Neural Network-Based System Identification and Controller Synthesis for an Industrial Sewing Machine

Il-Hwan Kim, Stanley Fok, Kingsley Fregene, Dong-Hoon Lee, Tae-Seok Oh, and David W. L. Wang

Abstract: The purpose of this paper is to obtain an accurate nonlinear system model to test various control schemes for a motion control system that requires high speed, robustness and accuracy. An industrial sewing machine equipped with a Brushless DC motor is considered. It is modeled by a neural network that is configured as an output-error dynamical system. The identified model is essentially a one step ahead prediction structure in which past inputs and outputs are used to calculate the current output. Using the model, a 2 degree-of-freedom PID controller to compensate the effects of disturbance without degrading tracking performance has been designed. In this experiment, it is not preferable for safety reasons to tune the controller online on the actual machinery. Experimental results confirm that the model is a good approximation of sewing machine dynamics and that the proposed control methodology is effective.

Keywords: 2 DOF PID controller, genetic algorithm, neural network, system identification.

1. INTRODUCTON

To accurately control a system, it is beneficial to first develop a model of the system. The main objective for the modeling task is to obtain a good and reliable tool for analysis and control system development. A good model can be used in off-line controller design and implementation of new advanced control schemes. In some applications, such as in an industrial sewing machine, it may be time consuming or dangerous to tune controllers directly on the machinery. In such cases, an accurate model must be used off-line for the tuning and verification of the controller. While nearly all aspects of modeling and simulation in control systems have now reached a reasonable stage of development, the aspect which remains least satisfactory at the present time is that of representing the loads supplied from systems due to the very wide range of load types.

Most motion control systems driven by motors exhibit nonlinear behavior and are often difficult or unrealistic to model directly using laws of physics. Friction is the main nonlinear element in motion control systems. In general, a linear system allows the use of more sophisticated advanced control schemes to achieve higher performance. Lai 0 identified a nonlinear model with a combination of linear dynamics and friction for the Virtual Reality (VR) Mouse, and used a few friction compensation strategies to linearize the VR Mouse dynamics. Turner 0 applied a creep random search based on Genetic Algorithms to simultaneously identify the linear motor parameters and the nonlinear friction parameters for a stereo camera system. However, various other nonlinear elements exist in a motor driver system. The voltage-source pulse width modulation (PWM) amplifier is used and dead time is required to prevent the shoot-through phenomenon during switching. This dead time causes distorted output voltage and results in a nonlinear effect to the system. In an extreme case, the distorted output voltage produces torque pulsation and instability at low-speed. Hur *et al.* 0 proposed a 2 degree-of freedom (2 DOF) controller employing an inverse current dynamic model and a PI controller to compensate the effects of the dead time for induction motor control. The 2 DOF controllers have also been extensively studied in the area of motion control to suppress disturbances [1, 4-5].

In this paper, we propose a method to obtain an accurate nonlinear system model for a motion control system based on neural networks (NNs). Modeling techniques based on NNs have proven to be quite useful for building good quality models from measured data. If such an NN model is available, various

Manuscript received February 4, 2003; revised October 27, 2003; accepted December 25, 2003. Recommended by Editorial Board member Jae Weon Choi under the direction of Editor Keum-Shik Hong. This work was supported by Kangwon National University in 2001.

Il-Hwan Kim, Dong-Hoon Lee, and Tae-Seok Oh are with the Department of Electrical and Computer Engineering, Kangwon National University, 192-1 Hoyza2-dong, Chunchon, 200-701, Korea (e-mail: ihkim@kangwon.ac.kr, {dhcjs, ots}@cclab.kangwon.ac.kr).

Stanley Fork, Kingsley Fregene, and David Wang are with the Department of Electrical and Computer Engineering, University of Waterloo, Waterloo, Ontario, Canada, N2I 3G1 (e-mail: sfok@alumni.uwaterloo.ca, kocfrege@ieee.org, dwang@kingcong.uwaterloo.ca).

control synthesis approaches may be attempted, even if the controllers themselves are not implemented in neural networks. It is possible to use a number of conventional nonlinear design techniques such as feedback linearization, generalized predictive control, or model linearization followed by a linear design. Another approach is to use a neural network as the controller; e.g., direct inverse control or internal model control [6-9].

A model must be found that combines both robustness and accuracy to the desired extent. As well, the model should be computationally efficient and economical in order to be applied in mass-produced systems. To succeed in fulfilling these criteria, we apply a 2 DOF PID controller to compensate the effects of disturbance without degrading tracking performance for a real-life system modeled in a NN.

For the experimental system we consider a commercial sewing machine. It requires high speed, robustness and accuracy. It is equipped with a BLDC (BrushLess Direct Current) motor. The BLDC motor is widely used as an actuator since it has a high torque-to-weight ratio, is easy to control, and has high efficiency and negligible maintenance requirements. Torque ripples are, however, one of the disadvantages of the BLDC motor and may be considered as a nonlinearity. In the present work, we model an industrial sewing machine, which has a BLDC motor, using NNs and then proceed to develop suitable controller synthesis techniques for such a system. The entire approach is demonstrated experimentally.

Apart from this introductory section, the article is organized as follows: Section 2 describes the setup used for data acquisition in our experiments. The system identification procedure for the sewing machine is detailed in Section 3, where controller synthesis methodology and experimental results are also presented. Concluding remarks are given in Section 4.

2. DATA ACQUISITION

To control and monitor the input command signal and encoder readings, the sewing machine is connected to a computer using a Quanser PCI MultiQ I/O board (www.quanser.com). The foot pedal command signal is intercepted and replaced by an analog output line from the MultiQ board. This allows the foot pedal command signal to be simulated by varying the MultiQ output voltage. The MultiQ board also intercepts the encoder readings from the sewing machine. A diagram of the experimental setup is shown in Fig. 1. This figure also shows the software-block diagram used to control and record the sewing machine response.

Using this system setup, several test command signals were generated and applied to the sewing machine. The test signals included a step function, two chirp functions and a random-type signal. The step response will be used to determine the gains of the designed controller. The chirp functions have a frequency sweep between 0.1 Hz and 3 Hz, since these are typical minimum and maximum rates of a human operated command input. The velocity range in the chirp signal used for the system identification has a maximum of 3,000 RPM. Chirp signals were employed in an attempt to exalt all nonlinear dynamics within the system.

The resulting encoder readings were recorded and converted into velocity, with unit's counts per sample (CPS). The encoder resolution is 1572 counts/ revolution and the sampling time is 1 ms. Thus, 1 CPS is equal to 38.168 RPM. Fig. 2 through Fig. 5 shows the

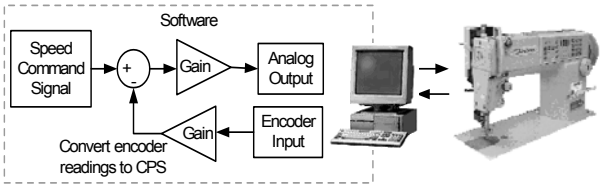

Fig. 1. System diagram.

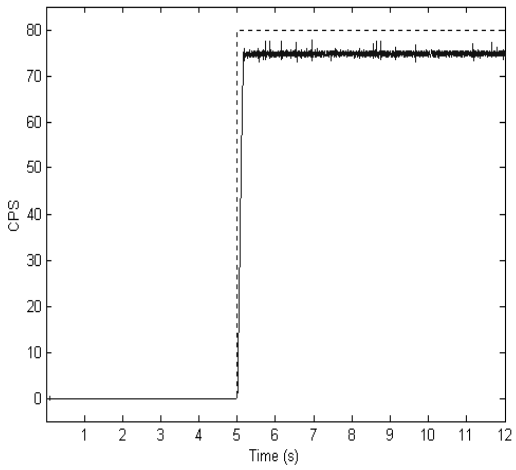

Fig. 2. Step function and response.

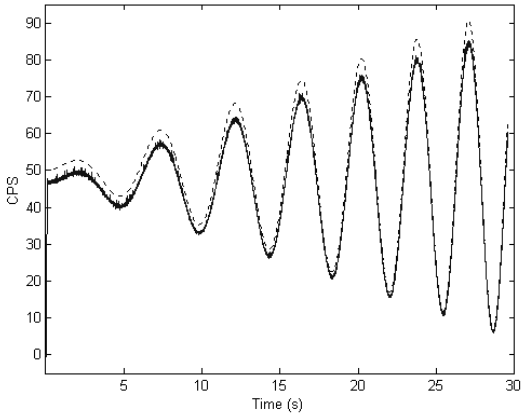

Fig. 3. Chirp (Lo-Hi) function and response.

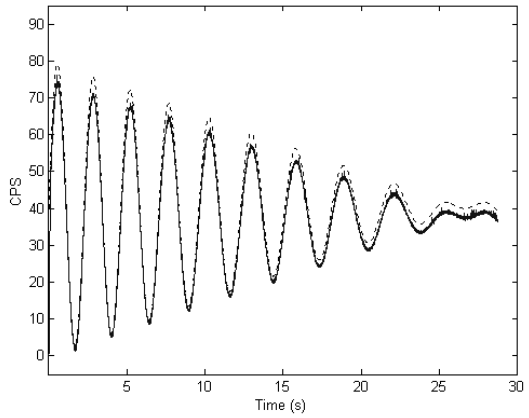

Fig. 4. Chirp (Hi-Lo) function and response.

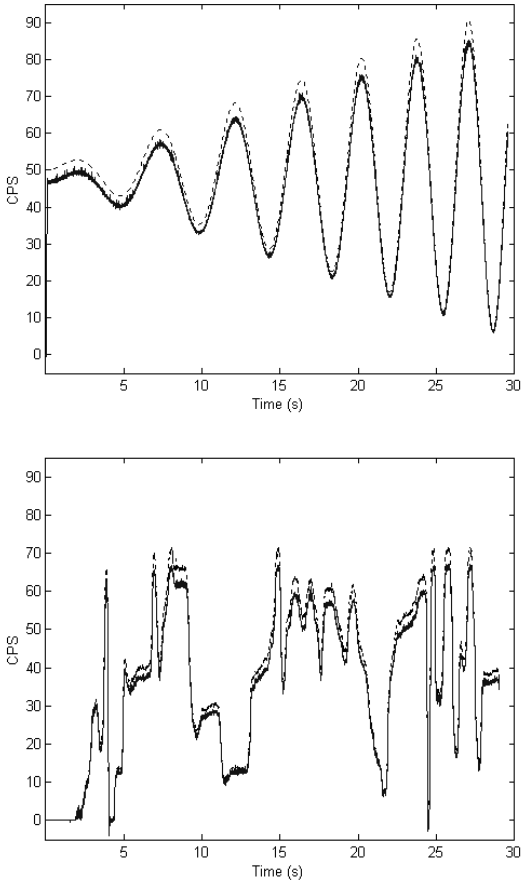

Fig. 5. Random function and response.

input command signals and the sewing machine responses. (In these plots, the dashed line is the reference command signal and the solid line is the sewing machine response.) This set of input-output data will be used for system identification in the next section.

3. SYSTEM IDENTIFICATION

The objective is to carry out system identification of the sewing machine motion system by using neural networks. The input (u) of interest is the voltage applied to the BLDC motor while the output (y) is the rotor speed in RPM. Two sets of data were collected experimentally. For the first set, a chirp signal, which grows progressively in amplitude and frequency, was applied to the motor and the corresponding RPM outputs logged. In the second experiment, the chirp signal shrinks in amplitude with increasing frequency. Only the data obtained from the first experiment was used for training the neural net. Validation is performed using the second data set, which were not used for training. We shall refer to the second data set as the *test data*. It is important to use the test data for validation to ensure that our neural network model does replicate the sewing machine system in general rather than memorize a specific data set.

3.1. System identification using a neural network output error model

In the identification framework, we assume that the sewing machine model can be represented in discrete input-output form by the identification structure:

$$\hat{y}[k] = \hat{g}[y(k-1),...,y(k-n_a), \\ u(k-n_k),...,u(k-n_b - n_k + 1)], \quad (1)$$

where $\hat{y}[k]$ is the one-step ahead prediction of the output; and n_a, n_b, n_k are system order and delay, respectively. This is essentially a *one-step* ahead prediction structure in which we use past inputs and outputs to predict the current output.

Using our intuition concerning the input-output model for the BLDC motor in the sewing machine, a second order system is selected for the identification structure. Therefore, $n_a = n_b = 2$ and $n_k = 1$ in the structure above.

We use the neural network $\hat{g}[\cdot]$ to model $g[\cdot]$. The $[\cdot]$ contains the *regressor structure*, which is implemented as Tapped Delay Lines (TDLs) in code. Therefore, the regressor structure for this network is given by:

$$\phi(k) = [\hat{y}(k-1),...,\hat{y}(k-n_a), \\ u(k-n_k),...,u(k-n_b - n_k + 1)], \quad (2)$$

where $\hat{y}[k]$ are delayed versions of the predicted outputs and $u(\cdot)$ are delayed inputs to the system. At every instant, the predicted output is parameterized in terms of network weights Θ by:

$$y(k,\Theta) = g(\phi(k),\Theta), \quad (3)$$

and is depicted in Fig. 6. Note that the sampling instant k is equivalent to t in all of the figures.

In our model, the $\hat{g}$ network has eight hidden layer neurons with tanh activation functions and a single

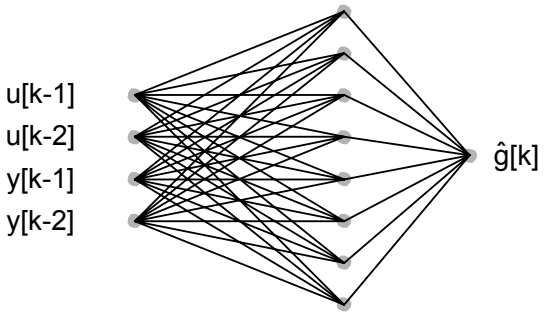

Fig. 6. The architecture for the $\hat{g}[\cdot]$ network.

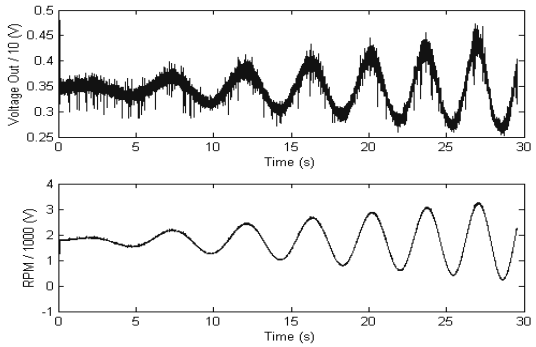

Fig. 7. The scaled training data - voltage input and RPM output.

saturated linear function in the output layer. Network training is first carried out offline in batch form using the Levenberg-Marquadt optimization (rather than conventional back propagation). The networks are is the *prediction error*. So, the algorithm essentially seeks to minimize the prediction error over the training data set. The input and output data sets used for training are obtained by multiplying a chirp signal with a ramp and then applying this to the trained to minimize the cost function

$$J = \frac{1}{2N}\sum_{k=1}^{N}[y(k)-\hat{y}(k)]^T[y(k)-\hat{y}(k)], \tag{4}$$

where

$$e(k) = [y(k)-\hat{y}(k)] \tag{5}$$

actual system. To facilitate training, the voltage input is scaled by 10 while the RPM output is scaled by 1000. The training data sets are depicted in Fig. 7.

One hundred (100) training iterations are performed at the end of which the cost function reduces to the order of 10^{-6}. At this point, the optimal network weights for the $\hat{g}[\cdot]$ networks are stored and used for validation. Validation and cross validation respectively consist of applying the training and test data to the neural identification model in order to see how closely it fits the experimental data from the sewing machine in each case.

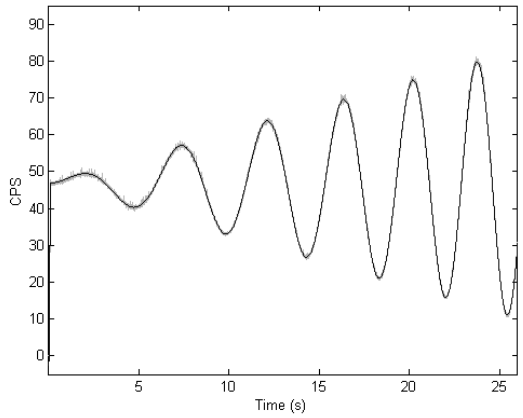

Fig. 8. Validation of the neural model on training data.

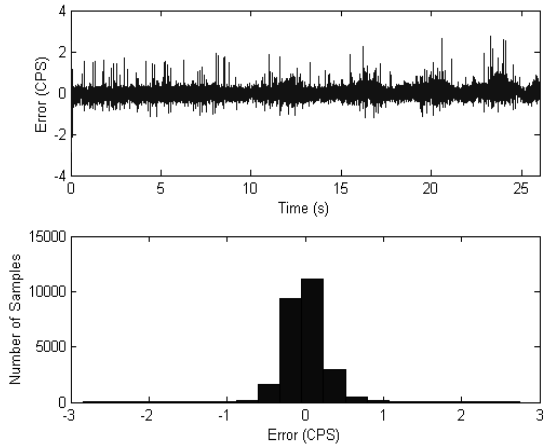

Fig. 9. Prediction errors over the training data.

3.2. Neural network identification results

In validation, we use the training data set as an input to the neural network model of the system and compare the outputs obtained with what was used during training. The voltage input is first scaled by a factor of 10 while the RPM outputs are scaled by a factor of 1000 before training the NN. Accordingly, to recover the original experimental data, we simply multiply the respective inputs and outputs by the scaling factor.

Fig. 8 shows the validation results for the RPM output while the prediction error over training data are shown (in standard and histogram form) in Fig. 9. In Fig. 8, the light gray line represents the actual system response while the dark line represents the predicted response. The results in Fig. 8 are excellent, but they do not necessarily relay an accurate story regarding the network's predictive capability since the networks merely received the same data they were trained on.

To better characterize the network's modeling ability, cross-validation is performed by applying the other data set not used for training to the network. Fig. 10 depicts the RPM predicted vs. actual RPM while Fig. 11 illustrates the prediction errors on the test data. Again, the light gray line represents the actual system

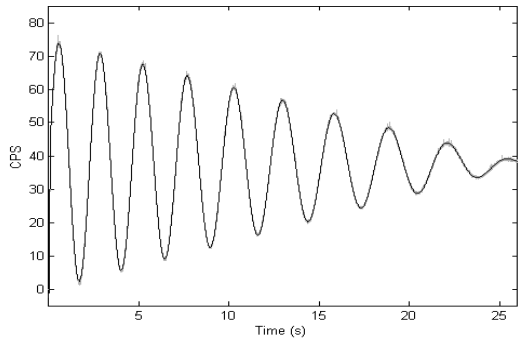

Fig. 10. Validation of the neural model on test data.

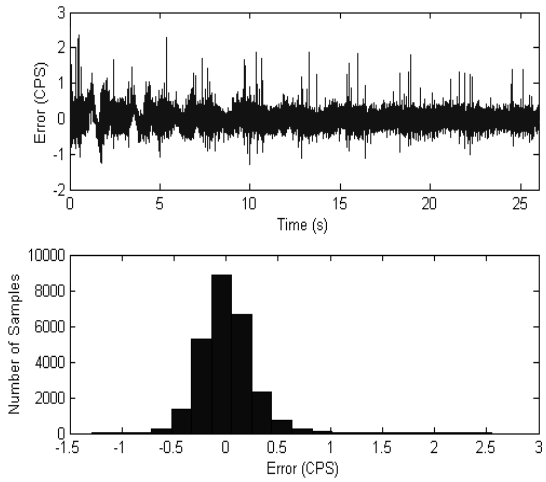

Fig. 11. Prediction errors over the test data.

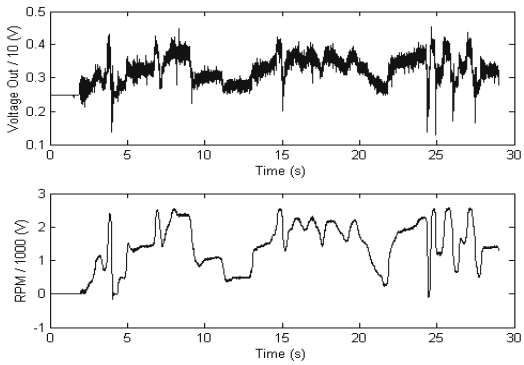

Fig. 12. The scaled secondary random test data - voltage input and RPM output.

response while the dark line represents the predicted response. Observe that the fit is almost perfect and the errors themselves are minimal.

Another experiment was carried out using richer signals that were significantly more random than the modified chirps used for the neural network training.

The scaled input and output sequences are depicted in Fig. 12 while the results that show the NN predicted RPM output versus the actual RPM outputs are shown in Fig. 13. Prediction errors over this test data (which were never used to train the network) are shown in Fig. 14. In all cases the neural network

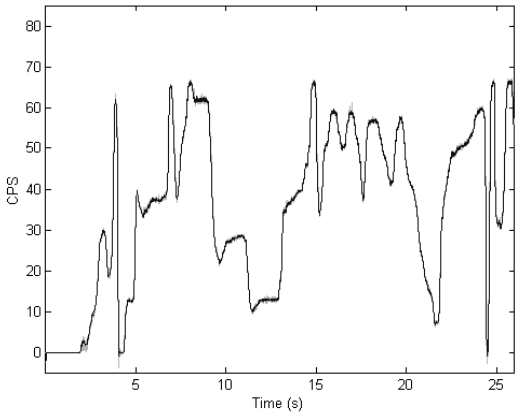

Fig. 13. Validation of the neural model using random test data not employed during network training.

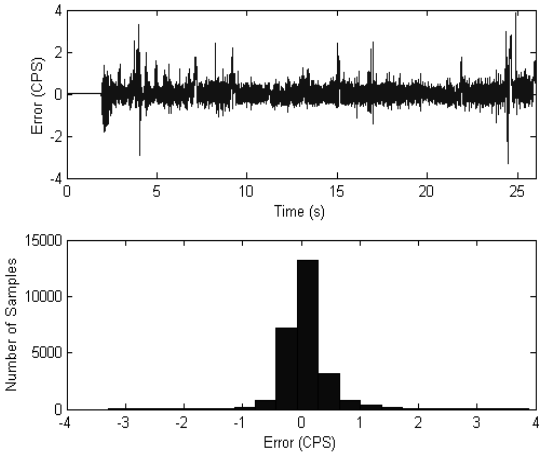

Fig. 14. Prediction errors over the test data.

model performed very well and the prediction errors were not unduly large.

3.3. System identification using a linear arx model

As a comparison, a system identification of theBLDC motor component of the sewing machine was also performed using an ARX model given by the transfer function:

$$G[z] = z^{-k}\frac{B[z]}{A[z]}, \tag{6}$$

where

$$\begin{aligned} A[z] &= 1 + a_1 z^{-1} + \ldots + a_m z^{-m}, \\ B[z] &= b_1 + b_2 z^{-1} + \ldots + b_n z^{-n+1}, \end{aligned} \tag{7}$$

and *m, n, k* are appropriately selected *system, order* and *delay* parameters. The ARX scheme determines a_i and b_i from measured input-output data of the system to be identified.

A 2nd-order linear model (i.e. $m = n = 2$, $k = 1$) was

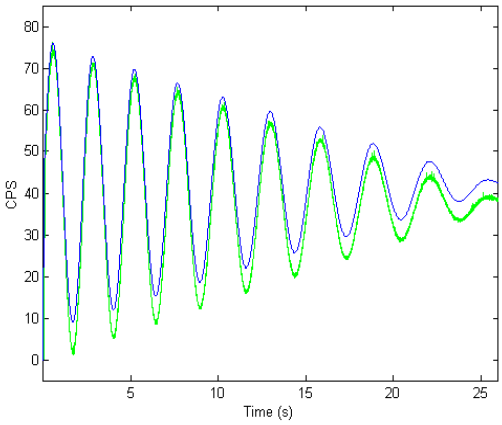

Fig. 15. Chirp (Hi-Lo) ARX model vs. real system response.

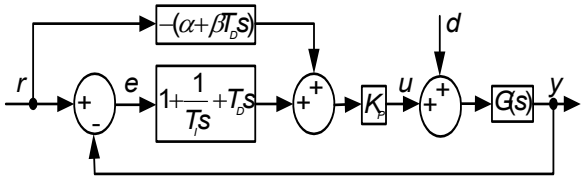

Fig. 16. 2 DOF PID controller (feedforward type).

was extracted by using a random input-output data set with a transfer function given by:

$$G[z]=\frac{0.1228z+0.1224}{z^2-1.87z+0.8705}, \tag{8}$$

and simulation results for this model (using a sampling rate of 1 ms) are depicted in Fig. 15. The gray line represents the actual system response and the dark line represents the predicted system response.

It is clear that the linear identification model does not work as well as the neural network model, although the general trends are noticeable. When random data sets are used, the identification results are significantly degraded.

3.4. Controller design

Fig. 16 shows a general 2 DOF PID controller of feedforward type. Note that the input-output relations are written in the form:

$$\begin{aligned} Wdy(s)&=\frac{G(s)}{1+F_1(s)G(s)},\\ Wry(s)&=\frac{\big(F_1(s)+F_2(s)\big)G(s)}{1+F_1(s)G(s)},\\ Wre(s)&=\frac{1-F_2(s)G(s)}{1+F_1(s)G(s)}, \end{aligned} \tag{9}$$

where,

$$F_1(s)=K_P\left(1+\frac{1}{T_I s}+T_D s\right),$$

$$F_2(s)=-K_p\left(\alpha+\beta T_D s\right),$$

and where $G(s)$ denotes the model for the actual plant, r is the input, y is the output and d is the disturbance. Two important control objectives are command tracking and disturbance rejection.

Command tracking and disturbance rejection are indicated by $Wry = 1$ and $Wdy = 0$, respectively. In a conventional PID controller it is impossible to modify the characteristics of tracking and disturbance rejection separately. However, from equation (9), Wry or Wre and Wdy can be adjusted separately by selecting the two filters F_1 (gain from the command r to the output y) and F_2 (feedforward compensator). This means that the performance of either the tracking or disturbance rejection can be tuned independently without affecting each other.

Insightful ideas concerning the design of F_1 and F_2 are difficult to come by in the case of parameter tuning and so in this paper we take two measures. First, we coarsely tune the parameters based on Table 1 presented by Araki 0. The plant transfer function $G(s)$ is approximated to a 1st order time delay system in the form:

$$G(s)=\frac{Ke^{-sL}}{1+sT}, \tag{10}$$

where K is the proportional gain, L is the delay time and T is the time constant. Normally we can determine these values from the step response, as shown in Fig. 2, since the step response of equation (10) is written by equation (11) below.

The parameters shown in Table 1 are obtained to minimize the evaluation function (12) where $E(s)$ is the Laplace transformation of the error signal $e(t)$:

$$y(t)=K(1-e^{-(t-L)/T}), \tag{11}$$

$$\vartheta=\int_0^{\infty}\left|\sqrt[4]{w}\left\{\frac{d^2}{ds^2}E(s)\Big|_{s=jw}\right\}\right|^2 dw\cdot \tag{12}$$

Table 1. Parameters for the 2 DOF PID controller.

L/T	K_pK	K_I/T	K_D/T	α	β
0.1	12.5	0.22	0.04	0.68	0.75
0.2	6.1	0.41	0.08	0.63	0.70
0.3	4.1	0.57	0.11	0.62	0.70
0.4	3.1	0.71	0.15	0.59	0.69
0.5	2.5	0.83	0.18	0.58	0.69
0.6	2.11	0.94	0.21	0.56	0.69
0.7	1.82	1.05	0.24	0.54	0.69
0.8	1.61	1.13	0.28	0.51	0.68
0.9	1.44	1.22	0.31	0.48	0.69
1.0	1.33	1.26	0.34	0.49	0.66

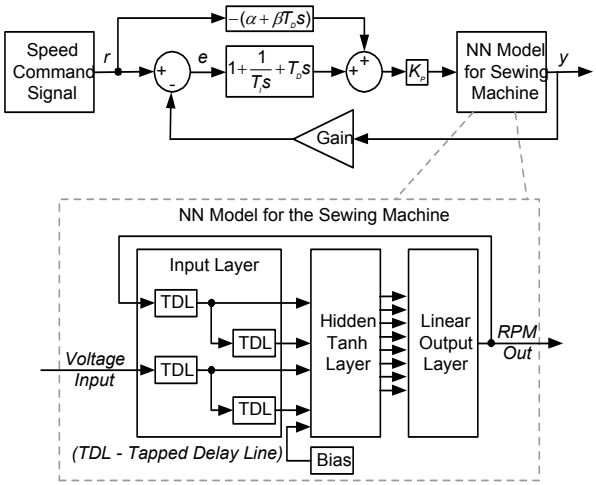

Fig. 17. Simulation block diagram.

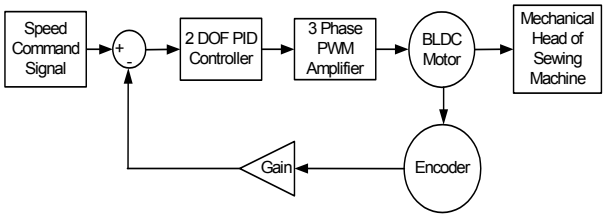

Fig. 18. Experimental system block diagram.

The second step is the fine-tuning of parameters determined by Table 1 through simulation using the identified nonlinear model in the previous section as shown in Fig. 17.

Our key focus is to minimize the delay of the system response so that the output y tracks the input as closely as possible. The controller gains, K_p, K_i and K_d, are tuned to provide the best tracking with minimalovershoot and vibration while keeping the control signal within the permitted voltage ranges for the MultiQ analog output.

In this paper the tuned gains are $K_p = 0.02$, $K_i = 0.02$, $K_d = 0.5$, $\alpha = 0.4$, $\beta = 0.2$ when the sampling time is 1 ms.

3.5. Experimental results

The results of applying the controller to the system simulation and the actual system are discussed in this section. Fig. 18 shows the experimental setup used. For these experiments, a TMS320F240 digital signal processor was used for the controller and PWM signals.

First, the identified system model is used with the designed 2 DOF PID controller. Then the same controller is applied to the actual system. Fig. 19 through Fig. 22 indicate these results. The dark solid line refers to the reference signal, the dark dashed line refersline represents the actual real system response.

These results show that controlled velocity output of the simulated and real system matches the reference

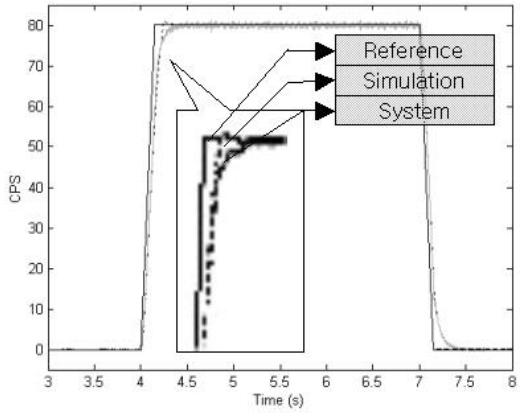

Fig. 19. Trapezoidal simulated and real controller tests.

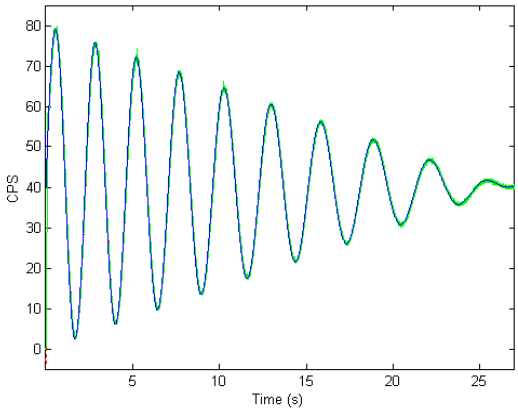

Fig. 20. Chirp (Hi-Lo) simulated and real controller tests.

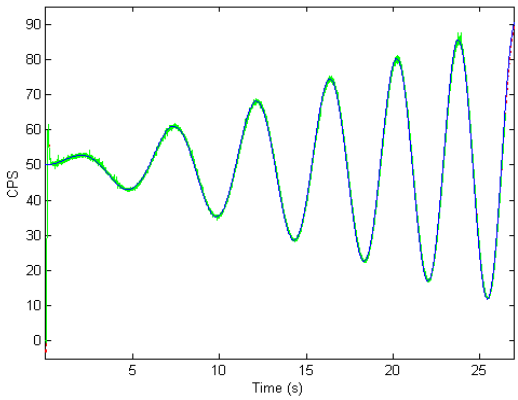

Fig. 21. Chirp (Lo-Hi) simulated and real controller tests.

signals very closely in the case of the chirp and random signals. Fig. 19 through Fig. 21 detail that the simulated and real responses of the sewing machine are almost exactly the same as the reference signals. The fact that the simulated and real controlled responses are practically identical further validate the identified system model.

The trapezoidal reference signal produced greater errors in both the simulated and actual systems. However,

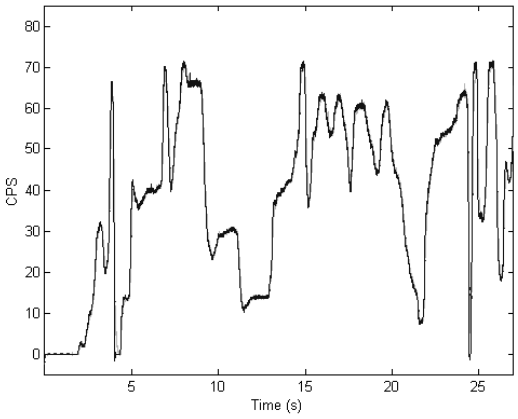

Fig. 22. Random simulated and real controller tests.

the resulting velocity profiles were still very close to the reference signal. The real trapezoidal response had a rise time of 150 ms. The simulated and real responses also differed slightly. This can be attributed to the system identification, which only used the chirp function data for modeling the system. This data did not contain any sharp edges or sudden changes and thus the identified system model does not behave like the real system when stimulated by very sharp velocity changes. Increasing the frequency sweep of the chirp signal should improve the identified system model for step and trapezoidal responses.

4. CONCLUSIONS

In this paper, we developed the nonlinear network model for a commercial sewing machine equipped with a BLDC motor. The identified model using neural networks is essentially a one step ahead prediction structure in which past inputs and outputs are used to predict the current output.

Using the model, a 2 degree-of-freedom PID controller to compensate the effects of disturbance without degrading tracking performance has been designed. With the experimental results, the model has been shown to be a good approximation of the sewing machine and the proposed method demonstrates the effectiveness for a motion control system that requires high speed, robustness and accuracy.

In further work, a Genetic Algorithm, which has been found particularly useful for optimization and searching, may be used to tune the gains of the 2 DOF PID controller to minimize the error between the command input and the identified system model output.

REFERENCES

[1] N. Hur, K. Nam, and S. Won, "A two-degrees-of-freedom current control scheme for deadtime compensation," *IEEE Trans. on Industrial Electronics*, vol. 47, pp. 557-564, June 2000.

[2] G. M. Y. Lai, *Investigation On a Haptic Device for Teleoperation*, Master's Thesis, University of Waterloo, 1999.

[3] C. F. Turner, *Development of an Internet Visual Telepresence System*, Master's Thesis, University of Waterloo, 2002.

[4] T. Sugie and T. Yoshikawa, "General solution of robust tracking problem in two-degree-of-freedom control system," *IEEE Trans. on Automat. Contr.,* vol. AC-31, pp. 552-554, 1986.

[5] K. Araki, "Two-degree-of-freedom controller I-PID, differential feedforward, I-PD controller," *System and Control,* vol. 29, pp. 649-656, 1985.

[6] Zhang Huaguang and Quan Yongbing, "Modeling, identification, and control of a class of nonlinear systems," *IEEE Trans. on Fuzzy Systems*, vol. 9, no. 2, pp. 349-353, 2001.

[7] K. K. Safak and O. S. Turkay, "Experimental identification of universal motor dynamics using neural networks," *Mechatronics,* vol. 10, pp. 881-896, 2000.

[8] B. M. Novakovic, "Discrete time neural network synthesis using input and output activation functions," *IEEE Trans. on SMC*, vol. 26, no. 4, pp. 533-541, 1996.

[9] D. Schroder, C. Hintz, and M. Rau, "Intelligent modeling, observation, and control for nonlinear systems," *IEEE Trans. on Mechatronics*, vol. 6, no. 2, pp. 122-131, 2001.

Il-Hwan Kim received the B.S. and M.S. degrees in Control and Instrument Engineering from Seoul National University in 1982 and 1985 respectively and the Ph.D. at Tohoku University in 1993. In 1995, he joined the Department of Electrical and Computer Engineering at Kangwon National University and is currently an Associate Professor. His research interests include control, mechatronics, and human interfaces.

Stanley Fok received the B.A.Sc degree in Computer Engineering from the University of Waterloo, Canada in 2001. He also completed the M.A.Sc. degree in Electrical & Computer Engineering at the same institution in 2002. His research interests are in controls, digital signal processing, image and video processing and compression.

Kingsley Fregene received the Bachelors degree (summa cum laude) from the Federal University of Technology, Owerri, Nigeria and the M.A.Sc. and Ph.D. degrees from the University of Waterloo, Canada in 1996, 1999 and 2002 respectively, all in Electrical Engineering. He is a member of the IEEE Control Systems Society and the ASME. He is currently a Research Scientist with Honeywell Labs in Minneapolis, USA. His research interests include intelligent control and distributed multi-agent control. He was co-chair of the Neural Networks track at the 1999 IEEE International Symposium on Intelligent Control and the author or co-author of several refereed technical publications.

Dong- Hoon Lee received the B.S. and M.S. degrees in Control and Instrument Engineering from Kangwon National University in 1998 and 2001 respectively. He is currently working toward the Ph.D. degree at the same institution. His research interests include motor control and mechatronics.

Tae-Seok Oh received the B.S. and M.S. degrees in Control and Instrument Engineering from Kangwon National University in 1998 and 2001 respectively. He is currently working toward the Ph.D. degree at the same institution. His research interests include motor control and mechatronics.

David W. L. Wang received the B.E. at the University of Saskatchewan in 1984, and the M.A.Sc. and Ph.D. at the University of Waterloo in 1986 and 1989 respectively. In 1990, he joined the Department of Electrical and Computer Engineering at the University of Waterloo and is currently an Associate Professor. His research interests include nonlinear control, mechatronics, flexible manipulators/structures, shape memory alloy actuators and haptic interfaces.