Dynamic Data Rectification by Recurrent Neural Networks vs. Traditional Methods

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

Recurrent neural networks are used to demonstrate the dynamic data rectification of process measurements containing Gaussian noise. The performance of these networks is compared to the traditional extended Kalman filtering approach and to published results for model-based nonlinear programming techniques for data reconciliation. The recurrent network architecture is shown to provide comparable, if not superior, results when compared to traditional methods. The networks used were trained using conventional nonlinear programming techniques in a batch fashion.

Introduction

In a modern chemical plant, a wide variety of measurements of the process variables are collected for the purposes of cost accounting, process control, statistical quality control, and performance evaluation. These measurements are often contaminated in the sense that random noise and gross errors may be present due to measurement irreproducibility, instrument degradation, human error, or other unmeasured disturbances.

Rarely does the measured amount of raw material processed and energy furnished to a modern chemical plant or refinery exactly match the measured amount of product produced or energy consumed. Data rectification methods were first developed in order to reconcile inaccurate and inconsistent plant data using steady-state mass and energy balances as constraints. In the literature, the term *data reconciliation* is commonly used to refer to the reduction of additive random noise from process measurements by adjusting the measurements to fit a prescribed model usually via constrained least squares. We prefer the broader term *data rectification* which means removal/reduction of not only noise but other types of contaminants in the data such as biases and gross errors. This term includes *data reconciliation* as a subset but also includes filtering, and/or prediction, and/or hypothesis testing as well as other methods. A typical example of data *reconciliation* in the literature is to select a process model, add normally distributed random noise to the measured process inputs and outputs, and minimize the squares of the deviations between the simulated process measurements and the predicted (rectified) measurements at each time t by using the process model as a constraint in the least-squares calculation. In contrast, rectification methods do not necessarily adjust the data collected at time t to satisfy a model. In fact, in our work with artificial recurrent neural networks (RNN), we do not adjust data at time t to get the rectified values at time t as will be explained below.

We focus here on the removal of random errors via three techniques: (1) artificial neural networks, (2) extended Kalman filtering, and (3) constrained nonlinear programming. We first present some background on data rectification for dynamic processes, then explain how Kalman filtering, artificial neural networks, and nonlinear programming can be applied to the problem. Finally, we provide an example to help evaluate the three techniques.

Data Rectification

Statistical methods have been the mainstay of data rectification in the chemical process industries. Most research to date has centered on the detection and removal of gross errors and filtering of measurement noise from somewhat idealized steady-state plant data. Recently, a few investigators have turned their attention toward dynamic systems and serially correlated data with some success (Narasimhan and Mah, 1988a,b; Darouach and Zasadzinski, 1991; Liebman et al., 1992; Kao et al., 1992; Ramamurthi et al., 1993).

Data rectification is traditionally performed in two steps. First, nonrandom measurement errors must be detected and their presence eliminated. Then the 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. Unfortunately, accurate process models are not always available in practice.

Steady-State Processes

Most of the techniques developed for data rectification have been for steady-state material balances. In the absence of gross or other types of nonrandom errors, the goal of data rectification is to make adjustments to the measurements in order to satisfy the process constraints. One would like to minimize some criterion such as the norm of the error involved, or the absolute value of the error. Because the "true" values of these variables are not known, the reference value for which the error is calculated usually is the reconciled value of a variable. If the constraints are linear, the problem can be posed mathematically as:

$$\underset{\hat{y}}{\text{Minimize:}}\ (y-\hat{y})^T\Sigma^{-1}(y-\hat{y})$$

$$\text{Subject to:}\ A\hat{y}=b \tag{1}$$

where y is the vector of actual measurements, $\hat{y}$ is the vector of reconciled measurements, Σ^{-1} is the inverse of the covariance matrix of the measurement noise, and $A\hat{y}=b$ is a set of linear balance equations. This minimization problem has the closed form solution:

$$\hat{y}=y-\Sigma A^T(A\Sigma A^T)^{-1}(Ay-b) \tag{2}$$

but this equation is not valid unless gross errors have been detected and removed. Numerous authors have developed statistical tests to detect gross errors in linearly constrained systems (Tamhane and Mah, 1985; Serth and Heenan, 1986; Rosenberg et al., 1987; Narasimhan and Mah, 1989; Jongenelen et al., 1988) most of which are based on analysis of the constraint residuals. Unfortunately, these statistical tests are founded on statistical assumptions frequently violated by real plant data.

More recently, Tjoa and Biegler (1991) have sought to simultaneously detect and remove gross errors while reducing random measurement errors by using the maximum likelihood principle to generate a modified objective function. An analytic closed form solution is no longer possible with departure from the least-squares objective function. In addition, this method requires prior knowledge of the nature and distribution of both the random and nonrandom measurement errors; such information is generally not available in real data.

If measurements are missing, Crowe et al. (1983, 1992) have demonstrated how unmeasured variables can be eliminated from the linear balance equations via matrix projection to form a reduced set of balance equations. If the constraints of problem 1 are nonlinear, such as the constraints that arise when the measurements for both a concentration and flow rate must be rectified for the same process stream, or if the objective function is not quadratic, nonlinear programming techniques must be applied for data rectification (after gross errors are deleted).

Dynamic Processes

Much less has been published for dynamic processes than for steady-state rectification. Table 1 lists most of the recent literature treating the problem of the rectification of data for dynamic processes. Even processes that are characterized as "steady state" in practice continually fluctuate about a nominal steady-state value. One would expect that data rectification techniques that account for process dynamics would show better performance than algorithms based on the assumption of steady state, even for "steady-state" processes. Almost all work in rectification for dynamic processes falls either within the general categories of filtering methods or iterative, model-based estimation via nonlinear programming techniques.

Table 1. Data Rectification for Dynamic Systems

Authors	Model Based	L or NL System*	Gross Error Incl.	Error Covar. Est.
Adams (1988)	Yes	L	No	Yes
Almasy (1990)	Yes	L	No	Yes
Barash (1987)	Yes	L	No	Yes
Cameron et al. (1991)	Yes	NL	No	Yes
Darouach and Zasadzinski (1991)	Yes	L	No	Yes
Gertler and Almasy (1973)	Yes	L	No	Yes
Gertler and Singer (1989)	Yes	L	No	Yes
Kao et al. (1992)	Yes	L	Yes	Yes
Karjala et al. (1992)	No	NL	No	No
Liebman et al. (1992)	Yes	NL	No	No
Miyakoda and Talagrand (1971)	Yes	NL	No	Yes
Narasimhan and Mah (1988a)	Yes	L	Yes	Yes
Ragot et al. (1992)	Yes	L	No	Yes
Ramamurthi et al. (1993)	Yes	NL	No	?
Stanley (1982)	Yes	L	No	No
Vaclavek et al. (1980)	Yes	L	No	Yes
Wolf et al. (1990)	Yes	L	No	Yes

*L = linear; NL = nonlinear

Kalman Filtering Methods

Filtering is one way to estimate the best values of process variables and their confidence limits. The literature in the field is enormous, and several excellent textbooks have been written (Sage and Melsa, 1971; Gelb, 1974; Lewis, 1986). Although the majority of these publications have been related to the Kalman-Bucy filter, considerable theoretical attention has been devoted to other concepts as well. In general, the goal of filtering has been to make minimum variance estimates that satisfy the dynamic model equations, provide estimates of both the input and output variables for the process, and minimize a performance index such as a cost function or the variance of the estimation error for all of the variables (appropriately weighted). However, such estimators (filters) are based on specific discrete, or sometimes continuous, linear models of the physical system that are often restricted in nature to be lumped parameter and time invariant with additive random noise that is stationary with zero mean and often only white (uncorrelated in time) noise. Furthermore, it is often assumed that the covariance matrices of the noises are known. Algorithms appropriate for slight deviations from these basic concepts have been published, but the results are still quite limited.

The most commonly used filtering algorithm in chemical engineering is the recursive estimation scheme known as the extended Kalman filter (Sorenson, 1985). Kalman filtering methods suppose full knowledge of the process dynamics. The goal is to make minimum variance, unbiased estimates of the

process states, $x(t)$, that satisfy (for a continuous model) the dynamic model equation:

$$\dot{x} = f[x(t), u(t), t] + w(t) \tag{3}$$

where u is a vector of process or control inputs, t is time, w_t is a vector of process noises, and f is a nonlinear vector valued function. The measured variables, y_t, are related to the process states by the discrete measurement equation:

$$y_t = h[x_t] + v_t \tag{4}$$

where h is a vector valued nonlinear function and v_t is a vector of measurement noises.

A number of dynamic rectification techniques are based on variations of the Kalman or extended Kalman filter (Darouach and Zasadzinski, 1991; Almasy, 1990; Barash, 1987). Kalman and extended Kalman filtering require a linear model or a locally linearized model, and modeling errors can result in significant bias. Kalman filters require parameter tuning and prior knowledge of the covariance matrices of the measurement and process noise vectors, but are computationally efficient and can be implemented in an on-line environment. However, there is reason to believe that the accuracy of a recurrent net in data rectification exceeds that of a Kalman filter when the assumptions underlying the filter are violated.

Nonlinear Programming Techniques

Nonlinear programming (NLP) techniques (Liebman and Edgar, 1990; Jang et al., 1986; Kim et al. 1990; Liebman et al., 1992) are essentially extensions of the methods for steady-state process rectification mentioned above. The steady-state data rectification problem of problem 1 is recast as:

$$\begin{aligned} &\text{Minimize: } \Phi(y_t, \hat{y}_t, y_{t-1}, \hat{y}_{t-1}, \ldots) \\ &\hat{y} \\ &\text{Subject to: } f(\dot{x}_t, x_t, \mathbf{u}_t, t) = 0 \\ &\qquad\qquad h(x_t, t) = 0 \\ &\qquad\qquad g(x_t, t) \geq 0 \end{aligned} \tag{5}$$

where Φ is a generalized objective function (most often the sum of squares), x_t is a vector of state variables at time t (not all of which are measured), y_t and $\hat{y}_t$ are the measurements and rectified measurements, f is the dynamic process model, g is a vector of inequality constraints (including bounds on the variables), and h is a vector of known equality constraints. The model constraint equations f are typically expressed as dynamic differential equations that are solved via orthogonal collocation on finite elements. Minimization problem 5 is often performed using a moving window of past measurements. This window must be long enough to capture relevant process dynamics but brief enough to keep the computation load for the NLP problem tractable. The computation time required for the NLP solution method for dynamic data rectification can be significant, and, generally speaking, is much longer than for Kalman filtering.

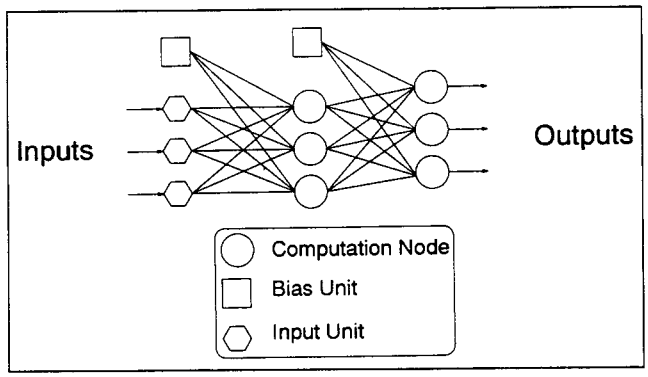

Figure 1. Standard feedforward architecture.

Neural Network Approach

Neural network methods have been proposed as useful tools for process modeling, diagnosis, and control by a number of researchers (Hoskins and Himmelblau, 1988; Watanabe et al., 1989; Bhat and McAvoy, 1989, 1990a,b; Kramer and Leonard, 1990; Hernandéz and Arkun, 1990; Psichogios and Ungar, 1990, 1992; Su and McAvoy, 1991, 1992; Qin et al., 1992). The strength of neural networks methods stems from their pattern recognition capability and role as nonparametric models. One of the main attractions of these methods is the potential to rapidly and easily perform nonlinear system identification solely on the basis of historical data.

Feedforward neural networks

The conventional artificial neural network architecture used by the majority of researchers is depicted in Figure 1. Signals "feed forward" in the sense that the inputs feed forward from the input nodes to the hidden nodes via weighted connections. In a similar fashion, the signals from the hidden (intermediate) nodes feed forward to the output nodes. Computation via a transfer function occurs only in the hidden and output nodes; the transfer function in the input units is unity if the inputs themselves are properly scaled. The bias nodes have a constant output of 1.0. The weights of a network contain representational information and are the decision variables adjusted during training.

Feedforward networks have traditionally been trained using an algorithm known as backpropagation of error that has been discussed at length in many sources (Hertz et al., 1991). Back propagation is essentially a first-order, gradient descent based optimization technique that adjusts the network weights (trains the net) to minimize the mean-squared error between the network outputs and target outputs for a set of training data, typically sequentially on an input-output pattern pair basis. A fixed step size in the direction of a linear combination of the current negative gradient and the past search directions is taken at each step. Like all first-order methods, back propagation is a significantly less efficient optimization method than are second-order methods such as the conjugate gradient or quasi-Newton algorithm. This has been demonstrated for neural networks by Watrous (1988).

The feedforward networks discussed above perform a static nonlinear mapping of the input space to the output space. This mapping may be adequate for pattern recognition and steady-

state prediction, but for dynamic process modeling and prediction it is necessary to either explicitly or implicitly represent time in a network. Many neural network researchers have chosen to use a moving window of input values introduced into a standard feedforward network (Bhat and McAvoy, 1989, 1990a; Ungar et al., 1990). The past values for the last n time steps of the m input variables are used as the vector of network inputs resulting in $n \times m$ input nodes and a corresponding increase in the number of weights. The outputs of such a network are typically the predicted variables of interest at the next time step. In general, the size of the data window must be determined by trial and error (each variable in a multivariate process might well have its own optimal data window).

Often researchers add the network outputs from the previous time step to the input window vector. Networks of this type are often called externally recurrent but have the same feedforward structure described above and are trained using the same techniques. Networks of this type have shown better performance than purely feedforward nets when used for system identification of chemical processes (Su and McAvoy, 1991, 1992; Qin et al., 1992).

Internally recurrent networks

An alternative type of net that can be used to model dynamic process is an internally recurrent (or recursive) network. These networks are characterized by internal feedback as well as feedforward connections between processing units. The presence of feedback connections provides the network with internal states and a form of memory. The output of such networks no longer depends simply upon the current input but also on the past outputs.

Figure 2 shows an example of a simple internally recurrent network architecture. This specific architecture is often referred to as the Elman architecture after its originator (Elman, 1990). An Elman net is similar to the standard feedforward architecture with layers of input, hidden, and output units, but also includes a set of *context units* that save the prior output of the hidden units and feed back their outputs as inputs to the hidden units on each successive calculation cycle.

In contrast to the standard feedforward or externally recurrent network, an internally recurrent network can be much smaller in size and use fewer parameters (weights). A moving window of vectors of process variables at times $\{t, t-1, t-2, \ldots, t-n\}$ is not used as input to the net. Instead, a single vector of process variables at time t is used to predict the process variables at time $t+1$. Recurrent networks have been demonstrated to be capable of learning process dynamics in the presence of noise and making significant reductions in the noise level of their predictions given noisy inputs (Karjala et al., 1992). Comparisons between externally recurrent networks and feedforward networks used for dynamic modeling have indicated that externally recurrent networks are less sensitive to noise and give better predictions (Qin et al., 1992). Internally recurrent networks can be expected to give comparable if not superior performance due to their smaller parameter count and the cumulation of information somewhat analogous to integration.

The presence of feedback connections in recurrent networks

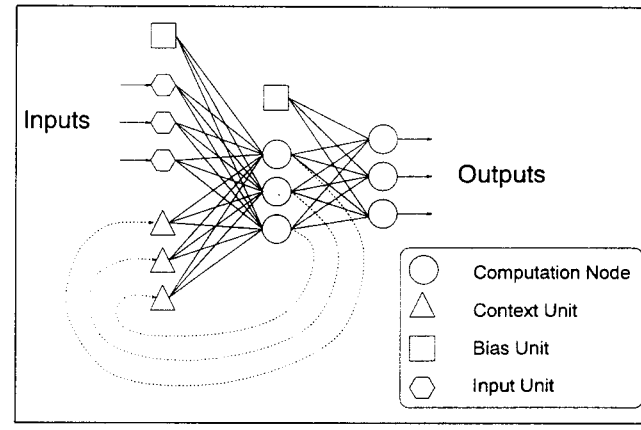

Figure 2. Simple recurrent network (Elman, 1990).

complicates the calculation of error gradients which in turn complicates estimation of the network weights. The derivatives of the network error with respect to a given weight depend not only on the current network input vector but also on all of the previous network inputs. This feature complicates the formulation of the back-propagation algorithm favored by many researchers. A modified algorithm, known as *back-propagation through time or unfolding in time*, is a particularly efficient algorithm for calculation of the error gradient (Hertz et al., 1991). A detailed discussion including the concept of "ordered derivatives" can be found in Werbos (1990). A large and growing body of work is devoted to the training and use of recurrent networks (Cleeremans et al., 1989; Gherrity, 1989; Pearlmutter, 1989, 1990; Pineda, 1987; Watrous et al., 1988; Williams and Peng, 1990; Williams and Ziper, 1989a,b).

Recurrent artificial neural networks with coefficients trained nonsynchronously are models that can represent the dynamics of complex processes. They serve the same role in process modeling as do first principles models, state space models, time-series models, transfer functions, and so on. All are approximations of the process operating characteristics of interest, and all involve coefficients whose values are estimated from process input-output data. However, artificial neural networks are nonparametric models, that is, the form of the model is constructed along with the estimation of the values of the coefficients, whereas with first principles models the form is prescribed prior to estimating the values of the coefficients. Because of the large number of coefficients and the more complex structure of an artificial neural network model (which is composed of basis functions), the net frequently can model a process more faithfully in the region of the given data (by interpolation) than can a first principles model, or a state-space model such as used in Kalman filtering or nonlinear programming. The prospect exists that an artificial neural network can rectify process data satisfactorily in the sense that the data fit a good model of the process during the optimization for rectification. Thus, as long as any model is of assistance in rectifying data, artificial neural networks can serve the same purpose.

One of the disadvantages of artificial neural networks as models is due to the substantial number of interconnections between nodes, each one of which has an associated coefficient. Considerably more data is needed to estimate the values of the

coefficients in a net than would be for the coefficients in a first principles model. To build a model that can predict the response of a dynamic process, representative data must be used from over the entire operating region so that an artificial neural network can interpolate in making predictions rather than extrapolate in making predictions of the process response in the desired regions of operation. A first principles model can probably do a better job of extrapolation than an artificial neural network, but even for a first principles model the data has to be collected properly to get unbiased estimates of the coefficients. The problem is if the net has to extrapolate, the predictive ability of the net may degrade unacceptably. Consequently, large amounts of data are needed to make sure that this does not occur.

Data rectification via neural networks

Very little has been published on this topic to date. Kramer has proposed the use of "autoassociative neural networks" (Kramer, 1991, 1992) for the purposes of sensor noise reduction and sensor validation. These networks are essentially two ordinary feedforward networks chained together with an intermediate "bottleneck" layer that accepts as input the output vector of the first network, and transmits its output into the input layer of the second network. This bottleneck is of reduced dimensionality with respect to the network inputs and outputs and thus implements data compression. The first network operates as a nonlinear principal component taxonomy for the input vector that in turn is reconverted to a vector of the original dimension of the inputs by the second network. In this operation, the measurement noise is reduced. Methods were proposed to detect and eliminate gross errors as well (Kramer, 1992). These techniques were applied to the data rectification of tray temperatures for a steady-state distillation column, and the approach worked well for that example. It is not clear whether these methods would be as effective when applied to less homogeneous measurements or to dynamic processes.

Karjala et al. (1992) were the first to propose the use of recurrent neural networks for dynamic data rectification. Gaussian noise was successfully filtered from simulated process measurements from a draining tank by training the network to make one step ahead predictions. The essential idea underlying the technique is that only the deterministic portion of the measurements y_t is modeled because the measurement errors from t to $t+1$ are uncorrelated. In contrast to traditional methods, this technique does not utilize the measurements y_t to estimate the rectified value $\hat{y}_t$. If the true values were available, they could be used as targets during training and the recurrent neural network could estimate $\hat{y}_t$ directly from y_t and the internal states of the network. But, because the true values of process measurements are not available for neural network training, you cannot train neural networks using the measurement y_t as both the network input and target vector. The result would be the trivial identity mapping because the strong (1-to-1) correlation of the measurement noise prevents the network from distinguishing between deterministic and random portions of the measurements. The "autoassociative" neural networks of Kramer are trained in this fashion but are prevented from learning the identity mapping by the "bottleneck" layer. Such "bottleneck" neural networks would not perform well for the dynamic, nonhomogeneous process measurements under consideration in this article. Our results obtained through one-step-ahead prediction, however, compare quite favorably to more complex data rectification methods which utilize current information when estimating $\hat{y}_t$ as demonstrated below.

Artificial neural networks accommodate zero ensemble mean random noise if the noise is uncorrelated. The noise acts as a regularization term added to the usual objective function that smoothes the function approximation. Random noise specifically functions as an additive norm of the first derivatives. If gross errors are defined as intermittent large but short duration deviations from the true measurement, then if they have zero ensemble mean, gross errors have the same effect as random noise. However, if a substantial fraction of the time record includes gross errors, and they do not have zero mean, then in addition to their smoothing influence a bias in the rectified values can occur just as if biased measurements were used to train a net. If a net is trained with unbiased targets, and significant biased measurements occur, the deviation between the prediction by the net and the measurements will signal that the measurement is biased. How to correct such measurements to obtain unbiased rectified values is beyond the scope of this work, but certainly because of analytical redundancy in the modeling, the predicted value of the measurement might be preferred.

Example Problem

For an engineer to use a novel technique in data rectification, he or she must have confidence that the method is valid and reliable. To build trust in the use of internally recurrent artificial neural networks for data rectification and to demonstrate their data rectification capabilities, we have applied them to a popular test problem first used by Seinfeld (1970) to demonstrate the benefit of extended Kalman filtering in optimal control. Simulations are used because only by knowing the "true" values for the process variables can an accurate evaluation be made of any proposed rectification method. The test problem consisted of a continuous stirred tank reactor (CSTR) with a first-order exothermic reaction and heat removal by a coil or jacket. Jang et al. (1986) used this example to compare extended Kalman filtering to a nonlinear programming approach for state and parameter estimation. They integrated the model equations over a measurement window. In a similar approach, Liebman et al. (1992) used nonlinear programming constrained by model equations which were solved by orthogonal collocation over a measurement window to estimate the process states.

The state equations for this system are:

$$\frac{dT}{dt}=\frac{q}{V}(T_0-T)-\frac{\Delta H_r}{\rho C_p}k_0C\exp\left(\frac{-E_a}{T}\right)-\frac{UA_r}{\rho C_pV}(T-T_c) \quad (6)$$

$$\frac{dC}{dt}=\frac{q}{V}(C_0-C)-k_0C\exp\left(\frac{-E_a}{T}\right) \quad (7)$$

For consistency of comparison with Liebman (1992), the temperatures and concentrations were scaled with reference values T_r and C_r to yield:

Table 2. CSTR Model Parameters

Parameter	Value	Units
q	10.0	$cm^3 \cdot s^{-1}$
V	1,000.0	cm^3
ΔH_r	−27,000.0	$cal \cdot mol^{-1}$
ρ	0.001	$g \cdot cm^{-3}$
C_p	1.0	$cal\ (g \cdot K)^{-1}$
U	5.0×10^{-4}	$cal\ (cm^2 \cdot s \cdot K)^{-1}$
A_r	10.0	cm^2
T_c	340.0	K
k_0	7.86×10^{12}	s^{-1}
E_a	14,090.0	K
C_r	1.0×10^{-6}	$mol \cdot cm^{-3}$
T_r	100.0	K

$$\frac{dT'}{dt} = \frac{q}{V}(T_0' - T') - \frac{\Delta H_r}{\rho C_p T_r} k_0 C' \exp\left(\frac{-E_a}{T'}\right)$$

$$- \frac{UA_r}{\rho C_p V}(T' - T_c') \quad (8)$$

$$\frac{dC'}{dt} = \frac{q}{V}(C_0' - C') - k_0 C' \exp\left(\frac{-E_a}{T}\right) \quad (9)$$

Table 2 lists the values of the parameters used in these models. The above equations were integrated through time in order to generate data that were used to compare the performance of a simple recurrent network and an extended Kalman filter with some literature results. As in Liebman et al. (1992), the simulated measurements consisted of the state variables T' and C' and the inputs T_0' and C_0' sampled every 2.5 s. All four measurements were corrupted by zero mean Gaussian measurement noise having a standard deviation of 0.05. Nonzero mean noise and autocorrelated noise both yield biased results for rectification as might be expected, and add a higher order of complexity to the rectification problem. These types of noise are not dealt with in this article. The recurrent network and extended Kalman filter were tested on the same examples as in Liebman et al. (1992). In some instances at low values of concentration, because of the added noise, the measured values of the concentration became negative, but these values were not suppressed in the simulations to avoid censoring the data.

Recurrent Neural Network

A simple recurrent network of the type described earlier was trained as a one step ahead predictor. Given the input measurement vector $[C'\ T'\ C_0'\ T_0']^T$ at time $t-1$, the measurement vector at the next time step was used as the target during training. Training of the net was accomplished using a standard nonlinear programming code in a batch fashion. The network consisted of four input nodes, four hidden nodes, four context nodes, and four output nodes. The number of input and output nodes is dictated by the size of the measurement vector. Four hidden and context nodes were the minimum number required for this four variable problem. In addition there was a bias unit for the hidden layer and input layer. This arrangement resulted in a total of 56 adjustable parameters.

The training set was created by integrating the response of the model Eqs. 8 and 9 in response to random step changes in both input variables within the region of interest. In this case T_0' ranged from 6.0 to 9.0 and C_0' ranged from 3.4 to 3.6. The training set was 12,824 time steps in length corresponding to 89 hours of simulated data and contained 96 step changes. Both the input vectors and target output vectors contained Gaussian noise with a standard deviation of 0.05. Training times for networks of this type and for this training set ranged from 90 to 150 minutes on a 33 Mhz Sun workstation.

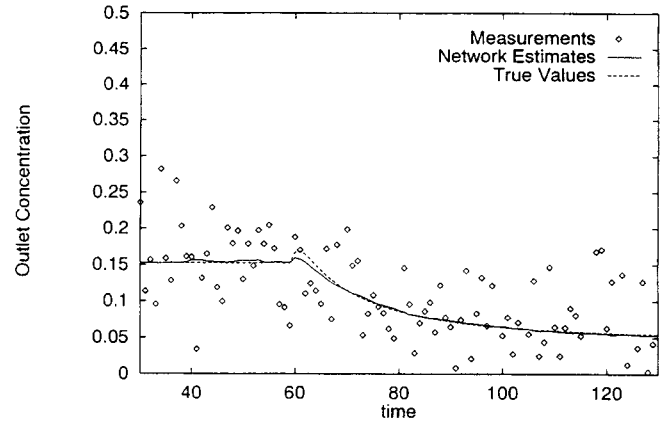

Figure 3. Recurrent network rectification of the outlet concentration.

The performance of the recurrent network was tested for the same example as in Liebman et al. (1992) and for the same variables. The simulation began at a steady-state operating point where $C_0' = 6.5$, $T_0' = 3.5$, $C' = 0.15$, and $T' = 4.6$. In order to build up the correct internal representation of the process within the net, the initial measurement vector was repeatedly used as input to the net until the internal states of the network (the context nodes) and hence the outputs stabilized. This procedure is necessary because just as in a recursive filter or data window based estimation procedure, additional data points beyond the initial measurement at $t = 0$ are needed to obtain valid output since the outputs of a recurrent net depend on more than just the current point. Subsequent measurement vectors were then presented sequentially as usual. At time step 30, the feed concentration was stepped from 6.5 to 7.5. Figure 3 shows the estimates by the net of the CSTR outlet

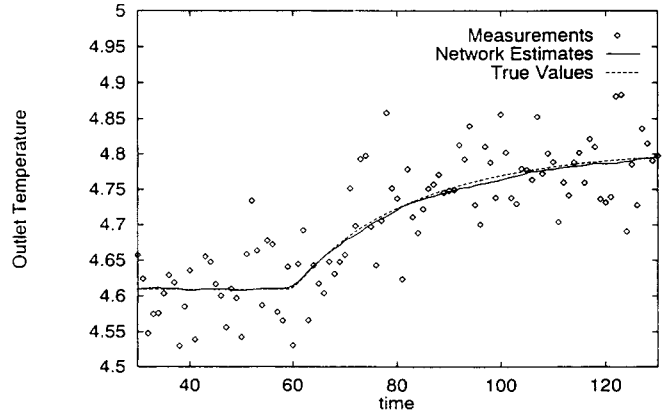

Figure 4. Recurrent network rectification of the outlet temperature.

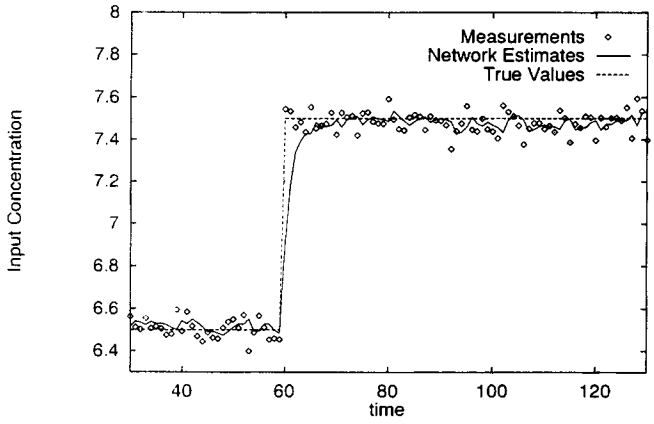

Figure 5. Recurrent network rectification of the inlet concentration.

concentration, the true values, and the unreconciled original measurements. Figures 4, 5, and 6 show the network estimates, true values, and unreconciled measurements for the CSTR outlet temperature, inlet concentration, and inlet temperature, respectively.

Extended Kalman Filter

An extended Kalman Filter (EKF) was also designed and was tested for the same example. The two process states, C' and T', were augmented by the process inputs C_0' to T_0' to form the state vector $\boldsymbol{x}$. The process inputs are assumed constant ($\partial C_0'/\partial t = 0$; $\partial T_0'/\partial t = 0$).

The process and measurement models consisted of:

$$\dot{\boldsymbol{x}} = \boldsymbol{f}(\boldsymbol{x}, t) + \boldsymbol{w} \tag{10}$$

$$\boldsymbol{y}_t = \boldsymbol{x}_t + \boldsymbol{v}_t \tag{11}$$

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

where $\boldsymbol{f}$ is a vector of nonlinear equations with elements consisting of Eqs. 8 and 9 in the first and second elements and zeros in the third and fourth elements, $\boldsymbol{w}$ is a vector of process noise, $\boldsymbol{y}_t$ is a vector of actual measurements, and $\boldsymbol{v}_t$ is a vector of the measurement errors. From time $t-1$ to time t, the process states and error covariance matrix $\boldsymbol{P}_t$ were first estimated by integrating the following process and error covariance update equations:

$$\dot{\hat{\boldsymbol{x}}} = \boldsymbol{f}(\hat{\boldsymbol{x}}, t) \tag{12}$$

$$\dot{\boldsymbol{P}} = \boldsymbol{F}(\hat{\boldsymbol{x}}, t)\boldsymbol{P} + \boldsymbol{P}\boldsymbol{F}^T(\hat{\boldsymbol{x}}, t) + \boldsymbol{Q} \tag{13}$$

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

over the interval Δt to yield the predicted $\hat{\boldsymbol{x}}_t^-$ and $\boldsymbol{P}_t^-$ at time t. (The superscript "$-$" denotes quantities obtained in the prediction phase of the Kalman filter (Eqs. 12-14). These quantities are then updated using Eqs. 15-17 to yield the final estimates at time t.)

Then, given the measurements $\boldsymbol{y}_t$ at the new time t, the updated state estimates and covariance matrix were obtained from:

$$\hat{\boldsymbol{x}}_t = \hat{\boldsymbol{x}}_t^- + \boldsymbol{K}_t[\boldsymbol{y}_t - \hat{\boldsymbol{x}}_t^-] \tag{15}$$

$$\boldsymbol{P}_t = [\boldsymbol{I} - \boldsymbol{K}_t]\boldsymbol{P}_t^- \tag{16}$$

$$\boldsymbol{K}_t = \boldsymbol{P}_t^-[\boldsymbol{P}_t^- + \boldsymbol{R}_t]^{-1} \tag{17}$$

where the third equation is the equation for the Kalman gain, $\boldsymbol{K}_t$. These equations have been simplified from the usual EKF equations because all of the process states are measured variables in this example.

Tuning of the EKF proved simple. The elements of the two matrices $\boldsymbol{Q}$ and $\boldsymbol{R}$ must be specified in order to balance the speed of response of the filter with the precision of the estimates. Here known error variances were used so that $\boldsymbol{R} = (0.0025)\boldsymbol{I}$. In practice, the covariance matrix of the measurement errors, $\boldsymbol{Q}$, can be estimated from a steady-state operating region. As the exact model was known for the simulated test data and there was no process noise, the diagonal elements of the process noise covariance matrix were set to small numbers, $\boldsymbol{Q} = \text{diag}([0.00001\ 0.0001\ 0.00001\ 0.00001]^T)$ to balance noise reduction with reasonable tracking of EKF state variables.

Figures 7-10 show the performance of the EKF on the test problem. The performance of the extended Kalman filter was similar to that of the recurrent neural network.

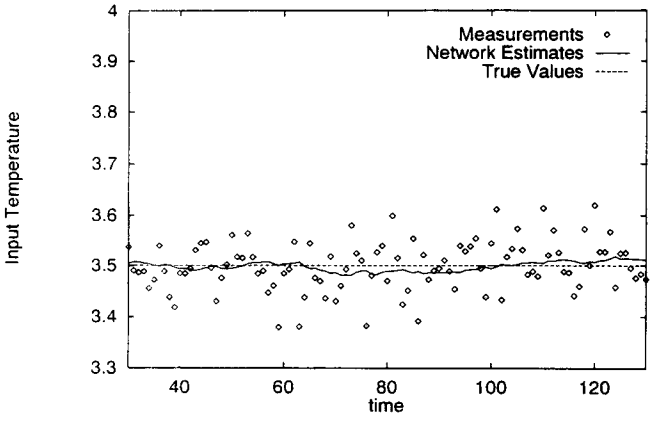

Figure 6. Recurrent network rectification of the inlet temperature.

Discussion of Results

Table 3 lists the standard deviations of the measurement errors before rectification and after rectification by the recurrent neural network and extended Kalman filter as well as some results from Liebman et al. (1992) who used NLP methods in this example. Table 4 shows the reduction in measurement error expressed as percentages. Note that both the RNN and EKF achieved greater reduction in measurement errors than did the NLP approach for the variables C', T', and T_0'. All three methods yielded an increase in standard deviation for the variable C_0' due to an inability to track exactly a step change in the feed concentration, but the RNN and EKF ex-

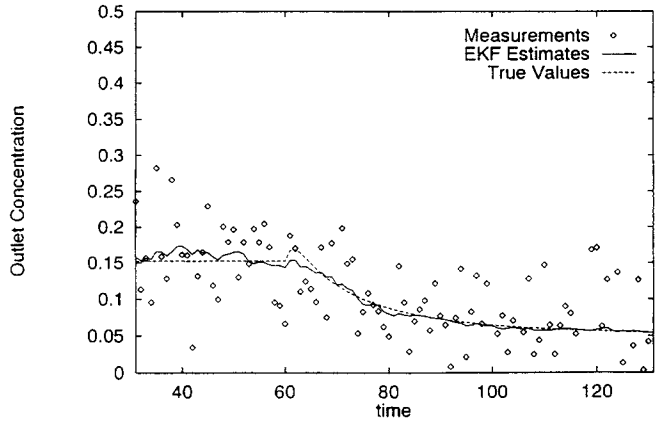

Figure 7. Extended Kalman filtering of the outlet concentration.

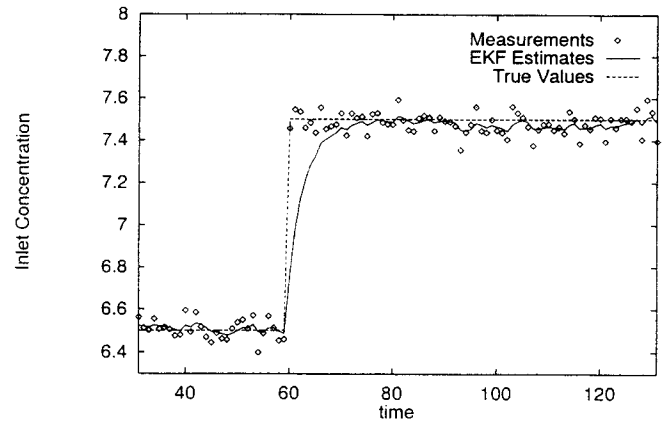

Figure 9. Extended Kalman filtering of the inlet concentration.

hibited smaller increases due to better tracking of the step change.

The recurrent network in general made smoother estimates of the process measurements as seen in Figures 3, 4, and 6 than did the EKF (Figures 7, 8 and 10). For rectification of the input concentration shown in Figures 5 and 9, the EKF seems to provide smoother estimates although the RNN exhibits faster response time to the step changes. Of course, the elements in the matrices $\boldsymbol{Q}$ and $\boldsymbol{R}$ in the EKF can be changed to give faster but rougher response. The fact that the results of the EKF and recurrent neural network are similar shows that the recurrent net did a good job of modeling the process specified by the set of differential Eqs. 8 and 9. The exact model was used in the EKF hence no model mismatch occurred, and the EKF simply had to filter out the measurement noise.

Model mismatch

To illustrate the dependency of model based data rectification on having an accurate process model, two runs of the extended Kalman filter were performed using biased models. Suppose that the activation energy E_a is estimated from process data and assume that the estimate deviates from the true value by $\pm 5\%$. The results of rectification by the Kalman filter are illustrated for the variables C' and T' for two cases, one in

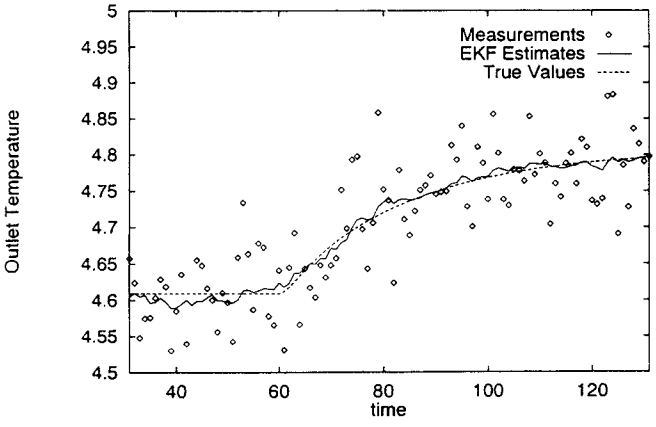

Figure 8. Extended Kalman filtering of the outlet temperature.

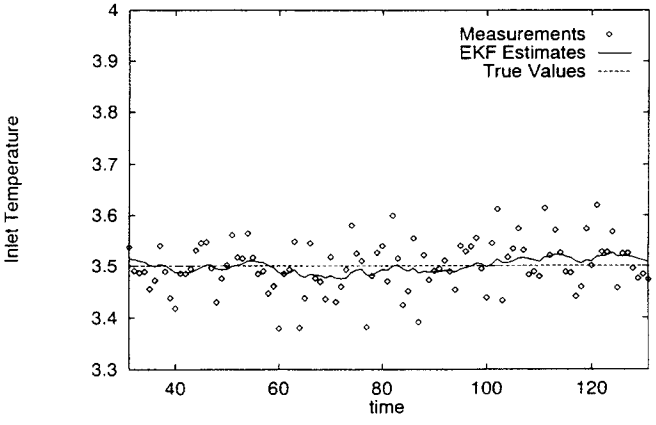

Figure 10. Extended Kalman filtering of the inlet temperature.

Table 3. Standard Deviations Before and After Rectification

	Present Work			Liebman et al. (1992)	
Variable	Before Rectification	RNN After Rectificaion	EKF After Rectification	Before Rectification	NLP After Rectification
C'	0.0515	0.00234	0.0055	0.0559	0.0068
T'	0.0486	0.00327	0.0078	0.0546	0.0125
C_0'	0.0498	0.05508	0.0740	0.0484	0.1757
T_0'	0.0507	0.00945	0.0117	0.0536	0.0184

Table 4. Percent Change in Measurement Error

	Present Work		Liebman et al. (1992)
Variable	RNN Method	EKF Method	NLP Method
C'	−95.5	−89.3	−87.8
T'	−93.3	−84.0	−77.1
C_0'	+10.6	+48.6	+263.0
T_0'	−81.4	−76.9	−65.7

which the activation energy E_a is biased upward by five percent (Figures 12 and 11) and one in which E_a is biased downward by five percent (Figures 14 and 13). Model mismatch significantly affects the accuracy of the Kalman filter which yields biased estimates of the outlet temperature and concentration. The EKF estimates of the process inputs, C_0' and T_0', are not significantly affected by model mismatch because they are assumed constant by the EKF on each iteration. We did not simulate the effect of model mismatch on rectification via nonlinear programming as it was beyond the scope of this article, but similarly biased estimates of the variables can be expected.

One-step-ahead prediction

As mentioned previously, the recurrent neural networks used in this research were trained as one-step-ahead predictors. Because this is a prediction approach, y_t was not used to estimate $\hat{y}_t$. Our procedure requires adequate data to train the net prior to it being placed in service, far more data than required to estimate the coefficients in a first principles model. But once trained, a net provides values of the rectified variables much faster than traditional methods. Although current data from t was not used to estimate $\hat{y}_t$ we can state:

(1) The RNN represents the input-output relation of the process faithfully in the domain in which data are available.

(2) The RNN predicts the inputs as well as the outputs reasonably well.

(3) In our simulations, the rectified measurements generated by the approximate RNN model had error less than or equal to the error in the rectified values generated by the NLP-based data *reconciliation* procedure, even when that procedure utilized the exact process model.

The performance of rectification methods based on prediction might be expected to degrade as the sampling interval increases. However, both extended Kalman filtering and dynamic data reconciliation methods will also experience degraded performance. In practice, on-line process measurements are usually collected in measurement intervals that are short compared to the time constants of the process dynamics. If the sampling rate for a given variable is high, one might be tempted to use a moving average to filter noise from the signal. This type of procedure might provide adequate estimates of $\hat{y}_t$ for steady-state data but moving averages would provide biased, lagged estimates for dynamic data. Predictive methods such as the RNN method outlined in this article anticipate the process trajectory and ameliorate this difficulty.

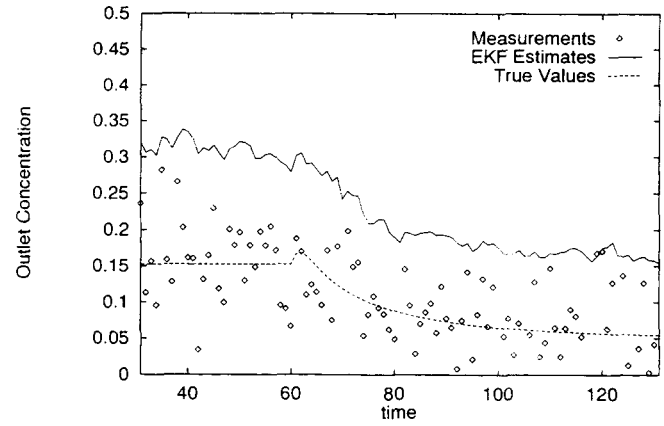

Figure 12. Extended Kalman filtering of the outlet concentration when E_a is biased upward by 5.0%.

Conclusions

Both the extended Kalman filter approach demonstrated here and the NLP approaches proposed by earlier researchers require an exact process model if model mismatch is not to contaminate the rectification. The results obtained for both of these techniques are no doubt better than would be achieved for actual process data because of the absence of model mismatch. In addition, both approaches utilize exact knowledge

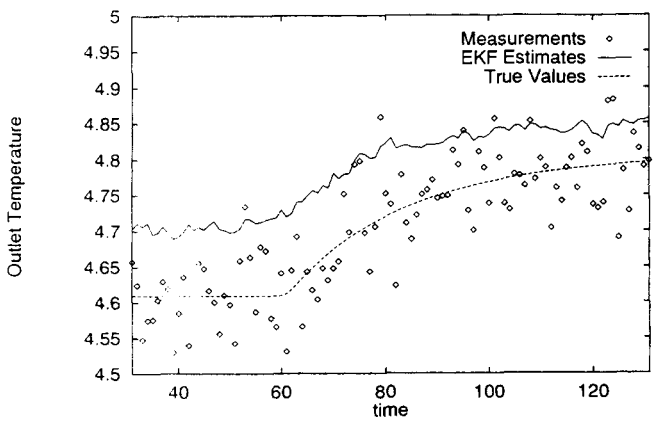

Figure 11. Extended Kalman filtering of the Kalman temperature when E_a is biased upward by 5.0%.

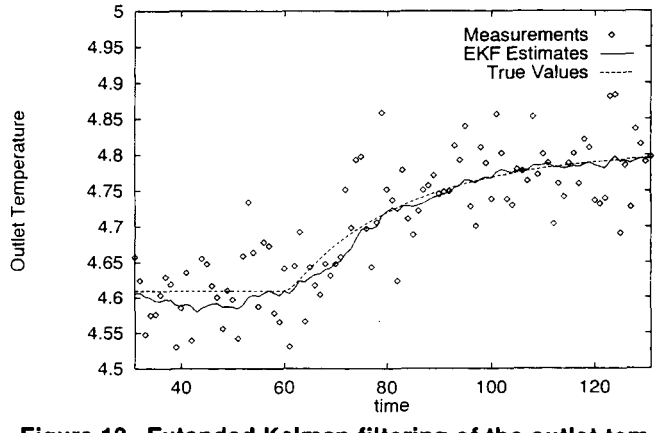

Figure 13. Extended Kalman filtering of the outlet temperature when E_a is biased downward by 5.0%.

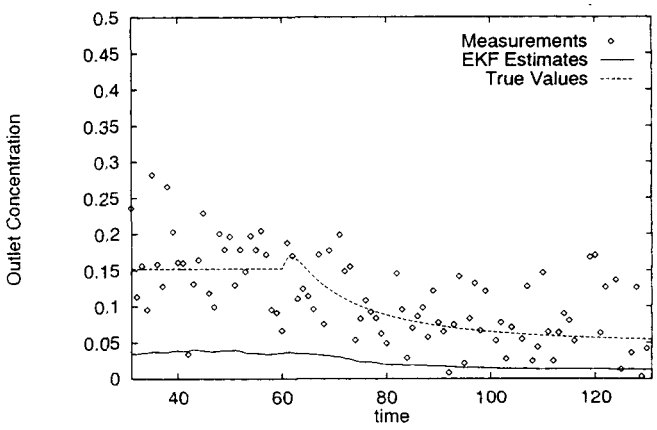

Figure 14. Extended Kalman filtering of the outlet concentration when E_a is biased downward by 5.0%.

of the covariance matrix of the measurement errors for the CSTR example. For real processes, the covariance matrix is normally not known and may be difficult to estimate. One might expect then that the performance of both methods would degrade when applied to measurements from real process data. The recurrent neural network approach, being a nonparametric modeling method, has the advantage that a prespecified process model and definitive knowledge of the measurement error statistics is not required.

Another important consideration in the comparison of data rectification methods is ease of implementation. The various iterative NLP methods of data rectification that use integration through time or orthogonal collocation are mathematically complex and may not be easily understood by plant engineers and operators. As a solution of a nonlinear program is required at each time step, these approaches can be computationally intensive to the extent that their use might be prohibitive in an on-line environment. Extended Kalman filtering is also a complex algorithm, but only a single matrix inversion is required at each time step and it can be easily implemented on-line. A recurrent neural network used for rectification could be implemented on-line in even the simplest hardware because only function evaluations and looping are required.

On the other hand, neural networks for data rectification require ample process data in the operating region of interest before they can be implemented. If process conditions change, a new net must be trained. Recurrent network training would best be accomplished off-line in batch mode using relevant portions of the operating record. The use of nonlinear programming for training makes it possible to sidestep some of the difficulties other researchers have experienced when using backpropagation in calculating the values of the weights.

Acknowledgment

This work was partially supported by NSF grant CTS-9216380.

Notation

A = state transition matrix
A = coefficient matrix in balance equations
A_r = heat-transfer area, cm^2
b = vector of constants
B = input matrix
C = concentration in reactor, $mol \cdot cm^{-3}$
C_0 = feed concentration, $mol \cdot cm^{-3}$
C_p = heat capacity of reactor contents, $cal\ (g \cdot K)^{-1}$
C_r = reference concentration, $mol \cdot cm^{-3}$
E_a = activation energy, K
f = dynamic process model equations
F = process model Jacobian
$h[x_i]$ = vector valued function
H = measurement matrix
ΔH_r = heat of reaction, $cal \cdot mol^{-1}$
g = vector of inequality constraints
h = vector of equality constraints
k_o = pre-exponential factor, s^{-1}
K_f = Kalman gain matrix
P = error covariance matrix
q = volumetric flow rate, $cm^3 \cdot s^{-1}$
Q = covariance matrix of process noise
R = covariance matrix of measurement noise
t = time, time step, s
T = reactor temperature, K
T_c = coolant temperature, K
T_0 = feed temperature, K
T_r = reference temperature, K
U = heat-transfer coefficient, $cal\ (cm^2 \cdot s \cdot K)^{-1}$
V = volume of reactor, cm^3
u = process or control input vector
v = measurement noise vector
w = process noise vector
x = vector of process states
$\hat{x}$ = process state estimates
y = measurement vector
$\hat{y}$ = reconciled measurement vector

Greek letters

ρ = density of reactor contents, $g \cdot cm^{-3}$
Σ = measurement noise covariance matrix
Φ = generalized objective function

Literature Cited

Adams, E., "Enclosure Methods and Scientific Computation," *Conf. of World Cong. on Scientific Computation*, IMACS, Paris (1988).

Almasy, G. A., "Principles of Dynamics Balancing," *AIChE J.*, **36**(9) (1990).

Barash, M. M., "Approximate Decomposition of the Problems of Computing Material Balance for Continuous Processes in Automatic Control Systems During Production," *Automatika i Telemekhanika*, No. 2, 152 (1987).

Bhat, N., and T. J. McAvoy, "Use of Neural Networks for Dynamic Modeling and Control of Chemical Process Systems," *Proc. Amer. Control Conf.*, p. 1342 (1989).

Bhat, N., and T. J. McAvoy, "Use of Neural Networks for Dynamic Modeling and Control of Chemical Process Systems," *Comput. Chem. Eng.*, **14**, 573 (1990a).

Bhat, N. V., and T. J. McAvoy, "Information Retrieval from a Stripped Back-Propagation Network," AIChE Meeting, Chicago (1990b).

Cameron, D. B., W. R. Paterson, and W. Morton, "Dynamic Data Reconciliation Using a Differential Equation-based Simulator," *PSE Proc.*, p. II.4.1, Montebello, Canada (1991).

Cleeremans, A. D., D. Servan-Schreiber, and J. L. McClelland, "Finite State Automata and Simple Recurrent Networks," *Neural Comput.*, **3**, 372 (1989).

Crowe, C. M., "The Maximum-Power Test for Gross Errors in the Original Constraints in Data Reconciliation," *Can. J. of Chem. Eng.*, **70**(10), 1030 (1992).

Crowe, C. M., Y. A. G. Compos, and A. Hrymak, "Reconciliation of Process Flow Rates by Matrix Projection," *AIChE J.*, **29**(6), 881 (1983).

Darouach, M., and M. Zasadzinski, "Data Reconciliation in Generalized Linear Dynamic Systems," *AIChE J.*, **37**(2) (1991).

Elman, J. L., "Finding Structure in Time," *Cognitive Sci.*, **14**(2), 179 (1990).

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

Gertler, J., and G. A. Almasy, "Balance Calculations through Dynamic System Modeling," *Automatika*, **9**, 79 (1973).
Gertler, J., and D. Singer, "Augmented Models for Statistical Fault Isolation in Complex Dynamic Systems," *ACC Proc.*, Boston (1989).
Gherrity, M., "A Learning Algorithm for Analog, Fully Recurrent Neural Networks," *Proc. IEEE-IJCNN89*, p. I, 643 (1989).
Hernandéz, E., and Y. Arkun, "Neural Network Modeling and an Extended dmc Algorithm to Control Nonlinear Processes," *Proc. Amer. Control Conf.*, **3**, 2454 (1990).
Hertz, J. A., A. S. Krogh, and R. G. Palmer, *Introduction to the Theory of Neural Computation*, Addison Wesley, New York (1991).
Hoskins, J. C., and D. M. Himmelblau, "Artificial Neural Network Models of Knowledge Representation in Chemical Engineering," *Comput. and Chem. Eng.*, **12**, 881 (1988).
Jang, S.-S., B. Joseph, and H. Mukai, "Comparison of Two Approaches to On-Line Parameter and State Estimation of Nonlinear Systems," *Ind. Eng. Chem. Process Des. Dev.*, **25**, 809 (1986).
Jongenelen, E. M., C. den Heijer, and G. A. van Zee, "Detection of Gross Errors In Process Data Using Studentized Residuals," *Comput. Chem. Eng.*, **12**(8), 845 (1988).
Kao, C.-S., A. C. Tamhane, and R. S. H. Mah, "Gross Error Detection in Serially Correlated Process Data 2: Dynamic Systems," *Ind. Eng. Chem. Res.*, **31**, 254 (1992).
Karjala, T. W., D. M. Himmelblau, and R. Miikkulainen, "Data Rectification using Recurrent (Elman) Neural Networks," *Proc. IEEE-IJCNN*, Baltimore (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).
Kramer, M. A., "Nonlinear Principal Component Analysis using Autoassociative Neural Networks," *AIChE J.*, **37**, 233 (1991).
Kramer, M. A., "Autoassociative Neural Networks," *Comput. Chem. Eng.*, **16**(4), 313 (1992).
Kramer, M. A., and J. A. Leonard, "Diagnosis Using Backpropagation Neural Networks—Analysis and Criticism," *Comput. and Chem. Eng.*, **14**(12), 1323 (1990).
Lewis, F. L., *Optimal Estimation with an Introduction to Stochastic Control Theory*, Wiley, New York (1986).
Liebman, M. J., and T. F. Edgar, "Data Reconciliation for Dynamic Processes Using Nonlinear Programming Techniques," AIChE Meeting, Orlando, FL (1990).
Liebman, M. J., T. F. Edgar, and L. S. Lasdon, "Efficient Data Reconciliation and Estimation for Dynamic Processes Using Nonlinear Programming Techniques," *Comput. Chem. Eng.*, **16**(11/12), 963 (1992).
Miyakoda, K., and O. Talagrand, "The Assimilation of Past Data in Dynamical Analysis: I," *Tellus*, **23**, 310 (1971).
Narasimhan, S., and R. S. H. Mah, "Generalized Likelihood Ratios for Gross Error Identification in Dynamic Processes," *AIChE J.*, **34**, 1321 (1988a).
Narasimhan, S., and R. S. H. Mah, "Gross Error Identification in Closed Loop Dynamic Processes Using Generalized Likelihood Ratio Method," *Int. Symp. on Process Systems Eng.*, Sydney (1988b).
Narasimhan, S., and R. S. H. Mah, "Treatment of General Steady State Process Models in Gross Error Identification," *Comput. and Chem. Eng.*, **13**, 851 (1989).
Pearlmutter, B., "Learning State Space Trajectories in Recurrent Neural Networks," *Neural Comput.*, **2**, 263 (1989).
Pearlmutter, B., "Two New Learning Procedures for Recurrent Networks," *Neural Networks*, **3**, 99 (1990).
Pineda, F. J., "Generalization of Back-propagation in Recurrent Networks," *Physical Rev. Lett.*, **29**, 2229 (1987).
Psichogios, D., and L. Ungar, "A Hybrid Neural Network-First Principles Approach to Process Modeling," *AIChE J.*, **38**, 1499 (1992).
Qin, S.-Z., H.-T. Su, and T. J. McAvoy, "Comparison of Four Neural Net Learning Methods for Dynamic System Identification," *IEEE Trans. on Neural Networks*, **3**(1), 122 (1992).
Ragot, J., D. Maquin, and D. Sauter, "Data Validation Using Orthogonal Filters," *IEEE Proceed-D.*, **139**, 47 (1992).
Ramamurthi, Y., P. B. Sistu, and B. W. Bequette, "Control-Relevant Dynamic Data Reconciliation and Parameter Estimation," *Comput. Chem. Eng.*, **17**(1), 41 (1993).
Rosenberg, J., R. S. H. Mah, and C. Iordache, "Evaluation of Schemes for Detecting and Identifying Gross Errors in Process Data," *Ind. Eng. Chem. Res.*, **26**, 555 (1987).
Sage, A. P., and J. L. Melsa, *Estimation Theory with Applications to Communications and Control*, McGraw-Hill, New York (1971).
Seinfeld, J. H., "Optimal Stochastic Control of Nonlinear Systems," *AIChE J.*, **16**, 1016 (1970).
Serth, R. W., and W. A. Heenan, "Gross Error Detection and Data Reconciliation in Steam-Metering Systems," *AIChE J.*, **32**, 733 (1986).
Sorenson, H. W., ed., *Kalman Filtering: Theory and Applications*, IEEE Press, New York (1985).
Stanley, G. M., "Online Data Reconciliation for Process Control," AIChE Meeting, Los Angeles (1982).
Su, H.-T., and T. J. McAvoy, "Identification of Chemical Processes using Recurrent Networks," *Proc. Amer. Control Conf.*, 2314 (1991).
Su, H.-T., and T. J. McAvoy, "Long-Term Predictions of Chemical Processes Using Recurrent Neural Networks: A Parallel Training Approach," *Ind. Eng. Chem. Res.*, **31**(5), 1338 (1992).
Tamhane, A. C., and R. S. H. Mah, "Data Reconciliation and Gross Error Detection in Chemical Process Networks," *Technometrics*, **27**(4), 409 (1985).
Tjoa, I. B., and L. T. Biegler, "Simultaneous Strategies for Data Reconciliation and Gross Error Detection of Nonlinear Systems," *Comput. Chem. Eng.*, **15**(10), 679 (1991).
Ungar, L. H., B. A. Powell, and S. N. Kamens, "Adaptive Networks for Fault Diagnosis and Process Control," *Comput. Chem. Eng.*, **14**(4/5), 561 (1990).
Vaclavek, V., A. Petrichova, H. Janouskovcova, and L. Kaspar, "Dynamic Balance Computations of Chemical Processing Using Time and Topological Redundancies," *Comput. in the Design and Erection of Chem. Plants*, EFCE Publ. No. 10, 412 (1980).
Watanabe, K., I. Matsuura, M. Abe, M. Kubota, and D. M. Himmelblau, "Incipient Fault Diagnosis of Chemical Processes via Artificial Neural Networks," *AIChE J.*, **35**, 1803 (1989).
Watrous, R., B. Ladendorf, and G. Kuhn, "Complete Gradient Optimization of a Recurrent Network Applied to /b/, /d/, /g/ Discrimination," *J. Acoust. Soc. Am.*, **87**, 1301 (1988).
Watrous, R. L., "Learning Algorithms for Connectionist Networks: Applied Gradient Methods of Nonlinear Optimization," Technical Report MS-CIS-88-62, Dept. of Computer and Information Science, Univ. of Pennsylvania, Philadelphia (1988).
Werbos, P. J., "Backpropagation Through Time: What it Does and How to Do It," *Proc. IEEE*, **78**(10), 1550 (1990).
Williams, R., and D. Ziper, "A Learning Algorithm for Continually Running Fully Recurrent Neural Networks," *Neural Comput.*, **1**, 270 (1989a).
Williams, R. J., and J. Peng, "An Efficient Gradient Based Algorithm for On-Line Training of Recurrent Network Trajectories," *Neural Comput.*, **2**, 490 (1990).
Williams, R. J., and D. Ziper, "Experimental Analysis of the Real-Time Recurrent Learning Algorithm," *Connection Sci.*, **1**, 87 (1989b).
Wolf, P. J., S. N. Mohan, F. M. Stienan, and G. J. Bierman, "Computation of Factorized Error Covariance of the Difference Between Correlated Estimators," *IEEE Trans. Auto Control*, **35**, 1284 (1990).

Manuscript received Feb. 1, 1993, and revision received Dec. 10, 1993.