ABSTRACT

In this paper, we introduce the neural network to reduce the ripple of error occurred when we control a linear pulse motor. In general, conventional position controllers for linear pulse motor disregard the modeling error and load variations, which cause inaccuracy in position control. We propose a neural network-based learning controller that modifies the current commands applied into linear motor in order to reduce error due to these factors. The experiment results show that the proposed controller works efficiently for accurate position control of linear pulse motor.

Keywords: linear pulse motor, neural network, unmodeled dynamics, force ripple

1. INTRODUCTION

Recently the importance and application examples of a linear motor are increasing in industry field.[1,2] A linear motor is a direct drive motor that has good performance in terms of accuracy, velocity and acceleration compared with the conventional linear motion system consisting of rotary motor, gear and ball screw. A Linear Pulse Motor(LPM), one of the linear motor, has some merits such as compactness, high reliability and free maintenance comparing with linear motion system with rotary motor, gear and belt. However, some highly nonlinear characteristics caused by inherent leakage flux and damping force delimit the enhancement for precision control with conventional control algorithm.

There are many excitation methods in order to control a linear pulse motor like conventional rotary pulse motor. Recently, in order to reduce force ripple the micro-step excitation method is often used.[3] In the micro-step control, the sinusoidal waves with 90 degrees phase difference are used. However, even though the sine waves are applied into the LPM, it is well known to occur the force ripple[3], because the sinusoidal current commands are extracted from the model without considering many nonlinear factors of LPM. So many studies for reducing the force ripple have been continued.[3,4] These studies found out the exact current commands to reduce force ripple by finding the accurate model via either experiments or theoretical studies for dynamics of a LPM.

However, it is difficult for this controller to be applied into the LPM with different model parameters. At low speed, in special, the conventional motion control system has a big position error by force ripple. if we disregard these factors, the force ripple has a bad effect on the motion control system. To get a better performance, we need to find the current commands considering these factors. In this paper, we introduce the neural network-based current controller of LPM in order to get current commands considering these factors at a time. And the validity of the proposed controller using neural network will be confirmed by applying it experimentally to control LPM.

2. DESIGN OF THE CONTROLLER

After we investigate the basic principles of the LPM and propose a neural network-based current controller.

2.1 PRINCIPLE OF HYBRID TYPE LPM

Fig. 1 depicts the working principle of a hybrid type LPM. Its principle is the same that of rotary type step motor. If the current is applied into phase A, the flux path is made by the mutual action of permanent magnet and current as shown in Fig.1 (a). Then the mover stops on stable equilibrium position that the magnetic resistance is a minimum. After that the current is applied into phase B

Fig. 1 The working principle of a hybrid linear pulse motor.
as shown in Fig. 1 (b), then the mover stops on next equilibrium position that shifted right by 1/4 pitch by same principle. Next, the opposite currents are applied into phase A and phase B continuously as Fig. 1 (c) and (d). The mover is shifted to right side by each 1/4 pitch. If these steps are repeated, the mover moves to right side continuously. This principle represents an one phase excitation method. Recently, however, to reduce force ripple, the micro step excitation method using sinusoidal waves which have 90 degrees phase difference as current command is often used. The phases are continuously excited by sinusoidal current commands.

2.2 NEURAL NETWORK-BASED CURRENT CONTROLLER

Assume a linear motor model

\[ M \frac{d^2x}{dt^2} + B \frac{dx}{dt} + F_e = F_e, \quad (1) \]

where

- \( M \) : mass of mover and load
- \( B \) : viscous frictional coefficient
- \( F_e \) : developed force by motor
- \( F_L \) : force by load
- \( x \) : position of mover

And the electrical force generated by motor is given by

\[ F_e(i_A, i_B, x) = K \left( - \sin \left( \frac{2\pi}{p} x \right) i_A + \cos \left( \frac{2\pi}{p} x \right) i_B \right) \quad (2) \]

where, \( p \) is pitch length. In this case, the current commands, \( i_A \) and \( i_B \), to get constant force become

\[ i_A = - \sin \left( \frac{2\pi}{p} x \right), \quad i_B = \cos \left( \frac{2\pi}{p} x \right) \quad (3) \]

But, current commands above were approximated by many assumptions in modeling. If we use the current commands in equation (3), the force ripple is generated by unmodeled dynamics. Since this has bad effect on precise position control, we propose a neural network-based current control scheme to resolve this problem.

There are many structures of neural network applied in control fields, for example, a feedback error learning type, a direct inverse model learning type, an indirect model learning type, and so on.[4-8] In this paper, we introduce the feedback error learning structure.[1,4,8] The neural network used in this paper is a 3 layer feedforward network using back-propagation for learning.

Fig. 2 shows the structure of a proposed controller.

To evaluate the performance for several structures, two neural networks are introduced. Each neural network is used exclusively and the learning structure of each neural network is feedforward learning controller using feedback error. Each structure of neural networks represents in Fig. 3. The main differences between neural network 1 (NN1) and neural network 2 (NN2) are inputs of them and applying point of additional control input from neural network output. The PID type feedback controller is inserted to enhance the robustness and stability of the system.

Bipolar sigmoid activation function is used. And to get better convergence rate, momentum term is used. To minimize the PID output of feedback controller, neural network is learned. The weights of neural network are modified by standard back-propagation algorithm.

3. EXPERIMENTS

Since the response of current controller is very rapid,
we implement our algorithm with DSP board with TMSC320C31. We investigate the performance of NN1 and NN2 respectively at both high and low speed.

3.1 EXPERIMENTAL ENVIRONMENTS

The current commands that calculated by DSP board according to the proposed algorithm are applied into current driver. Fig. 4 represents schematic diagram of experimental apparatus. The experimental system consists of LPM(2HRM0205-1020L) with PWM current driver, linear scale with 20㎛ resolution, DSP board and etc. The length of pitch and stroke of LPM are 0.8mm and 1m, respectively. Sampling time is 2.5msec. Fig. 5 represents the functional diagram of PWM current driver.

![Schematic diagram of experimental apparatus](image)

Fig. 4. Schematic diagram of experimental apparatus

![Functional diagram of PWM current driver](image)

Fig. 5 The blockdiagram of current driver

3.2 EXPERIMENT RESULTS

The learning and momentum coefficients of NN1 are 0.3 and 0.9 respectively. And coefficients of NN2 are 0.1 and 0.9. Initial weights of NN1 and NN2 are randomly chosen as values from –0.1 to +0.1.

Fig. 6 and Fig. 7 show position and velocity references at low and high speed, respectively. At low speed, an LPM moves 10 pitches.

Fig. 8 shows the position error when the only open loop control is performed. We can see that the ripple of error is big and constant offset error is also big.

![Desired velocity profile at low speed](image)

Fig. 6 Desired velocity profile at low speed

![Desired velocity profile at high speed](image)

Fig. 7 Desired velocity profile at high speed

![The position error when the only open loop control is performed at low speed](image)

Fig. 8 The position error when the only open loop control is performed at low speed

Fig. 9 is the result when the PI and feedforward controllers are operated at same time. The offset error is decreasing according to time because of I action of PI...
controller. But, the ripple magnitude is still big.

From Fig. 10, we can see that the constant offset error is decreased to 0 by using PI+FF+NN1 controller. But the ripple amplitude is still big. Fig. 11 shows the result when PI+FF+NN2 controller are used at the same time after learning. The peak-to-peak ripple is decreased and offset error becomes almost zero. From the results above, we conjecture that the PI+FF+NN2 controller has best performance at low speed.

Fig. 12 represents the change of the mean squared error (MSE) for PI+FF+NN2 controller according to learning iterations. Where, the one iteration means a period that a velocity profile cycle is performed. The whole results at low speed are summarized in Table 1.

<table>
<thead>
<tr>
<th>Controller</th>
<th>Peak to peak error (mm)</th>
<th>Offset error (mm)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Open loop</td>
<td>0.095</td>
<td>0.040</td>
</tr>
<tr>
<td>PI + FF</td>
<td>0.075</td>
<td>0.020</td>
</tr>
<tr>
<td>PI + FF + NN1</td>
<td>0.075</td>
<td>0.010</td>
</tr>
<tr>
<td>PI + FF + NN2</td>
<td>0.050</td>
<td>0.005</td>
</tr>
</tbody>
</table>

Up to now, we investigated the results for low speed. Now, let’s consider the results for high speed profile as shown in Fig. 7.
The results for high speed is similar to that of low speed in point of view that the offset position error is decreased by adding neural networks. The whole results are shown in Table 2. At high speed, we did not use NN1.

From the whole experimental results, neural networks play a role in reducing the offset position error. At low speed, the performance of NN2, the amplitude of current commands is varying directly, is better than that of NN1. And the ripple amplitude is also reduced a little. But the ripple amplitude is still big even though the neural network is added. To reduce peak-to-peak ripple, we have to develop better neural network to operate for periodic error.

Table 2  Comparison of position error at high velocity

<table>
<thead>
<tr>
<th>Controller</th>
<th>Peak to peak error (mm)</th>
<th>Offset error (mm)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Open loop</td>
<td>0.17</td>
<td>0.100</td>
</tr>
<tr>
<td>PI + FF</td>
<td>0.13</td>
<td>0.030</td>
</tr>
<tr>
<td>PI + FF + NN2</td>
<td>0.11</td>
<td>0.015</td>
</tr>
</tbody>
</table>

4. CONCLUSIONS

In this paper, we introduced a neural network of feedback error learning type. The neural network changes a current command to improve position accuracy. As a result of experiments, we could see the fact that the introduced neural network works efficiently for reducing the position ripple error of an LPM at low speed compared to conventional controller. But, since the position error has still large ripple, we wish to expand the capabilities of the neural network to account for reducing ripples. To do this, we must develop better neural network to operate for periodic error.

References