Dynamic Rectification of Data via Recurrent Neural Nets and the Extended Kalman Filter

Thomas W. Karjala and David M. Himmelblau
Dept. of Chemical Engineering, University of Texas, Austin, TX 78712

The presence of autocorrelated measurement errors and/or measurement bias in process measurements poses serious problems in the rectification of data taken from dynamic processes. The proposed procedure to resolve these problems involves the use of recurrent neural networks (RNN) and the extended Kalman filter (EKF). By interpreting RNNs within a nonlinear state-space context, a state-augmented EKF can be used to optimally estimate both the states of the RNNs and noise and bias models. RNN models can be identified off-line and utilized for data rectification within the extended Kalman filter in process environments in which badly autocorrelated measurement errors exist in the data. The same technique is also used to estimate measurement bias present in both process input and output variables. This approach has the advantage that models developed from "first principles" are not required and that rectification can be performed solely on the basis of the contaminated dynamic data.

Introduction

Process measurements are often contaminated in the sense that they include random noise and biases arising from measurement irreproducibility, instrument degradation, human error, as well as unmeasured factors. Many techniques have been proposed to adjust inaccurate and inconsistent plant data to conform with process models so as to yield improved estimates of the values of the measured (or unmeasured) variables. In the literature, the term *data reconciliation* has been commonly used to refer to the reduction of additive Gaussian noise from process measurements by adjusting the measurements to agree with a first-principles model of the process. For our work we prefer to use the more general term *data rectification*, meaning here to "correct by making the necessary adjustments" so that rectification includes reconciliation as a specific rectification technique. Rectification includes the removal/reduction of not only Gaussian noise but other types of contaminants in the data such as gross errors and biases, and is not limited to the use of a first-principles, parametric model as in conventional data reconciliation. Data rectification has traditionally been performed in two steps. First, nonrandom measurement errors must be detected and their presence eliminated. Then, the presumed remaining random-measurement noise is reduced, typically through some sort of filtering or data smoothing. Traditional techniques demand an accurate process model in order to yield unbiased results, and have generally only been applied to measurements containing uncorrelated errors.

Rectification of simulated data has the advantage over rectification of real plant data in that the values of the variables resulting from rectification can be compared with their corresponding true values that served as the basis of the simulation. By repeated use of rectification strategies on simulated data from multiple processes, it should be possible to identify sound strategies and build up trust that they may be effective for real data under various realistic assumptions about noise, gross errors, bias, and so on.

Our earlier work (Karjala et al., 1992; Karjala and Himmelblau, 1994a) involving recurrent neural networks (RNN) used only a prediction mechanism for nonparametric rectification of data taken from a dynamic process. Our more recent work (Karjala and Himmelblau, 1994b) included a new approach to data rectification by combining recurrent networks with the prediction/correction framework of Kalman filtering. What makes the work described here novel and unique is that we now examine data containing autocorrelated noise where our previous work and essentially all the literature assumes that data contains only Gaussian noise, a situation that can be treated by recurrent neural networks (RNN) alone. We also treat time-varying measurement bias. By interpreting an RNN as a state-space model we can carry

out rectification of data from dynamic processes without recourse to a "perfect model" (as is so often the case in the literature) for instances in which an RNN alone will fail.

We begin by reviewing the most commonly encountered types of RNNs, their state-space formulation, and methods for identifying them using prediction error techniques. Data rectification using RNNs is discussed next along with the problems encountered when RNNs alone are applied to measurements containing autocorrelated measurement noise. An example of a gravity-flow tank/pipeline system illustrates the identification procedure. Extended Kalman filtering is then used along with state augmentation techniques to treat data containing autocorrelated noise from two examples (1) a gravity-flow tank/pipeline, and (2) a packed distillation column. Finally, we demonstrate how these same ideas allow estimation and removal of measurement bias from process input and output variables. The examples demonstrate the benefits of combining Kalman filtering with neural nets for rectification of data taken from a dynamic process.

Data Rectification and System Identification Using RNNs

Recurrent neural networks (RNNs) have become an increasingly popular nonparametric empirical modeling tool for nonlinear system identification. RNNs are similar in structure to the standard feedforward artificial neural network architecture that comprises layers of nodes connected by weighted feedforward connections, but RNNs also contain time-delayed feedback, or recurrent, connections.

Two variations of RNNs are commonly found. The first are commonly called internally recurrent networks (IRNs), and are characterized by time-delayed feedback connections between the hidden nodes. The remainder of the network is a standard feedforward architecture. The second type are called externally recurrent networks (ERN), networks that contain time-delayed connections from the nodes of the output layer to the nodes of the hidden layer. One can also envision a hybrid recurrent network that contains both types of recurrent connections that can be described as an internal–external recurrent network (IERN). Such a network is pictured in Figure 1. Keep in mind that the nodes in the network represent basis functions.

Simulation studies (MacMurray, 1993), both published and unpublished, have indicated that no clear advantage exists in using IRN vs. ERN or even IERN for dynamic modeling. Both IRN and ERN modeling techniques seem to work about equally well. Our experience has indicated that internally recurrent networks hold a slight advantage for most modeling problems, with externally recurrent networks coming a close second. Each method has its respective advantages and disadvantages for a particular modeling problem.

The simplest approach to rectification of process measurements from dynamic process using RNNs, and the approach originally explored by Karjala et al. (1992) and Karjala and Himmelblau (1994a), is to employ a one-step-ahead prediction model. The idea in using a neural network for rectification is to find a nonlinear mapping of the process measurements to some hypothesized "true" values of the measurements. If a recurrrent network is used as a one-step-ahead predictive filter for the process measurements, the rectified values of the process measurements are related to the measurements themselves by

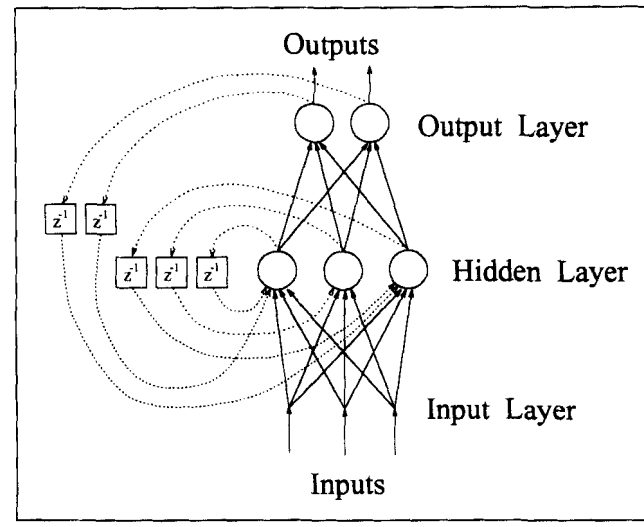

Figure 1. Internally/externally recurrent neural network (IERN).

Circles represent computation nodes, lines represent weighted connections, and z^{-1} indicates time delay. For clarity, not all of the recurrent connections are shown.

$$\hat{\boldsymbol{m}}_t = \boldsymbol{g}(\boldsymbol{m}_{t-1}, \boldsymbol{m}_{t-2}, \boldsymbol{m}_{t-3}, \ldots, \boldsymbol{m}_0), \tag{1}$$

where the vector $\hat{\boldsymbol{m}}_t$ is the best estimate of the process measurement vector $\boldsymbol{m}_t$ at time t, and $\boldsymbol{g}(\cdot)$ is the aforementioned nonlinear mapping. Although IRN models depend explicitly only on $\boldsymbol{m}_{t-1}$ as inputs, they implicitly depend on all previous measurement vectors through the autoregressive nature of the feedback connections within the network. However, measurement vectors from the distant past have little effect on the current predictions of the network.

Note that in the model shown by Eq. 1 the current estimate $\hat{\boldsymbol{m}}_t$ is not calculated from the current measurement $\boldsymbol{m}_t$. You cannot use $\boldsymbol{m}_t$ as an independent variable in the nonparametric model $\boldsymbol{g}$, because system identification would then yield the trivial identity mapping of $\hat{\boldsymbol{m}}_t = \boldsymbol{m}_t$. Because the noise in each measurement is assumed uncorrelated with the noise in previous measurements, it is possible to identify a system that describes the evolution of the measurement vectors in time using a prediction error model. Note also that because the model input $\boldsymbol{m}_{t-1}$ is not deterministic and contains measurement noise, the parameter estimates for the RNN model will be biased. How much bias exists is problem dependent, but in our simulations we have obtained good results for nonlinear processes in which the process measurements are corrupted by Gaussian measurement errors (Karjala et al., 1992; Karjala and Himmelblau, 1994a).

State-space representation

While the majority of neural-network researchers speak in terms such as training patterns, test sets, connection weights, and hidden layers, it is useful to present artificial neural network models in terms of nonlinear system identification. If

we allow the vectors $\boldsymbol{u}_t$, $\boldsymbol{x}_t$, and $\boldsymbol{y}_t$ to denote the vector outputs of the input, hidden, and output nodes, respectively, at time t, we can formulate an IRN network as a nonlinear discrete-time model

$$\boldsymbol{x}_{t+1} = \boldsymbol{\varphi}_h(\boldsymbol{A}\boldsymbol{x}_t + \boldsymbol{B}\boldsymbol{u}_t + \boldsymbol{b}_h) \tag{2}$$

$$\boldsymbol{y}_{t+1} = \boldsymbol{\varphi}_o(\boldsymbol{C}\boldsymbol{x}_{t+1} + \boldsymbol{b}_o), \tag{3}$$

where $\boldsymbol{\varphi}_h(\cdot)$ and $\boldsymbol{\varphi}_o(\cdot)$ are the vector valued functions corresponding to the activation functions in the hidden and output layers, respectively. The scalar elements of the Gaussian activation function for each hidden node are defined as

$$\varphi_i(v_i) = \exp\left(\frac{-v_i^2}{2}\right), \tag{4}$$

where v_i is the total input to each node. Usually all the activation functions in one layer are made identical for simplicity. Linear activation functions are typically used in the output layer. The matrices $\boldsymbol{A}$, $\boldsymbol{B}$, and $\boldsymbol{C}$ are the matrices of connection weights for the hidden-to-hidden recurrent connections, input to hidden, and hidden-to-output connections, respectively, and the vectors $\boldsymbol{b}_h$ and $\boldsymbol{b}_o$ are the bias vectors for the hidden and output layers. The common IRN architecture does not normally include direct connections from the input to the output layers, but these connections could be easily added. By posing the IRN model in the preceding form, we see that this type of recurrent neural network is a nonlinear extension of the standard linear state-space model in which the hidden layer outputs, $\boldsymbol{x}_t$, are the model states.

In a similar fashion we can write nonlinear state-space equations to represent the ERN. Whereas in the IRN model the states are the outputs of the hidden nodes, in the ERN model the states are the outputs of the nodes in the output layer so that the state-space equations are

$$\boldsymbol{x}_{t+1} = \boldsymbol{\varphi}_0[\boldsymbol{C}\boldsymbol{\varphi}_h(\boldsymbol{D}\boldsymbol{x}_t + \boldsymbol{B}\boldsymbol{u}_t + \boldsymbol{b}_h) + \boldsymbol{b}_o] \tag{5}$$

$$\boldsymbol{y}_{t+1} = \boldsymbol{x}_{t+1} \tag{6}$$

where the matrices $\boldsymbol{B}$ and $\boldsymbol{C}$ and vectors $\boldsymbol{b}_h$ and $\boldsymbol{b}_o$ have the same meaning for the ERN as the IRN, and the matrix $\boldsymbol{D}$ is the matrix of weights for the recurrent connections from the output layer at time $t-1$ to the inputs of the hidden layer at time t.

Although the ERN and IRN can exhibit comparable modeling performance, they have different features that may make one more desirable than the other for a particular process. The IRN does not have any structural limit on the number of model states, as the number of hidden nodes can be freely varied. The ERN, however, can only have the same number of states as model outputs because the outputs are the states. The IRN thus tends to be flexible in modeling. The ERN has the advantage that the model states have a clear physical interpretation in that they are the measured variables of the process itself, whereas the states of IRN are hypothetical and neither unique nor canonical.

Since both types of models are difference equations, to complete the model, a vector of initial values of the model states must be specified. Initialization of ERN models is simple because the user can observe the current values of the process outputs and use those values to initialize the states. Just as with linear state-space models, IRN models are more difficult to initialize, as the states normally lack physical meaning. Neural-network researchers normally initialize the states of IRN models with the median value of the activation function of the hidden nodes (0.5 if the activation function ranges from 0 to 1.0). Inaccuracies in the initialization of the states of the IRN typically result in initial inaccuracies in the model predictions, but the errors die out in a time span of the order of the dominant time constant of the process being modeled. Startup errors can be minimized by holding the network inputs constant at the initial input vector, and evaluating the IRN model until the output of the network reaches steady state. This is equivalent to assuming that $\boldsymbol{u}_t = \boldsymbol{u}_o$ for all $t < 0$.

Model identification

If you choose one of the RNN structures, Eqs. 2 and 3, or 5 and 6, how do you estimate the values of model parameters (the weights) of the network? The standard way from the perspective of investigators using neural networks is to train the networks to reproduce the desired dynamic behavior using the backpropagation-through-time algorithm (Hertz et al., 1991). Closer examination of this technique reveals that what is really being carried out is conventional prediction error estimation (Ljung, 1987), which will be briefly described here.

Let the parameter vector in the RNN nonlinear state-space model be denoted by $\boldsymbol{\theta}$, where

$$\boldsymbol{\theta} = \{\boldsymbol{A}; \boldsymbol{B}; \boldsymbol{C}; \boldsymbol{b}_h; \boldsymbol{b}_o\} \tag{7}$$

for the IRN model and

$$\boldsymbol{\theta} = \{\boldsymbol{B}; \boldsymbol{C}; \boldsymbol{D}; \boldsymbol{b}_h; \boldsymbol{b}_o\} \tag{8}$$

for the ERN model. Let the vector of prediction errors of either model be

$$\boldsymbol{\epsilon}_t(\boldsymbol{\theta}) = \boldsymbol{y}_t - \hat{\boldsymbol{y}}_t(\boldsymbol{\theta}), \tag{9}$$

where $\boldsymbol{y}_t$ is the vector of observed process outputs and $\hat{\boldsymbol{y}}_t(\boldsymbol{\theta})$ is the vector of predictions from the model. The observed data from the process being modeled is the set of input–output vector pairs

$$\boldsymbol{Z}^N = \{\boldsymbol{y}_1, \boldsymbol{u}_1, \boldsymbol{y}_2, \boldsymbol{u}_2, \ldots, \boldsymbol{y}_t, \boldsymbol{u}_t, \ldots, \boldsymbol{y}_N, \boldsymbol{u}_N\}, \tag{10}$$

where N is the number of data samples and $\boldsymbol{u}_t$ is the process input vector. The goal in prediction-error identification is to minimize the prediction error of the model for the data set $\boldsymbol{Z}^N$ by adjusting the parameter vector $\boldsymbol{\theta}$, that is,

$$\min_{\boldsymbol{\theta}} J(\boldsymbol{\theta}, \boldsymbol{Z}^N) = \frac{1}{N}\sum_{t=1}^{N} \boldsymbol{\epsilon}_t^T(\boldsymbol{\theta})\boldsymbol{\epsilon}_t(\boldsymbol{\theta}). \tag{11}$$

Equation 11 is the standard unweighted least-squares objective function.

Because RNN models are nonlinear in the coefficients, iterative methods must be used to minimize Eq. 11. The backpropagation algorithm is a gradient descent scheme that is well suited for parallel implementation in hardware, as each stage uses only local information about the inputs and outputs of each activation node. For simulation on a serial computer, more efficient optimization techniques such as the BFGS or conjugate gradient method are preferred. Although analytic formulation of the gradients of $J(\boldsymbol{\theta}, Z^N)$ with respect to $\boldsymbol{\theta}$ given the equations for the IRN or ERN model is quite complex because of the existence of state feedback, use of the gradient calculation as done in the BP algorithm (Hertz et al., 1991; Werbos, 1990) is both intuitive and computationally efficient. Analytical gradients of the objective function can be combined with an efficient quasi-Newton optimization code [NPSOL (Gill et al., 1986)] to yield rapid parameter identification, as was done in our work.

The parameter estimation scheme described earlier is known as prediction-error estimation. An inherent assumption behind this strategy is that the process output measurements, y_t, only contain additive white noise (noise uncorrelated in time), while the process inputs are assumed deterministic. In reality, these assumptions are rarely met, and it can be shown that even when a simple linear regression is used to model a steady-state process, the presence of noise in the independent variable will yield biased parameter estimates and biased predictions (Draper, 1991). Noise in the inputs is also a serious problem in the identification of linear dynamic models, since when the effect of input noise is neglected, and it exists, predictor-error methods cannot give consistent parameter estimates (Söderström, 1981). If the noise characteristics of the process measurements are known, this problem can be ameliorated to a degree, but in general how to resolve the problem is still open (Ljung, 1987). For nonlinear, nonparametric system identification the problem of bias similarly exists, and is further complicated by the nonlinearity of the model. In the case of nonlinear systems modeled by parametric models, various linearization-based error-in-the-variables methods have been proposed (Kim et al., 1990). Similar methods could be applied to neural-network models if model bias became a serious problem.

Extended Kalman Filtering

Nonparametric data rectification by RNN uses only information from previous time steps to predict the best values of the current measurements. The most current information about the "true" values of the current measurements, namely the current measurements themselves, is not used. Given the fact that RNN are nothing more than nonlinear state-space models, as discussed in the subsection on state-space representation, the primary tool of optimal estimation theory, the Kalman filter, can be directly applied in RNN-based data rectification.

For linear dynamic systems in which the states and measurements are corrupted by additive white noise, and for which a perfect model exists, the Kalman filter provides optimal, unbiased, minimum-variance estimates of the process states, and hence the process measurements (Gelb, 1974; Lewis, 1986). For nonlinear processes, however, derivation of the optimum, minimum variance estimator results in coupled, stochastic differential equations that lack finite-dimensional solutions (Sage and Melsa, 1971). A variety of suboptimal estimation procedures exist, most of which approximate the dynamic and measurement nonlinearities by first- and second-order Taylor series expansions. The most commonly used method is the extended Kalman filter (EKF) (Gelb, 1974).

Consider a general nonlinear discrete-time process modeled as follows

$$\boldsymbol{x}_{t+1} = \boldsymbol{f}(\boldsymbol{x}_t, \boldsymbol{u}_t) + \boldsymbol{G}_t \boldsymbol{w}_t \tag{12}$$

$$\boldsymbol{y}_t = \boldsymbol{h}(\boldsymbol{x}_t, \boldsymbol{u}_t) + \boldsymbol{v}_t \tag{13}$$

$$\boldsymbol{w}_t \sim (\boldsymbol{0}, \boldsymbol{Q}_t), \qquad \boldsymbol{v}_t \sim (\boldsymbol{0}, \boldsymbol{R}_t),$$

where $\boldsymbol{f}(\cdot)$ and $\boldsymbol{h}(\cdot)$ are general nonlinear functions of the process states and inputs, $\boldsymbol{w}_t$ and $\boldsymbol{v}_t$ are zero mean white noise sequences having covariance matrices $\boldsymbol{Q}_t$ and $\boldsymbol{R}_t$, respectively, and $\boldsymbol{G}_t$ is a process noise matrix. If we define the Jacobians

$$\boldsymbol{F}(\boldsymbol{x}_t, \boldsymbol{u}_t) = \frac{\partial \boldsymbol{f}(\boldsymbol{x}_t, \boldsymbol{u}_t)}{\partial \boldsymbol{x}_t} \tag{14}$$

$$\boldsymbol{H}(\boldsymbol{x}_t, \boldsymbol{u}_t) = \frac{\partial \boldsymbol{h}(\boldsymbol{x}_t, \boldsymbol{u}_t)}{\partial \boldsymbol{x}_t}, \tag{15}$$

it is possible to derive a discrete extended Kalman filter (DEKF) having the time update equations

$$\hat{\boldsymbol{x}}_{t+1}^- = \boldsymbol{f}(\hat{\boldsymbol{x}}_t, \boldsymbol{u}_t) \tag{16}$$

$$\boldsymbol{P}_{t+1}^- = \boldsymbol{F}(\hat{\boldsymbol{x}}_t, \boldsymbol{u}_t)\boldsymbol{P}_t\boldsymbol{F}^T(\hat{\boldsymbol{x}}_t, \boldsymbol{u}_t) + \boldsymbol{G}_t\boldsymbol{Q}_t\boldsymbol{G}_t^T \tag{17}$$

and measurement update equations

$$\boldsymbol{K}_{t+1} = \boldsymbol{P}_{t+1}^-\boldsymbol{H}^T(\hat{\boldsymbol{x}}_{t+1}^-, \boldsymbol{u}_{t+1}) \times [\boldsymbol{H}(\hat{\boldsymbol{x}}_{t+1}^-, \boldsymbol{u}_{t+1})\boldsymbol{P}_{t+1}^-\boldsymbol{H}^T(\hat{\boldsymbol{x}}_{t+1}^-, \boldsymbol{u}_{t+1}) + \boldsymbol{R}_t]^{-1} \tag{18}$$

$$\boldsymbol{P}_{t+1} = [\boldsymbol{I} - \boldsymbol{K}_{t+1}\boldsymbol{H}(\hat{\boldsymbol{x}}_{t+1}^-, \boldsymbol{u}_{t+1})]\boldsymbol{P}_{t+1}^- \tag{19}$$

$$\hat{\boldsymbol{x}}_{t+1} = \hat{\boldsymbol{x}}_{k+1}^- + \boldsymbol{K}_{t+1}[\tilde{\boldsymbol{y}}_{t+1} - \boldsymbol{h}(\hat{\boldsymbol{x}}_{t+1}^-, \boldsymbol{u}_{t+1})] \tag{20}$$

with the initial conditions

$$\boldsymbol{P}_0 = \boldsymbol{P}_{x_0}, \qquad \hat{\boldsymbol{x}}_0 = \bar{\boldsymbol{x}}_0.$$

The matrix $\boldsymbol{P}_t$ is the estimation error covariance matrix for the process states, and $\boldsymbol{K}_t$ is the Kalman gain matrix. Symbols with $-$ superscripts refer to predicted quantities before measurement updates are made. The DEKF can be derived a number of ways. The simplest but least rigorous is to linearize the process and measurement model, and directly apply the discrete time Kalman filter for linear systems (Sage and Melsa, 1971).

For each time step, implementation of the Kalman filter can be thought of as consisting of two phases. First, the state variables and error covariance matrix $\boldsymbol{P}$ are *predicted* for time $t+1$ from the estimates and inputs at time t via Eqs. 16 and 17. Then, the states and covariance matrix are *corrected* when

the measurements $\tilde{y}_{t+1}$ arrive by calculating the Kalman gain and carrying out the update using Eqs. 18, 19, and 20. Because the Jacobians are *functions* of the model states, the Kalman gain must be recalculated at each time step as the states change rather than calculated once, off-line, as is commonly done with linear state-space models.

Previously (Karjala and Himmelblau, 1994b) we demonstrated that it is possible to improve the rectification of measurements containing Gaussian noise by embedding the IRN model within an extended Kalman filter. In the next section we extend the concept to processes in which the measurements contain autocorrelated noise. We will use externally recurrent neural networks rather than internally recurrent nets because an ERN model has a fixed (by the number of model outputs) number of states to estimate and because only linearization of the state equation is required rather than linearization of both the state equation and the output/measurement equation. These advantages become even more important if you want to apply EKF techniques to more complicated problems requiring larger neural-network-based models.

Autocorrelated noise and state-vector augmentation

If straightforward RNN modeling by the prediction method outlined in the section on data rectification and system identification and Eq. 1 is used to rectify corrupt measurements, and the measurement errors are highly correlated from time step to time step, the ERN will model not only the deterministic relationship between variables but also the deterministic, autoregressive portion of the correlated noise. As a result, the network will track both the noise and the underlying deterministic value of a process variable. The net result will be poor estimation of the "true" value of each measurement and little reduction in the estimation error.

Application of the Kalman filter when the process and/or measurement noise is autocorrelated has been extensively studied since the mid 1960s (Sage and Melsa, 1971; Gelb, 1974; Lewis, 1986). Various state-augmentation and -derived measurement techniques have been developed, but here we focus on state-vector augmentation. In state-vector augmentation, the Kalman filter is used to estimate both states of the process model and the additional states of the noise model. General discussion of this technique can be found elsewhere (Sage and Melsa, 1971; Gelb, 1974; Lewis, 1986). We demonstrate application of this method to (1) a gravity-flow tank/pipeline example, and (2) to a packed distillation column example.

Using Eqs. 12 and 13, an ERN model can be written in stochastic state-space form as

$$\boldsymbol{x}_{t+1} = \boldsymbol{f}(\boldsymbol{x}_t, \boldsymbol{u}_t) + \boldsymbol{G}\boldsymbol{w}_t \tag{21}$$

$$\boldsymbol{y}_t = \boldsymbol{x}_t + \boldsymbol{v}'_t, \tag{22}$$

where $\boldsymbol{f}(\boldsymbol{x}_t, \boldsymbol{u}_t)$ for the ERN is

$$\boldsymbol{f}(\boldsymbol{x}_t, \boldsymbol{u}_t) = \boldsymbol{C}\boldsymbol{\varphi}(\boldsymbol{D}\boldsymbol{x}_t + \boldsymbol{B}\boldsymbol{u}_t + \boldsymbol{b}_h) + \boldsymbol{b}_o \tag{23}$$

as explained in the subsection on state-space representation. The vector $\boldsymbol{w}_t$ is the process noise vector and $\boldsymbol{G}$ is the process noise matrix. The vector $\boldsymbol{v}'_t$ is the measurement noise vector. If $\boldsymbol{v}'_t$ is uncorrelated and Gaussian, we can use the discrete EKF algorithm directly on this system to "optimally" estimate $\boldsymbol{x}_t$. If $\boldsymbol{v}'_t$ is autocorrelated, the EKF cannot be directly applied to the example process. When $\boldsymbol{v}'_t$ is autocorrelated, we must model the autocorrelations and reformulate the stochastic state-space equations, Eqs. 21 and 22, as an augmented system.

The autocorrelated noise $\boldsymbol{v}'_t$ of Eq. 22 can be modeled in general state-space form as

$$\boldsymbol{v}'_{t+1} = \boldsymbol{A}'\boldsymbol{v}'_t + \boldsymbol{G}'\boldsymbol{w}'_t \tag{24}$$

where $\boldsymbol{v}'_t$ is the autocorrelated noise and $\boldsymbol{w}'_t$ is zero mean Gaussian noise $[\boldsymbol{w}'_t \sim N(\boldsymbol{0}, \boldsymbol{Q}')]$, which drives the $\boldsymbol{v}'_t$ process. $\boldsymbol{A}'$ is the state transition matrix and $\boldsymbol{G}'$ is the process noise matrix for this noise process. The primes are used to distinguish matrices and vectors associated with the noise model from those of the overall state-space model in what follows. With Eq. 24 we can then construct an augmented system for both the ERN model (Eqs. 22 and 21) and noise model, Eq. 24, by appending the noise states $\boldsymbol{v}'_t$ to the ERN model states $\boldsymbol{x}_t$. The augmented state equation becomes

$$\begin{bmatrix} \boldsymbol{x}_{t+1} \\ \boldsymbol{v}'_{t+1} \end{bmatrix} = \begin{bmatrix} \boldsymbol{f}(\boldsymbol{x}_t, \boldsymbol{u}_t) \\ \boldsymbol{A}'\boldsymbol{v}'_t \end{bmatrix} + \begin{bmatrix} \boldsymbol{G} & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{G}' \end{bmatrix} \begin{bmatrix} \boldsymbol{w}_t \\ \boldsymbol{w}'_t \end{bmatrix}, \tag{25}$$

with the measurement equation

$$\boldsymbol{y}_t = [\boldsymbol{I}|\boldsymbol{I}] \begin{bmatrix} \boldsymbol{x}_t \\ \boldsymbol{v}'_t \end{bmatrix} + \boldsymbol{v}^*_t, \tag{26}$$

where $\boldsymbol{v}^*_t$ denotes the hypothetical measurement noise vector for the augmented system and $\boldsymbol{I}$ is the identity matrix. The process or state noise for the augmented system can be denoted as $\boldsymbol{w}^*_t = [\boldsymbol{w}_t\, \boldsymbol{w}'_t]^T$, and is zero mean, white, and Gaussian with a covariance matrix

$$\boldsymbol{Q}^* = \begin{bmatrix} \boldsymbol{Q} & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{Q}' \end{bmatrix}. \tag{27}$$

As this system of equations has been formulated, the measurement noise vector $\boldsymbol{v}^*_t$ does not exist. The associated covariance matrix $\boldsymbol{R}^*$ becomes a matrix of nonzero tuning parameters in the DEKF implementation for the augmented system. Alternate versions of the Kalman filter exist that are designed for the case of zero measurement noise, but we chose to stick to the standard algorithm.

Because the augmented system is nonlinear in $\boldsymbol{x}_t$, the system must be linearized. Linearization of the augmented state-space model yields the linearized state-transition matrix used in the EKF

$$\boldsymbol{F}^*(\hat{\boldsymbol{x}}_t, \boldsymbol{u}_t) = \begin{bmatrix} \boldsymbol{F}(\hat{\boldsymbol{x}}_t, \boldsymbol{u}_t) & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{A}' \end{bmatrix}. \tag{28}$$

Implementation of the discrete Kalman filter requires the time and measurement update equations of Eqs. 16–20 with

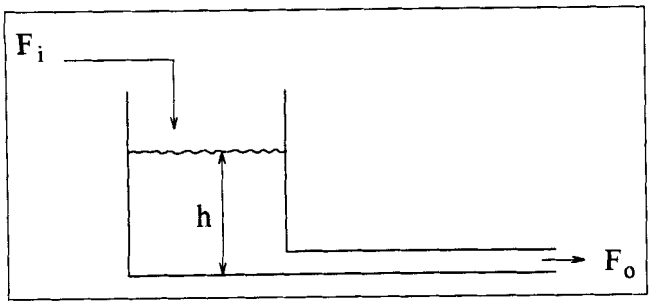

Figure 2. Gravity-flow tank/pipeline combination.

$F^*(\hat{x}_t, u_t)$, H^*, G^*, Q^*, and R^* replacing $F(\hat{x}_t, u_t)$, H, G, Q, and R, where

$$H^* = [I \quad I] \tag{29}$$

and

$$G^* = \begin{bmatrix} G & 0 \\ 0 & G' \end{bmatrix}. \tag{30}$$

Gravity-Flow Tank/Pipeline Example

The first example of rectification uses the gravity-flow tank example of Luyben (1990). This system is characterized by time-varying flow into the tank with gravity-induced flow out of the tank through a long pipeline where both momentum and friction effects are present (Figure 2). The equations governing this system are

$$\frac{dF_o}{dt} = \frac{A_p g}{L} h - \frac{K_f}{\rho A_p^2} F_o^2, \tag{31}$$

which is derived from a force balance, and the following mass-balance equation

$$\frac{dh}{dt} = \frac{1}{A_t}(F_i - F_o). \tag{32}$$

Here F_o is the volumetric flow rate of fluid leaving the tank via the pipe, F_i is the volumetric flow rate of fluid entering the tank, t is time, g is the gravitational acceleration, L is the length of the pipe, h is the height of liquid in the tank, K_f is the friction factor, ρ is the density of the liquid, A_p is the cross-sectional area of the pipe, and A_t is the cross-sectional area of the tank. Using the parameters of Table 1, these equations reduce to

$$\frac{dF_o}{dt} = (7.00 \times 10^{-3})h - (1.04 \times 10^{-2})F_o^2 \tag{33}$$

Table 1. Gravity-Flow Tank Parameters

g	9.81 m/s^2
L	914 m
K_f	4.41 N·s^2/m^3
ρ	998 kg/m^2
A_p	0.653 m^2
A_t	10.5 m^2

$$\frac{dh}{dt} = (9.52 \times 10^{-2})(F_i - F_o). \tag{34}$$

The measurement vector m_t consists of observations of F_o, h, and F_i

$$m_t = [F_o \quad h \quad F_i]^T + w_t, \tag{35}$$

where w_t is the measurement error vector.

In cases in which independent autocorrelated noise was added to a measurement, the individual noise model for each measurement error was

$$w_t = \alpha w_{t-1} + \beta e_t, \tag{36}$$

where e_t is Gaussian noise and α and β are constants. In the Z domain this is equivalent to the transfer function

$$w(z) = \frac{\beta}{1 + \alpha z^{-1}} e(z). \tag{37}$$

This noise model is stable for $\alpha < 1$. For the examples described here, α and β were chosen to be equal to $\sqrt{1/2}$ so that w_t would have approximately the same variance as e_t. The standard deviation for e_t was chosen so that the standard deviation of each of the elements of the autocorrelated noise vector w_t would be approximately 5% of the range of each measured variable. The noise model might be identified from steady-state process measurements or by iterative RNN/noise modeling, but will be assumed known for the examples.

Simulated measurements including the preceding autocorrelated measurement errors were generated by first integrating the model equations, Eqs. 31 and 32, in response to 50 random step changes in F_i of random duration using a 10-s sampling time. Then autocorrelated or Gaussian measurement errors were added to each simulated measurement, depending on the case being studied. Two different data sets were created—a 742 sample identification set and a 747 sample test set. In the subsequent tables that list the performance of our strategy, the integral squared error and standard deviation of the error are based on the entire test set. To enable a reader to see the individual data points along with the rectified data, subsets of 100 typical data points have been used in the figures.

Based on our previous experience, we focused our attention on the process outputs F_o and h, as we lacked a model for the process input F_i, and the EKF approach is a model-based technique. Furthermore, we only sought to estimate the autocorrelated noise states for these two output variables. If we combine the individual noise models into the state-space form of Eq. 24, the state-transition and state-noise matrices become

$$A' = G' = \begin{bmatrix} \frac{1}{\sqrt{2}} & 0 \\ 0 & \frac{1}{\sqrt{2}} \end{bmatrix}. \tag{38}$$

Table 2. Type of Noise Present in Data Sets Used for Model Identification and Testing of Rectification Algorithms for the Tank/Pipeline Example

	Identification Data Set		Test Data Set	
	Input Variable F_i	Output Variables F_o, h	Input Variable F_i	Output Variables F_o, h
Case 1	Deterministic	Gaussian	Deterministic	Autocorrelated
Case 2	Autocorrelated	Autocorrelated	Autocorrelated	Autocorrelated

In order to compare different data-rectification techniques and examples, scalar performance measures are needed. If we define the estimation error $\delta_i(t)$ as the deviation between individual measurement estimates and the true values of these measurements

$$\delta_i(t) = \hat{m}_i(t) - m_i(t)_{\text{true}}, \tag{39}$$

then we can define the integral squared error (ISE) as

$$\text{ISE}_i = \sum_{t=1}^{N} \delta_i(t)^2, \tag{40}$$

and the sample standard deviation (s_i) of the estimation error as

$$s_i = \sqrt{\frac{N\sum_{t=1}^{N}\delta_i(t)^2 - \left[\sum_{t=1}^{N}\delta_i(t)\right]^2}{N(N-1)}}. \tag{41}$$

Both of these measures are used to evaluate rectification performance in the examples and were calculated for the test data set.

In the next two subsections we present simulation results for rectification of measurements containing autocorrelated errors using externally recurrent network models that contained 14 adjustable parameters (weights), 2 Gaussian hidden nodes, 2 linear output nodes, and a single input node, together with the discrete extended Kalman filtering. To test the performance of the ERN-DEKF technique, several different cases were run that ranged from optimal but unrealistic to suboptimal but realistic. The characteristics of each case are summarized in Table 2.

Tank/pipeline case 1

In the first example of ERN-DEKF rectification, we sought to determine the best performance of this technique under optimal conditions. We began by separating the process input F_i from the process outputs h and F_o, and sought only to estimate the true values of the liquid level in the tank and the flow rate out of the tank. An externally recurrent neural network was identified using a training data set containing deterministic inputs $u_t = (F_i)_t$ and outputs $y_t = [F_o\, h]_t^T$ with added 5% uncorrelated Gaussian noise. As a result, none of the implicit assumptions of least squares were violated so that the best possible ERN model was obtained. In the test data set, the input variable F_i was deterministic, but both of the output variables, F_o and h contained autocorrelated measurement errors. None of the underlying Kalman filter assumptions were violated for the augmented system so that best possible performance was obtained. Figure 3 shows the results for the liquid level in the tank and Figure 4 shows the

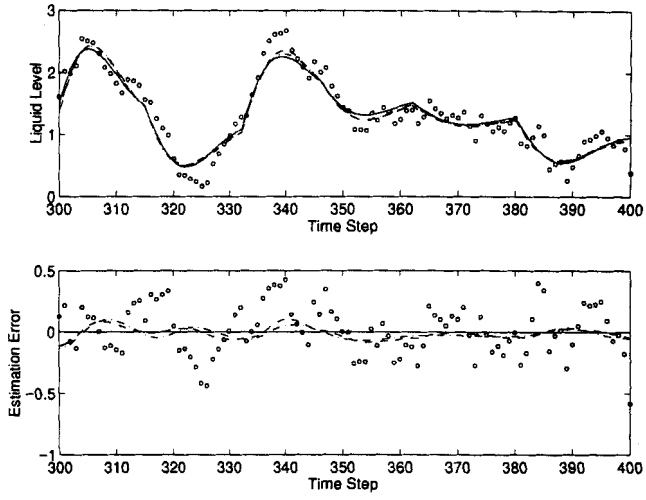

Figure 3. Case 1: Results for the rectification of tank level when the process input was deterministic and the model was accurate.

o = measurements; —— = true values; -·- = rectification by the ERN; - - - = rectification by the combined ERN-DEKF.

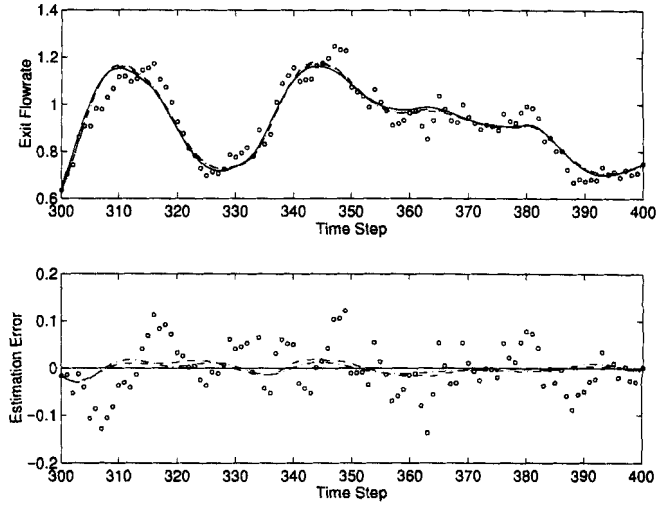

Figure 4. Case 1: Results for the rectification of exit flow rate when the process input was deterministic and the model was accurate.

o = measurements; —— = true values; -·- = rectification by the ERN; - - - = rectification by the combined ERN-DEKF.

Table 3. Case 1: Performance of the Data-Rectification Methods Given Deterministic Inputs, Outputs Containing Autocorrelated Noise, and an Accurate ERN Model

	Integral Squared Error	
	Liquid Level	Flow Out
ERN-DEKF	0.1137	1.9881
ERN	0.1183	2.1143
Original noise	2.3107	31.3504
	Standard Deviation of Error	
	Liquid Level	Flow Out
ERN-DEKF	0.0122	0.0516
ERN	0.0125	0.0532
Original noise	0.0556	0.2047

results for the exit flow rate from the tank for the same portion of the test data. Because the ERN models the liquid level and exit flow rate as a function of the deterministic inlet flow rate, and is a good model, it does a good job of estimating these two variables by itself. The ERN-DEKF, however, does better yet as measured by the numerical results of Table 3, and by the slightly visible difference in the curves in the estimation error graphs in Figures 3 and 4. Ideally the trajectories should track the correlated portion of the measurement error exactly, but they do so only partially. Empirical modeling via RNN is never exact, and subtle model bias is reflected in these estimates. Nonetheless, the rectification/estimation results for this example were very good. By good we mean the error in the measurements was reduced by rectification. Naturally, the end use of the rectified measurements determines how effective the rectification is for a user.

Tank/pipeline case 2

The second tank/pipeline example treats the most realistic and difficult case, one in which all three measurements, F_i, h, and F_o, contain correlated noise, and an ERN model must

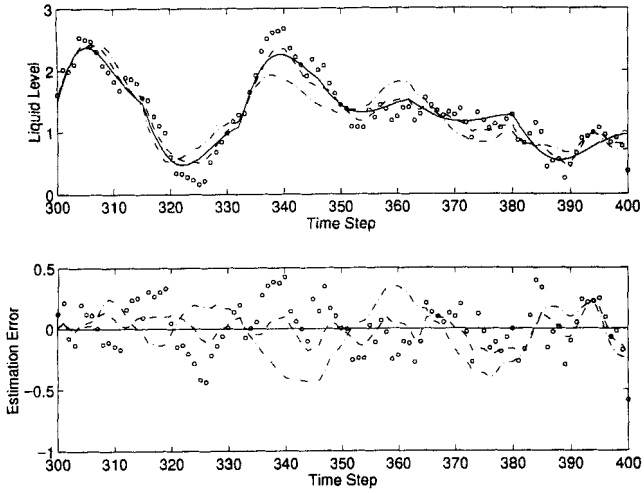

Figure 5. Case 2: Results for the rectification of tank level when the process input contains autocorrelated noise.

○ = measurements; —— = true values; -·- = rectification by the ERN; – – – = rectification by the combined ERN-DEKF.

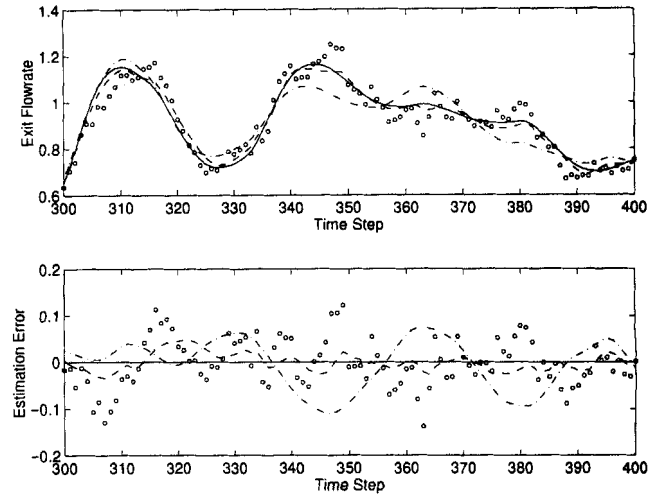

Figure 6. Case 2: Results for the rectification of exit flow rate when the process input contains autocorrelated noise.

○ = measurements; —— = true values; -·- = rectification by the ERN; – – – = rectification by the combined ERN-DEKF.

be identified from these data. Once again the decision was made to focus only on the rectification of h and F_o as no model can be developed to predict the process input F_i. It is important to recognize that when an ERN model or any model is identified from data containing autocorrelated noise in the model input, a biased model can be expected to result, as one of the assumptions underlying least squares is violated (no noise in the process inputs). If autocorrelated noise is added to the process outputs, another assumption underlying least squares is violated, namely that only independent noise exists in the process outputs. As Figures 5 and 6 indicate, the predictions by the ERN for the test data deviate from both the measurements of the process outputs and the true values themselves. Part of the lack of accuracy is caused by the bias present in the model, but part is also due to the autocorrelated measurement errors in the input variable F_i. The DEKF assumption of noise-free process inputs was also violated, but augmentation of the ERN model with the DEKF resulted in a significant improvement over the ERN model alone. The ERN-DEKF estimates of h and F_o are closer to their respective true values than either the measurements or the ERN predictions alone, as reflected in Table 4. Given the fact that

Table 4. Case 2: Performance of the Data-Rectification Methods Given Both Inputs and Outputs Containing Autocorrelated Noise and a Biased ERN Model

	Integral Squared Error	
	Liquid Level	Flow Out
ERN-DEKF	0.5278	11.8363
ERN	1.5791	29.3342
Original noise	2.3107	31.3504
	Standard Deviation of Error	
	Liquid Level	Flow Out
ERN-DEKF	0.0266	0.1257
ERN	0.0459	0.1980
Original noise	0.0556	0.2047

assumptions are being violated in both identification of the ERN and application of the DEKF, the results from the ERN-DEKF combination do improve the quality of the data.

Packed Distillation Column Example

The second example of this article is based on simulation of a first-principles dynamic model of a packed distillation column developed by Patwardhan and Edgar (1993) as a distributed parameter system. Figure 7 is a sketch of the column in which a binary mixture of cyclohexane and *n*-heptane are separated. Patwardhan derived a model for the column under the simplifying assumptions of equimolar counterdiffusion and negligible liquid-phase mass-transfer resistance. The Appendix lists the partial differential and algebraic equations of the model, and the values of the model parameters can be found in Patwardhan (1991). Of primary importance for any type of feedback control scheme for this process is the estimation of the true values of the distillate and bottoms concentration, and our rectification examples focus on this task.

The model equations were solved by discretization via orthogonal collocation on finite elements in space yielding a system of 89 coupled differential/algebraic equations to be integrated through time. Incorporation of the full quasi first-principles model within a traditional data-reconciliation scheme in order to reduce even Gaussian measurement errors would be a difficult task, both mathematically and numerically. We used the model in the Appendix as the "true" process, and applied our proposed rectification method to simulated data so that rectified and true values of the variables could be compared.

Six process variables of interest occur in the packed-column example: the feed flow rate (**feed**), the feed composition (***xf***), the vapor boil-up rate (***vbr***), the distillate rate (**dist**), the distillate composition (***xd***), and the bottoms composition

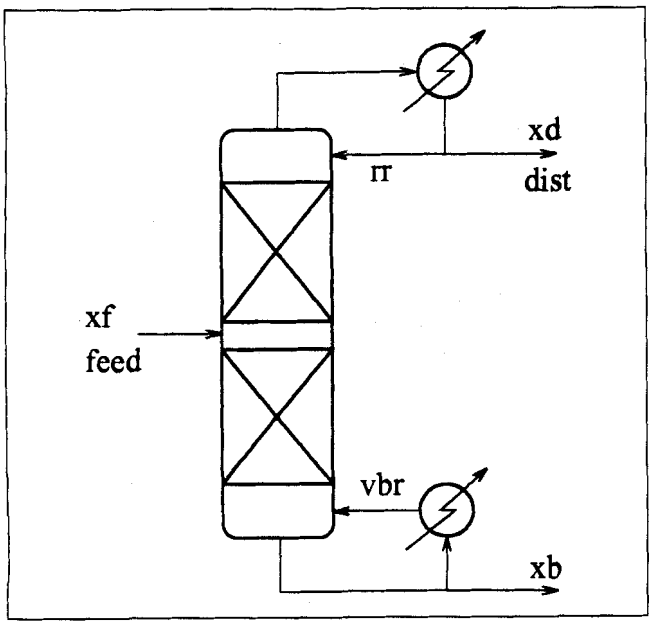

Figure 7. Packed distillation column.

xf is feed composition; **feed** is feed rate; ***rr*** is reflux ratio; ***xd*** is distillate composition; **dist** is distillate rate; ***vbr*** is vapor boil-up rate; and ***xb*** is bottoms composition.

Table 5. Extended Operating Range for the Packed-Distillation-Column Example

Variable	Units	Lower Limit	Upper Limit
Feed composition	mole fraction	0.4	0.6
Feed rate	mol/s	1.25	1.55
Distillate rate	mol/s	0.5	1.2
Vapor boil-up rate	mol/s	3.0	17.0

(***xb***). The vapor boil-up rate and the distillate rate were the manipulated variables and were assumed deterministic. The feed flow rate and feed composition both served as measured disturbances, and were made to contain autocorrelated measurement errors. The distillate composition and bottoms composition were the two process output variables of primary interest, as these variables would be the controlled variables under closed-loop conditions. The simulated measurements of both of these variables (***xd***, ***xd***) also contained autocorrelated measurement errors.

An interesting feature of this packed-column model is that the steady-state gain is highly nonlinear and undergoes a sign change (Patwardhan and Edgar, 1993; MacMurray, 1993) for the extended operating range of Table 5. This type of behavior is not generally encountered in distillation columns containing trays or random packing, but appears to be specific to the structured packing used in the experimental system for which this model was derived.

The data sets used to identify the ERN model and test the ERN-DEKF algorithm were generated by using Patwardhan's model to generate the response of the column in terms of distillate and bottoms composition to open-loop random step changes in feed rate, feed composition, distillate rate, and vapor boil-up rate. Due to numerical instability in the FORTRAN code used to solve the model, it was necessary to limit the rate of change of the input variables to 5% of their range (see Table 5) per 5-min sampling interval. The result was ramp-type input data where all four input variables were perturbed simultaneously every 20 samples. Five percent autocorrelated noise was added to the process inputs ***xf*** and **feed** and the process outputs ***xd*** and ***xf*** as mentioned earlier and summarized in Table 6. The identification and test data sets made up 2,040 samples each, and contained 101 simultaneous ramp changes in each input variable.

Although the data used may seem excessive, the problem in estimating the coefficients in a recurrent neural net is the same as fitting any dynamic model. You have to cover the space of the variables sufficiently well to model the expected domain of operation (both steady and unsteady state). A net can be fit with much less data by sacrificing the range of sat-

Table 6. Type of Noise Present in Data Sets Used for Model Identification and Testing of the Rectification Algorithms for the Packed-Column Example*

Identification Data Set		Test Data Set	
Input Variables **feed**, ***xf***	Output Variables ***xd***, ***xb***	Input Variables **feed**, ***xf***	Output Variables ***xd***, ***xb***
Autocorrelated	Autocorrelated	Autocorrelated	Autocorrelated

*The manipulated variables *vbr* and *dist* were deterministic.

isfactory use. Furthermore, although the amount of data needed to fit RNN is usually more than that needed to fit first-principles models (because the nets have many more coefficients), RNNs can often be developed in much less time than can first-principles models for an equivalent error in prediction.

Previous attempts to model the column with externally recurrent networks (MacMurray, 1993) required the use of very large data sets (30,000 samples), and rather large ERN networks (8 input nodes, 15 hidden nodes, 2 outputs, and 197 parameters) incorporating finite impulse-response history windows in the ERN inputs. By using Gaussian rather than sigmoid activation functions, reducing the sampling rate, and using more robust software, we were able to model the operating region of this process with an ERN model having only 4 inputs, 3 hidden nodes, and 2 outputs, and a total of 29 parameters with much less data.

Figures 8 and 9 demonstrate the performance of the ERN-DEKF rectification technique applied to the simulated measurements of the packed distillation column. Some of the graphs show negative values because of the low (essentially zero) concentration involved and the high level of noise present. The true value of the concentration never went negative. We choose to use these negative values within the rectification scheme rather than censure the raw measurements by setting negative values equal to zero. The results shown in these figures are analogous to Case 2 for the tank/pipeline example because the measured inputs xf and **feed** both contained autocorrelated noise. The packed-column system was more complex and difficult to model, as it had four inputs and two outputs compared to the one input, two output tank/pipeline model. In addition, the presence of autocorrelated noise in the inputs as well as the outputs violated the basic assumptions for both the least-squares identification and the EKF. Nevertheless, the values obtained by rectification for both the distillate composition (Figure 8) and the bottoms composition (Figure 9) measurements seem to be reasonable,

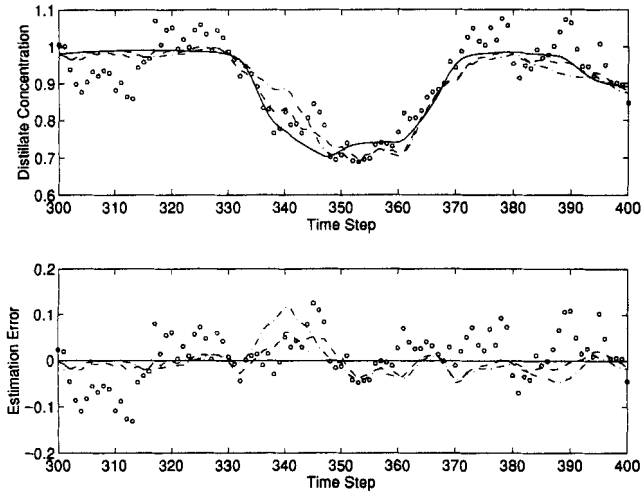

Figure 8. Packed column: results for the rectification of distillate composition when the process inputs contain autocorrelated noise.

○ = measurements; —— = true values; -·- = rectification by the ERN; – – – = rectification by the combined ERN-DEKF.

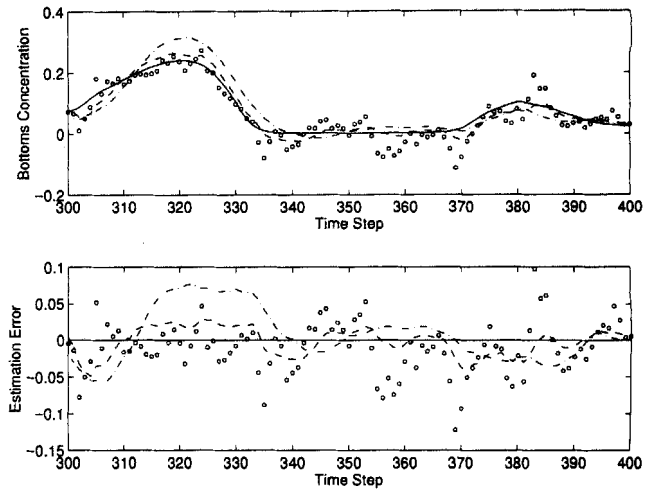

Figure 9. Packed column: results for the rectification of bottoms composition when the process inputs contain autocorrelated noise.

○ = measurements; —— = true values; -·- = rectification by the ERN; – – – = rectification by the combined ERN-DEKF.

as indicated by the data for the ERN-DEKF combination in Table 7. The ERN-DEKF combination showed only slight improvement in terms of s, the standard deviation of the estimation error, for the distillate composition, but showed more substantial reduction for the bottoms composition. We did not include transport or measurement delay in our simulations. Nevertheless, if delay occurs in a process that is related to a process variable, the recurrent net can automatically accommodate the delay (Cheng et al., 1995).

Because noise in the input data, particularly autocorrelated noise as shown in the two previous examples, leads to identification of biased models, an engineer must keep in mind that rectification of data from processes with input noise will not be as satisfactory as instances in which the input noise can be eliminated by modeling, filtering, physical changes, and so on. Nevertheless, our simulation results indicate that improvement can be achieved, that is, good prediction obtained from sparsely sampled data with noisy inputs. One reason for this outcome may be that the additive noise in the inputs to the EKF may merge with the state noise vector w_t to yield an

Table 7. Performance of the Data-Rectification Methods for the Packed-Column Example Given Both Inputs and Outputs Containing Autocorrelated Noise

	Integral Squared Error	
	Distillate Composition	Bottoms Composition
ERN-DEKF	0.4021	0.1545
ERN	1.8605	1.1602
Original noise	5.5341	2.9675
	Standard Deviation of Error	
	Distillate Composition	Bottoms Composition
ERN-DEKF	0.0275	0.0176
ERN	0.0291	0.0237
Original noise	0.0518	0.0380

effective noise term with zero mean but a different variance. Since the covariance matrix of w_t is effectively a set of tuning parameters in practice, the robustness of our proposed strategy may not be degraded very much.

Estimation of Measurement Bias Using the Kalman Filter

As we have seen in the previous examples, formulation of recurrent neural-network models as state-space models and incorporation of these models within a state-augmented Kalman filtering framework allows us to address the dynamic rectification of process measurements containing autocorrelated measurement error. State augmentation and Kalman filtering can also be applied in other circumstances, namely the estimation of measurement bias in both process input and output variables. This approach can be contrasted with other approaches in which two separate filters, one to estimate model states and one to estimate bias, are used in order to reduce computational complexity for high-order systems (Friedland, 1969; Zhou et al., 1993; Alouani et al., 1993).

Estimation of measurement bias in process outputs

When measurement bias exists in a process output variable and a good input–output model exists, it is tempting to simply assume that the predictions of the model are correct and to estimate the measurement bias as the difference between the measurement and the model prediction. Such a procedure is not optimal, however, as it does not utilize information available regarding the characteristics of the measurement and state noise, and very noisy estimates can result. In most cases it is better to employ a state-augmented Kalman filter to estimate the bias.

Because Kalman and extended Kalman filtering are model-based estimation techniques, it is necessary to model the bias expected in a process measurement. If knowledge of the bias dynamics is lacking, as is normally the case, the simplest approach is to model the measurement bias as a constant plus noise, which in state-space form is

$$b_{t+1} = b_t + w'_t, \tag{42}$$

where b_t is the bias vector at time t and w'_t is the state noise for the bias vector.

The augmented state equation for bias estimation in the process output is

$$\begin{bmatrix} x_{t+1} \\ b_{t+1} \end{bmatrix} = \begin{bmatrix} f(x_t, u_t) \\ b_t \end{bmatrix} + \begin{bmatrix} G & 0 \\ 0 & I \end{bmatrix} \begin{bmatrix} w_t \\ w'_t \end{bmatrix} \tag{43}$$

and the measurement equation is

$$y_t = [I|I] \begin{bmatrix} x_t \\ b_t \end{bmatrix} + v_t^*, \tag{44}$$

where the matrices and vectors have meanings similar to those explained in the subsection on autocorrelated noise and state-vector augmentation. Extended Kalman filtering can be directly and easily applied to this augmented system as demonstrated in the next example.

Estimation of bias in the output measurement for the packed distillation column

We used the packed column to demonstrate state-augmented bias estimation using the EKF. We began by identifying a ERN model using data free from measure bias but containing Gaussian errors in the measured variables **feed**, ***xf***, ***xd***, ***cb***. The values of ***vbr*** and **dist** were deterministic. Once identified, the ERN model used the variables **feed**, ***xf***, ***vbf***, and **dist** as inputs and predicted the process outputs ***xd*** and ***xb***.

In the next two cases, simulated noisy data were generated for the column that included a ramp bias (1) in the distillate composition (an output) and (2) in the feed composition. Once the simulated process data were generated in the modeling and prediction stage, we modeled the bias simply as a constant plus noise. We did decide to identify the variable in which bias occurred, but this was an arbitrary decision made simply to reduce the number of states in the EKF. We could just as well have assumed that a potential bias existed in each variable at the expense of estimating many more states. In an actual process each suspect sensor or variable could have an associated bias term introduced.

A slow, ramp-type measurement bias was introduced into the distillate composition (***xd***) at time step 300. This measurement bias increased linearly from zero to 20% of ***xd*** at time step 400 at which time the bias was removed. You can see in Figure 10 the change in the measurements compared with the "true" values, and the ERN and ERN-DEKF estimates of ***xd***. The presence of the output bias did not, of course, impact the ERN prediction, as the ERN prediction was not dependent on the measurement ***xd***. Figure 11 shows the estimates of the bias by the ERN-DEKF procedure. Although we modeled the bias as a constant plus noise, the ERN-DEKF does a good job of tracking the ramp as well as the step change in the bias when the bias was removed at

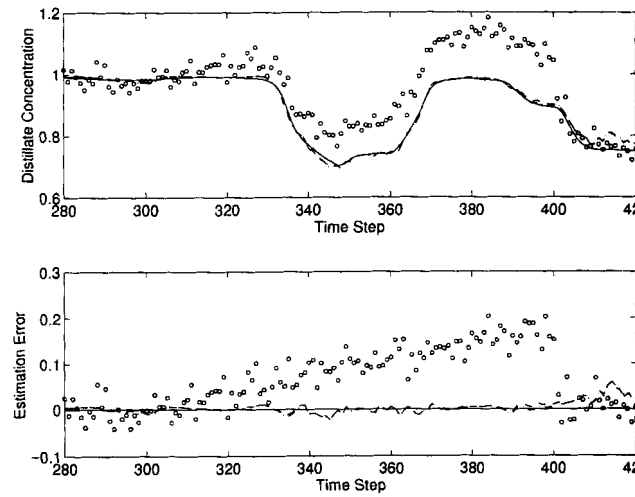

Figure 10. Packed column: results for the rectification of distillate composition when a ramp-type measurement bias is present in the distillate-composition measurement.

○ = measurements; —— = true values; –·– = rectification by the ERN; – – – = rectification by the combined ERN-DEKF.

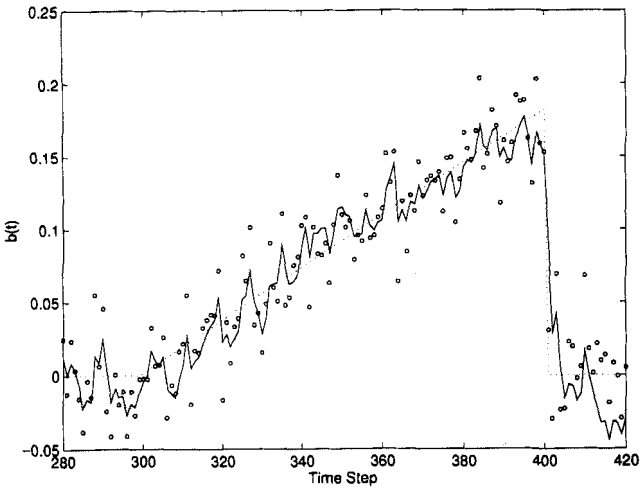

Figure 11. Packed column: estimation of the output measurement bias in the distillate-combined measurement.

○ = measurement errors including bias; —— = bias state estimates; · · · · = true value of bias.

time step 400. Figure 12 shows the results for the bottoms composition. No bias was present in this variable, and both the ERN and ERN-DEKF estimates were unaffected by bias.

Although the figures for this example indicate that the ERN alone makes reasonable estimates of the process outputs despite the existence of the bias in the distillate composition measurement, there are benefits to using the state-augmented ERN-DEKF combination. The primary benefit is optimal estimation of the magnitude of the bias, but, in addition, the estimates of ***xd*** and ***xb*** are improved, as are the numbers for integral squared estimation error and standard deviation of estimation error listed in Table 8.

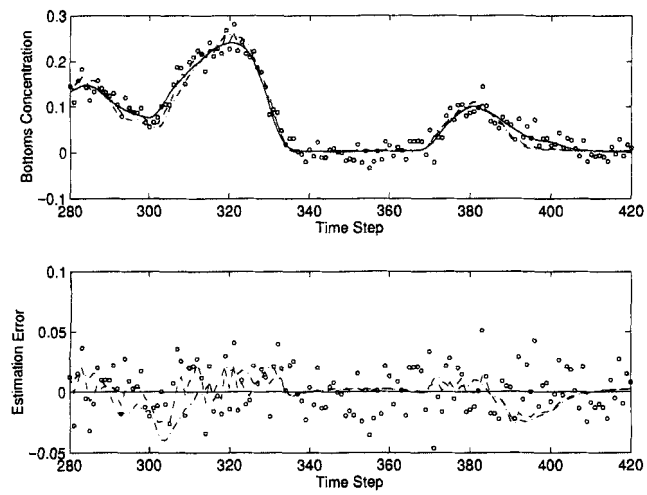

Figure 12. Packed column: results for the rectification of bottoms composition when a ramp-type measurement bias was present in the distillate-composition measurement.

○ = measurements; —— = true values; - · - = rectification by the ERN; – – – = rectification by the combined ERN-DEKF.

Table 8. Performance of the Data-Rectification Methods for the Packed-Column Example when a Single Ramp-Type Measurement Bias is Present in the Distillate-Composition Measurement

	Integral Squared Error	
	Distillate Composition	Bottoms Composition
ERN-DEKF	0.0975	0.0190
ERN	0.1028	0.0361
Noise + bias	1.4138	0.1730
	Standard Deviation of Error	
	Distillate Composition	Bottoms Composition
ERN-DEKF	0.0138	0.0062
ERN	0.0141	0.0085
Noise + bias	0.0495	0.0186

Estimation of measurement bias in process inputs

Measurement bias in the process input variables poses a more difficult problem in dynamic data rectification than does estimating measurement bias in the output variables. Because the predictions of an input–output model are absolutely dependent on the model inputs, measurement bias in the model inputs will seriously bias the predictions of such a model. State-augmented extended Kalman filtering can be used to alleviate these difficulties.

The augmented state equation for bias estimation in the process input variable is

$$\begin{bmatrix} \boldsymbol{x}_{t+1} \\ \boldsymbol{b}_{t+1} \end{bmatrix} = \begin{bmatrix} f(\boldsymbol{x}_t, \boldsymbol{u}_t - \boldsymbol{b}_t) \\ \boldsymbol{b}_t \end{bmatrix} + \begin{bmatrix} \boldsymbol{G} & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{I} \end{bmatrix} \begin{bmatrix} \boldsymbol{w}_t \\ \boldsymbol{w}'_t \end{bmatrix} \tag{45}$$

and the measurement equation is

$$\boldsymbol{y}_t = [\boldsymbol{I}|\boldsymbol{0}] \begin{bmatrix} \boldsymbol{x}_t \\ \boldsymbol{b}_t \end{bmatrix} + \boldsymbol{v}_t^*, \tag{46}$$

where the matrices and vectors have meanings similar to those given before and **0** is the null matrix. We subtracted the bias vector $\boldsymbol{b}_t$ from the process input vector $\boldsymbol{u}_t$ in order to compensate for its presence. The process outputs $\boldsymbol{y}_t$ in this case are solely a function of the states $\boldsymbol{x}_t$ in the ERN model. In the state equation the updated model states are a nonlinear function of both themselves and the bias states. This means that this augmented system is nonlinear in all of the states, unlike the case of output bias or the case of correlated noise. Additional linearization is required to implement the Jacobian for this augmented system.

Estimation of input measurement bias for the packed distillation column

Once again we began by identifying a 29-parameter ERN model in which $\boldsymbol{u}_t = [\textbf{feed } \boldsymbol{xf} \textbf{ dist } \boldsymbol{vbr}]^T$ and where $\boldsymbol{y}_t = [\boldsymbol{xf}\ \boldsymbol{xd}]^T$. The variables ***dist*** and ***vbr*** were deterministic, but the other four variables contained Gaussian noise.

A ramp-type bias similar to that used in the previous example was introduced into the feed composition measurements.

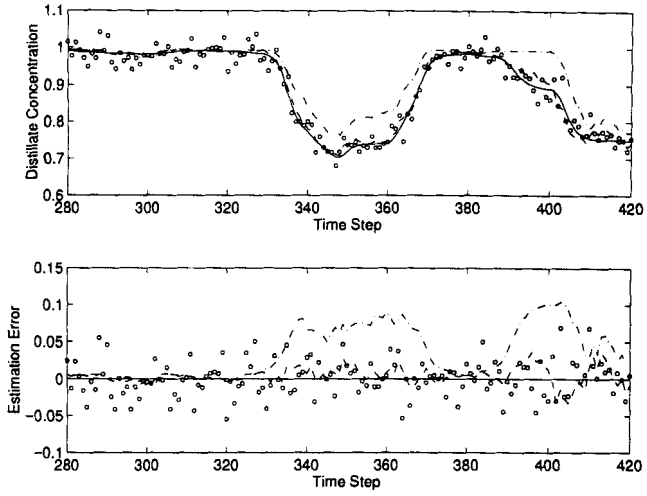

Figure 13. Packed column: results for the rectification of distillate composition when a ramp-type input measurement bias is present in the feed-composition measurement.

○ = measurements; —— = true values; – · – = rectification by the ERN; – – – = rectification by the combined ERN-DEKF.

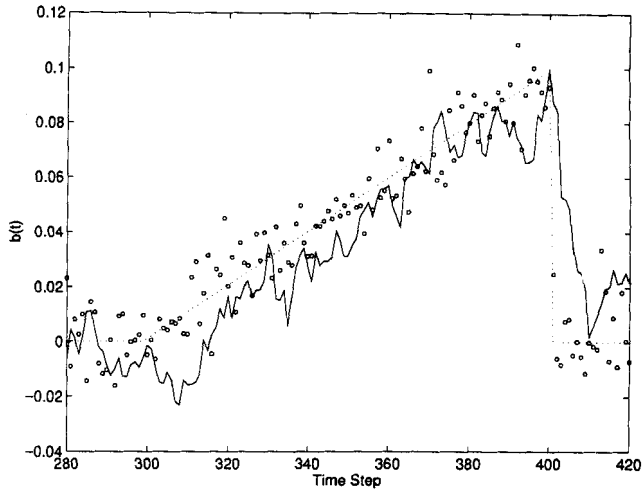

Figure 15. Packed column: estimation of the input measurement bias in the feed-composition measurement.

○ = measurement errors including bias; —— = bias state estimates; · · · · = true value of bias.

The ramp began at time step 300 and increased linearly until time step 400, when it was removed. The effect of the bias in the inputs of the ERN model was significant and affected both outputs. Figures 13 and 14 show that the effect of the ramp bias in *xf* on the ERN predictions was serious and clearly did not appear as a simple ramp in the outputs. The ERN-DEKF combination was able to compensate for the bias, and continued to make excellent estimates of the true values of *xd* and *xb*. Figure 15 shows the ERN-DEKF estimates of the input bias. The ERN-DEKF did a reasonable job of estimating this bias, but did not do quite as well as for the case of output bias. The difference in outcomes could be caused by several factors including the necessary linearization of the ERN states with respect to the bias, model mismatch between the ramp bias and the bias model used, or even model bias in the ERN model. The tuning of the Kalman filter can be adjusted to allow faster tracking of the bias, but at the expense of a higher variance estimate. The numbers in Table 9 indicate that the ERN-DEKF combination is very effective in reducing the estimation error when bias is present in the measurements of the process inputs.

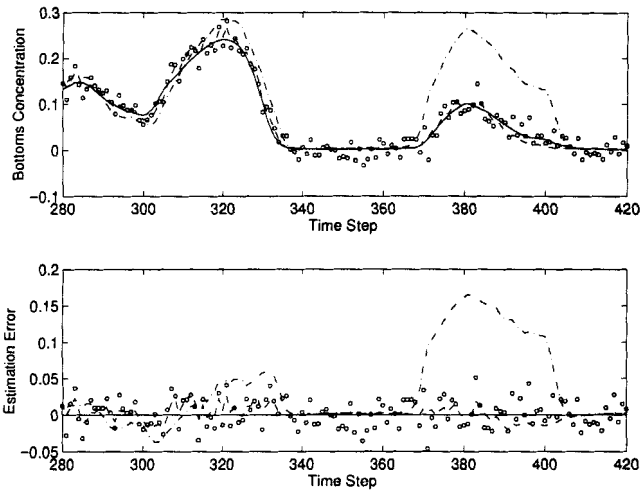

Figure 14. Packed column: results for the rectification of bottoms composition when a ramp-type measurement bias was present in the feed-composition measurement.

○ = measurements; —— = true values; – · – = rectification by the ERN; – – – = rectification by the combined ERN-DEKF.

Discussion

As demonstrated by the examples, state-augmented Kalman filtering is very useful in neural-network-based data rectification. RNNs and the EKF can work well for rectification of measurements from a dynamic process containing autocorrelated error when a noise model is available. In the examples in this article we assumed a noise model. Ordinarily, the noise

Table 9. Performance of Data-Rectification Methods for the Packed-Column Example when a Single Ramp-Type Measurement Bias is Present in the Feed-Composition Measurement

	Integral Squared Error	
	Distillate Composition	Bottoms Composition
ERN-DEKF	0.0711	0.0230
ERN	0.4259	0.6309
Noise	0.2826	0.1730
	Standard Deviation of Error	
	Distillate Composition	Bottoms Composition
ERN-DEKF	0.0119	0.0067
ERN	0.0285	0.0340
Noise	0.0238	0.0186

model must be identified along with the process model, typically by using linear system identification techniques and the prediction residuals created during identification of the nonlinear RNN model. Bias estimation/compensation also requires a model. In the examples that we have evaluated, modeling the bias simply as a constant plus noise was successful.

In the generation of simulated data we used a coefficient of variation of 0.05 for the Gaussian noise as being significant but not physically unreasonable. Asymptotically smaller values for the noise would lead to the results for deterministic measurements (for which rectification would not be needed), whereas larger values of the noise would lead to less reduction of variance in the rectified measurements.

Because of the relatively small identification data sets we used and the efficiency of the software, identification of the recurrent neural-network models employed for the examples presented here took only about 100 seconds of CPU time on an entry-level UNIX workstation. Similarly, rectification of the measurements of both the tank/pipeline and packed-column examples in the test data using the ERN-DEKF combination took a comparable amount of time using an interpreted simulation language (MATLAB, Registered Trademark of MathWorks, Inc.). Implementation of recurrent neural-network models, once identified, requires only simple function evaluations. The discrete Kalman filter requires a single matrix inversion at each sampling period. Both techniques can be easily implemented in an on-line environment.

Conclusions

The recurrent neural-network approaches to nonlinear, dynamic data rectification presented in this and previous work provide an attractive alternative to more traditional approaches. By examining recurrent neural networks from a state-space perspective, optimal estimation tools such as the Kalman filter can be easily used to improve neural-network performance in nonlinear, dynamic data rectification, and allow extension of these methods to situations involving correlated measurement errors and measurement bias.

Acknowledgment

This work was supported by NSF grant CTS-9216380.

Notation

$\boldsymbol{F}$ = Jacobian of state equation
$\boldsymbol{F}^*$ = Jacobian of augmented state equation
$\boldsymbol{G}^*$ = process noise matrix for augmented model
$\boldsymbol{H}$ = Jacobian of measurement equation
$\boldsymbol{H}$ = Jacobian of augmented measurement equation
z^{-1} = time delay

Literature Cited

Alouani, A., P. Xia, T. Rice, and W. D. Blair, "On the Optimality of Two-Stage State Estimation in the Presence of Random Bias," *IEEE Trans. Automat. Contr.*, **AC-38**, 1279 (1993).

Cheng, Y., T. W. Karjala, and D. M. Himmelblau, "The Identification of Nonlinear Dynamic Processes with Unknown and Variable Dead Time Using Internally Recurrent Networks," *Ind. Eng. Chem. Res.*, **34**(5), 1735 (1995).

Draper, N., "Straight Line Regression where Both Variables are Subject to Error," *Proc. Kansas State Univ. Conf. on Applied Statistics in Agriculture*, Manhattan, KS, p. 1 (1991).

Friedland, B., "Treatment of Bias in Recursive Filtering," *IEEE Trans. Automat. Contr.*, **AC-14**, 359 (1969).

Gelb, A., ed., *Applied Optimal Estimation*, MIT Press, Cambridge, MA (1974).

Gill, P. E., W. Murray, M. A. Saunders, and M. H. Wright, "User's Guide for NPSOL (Version 4.0): A Fortran Package for Nonlinear Programming," Tech. Rep., Systems Optimization Laboratory, Dept. of Operations Research, Stanford Univ., Stanford, CA (1986).

Hertz, J. A., A. S. Krogh, and R. G. Palmer, *Introduction to the Theory of Neural Computation*, Addison-Wesley, Reading, MA (1991).

Karjala, T. W., and D. M. Himmelblau, "Dynamic Data Rectification via Recurrent Neural Networks vs. Traditional Methods," *AIChE J.*, **40**(11), 1865 (1994a).

Karjala, T. W., and D. M. Himmelblau, "Dynamic Data Rectification Using the Extended Kalman Filter and Recurrent Neural Networks," *Proc. IEEE-ICNN*, Vol. V, Orlando, FL, p. 3244 (1994b).

Karjala, T. W., D. M. Himmelblau, and R. Miikkulainen, "Data Rectification using Recurrent (Elman) Neural Networks," *Proc. IEEE-IJCNN*, Vol. II, Baltimore, p. 901 (1992).

Kim, I.-W., M. J. Liebman, and T. F. Edgar, "Robust Error-in-Variables Estimation Using Nonlinear Programming Techniques," *AIChE J.*, **36**(7), 985 (1990).

Lewis, F. L., *Optimal Estimation with an Introduction to Stochastic Control Theory*, Wiley, New York (1986).

Ljung, L., *System Identification, Theory for the User*, Prentice-Hall, Englewood Cliffs, NJ (1987).

Luyben, W. L., *Process Modeling, Simulation and Control for Chemical Engineers*, McGraw-Hill, New York (1990).

MacMurray, J. C., "Modeling and Control of a Packed Distillation Column Using Artificial Neural Networks," Master's thesis, Univ. of Texas at Austin (1993).

Patwardhan, A. A., "Modeling and Control of a Packed Distillation Column," PhD Thesis, The Univ. of Texas at Austin (1991).

Patwardhan, A. A., and T. F. Edgar, "Nonlinear Model Predictive Control of a Packed Distillation Column," *Ind. Eng. Chem. Res.*, **32**(10), 2345 (1993).

Sage, A. P., and J. L. Melsa, *Estimation Theory with Applications to Communications and Control*, McGraw-Hill, New York (1971).

Söderström, T., "Identification of Stochastic Linear Systems in Presence of Input Noise," *Automatica*, **17**, 713 (1981).

Werbos, P. J., "Backpropagation Through Time: What it Does and How to Do It," *Proc. IEEE*, **78**(10), 1550 (1990).

Zhou, D., Y. X. Sun, Y. G. Xi, and Z. J. Zhang, "Extension of Friedland's Separate-Bias Estimation to Randomly Time-Varying Bias of Nonlinear Systems," *IEEE Trans. Automat. Contr.*, **AC-38**, 1270 (1993).

Appendix: Packed Distillation Column Model

Liquid-phase material balance for the light component:

$$A\rho_L\phi_L\frac{\partial x}{\partial t} = -L\frac{\partial x}{\partial z} + AK_y^0 a_m(y - y^*). \qquad \text{(A1)}$$

Vapor-phase material balance for the light component:

$$A\rho_V\phi_V\frac{\partial y}{\partial t} = V\frac{\partial y}{\partial z} - AK_y^0 a_m(y - y^*). \qquad \text{(A2)}$$

Vapor–liquid equilibrium relationship at the interface:

$$\alpha = \frac{y^*(1-x)}{x(1-y^*)}. \qquad \text{(A3)}$$

Material balance at the feed location:

$$\text{feed}_L + L_R = L_S \tag{A4}$$

$$\text{feed}_V + V_S = V_R \tag{A5}$$

$$\text{feed}_L xf_L + L_R x_{BR} = L_S x_{TS} \tag{A6}$$

$$\text{feed}_V xf_V + V_S y_{TS} = V_R y_{BR}. \tag{A7}$$

Energy balance at the feed location:

$$\text{feed}_L T_F + L_R T_{L_{BR}} = L_S T_{L_{TS}} \tag{A8}$$

$$\text{feed}_V T_F + V_S T_{V_{TS}} = V_R T_{V_{BR}}. \tag{A9}$$

Material balance around the overhead accumulator:

$$H_{\text{drum}}\,\rho_L \frac{dx_{TR}}{dt} = V_R(y_{TR} - xd) \tag{A10}$$

$$V_R = D + L_R. \tag{A11}$$

Material balance around the sump:

$$H_{\text{sump}}\,\rho_L \frac{dxb}{dt} = L_S(x_{BS} - xb). \tag{A12}$$

Material balance around the total reboiler:

$$H_{\text{reb}}\,\rho_L \frac{dx_{\text{reb}}}{dt} = V_S(xb - y_{BS}). \tag{A13}$$

Vapor–liquid equilibrium relationship for the reboiler:

$$\alpha = \frac{y_{BS}(1 - x_{\text{reb}})}{x_{\text{reb}}(1 - y_{BS})}. \tag{A14}$$

Choice of manipulated variables:

D = dist (distillate rate), or

L_R = rr (reflux rate), and

V_S = vbr (vapor boilup rate).

Manuscript received Dec. 15, 1994, and revision received Dec. 13, 1995.