7-1991

Nonlinear System Identification Using Recurrent Networks

Hyungkeun Lee

Y. Park

Kishan Mehrotra
Syracuse University, mehrtra@syr.edu

Sanjay Ranka
Syracuse University

SU-CIS-91-20

Nonlinear System Identification Using Recurrent Networks

H. Lee, Y. Park, K. Mehrotra, C. Mohan, and S. Ranka

July 1991

School of Computer and Information Science
Syracuse University
Suite 4-116, Center for Science and Technology
Syracuse, New York 13244-4100

Nonlinear System Identification Using Recurrent Networks

Hyukjoon Lee, Yongseok Park [1]

Kishan Mehrotra, Chilukuri Mohan, Sanjay Ranka

Abstract

This paper presents empirical results on the application of neural networks to system identification and inverse system identification. Recurrent and Feedforward network models are used to build an emulator of a simple *nonlinear* gantry crane system, and for the inverse dynamics of the system. Recurrent networks were observed to perform slightly better than feedforward networks for these problems.

1 Introduction

Classical linear control provides robustness over a relatively small range of uncertainty. Adaptive control techniques have been developed for systems that must perform over large ranges of uncertainties due to large variations in parameter values, environmental conditions, and signal inputs. Neural networks are employed in adaptive control systems to increase the range of uncertainty that can be tolerated without sacrificing fast response, and without requiring human intervention.

Many successful works in the application of neural networks to various control problems have been reported, e.g., pole-balancing [1], robot arm control [2], truck backing-up [3], and inverse robot kinematics [5]. These systems have been successful due to (i) realization of fast decision making and control by parallel computation, (ii) fast adaptation to a large number of parameters as the convergence rate to a steady state is independent of the number of neurons in the network, (iii) adaptation to parameter variations over continuous and

[1]Currently at the Dept. of Computer Engg., Purdue University

discrete domains (iv) natural fault tolerance due to the distributed representation of information, (v) robustness to variations in parameters not modeled, due to the generalization properties of networks.

The process of incorporating a neural network into an adaptive control system consists of building neural networks that can recognize change in parameter values as well as estimate the inverse function of a system. Nonlinear functions applicable to control problems may be *serial order* or *time-varying,* and neural networks with recurrent connections have been known to have good performance for such problems in other fields. Nevertheless only a few applications of recurrent networks to practical problems have been reported (and almost none in control), e.g., in speech recognition [7], temporal pattern recognition [8, 9], and forecasting sunspot numbers [10].

In this paper we compare a recurrent network and a multilayer feedforward network (trained by error backpropagation [4]) to emulate and approximate the direct and inverse transfer function of a simple time-varying system, which can be represented as

$$\dot{X}(t) = A[X(t), U(t), t] \tag{1}$$

$$Y(t) = B[X(t), t] \tag{2}$$

The purpose of system identification is to find the optimal solution for A and B from the data of input $U(t)$ and output $Y(t)$.

2 Gantry Crane Problem and Recurrent Networks

We address the problem of controlling a gantry crane system. The gantry crane is used to move large parts and assemblies from one location to others on a factory floor. A cable is attached to the load to be moved which is then hoisted several feet in the air (See Figure 1). The control system is responsible for controlling the horizontal motion of the crane and load so that (i) the load is moved to a new site specified by given coordinates; (ii) load motion is well damped using position and velocity sensors; (iii) the closed loop bandwidth

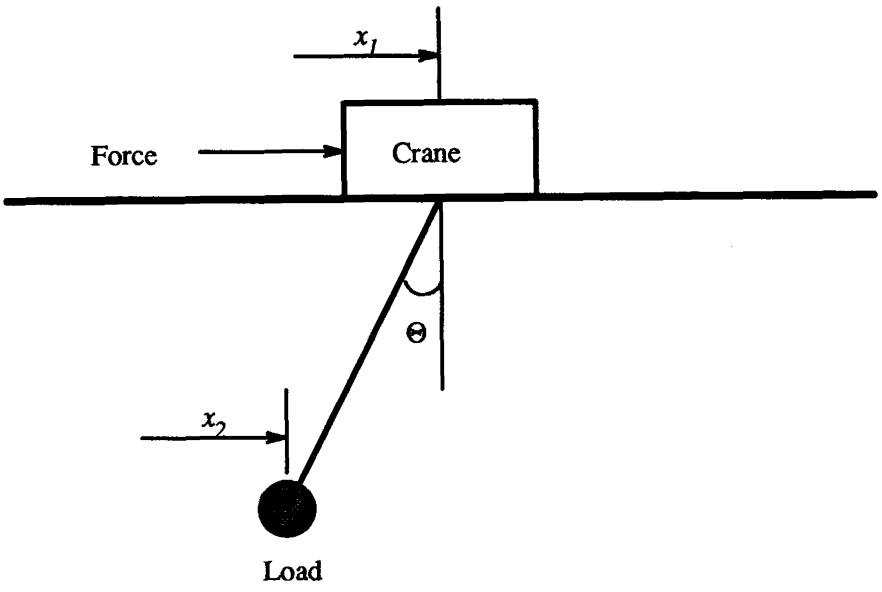

Figure 1: Gantry Crane System

is as large as possible while achieving reasonable crane (and load) stability; (iv) it can cope with variable load mass and cable length. This system can be represented by nonlinear differential equations as

$$F = M\ddot{X} + m[\ddot{X} + L(\ddot{\theta}\cos\theta - \dot{\theta}^2\sin\theta)] + c_1\dot{X} + c_2(\dot{X} + L\dot{\theta}\cos\theta) \tag{3}$$

$$m[L\ddot{\theta} + \ddot{X}\cos\theta] + mg\sin\theta + c_2[L\ddot{\theta} + \ddot{X}\cos\theta] = 0 \tag{4}$$

where X is the position of crane, θ is the angle of the cable, L is the cable length, M is the crane mass, m is the load mass, g is the gravity constant, c_1 is the viscous damping of the crane, c_2 is the viscous damping of the load, and F is the force applied to crane.

Networks with recurrent connections have been known to have important capabilities not found in feedforward networks [4]. Recurrent connections allow information about events occurring at arbitrary times in the past to be retained and used in current computations. Recurrent connections also allow networks to produce complex, time-varying outputs in response to simple static input, an important component in generating complex behaviors. For the gantry crane problem, performance of a feedforward network trained by the well-known error back-propagation algorithm was compared with the training al-

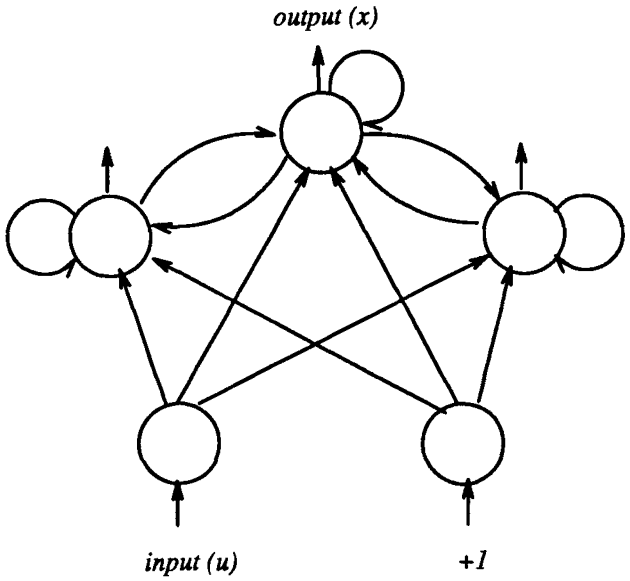

Figure 2: A fully recurrent network with one input, one output and two hidden units

gorithm described by Williams and Zipser [6] for a fully recurrent network in which any unit can receive external input (See Figure 2). These recurrent networks run continually in the sense that they sample their inputs on every update cycle, and any unit can receive training signals on any cycle.

3 Experiments and Results

By rewriting Eq. (3) and (4), we obtain expressions for the velocity of the crane $v_1(t+1)$ and the velocity of the load $v_2(t+1)$ at time $t+1$ as functions of voltage applied to the crane's motor $u(t)$, the velocities $v_1(t)$ and $v_2(t)$ at time t, and other parameters. We assume that the voltage and the velocities of crane and load can be measured while the system is driven. All the other parameters of the system are to be realized by the internal representations of neural networks. We generated 240 data points. The first 120 were used in training networks and the remaining 120 in testing. The value of voltage at each time point was generated randomly ranging from 0 to 200 and the two corresponding velocity measures were generated accordingly. In the case of test data, sinusoidal voltage ($sin(t/3)$) was also

Velocities	MSE in Training	MSE in Test	
		Random Input	Sinusoidal Input
Crane	0.000605	0.000493	0.000635
Load	0.000578	0.000296	0.001126

Table 1: Mean square errors in system identification by a recurrent net

used to compare the results in more realistic situations. The values of the constants i.e., M, m, L, c_1, c_2 were chosen arbitrarily. Then a recurrent network was trained to emulate the system behavior. Using the same data set, we also trained a feedforward network using the back-propagation algorithm. All values were normalized to range between 0.01 and 0.99 to be used by neural networks.

3.1 System Identification

A recurrent network with Williams and Zipser's training algorithm was implemented and run on the data set produced by the above description. A network with one input (for input voltage), two output (for crane's velocity or load's velocity) and five hidden units was trained. The number of hidden layers and nodes in the network were chosen after trying many possibilities. Performance results on training data were very good (but graphs portraying them have been omitted due to lack of space.) Results on test cases are shown in Figure 3a and 4a.

Next a feedforward network with 4 hidden nodes was trained and tested. The inputs to the network were $u(t)$, $v_1(t)$, $v_2(t)$, $v_1(t-1)$, and $v_2(t-1)$. The outputs from the network were $v_1(t+1)$ and $v_2(t+1)$. The test results are shown in Figure 3b and 4b. The graphs show that the feedforward networks did not perform as well as the recurrent networks in the test, although the former performed better than the latter in the training. The mean square errors for each case are given in Tables 1 and 2 for the case of recurrent and feedforward networks, respectively.

Velocities	MSE in Training	MSE in Test	
		Random Input	Sinusoidal Input
Crane	0.000364	0.000245	0.000966
Load	0.000560	0.000348	0.000784

Table 2: Mean square errors in system identification by a feedforward net

Neural Networks	MSE in Training	MSE in Test	
		Random Input	Sinusoidal Input
Recurrent	0.000494	0.000939	0.001419
Feedforward	0.000554	0.001168	0.002168

Table 3: Mean square errors in approximation of the inverse system

3.2 Inverse System Identification

For inverse system identification the previously generated data set was used to train and test performances of the networks. For the recurrent network with three hidden nodes the inputs were $v_1(t)$, $v_2(t)$ and the output was $u(t-1)$. The test results are depicted in Figure 3a and 4a (bottom). A feedforward network was applied to the same task with inputs $v_1(t)$, $v_2(t)$, $v_1(t-1)$, $v_2(t-1)$, $v_1(t-2)$ and $v_2(t-2)$, output $u(t-1)$, and two nodes in the hidden layer (so that the total number of weights in the feedforward network and that of the recurrent network are approximately equal). The graphs for test results are given in Figure 3b and 4b (bottom), and the mean square errors are given in Table 3. The recurrent network performed slightly better than the feedforward net.

4 Conclusions

We studied the applications of two kinds of neural networks in two important problems in control. The gantry crane system was chosen as the model for a simple non-linear time-varying system. The relevant data were artificially generated from the differential equations describing the system. Two major problems of interest — i.e., system identification and in-

verse system identification — were successfully solved using both recurrent and feedforward networks.

Our experimental results show that recurrent networks performed marginally better than feedforward networks, in terms of the mean square errors, for the system identification problem, as well as for the inverse system identification problem.

There are other advantages of recurrent networks over a feedforward networks. First, a recurrent network does not require *a priori* knowledge about the time structure of the system which is essential in using the feedforward network to determine the number of past data as the input to the network. Second, the past data need not be fed into a recurrent network while it must be explicitly given to the feedforward network. While the results from a single case study cannot be overly generalized, our work has shown that recurrent neural networks can be successfully used to solve practical control problems.

References

1. V.V. Tolat and B. Widrow, "An Adaptive Broom Balancer with Visual Inputs," *Proceedings of IEEE 1988 Intl. Conf. Neural Networks,* pp. 641-647.

2. Guez, Eilbert, and Kam, "Neural Network Architecture for Contol," *IEEE Control Systems Magazine,* pp. 22-25, 1988.

3. D.H. Nguyen and B. Widrow, "Neural Networks for Self-Learning Control Systems," *IEEE Control Systems Magazine,* pp. 18-23, 1990.

4. D.E. Rumelhart, G.E. Hinton, and R.J. Williams, "Learning Internal Representations by Error Propagation," *Parallel Distributed Processing,* Vol. 1, Chap. 8, Cambridge, MA., MIT Press, 1986.

5. G. Josin, D. Charney and D. White, "Robot Control Using Neural Networks," *Proceedings of IEEE 1988 Intl. Conf. Neural Networks,* pp. 625-631.

6. R.L. Williams and D. Zipser, "A Learning Algorithm for Continually Running Fully Recurrent Neural Networks," ICS Report 8805, Oct. 1988.

7. J.L. Jordan, "Attractor Dynamics and Parallelism in a Connectionist Sequential Machine," *Proc. Eighth Annual Conference of the Cognitive Science Society,* 1987.

8. M.C. Mozer, *A Focused Back-Propagation Algorithm for Temporal Pattern Recognition,* Technical Report, Dept. of Psychology and Computer Science, University of Toronto, 1988.

9. W.S. Stornetta, T. Hogg, and B.A. Huberman, "A dynamical Approach to Temporal Pattern Processing," *Proc. IEEE Conf. Neural Info. Processing Systems,* 1978.

10. M. Li, K. Mehrotra, C.K. Mohan, S. Ranka, "Forecasting Sunspot Numbers using Neural Networks," *Proc. 5th IEEE Intl. Symp. on Intelligent Control*, 1990.

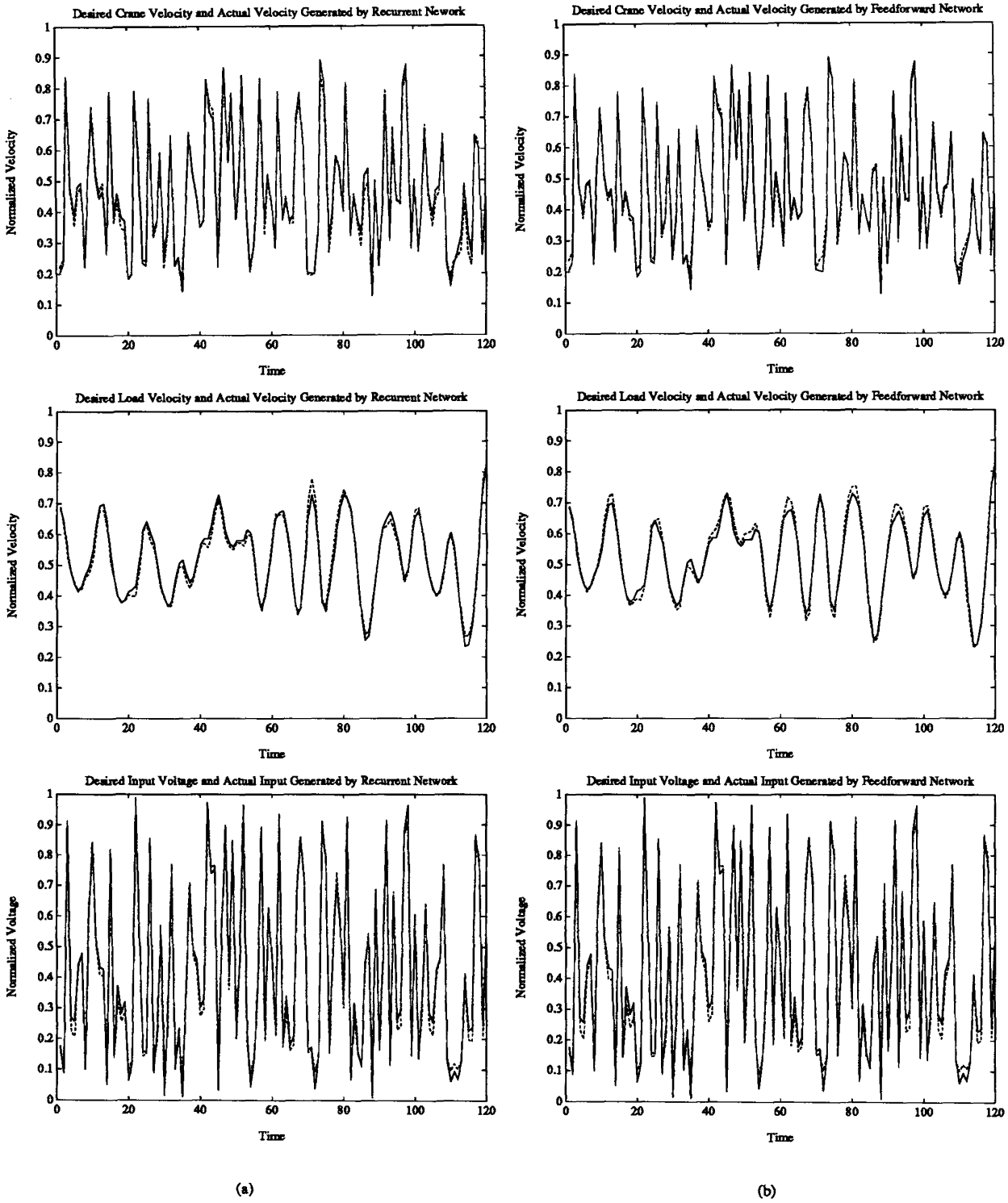

Figure 3: (a) Performance of the Recurrent Network, (b) Performance of the Feedforward Network with Sinusoidal Input Voltage; Solid lines and dashed lines represent the desired values and the actual values, respectively.

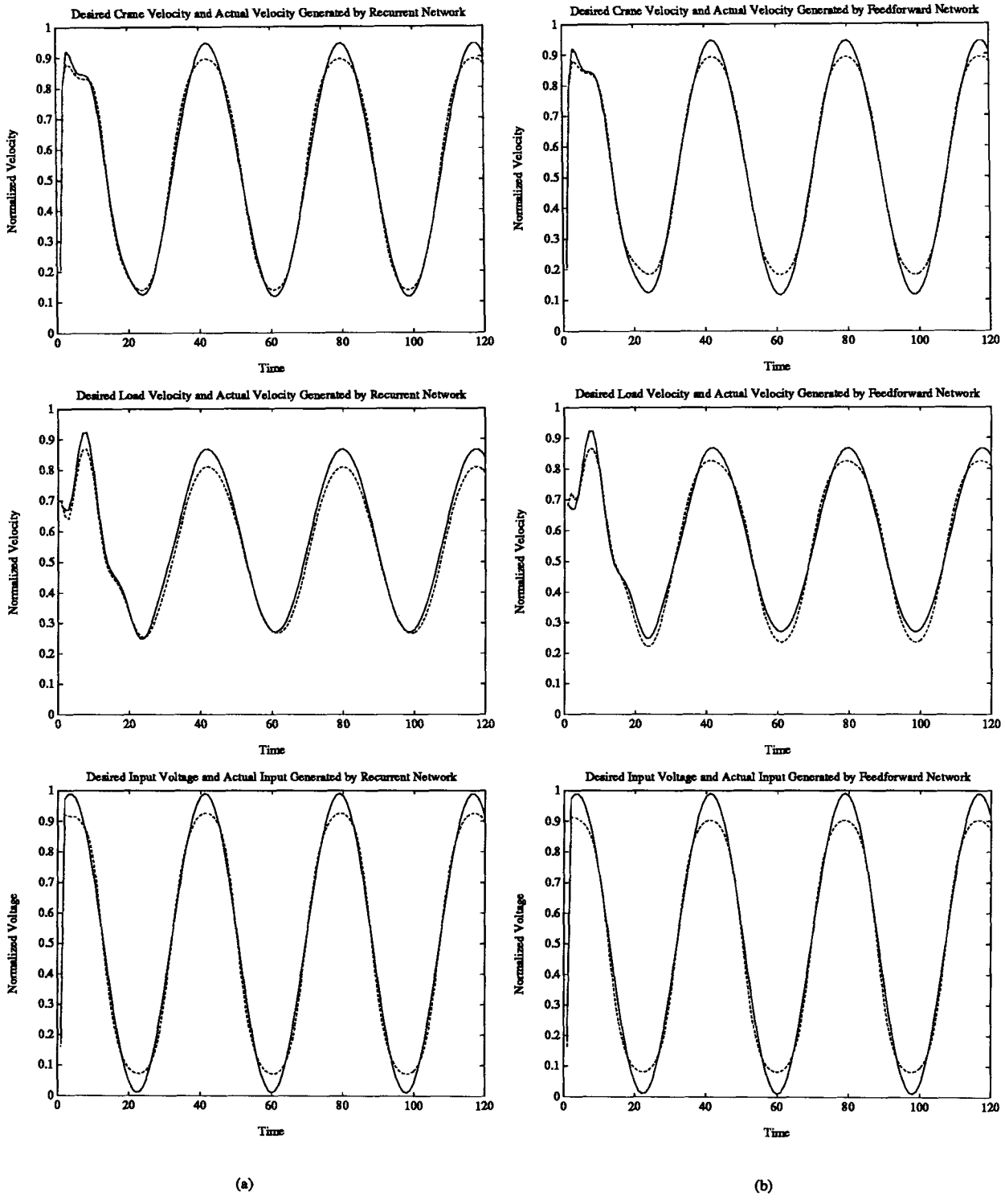

Figure 4: (a) Performance of the Recurrent Network, (b) Performance of the Feedforward Network with Sinusoidal Input Voltage; Solid lines and dashed lines represent the desired values and the actual values, respectively.