Nonlinear Function Approximation for Mechanical System Modeling using Fourier Series Neural Network

Kajorndaj Phimphilai
Department of Mechanical Engineering, Faculty of Engineering, Chiang Mai University
Chiang Mai 50200, Thailand
Tel: 0-5394-4146, Fax: 0-5394-4145, E-mail: kphimph@yahoo.com

Abstract

Mostly, physical systems are nonlinear and their dynamics do not lead to straightforward system analysis procedures. The modeling of mechanical systems requires accurate knowledge and understanding of their dynamics. However, the procedures to identify and analyze nonlinear systems are often restricted by limited and inaccurate system knowledge. From these reasons, the problem can be addressed by considering new methodology that is not totally dependent on precise system knowledge in order to obtain accurate system dynamical model. This research is proposed to apply Fourier Series Neural Network (FSNN) that has universal function approximation properties to learn nonlinearity in mechanical systems. Utilization of the network promises to provide auxiliary tool for identifying and modeling nonlinear dynamical effects consisting in mechanical systems. Performance based on simulation for the FSNN compensation adaptive controller was conducted on a rotational wheel system together with brake-pad device for generating stick-slip frictional effects. The FSNN was employed as a nonlinear dynamics approximation controller in the control strategy that identifies friction embedded in the system. The weights adaptation and stability of control system were confirmed using Lyapunov analysis. The tracking error was proved to be bounded and PD controller is necessary to ensure the overall stability.

Keywords: Fourier series neural network, Adaptive controller, Function approximation, System identification, System modeling

1. Introduction

Adaptive controllers are implemented to learn and adapt to unknown constant parameters from known system dynamic function. The well-developed adaptive controllers represent high performance controllers for system control [1]. The accuracy of controlled system improves along with task performing because of the adaptation mechanism. The controllers also give consistent performance in situation of environment variations [2]. The adaptive computed-torque method is modified from approximated computed torque controller by applying adaptive updated rule for parameter estimation [3]. The PD controller is implemented with adaptive controller to ensure the stability of the overall system in case there existed mismatch between approximated and exact dynamics equation.

Recently, there has been increasing interest in the use of intelligent control technique such as fuzzy logic and neural networks. Generally, the controls rely on learning the input-output-behavior of the plant to be controlled. Neural networks have been applied extensively to identify and control of nonlinear dynamic systems. The convergence of neural networks is never assured because of the nonlinear activation function presence within the networks [4]. The networks also have another limitation that required initial weights to be properly chosen to achieve an appropriate convergence. However, the neural network is still a convenient approach to address complex and nonlinear dynamic systems.

A multilayered neural network was proposed for identify a nonlinear dynamic plant with the presence of disturbance [5]. Due to neural network ability to model nonlinear dynamical systems, it has been investigated for various control applications [6]. Various neural network architectures have been developed and applied for dynamic system identification. Recurrent networks with different feedback have been widely applied in the area of nonlinear system modeling. A hybrid neural network was implemented with a conventional two-layered feedforward neural network to identify viscous torque and asperity contact torque in wet friction component in transmission system [7]. The neural network control scheme was successfully capturing friction characteristic of the friction component as a function of time.

Orthogonal activation function neural networks have an advantage over conventional neural networks. One unique characteristic is that they demonstrate no local minima. The absence of local minima assures the convergence of the training that is the limitation of conventional neural networks. For orthogonal activation functions, the auto correlation matrix is a diagonal matrix with all equal eigenvalues. These equal eigenvalues result in an extraordinary characteristic that they have hyperspheroid error contour projection surface in n-dimensional network weight space. The symmetry of error contour enhances the possibility for small number of training cycles [8].

An orthonormal activation function neural network such as the Fourier series neural network (FSNN) has been used for dynamic system identifier [8,9,10,11]. A Single-Input-Single-Output (SISO) FSNN was developed and analyzed their coefficients of *sin* and *cos* terms, including identification of system transfer function and describing functions. The FSNN was compared with other orthonormal activation function neural networks and showed good performance in work by Shukla and Paul [8]. The FSNN stability is guaranteed if the learning rate does not exceed the maximum-learning rate. This work showed that the Fourier series neural network is another approach to identify nonlinear dynamic systems. The network also provides flexibility to implement to dynamic systems without a prior knowledge of such systems.

2. Statement of the problem

An accurate system model is very important for precision and performance in such engineering application. The modeling of mechanical systems requires accurate knowledge and understanding of their dynamics. However, the procedures to identify and analyze nonlinear systems are often restricted by limited and inaccurate system knowledge. Frequently, the nonlinearity embedding in physical systems enhances complexity and difficulties for the analysis. This kind of systems can be found in many control applications such as robot, vehicles and motor-load systems. Control of these physical systems is crucial for any tasks that the systems involve. For example, the robot will be useless if the controller cannot command the robot arm to follow such desired trajectory. In addition, it can damage product items that the robot is assigned to handle. These systems have complex nonlinear dynamics that make accurate and robust control difficult. From these reasons, control of physical systems with nonlinear dynamics requires well-developed adaptive controllers. The objective of this research is to understand, verify and demonstrate the applicability of Fourier Series Neural Network (FSNN) nonlinear adaptive control. Based on this study, the research will consider the learning characteristics of FSNN.

The proposed control architecture consists of a PD controller and FSNN for fast learning and guarantee convergent of Fourier series neural network characteristics. The control scheme has a benefit from FSNN in enhancing the performance of a conventional PD control strategy for uncertainties compensation. Examples of uncertainties can be imperfection of system dynamic model, load fluctuation and friction.

The FSNN receives desired trajectory signal and error signal as inputs for learning system dynamics by adjusting its state weights inside the network. The weight adaptation objective is to minimize the feedback error from the controlled system. While the FSNN compensator is trained, it generates sending out signal as an additional control force onto the system. Eventually, the FSNN will gain knowledgeable of system dynamics and then provides an appropriate force signal for the system. At this point, the FSNN applies the force signal that compensates for imperfect knowledge of the controlled system dynamic model and any other uncertainties that occur during the operation.

Thus, the advantages of utilizing FSNN compensator within the control architecture are more accurate performance and more flexibility in order to control different mechanical systems from the same control platform. In addition, the controller does not need either a priori or near perfect knowledgeable of the controlled system in order to achieve an accurate trajectory output because the FSNN compensator within the control architecture has an ability to compensate for the imperfection of the system model and uncertainties.

3. Fourier series neural network

Fourier series expansion contains an orthogonal set which has a fixed structure depending on the number of variables and based frequencies. The multiple Fourier series is comparable to a feedforward neural network with a single hidden layer as shown in Figure 1. This particular neural network is a Fourier series neural network (FSNN). Inputs fed into the FSNN at the input nodes are variables of the function. They will be arranged to be harmonic functions, *sin* and *cos* terms, within harmonic neurons. The harmonic terms are multiplied to generate an orthogonal set which are compatible with multiple Fourier series expansion terms. Each orthogonal term is then multiplied by a state weight. Finally all product terms will be combined together with a bias weight to generate the network output.

The state weights and the bias weight, which are well matched with the coefficient of each Fourier series term, are determined utilizing the Delta Rule (DR) method. The DR is a product-learning rule for a feedforward, single-layer, structured neural network using gradient descent to achieve training or learning by error correction. The network's weights are adjusted in the direction of minimizing the difference between the desired value and actual output from the network.

Suppose the best fit Fourier series that represents a function is

$$\hat{y}(X,W) = \sum_{n1=0}^{N1} \sum_{nm=0}^{Nm} W_{n1..nm} h_{n1}(x_1)..h_{nm}(x_m) \quad (1),$$

where *h(.)* is a complex harmonic activation function with *W*'s as the state weights.

The equation approximates a multiple Fourier series if weights $W_{n1..nm}$ are trained to approach the coefficients of Fourier series. The FSNN model with real sine and cosine activation functions can be expressed as

$$\hat{y}(X,W) = \sum_{n1=0}^{N1} \sum_{nm=0}^{Nm} [W^{c}_{n1..nm} \cos(\Omega_{n1..nm} \bullet X) + W^{S}_{n1..nm} \sin(\Omega_{n1..nm} \bullet X)]$$

(2),

where vector $\Omega_{n1..nm}$ = ($n_1\omega_{01}$, …, $n_m\omega_{01}$) and the '●' between $\Omega_{n1..nm}$ and *X* denotes the dot product operation of the two vectors.

The multiple Fourier series for a function can be reformed with the coefficients given by

$$y_d(X) = \sum_{n1=0}^{\infty} \sum_{nm=0}^{\infty} [A_{n1..nm} \cos(\Omega_{n1..nm} \bullet X) + B_{n1..nm} \sin(\Omega_{n1..nm} \bullet X)]$$

(3)

Let the FSNN learning error E_l is the difference between the desired output y_d and actual output from FSNN model $\hat{y}$

$$E_l = y_d - \hat{y}$$

(4)

$$= \sum_{n1=0}^{N1} \sum_{nm=0}^{Nm} [(A_{n1..nm} - W^{c}_{n1..nm}) \cos(\Omega_{n1..nm} \bullet X) + (B_{n1..nm} - W^{S}_{n1..nm}) \sin(\Omega_{n1..nm} \bullet X)] + \sum_{n1=N1+1}^{\infty} .. \sum_{nm=Nm+1}^{\infty} [A_{n1..nm} \cos(\Omega_{n1..nm} \bullet X) + B_{n1..nm} \sin(\Omega_{n1..nm} \bullet X)]$$

(5)

The second multiple contains higher order terms, which can be neglected if N_i's (*i* = 1,.., *m*) are large. Thus the learning error can be approximated by

$$E_l = \sum_{n1=0}^{N1} \sum_{nm=0}^{Nm} [(A_{n1..nm} - W^{c}_{n1..nm}) \cos(\Omega_{n1..nm} \bullet X) + (B_{n1..nm} - W^{S}_{n1..nm}) \sin(\Omega_{n1..nm} \bullet X)]$$

(6)

From Delta rule, the state weight is updated as

$$\Delta W_x = -\eta \frac{\partial V}{\partial W_x}$$

(7),

where W_x = state weight for a particular orthogonal term and η = learning rate

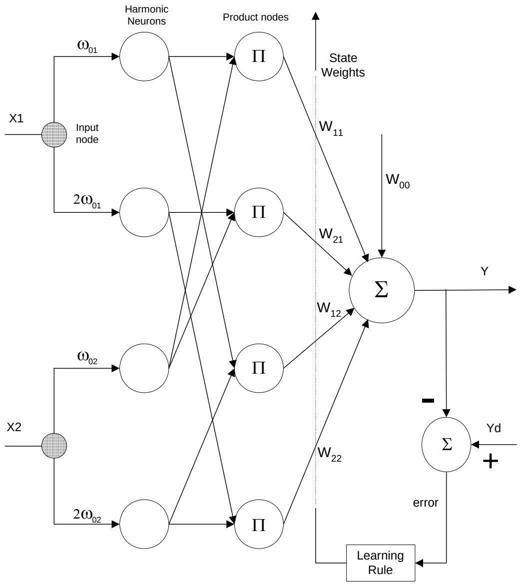

Figure 1. FSNN architecture (2 inputs)

4. Model equation and control scheme

A rotational wheel system together with brake-pad device for generating stick-slip frictional effects using in this work is shown in Figure 2. The system itself is not complicated; however, it includes comparable nonlinearities in the real mechanical system as shown in Figure 3. We assume that there is no disturbance torque applied onto the system. The system dynamical equation can be written as follow

$$\tau = J\ddot{q} + B\dot{q} + F(\dot{q})$$

(8)

where J is inertia term

B is damping coefficient

$F(\dot{q})$ is frictional effect

Figure 2. Rotational wheel system with brake-pad device

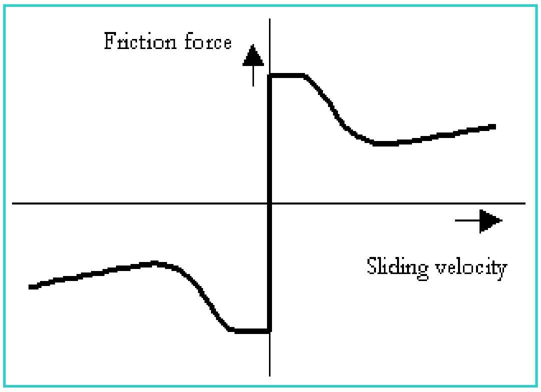

Figure 3. Stick-slip friction with stribeck effect

The Fourier series neural network (FSNN) controller was implemented to control rotational wheel system model in order to verify its performance. Fourier series neural network (FSNN) controller diagram is shown in Figure 3. In this controller, we proposed to use a filtered-error-based approach, employing the FSNN to approximate unknown nonlinear functions in the system dynamics, there by overcoming some limitations of the adaptive control. Instead of requiring knowledge of the system dynamics, as needed in both regression adaptive control and the robust control, the FSNN is responsible to adapt its weights on-line to learn the unknown system dynamics. The controller may pledge to provide a model-free learning controller for a class of nonlinear adaptive control systems.

The control input for the FSNN controller can be written as

$$\tau = \hat{W}^T \phi(x) + K_v r$$

(9),

where $\hat{W}^T \phi(x)$ is control input from the FSNN.

$K_v r$ is PD control input

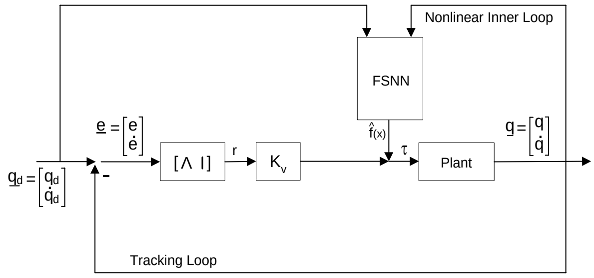

Figure 4. FSNN controller diagram

5. Stability analysis

Dynamical equation of a rotational wheel system together with stick-slip frictional effects can be generally written as Equation 8. Assign error term (*e*) as difference between desired and actual angular displacement

$$e = q_d - q$$

(10)

$$r = \dot{e} + \lambda e$$

(11)

$$\dot{r} = \ddot{e} + \lambda \dot{e}$$

(12)

where *r* is filter error term

substitutes Equation 10 into Equation 8 we obtain

$$J(\ddot{q}_d - \ddot{e}) + B(\dot{q}_d - \dot{e}) + F(\dot{q}) = \tau$$

(13)

substitutes Equations 11 and 12 into Equation 13 we obtain

$$J(\ddot{q}_d - \dot{r} + \lambda \dot{e}) + B(\dot{q}_d - r + \lambda e) + F(\dot{q}) = \tau \quad (14)$$

rearrange the above equation in filter error expression.

$$J\dot{r} = -Br + J(\ddot{q}_d + \lambda \dot{e})$$
$$+ B(\dot{q}_d + \lambda e) + F(\dot{q}) - \tau$$

(15)

Finally, we obtain dynamic equation of the rotational wheel system express in filter error form as follows:

$$J\dot{r} = -Br + f(\dot{q}) - \tau$$

(16)

where

$$f(\dot{q}) = J(\ddot{q}_d + \lambda \dot{e}) + B(\dot{q}_d + \lambda e) + F(\dot{q}) \quad (17)$$

The FSNN approximates nonlinear function (Equation 17) with ideal weights can be written as

$$f(\dot{q}) = W^T \phi(\dot{q}) + \varepsilon \quad (18),$$

where ε is net functional reconstruction error.

Substitutes Equations 9 and 18 into Equation 16, we obtain system dynamic equation as followed:

$$J\dot{r} = -Br + W^T \phi(\dot{q}) + \varepsilon - \hat{W}^T \phi(\dot{q}) - K_v r \quad (19)$$

$$J\dot{r} = -(K_v + B)r + \tilde{W}^T \phi(\dot{q}) + \varepsilon$$

(20)

where $\tilde{W} = W - \hat{W}$ (21)

Assign Lyapunov function as:

$$L = \tfrac{1}{2} r^T J r + \tfrac{1}{2}\left\{\tilde{W}^T F^{-1} \tilde{W}\right\}$$

(22)

Differentiates the Lyapunov function, we obtain

$$\dot{L} = r^T J\dot{r} + \tfrac{1}{2} r^T \dot{J} r + \left\{\tilde{W}^T F^{-1} \dot{\tilde{W}}\right\} \quad (23)$$

We first assume that there is no net functional reconstruction error. Substitutes dynamic equation in filter error form (Equation 20) into Equation 23

$$\dot{L} = r^T\left[-(K_v + B)r + \tilde{W}^T \phi(\dot{q})\right] + \tfrac{1}{2} r^T \dot{J} r + \left\{\tilde{W}^T F^{-1} \dot{\tilde{W}}\right\} \quad (24)$$

$$\dot{L} = -r^T K_v r + \left\{\tilde{W}^T \left(F^{-1}\dot{\tilde{W}} + \phi(\dot{q}) r^T\right)\right\} \quad (25)$$

if we design term

$$\dot{\tilde{W}} = -F\phi(\dot{q}) r^T \quad (26)$$

which causes the second term to be zero.

We differentiate Equation 21 to obtain

$$\dot{\tilde{W}} = \dot{W} - \dot{\hat{W}} = -\dot{\hat{W}} \quad (27)$$

thus

$$\dot{\hat{W}} = F\phi(\dot{q}) r^T$$

(28)

which is Fourier series neural network weights adaptation algorithm

Differentiate of Lyapunov function in Equation 25 can be shorten to

$$\dot{L} = -r^T K_v r \quad (29)$$

Since $L > 0$ and $\dot{L} \le 0$ resulting in *r* and $\tilde{W}$ are bounded. The system is stable in the sense of Lyapunov (SISL).

From Barbalet's lemma, if $\dot{L}$ is uniformly continuous, then $\dot{L} \to 0$ as $t \to \infty$.

$\dot{L}$ is uniformly continuous only if $\dot{L}$ is continuous and $\ddot{L}$ is bounded.

Differentiate Equation 29 we obtain

$$\ddot{L} = -2r^T K_v \dot{r}$$

(30)

From system dynamic equation (Equation 20), we can rearrange it into the following form

$$\dot{r} = J^{-1}\left(-(K_v + B)r + \tilde{W}^T \phi(\dot{q})\right) \quad (31)$$

We know that *r*, $\tilde{W}$, $M(q)^{-1}, B$ and K_v are bounded. So $\dot{r}$ showing in Equation 35 is bounded.

From these reasons, $\ddot{L}$ is bounded, $\dot{L}$ is continuous so $\dot{L}$ is uniformly continuous. $\dot{L} \to 0$ as $t \to \infty$. $r(t) \to 0$ as $t \to \infty$

If there is no net functional reconstruction error, the error will go to zero as time goes to infinity.

In real application, there still exists net reconstruction error from the network. Then, the system dynamic Equation 20 can be rewritten as

$$J\dot{r} = -(K_v + B)r + \tilde{W}^T \phi(\dot{q}) + \varepsilon$$

(32) Assign Lyapunov function as

$$L = \tfrac{1}{2} r^T J r + \tfrac{1}{2}\left\{\tilde{W}^T F^{-1} \tilde{W}\right\} \quad (33)$$

differentiates

$$\dot{L} = r^T J\dot{r} + \tfrac{1}{2} r^T \dot{J} r + \left\{\tilde{W}^T F^{-1} \dot{\tilde{W}}\right\}$$

(34)

substitutes Equation 32 into Equation 34

$$\dot{L} = r^T\left[-(K_v + B)r + \tilde{W}^T \phi(\dot{q}) + \varepsilon\right] + \tfrac{1}{2} r^T \dot{J} r + \left\{\tilde{W}^T F^{-1} \dot{\tilde{W}}\right\}$$

(35)

$$\dot{L} = -r^T K_v r + \left\{\tilde{W}^T \left(F^{-1}\dot{\tilde{W}} + \phi(\dot{q}) r^T\right)\right\} + r^T \varepsilon \quad (36)$$

if we design term

$$\dot{\tilde{W}} = -F\phi(\dot{q}) r^T$$

(37)

this will cause the second term to be zero.

Since $\dot{\tilde{W}} = \dot{W} - \dot{\hat{W}} = -\dot{\hat{W}}$ so that

$$\dot{\hat{W}} = F\phi(\dot{q}) r^T \quad (38)$$

which Equation 38 is Fourier series neural network weight adaptation algorithm.

So differentiate of Lyapunov function in Equation 36 will be shorten to

$$\dot{L} = -r^T K_v r + r^T \varepsilon$$

(39)

We can see from Equation 39 that if the first term larger or equal the second term, $\dot{L}$ will be negative semidefinite which results in stable system. So if K_v is designed to be large enough, then $L > 0$ and $\dot{L} \le 0$. This resulting in *r* and $\tilde{W}$ are bounded. The system is stable in the sense of Lyapunov (SISL).

6. Simulations result

The Fourier series neural network (FSNN) controller was implemented to control rotational wheel system together with stick-slip friction model in order to verify its performance. The controller consists of partial plant inverse dynamic that exclude friction model and FSNN compensation controller to learn friction characteristics. The PD controller is also used for guarantee stability of overall system. As a result, the controller needs partial prior knowledge of the system dynamics.

Parameters of the rotational wheel system for the simulation are designed as followed: J = 0.003164 kg-m^2, $F(\dot{q}) = 10 sign(\dot{q})$ N-m.

The desired input trajectory of the rotational wheel is given by

$$q_d = 0.5\sin(t) \quad (40)$$

where q_d represent angular position the rotational wheel

The plot for desired input trajectory for rotational wheel is shown in Figures 5. The simulations were performed using Simulink simulation software (The Math Works, Inc.).

The data was collected at the interval of 0.01 sec. Both PD controller gains, K_v and K_p, were designed to be 2 and 30, respectively. The gains were selected such that the close loop system response behaves critical damped.

All simulations were performed from 0 to 40 sec. The simulation time length was set long enough in order to let FSNN acquired friction characteristics. There were 2 different size FSNN; 2-frequency-based with 17 neurons and 5-frequency-based with 101 neurons, in order to observed FSNN behavior and evaluated their performance.

In the first simulation, 2-frequency-based with 17 neurons, the best possible learning rate for FSNN weight adaptation was determined and set at 25% of maximum permissible learning rate. We can see that the trajectory friction approximation start decreasing right after the networks was adapting their weights in order to minimize the feedback error from the rotational wheel.

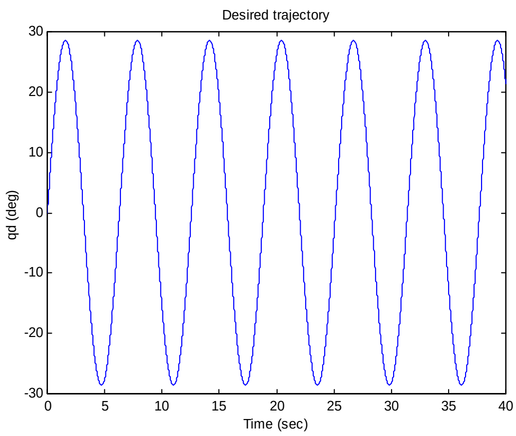

Figure 5. Desired input trajectory

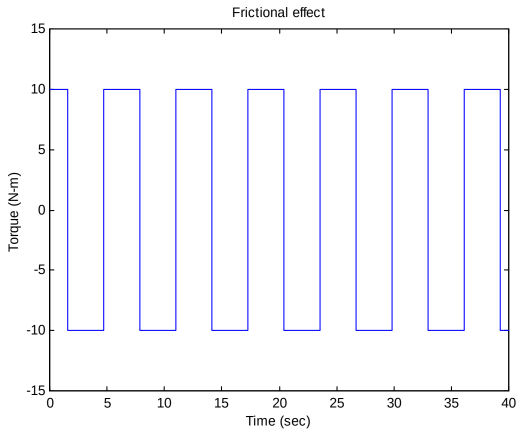

Figure 6. Friction characteristics plot

Approximately after 25 sec, the FSNN gains proper knowledge of system dynamics by which their weight values were converged. Eventually FSNN controller provided an appropriate compensation control torque for friction effect onto the rotational wheel using its available weight value. The control torque plot from FSNN was shown in Figure 7. The best approximated frictional effect from FSNN is not as sharp as in the model one because the simulation uses the 2-frequency-based FSNN.

In the other simulation, 5-frequency-based with 101 neurons, the best possible learning rate for FSNN weight adaptation was also determined and set at 25% of maximum permissible learning rate. As we found before, the trajectory friction approximation start decreasing right after the networks was adapting their weights in order to minimize the feedback error from the rotational wheel. About after 25 sec, the FSNN gains proper knowledge of system dynamics by which their weight values were converged. Eventually FSNN controller provided an appropriate compensation control torque for friction effect onto the rotational wheel using its available weight value. The control torque plot from FSNN was shown in Figure 8. The final approximated frictional effect from FSNN is better than in the previous one because the simulation uses higher number of frequency-based FSNN.

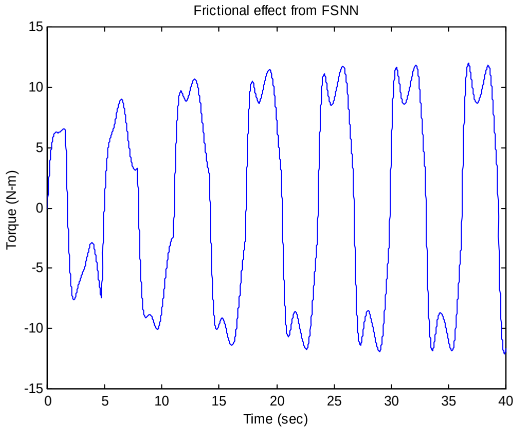

Figure 7. Friction model from 2-freq-based FSNN

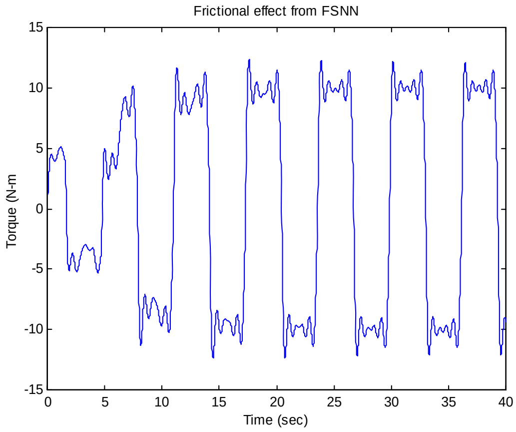

Figure 8. Friction model from 5-freq-based FSNN

7. Conclusions

Prior controlled system dynamics knowledge is unnecessary for the proposed controller. The FSNN required training period to gain system dynamics knowledge. After the network was properly learned, the simulation results indicated FSNN learning ability in system dynamics and prolonging the tracking error after the weight adaptation was terminated. The weights adaptation and stability of control system were confirmed using Lyapunov analysis. The tracking error was proved to be bounded and PD controller is necessary to ensure the overall stability. Since FSNN is approximating missing system dynamic of controlled system, still, there exists mismatching between best approximation from the network and exact system dynamic. As a result, the FSNN controller shows comparable stability to the regression adaptive controller with presence of system model mismatching. The FSNN controller can be employed without precise knowledge of controlled system dynamics. By its characteristic, the FSNN controller shows strong advantage for utilizing in nonlinear system control applications while acquire unknown system dynamics knowledge.

Acknowledgments

This research is financially supported by the National Science and Technology Development Agency (NSTDA), contract no. F-31-110-10-01

References

[1] Li, W. and Slotine, J., 1985. Parameter Estimation Strategies for Robotic Applications. ASME Winter annual Meeting, Boston.

[2] Astrom, K. and Wittenmark, 1989. Adaptive Control, Addison-Wesley.

[3] Craig, J., 1985. Adaptive Control of Mechanical Manipulators, Reading, MA., Addison-Wesley.

[4] Declercq, F., Dumortier, F., Keyser, R. and Cauwenberghe, A. V., 1997. Real-Time Control of a Robot using Neural Networks. Proceeding of the 1994 IEEE International Conference on Control Applications, vol. 2 part 2 (of 3), Apr., pp. 1061-1066.

[5] Song, Q., 1998. Robust Training Algorithm of Multilayered Neural Networks for Identification of Nonlinear Dynamic Systems", IEE Proceeding in Control theory and Application, vol.145, Jan., pp. 41-46.

[6] Narendra, K. S. and Parthasarathy, K., 1990. Identification and Control of Dynamical Systems using Neural Networks, IEEE Trans. Neural Network, 1(1), pp. 4-27.

[7] Cao, M., Wang, K. W., Fujii, Y. and Tobler, W. E., 2004. A Hybrid Neural Network Approach for the Development of Friction Component Dynamic Model, ASME Trans. Dynamic Systems, Measurement and Control, vol. 126, March, pp. 144-153.

[8] Shukla D. and Paul, F.W., 1995. Neural Networks with Orthogonal Activation Functions for Non-Linear Dynamic System Identification. Proceeding of the Artificial Neural Networks in Engineering Conference, vol. 5, Nov., pp. 517-524.

[9] Haque, I. and Schuller, J., 1999. Fourier Series Neural Network for Vehicle System Identification, Vehicle Design and Development, DE-vol. 101, Nov.

[10] Phimphilai, K., 2006. Two-Rigid Link Planar Robot Inverse Kinematics Approximation using Fourier Series Neural Network. Proceeding of the Tenth Annual National Symposium on Computational Science and Engineering conference, pp. 258-263, March 22-24, Chiang Mai, Thailand.

[11] Phimphilai, K., 2007. Model Free Nonlinear Adaptive Control of Robot Manipulators using Fourier Series Neural Network. Proceeding of the International Conference on Control, Instrumentation and Mechatronics Engineering, pp.108-115, May 28-29, Johor Bahru, Johor, Malaysia.