

## Automatic Tuning of Pulse Timings Using Neural Network for Rotor Oscillation Damping of a Stepping Motor

Kenshi Matsuo<sup>\*a)</sup> Member, Zheng Xiang<sup>\*</sup> Non-member  
Takeshi Miura<sup>\*</sup> Member, Katsubumi Tajima<sup>\*</sup> Member

(Manuscript received May 8, 2018)

To suppress the rotor oscillation when positioning a stepping motor, we propose a novel scheme that automatically tunes the pulse timings in back-phasing damping using a neural network, from the input of only a step response. Here, the network is learned from the response and its timings for a few loads. To validate the scheme, the drives that satisfactorily damp the rotor oscillation are demonstrated using the timings estimated with unknown loads attached.

**Keywords:** stepping motor, rotor oscillation damping, back-phasing damping, neural network

### 1. Introduction

In stepping motors, the rotor oscillation occurs in positioning, which causes out of synchronism and increases the settling time. Back-phasing damping<sup>(1)</sup> has been known as a damping method, which has been achieved by properly tuning the timing of pulses switching the excitation phase. However, this method needs to tune the timings whenever the inertia moment of load with the rotor changes. Moreover, it is difficult to grasp the value of moment in actual systems.

In this paper, we propose a scheme that automatically tunes the pulse timings using a neural network. The network outputs the proper timings estimated on the basis of the load when only a step response of one step angle in one-phase excitation drive is given as input. Here, the network is learned using the response and optimal timings that are manually sought for a few loads in advance. To validate the scheme, the drives using the timings estimated with unknown loads attached are conducted, and the suppression performance is evaluated.

### 2. Tuning Design

A two-phase hybrid type stepping motor is used.

**2.1 Back-phasing Damping** This method enables the rotor to rotate for one step angle without oscillation<sup>(1)</sup>. First, the rotor comes to rest at the equilibrium position with phase A excited as shown in Fig. 1. Next, give a clockwise (CW) pulse in the drive circuit, and the excitation phase is switched from phase A to B. The rotor moves to the next equilibrium position while accelerating. Then, for preventing the overshoot, i.e., decelerating, give a counter clockwise (CCW) pulse, and the excitation phase is switched back to phase A. Finally, after sufficient deceleration, give a CW pulse again, and the rotor can stop at the next position without oscillation. The pulse timings are set as time  $t_1$  between first CW and CCW pulses and  $t_2$  between first and second CW pulses.

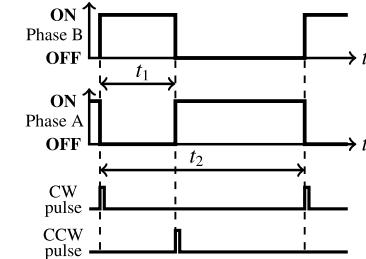


Fig. 1. Pulse timings and excitation phase in back-phasing damping<sup>(1)</sup>

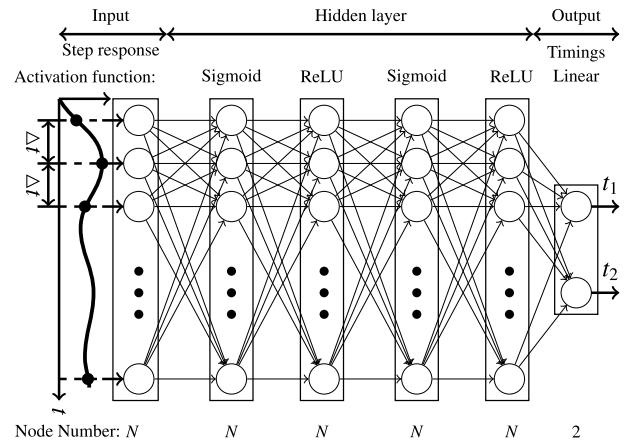


Fig. 2. Feedforward neural network structure

**2.2 Neural Network** Back-phasing damping needs to tune the pulse timings whenever the inertia moment of load with the rotor changes. As shown in Fig. 2, a feedforward neural network (FNN)<sup>(2)</sup> is used for the tuning. The FNN is a six-layer structure that is composed of an input layer with  $N$  nodes, four hidden layers with each  $N$  nodes and an output layer with two nodes, which outputs the timings. The nodes of each layer are fully connected to the nodes of the next layer with weights. Then, for the  $N$  nodes  $\{x_1, \dots, x_N\}$  of the layer in the input and hidden layers, each node of the next layer  $y_j$  ( $j = 1, \dots, P$ ) is calculated as  $y_j = f(\sum_{i=1}^N w_{ij}x_i)$ ,

a) Correspondence to: Kenshi Matsuo. E-mail: matsuo@gipc.akita-u.ac.jp

\* Graduate School of Engineering Science, Akita University  
1-1, Tegata Gakuen-machi, Akita 010-8502, Japan

Table 1. The optimal and estimated pulse timings for loads attached with rotor

Load $J \times 10^{-4}$ [Nm s <sup>2</sup> /rad]	No Load*	Load1	Load2	Load3	Load4	Load5	Load6	Load7*	Load8	Load9	Load10*	
Optimal timings	$t_{1t}$ [ms]	1.4	1.6	1.9	2.3	2.5	2.6	2.8	3.0	3.3	3.7	4.0
	$t_{2t}$ [ms]	2.3	2.8	3.3	4.2	4.5	4.8	5.1	5.5	6.0	6.7	7.2
Estimated timings	$t_1$ [ms]	1.4	1.6	1.8	2.1	2.3	2.5	2.8	3.0	3.1	3.5	4.0
	$t_2$ [ms]	2.3	2.8	3.3	3.8	4.1	4.6	5.1	5.5	5.7	6.4	7.2

\* Used to learn FNN.

where  $P = N$  in the hidden layers and  $P = 2$  in the output layer,  $w_{ij}$  is the weight between  $x_i$  and  $y_j$  and  $f(\cdot)$  means an activation function. Each activation function in the hidden layers as in Fig. 2 is a sigmoid ( $f(z) = 1/(1 - e^{-z})$ ) and ReLU ( $f(z) = \max(0, z)$ )<sup>(2)</sup>, and in the output layer is a linear function ( $f(z) = z$ ), where  $z$  is a real number. When the waveform of a step response of one step angle sampled at time  $\Delta t$  is given in the input layer, the timings  $t_1$  and  $t_2$  are estimated from the output layer through the hidden layers. Because the response includes the effect of the load, the timing based on the load can be estimated.

To train the weights in the FNN, the teacher data are needed. The data are the step response and optimal pulse timings  $t_1$  and  $t_2$  (denoted here as  $t_{1t}$  and  $t_{2t}$ ) sought manually for the inertial moment of loads with the rotor. Then the weights are updated such that the mean squared error  $\{(t_1 - t_{1t})^2 + (t_2 - t_{2t})^2\}/m$  for the  $m$  inputs is reduced using back propagation<sup>(2)</sup>.

**2.3 Tuning Scheme** A tuning scheme is proposed as follows. Firstly, a CW pulse is given with the rotor stopped and the motor runs in one phase excitation drive. Next, for the obtained step response of one step angle, the FNN outputs the pulse timing  $t_1$  and  $t_2$ . Finally, after the tuning, give the CW and CCW pulses of the timings in open-loop control, and the rotor will move to the next equilibrium position and come to rest with oscillation suppressed.

### 3. Experiments

The used stepping motor is Oriental Motor PK244-02B, where the step angle is 1.8 deg. For the training and validation, the optimal timings  $t_{1t}$  and  $t_{2t}$  tuned manually for no load and 10 loads are obtained in advance as in Table 1.

**3.1 Learning Neural Network** The FNN are learned using the step responses of one step angle in one-phase excitation as input and the pulse timings  $t_{1t}$  and  $t_{2t}$  as output for only three loads which are No Load (min moment), Load7 (intermediate) and Load10 (max). Here, the step responses are for 300 ms at sampling time  $\Delta t = 0.1$  ms, i.e.,  $N = 3000$  samples. Actually, when the responses are inputted in the input nodes, each angle  $\theta_i$  ( $i = 1, \dots, N$ ) sampled at  $\Delta t$  is converted into  $\theta_i/\theta_0$ , where  $\theta_0$  is the final stop position. The FNN is trained and implemented in Python 2.7 using Keras<sup>(3)</sup> and Theano<sup>(4)</sup> as backend. It is trained for 10000 epochs and optimized using stochastic gradient descent<sup>(3)</sup> (learning rate: 0.01).

**3.2 Validation Experiment** After the weights are trained, the FNN outputs the estimated timings  $t_1$  and  $t_2$  from each step response for the other seven loads which are unused in the training. The results are shown in  $t_1$  and  $t_2$  of Table 1. In the cases (A) of Load1, Load2 and Load6, the estimated  $t_1$  and  $t_2$  are optimal or nearly optimal timings, compared with  $t_{1t}$  and  $t_{2t}$ . While, in the cases (B) of Load3, Load4, Load5, Load 8 and Load 9, the timings are not optimal. To validate

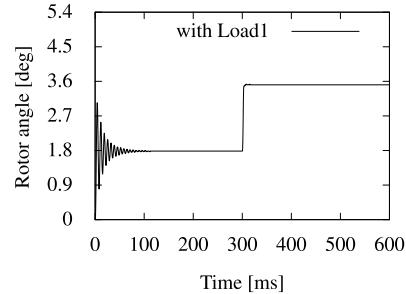


Fig. 3. The step response before and after tuning pulse timings in case of Load1

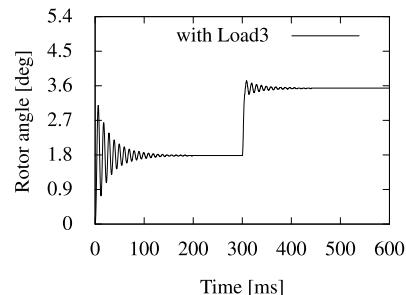


Fig. 4. The step response before and after tuning pulse timings in case of Load3

the scheme in term of step response, the experiment is conducted as follows. First, a step response of one step angle in one-phase excitation drive is derived from 0 to 300 ms. Next, the FNN outputs the pulse timings from the response. Finally, the motor drives for one step angle using the timings in back-phasing damping, and then its response is obtained from 300 to 600 ms. The cases of Load1 and Load3 as the best and worst estimated results of (A) and (B) are shown in Figs. 3 and 4, respectively. In Load1, because the optimal timings are estimated, the rotor oscillation hardly appears. In Load3, though the response is not optimal, the oscillation is satisfactorily damped on the basis of the load. As a result, the estimated timings are effective for the oscillation damping.

### 4. Conclusion

The drives using the pulse timings in back-phasing damping estimated by the FNN demonstrated the rotor oscillation damping in positioning. Thus, the scheme is effective.

### References

- (1) T. Kenjo and A. Sugawara: Stepping Motors and Their Microprocessor Controls, 2nd ed., Oxford University Press (1994)
- (2) I. Goodfellow, Y. Bengio, and A. Courville: Deep Learning, MIT Press (2016)
- (3) F. Chollet: Deep Learning with Python, Manning Publications (2017)
- (4) Theano Development Team: “Theano: A Python framework for fast computation of mathematical expressions”, arXiv preprint, arXiv:1605.02688 (2016)