Optimal Currents Based on Adalines to control a Permanent Magnet Synchronous Machine

Damien Flieller*, Julien Gressier*, Guy Sturtzer*, Djaffar Ould Abdeslam**, Patrice Wira**.

*GREEN UMR CNRS 7037 antenne de Strasbourg – France 24 boulevard de la victoire, 67084 Strasbourg cedex {damien.flieller, julien.gressier} @insa-strasbourg.fr
**MIPS Laboratory - University of Mulhouse 4 rue des Frères Lumière, 68093 Mulhouse – France {djaffar.ould-abdeslam, patrice.wira}@uha.fr

Abstract- This paper describes a way to find optimum currents to control a Permanent Magnet Synchronous Machine (PMSM) using neural methods. Here the aim is to minimize torque and speed undulations but the proposed method enter in the global problematic of current compensation. Many excellent authors have already written articles on this matter, this one adds a new contribution with the advantage of being flexible toward the process to control. The neural algorithm has been implemented and the effectiveness of the adaptive torque ripple compensation demonstrated by experiments.

NOMENCLATURE

- $i_u$, $i_v$, $i_w$: Currents on the u, v and w phases.
- $e_u$, $e_v$, $e_w$: Back emf on the u, v and w phases.
- $T$: Electromagnetic torque.
- $\omega$: Electrical angular frequency of the motor.
- $\Omega$: Rotor velocity.
- $\lambda_1$, $\lambda_2$: Lagrange parameters.
- $e_u$, $e_v$, $e_w$: Coordinates of $e_u$, $e_v$, $e_w$ in the $\alpha$, $\beta$ frame.
- $i_u$, $i_v$: Coordinates of $i_u$, $i_v$, $i_w$ in the $\alpha$, $\beta$ frame.
- $T_{32}$: Concordia transform matrix:
  \[
  T_{32} = \begin{bmatrix}
  1 & 0 \\
  \frac{1}{2} & \frac{\sqrt{3}}{2} \\
  \frac{1}{2} & -\frac{\sqrt{3}}{2}
  \end{bmatrix}
  \]
- $P(\omega t)$: Park transform matrix:
  \[
  P(\omega t) = \begin{bmatrix}
  \cos(\omega t) & -\sin(\omega t) \\
  \sin(\omega t) & \cos(\omega t)
  \end{bmatrix}
  \]
- $\Theta$: Rotor’s angular position.
- $F_{ref}$: Periodic function from the neural process.
- $p$: Pole pairs.
- $P_n$: Synaptic weight at the $n^{th}$ iteration.
- $\eta$: Learning coefficient.
- $\xi_n$: Process error at the $n^{th}$ iteration.
- $X_n$: Input of the neuron at the $n^{th}$ iteration.
- $R$: Stator winding resistance.
- $\Psi_0$: PM Flux magnitude.
- $T_o$: Electrical period.

I. INTRODUCTION

Permanent Magnet Synchronous Machine (PMSM) drives have become more and more popular for many servo control and robotic applications. PMSMs are still preferred for their efficiency due to the absence of magnetizing current and to their very small copper losses. However the main inconvenient is the presence of a torque ripple. This phenomenon could be amplified by a non sinusoidal back emf, non constant load torque and possible defect on the inverter. That is why high performance in PMSM control systems needs to know exactly what kind of references currents are to be generated and how to find them. High frequency electromagnetic torque components can be created by fast current control for the compensation of torque ripple [1]. But a simpler way is to build an Iterative Learning Control (ILC) for torque compensation using the fact that torque ripples are periodic [2] and [3]. This document offers another quite simple method of calculation for the three reference currents in order to control and obtain an almost constant speed or torque based on a similar point. The main interest of this method is to be able to adapt the form of these currents to a majority of possible defects that can occur in the controlled system. We will first present various ways of finding the optimum current formulation, and how to demonstrate it.

We will consider two different cases; the first one is the calculation of these currents with homopolar component and the second one without. Then, we will explore the neural method based on an Adaline. We will explain the algorithm used to find the currents. A paragraph is also focused on the interactions with the controlled system and the influence of the different parameters. Finally, we will examine an example of the neural method for current compensation on a PMSM with speed ripple optimization. This method gives some relatively good results with a quite simple control algorithm.

II. EXPRESSION OF OPTIMUM CURRENTS

The optimum currents that have to be created to control the BLDC motor are the following. We can consider two cases; the first one is when currents are calculated using the homopolar current:

\[
i_u = \frac{e_u T \Omega}{e_u^2 + e_v^2 + e_w^2}
\]
\[ i_v = \frac{e_v T\Omega}{e_u^2 + e_v^2 + e_w^2} \]  
\[ i_w = \frac{e_w T\Omega}{e_u^2 + e_v^2 + e_w^2} \]  

The second one is the calculation without the homopolar component:
\[ i_u = \frac{(e_u - e_v) + (e_u - e_w)}{(e_u - e_v)^2 + (e_v - e_w)^2 + (e_u - e_w)^2} T\Omega \]  
\[ i_v = \frac{(e_v - e_u) + (e_v - e_w)}{(e_u - e_v)^2 + (e_v - e_w)^2 + (e_u - e_w)^2} T\Omega \]  
\[ i_w = \frac{(e_w - e_u) + (e_w - e_u)}{(e_u - e_v)^2 + (e_v - e_w)^2 + (e_u - e_w)^2} T\Omega \]

III. DEMONSTRATION OF THE REFERENCE CURRENTS

A. Lagrange’s Method

This paragraph presents how the currents references expression (2) without homopolar could be found. We consider the copper losses equation (3), the power equation (2) and the fact that there is no homopolar component (5):
\[ R i_u^2 + R i_v^2 + R i_w^2 = P_j \]  
\[ e_u i_u + e_v i_v + e_w i_w = T\Omega \]  
\[ i_u + i_v + i_w = 0 \]

A way to find the optimum currents at the meaning of the minimum copper losses is to use the Lagrange’s method. Here the extended criterion to be minimized is:
\[ i_u^2 + i_v^2 + i_w^2 + \lambda_1 (i_u + i_v + i_w) + \lambda_2 (u i_u + v i_v + w i_w - T\Omega) \]

The derivation by \( i_u, i_v, i_w \) respectively gives:
\[ 2 i_u = -\lambda_1 + \lambda_2 e_u \]  
\[ 2 i_v = -\lambda_1 + \lambda_2 e_v \]  
\[ 2 i_w = -\lambda_1 + \lambda_2 e_w \]

Replacing \( i_u, i_v, i_w \) in (5) and (4) gives:
\[ \lambda_1 = \frac{2 (e_u + e_v + e_w) T\Omega}{3 (e_u^2 + e_v^2 + e_w^2) - (e_u + e_v + e_w)^2} \]  
\[ \lambda_2 = \frac{-6 T\Omega}{3 (e_u^2 + e_v^2 + e_w^2) - (e_u + e_v + e_w)^2} \]

These expressions are introduced in (7) and lead to:
\[ i_u = \frac{(e_u - e_v) + (e_u - e_w)}{2 e_u^2 + 2 e_v^2 + 2 e_w^2 - 2 e_u e_v - 2 e_u e_w - 2 e_v e_w} T\Omega \]  

We find the expression:
\[ i_u = \frac{(e_u - e_v) + (e_u - e_w)}{(e_u - e_v)^2 + (e_v - e_w)^2 + (e_u - e_w)^2} T\Omega \]  

B. Iso-curve method

In this part, we will take a look at how to find the reference current expression with homopolar component (1). A simple way is to find it with the iso torque method. The iso-torque curves are defined by the set of points \( M \) giving a constant electromagnetic torque for an angle \( \theta \) (see Fig. 1). In order to minimize the copper losses it is necessary to choose the point \( M = \tilde{M} \) on the curve which gives an \( \tilde{OM} \) vector as small as possible [4]. Equation (4) has the following expression in the \( ab\)-frame:
\[ e_a i_a + e_b i_b = T\Omega \]

Graphic presentation to find the \( \tilde{OM} \) optimum:
\[ \tilde{OM} = \frac{T\Omega}{e_a} \sin(\alpha) \]  
\[ \sin(\alpha) = \frac{e_a}{\sqrt{e_a^2 + e_b^2}} \]

We can deduce from (15) and (16), therefore:
\[ i_a = \frac{T\Omega}{e_a} \sin^2(\alpha) = \frac{T\Omega}{e_a^2 + e_b^2} \]

The expression of \( i_b \) is:
\[ i_b = \frac{T\Omega e_b}{e_a^2 + e_b^2} \]

The relations between \( i_u, i_v, i_w, i_a, i_b, e_u, e_v, e_w, e_a \) and \( e_b \) are:
\[ \begin{bmatrix} i_u \\ i_v \\ i_w \end{bmatrix} = T_{32} \begin{bmatrix} i_a \\ i_b \end{bmatrix} \]  
\[ \begin{bmatrix} e_u \\ e_v \\ e_w \end{bmatrix} = T_{32} \begin{bmatrix} e_a \\ e_b \end{bmatrix} \]
\[
\begin{bmatrix}
i_u \\
i_v \\
i_w
\end{bmatrix} = \frac{T \Omega}{e_a^2 + e_b^2} T_{32} \begin{bmatrix} e_a \\ e_b \end{bmatrix}
\]  
(21)

and
\[
e_a^2 + e_b^2 = [e_a \ e_b] \begin{bmatrix} e_a \\ e_b \end{bmatrix}
\]  
(22)

\[
e_a^2 + e_b^2 = [e_u \ e_v \ e_w] T_{32}^T T_{32} \begin{bmatrix} e_u \\ e_v \\ e_w \end{bmatrix}
\]  
(23)

\[
e_a^2 + e_b^2 = e_u^2 + e_v^2 + e_w^2
\]  
(24)

Consequently the relation between the currents and back emf using homopolar component is:
\[
\begin{bmatrix}
i_u \\
i_v \\
i_w
\end{bmatrix} = \frac{T \Omega}{e_u^2 + e_v^2 + e_w^2} \begin{bmatrix} e_u \\ e_v \\ e_w \end{bmatrix}
\]  
(25)

IV. OPTIMUM CURRENTS IN NORMAL OPERATING CONDITIONS

A. Sinusoidal behavior without defect
We consider the following case; the PMSM runs with three sinusoidal back emf:
\[
\begin{bmatrix} e_u \\ e_v \\ e_w \end{bmatrix} = \begin{bmatrix}
\Psi_0 \omega \cos (\omega t) \\
\Psi_0 \omega \cos (\omega t - \frac{2\pi}{3}) \\
\Psi_0 \omega \cos (\omega t + \frac{2\pi}{3})
\end{bmatrix}
\]
(26)

\[
e_u^2 + e_v^2 + e_w^2 = \frac{3}{2} (\Psi_0 \omega)^2
\]  
(27)

According to (1), the calculation of optimum currents gives:
\[
\begin{bmatrix}
i_u \\
i_v \\
i_w
\end{bmatrix} = \frac{T}{\frac{3}{2} \Psi_0} \begin{bmatrix}
\cos (\omega t) \\
\cos (\omega t - \frac{2\pi}{3}) \\
\cos (\omega t + \frac{2\pi}{3})
\end{bmatrix}
\]  
(28)

In the sinusoidal behavior the reference currents which have been found have the classical form, in phases with the back emf.

B. Non-Sinusoidal behavior without defect
In this case, we consider the different harmonics of the back emf (Table 1):
\[
\begin{bmatrix} e_u \\ e_v \\ e_w \end{bmatrix} = \sum_{k=1}^{k=9} A_k \begin{bmatrix}
\cos (k\omega t) \\
\cos (k\omega t - \frac{2\pi}{3}) \\
\cos (k\omega t + \frac{2\pi}{3})
\end{bmatrix} + B_k \begin{bmatrix}
\sin (k\omega t) \\
\sin (k\omega t - \frac{2\pi}{3}) \\
\sin (k\omega t + \frac{2\pi}{3})
\end{bmatrix}
\]  
(29)

With a back emf waveform which has half-wave symmetry, only odd harmonics are present [5]. Fig. 2 shows for example the non sinusoidal waveform of e_u. The FFT data have been normalized by dividing by the magnitude of fundamental component (A_1^2 + B_1^2 = E_1^2 = 1).

<table>
<thead>
<tr>
<th>Harmonic</th>
<th>A_k</th>
<th>B_k</th>
<th>E_k</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>-0.0678</td>
<td>-0.9977</td>
<td>1</td>
</tr>
<tr>
<td>3</td>
<td>-0.0405</td>
<td>-0.2107</td>
<td>0.2146</td>
</tr>
<tr>
<td>5</td>
<td>-0.0063</td>
<td>-0.0216</td>
<td>0.0225</td>
</tr>
<tr>
<td>7</td>
<td>0.0300</td>
<td>0.0315</td>
<td>0.0435</td>
</tr>
<tr>
<td>9</td>
<td>0.0365</td>
<td>0.0288</td>
<td>0.0465</td>
</tr>
</tbody>
</table>

Table 1. Harmonics amplitudes
The non-sinusoidal operating conditions are illustrated with and without homopolar current with respectively Fig. 3 and Fig. 4. A Joule-losses criteria can be used to evaluate the performances in the case of these operating condition:

\[
\frac{(E_1)^2}{R(TΩ)^2}P_1 = 0.2227 \tag{30}
\]
\[
\frac{(E_2)^2}{R(TΩ)^2}P_2 = 0.2314 \tag{31}
\]

According to (30) and (31), the best way to minimize copper losses is to choose the expression with homopolar current, if it is possible with the inverter.

C. Optimum currents in faulty conditions
We consider that the PMSM just runs on the v and w lines; this means that there is no current on the u phase. This is illustrated by Fig. 5. The resulting optimum currents are represented by Fig. 6 and are given by:

\[
\begin{bmatrix}
i_u \\
i_v \\
i_w
\end{bmatrix} = \frac{TΩ}{e_v^2 + e_w^2} e_v \tag{32}
\]

As it is not always possible to get access to the all parameters needed for the calculation of the currents (back emf, rotor velocity, Torque, etc.), the next paragraph explain how to find them just with Ω given by the incremental coder in the case of a speed feedback. This principle is also valid with the measure of Ω and T in the case of a torque feedback.

V. NEURAL UNIT FOR SPEED RIPPLE COMPENSATION

A. Principle
The global system is subjected to a required torque and it measures constantly the torque generated by the PMSM (Fig. 7). The difference between them gives rise to an error ξ varying in the course of time. The neural process measures this error and adapts his synaptic weights accordingly.

These weights Ps are in fact the amplitudes of the Fourier transform of the function which we try to generate to control the motor. Thanks to these coefficients the neural process rebuilds the researched function, which can be called "Fref" (Fig. 8). This periodic function is then multiplied by each emf eu, ev, and ew respectively to create the three reference currents for the motor that are iu, iv, and iw.

The reference currents are chosen collinear to the back emf to minimize copper losses according with (1). These currents are then measured to rebuild the torque from the angular rotor position:

\[
i_u = F_{ref} e_u \tag{33}
\]
\[
i_v = F_{ref} e_v \tag{34}
\]
\[
i_w = F_{ref} e_w \tag{35}
\]

By replacing the expressions (33), (34) and (35) in the power equation (4) we acquire:

\[
TΩ = F_{ref} (e_u^2 + e_v^2 + e_w^2) \tag{36}
\]

If the load torque is:

\[
T = -Tc - \bar{T}(θ) \tag{37}
\]

with Tc is constant torque and \(\bar{T}(θ)\) a torque variable in function of θ, we have to create a torque Tc + \(\bar{T}(θ)\) to obtain a constant speed. The term \(\bar{T}(θ)\) is quite never known and it can change while the motor turns, so F_{ref} must be adapted in function of it.

According to (36):
\[
T_c + \bar{T}(\theta) = F_{\text{ref}}(e_u^2 + e_v^2 + e_w^2) \tag{38}
\]

So,
\[
\frac{T_c}{(e_u^2 + e_v^2 + e_w^2)} + \frac{\bar{T}(\theta)}{(e_u^2 + e_v^2 + e_w^2)} = F_{\text{ref}} \tag{39}
\]

A block diagram illustrating the weight update algorithm in the neural process is depicted by Fig. 9. \( \xi_n \) is the error between the measured torque and the required one; “rank” corresponds to the number of harmonics of the \( F_{\text{ref}} \) function.

The neural controller is an Adaline neural network [6] presented by Fig. 10. The number of weights depends on the rank chosen by the user. The neural unit is built to adapt its weights in function of the error of the feedback process and a learning coefficient. The optimum learning coefficient depends on this number. It will affect the system dynamic and the precision. The mathematical formulation to update the weights of the neural controller is the Least Mean Square Algorithm (LMS). If the error in the \( n \)th iteration is not void, the weights of the present iteration are given by:
\[
P_{n+1} = P_n + \eta \xi_n X_n \tag{40}
\]
When one of the phases is disconnected the system becomes unsettled but finds a proper resolution to pilot the motor in the operating conditions. Indeed the neural process search automatically the \( F_{\text{ref}} \) function which is going to cancel the error measured between the required torque and the measured one. This system of currents allows after a learning phase to make converge the torque measured towards the required one.

**B. Experiment**

The aim of this experiment is to show the characteristics of this algorithm on a PMSM whose load is not constant. The experiment parameters are recapitulated in Table 2. The neural controller needs a learning time to adapt its weights. In a first time we see how the motor with no compensation runs. Here the number of pole pairs is three and rank is five. The reference currents are shown by Fig. 11.

![Output power 720 W](image-url)

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Output power</td>
<td>720 W</td>
</tr>
<tr>
<td>Voltage constant</td>
<td>98 Vrms/Krpm</td>
</tr>
<tr>
<td>Rating current</td>
<td>1.42 A</td>
</tr>
<tr>
<td>Torque constant</td>
<td>1.6 Nm/Arms</td>
</tr>
<tr>
<td>Rotate speed</td>
<td>3000 Rpm</td>
</tr>
<tr>
<td>Pole pairs</td>
<td>3</td>
</tr>
<tr>
<td>Resistance</td>
<td>12.25 Ω</td>
</tr>
<tr>
<td>Inductance</td>
<td>0.033 H</td>
</tr>
</tbody>
</table>

Table 2. Rating and constants of the experimental motor

![Fig. 11. Compensation currents before the learning process](image-url)

The \( F_{\text{ref}} \) function is constant so the three reference currents are sinusoidal. The speed is given by Fig. 12.

![Fig. 12. Compensation currents after the learning process](image-url)

The speed has an undulation, due to the load, between 102.5% and 97.5%, which means an error of 5% at the beginning of the process. In a second time the compensation starts to run. The reference currents are given by Fig. 13.

![Fig. 13. Compensation currents after the learning process](image-url)

As the error on the measured speed is not void, each weight of the neural controller start to minimize this one. So, as these weights are the Fourier coefficients of the \( F_{\text{ref}} \) function and as the reference currents are created by \( F_{\text{ref}} \), their form changes and the undulations are reduced. Twenty seconds after the beginning of the learning process the error on the speed has been reduced from 50% towards the initial ones. Then, as the time passes, the neural controller gives a \( F_{\text{ref}} \) function which minimizes the speed ripple for this particular load defect. If the load changes, the neural controller will recalculate the
weights to find another optimal solution for the speed. After application of this method, the speed only takes its values between 0.996 and 1.005, which means less than 1% of ripple. We can see on Fig. 14 this residual error. This error can be reduced by using more harmonics that means more weights in the neural process.

![Fig. 12. Speed ripple before the learning process. (Experimental result)](image1)

![Fig. 13. Compensation currents after the learning process. (Experimental result)](image2)

![Fig. 14. Speed ripple after the learning process. (Experimental result)](image3)

VI. CONCLUSION

This paper introduces optimal currents for a PMSM in order to keep the torque constant or the speed undulations null with minimal Joules losses whatever the load. We show that these optimal currents can be analytically expressed in all cases, i.e., sinusoidal or not. Their expression can be simplified in the presence of homopolar components. It appears from these formulas that the optimal currents are proportional to the emf with a coefficient $F_{ref}$ depending of $\theta$. An ideal machine leads to sinusoidal emf and a constant $F_{ref}$ coefficient. The proposed method is also valid for a machine with faults. For example, the torque remains constant with a disconnected phase. In the non sinusoidal case, $F_{ref}$ depends from the emf, from the desired torque and load torque, and can theoretically be expressed by a Fourier series expansion. The load torque is generally unknown, we therefore propose to use a neural network to learn $F_{ref}$. The approach has been successfully implemented with simulation and experiments. The results clearly show that the speed undulations are reduced by 80% with an unknown load and uncertain parameters.

REFERENCES


