2008 American Control Conference
Westin Seattle Hotel, Seattle, Washington, USA
June 11-13, 2008

Improved Nonlinear Predictive Control Performance using Recurrent Neural Networks

Matthew Kuure-Kinsey and B. Wayne Bequette[1]
Isermann Department of Chemical & Biological Engineering, Rensselaer Polytechnic Institute, Troy, NY 12180-3590

***Abstract*— Recurrent neural networks are known to have better multi-step predictive capability compared to feedforward neural networks, with the disadvantage that they are more difficult to train. This paper develops a novel recurrent neural network architecture, the structure of which allows formulation as a time varying linear model. Based on a quadruple tank challenge problem, the proposed recurrent neural network is shown to have superior performance compared to a similarly designed feedforward neural network.**

1. Introduction

For complex nonlinear systems, analytical and empirical models are two common techniques available for quantifying underlying physical responses and behavior. Analytical, or first principle, models are based on known physical relationships and equations, which allows modeling results to have physical meaning. The benefit in basing models on these physical relationships is often outweighed by the time required to accurately determine all equations and associated parameters. The alternative is empirical modeling, which uses equations whose coefficients and parameters have no physical meaning. Because of this, the complexity and time required to generate the model is often significantly reduced.

Among empirical modeling, several primary approaches have emerged. Fuzzy models attempt to codify heuristic rule and "if-then-else" based systems. Fuzzy models have been used to model robotic systems (Vachkov and Fukuda, 2001), with a key limitation being the detailed amount of hierarchical knowledge required to build the model. If only the structure of the modeling equations is known, there are several approaches that can be used to determine the parameters. Genetic algorithms use a stochastic approach to parameter estimation (Yeh and Jang, 2006) and partial least squares regression uses an optimization driven approach (Qin, 1993).

This research focuses on the use of neural networks as the empirical model. No knowledge of model structure is needed, only input–output data is required. A series of interconnecting nodes and activation functions are used to model the system from the input–output data. Nodes and activation functions combine to form layers or neurons, which mimic the connectivity of the human brain. By connecting larger numbers of layers in series, neural networks have the ability to model any nonlinear system to an arbitrary degree of accuracy (Cotter, 1990). It is this "universal approximation" capability that makes neural networks such an appealing for technique modeling complex nonlinear systems.

Neural network architectures are classified by the direction of flow of information between layers. Neural networks where information flows in only in the forward direction, from inputs through layers to outputs are known as feedforward neural networks. The output of one layer becomes the input to the next layer. Neural networks where information flows both forward from input through layers to output and backwards between layers are called recurrent neural networks. The purpose of the backward, or recurrent, connection between layers is to introduce internal dynamics to the model, allowing the recurrent neural network to model dynamic input–output relationships (Chao-Chee and Lee, 1995). In contrast, feedforward neural networks have no internal dynamics, resulting in an overall static input–output relationship.

Model predictive control is an advanced control strategy that uses a model, such as a neural network, to predict future system behavior when calculating control actions. Both recurrent and feedforward neural networks have been used in a model predictive control framework. Medinelli and Rojas (2007) use a feedforward neural network to model and control a solar energy water heater. Pappa *et al.* (2005) use a recurrent neural network based strategy to control a double-pipe heat exchanger, with improved performance compared to an existing PID control strategy.

There are two primary contributions in this paper. A novel recurrent neural network architecture is developed that yields an analytical solution under unconstrained model predictive control. This recurrent neural network model is used with a similar feedforward model to determine the effect of multi step ahead predictive ability on model predictive control performance. The paper is structured as follows: section 2 discusses the neural network architectures and training styles, section 3 details the model predictive control strategy, section 4 details the example system studied with training and validation results, section 5 compares the accuracy of multi step ahead

[1] bequette@rpi.edu, (518) 276-6683 (voice), (518) 276-4030 (fax)

predictions for the neural network architectures and section 6 compares control performance.

2. Neural Network Architecture and Training

Feedforward and recurrent neural networks belong to a class of neural networks known as multi layer perceptrons (MLP's), which consist of input, hidden and output layers.

2.1 Feedforward Architecture

To use neural networks in control applications, it is critical that the architecture is capable of representing the dynamic models found in complex nonlinear systems. Feedforward neural networks, which are inherently static in nature, require the addition of external feedback to model dynamic behavior. Kuure-Kinsey *et al.* (2006) develop a feedforward neural network capable of representing dynamic behavior, with the network architecture shown in figure 1.

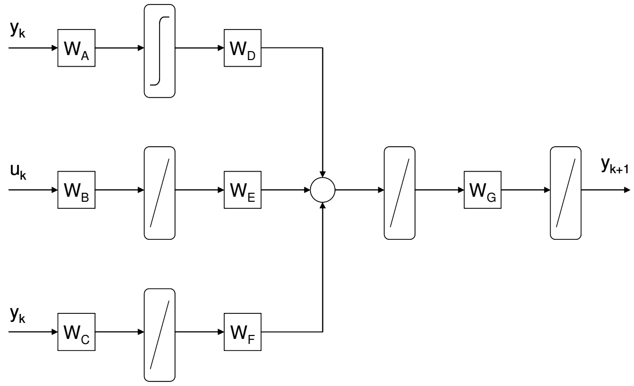

Figure 1: Feedforward neural network architecture capable of representing dynamic behavior.

The external feedback to the feedforward neural network is denoted in figure 1 as y_k. It is important to note that only the external feedback passes through a nonlinear activation function. This results in the nonlinear state space model given in (1).

$$\hat{x}_{k+1} = W_F W_C \hat{x}_k + W_E W_B u_k + W_D \tanh(W_A y_k)$$
$$\hat{y}_{k+1} = W_G \hat{x}_{k+1} \qquad (1)$$

To train the feedforward architecture, external feedback is supplied at each time step. Because the feedback signal in figure 1 is externally supplied at each time step, the feedforward neural network is trained to predict dynamic behavior only at the next time step in the future. This is known as one step ahead prediction.

2.2 Recurrent Architecture

Feedforward neural networks require external feedback to model dynamic systems. One benefit of recurrent neural networks is that the internal feedback present is inherently capable of representing dynamic behavior. There are a number of recurrent neural network architectures in the literature, but to make any comparisons to the feedforward architecture in section 2.1 as equivalent and accurate as possible, it makes sense to develop an architecture similar to the feedforward architecture. The developed recurrent architecture is shown in figure 2.

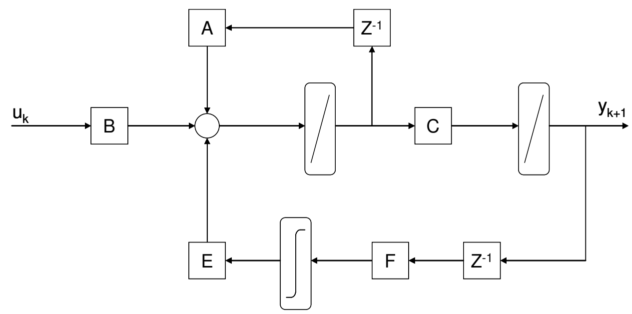

Figure 2: Recurrent neural network architecture

There are two recurrent layer connections in figure 2. The recurrent connection in layer 1 from the output y_{k+1} back to the input to layer 2 mimics the external feedback supplied to the feedforward neural network in figure 1. The recurrent connection from the output of layer 2 back to the input to layer 2 adds a state space characteristic to the architecture. The resulting input-output relationship for the recurrent architecture is given in (2).

$$\hat{x}_{k+1} = A\hat{x}_k + Bu_k + E\tanh(F\hat{y}_k)$$
$$\hat{y}_{k+1} = C\hat{x}_{k+1} \qquad (2)$$

The important thing to note about (2) is that the structure of the nonlinear state space equation is the exact same as the feedforward architecture in (1). The only difference between the nonlinear state space models in (1) and (2) is the feedback signal. The feedback signal y_k in (1) comes from an external measurement while $\hat{y}_k$ in (2) comes from internal feedback. Because the feedback signal in figure 2 is internally supplied at each time step, the recurrent neural network is trained to predict dynamic behavior beyond the next time step in the future. This is known as multi step ahead prediction.

2.3 Training Styles

While the dynamic model structure is equivalent between the feedforward and recurrent architectures in (1) and (2) respectively, that does not mean that the training is the same. Feedforward neural networks require an external feedback source to model dynamic systems whereas recurrent neural networks use internal feedback. This difference in feedback source is illustrated in figure 3.

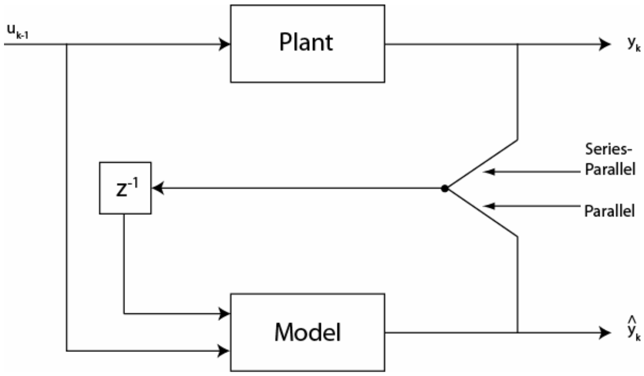

Figure 3: Difference in source of feedback required by neural networks to model dynamic systems

The reason the difference in figure 3 is important is the intended use of the neural network model. If only used for one step ahead predictions, then neither neural network has an advantage. In model predictive control however, the neural network is required to predict over a horizon greater than one, which results in a multi step ahead prediction with no source of external feedback. With only internal feedback available, the nonlinear state space model representing the feedforward neural network becomes:

$$\begin{aligned} \hat{x}_{k+1} &= W_F W_C \hat{x}_k + W_E W_B u_k + W_D \tanh\left(W_A \hat{y}_k\right) \\ \hat{y}_{k+1} &= W_G \hat{x}_{k+1} \end{aligned} \tag{3}$$

It is important to note that under model predictive control, the feedforward neural network model in (3) has a model predicted feedback source, making the nonlinear state space models in (2) and (3) identical in both structure and signals.

Su *et al.* (1992) study the predictive capabilities of feedforward and recurrent neural networks, trained using the series-parallel and parallel connections in figure 3, and find that recurrent neural networks provide superior predictive capabilities for multi step ahead predictions. Su *et al.* (1992) also theorize that the improved multi step ahead predictions in recurrent neural networks leads to better performance in model predictive control, though no results are shown to support the claim. With feedforward and recurrent neural networks developed that have the same dynamic model structure, it is now possible to determine if the source of feedback during training and subsequent multi step ahead predictive abilities have an effect on model predictive control performance.

3. Model Predictive Control

Model predictive control, or receding horizon control, is an advanced control technique that takes advantage of a model's inherent ability to predict system behavior into the future. At each time step, an optimization problem is formulated and solved. The objective function is to minimize control action over *p* time steps, where *p* is known as the prediction horizon. The decision variables are *m* control moves, where *m* is the control horizon. Only the first control move is applied to the system, the model is updated, and the entire process is repeated at the next time step. The objective function used in this work is a sum of squares, given in (4).

$$\begin{aligned} \Phi = &\sum_{i=1}^{p} \left(r_{k+i} - \hat{y}_{k+i|k}\right)^T W_y \left(r_{k+i} - \hat{y}_{k+i|k}\right) \\ &+ \sum_{i=0}^{m-1} \Delta u_{k+i}^T W_u \Delta u_{k+i} \end{aligned} \tag{4}$$

Where r_{k+i} is the setpoint, Δu_{k+i} is the control action and $\hat{y}_{k+i}$ is the output prediction. A number of different types of models are used in a model predictive control framework, with this work using the neural network models developed in section 2. The model structures in (2) and (3) are equivalent under model predictive control and known as a nonlinear state space model with a feedback path nonlinearity. Having both architectures result in this particular nonlinear state space model is important, as Kuure-Kinsey *et al.* (2006) show that a computationally efficient model predictive control formulation is possible. A succinct review of the formulation follows, for a more detailed discussion of the formulation and verification against nonlinear optimization of the nonlinear state space model, the reader is referred to Kuure-Kinsey *et al.* (2006).

The primary benefit to the structure of the nonlinear state space model in (2) and (3) is that the nonlinearity is static and only involves the output term. This allows for the static nonlinearity to be mapped across the prediction horizon into a time varying linear term.

$$\begin{aligned} \hat{x}_{k+1} &= A_k \hat{x}_k + B u_k + g_k \\ \hat{y}_k &= C \hat{x}_k \end{aligned} \tag{5}$$

Where the terms A_k and g_k are defined in (6)-(7).

$$A_k = A + E\left(1 - \tanh\left(Fr_k\right)^2\right)FC \tag{6}$$

$$g_k = E\tanh\left(Fr_k\right) - E\left(1 - \tanh\left(Fr_k\right)^2\right)Fr_k \tag{7}$$

The nonlinear state space model in (2) and (3) has now been cast into a linear time varying model. The linear time varying model in (5) does not perfectly represent the system at all times, as there is always a degree of parameter uncertainty and measurement noise. To account for this plant-model mismatch, there are two common state estimation approaches employed in model predictive control: an additive output disturbance assumption and a step input disturbance assumption. Muske and Badgwell (2002) show that the step input disturbance model, which estimates an augmented disturbance state using a Kalman filter, provides better rejection of step input disturbances, and is therefore the state estimation approach used in this formulation. The linear time-varying model in (5) is augmented with a step input disturbance in (8).

$$\underbrace{\begin{bmatrix} \hat{x}_k \\ \hat{d}_k \end{bmatrix}}_{\hat{x}^a_{k|k-1}} = \underbrace{\begin{bmatrix} A_{k-1} & \Gamma^d \\ 0 & I \end{bmatrix}}_{A^a_k} \underbrace{\begin{bmatrix} \hat{x}_{k-1} \\ \hat{d}_{k-1} \end{bmatrix}}_{\hat{x}^a_{k-1|k-1}} + \underbrace{\begin{bmatrix} B \\ 0 \end{bmatrix}}_{B^a} u_k + \underbrace{\begin{bmatrix} I \\ 0 \end{bmatrix}}_{G^a} g_k$$
$$\hat{y}_k = \underbrace{\begin{bmatrix} C & 0 \end{bmatrix}}_{C^a} \underbrace{\begin{bmatrix} \hat{x}_k \\ \hat{d}_k \end{bmatrix}}_{\hat{x}^a_{k|k}} \tag{8}$$

The augmented model in (8) is based on information available through timestep k-1. Once a measurement is taken at the current timestep k, the augmented model is updated through the predictor and corrector equations in (9) and (10).

$$\hat{x}^a_{k|k-1} = A^a_k \hat{x}^a_{k-1|k-1} + B^a u_k + G^a g_k \tag{9}$$
$$\hat{x}^a_{k|k} = \hat{x}^a_{k|k-1} + L\left(y_k - C^a \hat{x}^a_{k|k-1}\right) \tag{10}$$

In (10), L is the steady-state Kalman gain, calculated by solving the steady-state Ricatti equation. The augmented model in (8) is then used as the model in the objective function in (4), which results in an analytical solution to the unconstrained problem. For constrained systems, the objective function (4) and augmented model (8) yield a quadratic programming problem, for which efficient computational routines exist.

4. Quadruple Tank Details

To compare the performance under model predictive control of the feedforward and recurrent neural networks, the quadruple tank is chosen as an example system for the presence of nonminimum phase behavior, a well known control challenge. The quadruple tank is a two input, two output system, described by the nonlinear differential equations in (11)-(15), with the parameters and nominal steady state values given in table 1.

$$\frac{dh_1}{dt} = \frac{-a_1}{A_1}\sqrt{2gh_1} + \frac{a_3}{A_1}\sqrt{2gh_3} + \frac{\gamma_1 k_1}{A_1} v_1 \tag{11}$$
$$\frac{dh_2}{dt} = \frac{-a_2}{A_2}\sqrt{2gh_2} + \frac{a_4}{A_2}\sqrt{2gh_4} + \frac{\gamma_2 k_2}{A_2} v_2 \tag{12}$$
$$\frac{dh_3}{dt} = \frac{-a_3}{A_3}\sqrt{2gh_3} + \frac{(1-\gamma_2)k_2}{A_3} v_2 \tag{13}$$
$$\frac{dh_4}{dt} = \frac{-a_4}{A_4}\sqrt{2gh_4} + \frac{(1-\gamma_1)k_1}{A_4} v_1 \tag{14}$$
$$y = \begin{bmatrix} k_c & 0 & 0 & 0 \\ 0 & k_c & 0 & 0 \end{bmatrix} \begin{bmatrix} h_1 \\ h_2 \end{bmatrix} \tag{15}$$

The manipulated inputs are control voltages to pumps, v_1 and v_2, that regulate the flow of liquid to the system. The measured outputs are the heights of tanks 1 and 2. The γ_1 and γ_2 terms represent the fraction of flow from pumps 1 and 2 that go to the lower tanks versus the upper tanks. For additional information about the quadruple tank system, see Johansson (2000).

Table 1: Parameters and steady state values for the quadruple tank

Parameter	Value	Parameter	Value
A_1	28 cm^2	A_3	28 cm^2
A_2	32 cm^2	A_4	32 cm^2
$h_{1,ss}$	12.6 cm	$h_{3,ss}$	4.8 cm
$h_{2,ss}$	13.0 cm	$h_{4,ss}$	4.9 cm
k_c	0.5 V / cm	g	981 cm / s^2
a_1	0.071 cm	a_3	0.071 cm
a_2	0.057 cm	a_4	0.057 cm
$v_{1,ss}$	3.15 V	k_1	3.14 cm^3 / Vs
$v_{2,ss}$	3.15 V	k_2	3.29 cm^3 / Vs
γ_1	0.43	γ_2	0.34

To ensure a valid comparison of multi step ahead predictions and performance under model predictive control, the same quadruple tank data is used to train and verify the feedforward and recurrent neural networks. Dynamic step test data is generated from the model in (11)-(15) and parameters in table 1, with training and verification results for the feedforward and recurrent neural networks given in figures 4 and 5.

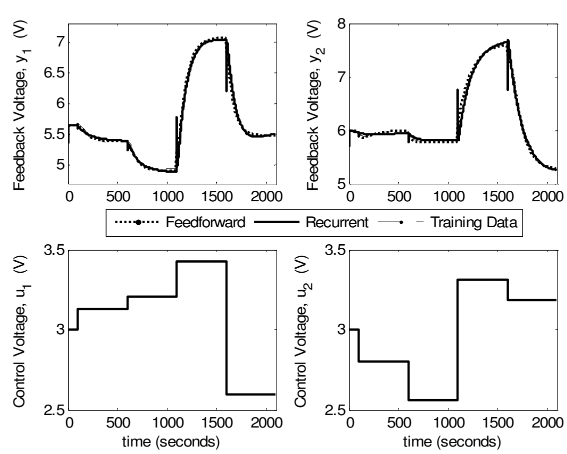

Figure 4: Training results for the feedforward and recurrent neural network models.

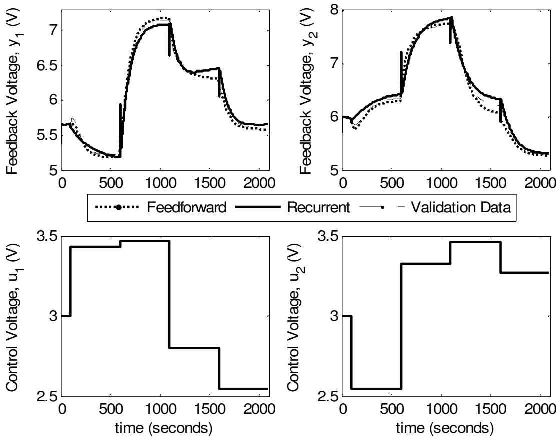

Figure 5: Training results for the feedforward and recurrent neural network models.

Note that the time intervals between step changes in the training and validation input sequences are selected based on the dominant time constant of the quadruple tank. The results in figures 4 and 5 demonstrate that both the feedforward and recurrent neural networks are able to successfully model the quadruple tank.

5. Multi Step Ahead Predictions

In section 4, feedforward and recurrent neural networks are trained and validated for the quadruple tank system. To confirm the multi step ahead predictive ability of recurrent neural networks demonstrated in Su *et al.* (1992), each model is propagated through a prediction horizon of increasing length. This is done by propagating each model through a prediction horizon of increasing length. In both neural networks, the feedback supplied to the network is from the neural network model, not from external measurements. This forces both neural networks to operate in series-parallel mode, similar to the prediction horizon calculation in model predictive control.

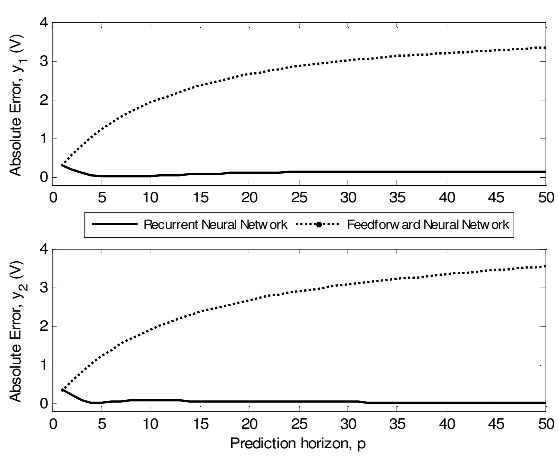

Figure 6: Comparison of mean absolute deviation for feedforward and recurrent neural networks as a function of prediction horizon for the quadruple tank

The quadruple tank is tested using a step input sequence from the validation data set. For a change in v_1 from 3.43 to 3.47 and a change in v_2 from 2.55 to 3.33, the mean absolute deviation (MAD) from the underlying differential equation based model is calculated for the two neural networks. For a prediction horizon range of 1 to 50, the comparisons for the two outputs of the quadruple tank are shown in figure 6.

The results in figure 6 confirm, for non-minimum phase operation of the quadruple tank, the observation of Su *et al.* (1992) that recurrent neural networks have better predictive ability than feedforward neural networks. For small prediction horizons, the one-step-ahead trained feedforward neural network gives adequate performance, with comparable performance to the multi step ahead trained recurrent neural network. Once the prediction horizon increases, the recurrent neural network provides increasingly better predictions compared to the feedforward neural network. This result illustrates the effect of propagating the internal model predictions on the multi step ahead performance of the feedforward network and motivates the evaluation of model predictive control performance for the two neural network architectures.

6. Control Performance

The performance under model predictive control of the two neural network architectures is tested using the control formulation presented in section 3. To make the comparison valid, all control parameters are held constant between the two neural networks. The control performance is tested for setpoint regulation of the quadruple tank, with figure 7 showing the performance comparison for a specific setpoint profile.

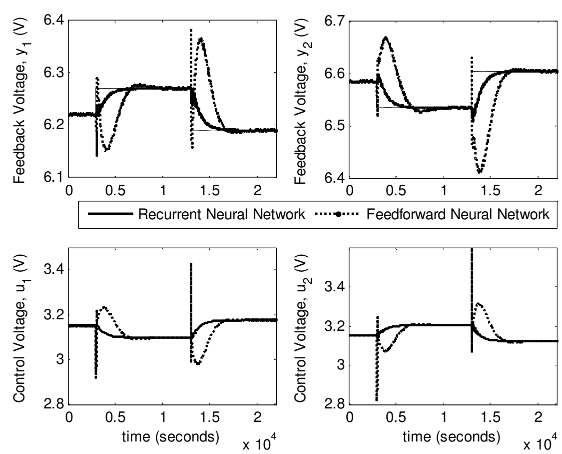

Figure 7: Setpoint regulation results comparing feedforward and recurrent neural networks. p = 20, m = 3, Δt = 1 sec, Q/R = 0.01

The results in figure 7 show that the series-parallel trained recurrent neural network has superior performance compared to the series trained feedforward neural network.

The mean absolute deviation of 0.0065 V for the recurrent neural network is 22% of the 0.0289 V for the feedforward neural network. This result is for a specific set of control parameters. To determine the effect of prediction and control horizon, two critical parameters in model predictive control, the setpoint regulation problem is run over a range of prediction and control horizons. The effect of different prediction and control horizons on the MAD values is shown in figure 8.

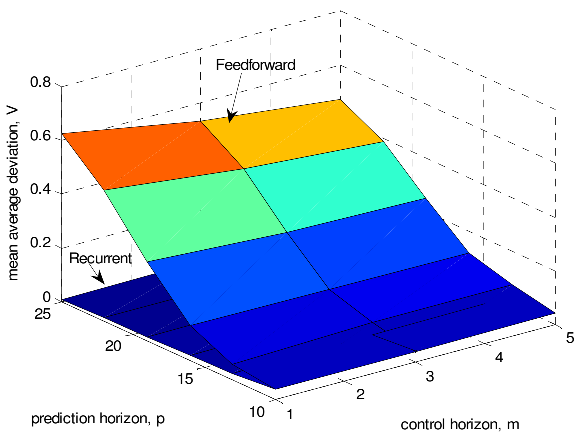

Figure 8: Comparison of mean absolute deviation between recurrent and feedforward neural networks for a range of prediction and control horizons. $\Delta t = 1$ sec, $Q/R = 0.01$

The MAD values are stronger functions of the prediction horizon and the MAD values for the recurrent neural network are uniformly lower than for the feedforward neural network. The dependence on prediction horizon matches with the training of the respective neural networks. The feedforward neural network, trained for one step ahead predictions, show progressively worse behavior as prediction horizon increases. This is due to the increasing multi step ahead prediction required for the model predictive control calculation. The recurrent neural network, trained exactly for the type of multi step ahead predictions required for model predictive control, has a consistent performance not dependant on prediction horizon. This demonstrates the performance difference between the two neural networks when used in model predictive control.

7. Summary

This research develops a novel recurrent neural network trained using a series-parallel approach to output feedback. The network has the same dynamic model structure as a previously developed feedforward neural network trained using a series approach to output feedback. The recurrent neural network represents training for multi step ahead predictions and the feedforward neural network represents one step ahead predictions. The neural networks are used in model predictive control to compare performance, with the series-parallel trained recurrent neural network providing superior performance for a nonminimum phase multivariable quadruple tank system.

Acknowledgements

The authors gratefully acknowledge the National Science Foundation (BES-0411693 and DGE-0504361) for partial financial support of this research, and Hyunjin Lee for assistance with the generation of input sequences for training and validation of the neural networks.

Literature Cited

1. Chao-Chee, K., Lee, K., "Diagonal recurrent neural networks for dynamic systems control", IEEE Trans Neural Networks, 6 144-156 (1995)
2. Cotter, N. "The Stone-Weierstrass theorem and its application to neural networks", IEEE Trans Neural Networks, 1 290-295 (1990)
3. Johansson, K., "Quadruple tank process: a multivariable laboratory process with an adjustable zero", IEEE Trans Control Syst Technol, 8 3 456-465 (2000)
4. Kuure-Kinsey, M., Cutright, R., Bequette, B.W., "Computationally efficient neural predictive control based on a feedforward architecture", Ind Eng Chem Res, 45 25 8575-8582 (2006)
5. Medinelli, L., Rojas, R., "Modeling and identification of solar energy water heating system incorporating nonlinearities", Sol Energy, 81 5 570-580 (2007)
6. Muske, K., Badgwell, T., "Disturbance modeling for offset free linear model predictive control", J Process Control, 12 5 617-632 (2002)
7. Pappa, N., Kaliraj, G., Shanmugan, J., "Real-time implementation of model predictive neural controller for heat exchanger", Control Intell Syst, 33 3 184-189 (2005)
8. Su, H., Werbos, P., McAvoy, T., "Long term predictions of chemical processes using recurrent neural networks: a parallel training approach", Ind Eng Chem Res, 31 1338-1352 (1992)
9. Qin, J., "Partial least squares regression for recursive system identification" Proc IEEE Conf Decis Control, 3 2617-2621 (1993)
10. Vachkov, G., Fukuda, T., "Structured learning and decomposition of fuzzy models for robotic control applications" J Intell Rob Syst Theor Appl 32 1 1-21 (2001)
11. Yeh, C., Jang, S., "Chemical process model parameter estimation using an information guided genetic algorithm" J Chem Eng Japan 39 2 197-210 (2006)