International Journal of Neural Systems, Vol. 8, No. 4 (August, 1997) 399–415
Special Issue on Data Mining in Finance

A CONSTRAINED NEURAL NETWORK KALMAN FILTER FOR PRICE ESTIMATION IN HIGH FREQUENCY FINANCIAL DATA

PETER J. BOLLAND* and JEROME T. CONNOR†
London Business School, Department of Decision Science,
Sussex Place, Regents Park, London NW1 4SA, UK

In this paper we present a neural network extended Kalman filter for modeling noisy financial time series. The neural network is employed to estimate the nonlinear dynamics of the extended Kalman filter. Conditions for the neural network weight matrix are provided to guarantee the stability of the filter. The extended Kalman filter presented is designed to filter three types of noise commonly observed in financial data: process noise, measurement noise, and arrival noise. The erratic arrival of data (arrival noise) results in the neural network predictions being iterated into the future. Constraining the neural network to have a fixed point at the origin produces better iterated predictions and more stable results. The performance of constrained and unconstrained neural networks within the extended Kalman filter is demonstrated on "Quote" tick data from the $/DM exchange rate (1993–1995).

1. Introduction

The study of financial tick data (trade data) is becoming increasingly important as the financial industry trades on shorter and shorter time scales. Tick data has many problematic features, it is often heavy tailed (Dacorogna 1995, Butlin and Connor 1996), it is prone to data corruption and outliers (Chung 1991), and it's variance is heteroscedastic with a seasonal pattern within each day (Dacorogna 1995). However the most serious problem with applying conventional methodologies to tick data is it's erratic arrival. The focus of this study is the prediction of erratic time series with neural networks. The issues of robust prediction and non-stationary variance are explored in Bolland and Connor (1996a) and Bolland and Connor (1996b).

There are three distinct types of noise found in real world time series such as financial tick data:

Process noise represents the shocks that drive the dynamics of the stochastic process. The distribution of the process/system noise is generally assumed to be Gaussian. For financial data the noise distributions can often be heavy tailed.

Measurement noise is the noise encountered when observing and measuring the time series. The measurement error is usually assumed to be Gaussian. The measurement of financial data is often corrupted by gross outliers.

Arrival noise reflects uncertainty concerning whether an observation will occur at the next time step. Foreign exchange quote data is strongly effected by erratic data arrival. At times the quotes are missing for forty seconds, at other times several ticks are contemporaneously aggregated.

These three types of noise have been widely studied in the engineering field for the case of a known deterministic system. The Kalman filter was invented to estimate the state vector of a linear deterministic system in the presence of the process, measurement, and arrival noise. The Kalman filter has been applied in the field of econometrics for the case

*E-mail: pbolland@medici-capm.com
†E-mail: Jerome.Connor@lebpa.sbi.com

when a deterministic system is unknown and must be estimated from the data, see for example Engle and Watson (1987). In Sec. 2, we give a brief description of the workings of the Kalman filter on linear models.

Neural networks have been successfully applied to the prediction of time series by many practitioners in a diverse range of problem domains (Weigend 1990). In many cases neural networks have been shown to yield significant performance improvements over conventional statistical methodologies. Neural networks have many attractive features compared to other statistical techniques. They make no parametric assumption about the functional relationship being modeled, they are capable of modeling interaction effects and the smoothness of fit is locally conditioned by the data. Neural networks are generally designed under careful laboratory conditions which while taking into consideration process noise, usually ignore the presence of measurement and arrival noise. In Sec. 3, we show how the extended Kalman filter can be used with neural network models to produce reliable predictions in the presence of process, measurement and arrival noise.

When observations are missing, the neural networks predictions are iterated. Because the iterated feedforward neural network predictions are based on previous predictions, it acts as a discrete time recurrent neural network. The dynamics of the recurrent network may converge to a stable point; however it is equally possible that the recurrent network could oscillate or become chaotic. Section 4 describes a neural network model which is constrained to have a stable point at the origin. Conditions are determined for which the neural network will always converge to a single fixed point. This is the neural network analogue of a stable linear system which will always converge to the fixed point of zero. The constrained neural network is useful for two reasons:

(1) The extended Kalman filter is guaranteed to have bounded errors if the neural network model is observable and controllable (discussed in Sec. 3). The existence of a stable fixed point will help make this the case.
(2) It reflects our belief that price increments beyond a certain horizon are unpredictable for the Dollar–DM foreign exchange rates.

For different problems, a neural network with a fixed point at zero may not make sense, in which case we do not advocate the constrained neural network. However, for modeling foreign exchange data, this constrained neural network should yield better results.

In Sec. 5, the performance of the extended Kalman filter with a constrained neural network is shown on Dollar–Deutsche Mark foreign exchange tick data. The performance of the constrained neural network is shown to be both quantitatively and qualitatively superior to the unconstrained neural network.

2. Linear Kalman Filter

Kalman filters originated in the engineering community with Kalman and Bucy (1960) and have been extensively applied to filtering, smoothing and control problems. The modeling of time series in state space form has advantages over other techniques both in interpretability and estimation. The Kalman filter lies at the heart of state space analysis and provides the basis for likelihood estimation. The general state space form for a multivariate time series is expressed as follows. The observable variables, a $n \times 1$ vector $\mathbf{y}_t$, are related to an $m \times 1$ vector $\mathbf{x}_t$, known as the state vector, via the observation equation,

$$\mathbf{y}_t = \mathbf{H}_t \mathbf{x}_t + \varepsilon_t \qquad t = 1, \ldots, N \tag{1}$$

where $\mathbf{H}_t$ is the $n \times m$ observation matrix and ε_t, which denotes the measurement noise, is a $n \times 1$ vector of serially uncorrelated disturbance terms with mean zero and covariance matrix $\mathbf{R}_t$. The states are unobservable and are known to be generated by a first-order Markov process,

$$\mathbf{x}_t = \Phi_t \mathbf{x}_{t-1} + \Gamma_t \eta_t \qquad t = 1, \ldots, N \tag{2}$$

where Φ_τ is the $m \times m$ state matrix, Γ_t is an $m \times g$ matrix, and η_t a $g \times 1$ vector of serially uncorrelated disturbance terms with mean zero and covariance matrix $\mathbf{Q}_t$. A linear AR(p) time series model can be represented in the general state form by

$$\mathbf{x}_t = \begin{bmatrix} \phi_1 & \phi_2 & \phi_3 & \cdots & \phi_{p-1} & \phi_p \\ 1 & 0 & 0 & \cdots & 0 & 0 \\ 0 & 1 & 0 & \cdots & 0 & 0 \\ 0 & 0 & 1 & & 0 & 0 \\ \vdots & \vdots & & & & \vdots \\ 0 & 0 & 0 & & 1 & 0 \end{bmatrix} \mathbf{x}_{t-1} + \Gamma_t \eta_t \tag{3}$$

$$\mathbf{y}_t = [1 \quad 0 \quad 0 \quad \cdots \quad 0]\mathbf{x}_t + \varepsilon_t \tag{4}$$

where Γ_t is 1 for the element $(1,1)$ and zero elsewhere. The state space model given by (3) and (4) is known as the phase canonical form and is not unique for AR(p) models. There are three other forms which differ in how the parameters are displayed but all representations give equivalent results, see for example Aoki (1987) or Akaike (1975).

The Kalman filter is a recursive procedure for computing the optimal estimates of the state vector at time t, based on the information available at time t. The Kalman filter enables the estimate of the state vector to be continually updated as new observations become available. For linear system equations with normally distributed disturbance terms the Kalman filter produces the minimum mean squared error estimate of the state $\mathbf{x}_t$. The filtering equation written in the error prediction correction format is,

$$\tilde{\mathbf{x}}_{t+1|t+1} = \hat{\mathbf{x}}_{t+1|t} + \mathbf{K}_{t+1}\bar{\mathbf{y}}_{t+1|t} \tag{5}$$

where $\hat{\mathbf{x}}_{t+1|t} = \Phi_t\tilde{\mathbf{x}}_{t|t}$ is the predicted state vector based on information available at time t, $\mathbf{K}_{t+1}$ is the $m \times n$ Kalman gain matrix, and $\bar{\mathbf{y}}_{t+1|t}$ denotes the innovations process given by $\bar{\mathbf{y}}_{t+1|t} = \mathbf{y}_{t+1} - \mathbf{H}_{t+1}\hat{\mathbf{x}}_{t+1|t}$. The estimate of the state at $t+1$, is produced from the prediction based on information available at time t, and a correction term based on the observed prediction error at time $t+1$. The Kalman gain matrix is specified by the set of relations,

$$\mathbf{K}_{t+1} = \mathbf{P}_{t+1|t}\mathbf{H}'_{t+1}[\mathbf{H}_{t+1}\mathbf{P}_{t+1|t}\mathbf{H}'_{t+1} + \mathbf{R}_{t+1}]^{-1} \tag{6}$$

$$\mathbf{P}_{t+1|t} = \Phi_{t+1|t}\mathbf{P}_{t|t}\Phi'_{t+1|t} + \Gamma_{t+1|t}\mathbf{Q}_t\Gamma'_{t+1|t} \tag{7}$$

$$\mathbf{P}_{t|t} = [\mathbf{I} - \mathbf{K}_t\mathbf{H}_t]\mathbf{P}_{t|t-1} \tag{8}$$

where $\mathbf{P}_{t|t}$ is the filter error covariance. Given the initial conditions, $\mathbf{P}_{0|0}$ and $\hat{\mathbf{x}}_{0|0}$, the Kalman filter gives the optimal estimate of the state vector as each new observation becomes available.

The parameters of the system of the Kalman filter, represented by the vector ψ, can be estimated using maximum likelihood. For the typical system the parameters, ψ, would consist of the elements of $\mathbf{Q}_t$ and $\mathbf{R}_t$, auto-regressive parameters within Φ_t and sometimes parameters from within the observation matrix $\mathbf{H}_t$. The parameters will depend upon the specific formulation of the system being modeled. For normally distributed disturbance terms, writing the likelihood function $L(\mathbf{y}_t; \psi)$, in terms of the density of $\mathbf{y}_t$ conditioned on the information set at time $t-1$, $\mathbf{Y}_{t-1} = \{\mathbf{y}_{t-1}, \ldots, \mathbf{y}_1\}$, the log likelihood function is,

$$\log L = -\frac{1}{2}\sum_{t=1}^{N}[\bar{\mathbf{y}}'_{t|t-1_\psi}N^{-1}_{t|t-1_\psi}\bar{\mathbf{y}}_{t|t-1_\psi} + \ln|N_{t|t-1_\psi}|] \tag{9}$$

where $\bar{\mathbf{y}}_{t|t-1_\psi}$ is the innovations process and $N_{t|t-1_\psi}$ is the covariance of that process,

$$N_{t|t-1_\psi} = \mathbf{H}_{t_\psi}\mathbf{P}_{t|t-1_\psi}\mathbf{H}'_{t_\psi} + \mathbf{R}_{t_\psi}\,. \tag{10}$$

The log likelihood function given in (9) depends on the initial conditions $\mathbf{P}_{0|0}$ and $\hat{\mathbf{x}}_{0|0}$. de Jong (1989) showed how the Kalman filter can be augmented to model the initial conditions as diffuse priors which allow the initial conditions to be estimated without filtering implications. After τ time steps, where τ is specified by the modeller, a proper prior for state vectors will be estimated and the log likelihood can then be described as in (9) and (10).

The Kalman filter described is discrete, (as tick data is quantisized into time steps, i.e. seconds), however the methodology could be extended to continuous time problems with the Kalman–Bucy filter (Meditch 1969).

2.1. *Missing data*

Irregular (erratic) times series presents a serious problem to conventional modeling methodologies. Several methodologies for dealing with erratic data have been suggested in the literature. Muller *et al.* (1990), suggest methods of linear interpolation between erratic observations to obtain a regular homogenous times series. Other authors (Ghysels and Jasiak 1995) have favored nonlinear time deformation ("business- time" or "tick-time). The Kalman filter can be simply modified to deal with observations that are either missing or subject to contemporaneous aggregation. The choice of time discretization (i.e. seconds, minutes, hours) will depend on the specific irregular time series. Too small a discretization and the data set will be composed of mainly missing observation. Too long a discretization and non-synchronous data will be aggregated. The timing (time-stamp) of financial transactions is

only measured to a precision of one second. A time discretization of one second is a natural choice for modeling financial time series as aggregation will be across synchronous data and missing observations will be in the minority.

Augmenting the Kalman filter to deal with erratic time series is achieved by allowing the dimensions of the observation vector $\mathbf{y}_t$ and the observation errors ε_t, to vary at each time step $(n_t \times 1)$. The observation equation dimensions now vary with time so,

$$\mathbf{y}_t = \mathbf{W}_t \mathbf{H}_t \mathbf{x}_t + \varepsilon_t \qquad t = 1, \ldots, N \tag{11}$$

where $\mathbf{W}_t$ is an $n_t \times n$ matrix of fixed weights. This gives rise to several possible situations:

- Contemporaneous aggregation of the first component of the observation vector, in addition to the other components. So the weight matrix is $((n+1) \times n)$ and has a row augment for the first component to give rise to the two values $y^1_{t,1}$ and $y^1_{t,2}$.

$$y_t = \begin{bmatrix} y^1_{t,1} \\ y^1_{t,2} \\ \vdots \\ y^n_t \end{bmatrix}_{n+1},$$

$$\mathbf{W}_t = \begin{bmatrix} 1 & 0 & 0 & \ldots & 0 \\ 1 & 0 & 0 & \ldots & 0 \\ 0 & 1 & 0 & \ldots & 0 \\ 0 & 0 & 1 & & 0 \\ \vdots & \vdots & & \ddots & \\ 0 & 0 & 0 & & 1 \end{bmatrix}_{(n+1)\times n} \tag{12}$$

- All components of the observation equation occur, where the weight matrix is square $(n \times n)$ and an identity.

$$\mathbf{y}_t = \begin{bmatrix} y^1_t \\ \vdots \\ y^n_t \end{bmatrix}_n, \quad \mathbf{W}_t = \begin{bmatrix} 1 & \ldots & 0 \\ \vdots & 1 & \vdots \\ 0 & \ldots & 1 \end{bmatrix}_{n\times n} \tag{13}$$

- The ith component of the observation vector is unobserved, so n_t is $(n-1)$ and the weight matrix $(n \times (n-1))$ has a single row removed.

$$\mathbf{y}_t = \begin{bmatrix} y^1_t \\ \vdots \\ y^{i-1}_t \\ y^{i+1}_t \\ \vdots \\ y^n_t \end{bmatrix}_{n-1},$$

$$\mathbf{W}_t = \begin{bmatrix} & 0 & \\ \mathbf{I}_{i-1} & \vdots & 0 \\ & \vdots & \\ & \vdots & \\ 0 & \vdots & \mathbf{I}_{n-i} \\ & 0 & \end{bmatrix}_{n\times(n-1)} \tag{14}$$

- All components of the observations vector are unobserved, n_t is zero, and the weight matrix is undefined.

$$\mathbf{y}_t = [NULL], \quad \mathbf{W}_t = [NULL] \tag{15}$$

The dimensions of the innovations $\bar{\mathbf{y}}_{t|t-1_\psi}$, and their covariance $N_{t|t-1_\psi}$ also vary with n_t. Equation (10) is undefined when n_t is zero. When there are no observations on a given time t, the Kalman filter updating equations can simply be skipped. The resulting predictions for the states and filtering error covariance's are, $\hat{\mathbf{x}}_t = \hat{\mathbf{x}}_{t|t-1}$ and $\mathbf{P}_t = \mathbf{P}_{t|t-1}$. For consecutive missing observations $(n_t = 0, \ldots, n_{t+l} = 0)$ a multiple step prediction is required, with repeated substitution into the transition Eq. (2) multiple step predictions are obtained by

$$\mathbf{x}_{t+l} = \left[\prod_{j=1}^{l} \Phi_{t+j}\right] \mathbf{x}_t + \sum_{j=1}^{l-1} \left[\prod_{i=j+1}^{l} \Phi_{t+i}\right]_t \Gamma_{t+j}\eta_{t+j} + \Gamma_{t+l}\eta_{t+l}\,. \tag{16}$$

The conditional expectations at time t of (16) is given by

$$\hat{\mathbf{x}}_{t+l|t} = \left[\prod_{j=1}^{l} \Phi_{t+j}\right] \hat{\mathbf{x}}_t \tag{17}$$

and similarly the expectation of the multiple step ahead prediction error covariance for the case of time invariant system is given by

$$\mathbf{P}_{t+l|t} = \Phi^l \mathbf{P}_t \Phi^l + \sum_{j=0}^{l-1} \Phi^j \Gamma \mathbf{Q}\, \Gamma' \Phi^j\,. \tag{18}$$

The estimates for multiple step predictions of $\hat{\mathbf{y}}_{t+l|t}$ and $\hat{\mathbf{x}}_{t+l|t}$ can be shown to be the minimum mean square error estimators $\mathbf{E}_t(\mathbf{y}_{t+l})$.

2.2. *Non-Gaussian Kalman filter*

The Gaussian estimation methods outlined above can produce very poor results in the presence of gross outliers. The Kalman filter can be adapted to filter processes where the disturbance terms are non-Gaussian. The distributions of the process and measurement noise for the standard Kalman filter is assumed to be Gaussian. The Kalman filter can be adapted to filter systems where either the process or measurement noise is non-Gaussian (Mazreliez 1973). Mazreliez showed that Kalman filter update equations are dependent on the distribution of the innovations and its score function. For a known distribution the optimal minimum mean variance estimator can be produced. In the situation where the distribution is unknown, piecewise linear methods can employed to approximate it (Kitagawa 1987). Since the filtering equations depends on the derivative of the distribution, such methods can be inaccurate. Other methods take advantage of higher moments of the density functions and require no assumptions about the shape of the probability densities (Hilands and Thomoploulos 1994). Carlin *et al.* (1992), demonstrate the Kalman filter robust to non-Gaussian state noise. For robustness, heavy tailed distributions can be consider as a mixture of Gaussians, or an ε-contaminated distribution. If the bulk of the data behaves in a Gaussian manner then we can employ a density function which is not dominated by the tails. The Huber function (1980), can be shown to be the least informative distribution in the case of ε-contaminated data.

In this paper we achieve robustness to measurement outliers by the methodology first described in Bolland and Connor (1996a). A Huber function is employed as the score function of the residuals. For spherically ε-contaminated normal distributions in R^3 the score function g_0 for the Huber function is given by,

$$\begin{aligned} g_0 &= r \qquad \text{for} \quad r < c \\ &= c \qquad \text{for} \quad r \geq c\,. \end{aligned} \tag{19}$$

The Huber function behaves as a Gaussian for the center of the distribution and as an exponential in the tails. The heavy tails will allow the robust Kalman filter to down weight large residuals and provide a degree of insurance against the corrupting influence of outliers. The degree of insurance is determined by the distribution parameter c in (19). The parameter can be related to the level of assumed contamination as shown by Huber (1980).

2.3. *Sufficient conditions*

For a Kalman filter to estimate the time varying states of the system given by (1) and (2), the system must be both observable and controllable.

An observable system allows the states to be uniquely estimated from the data. For example, in the no noise condition ($\varepsilon_t = 0$) the state $\mathbf{x}_t$ can be determined uniquely from future observations, if the observability matrix given by $\mathbf{O} = [\mathbf{H}' \quad \Phi'\mathbf{H}' \quad \Phi'^2\mathbf{H}'' \quad \Phi'^3\mathbf{H}' \quad \cdots]$ has $\text{Rank}(\mathbf{O}) = m$ where m is the number of elements in the state vector $\mathbf{x}_t$. The condition for observability is also equivalent to (see Aoki, 1990)

$$G_O = \sum_{k=0}^{\infty} (\Phi')^k \mathbf{H}'\mathbf{H}\Phi^k > 0 \tag{20}$$

where the observability Grammian, $\mathbf{G}_O$, satisfies the following Lyaponov equation, $\Phi'\mathbf{G}_O\Phi - \mathbf{G}_O = -\mathbf{H}'\mathbf{H}$.

The notion of controllability emerged from the control theory literature where v_t denotes an action that can be made by an operator of a plant. If the system is controllable, any state can be reached with the correct sequence of actions. A system Φ is controllable if the reachability matrix defined by $\mathbf{C} = [\Gamma \quad \Phi\Gamma \quad \Phi^2\Gamma \quad \Phi^3\Gamma \quad \cdots]$ has $\text{Rank}(\mathbf{H}) = m$ which is equivalent to (see Aoki, 1990)

$$\mathbf{G}_C = \sum_{k=0}^{\infty} \Phi^k \Gamma\Gamma' (\Phi')^k > 0 \tag{21}$$

where the controllability Grammian, $\mathbf{G}_C$, satisfies the following Lyaponov equation, $\Phi\mathbf{G}_C\Phi' - \mathbf{G}_C = -\Gamma\Gamma'$.

When a state space model is in one of the four canonical variate forms such as (3) and (4), the Grammians $\mathbf{G}_O$ and $\mathbf{G}_C$ are identical and the Controllability and Observability requirements are the same. If Φ is non-singular and the absolute values of eigenvalues are less than one, the Observability and

Controllability requirements of (20) and (21) will be met and the Kalman filter will estimate a unique sequence of underlying states. In the next two sections a neural network analogue of the stable linear system is introduced which will be used within the extended Kalman filter. If the eigenvalues are greater than one, the Kalman filter may still estimate a unique sequence of underlying states but this must be evaluated on a system by system basis. See for example the work by Harvey on the random trend model.

3. Extended Kalman Filter

The Kalman filter can be extended to filter nonlinear state space models. These models are not generally conditionally Gaussian and so an approximate filter is used. The state space model's observation function $h_t(\mathbf{x}_t)$ and the state update function $f_t(\mathbf{x}_{t-1})$ are no longer linear functions of the state vector. Using a Taylor expansion of these nonlinear functions around the predicted state vector and the filtered state vector we have,

$$h_t(\mathbf{x}_t) \approx h_t(\hat{\mathbf{x}}_{t|t-1}) + \hat{\mathbf{H}}_t(\mathbf{x}_t - \hat{\mathbf{x}}_{t|t-1})$$
$$\hat{\mathbf{H}}_t = \left.\frac{\partial h_t(\mathbf{x}_t)}{\partial \mathbf{x}'_{t-1}}\right|_{\mathbf{x}_t=\hat{\mathbf{x}}_{t|t-1}}, \tag{22}$$

$$f_t(\mathbf{x}_{t-1}) \approx f_t(\hat{\mathbf{x}}_{t-1}) + \hat{\Phi}_t(\mathbf{x}_{t-1} - \hat{\mathbf{x}}_{t-1})$$
$$\hat{\Phi}_t = \left.\frac{\partial f_t(\mathbf{x}_{t-1})}{\partial \mathbf{x}'_{t-1}}\right|_{\mathbf{x}_{t-1}=\hat{\mathbf{x}}_{t-1}}. \tag{23}$$

The extended Kalman filter (Jazwinski 1970) is produced by modeling the linearized state space model using a modified Kalman filter. The linearized observation equation and state update equation are approximated by,

$$\mathbf{y}_t \approx \hat{\mathbf{H}}_t\mathbf{x}_t + \hat{\mathbf{d}}_t + \varepsilon_t$$
$$\hat{\mathbf{d}}_t = \mathbf{h}_t(\hat{\mathbf{x}}_{t|t-1}) - \hat{\mathbf{H}}_t\mathbf{x}_{t|t-1} \tag{24}$$

$$\mathbf{x}_t \approx \hat{\Phi}_t\mathbf{x}_{t-1} + \hat{\mathbf{c}}_t + \eta_t$$
$$\hat{\mathbf{c}}_t = f_t(\hat{\mathbf{x}}_{t-1}) - \hat{\Phi}_t\hat{\mathbf{x}}_{t-1} \tag{25}$$

and the corresponding Kalman prediction equation and the state update equation (in prediction error correction format) are,

$$\hat{\mathbf{x}}_{t|t-1} = f_t(\hat{x}_{t-1})$$
$$\hat{\mathbf{x}}_{t|t} = \hat{\mathbf{x}}_{t|t-1} + \mathbf{P}_{t|t-1}\hat{\Phi}_t N_t^{-1}[y_t - h_t(\hat{\mathbf{x}}_{t|t-1})]. \tag{26}$$

The quality of the approximation depends on the smoothness of the nonlinearity since the extended Kalman filter is only a first order approximation of $\mathbf{E}\{\mathbf{x}_t|\mathbf{Y}_{t-1}\}$. The extended Kalman filter can be augmented as described in Eqs. (11)–(15) to deal with missing data and contemporaneous aggregation. The functional form of $h_t(\mathbf{x}_t)$ and $f_t(\mathbf{x}_{t-1})$ are estimated using a neural network described in Sec. 5.

3.1. *Sufficient conditions*

The nonlinear analog of the AR(p) model expressed in phase canonical form given by (3) and (4) is expressed as

$$\mathbf{x}_t^T = [x_t \quad x_{t-1} \quad x_{t-2} \quad \cdots \quad x_{t-p+1}] \tag{27}$$

$$\mathbf{f}(\mathbf{x}_{t-1})^T = [f(\mathbf{x}_{t-1}) \quad x_{t-1} \quad x_{t-2} \quad \cdots \quad x_{t-p+1}] \tag{28}$$

$$y_t = \mathbf{H}\mathbf{f}(\mathbf{x}_{t-1})^T + \varepsilon_t \tag{29}$$

with $\mathbf{H} = [1 \quad 0 \quad 0 \quad \cdots \quad 0]$ and $\Gamma = [1 \quad 0 \quad 0 \quad \cdots \quad 0]$ as in the linear time series model of (3) and (4).

The Extended Kalman Filter has a long history of working in practice, but only recently has theory produced bounds on the resulting error dynamics. Baras *et al.* (1988) and Song and Grizzle (1995) have shown bounds on the EKF error dynamics of deterministic systems in continuous and discrete time respectively. La Scala, Bitmead, and James (1995) have shown how the error dynamics of an EKF on a general nonlinear stochastic discrete time are bounded. The nonlinear system considered by La Scala *et al.* has a linear output map which is also true for our system described in (27)–(29). As in the case of the linear Kalman filter, the results of La Scala *et al.* require that the nonlinear system be both observable and controllable.

The nonlinear observability Grammian is more complicated than its linear counterpart in (21) and must be evaluated upon the trajectory of interest ($t_1 \to t_2$). The nonlinear observability Grammian as defined by La Scala *et al.* and applied to the system given by (27)–(29) is

$$\mathbf{G}_O(t, M) = \sum_{i=t-M}^{t} S(i, t)'\mathbf{H}'\mathbf{R}_i^{-1}\mathbf{H}'\mathbf{S}(i, t) \tag{30}$$

where $\mathbf{S}(t_2,\, t_1) = \mathbf{F}_y(t_2-1)\mathbf{F}_y(t_2-2)\cdots\mathbf{F}_y(t_1)$ and $\mathbf{F}_y(t) = \partial f/\partial \mathbf{x}(\mathbf{y}(t))$. The nonlinear controllability Grammian as defined by La Scala *et al.* and applied to the system given by (27)–(29) is

$$\mathbf{G}_C(t,\, M) = \sum_{i=t-M}^{t-1} \mathbf{S}(t,\, i+1)\mathbf{Q}_i\mathbf{S}(t,\, i+1)'\,. \quad (31)$$

As in the linear case of Sec. 2.3, the requirements that the nonlinear observability and controllability Grammians given by (30) and (31) are positive are identical with the appropriate choices of t and M due to $\mathbf{H}'\mathbf{R}_i^{-1}\mathbf{H}' \sim \mathbf{Q}_i'$. If the Grammians of (30) and (31) are both positive and finite then the system is said to be controllable (observable).

Because of the similarity between observability and controllability criterions, only the observability criterion is examined closely. If the observability criterion is positive definite and finite for some values of M, the system is observable. For the nonlinear auto-regressive system given by (27)–(29),

$$\mathbf{F}_y(k) = \begin{bmatrix} \dfrac{\partial f(y)}{\partial x_1} & \dfrac{\partial f(y)}{\partial x_2} & \cdots & \dfrac{\partial f(y)}{\partial x_{p-1}} & \dfrac{\partial f(y)}{\partial x_p} \\ 1 & 0 & \cdots & 0 & 0 \\ 0 & 1 & \cdots & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & \cdots & 1 & 0 \end{bmatrix}. \quad (32)$$

The observability Grammian, (30), can be shown to be finite if $\mathbf{S}(t+\Delta,\, t)$ converges exponentially to zero. $\mathbf{S}(t+\Delta,\, t)$ will converge exponentially to zero if all the eigenvalues of $\mathbf{F}_y(k)$ are inside the unit circle for all possible values of y. For any fixed choice of y, Eq. (32) is equivalent to an AR(p) updating equation in state space form, this equivalence is easily understood because $\mathbf{F}_y(k)$ corresponds to the linearization of a nonlinear AR(p) model given by (27)–(29). If the corresponding AR(p) model converges exponentially to zero for all values of y, then $\mathbf{S}(t+\Delta,\, t)$ will converge exponentially to zero also for this y. An AR(p) time series model given by $w_t = \sum_{i=1}^{p} a_i w_{t-i} + e_t$ will converge exponentially to zero if all of the zeros of the polynomial $\prod_{i=1}^{p}(1 - a_i z^{-1})$ are inside the unit circle, see for example Roberts and Mulis (1987). Thus, the nonlinear autoregressive system, $\mathbf{S}(t+\Delta,\, t)$ will converge exponentially to zero provided all the roots of the polynomial given in (33) are inside the unit circle

$$g(z) = \prod_{i=1}^{p}(1 - \partial f(y)/\partial x_{i1} z^{-1})\,. \quad (33)$$

If in addition, $\partial f(y)/\partial x_i$ is non-zero, G_O is finite positive definite and there exists an optimal choice of states which can be estimated with the EKF. In Sec. 4, conditions are discussed which guarantee the observability of a neural network system.

The notions of observability and controllablility are not unique to Kalman filtering. Both notions are used extensively within control theory. For information related to neural networks and control theory, see for example Levin and Narendra (1993) and Levin and Narendra (1996).

4. Constrained Neural Networks

Often data generating processes have symmetries or constraints which can be exploited during the estimation of a neural network model. The problem is how to constrain a neural network to better exploit these symmetries. One of the key strengths of neural networks is the ability to approximate any function to any desired level of accuracy whether the function has or lacks symmetries, see for example Cybenko (1989). In this section a constraint is examined which is both natural to the financial problems investigated and desirable from a Kalman filtering perspective.

A neural network embedded within a Kalman Filter is used in an iterative fashion in the event of missing data. As mentioned earlier, linear models will either diverge if they are unstable or go to zero if they are stable. In addition, the Kalman Filter will converge on a unique state sequence if the system is stable, if it is unstable the Kalman Filter may or may not converge on a unique state sequence. It is thus natural when dealing with linear systems to want to constrain the system to have stable behavior. Constraining a linear model to be stable is as simple as insuring the eigenvalues of the state transition matrix are between ± 1.

For neural networks the dynamics of the system can exhibit complicated behavior where the state trajectories can be cyclical, chaotic, or converge to one of multiple fixed point attractors. For the problems we are interested in, a limiting point of zero is desirable. From the perspective of a Kalman filter, the

system is stable. From view of the financial example presented later, a limiting point of zero will correspond to future price increments being unknown beyond a certain distance into the future.

There are two ways of putting symmetry constraints in neural networks, a *hard* symmetry constraint obtained by direct embedding of the symmetry in the weight space and a *soft* constraint of pushing a neural network towards a symmetric solution but not enforcing the final solution to be symmetric.

The soft constraint of biasing towards symmetric solutions can be viewed from two perspectives, providing hints and regularization. Abu-Mostafa (1990), (1993), and (1995), showed among other things that it is possible to augment a data set with examples obtained by generating new data under the symmetry constraint. The neural network is then trained on the augmented training set and is likely to have a solution that is closer to the desired symmetric solution then would otherwise be the case. The soft constraint can come in the form of extra data generated from a constraint or as a constraint within the learning algorithm itself. Alternatively, neural networks which drift from the symmetric solution can be penalized by a regularization term, see for example the tangent prop by Simard, Victorri, La Cunn, and Denker (1992). Both the hint and the regularization approach to soft constraints were shown to be related by Leen (1995).

The alternative of hard constraints was first proposed by Giles *et al.* (1990) in which a neural network was made invariant under geometric transformations of the input space. We propose to incorporate a hard constraint in which a neural network is forced to have a fixed point at the origin, producing a forecast of zero when past observations, y_{t-i} for $i = 1, \ldots, p$ are equal to zero.

The imposition of a fixed point in the neural network will have the largest effect when the predictor is being iterated. The iterated predictor is, as will be described in Sec. 4.1, a recurrent neural network. A fixed point need not alter the estimated neural network significantly. Jin *et al.* (1994) show there must be at least one fixed point in a recurrent network anyway. As the example in Sec. 5 demonstrates, an unconstrained iterated predictor will often converge to a fixed point in any case, the problem is that the resulting fixed point is found to be undesirable.

Other possible dangers exist with recurrent neural networks. The existence of a fixed point does not preclude chaos, the fixed point may be unstable or only stable locally. Often recurrent neural networks can perform oscillations or more complicated types of stable or chaotic patterns, see for example Marcus and Westervelt (1989), Pearlmutter (1989), Pineda (1989), Cohen (1992), Blum and Wang (1992) and many others. Under some circumstances this complicated behavior in an iterated predictor could be desirable, however it is the view of this paper that any advantages are outweighed by the dangers of using a Kalman filter on a poorly understood nonstable system.

A fixed point at $\hat{y}_t = 0$ is achieved by augmenting a neural network with a constrained output bias parameter. For a network consisting of H hidden units and with activation function f, the functional form of the constrained network is given by,

$$\hat{y}_t = \sum_{i=1}^{H} W_i f\left(\sum_{j=1}^{p} w_{ij} y_{t-j} + \theta_i\right) - \sum_{i=1}^{H} W_i f(\theta_i)\,. \tag{34}$$

where parameters of the network W_i, w_{ij}, and θ_i, represent the output weights, input weights and the input biases. The first term of (34) describes a standard feedforward neural network and the second term represents the "hard wired" constraint. Conditions which will ensure that the fixed point is a stable attractor will be discussed in Sec. 4.2. The fixed point will only be guaranteed to be a local attractor. Outside of the local area surrounding the origin, the behavior of the neural network model will be determined by the data used for training.

The estimation of the parameters can be achieved by using a slightly augmented back-propagation algorithm. For a mean squared error cost function the fixed point constraint leads to a slightly more complicated learning rule based on the following derivatives,

$$\frac{d\hat{y}_t}{dW_i} = f\left(\sum_{j=1}^{p} w_{ij} y_{t-j} + \theta_i\right) - f(\theta_i) \tag{35}$$

$$\frac{d\hat{y}_t}{d\theta_i} = W_i f\left(\sum_{j=1}^{p} w_{ij} y_{t-j} + \theta_i\right) \times \left(1 - f\left(\sum_{j=1}^{p} w_{ij} y_{t-j} + \theta_i\right)\right) - W_i f(\theta_i)(1 - f(\theta_i)) \quad (36)$$

with the derivative of the prediction w.r.t. the inputs, $d\hat{y}_t/dw_{ik}$, being unaffected by the constraint on the neural network. The neural network training algorithm assumes that the initial weight matrix satisfies the fixed point constraint.

4.1. *Iterated behavior of neural network*

The neural network in (34) is a simple feedforward neural network with no feedback connections. As the neural network stops receiving new data and begins running in an iterated manner, the neural network predictions will be based on past neural network predictions

$$\hat{x}_t = f_{NN}(\hat{x}_{t-1}, \ldots, \hat{x}_{t-k}, x_{t-k-1}, \ldots, x_{t-p}). \quad (37)$$

Recurrent connections are implicitly being added when the neural network is running iteratively within the Extended Kalman filtering system. After p time steps have been processed without receiving any data, the system is no longer receiving data from outside the system. The iterated system is as depicted in Fig. 1. The recurrent activations of the neural units can be divided into three sets; the activations of the hidden units, s_i for $1 \le i \le H$; the activation of the output unit, s_{H+1} for $i = H + 1$; the activation of the input units, s_{H+I} for $H + 2 \le i \le H + p$.

$$s_i(t+1) = f\left(\theta_i + \sum_{j=1}^{p} w_{ij} s_{H+j}(t)\right), \quad 1 \le i \le H \quad (38)$$

$$s_{(H+1)}(t+1) = \sum_{j=1}^{H} W_j s_j(t), \quad i = H + 1 \quad (39)$$

$$s_{H+i}(t+1) = s_{H+i-1}(t) \quad H + 2 \le i \le H + p \quad (40)$$

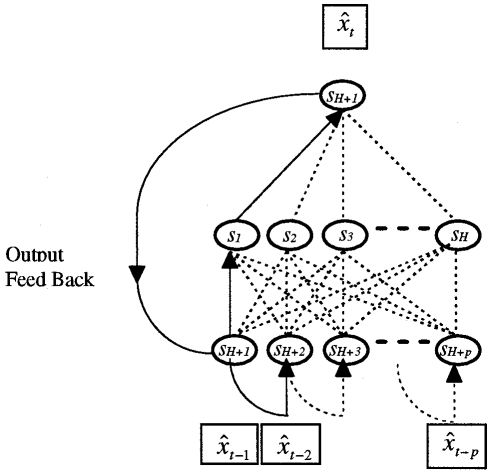

Fig. 1. Iterated neural network.

where the parameters $\mathbf{W}$ and $\mathbf{w}$ are identical to the system given by (34). The recurrent neural network is different from the feedforward neural network because of the addition of p neurons, $s_{H+1}(t), \ldots, s_{H+p}(t)$, which enable the previous predictions to be stored in the system via (39) and (40) and used as a basis for further predictions. The resulting recurrent connections are much more limited than considered in most recurrent network studies.

Whether the behavior of the recurrent network given by (38)–(40) is stable, oscillatory or chaotic is determined by the weights. The next section will state under what conditions the neural network will converge to a fixed point.

4.2. *Absolute stability of discrete recurrent neural networks*

The strongest results on the stability of recurrent networks are for the continuous time versions. The popular Hopfield network (Hopfield 1984) is guaranteed to be stable because the weight matrix is confined to be symmetric. The convergence of non-symmetric recurrent neural networks in continuous time has been investigated by Kelly (1990) and others. Matsuoka (1992) in particular has shown how weights of a non-symmetric recurrent neural network can be extremely large and convergence can still be guaranteed in some cases.

The guarantees for convergence of discrete time recurrent networks are not as strong as that of

continuous time neural networks. Many weight configurations which lead to convergence in continuous time neural networks do not produce convergence for the case of discrete time recurrent networks.

For a discrete time recurrent network to be absolutely stable, it must converge to a fixed point independent of the initial starting conditions. Results of Jin *et al.* (1994) for fully recurrent networks will be reviewed. These results will be applied to the special case of the constrained network. Jin *et al.* (1994) reported that a fully recurrent network of the form

$$s_i(t+1) = f\left(\theta_i + \sum_{j=1}^{H} \omega_{ij} s_j(t)\right) \quad i = 1, \ldots, H \tag{41}$$

will converge to a fixed point if all of the eigenvalues of the matrix $\mathbf{W}$ of network weights ω_{ij} fall within the unit circle. Jin *et al.* then generalize the results by noting that the eigenvalues are not change by the transformation $\mathbf{P}^{-1}\mathbf{W}\mathbf{P}$ when is a non-singular matrix of equivalent dimensions. Jin *et al.* consider the transformation $\mathbf{P} = \mathrm{diag}(p_1, \ldots, p_N)$ where p_i are all positive which leads to leads to following guarantee of stability

$$|\omega_{ii}| + \frac{1}{p_i^{2\gamma-1}} (R_i^p)^{\gamma} (C_i^p)^{1-\gamma} < c_i^{-1} \tag{42}$$

with c_i denoting the maximum slope of the ith nonlinear neuron transfer function, $\gamma \in [0, 1]$, R_i^p and C_i^p are given by

$$R_i^p = \sum_{j \neq i} p_j |\omega_{ij}|\,, \tag{43}$$

$$C_i^p = \sum_{j \neq i} \frac{1}{p_j} |\omega_{ji}| \tag{44}$$

and $(p_1, \ldots, p_n)$ are all positive.

4.2.1. *The general case*

Stability guarantees for neural networks using (41) are typically quoted for two cases, $\gamma = 0$ and $\gamma = 1$. The fully recurrent network given by (41) is more general than the iterated neural network given in (38)–(40). For the iterated neural network, the stability guarantees (38) reduces to the following two cases.

With a choice of $\gamma = 0$:

$$\frac{p_i}{p_{H+1}} |W_i| < c_i^{-1}, \quad i = 1, \ldots, H \tag{45}$$

$$p_i \left(\frac{1}{p_{i+1}} + \sum_{j=1}^{H} \frac{|w_{j,i-H}|}{p_j} \right) < 1, \quad i = H+1, \ldots, H-p-1 \tag{46}$$

$$p_i \sum_{j=1}^{H} \frac{|w_{j,i-H}|}{p_j} < 1, \quad i = H+p \tag{47}$$

This is equivalent to a weighted sum of the absolute value of the weights leaving each neuron multiplied by the maximum slope of a neuron nonlinearity being less than one.

With a choice of $\gamma = 1$.

$$\frac{1}{p_i} \sum_{j=1}^{p} p_{H+j} |w_{ij}| < c_i^{-1} \qquad i = 1, \ldots, H \tag{48}$$

$$\frac{1}{p_{H+1}} \sum_{j=1}^{H} p_j |W_j| < 1 \qquad i = H+1 \tag{49}$$

$$\frac{p_{i-1}}{p_i} < 1 \qquad i = H+2, \ldots, H+p\,. \tag{50}$$

This is equivalent to a weighted sum of the absolute value of the weights going entering each neuron multiplied by the maximum slope of a neuron nonlinearity being less than one.

Any choice of positive $(p_1, \ldots, p_n)$ is allowable, two natural choices for the iterated neural network are now listed.

4.2.2. *All p_i equal*

The choice of $p = (p_0, p_0, \ldots, p_0)$ will not satisfy (45)–(47) or (48)–(50) because of the recurrent connections which are equal to 1. However, if $p_i = p_0 + (i - H - 1)\varepsilon$ is used instead for $i = H+2, \ldots, H+p$ where ε is vanishingly small, the stability guarantees of (48)–(50) reduce to

$$\sum_{j=1}^{p} |w_{ij}| < c_i^{-1}, \quad i = 1, \ldots, H\,, \tag{51}$$

$$\sum_{j=1}^{H} |W_j| < 1, \quad i = H+1\,. \tag{52}$$

Equations (45)–(47) do not have an equally parsimonious expression for the p_i being equal case.

4.2.3. *Scale invariance*

A transformation which will not alter the recurrent network is obtained by taking advantage of its autoregressive structure. The output of the network can be scaled as long as the weights connected to past predictions are divided by the same constant, this leads to a network with equivalent dynamics and weights given by $W_i' = kW_i$ and $w_{ij}' = k^{-1}w_{ij}$. The guarantee for stability given by (51) and (52) becomes

$$\sum_{j=1}^{p} |w_{ij}| < kc_i^{-1}, \quad i = 1, \ldots, H\,, \tag{53}$$

$$\sum_{j=1}^{H} |W_j| < k^{-1}, \quad i = H+1\,. \tag{54}$$

Equations (51) and (52) can easily be optimized by choosing $k^{-1} = \Sigma_{j=1}^{H}|W_j| + \varepsilon$ with ε being vanishingly small and positive and checking that (51) still holds.

4.3. *The relationship between absolute stability and observability*

In Sec. 3 the extended Kalman filter was said to be observable if the eigenvalues of the transition matrix were inside the unit circle. The stability results of the constrained neural network presented in Sec. 4.1 are equivalent to guaranteeing the roots of the polynomial $g(z)$ given in (33) are inside the unit circle. In addition, if (32) is invertible, a constrained neural network will be observable.

4.4. *Stability of the constrained neural network*

When the constrained neural network given in (34) is iterated it is the same form as the iterated neural network given in (38)–(40), with the exception of a bias term. Bias terms have no effect on the stability of an iterated constrained neural network, hence the above stability arguments apply also to the stability of the constrained neural network.

5. Application to "Tick" Data

The financial institutions are continually striving for a competitive edge. With the increase in computer power and the advent of computer trading, the financial markets have dramatically increased in speed. Technology is creating new trading horizons which give access to possible inefficiencies of the market. The volume of trading has increased hugely and the price series (tick data) of these trades offers a great source of information about the dynamics of the markets and the behavior of its practitioners. A major problem for financial institutions that trade at high frequency is estimating the true value of the commodity being traded at the instance of a trade. This problem is faced by both sides of transaction, the market maker and the investor. The uncertainty arises from many sources. There are several sources of additive noise. The state noise dominates, however bid ask bounce, price quantization, and typographic errors result in observation noise. In addition to these uncertainties the estimate of the value has to be based on information that may be several time periods old. The methodology we present addresses these problems and produces an estimate of the "true mid-price" irrespective of the data's erratic nature.

The Dollar/Deutsche Mark exchange rates were modeled. The data was obtained from Reuters FXFX pages, and covered the period March 1993–April 1995. As the data is composed of only quotes then the possible sources of noise are amplified. Compared to brokers data or actual transactions data the quality of quote data is very poor. The price spreads (bid/ask) are larger, the data is prone to miss-typing error and also some of the quotes are simply advertisements for the market makers. In general quotation have more process noise and more sources of measurement noise. However quotation data is readily available and the most liquid and accessible view of the market.

To eliminate some of the bid-ask noise in the series, the changes in the mid-price were modeled. The filtered states represent the estimated "true" mid-prices. For an NAR(p) model the one step ahead prediction is,

$$\hat{\mathbf{x}}_{t+1|t} = \mathbf{f}(\tilde{\mathbf{x}}_t) = \tilde{x}_t + \mathbf{g}(\tilde{x}_t - \tilde{x}_{t-1}, \ldots, \tilde{x}_{t-p+1} - \tilde{x}_{t-p}) \tag{55}$$

where **g** is the predicted change in state based on the changes in previous time periods. The function **g** was estimated by both a constrained feed-forward neural network and an unconstrained neural network. For the results shown here a simple NAR(1) model was estimated. A parsimonious neural network specification was used, with a 4 hidden unit feed forward network using sigmoidal activation functions.

An estimation maximization (EM) algorithm is employed at the center of a robust estimation procedure based on filtered data (for full details see Bolland and Connor 1996). The EM algorithm, see Dempster, Laird, and Rubin (1977), is the standard approach when estimating model parameters with missing data. The EM algorithm has been used in the neural network community before, see for example Jordan and Jacobs (1993) or Connor, Martin, and Atlas (1994). During the estimation step, the missing data, namely the $\mathbf{x}_t$, ε_t, and η_t of (1) and (2) must be estimated. This amounts to estimating parameters of the state update function f and the noise variance matrices $\mathbf{Q}_t$ and $\mathbf{R}_t$. With the estimated missing data assumed to be true, the parameters are then chosen by way of maximizing the likelihood. This procedure is iterative with new parameter estimates giving rise to new estimates of missing data which in turn give rise to newer parameter estimates. The iterative estimation procedure was initialized by constructing a contiguous data set (no arrival noise) and estimating a linear auto-regressive model. The variances of the disturbance terms are non-stationary. To remove some of this non-stationarity the intra day seasonal pattern of the variances were estimated (Bolland and Connor 1996). The parameters of the state update function were assumed to be stationary across the length of the data set.

Table 1. Non-iterated forecasts.

	r-Squared	Correlation
NN Unconstrained AR (1)	0.254995	0.505472
NN Constrained AR (1)	0.251663	0.505478

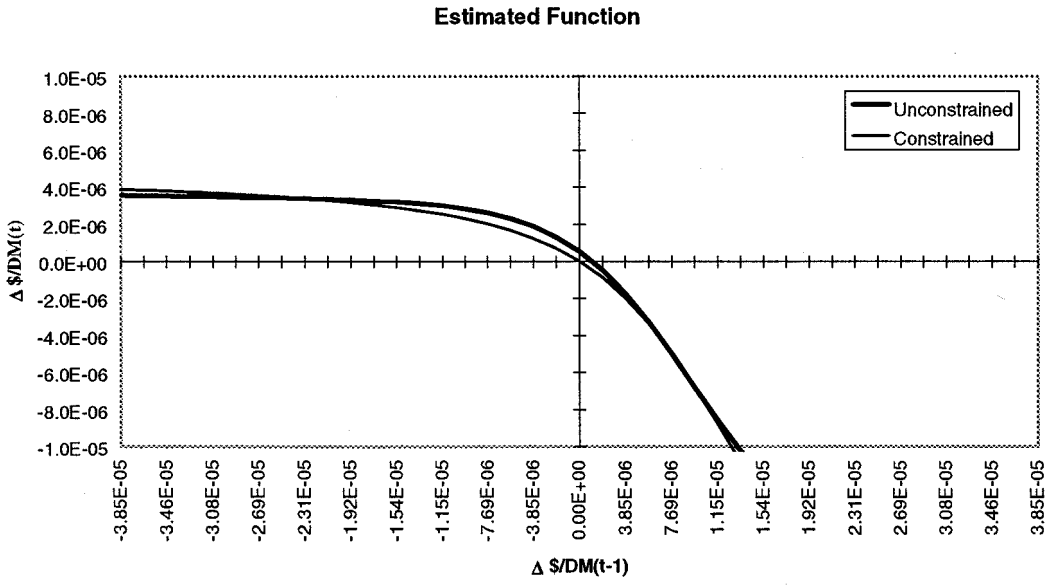

Fig. 2. Estimated function.

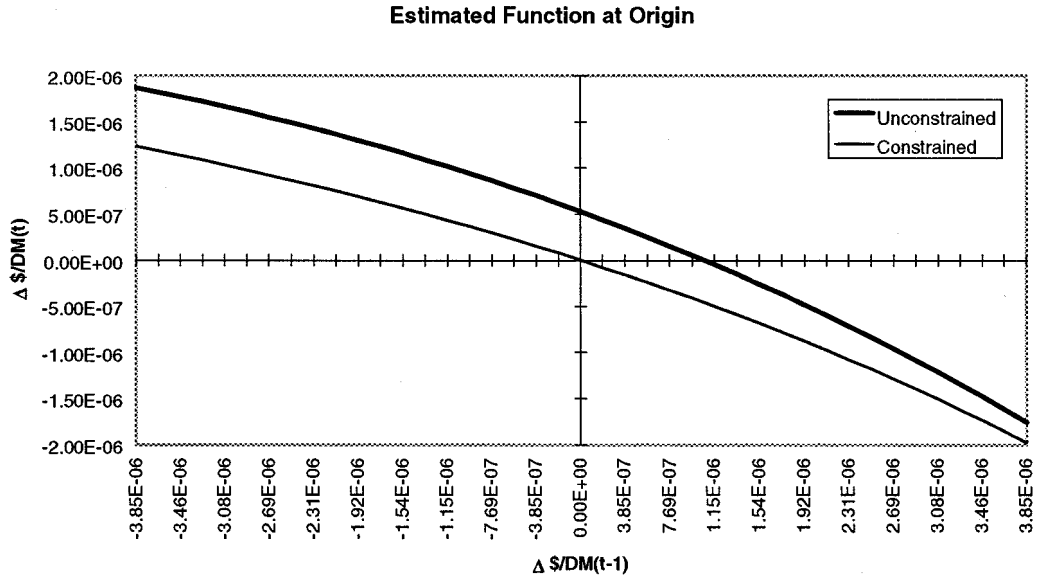

Fig. 3. Estimated function at origin.

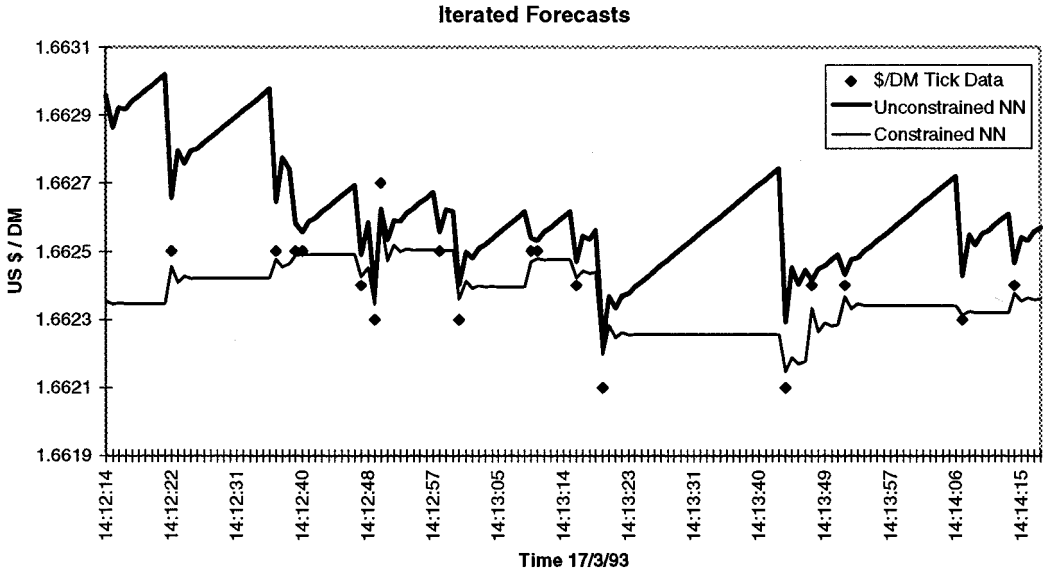

Fig. 4. Filtered tick data.

Table 1 gives the performance of the two models for non-iterated forecast. The constraints on the network are not detrimental to the overall performance, with the percentage variance explained (r-squared) and the correlation being very similar.

Figure 2 shows the fitted function of a simple NAR(1) model for the constrained neural network and the unconstrained neural network. The qualitative difference in the models estimated function are only slight.

Figure 3 shows the estimated function around the origin. At the origin the constrained network has a bias as it has been restrained from learning the mean of the estimation set. Although this bias is only very

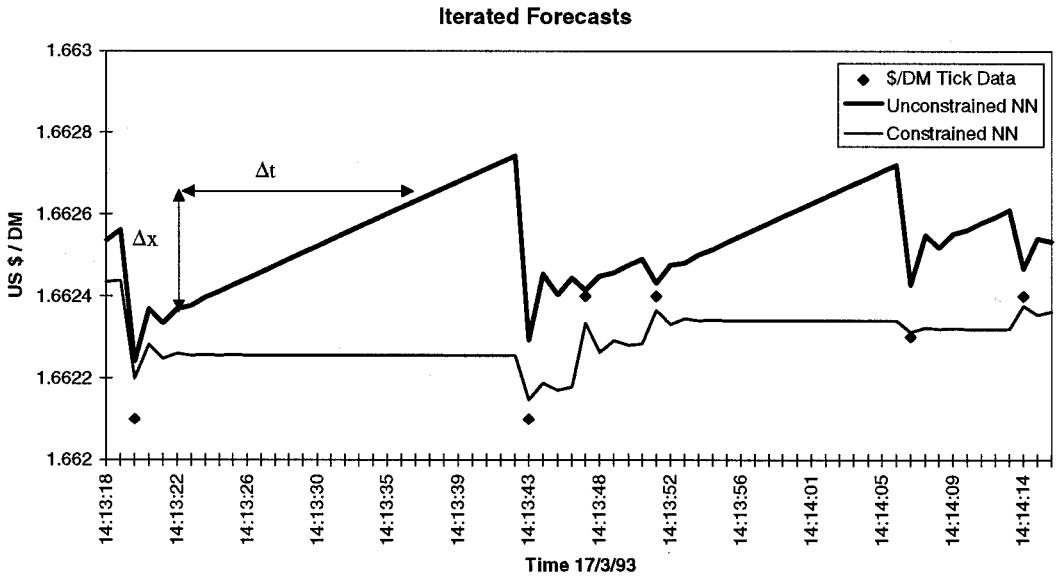

Fig. 5. Stable points of network.

Table 2. Test set performance.

	MAD	MSE
NN Unconstrained AR (1)	7.81897	1.776407
NN Constrained AR (1)	5.218660	0.701285

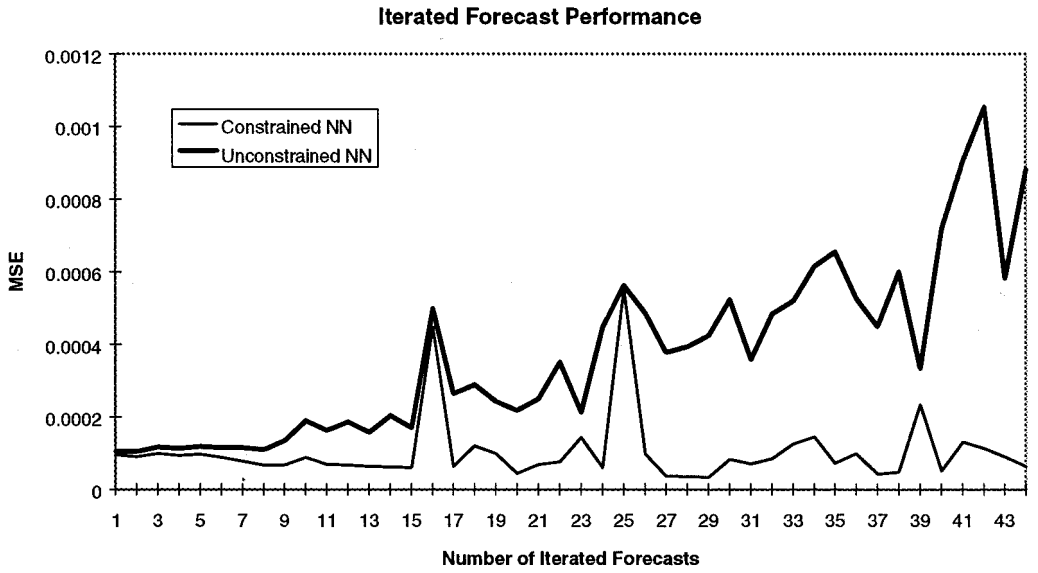

Fig. 6. Iterated forecast error.

small (for linear regression the bias is 5.12×10^{-7} with a t-statistic of 0.872), its effect large as it is compounded by iterating the forecast.

The filter produces estimated states (shown in Fig. 4) which can be viewed as the "true mid-prices," the noise due to market friction's has been estimated and filtered (bid-ask bounce, price quantization, etc.). The iterated forecasts reach the stable point after only small number of iterations (approx. 5).

Figure 5 shows a close up of the iterated forecasts of the two networks. The value of the stable point in the case of the simple NAR(1) is the final gradient of the iterated forecasts. The stable point of the constrained neural network is zero, and the stable point of unconstrained is 1.57×10^{-6}. This is the result of a small bias in the model. When the forecast is iterated this bias is accumulated and therefore the unconstrained network predictions trend. For the constrained network the iterated forecast soon reach the stable point zero, reflecting our prior belief in the long term predictability of the series. The mean squared error (MSE) as well as the median absolute deviations (MAD) of the constrained and unconstrained networks are given in Table 2, and shown in Fig. 6. As the forecast is iterated the MSE for the unconstrained grows rapidly. This is due to its trending forecast.

It is clear that the performance is improved by constraining the neural network. The MSE for the constrained neural network remains relatively constant with prediction. The accuracy of iterated prediction should decrease as the forecast is iterated. From Fig. 6 it is clear that the MSE is not increasing with number of iterations. However, only in periods of very low trading activity are forecasts iterated for 40 time steps also in periods of low trading activity the variance in the time series is low. So the errors in these periods are only small even though the time between observations can be large.

6. Conclusion and Discussion

Using neural networks within an extended Kalman filter is desirable because of the measurement and arrival noise associated with foreign exchange tick data. The desirability of using a stable system within a Kalman filter was used as an analogy for developing a "stable neural network" for use within an extended Kalman filter. The "stable neural network" was obtained by constraining the neural network to have a fixed point of zero input gives zero output. In addition, the fixed point at zero reflected our belief that price increments beyond a certain horizon are unknowable and a predicted price increment of zero is best (random walk). This constrained neural network is optimized for foreign exchange modeling, for other problems a constrained neural network with a fixed point at zero would be undesirable.

The behavior of the neural network within the extended Kalman filter under normal operating conditions is roughly the same as the unconstrained neural network. But in the presence of missing data, the iterated predictions of the constrained neural network far outperformed the unconstrained neural network in both quality and performance metrics.

References

Y. S. Abu-Mostafa 1990, "Learning from hints in neural networks," *J. Complexity* **6**, 192-198.

Y. S. Abu-Mostafa 1993, "A method for learning from hints," *Advances in Neural Information Processing 5*, ed. S. J. Hanson (Morgan Kaufmann), pp. 73–80.

Y. S. Abu-Mostafa 1995, "Financial applications of learning from hints," *Advances in Neural Information Processing 7*, eds. J. Tesauro, D. S. Touretzky and T. Leen (Morgan Kaufman), pp. 411–418.

H. Akaike 1975, "Markovian representation of stochastic processes by stochastic variables," *SIAM J. Control* **13**, 162–173.

M. Aoki 1987, *State Space Modeling of Time Series*, Second Edition (Springer–Verlag).

J. S. Baras, A. Bensoussan and M. R. James 1988, "Dynamic observers as asymptotic limits of recursive filters: Special cases," *SIAM J. Appl. Math.* **48**, 1147–1158.

E. K. Blum and X. Wang 1992, "Stability of fixed points and periodic orbits and bifurcations in analog neural networks," *Neural Networks* **5**, 577–587.

P. J. Bolland and J. T. Connor 1996, "A robust non-linear multivariate Kalman filter for arbitrage identification in high frequency data," in *Neural Networks in Financial Engineering* (Proceedings of the NNCM-95), eds. A-P. N. Refenes, Y. Abu-Mostafa, J. Moody and A. S. Weigend (World Scientific), pp. 122–135.

P. J. Bolland and J. T. Connor 1996b, "Estimation of intra day seasonal variances," Technical Report, London Business School.

S. J. Butlin and J. T. Connor 1996, "Forecasting foreign exchange rates: Bayesian model comparison using Gaussian and Laplacian noise models," in *Neural Networks in Financial Engineering* (*Proc. NNCM-95*), eds. A.-P. N. Refenes, Y. Abu-Mostafa, J. Moody and A. Weigend (World Scientific, Singapore), pp. 146–156.

B. P. Carlin, N. G. Polson and D. S. Stoffer 1992, "A Monte Carlo approach to nonnormal and nonlinear state-space modeling," *J. Am. Stat. Assoc.* **87**, 493–500.

P. Y. Chung 1991, "A transactions data test of stock index futures market efficiency and index arbitrage profitability," *J. Finance* **46**, 1791–1809.

M. A. Cohen 1992, "The construction of arbitrary stable dynamics in nonlinear neural networks," *Neural Networks* **5**, 83–103.

J. T. Connor, R. D. Martin and L. E. Atlas 1994, "Recurrent neural networks and robust time series prediction," *IEEE Trans. Neural Networks* **4**, 240–254.

G. Cybenko 1989, "Approximation by superpositions of a sigmoidal function," *Math. Control, Signals Syst.* **2**, 303–314.

M. M. Dacorogna 1995, "Price behavior and models for high frequency data in finance," Tutorial, NNCM conference, London, England, Oct, pp. 11–13.

P. de Jong 1989, "The likelihood for a state space model," *Biometrica* **75**, 165–169.

A. P. Dempster, N. M. Laird and D. B. Rubin 1977, "Maximum likelihood from incomplete data via the EM algorithm," *J. Royal Stat. Soc.* **B39**, 1–38.

R. F. Engle and M. W. Watson 1987, "The Kalman filter: Applications to forecasting and rational expectation models," in *Advances in Econometrics Fifth World Congress*, Volume I, ed. T. F. Bewley (Cambridge University Press).

E. Ghysels and J. Jasiak 1995, "Stochastic volatility and time deformation: An application of trading volume and leverage effects," *Proc. HFDF-I Conf.*, Zurich, Switzerland, March 29–31, Vol. 1, pp. 1–14.

C. L. Giles, R. D. Griffen and T. Maxwell 1990, "Encoding geometric invariance's in higher order neural networks," in *Neural Information Processing Systems*, ed. D. Z. Anderson (American Institute of Physics), pp. 301–309.

A. V. M. Herz, Z. Li and J. Leo van Hemmen, "Statistical mechanics of temporal association in neural networks with delayed interactions," *NIPS*, 176–182.

T. W. Hilands and S. C. Thomoploulos 1994, "High-order filters for estimation in non-Gaussian noise," *Inf. Sci.* **80**, 149–179.

J. J. Hopfield 1984, "Neurons with graded response have collective computational properties like those of two-state neurons," *Proc. Natl. Acad. Sci.* **81**, 3088–3092.

P. J. Huber 1980, *Robust Statistics* (Wiley, New York).

A. H. Jazwinshki 1970, *Stochastic Processes and Filtering Theory* (Academic Press, New York).

L. Jin, P. N. Nikiforuk and M. Gupta 1994, "Absolute stability conditions for discrete-time recurrent neural networks," *IEEE Trans. Neural Networks* **5**(6), 954–64.

M. I. Jordan and R. A. Jacobs 1992, "Hierarchical mixtures of experts and the EM algorithm," *Neural Comput.* **4**, 448–472.

R. E. Kalman and R. S. Bucy 1961, "New results in linear filtering and prediction theory," *Trans. ASME J. Basic Eng. Series D* **83**, 95–108.

D. G. Kelly 1990, "Stability in contractive nonlinear neural networks," *IEEE Trans. Biomed. Eng.* **37**, 231–242.

G. Kitagawa 1987, "Non-Gaussian state-space modeling of non-stationary time series," *J. Am. Stat. Assoc.* **82**, 1033–1063.

B. F. La Scala, R. R. Bitmead and M. R. James 1995, "Conditions for stability of the extended kalman filter and their application to the frequency tracking problem," *Math. Control Signals Syst.* **8**, 1–26.

Y. Le Cun, B. Boser, J. S. Denker, D. Henderson, R. E. Howard, W. Hubbard and L. D. Jackel 1990, "Handwritten digit recognition with a back-propagation network," in *Advances in Neural Information Processing Systems 2*, ed. D. S. Touretzky (Morgan Kaufmann), pp. 396–404.

T. K. Leen 1995, "From data distributions to regularization in invariant learning," in *Advances in Neural Information Processing 7*, eds. J. Tesauro, D. S. Touretzky and T. Leen (Morgan Kaufman), pp. 223–230.

A. U. Levin and K. S. Narendra 1996, "Control of nonlinear dynamical systems using neural networks — Part II: Observability, identification, and control," *IEEE Trans. Neural Networks* **7**(1).

A. U. Levin and K. S. Narendra 1993, "Control of nonlinear dynamical systems using neural networks: Controllability and stabilization," *IEEE Trans. Neural Networks* **4**(2).

C. M. Marcus and R. M. Westervelt 1989, "Dynamics of analog neural networks with time delay," *Advances in Neural Information Processing Systems 2*, ed. D. S. Touretzky (Morgan Kaufmann).

C. J. Mazreliez 1973, "Approximate non-Gaussian filtering with linear state and observation relations," *IEEE Trans. Automatic Control*, February.

K. Matsuoka 1992, "Stability Conditions for nonlinear continuous neural networks with asymmetric connection weights," *Neural Networks* **5**, 495–500.

J. S. Meditch 1969, *Stochastic Optimal Linear Estimation and Control* (McGraw-Hill, New York).

U. A. Muller, M. M. Dacorogna, R. B. Oslen, O. V. Pictet, M. Schwarz and C. Morgenegg 1990, "Statistical study of foreign exchange rates, empirical evidence of a price change scaling law, and intraday analysis," *J. Banking and Finance* **14**, 1189–1208.

B. A. Pearlmutter 1989, "Learning state-space trajectories in recurrent neural networks," *Neural Computation* **1**, 263–269.

F. J. Pineda 1989, "Recurrent backpropagation and the dynamical approach to adaptive neural computation," *Neural Computation* **1**, 161–172.

R. Roll 1984, "A simple implicit measure of the effective bid-ask spread in an efficient market," *J. Finance* **39**, 1127–1140.

P. Simard, B. Victorri, Y. La Cunn and J. Denker 1992, "Tangent prop- a formalism for specifying selective variances in an adaptive network," in *Advances in*

Neural Information Processing 4, eds. J. E. Moody, S. J. Hanson and R. P. Lippman (Morgan Kaufman), pp. 895–903.

Y. Song and J. W. Grizzle 1995, "The extended Kalman filter as a local observer for nonlinear discrete-time systems," *J. Math. Syst. Estim. Control* **5**, 59–78.

A. S. Weigend, B. A. Huberman and D. E. Rumelhart 1990, "Predicting the future: A connectionist approach," *Int. J. Neural Systems* **1**, 193–209.