A Two-Observation Kalman Framework for Maximum-Likelihood Modeling of Noisy Time Series

Alex T. Nelson and Eric A. Wan
Department of Electrical and Computer Engineering, Oregon Graduate Institute
P.O. Box 91000 Portland, OR 97291
atnelson@ece.ogi.edu, ericwan@ece.ogi.edu

Abstract— **Modeling a noisy time series requires the dual estimation of both the model parameters and the underlying clean time series. Most approaches estimate the model parameters by minimizing the mean squared prediction error, but estimate the time series by minimizing another cost function. We justify the use of the same maximum-likelihood cost for both parameter and time series estimation, and present a new weight update procedure for recursive minimization of this cost. This learning algorithm uses a two-observation form of the extended Kalman filter, and provides a natural extension of the *Dual Extended Kalman Filter* procedure previously developed by the authors.**

I. Introduction

The estimation and prediction of noisy time series have applications ranging from speech enhancement to financial and economic forecasting. While a model of the time series can be used for performing these tasks, it is well known that the presence of noise will cause standard regression techniques to produce a biased model of the underlying data [8], [9].

Several methods have been proposed for reducing this model bias. Some involve an *iterative* scheme of estimating first the model, and then the time series. These include Lim and Oppenheim's well-known approach to fitting linear predictive coding (LPC) models to degraded speech [3], Conner et al.'s use of an extended Kalman filter (EKF) to perform state-estimation interspersed with neural network training [1], as well as Weigend and Zimmerman's *Clearning* method [11].

Other methods involve *simultaneous* estimation of the model and the time series. These are often referred to as *recursive prediction error* methods, and have the advantage of offering on-line estimation. Nelson and Stear developed an adaptive estimator for linear systems using a Kalman filter for both state and parameter estimation [7]. Ljung and Söderström have presented several related techniques [4] for linear systems. In the neural network literature, Matthews [6] and Williams [12] have extended the approach to recurrent neural networks. Extensions and analysis in the context of neural estimation and prediction for speech enhancement and signal processing have been provided by Wan and Nelson [9], [10]. The authors have referred to their approaches as *Dual Extended Kalman Filtering*.

During the signal estimation phase of each of these approaches (iterative and simultaneous), a maximum-likelihood cost function involving both prediction and estimation error is minimized. However, during the model estimation phase of these algorithms, only the mean squared prediction error is minimized. The prediction targets are either the original noisy data or the current estimates of the data, and the prediction error is minimized using either Least Squares (in the linear case), backpropagation, or Kalman filter training. Note that all these approaches may also be related to the EM algorithm [2], which we will elaborate on in Section 2.

The main purpose of this paper is to propose minimization of the same maximum-likelihood cost function during both time-series *and* model estimation. This is accomplished by a new algorithm for model estimation based on a two-observation form of the Dual Extended Kalman Filter (Dual EKF).

We will begin in Section 2 with a derivation of the maximum-likelihood cost function. Section 3 presents a recursive algorithm for minimizing this cost function within the Dual EKF framework. In Section 4, we provide experimental results showing superior estimation properties of the new algorithm on short data sets.

II. Maximum-Likelihood Cost Function

A. The Model

We assume the noisy time series of interest is generated by a nonlinear autoregressive function with additive observation noise:

$$\begin{aligned} x(k) &= f\big(x(k-1), \ldots x(k-M), \mathbf{w}\big) + v(k) \\ y(k) &= x(k) + n(k), \qquad\qquad \forall k \in \{1 \ldots N\} \end{aligned} \tag{1}$$

where $x(k)$ corresponds to the true underlying time series driven by process noise $v(k)$, and $f(\cdot)$ is a nonlinear function (*e.g.*, a neural network) of past values of $x(k)$ parameterized by $\mathbf{w}$.

The only available observation is $y(k)$, which contains additive noise $n(k)$. Again, the dual estimation problem is to find estimates $\hat{x}(k)$ and $\hat{\mathbf{w}}$. To decouple the problem, we first consider estimation of each quantity separately, under the assumption that the other quantity is known.

B. Development of the Cost Function

We will begin by focusing on estimation of the clean time series, $\{x(k)\}_1^N$. If the model parameters $\mathbf{w}$ are known, the Bayesian approach is to find an estimate $\{\hat{x}(k)\}_1^N$ that maximizes the conditional probability density of $\{x(k)\}_1^N$ given the noisy data $\{y(k)\}_1^N$ and the weights $\mathbf{w}$. Defining $\mathbf{X}$,$\mathbf{Y}$, and $\mathbf{W}$ as the vectors of random variables associated with $\{x(k)\}_1^N$,$\{y(k)\}_1^N$, and $\mathbf{w}$, respectively, we can write this density function as $\rho_{\mathbf{X|YW}}(\{x(k)\}_1^N|\{y(k)\}_1^N,\mathbf{w})$, abbreviated as $\rho_{\mathbf{X|YW}}$. Using Bayes rule, we expand this as:

$$\rho_{\mathbf{X|YW}} = \frac{\rho_{\mathbf{YX|W}}}{\rho_{\mathbf{Y|W}}} \tag{2}$$

Because $\rho_{\mathbf{Y|W}}$ is independent of the time series itself, the estimates of $\{x(k)\}_1^N$ can be found by maximizing $\rho_{\mathbf{YX|W}}$ alone.

Similarly, to estimate $\mathbf{w}$ when the time series $\{x(k)\}_1^N$ and the data $\{y(k)\}_1^N$ are available, we consider the conditional density $\rho_{\mathbf{W|XY}}$. This can be expanded as:

$$\rho_{\mathbf{W|YX}} = \frac{\rho_{\mathbf{YX|W}} \cdot \rho_{\mathbf{W}}}{\rho_{\mathbf{YX}}}. \tag{3}$$

Since $\rho_{\mathbf{YX}}$ is independent of $\mathbf{w}$, the numerator alone can be maximized to estimate the weights. Furthermore, the term $\rho_{\mathbf{W}}$ can be dropped if we assume that no prior information is available on the distribution of the weights. This leaves $\rho_{\mathbf{YX|W}}$ as the likelihood function for the weights.

Hence, $\rho_{\mathbf{YX|W}}$ is the relevant term to maximize for estimation of both the weights $\mathbf{w}$ and the time series $\{x(k)\}_1^N$, when the other quantity is assumed to be known. From the definition of conditional densities, $\rho_{\mathbf{YX|W}}$ can be expressed as:

$$\rho_{\mathbf{YX|W}} = \rho_{\mathbf{Y|XW}} \cdot \rho_{\mathbf{X|W}}. \tag{4}$$

Assuming $v(k)$ and $n(k)$ have Gaussian statistics with variances σ_v^2 and σ_n^2, respectively, this in turn can be shown to be:

$$\rho_{\mathbf{YX|W}} = \frac{1}{\sqrt{(2\pi)^N(\sigma_n^2)^N}} \exp\left(-\sum_{k=1}^{N}\frac{(y(k)-x(k))^2}{2\sigma_n^2}\right) \cdot \frac{1}{\sqrt{(2\pi)^N(\sigma_v^2)^N}} \exp\left(-\sum_{k=1}^{N}\frac{(x(k)-x^-(k))^2}{2\sigma_v^2}\right),$$

where $x^-(k) \triangleq f(x(k-1),...x(k-M),\mathbf{w})$ denotes the predicted value of the time series. This makes no assumptions about the nature of $f(\cdot)$. We can maximize $\rho_{\mathbf{YX|W}}$ by taking the log and minimizing the cost function:

$$J(x,\mathbf{w}) = \sum_{k=1}^{N}\left(\frac{(y(k)-x(k))^2}{\sigma_n^2} + \frac{(x(k)-x^-(k))^2}{\sigma_v^2}\right). \tag{5}$$

We can put J in an alternate form by making the substitutions: $e(k) \triangleq y(k)-x(k)$, and $e^-(k) \triangleq y(k)-x^-(k)$, representing an *observation* and a *prediction* error, respectively. After a few steps of algebra, this results in:

$$J(x,\mathbf{w}) = \sum_{k=1}^{N}\left(\left(\frac{1}{\sigma_n^2}+\frac{1}{\sigma_v^2}\right)e(k)^2 - \frac{2}{\sigma_v^2}e(k)e^-(k) + \frac{1}{\sigma_v^2}e^-(k)^2\right). \tag{6}$$

Note that this cost is now expressed in terms of the squared observation error, the squared prediction error, and their cross-product.

Minimizing J with respect to $\mathbf{w}$ produces the maximum-likelihood estimate $\hat{\mathbf{w}}$. Similarly, minimizing J with respect to $\{x(k)\}_1^N$ produces maximum-likelihood estimates of the time series $\{\hat{x}(k)\}_1^N$ if the initial conditions $x(k)$, $-M < k \leq 0$ are taken to be unknown. In this case, $\rho_{\mathbf{X|W}}$ can be dropped from Equation 4, leaving maximization of $\rho_{\mathbf{Y|XW}}$, which is a likelihood function for $\{x(k)\}_1^N$.

As a side note, it is satisfying that maximizing $\rho_{\mathbf{YX|W}}$ with respect to the weights is equivalent to maximizing $\rho_{\mathbf{X|W}}$. This is because, in Equation 4, $\rho_{\mathbf{Y|XW}} = \rho_{\mathbf{Y|X}}$, which is independent of $\mathbf{w}$ when $\{x(k)\}_1^N$ is fixed.

C. Optimization and Alternate Costs

We have shown that J can be minimized for maximum-likelihood estimation of both the time series and the weights, $\mathbf{w}$. In fact, the *errors in variables* method (EIV) from the statistical regression literature [8], [9] can be adjusted for the time series model to yield the same cost function J. However, EIV is a batch algorithm that does not allow for on-line learning. In addition, it is especially inefficient for models with large numbers of parameters (*e.g.*, neural networks).

Our goal, then, is to develop an on-line algorithm that simultaneously minimizes $J(\hat{x},\mathbf{w})$ to find $\hat{\mathbf{w}}$, and minimizes $J(x,\hat{\mathbf{w}})$ to find $\{\hat{x}(k)\}_1^N$. We will show in the next section how the Dual EKF framework [9] allows for the development of such an algorithm.

It is worth noting that all of the methods listed in the introduction generate estimates $\{\hat{x}(k)\}_1^N$ of the time series by minimizing $J(x,\hat{\mathbf{w}})$[1]. However, for estimation of the weights $\mathbf{w}$, these methods minimize only the last term of J in Equation 6 (*i.e.*, the prediction error, $e^-(k)^2$).

When the model $f(\cdot)$ is linear, this is equivalent to maximizing the likelihood function $\rho_{\mathbf{Y|W}}$ (recall that $\rho_{\mathbf{YX|W}}$ is valid for nonlinear models as well). Although

[1]The *Clearning* method [11] proposed for on-line neural network training neglects the cross terms in the cost J for state estimation.

an advantage of the marginal $\rho_{\mathbf{Y}|\mathbf{W}}$ is that it can be expressed explicitly in terms of the maximum-likelihood estimates $\{\hat{x}(k)\}_1^N$, the variance of the weight estimate will always be higher than that of the estimate obtained from $\rho_{\mathbf{X}|\mathbf{W}}$.

The EM algorithm [2] also minimizes $\rho_{\mathbf{Y}|\mathbf{W}}$. However, rather than minimizing the squared prediction error $e^-(k)^2$, the EM algorithm operates on the fact that the log of the likelihood can be expanded as:

$$\log \rho_{\mathbf{Y}|\mathbf{W}} = \log \rho_{\mathbf{YX}|\mathbf{W}} - \log \rho_{\mathbf{X}|\mathbf{YW}}. \tag{7}$$

The conditional expectation is taken and a convexity argument is used to show that maximizing the *expectation* of the first term on the right results in a maximum of the log likelihood. In other words, in order to maximize $\rho_{\mathbf{Y}|\mathbf{W}}$, the EM algorithm maximizes $E[\log \rho_{\mathbf{YX}|\mathbf{W}}]$.

In the next section, we present an algorithm for the on-line minimization of the maximum-likelihood cost $J(x, \mathbf{w})$, with respect to both $\{x(k)\}_1^N$ and $\mathbf{w}$. This is followed by experimental results in Section 4.

III. Two-Observation Dual EKF

The on-line algorithm we present here takes the form of two extended Kalman filters run simultaneously: one for minimizing $J(x, \hat{\mathbf{w}})$ with respect to $\{x(k)\}_1^N$, and the other for minimizing $J(\hat{x}, \mathbf{w})$ with respect to $\mathbf{w}$.

A. State Estimation

When $\mathbf{w}$ is known and if $f(\cdot)$ is linear, the Kalman filter generates a series of estimates $\{\hat{x}(k)\}_1^N$ and predictions $\{\hat{x}^-(k)\}_1^N$, where each $\hat{x}(k)$ and $\hat{x}^-(k)$ is *causally* maximum-likelihood [2]. In other words, $\hat{x}(k)$ is found such that cost J is minimized up to the current time step k.

When $f(\cdot)$ is nonlinear, the *extended* Kalman filter is used, providing an approximate maximum-likelihood series $\{\hat{x}(k)\}_1^N$. The assumption is that the nonlinear system is modeled well by a time-varying linear one. To apply the EKF, we first put the autoregression of Equation 1 in state-space form:

$$\begin{aligned} \mathbf{x}(k) &= F[\mathbf{x}(k-1)] + Bv(k) \\ y(k) &= C\mathbf{x}(k) + n(k), \end{aligned} \tag{8}$$

where

$$\mathbf{x}(k) = \begin{bmatrix} x(k) \\ x(k-1) \\ \vdots \\ x(k-M+1) \end{bmatrix}, \quad B = \begin{bmatrix} 1 \\ 0 \\ \vdots \\ 0 \end{bmatrix},$$

$$F[\mathbf{x}(k)] = \begin{bmatrix} f(x(k), \ldots, x(k-M+1), \mathbf{w}) \\ x(k) \\ \vdots \\ x(k-M+2) \end{bmatrix},$$

and $C = B^T$. The EKF algorithm for estimating $\mathbf{x}(k)$ is then as follows:

$$\hat{\mathbf{x}}^-(k) = F[\hat{\mathbf{x}}(k-1), \mathbf{w}] \tag{9}$$

$$P_{\hat{\mathbf{x}}}^-(k) = A(k) P_{\hat{\mathbf{x}}}(k-1) A(k)^T + B\sigma_v^2 B^T \tag{10}$$

$$\text{where} \qquad A(k) \triangleq \frac{\partial F[\hat{\mathbf{x}}(k-1), \mathbf{w}]}{\partial \hat{\mathbf{x}}(k-1)} \tag{11}$$

$$K(k) = P_{\hat{\mathbf{x}}}^-(k) C^T \left(C P_{\hat{\mathbf{x}}}^-(k) C^T + \sigma_n^2 \right)^{-1} \tag{12}$$

$$P_{\hat{\mathbf{x}}}(k) = \left(I - K(k) C \right) P_{\hat{\mathbf{x}}}^-(k) \tag{13}$$

$$\hat{\mathbf{x}}(k) = \hat{\mathbf{x}}^-(k) + K(k) \left(y(k) - C\hat{\mathbf{x}}^-(k) \right). \tag{14}$$

Note, the derivative in Equation 11 corresponds to the linearization of $f(\cdot)$ at the current operation point. This can be found by a single application of standard back-propagation when $f(\cdot)$ is modeled as a neural network.

B. Weight Estimation

We now present a new recursive (on-line) algorithm for minimizing the cost J with respect to $\mathbf{w}$. We approach this problem by constructing a two-observation state-space formulation for the underlying weights as follows:

$$\begin{aligned} \mathbf{w}(k) &= \mathbf{w}(k-1) \\ \begin{bmatrix} y(k) \\ y(k) \end{bmatrix} &= \begin{bmatrix} x^-(k) \\ x(k) \end{bmatrix} + \begin{bmatrix} 1 & 1 \\ 1 & 0 \end{bmatrix} \begin{bmatrix} n(k) \\ v(k) \end{bmatrix}, \end{aligned} \tag{15}$$

where the state transition is simply an identity matrix, and the term $[x^-(k) \quad x(k)]^T$ plays the role of a time-varying nonlinear observation on $\mathbf{w}$. Letting $G \triangleq \left[\begin{smallmatrix} 1 & 1 \\ 1 & 0 \end{smallmatrix}\right]$, and $\Lambda \triangleq Cov([n(k) \ v(k)])$, these state-space equations for the weights allow us to estimate them with an EKF given by:

$$\hat{\mathbf{w}}^-(k) = \hat{\mathbf{w}}(k-1) \tag{16}$$

$$P_{\hat{\mathbf{w}}}^-(k) = P_{\hat{\mathbf{w}}}(k-1) \tag{17}$$

$$K_{\hat{\mathbf{w}}}(k) = P_{\hat{\mathbf{w}}}^-(k) H(k)^T \left(H(k) P_{\hat{\mathbf{w}}}^-(k) H(k)^T + G\Lambda G^T \right)^{-1} \tag{18}$$

$$P_{\hat{\mathbf{w}}}(k) = \left(I - K_{\hat{\mathbf{w}}}(k) H(k) \right) P_{\hat{\mathbf{w}}}^-(k) \tag{19}$$

$$\text{where} \quad H(k) \triangleq \left[\left. \frac{\partial x^-(k)}{\partial \hat{\mathbf{w}}} \right|_{\hat{\mathbf{w}}(k-1)}, \left. \frac{\partial x(k)}{\partial \hat{\mathbf{w}}} \right|_{\hat{\mathbf{w}}(k-1)} \right]^T \tag{20}$$

[2] For non-causal maximum-likelihood estimates, a forward-backward Kalman smoother may be used.

$$\hat{\mathbf{w}}(k) = \hat{\mathbf{w}}^-(k) + K_{\hat{\mathbf{w}}}(k)\left(\begin{bmatrix} y(k) \\ y(k) \end{bmatrix} - \begin{bmatrix} x^-(k) \\ x(k) \end{bmatrix}\right). \tag{21}$$

The EKF for weight estimation is an efficient second-order optimization method. The relationship between the EKF and Recursive Least Squares (RLS) can be used to show that this filter in fact minimizes the desired cost function, J.

The linearizations in Equation 20 are computed as *dynamic derivatives* (or full partial derivatives) to account for the recurrent nature of the state-estimation filter, including the dependence of the Kalman gain $K(k)$ on the weights. Computation of these derivatives accounts for a large part of the computational cost of the algorithm. The necessary equations are presented in Appendix A.

Note that the prediction error cost mentioned in Section 2 can be minimized with respect to $\mathbf{w}$ by using a single-observation form of the above EKF. This is the approach taken in our prior work on the Dual EKF [9].

C. State and Weight Estimation

Since $\mathbf{w}$ and $\{x(k)\}_1^N$ are not known in advance, the state EKF uses the current estimate of the weights, $\hat{\mathbf{w}}(k)$, and the weight EKF uses the current estimates of the state, $\hat{\mathbf{x}}(k)$. With each filter using the estimates produced by the other, the state EKF and weight EKF can be run in parallel over data until convergence. We will refer to this new algorithm as the Dual^2 EKF (shown in Figure 1). This name highlights the fact that the algorithm uses the *two*-observation form of the weight filter, as opposed to the standard Dual EKF, which uses the single-observation weight filter mentioned above.

Recall that this new weight filter provides a symmetry between the state and weight estimation, as both estimates attempt to minimize the same maximum-likelihood cost J at each time step. In contrast, the standard Dual EKF, like the other methods in the introduction, minimizes the prediction error for weight estimation, and J for state estimation.

IV. Experimental Results

We present results which demonstrate that on short data, using the cost function J improves performance over the usual prediction error cost. We compare the performance of the Dual^2 EKF algorithm against the standard Dual EKF. Because the algorithms differ primarily in the choice of cost function for weight estimation, differences in performance can more readily be attributed to this cost. However, similar performance gains can also be expected over other algorithms which also minimize only prediction error.

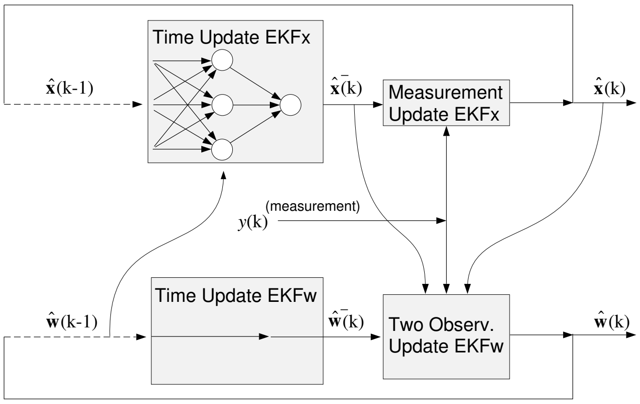

Fig. 1. The Two-Observation Dual Extend Kalman Filter (Dual^2 EKF). EKFx and EKFw represent the filters for the states and the weights, respectively.

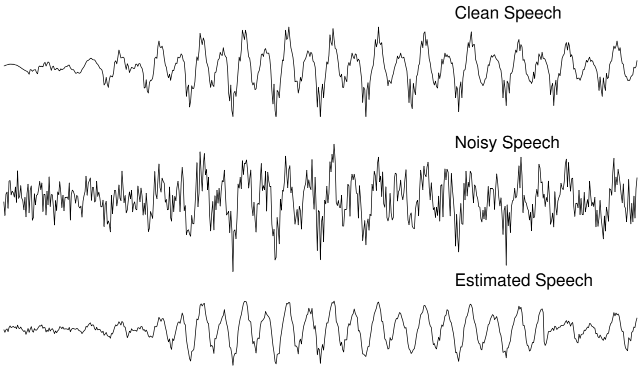

Fig. 2. Speech enhancement experiment. The Dual^2 EKF algorithm produced the result on the bottom.

A. Speech Data

The first data set we present is motivated by our interest in speech enhancement. Because of the nonstationarity of speech, it must be windowed into approximately stationary segments before the Dual EKF or Dual^2 EKF can be employed. For illustrative purposes, we have selected one such short segment of speech, consisting of 512 points of data (about 64 ms) representing the two phonemes /b/ and /I/.

Artificial stationary white noise is added to the clean speech signal to generate the noisy data (SNR 3.2 dB) provided to the algorithms. Signal-to-noise ratio improvements are shown in Table I. The estimated speech produced by the Dual^2 EKF algorithm is shown in Figure 2, along with the clean speech and noisy speech. Results for both algorithms were produced with a standard feedforward neural network (10 inputs, 8 hidden units, 1 output) where the variances σ_v^2 and σ_n^2 were assumed known. In addition, the convergence behavior is compared in Figure 3. These results indicate the superior performance of the Dual^2 EKF algorithm, which minimizes the full maximum-likelihood cost function.

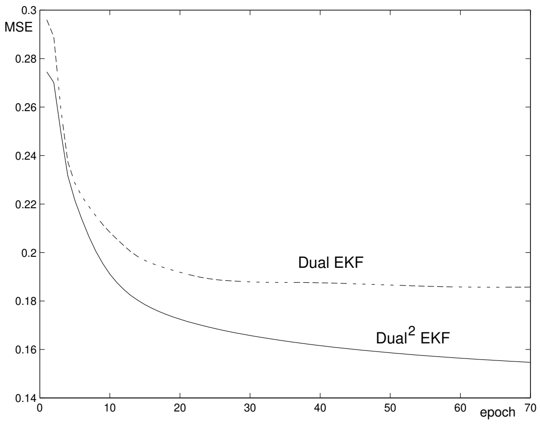

Fig. 3. Estimation MSE versus training epoch (speech data).

TABLE I

COMPARISON OF METHODS ON NOISY SPEECH DATA

Algorithm	SNR Improvement (dB)
Dual^2 EKF	4.934
Dual EKF	4.115

B. Noisy Mackey-Glass

In order to investigate the behavior of the algorithms on stochastic nonlinear time series, we generated a data set using the chaotic Mackey-Glass equations [5] with a delay parameter of 30. The series differs from the usual Mackey-Glass time series in that process noise (variance $\sigma_v^2 = 0.004$) was injected into the system equations. The continuous data was then sampled, and measurement noise with variance $\sigma_n^2 = 0.43$ was added to produce the noisy time series $\{y(k)\}_1^N$ with a normalized mean squared error of 0.5 (SNR equals 3dB).

Results for all algorithms were produced using a feedforward neural network (12 inputs, 8 hidden units, 1 output). Algorithms were trained on 500 points of data, and tested on the subsequent 1500 points. On the test-set, the standard EKF algorithm was used to generate predictions and estimations using the parameters learned by either the Dual^2 EKF or the Dual EKF in the training set.

The results produced by the Dual^2 EKF algorithm are shown in Figure 4. Normalized mean squared errors (both in and out of training set) are shown in Table II. While the Dual^2 EKF performance is better in all columns, the most noticeable difference is in the training set estimation performance. This is expected, since the weight estimation in the standard Dual EKF minimizes only the prediction error, which results in more rapid overfitting (as illustrated in Figure 5).

C. Discussion of Results

We have performed numerous experiments on different data sets with varying initial conditions. Although

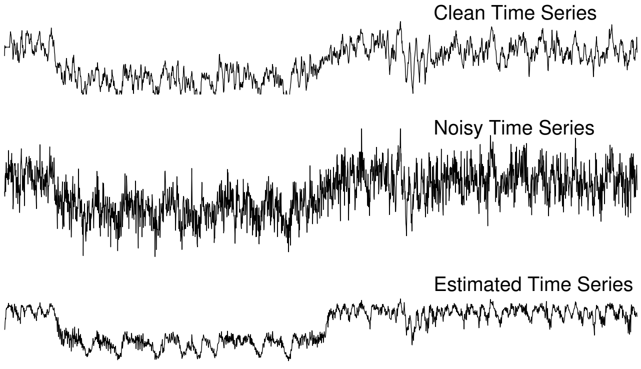

Fig. 4. Noisy Mackey-Glass Estimation. The Dual^2 EKF algorithm produced the result on the bottom, where the first quarter of the data was "in-sample," and the rest was used as a test set ($\hat{\mathbf{w}}$ fixed).

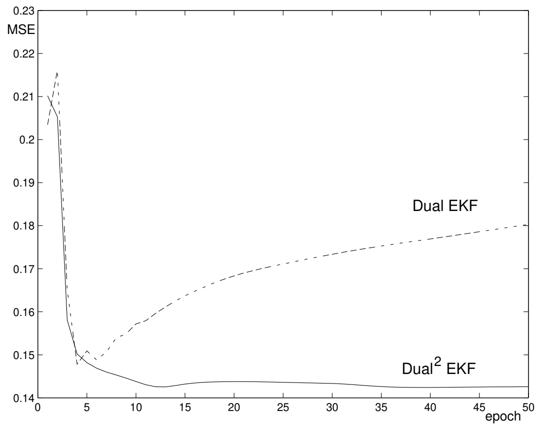

Fig. 5. In-sample Estimation MSE (Mackey-Glass data).

the Dual^2 EKF provides consistently better results than the Dual EKF on short data sets ($N < 1000$), results may not be superior on longer data sets.

Our hypothesis for this observation was alluded to earlier in this paper. Because the two-observation weight filter tries to maximize $\rho_{\mathbf{YX|W}}$, it should produce estimates $\hat{\mathbf{w}}$ with lower variance than those produced by the single-observation weight filter, which tries to maximize $\rho_{\mathbf{Y|W}}$. However, as the length of the training data is increased, the variances of the solutions produced by the two methods become nearly equivalent. In addition, the two-observation weight filter may be slightly biased by the replacement of x with $\hat{x}$ in the likelihood function $\rho_{\mathbf{YX|W}}$. Consequently, the Dual^2 EKF will tend to perform better on problems with

TABLE II

COMPARISON OF METHODS ON MACKEY-GLASS DATA

	Train NMSE		**Test NMSE**	
Algorithm	Estimat.	Predict.	Estimat.	Predict.
Dual^2 EKF	0.1421	0.1718	0.2384	0.3399
Dual EKF	0.1815	0.1995	0.2533	0.3813

higher variance in the weight estimate, $\hat{\mathbf{w}}$. (*e.g.*, shorter data sets). Another point of view is that J introduces additional constraints on the weight estimation, which reduces overfitting (as illustrated in Mackey-Glass example). We are currently working on a more formal analysis of these issues.

V. Conclusions

A maximum-likelihood cost function was presented for the dual state estimation and model estimation of noisy time series. We then presented the Dual^2 EKF method as an on-line algorithm for efficient minimization of this cost. The algorithm involves an EKF for maximum-likelihood estimation of the states, and a two-observation EKF for the maximum-likelihood estimation of the weights. While preliminary in nature, the results presented here indicate that on short data sets, significant advantages can be obtained over previous methods by this approach.

Acknowledgments.

This work was sponsored in part by the NSF under grant ECS-9410823, and grant IRI-9712346.

Appendix A: Dynamic Derivatives

Note that the derivatives in Equation 20, given by

$$H(k) = \left[C\frac{\partial \mathbf{x}^-(k)}{\partial \hat{\mathbf{w}}}\bigg|_{\hat{\mathbf{w}}(k-1)}, C\frac{\partial \mathbf{x}(k)}{\partial \hat{\mathbf{w}}}\bigg|_{\hat{\mathbf{w}}(k-1)} \right]^T$$

are computed recursively because both $\hat{\mathbf{x}}(k)$ and $\hat{\mathbf{x}}^-(k)$ are recursive functions of $\mathbf{w}$. This results in the following system of recursive equations:

$$\frac{\partial \hat{\mathbf{x}}^-(k+1)}{\partial \hat{\mathbf{w}}} = \frac{\partial F[\hat{\mathbf{x}}, \hat{\mathbf{w}}]}{\partial \hat{\mathbf{x}}(k)}\frac{\partial \hat{\mathbf{x}}(k)}{\partial \hat{\mathbf{w}}} + \frac{\partial F[\hat{\mathbf{x}}, \hat{\mathbf{w}}]}{\partial \hat{\mathbf{w}}(k)}, \tag{22}$$

$$\frac{\partial \hat{\mathbf{x}}(k)}{\partial \hat{\mathbf{w}}} = \big(I - K(k)C\big)\frac{\partial \hat{\mathbf{x}}^-(k)}{\partial \hat{\mathbf{w}}} + \frac{\partial K(k)}{\partial \hat{\mathbf{w}}}\big(y(k) - C\hat{\mathbf{x}}^-(k)\big), \tag{23}$$

where $\frac{\partial F[\hat{\mathbf{x}},\hat{\mathbf{w}}]}{\partial \hat{\mathbf{x}}(k)}$ and $\frac{\partial F[\hat{\mathbf{x}},\hat{\mathbf{w}}]}{\partial \hat{\mathbf{w}}(k)}$ again represent static linearizations of the neural network. The last term in Equation 23 may be dropped if we assume the Kalman gain K is independent of $\mathbf{w}$. This greatly simplifies the algorithm. However, for accurate computation of the dynamic derivatives, $\frac{\partial K(k)}{\partial \hat{\mathbf{w}}}$ is calculated as follows. The i^{th} column of $\frac{\partial K(k)}{\partial \hat{\mathbf{w}}}$ is the derivative of $K(k)$ with respect to the i^{th} element of $\hat{\mathbf{w}}$, and is given by:

$$\frac{\partial K(k)}{\partial \hat{\mathbf{w}}^{(i)}} = \frac{1}{CP^-_{\hat{\mathbf{x}}}(k)C^T + \sigma_n^2}\big(I - K(k)C\big)\frac{\partial P^-_{\hat{\mathbf{x}}}(k)}{\partial \hat{\mathbf{w}}^{(i)}}C^T, \tag{24}$$

$$\begin{aligned}\frac{\partial P^-_{\hat{\mathbf{x}}}(k)}{\partial \hat{\mathbf{w}}^{(i)}} &= \frac{\partial A(k-1)}{\partial \hat{\mathbf{w}}^{(i)}}P^-_{\hat{\mathbf{x}}}(k-1)A^T(k-1) \\ &+ A(k-1)\frac{\partial P^-_{\hat{\mathbf{x}}}(k-1)}{\partial \hat{\mathbf{w}}^{(i)}}A^T(k-1) \\ &+ A(k-1)P^-_{\hat{\mathbf{x}}}(k-1)\frac{\partial A(k-1)}{\partial \hat{\mathbf{w}}^{(i)}} \\ &- \left(\frac{\partial A(k-1)}{\partial \hat{\mathbf{w}}^{(i)}}K(k-1) + A(k-1)\frac{\partial K(k-1)}{\partial \hat{\mathbf{w}}^{(i)}}\right) \\ &\qquad \cdot CP^-_{\hat{\mathbf{x}}}(k-1)A^T(k-1) \\ &- A(k-1)K(k-1)C \\ &\quad \cdot \left(\frac{\partial P^-_{\hat{\mathbf{x}}}(k)}{\partial \hat{\mathbf{w}}^{(i)}}A^T(k-1) + P^-_{\hat{\mathbf{x}}}(k-1)\frac{\partial A(k-1)}{\partial \hat{\mathbf{w}}^{(i)}}\right).\end{aligned} \tag{25}$$

Note that $A(k-1)$ depends not only on the weights $\hat{\mathbf{w}}$, but also on the point of linearization, $\hat{\mathbf{x}}(k-1)$. Thus,

$$\frac{\partial A(k-1)}{\partial \hat{\mathbf{w}}^{(i)}} = \frac{\partial A(k-1)}{\partial \hat{\mathbf{w}}^{(i)}(k-1)} + \frac{\partial A(k-1)}{\partial \hat{\mathbf{x}}(k-1)}\frac{\partial \hat{\mathbf{x}}(k-1)}{\partial \hat{\mathbf{w}}^{(i)}}, \tag{26}$$

which represent a recursive equation for the derivative of the linearization of the neural network. The largest computational expense is incurred by the calculation of $\frac{\partial K(k)}{\partial \hat{\mathbf{w}}}$, which requires that $\frac{\partial P^-_{\hat{\mathbf{x}}}(k)}{\partial \hat{\mathbf{w}}^{(i)}}$ be computed for all $i \in \{1 \ldots \dim(\hat{\mathbf{w}})\}$. Whether this expense is worth the improvement in performance is clearly a design issue.

References

[1] J. T. Connor, R. D. Martin, and L. E. Atlas. Recurrent neural networks and robust time series prediction. *IEEE Transactions on Neural Networks*, March 1994.

[2] A. Dempster, N. M. Laird, and D. Rubin. Maximum-likelihood from incomplete data via the EM algorithm. *Journal of the Royal Statistical Society*, B39:1–38, 1977.

[3] J. S. Lim and A. V. Oppenheim. All-pole modeling of degraded speech. *IEEE Tr. ASSP*, 26(3):197–210, June 1978.

[4] L. Ljung and T. Söderström. *Theory and Practice of Recursive Identification.* MIT Press, Cambridge, MA, 1983.

[5] M. Mackey and L. Glass. Oscillations and chaos in physiological control systems. *Science*, 197(4300):287–9, 1977.

[6] M. B. Matthews and G. S. Moschytz. The identification of nonlinear discrete-time fading-memory systems using neural network models. *IEEE Transactions on Circuits and Systems-II*, 41(11):740–51, November 1994.

[7] L. W. Nelson and E. Stear. The simultaneous on-line estimation of parameters and states in linear systems. *IEEE Trans. on Automatic Control*, pages 94–8, Feb. 1976.

[8] G. Seber and C. Wild. *Nonlinear Regression.* John Wiley & Sons, 1989.

[9] E. A. Wan and A. T. Nelson. Dual Kalman filtering methods for nonlinear prediction, estimation, and smoothing. In *Advances in Neural Information Processing Systems 9*, 1997.

[10] E. A. Wan and A. T. Nelson. Neural dual extended Kalman filtering: Applications in speech enhancement and monaural blind signal separation. In *IEEE Workshop on Neural Networks for Signal Processing*, 1997.

[11] A. Weigend, H. G. Zimmermann, and R. Neuneier. Clearning. Technical Report CU-CS-772-95, University of Colorado Dept. of Computer Science, Boulder, CO, May 1995.

[12] R. J. Williams. Training recurrent networks using the extended Kalman filter. In *Proceedings International Joint Conference on Neural Networks*, volume 4, pages 241–6. IEEE and INNS, 1992.