curvature of \( E(\mathbf{x}) \) is not utilized in this method. This makes it hard to set \( \alpha_k \), which should be modified at each iteration to increase the speed.

Gauss–Newton algorithm (GNA) was proposed where the curvature as well as gradient information is used to improve the convergence speed. It is a modification of Newton’s method, which derives same iteration function as

\[
\mathbf{x}_{k+1} = \mathbf{x}_k - \mathbf{H}(\mathbf{x}_k)^{-1}\nabla E(\mathbf{x}_k) = \mathbf{x}_k - \mathbf{J}(\mathbf{x}_k)^{T} \mathbf{J}(\mathbf{x}_k) \tag{15}
\]

where \( \mathbf{H}(\mathbf{x}_k) \) is the Hessian matrix of \( \mathbf{F}(\mathbf{x}_k) \) defined as

\[
\mathbf{J}(\mathbf{x}_k) = \begin{bmatrix}
\frac{\partial f(i_d,i_q)}{\partial i_d} & \frac{\partial f(i_d,i_q)}{\partial i_q} \\
\frac{\partial g(i_d,i_q)}{\partial i_d} & \frac{\partial g(i_d,i_q)}{\partial i_q}
\end{bmatrix}.
\tag{14}
\]
Fig. 5. Current reference trajectories depending on calculation algorithm. (a) Gradient descent. (b) GNA. (c) LMA.

\[ x_{k+1} = x_k - \left\{ H(x_k) + \mu_k \cdot \text{diag}(H(x_k)) \right\}^{-1} \nabla E(x_k) \]
\[ = x_k - \left\{ H(x_k) + \mu_k \cdot \text{diag}(H(x_k)) \right\}^{-1} J(x_k)^T F(x_k) \]

where \( \mu_k \) is the damping coefficient. \( \mu_k \) is multiplied by the diagonal of Hessian to scale each component of the gradient.

LMA is more robust than GNA due to the damping factor \( \mu_k \cdot \text{diag}(H(x_k)) \), which helps \( x_k \) converge into \( x^* \) even if \( x_0 \) are far away from \( x^* \). Moreover, LMA can be explained as GNA with a trust-region method [30]. It means that LMA alleviates the ill-convergence risk by restricting a step size within the trust region. LMA commonly operates as gradient descent when \( x_k \) is far from \( x^* \), whereas it operates as GNA when \( x_k \) is closed to \( x^* \). Thus, it is appropriate for the highly saturated condition where the fast and robust numerical algorithm is required to extract accurate current references.

Fig. 5 shows current reference trajectories depending on the calculation algorithm when \( T^* \) is 1 p.u. and an iteration number \( N \) is 5. \( x_0 \) is marked as a cross and \( x_N \) as a circle. \( x_N \) should be approached to \( x^* = [-0.69, 0.72]^T \) wherever \( x_0 \) is located after several iterations. In Fig. 5(a), where gradient descent is applied and \( \alpha_k = 200 \), \( x_k \) converges to \( x^* \) too slowly, or stays far away from \( x^* \) depending on \( x_0 \). It shows that the performance of gradient descent is highly dependent on \( \alpha_k \). Therefore, \( \alpha_k \) should be modified in accordance with \( x^* \) and \( x_k \), which is difficult to calculate in real time. After applying GNA, \( x_k \) cannot converge to \( x^* \) because \( x_0 \) is far from \( x^* \), as shown in Fig. 5(b). In addition, it reveals oscillatory \( x_k \) near \( x^* \) since \( J(x_k) \) is ill conditioned under the highly saturated operating condition. However, \( x_k \) can converge to \( x^* \) regardless of \( x_0 \) and \( T^* \) by applying LMA, as shown in Fig. 5(c), where \( \mu_k = 1 \). It shows that applying LMA is helpful to stabilize the online MTPA control.

The numerical algorithms could be compared in aspect of the condition number, which shows its sensitivity to small perturbation of \( J(x) \). The condition number of LMA \( \kappa_{\mu}(J) \) is defined as

\[ \kappa_{\mu}(J) = ||H(x) + \mu \cdot \text{diag}(H(x))|| \cdot ||J(x)|| \]

where ||·|| means the norm and \( \mu \) is the damping coefficient [31].

Small \( \kappa_{\mu}(J) \) means that the iteration function is more robust to the errors in \( J(x) \). Fig. 6 shows \( \kappa_{\mu}(J) \) when \( \mu \) is set to 0 and 1, respectively, under the same condition with that of Fig. 5. The condition number of GNA is equal to that of LMA when \( \mu = 0 \). It shows that \( \kappa_{\mu}(J) \) is smaller as \( \mu \) is larger. It means that a large \( \mu \) can dampen the errors in \( J(x) \) as well as the effects of nonlinear residual, i.e., nonlinear characteristics of \( F(x) \). In summary, LMA with large \( \mu \) is more robust to \( F(x) \) variations compared with GNA.
C. Implementation of Calculation Algorithm

The damping coefficient $\mu_k$ can be modified according to the changes of the error during iteration process, which varies as a result of the update or degree of the saturation. The size of $\mu_k$ can be updated by a gain ratio, which is the ratio between the actual and predicted decrease in the objective function [32]. However, it requires extra updating process during operations. Thus, $\mu_k$ is being held constant in the DSP, which should be fixed to a minimum value to stabilize the online MTPA control even under the highly saturated condition, i.e., worst operating condition.

For the unsaturated condition, GNA could be a good choice because it reduces calculation burden as compared with LMA. Besides, a hybrid algorithm that combines GNA with LMA can be applied to reduce the burden of DSP, e.g., GNA for the low-torque references, whereas LMA for the high-torque references. The choice of numerical algorithms could be optimized with consideration for not only magnetic characteristics but also operating conditions of IPMSM.

Furthermore, the number of iteration $N$ and calculation rate $f_{\text{calc}}$ should be reduced to retain the calculation burden while keeping both the stability and accuracy of the proposed algorithm. The current references should be calculated faster than a current control bandwidth $\omega_{cc} = 2\pi \cdot f_{cc}$. However, $f_{cc}$ should be lower than the sampling frequency, $f_{\text{samp}}$ (about one-tenth), which means that the calculation of the optimal current references at each $f_{\text{samp}}$ is too excessive. Thus, $N$ is set to 1 for mitigating a steep rise of the calculation burden. Likewise, $f_{\text{calc}}$ should be set to a proper value between $f_{\text{samp}}$ and $f_{cc}$, i.e., $f_{cc} < f_{\text{calc}} < f_{\text{samp}}$. It gives enough convergence speed to track the references. In summary, the online MTPA control can be calculated by solving $2 \times 2$ matrix equation, as shown in (18), at each $f_{\text{calc}}$, where $I_q^*_{\text{ref}}$ means the current references calculated by LMA.

The proposed algorithms for MTPA region could be extended to a flux-weakening region by substituting MTPA equation with a voltage limit equation. The intersection of torque curve and voltage limit is a copper loss minimizing point under the flux-weakening region [19]. Likewise, the nonlinear simultaneous equation composed of torque and voltage limit equations can be easily solved by the numerical algorithms, e.g., gradient descent, GNA, LMA, as mentioned previously. However, because this paper is focused on enhancing the robustness of the online MTPA control, the extension of the proposed algorithm to operation at flux weakening is not covered.

IV. TORQUE REFERENCE LIMITER

A. Derivation of Maximum Torque Limit

The maximum available torque is constrained due to current and voltage limits. A large current magnitude would result in the excessive loss during the MTPA operation, which exceeds the thermal constraints [12]. Therefore, the current references should be limited under the current limit $I_{\text{max}}$ in case of the online MTPA control. $I_{\text{max}}$ can be determined by the motor and inverter current rating or active thermal management strategy as mentioned previously. As a result, the torque reference should be limited under the maximum torque, $T_{\text{cc, max}}$, in real time, which also can be defined as the constrained optimization problem. In MTPA region, the maximum torque is constrained by the current limit, which can be formulated as Eq. (18) shown at the bottom of the page.

$$\max T_e = \frac{3P}{2}(\lambda_d i_q - \lambda_q i_d), \text{ subject to } I_{\text{max}}^2 = i_d^2 + i_q^2. \tag{19}$$

Likewise, it can be solved by Lagrange multiplier method, where the Lagrange function is defined as

$$\mathcal{L}(i_d, i_q, \nu) = \frac{3P}{2}(\lambda_d i_q - \lambda_q i_d) + \nu(i_d^2 + i_q^2 - I_{\text{max}}^2) \tag{20}$$

where $\nu$ is the Lagrange multiplier.

$$i_{dq}^* = i_{dq} - \{J(i_{dq})^T J(i_{dq}) + \mu \cdot \text{diag}(J(i_{dq})^T J(i_{dq}))\}^{-1} J(i_{dq})^T F(i_{dq})$$

where $J(i_{dq}) = \begin{bmatrix} -\lambda_q - L_{qd}i_d + L_{dd}i_q & \lambda_d - L_{qq}i_d + L_{dq}i_q \\ \lambda_d + (L_{dd} - 2L_{qd})i_d + (L_{dq} + 2L_{qq})i_q & \lambda_q + (2L_{dq} + L_{qq})i_d + (L_{qq} - 2L_{dd})i_q \end{bmatrix}$

$$F(i_{dq}) = \begin{bmatrix} (\lambda_d i_q - \lambda_q i_d) - \frac{2T_e}{\nu} \\ (\lambda_d - L_{qq}i_d + L_{dq}i_q)i_d + (\lambda_q + L_{qd}i_d - L_{dd}i_q)i_q \end{bmatrix} \tag{18}$$
Necessary conditions for optimization are extracted as (21), which can be simplified to a pair of equations as (22) and (23)

\[
\begin{align*}
0 &= \frac{\partial \mathcal{E}}{\partial i_d} = i_d^2 + i_q^2 - I_{\text{max}}^2 \\
0 &= \frac{\partial \mathcal{E}}{\partial i_q} = \frac{3P}{2} \left( -\lambda_d - L_{dq}i_d + L_{dd}i_q \right) + \nu(2i_q) \\
0 &= i_d^2 + i_q^2 - I_{\text{max}}^2 = p(i_d, i_q) \quad (22) \\
0 &= (\lambda_d - L_{dq}i_d + L_{dd}i_q)i_d + (\lambda_q + L_{qd}i_d - L_{dd}i_q)i_q \\
&= q(i_d, i_q). \quad (23)
\end{align*}
\]

In conclusion, the maximum current circle is represented as \( p(i_d, i_q) \) and MTPA curve is \( q(i_d, i_q) \), where \( q(i_d, i_q) \) is identical with \( g(i_d, i_q) \) in (10). The intersection of the maximum current circle and MTPA curve can be derived by the numerical algorithms similar to the case of online MTPA control.

### B. Implementation of Calculation Algorithm

The maximum torque and its current are also derived by the numerical algorithms where the objective function is defined as

\[
R(i_d, i_q) = p(i_d, i_q)^2 + q(i_d, i_q)^2. \quad (24)
\]

The numerical algorithm should be determined by considering the tradeoff between the ill-convergence risks and convergence speed. Among several algorithms, LMA algorithm is also applied for the torque reference limiter because it converges to solutions in minimum steps without oscillations. On the basis of LMA, the maximum current references, \( i_{dq,\text{max}} \), are calculated by \( 2 \times 2 \) matrix as seen from (25). As a result, the maximum torque \( T_{e,\text{max}} \) is derived as follows: Eq. (25) shown at the bottom of the page.

\[
T_{e,\text{max}} = \frac{3P}{2} (\lambda_d i_{d,\text{max}}^* - \lambda_q i_{q,\text{max}}^*). \quad (26)
\]

Likewise, the minimum torque \( T_{e,\text{min}} \) can be derived from the same process because the Lagrangian function is identical

\[
\begin{align*}
\mathbf{i}_{dq,\text{max}}^* &= i_{dq} - \left\{ \mathbf{J}(i_{dq})^T \mathbf{J}(i_{dq}) + \mu \cdot \text{diag}(\mathbf{J}(i_{dq})^T \mathbf{J}(i_{dq})) \right\}^{-1} \mathbf{J}(i_{dq})^T \mathbf{F}(i_{dq}) \\
\text{where } \mathbf{J}(i_{dq}) &= \begin{bmatrix} 2i_d^2 & 2i_d^2 \\
\lambda_d + (L_{dd} - 2L_{dq})i_d + (L_{dq} + 2L_{qd})i_q & \lambda_q + (2L_{dq} + L_{qd})i_d + (L_{qq} - 2L_{dd})i_q \end{bmatrix} \\
\mathbf{F}(i_{dq}) &= \begin{bmatrix} \lambda_d - L_{dq}i_d + L_{dd}i_q & \lambda_q + L_{qd}i_d - L_{dd}i_q \end{bmatrix}
\end{align*}
\]
with (20). Thus, \( T_{e, \text{min}} \) can be calculated by the same equation as (26).

Fig. 7 shows the overall flowchart of the proposed MTPA algorithm. First, the torque reference limiter is implemented where MTPA equation and current limit are solved by LMA. After that, it is checked whether the torque command \( T_{e, \text{cmd}} \) is feasible or not and produces available \( T^* \) under the given current limit \( I_{\text{max}} \). If the magnitude of torque command \( |T_{e, \text{cmd}}| \) exceeds the maximum torque \( |T_{e, \text{max}}| \), \( T^* \) is substituted with \( T_{e, \text{max}} \). Subsequently, the current references \( i^*_{dq} \) are calculated by the online MTPA control as (18).

Each numerical algorithm, as (18) and (25), can be changed depending on the convergence of the algorithms, which relies on the characteristics of IPMSM. As mentioned previously, GNA could be a good option if \( J(i_{dq}) \) is well conditioned.

V. SIMULATION AND EXPERIMENTAL RESULTS

A. Simulation Results

To simulate actual behaviors of IPMSM including the cross-coupling and saturation effects, a flux-linkage-based machine model is used to simulate in MATLAB/Simulink where motor parameters are extracted via FEA [33]. Fig. 8 shows the overall block diagram of the proposed online MTPA algorithm. For the given torque command \( T_{e, \text{cmd}} \), the torque reference and current references \( T^* \) and \( i^*_{dq} \) are generated from the online calculation block where the calculation algorithms are implemented. \( \lambda_{dq} \) are obtained by a linear interpolation from two-dimensional LUTs, as shown in Fig. 1, and \( L_{dq} \) are extracted by the differences of \( \lambda_{dq} \). The switching and sampling frequency are set to 10 kHz. Calculation algorithms are executed per 0.5 ms, i.e., \( T_{\text{calc}} = 1/f_{\text{calc}} = 0.5 \text{ ms} \), where \( \mu \) and \( N \) are set to 1.

Fig. 9 shows current reference waveforms \( i^*_{dq} \) according to MTPA calculation algorithm during the transition of \( T_{e, \text{cmd}} \) from 0 to 1 p.u. and vice versa below the rated speed. The conventional method based on GNA [19] incurs oscillating calculated current references \( i^*_{dq, \text{Calc}} = (i^*_{d, \text{Calc}}, i^*_{q, \text{Calc}}) \) at full torque, whereas there is no oscillation under low-torque condition, as shown in Fig. 9(a). It shows that not only MTPA control but also torque accuracy cannot be achieved due to the excessive torque fluctuation under the high-torque references. Applying the proposed method based on LMA, \( i_{dq, \text{Calc}}^* \) converges to the optimal current references stored in LUT, \( i_{dq, \text{Table}}^* = (i_{d, \text{Table}}^*, i_{q, \text{Table}}^*) \), even under high-torque condition, as shown in Fig. 9(b). \( i_{dq, \text{Calc}}^* \) traces MTPA curve exactly even under transient conditions.

Fig. 10 shows the simulation results under the transition of \( T_{e, \text{cmd}} \) from 0 to -0.9 p.u. and vice versa, i.e., the under generating mode. In Fig. 10(a), where GNA is applied, the online MTPA control cannot be guaranteed when \( |T^*| \) exceeds about 80% of a peak torque. However, torque accuracy as well as MTPA control can be achieved by applying LMA, as shown in

![Fig. 10. Simulation 2: \( \lambda_{dq} \) waveforms under generating mode. (a) Conventional method (GNA) [19]. (b) Proposed method (LMA).](image-url)
Fig. 10(b). It shows sufficient dynamic performance enough to apply for the industrial and automotive applications.

Fig. 11 shows the performance of torque reference limiter under the variation of $I_{\text{max}}$ from 1 to 0.6 p.u. and vice versa, where $T_{e,\text{cmd}}$ is set to 0.9 p.u. The setting of $I_{\text{max}}$ can be updated in real time according to the thermal condition of inverter and motor. Both torque accuracy and MTPA control are achieved in case I where $T^*_{e}$ is matched with $T_{e,\text{cmd}}$ and $|i^*_{dq,\text{Calc}}|$ is below $I_{\text{max}}$. However, $T^*_{e}$ is limited by the torque reference limiter when $I_{\text{max}}$ is set to 0.6 p.u., as shown in case II, where the magnitude of $i^*_{dq,\text{Calc}}$ is set equal to $I_{\text{max}}$, i.e., $i^*_{dq,\text{Calc}} = I_{\text{max}}$, and $T^*_{e}$ is around 0.6 p.u. In addition, it shows high dynamic performance under the rapid change of $I_{\text{max}}$ and $T^*_{e}$.

### B. Experimental Results

The proposed and conventional methods were also tested in a practical system. All control algorithms are implemented digitally in the DSP, TMS320F28377D, where the control parameters such as $\mu$, $N$, and $T_{\text{calc}}$ are identical with those in the simulation. $\lambda_{dq}$ was extracted in advance based on the stator voltage equations under the steady-state operations. The extracted $\lambda_{dq}$ has been stored as two-dimensional LUTs, i.e., $\lambda_d(i_d, i_q)$ and $\lambda_q(i_d, i_q)$ [34]. The estimated torque $\hat{T}^*_{e}$ can be derived based on $\lambda_{dq}$ LUTs and $i_{dq}$ as (2). $L_{dq}$ can be simply extracted by the differences of $\lambda_{dq}$ LUTs during operations. In addition, MTPA operating points can be extracted from the pretests, which was stored in the LUT, $i^*_{dq,\text{Table}}$, for comparison [6].

Fig. 12 shows $i^*_{dq}$ waveforms under the same condition with those in Fig. 9, which shows the performance of MTPA calculation algorithms under $T^*_{e}$ transients. Likewise, the conventional method based on GNA causes incorrect $i^*_{dq,\text{Calc}}$ and $\hat{T}^*_{e}$ vibration when $T^*_{e}$ is set to a value near 1 p.u., as shown in Fig. 12(a). On the other hand, the proposed method based on LMA generates stable $i^*_{dq,\text{Calc}}$ and $\hat{T}^*_{e}$ under any operating condition, as shown in Fig. 12(b), where $i^*_{dq,\text{Calc}}$ matches precisely to $i^*_{dq,\text{Table}}$ and $\hat{T}^*_{e}$ matches to $T^*_{e}$.

Fig. 13 shows the experimental results under the same operating conditions with those in Fig. 10. Similar to the case of Fig. 10(a), the conventional method based on GNA produces oscillatory $i^*_{dq,\text{Calc}}$ when $T^*_{e}$ is closed to –0.9 p.u., which causes unwanted $\hat{T}^*_{e}$ vibration and acoustic noise, as shown in Fig. 13(a). Therefore, it is not appropriate for IPMSMs.
under harsh conditions, which require robust operation even under the highly saturated operating conditions. On the other hand, the proposed method based on LMA produces accurate $i_{dq,\text{Calc}}$ equivalent to $i_{dq,\text{Table}}$ and $T_e$ equivalent to $T_e^*$, as shown in Fig. 13(b). It verifies that the proposed method shows better performance under the high-torque references, where IPMSM is highly saturated.

Fig. 14 shows the experimental waveforms under the same condition with those in Fig. 11. Likewise, accurate $T_e^*$ and $i_{dq,\text{Calc}}$ are generated when $i_{dq,\text{Calc}}$ is below $I_{\text{max}}$, as shown in case I. However, $i_{dq,\text{Calc}}$ in case I would exceed $I_{\text{max}}$ after $0.3$ s when $I_{\text{max}}$ is set to $0.6$ p.u. In this case, $T_e^*$ is revised by the torque reference limiter, as seen from case II, where the maximum torque is generated under the current constraint. $i_{dq,\text{Calc}}$ matches with $i_{dq,\text{Table}}$ in both cases where $T_e^*$ is limited below $T_{e,\text{max}}$. It shows that the proposed online MTPA control worked well under the condition of $I_{\text{max}}$ variations.

VI. CONCLUSION

In this paper, the online MTPA control has been proposed where the saturation and cross-coupling effects are concerned. First, MTPA control was defined as the constrained optimization problem to fulfill both torque accuracy and minimum-copper-loss operation. It has been derived to two nonlinear simultaneous equations, i.e., torque and MTPA equations, which can be solved by the numerical algorithms. Among them, LMA has been adopted due to the robustness under the highly saturated operating condition of IPMSM. It alleviates the risks of ill-convergence due to the effects of damping factor. Besides, the torque reference limiter has been proposed to satisfy the thermal constraints in MTPA operation. The proposed algorithms have been optimized to minimize the calculation burden while keeping the accuracy of MTPA algorithm.

The effectiveness of the proposed methods has been verified by the simulation and experimental results. Thanks to the proposed algorithm, it has been shown that accurate MTPA control is feasible even under the deep saturation region of IPMSM with LUTs. Consequently, the proposed algorithms can offer the copper loss minimizing point for the torque-control applications with a little computational burden.

The proposed algorithms could be extended to the flux-weakening region by introducing the voltage limit and MTPV equation. Furthermore, the proposed algorithms can be applied together with the online parameter estimation algorithms to improve the torque accuracy and efficiency under parameter variations, especially against flux linkage variations caused by temperature effects and manufacturing tolerance.

REFERENCES


