Kalman Filtering and Neural Networks, Edited by Simon Haykin
Copyright © 2001 John Wiley & Sons, Inc.
ISBNs: 0-471-36998-5 (Hardback); 0-471-22154-6 (Electronic)

5

DUAL EXTENDED KALMAN FILTER METHODS

Eric A. Wan and Alex T. Nelson
Department of Electrical and Computer Engineering, Oregon Graduate Institute of Science and Technology, Beaverton, Oregon, U.S.A.

5.1 INTRODUCTION

The Extended Kalman Filter (EKF) provides an efficient method for generating approximate maximum-likelihood estimates of the state of a discrete-time nonlinear dynamical system (see Chapter 1). The filter involves a recursive procedure to optimally combine noisy observations with predictions from the known dynamic model. A second use of the EKF involves estimating the parameters of a model (e.g., neural network) given clean training data of input and output data (see Chapter 2). In this case, the EKF represents a modified-Newton type of algorithm for on-line system identification. In this chapter, we consider the *dual estimation* problem, in which both the states of the dynamical system and its parameters are estimated simultaneously, given only noisy observations.

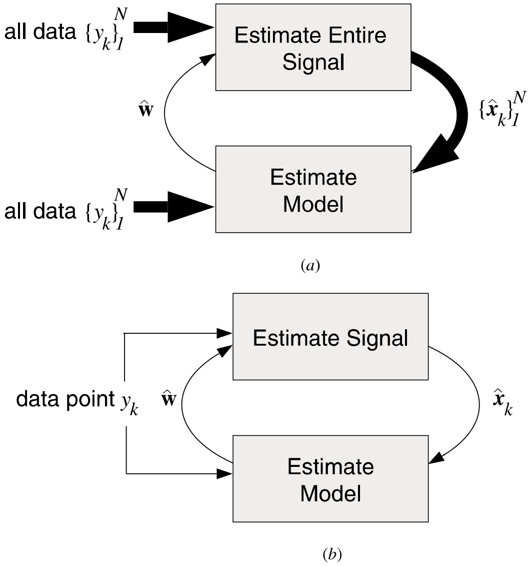
To be more specific, we consider the problem of learning both the hidden states $\mathbf{x}_k$ and parameters $\mathbf{w}$ of a discrete-time nonlinear dynamical system,

$$\begin{aligned} \mathbf{x}_{k+1} &= \mathbf{F}(\mathbf{x}_k, \mathbf{u}_k, \mathbf{w}) + \mathbf{v}_k, \\ \mathbf{y}_k &= \mathbf{H}(\mathbf{x}_k, \mathbf{w}) + \mathbf{n}_k, \end{aligned} \tag{5.1}$$

where both the system states $\mathbf{x}_k$ and the set of model parameters $\mathbf{w}$ for the dynamical system must be simultaneously estimated from only the observed noisy signal $\mathbf{y}_k$. The *process* noise $\mathbf{v}_k$ drives the dynamical system, *observation* noise is given by $\mathbf{n}_k$, and $\mathbf{u}_k$ corresponds to observed exogenous inputs. The model structure, $\mathbf{F}(\cdot)$ and $\mathbf{H}(\cdot)$, may represent multilayer neural networks, in which case $\mathbf{w}$ are the weights.

The problem of dual estimation can be motivated either from the need for a model to estimate the signal or (in other applications) from the need for good signal estimates to estimate the model. In general, applications can be divided into the tasks of modeling, estimation, and prediction. In *estimation*, all noisy data up to the current time is used to approximate the current value of the clean state. *Prediction* is concerned with using all available data to approximate a *future* value of the clean state. *Modeling* (sometimes referred to as *identification*) is the process of approximating the underlying dynamics that generated the states, again given only the noisy observations. Specific applications may include noise reduction (e.g., speech or image enhancement), or prediction of financial and economic time series. Alternatively, the model may correspond to the explicit equations derived from first principles of a robotic or vehicle system. In this case, $\mathbf{w}$ corresponds to a set of unknown parameters. Applications include adaptive control, where parameters are used in the design process and the estimated states are used for feedback.

Heuristically, dual estimation methods work by alternating between using the model to estimate the signal, and using the signal to estimate the model. This process may be either *iterative* or *sequential*. *Iterative* schemes work by repeatedly estimating the signal using the current model and all available data, and then estimating the model using the estimates and all the data (see Fig. 5.1*a*). Iterative schemes are necessarily restricted to off-line applications, where a batch of data has been previously collected for processing. In contrast, *sequential* approaches use each individual measurement as soon as it becomes available to update both the signal and model estimates. This characteristic makes these algorithms useful in either on-line or off-line applications (see Fig. 5.1*b*).



(*a*)

(*b*)

Figure 5.1 Two approaches to the dual estimation problem. (*a*) Iterative approaches use large blocks of data repeatedly. (*b*) Sequential approaches are designed to pass over the data one point at a time.

The vast majority of work on dual estimation has been for linear models. In fact, one of the first applications of the EKF combines both the state vector $\mathbf{x}_k$ and unknown parameters $\mathbf{w}$ in a joint bilinear state-space representation. An EKF is then applied to the resulting nonlinear estimation problem [1, 2]; we refer to this approach as the *joint extended Kalman filter*. Additional improvements and analysis of this approach are provided in [3, 4]. An alternative approach, proposed in [5], uses two separate Kalman filters: one for signal estimation, and another for model estimation. The signal filter uses the current estimate of $\mathbf{w}$, and the weight filter uses the signal estimates $\hat{\mathbf{x}}_k$ to minimize a prediction error cost. In [6], this *dual Kalman* approach is placed in a general family of recursive prediction error algorithms. Apart from these sequential approaches, some iterative methods developed for linear models include maximum-likelihood approaches [7–9] and expectation-maximization (EM) algorithms [10–13]. These algorithms are suitable only for off-line applications, although sequential EM methods have been suggested.

Fewer papers have appeared in the literature that are explicitly concerned with dual estimation for nonlinear models. One algorithm (proposed in [14]) alternates between applying a robust form of the

EKF to estimate the time-series and using these estimates to train a neural network via gradient descent. A joint EKF is used in [15] to model partially unknown dynamics in a model reference adaptive control framework. Furthermore, iterative EM approaches to the dual estimation problem have been investigated for radial basis function networks [16] and other nonlinear models [17]; see also Chapter 6. Errors-in-variables (EIV) models appear in the nonlinear statistical regression literature [18], and are used for regressing on variables related by a nonlinear function, but measured with some error. However, errors-in-variables is an iterative approach involving batch computation; it tends not to be practical for dynamical systems because the computational requirements increase in proportion to N^2, where N is the length of the data. A heuristic method known as *Clearning* minimizes a simplified approximation to the EIV cost function. While it allows for sequential estimation, the simplification can lead to severely biased results [19]. The dual EKF [19] is a nonlinear extension of the linear dual Kalman approach of [5], and recursive prediction error algorithm of [6]. Application of the algorithm to speech enhancement appears in [20], while extensions to other cost functions have been developed in [21] and [22]. The crucial, but often overlooked issue of sequential variance estimation is also addressed in [22].

Overview The goal of this chapter is to present a unified probabilistic and algorithmic framework for nonlinear dual estimation methods. In the next section, we start with the basic dual EKF prediction error method. This approach is the most intuitive, and involves simply running two EKF filters in parallel. The section also provides a quick review of the EKF for both state and weight estimation, and introduces some of the complications in coupling the two. An example in noisy time-series prediction is also given. In Section 5.3, we develop a general probabilistic framework for dual estimation. This allows us to relate the various methods that have been presented in the literature, and also provides a general algorithmic approach leading to a number of different dual EKF algorithms. Results on additional example data sets are presented in Section 5.5.

5.2 DUAL EKF–PREDICTION ERROR

In this section, we present the basic dual EKF prediction error algorithm. For completeness, we start with a quick review of the EKF for state estimation, followed by a review of EKF weight estimation (see Chapters

1 and 2 for more details). We then discuss coupling the state and weight filters to form the dual EKF algorithm.

5.2.1 EKF–State Estimation

For a *linear* state-space system with *known* model and Gaussian noise, the Kalman filter [23] generates optimal estimates and predictions of the state $\mathbf{x}_k$. Essentially, the filter recursively updates the (posterior) mean $\hat{\mathbf{x}}_k$ and covariance $\mathbf{P}_{\mathbf{x}_k}$ of the state by combining the predicted mean $\hat{\mathbf{x}}_k^-$ and covariance $\mathbf{P}_{\mathbf{x}_k}^-$ with the current noisy measurement $\mathbf{y}_k$. These estimates are optimal in both the MMSE and MAP senses. Maximum-likelihood signal estimates are obtained by letting the initial covariance $\mathbf{P}_{\mathbf{x}_0}$ approach infinity, thus causing the filter to ignore the value of the initial state $\hat{\mathbf{x}}_0$.

For nonlinear systems, the extended Kalman filter provides *approximate* maximum-likelihood estimates. The mean and covariance of the state are again recursively updated; however, a first-order linearization of the dynamics is necessary in order to analytically propagate the Gaussian random-variable representation. Effectively, the nonlinear dynamics are approximated by a time-varying linear system, and the linear Kalman filters equations are applied. The full set of equations are given in Table 5.1. While there are more accurate methods for dealing with the nonlinear dynamics (e.g., particle filters [24, 25], second-order EKF, etc.), the standard EKF remains the most popular approach owing to its simplicity. Chapter 7 investigates the use of the unscented Kalman filter as a potentially superior alternative to the EKF [26–29].

Another interpretation of Kalman filtering is that of an optimization algorithm that recursively determines the state $\mathbf{x}_k$ in order to minimize a cost function. It can be shown that the cost function consists of a weighted prediction error and estimation error components given by

$$\begin{aligned} J(\mathbf{x}_1^k) = \sum_{t=1}^{k} \{ & [\mathbf{y}_t - \mathbf{H}(\mathbf{x}_t, \mathbf{w})]^T (\mathbf{R}^\mathbf{n})^{-1} [\mathbf{y}_t - \mathbf{H}(\mathbf{x}_t, \mathbf{w})] \\ & + (\mathbf{x}_t - \mathbf{x}_t^-)^T (\mathbf{R}^\mathbf{v})^{-1} (\mathbf{x}_t - \mathbf{x}_t^-) \} \end{aligned} \tag{5.10}$$

where $\mathbf{x}_t^- = \mathbf{F}(\mathbf{x}_{t-1}, \mathbf{w})$ is the predicted state, and $\mathbf{R}^\mathbf{n}$ and $\mathbf{R}^\mathbf{v}$ are the additive noise and innovations noise covariances, respectively. This interpretation will be useful when dealing with alternate forms of the dual EKF in Section 5.3.3.

Table 5.1 Extended Kalman filter (EKF) equations

Initialize with:

$$\hat{\mathbf{x}}_0 = E[\mathbf{x}_0], \tag{5.2}$$

$$\mathbf{P}_{\mathbf{x}_0} = E[(\mathbf{x}_0 - \hat{\mathbf{x}}_0)(\mathbf{x}_0 - \hat{\mathbf{x}}_0)^T]. \tag{5.3}$$

For $k \in \{1, \ldots, \infty\}$, the time-update equations of the extended Kalman filter are

$$\hat{\mathbf{x}}_k^- = \mathbf{F}(\hat{\mathbf{x}}_{k-1}, \mathbf{u}_k, \mathbf{w}), \tag{5.4}$$

$$\mathbf{P}_{\mathbf{x}_k}^- = \mathbf{A}_{k-1}\mathbf{P}_{\mathbf{x}_{k-1}}\mathbf{A}_{k-1}^T + \mathbf{R}^{\mathrm{v}}, \tag{5.5}$$

and the measurement-update equations are

$$\mathbf{K}_k^{\mathbf{x}} = \mathbf{P}_{\mathbf{x}_k}^-\mathbf{C}_k^T(\mathbf{C}_k\mathbf{P}_{\mathbf{x}_k}^-\mathbf{C}_k^T + \mathbf{R}^{\mathrm{n}})^{-1}, \tag{5.6}$$

$$\hat{\mathbf{x}}_k = \hat{\mathbf{x}}_k^- + \mathbf{K}_k^{\mathbf{x}}[\mathbf{y}_k - \mathbf{H}(\hat{\mathbf{x}}_k^-, \mathbf{w})], \tag{5.7}$$

$$\mathbf{P}_{\mathbf{x}_k} = (\mathbf{I} - \mathbf{K}_k^{\mathbf{x}}\mathbf{C}_k)\mathbf{P}_{\mathbf{x}_k}^-, \tag{5.8}$$

where

$$\mathbf{A}_k \triangleq \left.\frac{\partial \mathbf{F}(\mathbf{x}, \mathbf{u}_k, \mathbf{w})}{\partial \mathbf{x}}\right|_{\hat{\mathbf{x}}_k}, \qquad \mathbf{C}_k \triangleq \left.\frac{\partial \mathbf{H}(\mathbf{x}, \mathbf{w})}{\partial \mathbf{x}}\right|_{\hat{\mathbf{x}}_k}, \tag{5.9}$$

and where $\mathbf{R}^{\mathrm{v}}$ and $\mathbf{R}^{\mathrm{n}}$ are the covariances of $\mathbf{v}_k$ and $\mathbf{n}_k$, respectively.

5.2.2 EKF–Weight Estimation

As proposed initially in [30], and further developed in [31] and [32], the EKF can also be used for estimating the parameters of nonlinear models (i.e., training neural networks) from *clean* data. Consider the general problem of learning a mapping using a parameterized nonlinear function $\mathbf{G}(\mathbf{x}_k, \mathbf{w})$. Typically, a training set is provided with sample pairs consisting of known input and desired output, $\{\mathbf{x}_k, \mathbf{d}_k\}$. The error in the model is defined as $\mathbf{e}_k = \mathbf{d}_k - \mathbf{G}(\mathbf{x}_k, \mathbf{w})$, and the goal of learning involves solving for the parameters $\mathbf{w}$ in order to minimize the expected squared error. The EKF may be used to estimate the parameters by writing a new state-space representation

$$\mathbf{w}_{k+1} = \mathbf{w}_k + \mathbf{r}_k, \tag{5.11}$$

$$\mathbf{d}_k = \mathbf{G}(\mathbf{x}_k, \mathbf{w}_k) + \mathbf{e}_k, \tag{5.12}$$

where the parameters $\mathbf{w}_k$ correspond to a stationary process with identity state transition matrix, driven by process noise $\mathbf{r}_k$. The output $\mathbf{d}_k$

Table 5.2 The extended Kalman weight filter equations

Initialize with:

$$\hat{\mathbf{w}}_0 = E[\mathbf{w}] \tag{5.13}$$

$$\mathbf{P}_{\mathbf{w}_0} = E[(\mathbf{w} - \hat{\mathbf{w}}_0)(\mathbf{w} - \hat{\mathbf{w}}_0)^T] \tag{5.14}$$

For $k \in \{1, \ldots, \infty\}$, the time update equations of the Kalman filter are:

$$\hat{\mathbf{w}}_k^- = \hat{\mathbf{w}}_{k-1} \tag{5.15}$$

$$\mathbf{P}_{\mathbf{w}_k}^- = \mathbf{P}_{\mathbf{w}_{k-1}} + \mathbf{R}_{k-1}^{\mathbf{r}} \tag{5.16}$$

and the measurement update equations:

$$\mathbf{K}_k^{\mathbf{w}} = \mathbf{P}_{\mathbf{w}_k}^- (\mathbf{C}_k^{\mathbf{w}})^T (\mathbf{C}_k^{\mathbf{w}} \mathbf{P}_{\mathbf{w}_k}^- (\mathbf{C}_k^{\mathbf{w}})^T + \mathbf{R}^{\mathbf{e}})^{-1} \tag{5.17}$$

$$\hat{\mathbf{w}}_k = \hat{\mathbf{w}}_k^- + \mathbf{K}_k^{\mathbf{w}} (\mathbf{d}_k - \mathbf{G}(\hat{\mathbf{w}}_k^-, \mathbf{x}_{k-1})) \tag{5.18}$$

$$\mathbf{P}_{\mathbf{w}_k} = (\mathbf{I} - \mathbf{K}_k^{\mathbf{w}} \mathbf{C}_k^{\mathbf{w}}) \mathbf{P}_{\mathbf{w}_k}^- . \tag{5.19}$$

where

$$\mathbf{C}_k^{\mathbf{w}} \triangleq \left. \frac{\partial \mathbf{G}(\mathbf{x}_{k-1}, \mathbf{w})^T}{\partial \mathbf{w}} \right|_{\mathbf{w} = \hat{\mathbf{w}}_k^-} \tag{5.20}$$

corresponds to a nonlinear observation on $\mathbf{w}_k$. The EKF can then be applied directly, with the equations given in Table 5.2. In the linear case, the relationship between the Kalman filter (KF) and the popular recursive least-squares (RLS) is given [33] and [34]. In the nonlinear case, the EKF training corresponds to a modified-Newton optimization method [22].

As an optimization approach, the EKF minimizes the *prediction error* cost:

$$J(\mathbf{w}) = \sum_{t=1}^{k} [\mathbf{d}_t - \mathbf{G}(\mathbf{x}_t, \mathbf{w})]^T (\mathbf{R}^{\mathbf{e}})^{-1} [\mathbf{d}_t - \mathbf{G}(\mathbf{x}_t, \mathbf{w})]. \tag{5.21}$$

If the "noise" covariance $\mathbf{R}^{\mathbf{e}}$ is a constant diagonal matrix, then, in fact, it cancels out of the algorithm (this can be shown explicitly), and hence can be set arbitrarily (e.g., $\mathbf{R}^{\mathbf{e}} = 0.5\mathbf{I}$). Alternatively, $\mathbf{R}^{\mathbf{e}}$ can be set to specify a weighted MSE cost. The innovations covariance $E[\mathbf{r}_k \mathbf{r}_k^T] = \mathbf{R}_k^{\mathbf{r}}$, on the other hand, affects the convergence rate and tracking performance. Roughly speaking, the larger the covariance, the more quickly older data are discarded. There are several options on how to choose $\mathbf{R}_k^{\mathbf{r}}$:

- Set $\mathbf{R}_k^{\mathbf{r}}$ to an arbitrary diagonal value, and anneal this towards zeroes as training continues.

- Set $\mathbf{R}_k^{\mathbf{r}} = (\lambda^{-1} - 1)\mathbf{P}_{\mathbf{w}_k}$, where $\lambda \in (0, 1]$ is often referred to as the "forgetting factor." This provides for an approximate exponentially decaying weighting on past data and is described more fully in [22].
- Set $\mathbf{R}_k^{\mathbf{r}} = (1 - \alpha)\mathbf{R}_{k-1}^{\mathbf{r}} + \alpha \mathbf{K}_k^{\mathbf{w}}[\mathbf{d}_k - \mathbf{G}(\mathbf{x}_k, \hat{\mathbf{w}})][\mathbf{d}_k - \mathbf{G}(\mathbf{x}_k, \hat{\mathbf{w}})]^T(\mathbf{K}_k^{\mathbf{w}})^T$, which is a Robbins–Monro stochastic approximation scheme for estimating the innovations [6]. The method assumes that the covariance of the Kalman update model is consistent with the actual update model.

Typically, $\mathbf{R}_k^{\mathbf{r}}$ is also constrained to be a diagonal matrix, which implies an independence assumption on the parameters.

Study of the various trade-offs between these different approaches is still an area of open research. For the experiments performed in this chapter, the forgetting factor approach is used.

Returning to the dynamic system of Eq. (5.1), the EKF weight filter can be used to estimate the model parameters for either $\mathbf{F}$ or $\mathbf{H}$. To learn the state dynamics, we simply make the substitutions $\mathbf{G} \rightarrow \mathbf{F}$ and $\mathbf{d}_k \rightarrow \mathbf{x}_{k+1}$. To learn the measurement function, we make the substitutions $\mathbf{G} \rightarrow \mathbf{H}$ and $\mathbf{d}_k \rightarrow \mathbf{y}_k$. Note that for both cases, it is assumed that the noise-free state $\mathbf{x}_k$ is available for training.

5.2.3 Dual Estimation

When the clean state is *not* available, a dual estimation approach is required. In this section, we introduce the basic dual EKF algorithm, which combines the Kalman state and weight filters. Recall that the task is to estimate *both* the state and model from only noisy observations. Essentially, two EKFs are run concurrently. At every time step, an EKF state filter estimates the state using the current model estimate $\hat{\mathbf{w}}_k$, while the EKF weight filter estimates the weights using the current state estimate $\hat{\mathbf{x}}_k$. The system is shown schematically in Figure 5.2. In order to simplify the presentation of the equations, we consider the slightly less general state-space model:

$$\mathbf{x}_{k+1} = \mathbf{F}(\mathbf{x}_k, \mathbf{u}_k, \mathbf{w}) + \mathbf{v}_k, \tag{5.22}$$

$$y_k = \mathbf{C}\mathbf{x}_k + n_k, \qquad \mathbf{C} = [1 \quad 0 \quad \ldots \quad 0], \tag{5.23}$$

in which we take the scalar observation y_k to be one of the states. Thus, we only need to consider estimating the parameters associated with a single

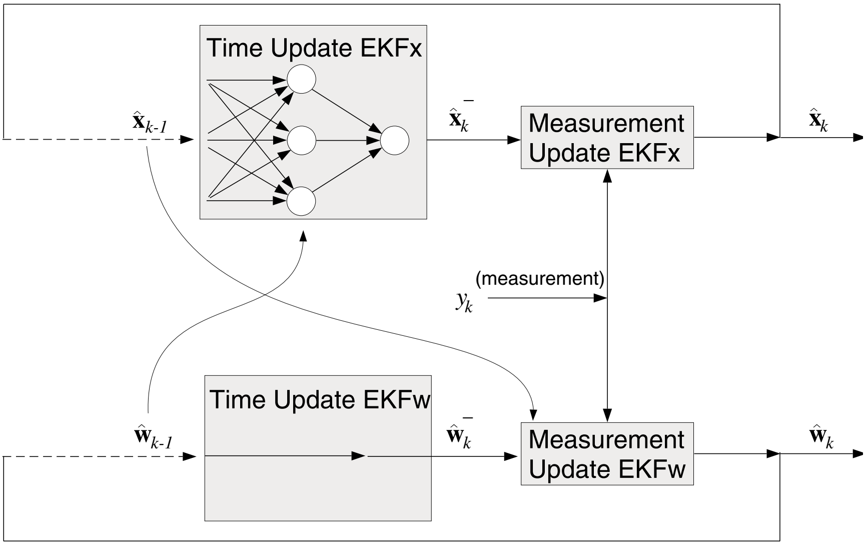


Figure 5.2 The dual extended Kalman filter. The algorithm consists of two EKFs that run concurrently. The top EKF generates state estimates, and requires $\hat{\mathbf{w}}_{k-1}$ for the time update. The bottom EKF generates weight estimates, and requires $\hat{\mathbf{x}}_{k-1}$ for the measurement update.

nonlinear function $\mathbf{F}$. The dual EKF equations for this system are presented in Table 5.3. Note that for clarity, we have specified the equations for the additive white-noise case. The case of colored measurement noise n_k is treated in Appendix B.

Recurrent Derivative Computation While the dual EKF equations appear to be a simple concatenation of the previous state and weight EKF equations, there is actually a necessary modification of the linearization $\mathbf{C}_k^{\mathbf{w}} = \mathbf{C}\partial\hat{\mathbf{x}}_k^- / \partial\hat{\mathbf{w}}_k^-$ associated with the weight filter. This is due to the fact that the signal filter, whose parameters are being estimated by the weight filter, has a recurrent architecture, i.e., $\hat{\mathbf{x}}_k$ is a function of $\hat{\mathbf{x}}_{k-1}$, and both are functions of $\mathbf{w}$.[1] Thus, the linearization must be computed using recurrent derivatives with a routine similar to real-time recurrent learning

[1] Note that a linearization is also required for the state EKF, but this derivative, $\partial\mathbf{F}(\hat{\mathbf{x}}_{k-1}, \hat{\mathbf{w}}_k^-)/\partial\hat{\mathbf{x}}_{k-1}$, can be computed with a simple technique (such as backpropagation) because $\hat{\mathbf{w}}_k^-$ is not itself a function of $\hat{\mathbf{x}}_{k-1}$.

Table 5.3 The dual extended Kalman filter equations. The definitions of $\mathfrak{e}_k$ and $\mathbf{C}_k^{\mathbf{w}}$ depend on the particular form of the weight filter being used. See the text for details

Initialize with:

$$\hat{\mathbf{w}}_0 = E[\mathbf{w}], \qquad \mathbf{P}_{\mathbf{w}_0} = E[(\mathbf{w} - \hat{\mathbf{w}}_0)(\mathbf{w} - \hat{\mathbf{w}}_0)^T],$$

$$\hat{\mathbf{x}}_0 = E[\mathbf{x}_0], \qquad \mathbf{P}_{\mathbf{x}_0} = E[(\mathbf{x}_0 - \hat{\mathbf{x}}_0)(\mathbf{x}_0 - \hat{\mathbf{x}}_0)^T].$$

For $k \in \{1, \ldots, \infty\}$, the time-update equations for the weight filter are

$$\hat{\mathbf{w}}_k^- = \hat{\mathbf{w}}_{k-1}, \tag{5.24}$$

$$\mathbf{P}_{\mathbf{w}_k}^- = \mathbf{P}_{\mathbf{w}_{k-1}} + \mathbf{R}_{k-1}^{\mathbf{r}} = \lambda^{-1}\mathbf{P}_{\mathbf{w}_{k-1}}, \tag{5.25}$$

and those for the state filter are

$$\hat{\mathbf{x}}_k^- = \mathbf{F}(\hat{\mathbf{x}}_{k-1}\mathbf{u}_k, \hat{\mathbf{w}}_k^-), \tag{5.26}$$

$$\mathbf{P}_{\mathbf{x}_k}^- = \mathbf{A}_{k-1}\mathbf{P}_{\mathbf{x}_{k-1}}\mathbf{A}_{k-1}^T + \mathbf{R}^{\mathbf{v}}. \tag{5.27}$$

The measurement-update equations for the state filter are

$$\mathbf{K}_k^{\mathbf{x}} = \mathbf{P}_{\mathbf{x}_k}^- \mathbf{C}^T(\mathbf{C}\mathbf{P}_{\mathbf{x}_k}^-\mathbf{C}^T + \mathbf{R}^{\mathbf{n}})^{-1}, \tag{5.28}$$

$$\hat{\mathbf{x}}_k = \hat{\mathbf{x}}_k^- + \mathbf{K}_k^{\mathbf{x}}(y_k - \mathbf{C}\hat{\mathbf{x}}_k^-), \tag{5.29}$$

$$\mathbf{P}_{\mathbf{x}_k} = (\mathbf{I} - \mathbf{K}_k^{\mathbf{x}}\mathbf{C})\mathbf{P}_{\mathbf{x}_k}^-, \tag{5.30}$$

and those for the weight filter are

$$\mathbf{K}_k^{\mathbf{w}} = \mathbf{P}_{\mathbf{w}_k}^-(\mathbf{C}_k^{\mathbf{w}})^T[\mathbf{C}_k^{\mathbf{w}}\mathbf{P}_{\mathbf{w}_k}^-(\mathbf{C}_k^{\mathbf{w}})^T + \mathbf{R}^{\mathbf{e}}]^{-1}, \tag{5.31}$$

$$\hat{\mathbf{w}}_k = \hat{\mathbf{w}}_k^- + \mathbf{K}_k^{\mathbf{w}} \cdot \mathfrak{e}$$

where

$$\mathbf{A}_{k-1} \triangleq \left.\frac{\partial \mathbf{F}(\mathbf{x}, \hat{\mathbf{w}}_k^-)}{\partial \mathbf{x}}\right|_{\hat{\mathbf{x}}_{k-1}}, \qquad \mathfrak{e}_k = (y_k - \mathbf{C}\hat{\mathbf{x}}_k^-), \qquad \mathbf{C}_k^{\mathbf{w}} \triangleq -\frac{\partial \mathfrak{e}_k}{\partial \mathbf{w}} = \mathbf{C}\left.\frac{\partial \hat{\mathbf{x}}_k^-}{\partial \mathbf{w}}\right|_{\mathbf{w}=\hat{\mathbf{w}}_k^-}. \tag{5.34}$$

(RTRL) [35]. Taking the derivative of the signal filter equations results in the following system of recursive equations:

$$\frac{\partial \hat{\mathbf{x}}_{k+1}^-}{\partial \hat{\mathbf{w}}} = \frac{\partial \mathbf{F}(\hat{\mathbf{x}}, \hat{\mathbf{w}})}{\partial \hat{\mathbf{x}}_k}\frac{\partial \hat{\mathbf{x}}_k}{\partial \hat{\mathbf{w}}} + \frac{\partial \mathbf{F}(\hat{\mathbf{x}}, \hat{\mathbf{w}})}{\partial \hat{\mathbf{w}}_k}, \tag{5.35}$$

$$\frac{\partial \hat{\mathbf{x}}_k}{\partial \hat{\mathbf{w}}} = (\mathbf{I} - \mathbf{K}_k^{\mathbf{x}}\mathbf{C})\frac{\partial \hat{\mathbf{x}}_k^-}{\partial \hat{\mathbf{w}}} + \frac{\partial \mathbf{K}_k^{\mathbf{x}}}{\partial \hat{\mathbf{w}}}(y_k - \mathbf{C}\hat{\mathbf{x}}_k^-), \tag{5.36}$$

where $\partial \mathbf{F}(\hat{\mathbf{x}}, \hat{\mathbf{w}})/\partial \hat{\mathbf{x}}_k$ and $\partial \mathbf{F}(\hat{\mathbf{x}}, \hat{\mathbf{w}})/\partial \hat{\mathbf{w}}_k$ are evaluated at $\hat{\mathbf{w}}_k$ and contain static linearizations of the nonlinear function.

The last term in Eq. (5.36) may be dropped if we assume that the Kalman gain $\mathbf{K}_k^{\mathbf{x}}$ is independent of $\mathbf{w}$. Although this greatly simplifies the algorithm, the exact computation of $\partial \mathbf{K}_k^{\mathbf{x}}/\partial \hat{\mathbf{w}}$ may be computed, as shown in Appendix A. Whether the computational expense in calculating the recursive derivatives (especially that of calculating $\partial \mathbf{K}_k^{\mathbf{x}}/\partial \hat{\mathbf{w}}$) is worth the improvement in performance is clearly a design issue. Experimentally, the recursive derivatives appear to be more critical when the signal is highly nonlinear, or is corrupted by a high level of noise.

Example As an example application, consider the noisy time-series $\{x_k\}_1^N$ generated by a nonlinear autoregression:

$$\begin{aligned} x_k &= f(x_{k-1}, \ldots x_{k-M}, \mathbf{w}) + v_k, \\ y_k &= x_k + n_k \qquad \forall k \in \{1, \ldots, N\}. \end{aligned} \tag{5.37}$$

The observations of the series y_k contain measurement noise n_k in addition to the signal. The dual EKF requires reformulating this model into a state-space representation. One such representation is given by

$$\mathbf{x}_k = \mathbf{F}(\mathbf{x}_{k-1}, \mathbf{w}) + \mathbf{B}v_k, \tag{5.38}$$

$$\begin{bmatrix} x_k \\ x_{k-1} \\ \vdots \\ x_{k-M+1} \end{bmatrix} = \begin{bmatrix} f(x_{k-1}, \ldots, x_{k-M}, \mathbf{w}) \\ \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & \ddots & 0 & \vdots \\ 0 & 0 & 1 & 0 \end{bmatrix} \cdot \begin{bmatrix} x_{k-1} \\ \vdots \\ x_{k-M} \end{bmatrix} \end{bmatrix} + \begin{bmatrix} 1 \\ 0 \\ \vdots \\ 0 \end{bmatrix} v_k,$$

$$\begin{aligned} y_k &= \mathbf{C}\mathbf{x}_k + n_k, \\ &= [1 \quad 0 \quad \ldots \quad 0]\mathbf{x}_k + n_k, \end{aligned} \tag{5.39}$$

where the state $\mathbf{x}_k$ is chosen to be lagged values of the time series, and the state transition function $\mathbf{F}(\cdot)$ has its first element given by $f(\cdot)$, with the remaining elements corresponding to shifted values of the previous state.

The results of a controlled time-series experiment are shown in Figure 5.3. The clean signal, shown by the thin curve in Figure 5.3*a*, is generated by a neural network (10-5-1) with chaotic dynamics, driven by white Gaussian-process noise ($\sigma_v^2 = 0.36$). Colored noise generated by a linear autoregressive model is added at 3 dB signal-to-noise ratio (SNR) to produce the noisy data indicated by + symbols. Figure 5.3*b* shows the

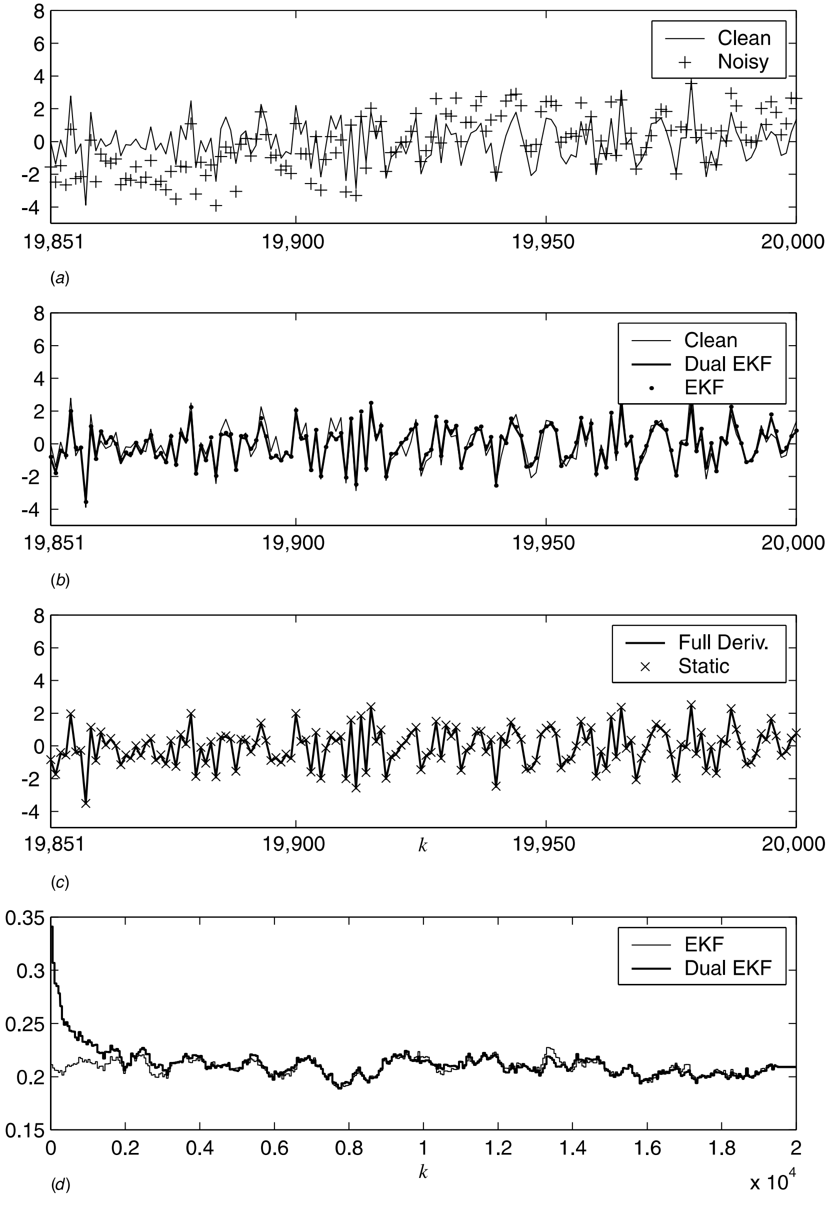


Figure 5.3 The dual EKF estimate (heavy curve) of a signal generated by a neural network (thin curve) and corrupted by adding colored noise at 3 dB (+). For clarity, the last 150 points of a 20,000-point series are shown. Only the noisy data are available: both the signal and weights are estimated by the dual EKF. (*a*) Clean neural network signal and noisy measurements. (*b*) Dual EKF estimates versus EKF estimates. (*c*) Estimates with full and static derivatives. (*d*) MSE profiles of EKF versus dual EKF.

time series estimated by the dual EKF. The algorithm estimates both the clean time series and the neural network weights. The algorithm is run sequentially over 20,000 points of data; for clarity, only the last 150 points are shown. For comparison, the estimates using an EKF with the *known* neural network model are also shown. The MSE for the dual EKF, computed over the final 1000 points of the series, is 0.2171, whereas the EKF produces an MSE of 0.2153, indicating that the dual algorithm has successfully learned both the model and the states estimates.[2]

Figure 5.3*c* shows the estimate when the static approximation to recursive derivatives is used. In this example, this static derivative actually provides a slight advantage, with an MSE of 0.2122. The difference, however, is not statistically significant. Finally, Figure 5.3*d* assesses the convergence behavior of the algorithm. The mean-squared error (MSE) is computed over 500 point segments of the time series at 50 point intervals to produce the *MSE profile* (dashed line). For comparison, the solid line is the MSE profile of the EKF signal estimation algorithm, which uses the true neural network model. The dual EKF appears to converge to the optimal solution after only about 2000 points.

5.3 A PROBABILISTIC PERSPECTIVE

In this section, we present a unified framework for dual estimation. We start by developing a probabilistic perspective, which leads to a number of possible cost functions that can be used in the estimation process. Various approaches in the literature, which may differ in their actual optimization procedure, can then be related based on the underlying cost function. We then show how a Kalman-based optimization procedure can be used to provide a common algorithmic framework for minimizing each of the cost functions.

MAP Estimation Dual estimation can be cast as a *maximum a posteriori* (MAP) solution. The statistical information contained in the sequence of data $\{y_k\}_1^N$ about the signal and parameters is embodied by the joint conditional probability density of the sequence of states $\{\mathbf{x}_k\}_1^N$ and weights

[2] A surprising result is that the dual EKF sometimes actually outperforms the EKF, even though the EKF appears to have an unfair advantage of knowing the true model. Our explanation is that the EKF, even with the known model, is still an *approximate* estimation algorithm. While the dual EKF also learns an approximate model, this model can actually be better matched to the state estimation approximation.

$\mathbf{w}$, given the noisy data $\{y_k\}_1^N$. For notational convenience, define the column vectors $\mathbf{x}_1^N$ and $\mathbf{y}_1^N$, with elements from $\{\mathbf{x}_k\}_1^N$ and $\{y_k\}_1^N$, respectively. The joint conditional density function is written as

$$\rho_{\mathbf{x}_1^N \mathbf{w} | \mathbf{y}_1^N}(\mathbf{X} = \mathbf{x}_1^N,\ \mathbf{W} = \mathbf{w} | \mathbf{Y} = \mathbf{y}_1^N), \tag{5.40}$$

where $\mathbf{X}$, $\mathbf{Y}$, and $\mathbf{W}$ are the vectors of random variables associated with $\mathbf{x}_1^N$, $\mathbf{y}_1^N$, and $\mathbf{w}$, respectively. This joint density is abbreviated as $\rho_{\mathbf{x}_1^N \mathbf{w} | \mathbf{y}_1^N}$. The MAP estimation approach consists of determining instances of the states and weights that maximize this conditional density. For Gaussian distributions, the MAP estimate also corresponds to the minimum mean-squared error (MMSE) estimator. More generally, as long as the density is unimodal and symmetric around the mean, the MAP estimate provides the *Bayes estimate* for a broad class of loss functions [36].

Taking MAP as the starting point allows dual estimation approaches to be divided into two basic classes. The first, referred to here as *joint estimation methods*, attempt to maximize $\rho_{\mathbf{x}_1^N \mathbf{w} | \mathbf{y}_1^N}$ directly. We can write this optimization problem explicitly as

$$(\hat{\mathbf{x}}_1^N, \hat{\mathbf{w}}) = \arg \max_{\mathbf{x}_1^N, \mathbf{w}} \rho_{\mathbf{x}_1^N \mathbf{w} | \mathbf{y}_1^N}. \tag{5.41}$$

The second class of methods, which will be referred to as *marginal estimation methods*, operate by expanding the joint density as

$$\rho_{\mathbf{x}_1^N \mathbf{w} | \mathbf{y}_1^N} = \rho_{\mathbf{x}_1^N | \mathbf{w} \mathbf{y}_1^N} \rho_{\mathbf{w} | \mathbf{y}_1^N} \tag{5.42}$$

and maximizing the two terms separately, that is,

$$\hat{\mathbf{x}}_1^N = \arg \max_{\mathbf{x}_1^N} \rho_{\mathbf{x}_1^N | \mathbf{w} \mathbf{y}_1^N}, \qquad \hat{\mathbf{w}} = \arg \max_{\mathbf{w}} \rho_{\mathbf{w} | \mathbf{y}_1^N}. \tag{5.43}$$

The cost functions associated with joint and marginal approaches will be discussed in the following sections.

5.3.1 Joint Estimation Methods

Using Bayes' rule, the joint conditional density can be expressed as

$$\rho_{\mathbf{x}_1^N \mathbf{w}|\mathbf{y}_1^N} = \frac{\rho_{\mathbf{y}_1^N|\mathbf{x}_1^N \mathbf{w}} \rho_{\mathbf{x}_1^N \mathbf{w}}}{\rho_{\mathbf{y}_1^N}} = \frac{\rho_{\mathbf{y}_1^N|\mathbf{x}_1^N \mathbf{w}} \rho_{\mathbf{x}_1^N|\mathbf{w}} \rho_{\mathbf{w}}}{\rho_{\mathbf{y}_1^N}}. \tag{5.44}$$

Although $\{y_k\}_1^N$ is *statistically* dependent on $\{\mathbf{x}_k\}_1^N$ and $\mathbf{w}$, the prior $\rho_{\mathbf{y}_1^N}$ is nonetheless *functionally* independent of $\{\mathbf{x}_k\}_1^N$ and $\mathbf{w}$. Therefore, $\rho_{\mathbf{x}_1^N \mathbf{w}|\mathbf{y}_1^N}$ can be maximized by maximizing the terms in the numerator alone. Furthermore, if no prior information is available on the weights, $\rho_{\mathbf{w}}$ can be dropped, leaving the maximization of

$$\rho_{\mathbf{y}_1^N \mathbf{x}_1^N|\mathbf{w}} = \rho_{\mathbf{y}_1^N|\mathbf{x}_1^N \mathbf{w}} \rho_{\mathbf{x}_1^N|\mathbf{w}} \tag{5.45}$$

with respect to $\{\mathbf{x}_k\}_1^N$ and $\mathbf{w}$.

To derive the corresponding cost function, we assume $\mathbf{v}_k$ and n_k are both zero-mean white Gaussian noise processes. It can then be shown (see [22]), that

$$\begin{aligned}\rho_{\mathbf{y}_1^N \mathbf{x}_1^N|\mathbf{w}} = {} & \frac{1}{\sqrt{(2\pi)^N (\sigma_n^2)^N}} \exp\left[-\sum_{k=1}^{N} \frac{(y_k - \mathbf{C}\mathbf{x}_k)^2}{2\sigma_n^2} \right] \\ & \times \frac{1}{\sqrt{(2\pi)^N |\mathbf{R}^{\mathrm{v}}|^N}} \exp\left[-\sum_{k=1}^{N} \frac{1}{2} (\mathbf{x}_k - \mathbf{x}_k^-)^T (\mathbf{R}^{\mathrm{v}})^{-1} (\mathbf{x}_k - \mathbf{x}_k^-) \right],\end{aligned} \tag{5.46}$$

where

$$\mathbf{x}_k^- \triangleq E[\mathbf{x}_k | \{\mathbf{x}_t\}_1^{k-1}, \mathbf{w}] = \mathbf{F}(\mathbf{x}_{k-1}, \mathbf{w}). \tag{5.47}$$

Here we have used the structure given in Eq. (5.37) to compute the prediction $\mathbf{x}_k^-$ using the model $\mathbf{F}(\cdot, \mathbf{w})$. Taking the logarithm, the corresponding cost function is given by

$$J = \sum_{k=1}^{N} \left[\log(2\pi\sigma_n^2) + \frac{(y_k - \mathbf{C}\mathbf{x}_k)^2}{\sigma_n^2} \right. \tag{5.48}$$

$$\left. + \log(2\pi|\mathbf{R}^{\mathrm{v}}|) + (\mathbf{x}_k - \mathbf{x}_k^-)^T (\mathbf{R}^{\mathrm{v}})^{-1} (\mathbf{x}_k - \mathbf{x}_k^-) \right]. \tag{5.49}$$

This cost function can be minimized with respect to any of the unknown quantities (including the variances, which we will consider in Section 5.4). For the time being, consider only the optimization of $\{\mathbf{x}_k\}_1^N$ and $\mathbf{w}$. Because the log terms in the above cost are independent of the signal and weights, they can be dropped, providing a more specialized cost function:

$$J^j(\mathbf{x}_1^N, \mathbf{w}) = \sum_{k=1}^{N} \left[\frac{(y_k - \mathbf{C}\mathbf{x}_k)^2}{\sigma_n^2} + (\mathbf{x}_k - \mathbf{x}_k^-)^T (\mathbf{R}^{\mathrm{v}})^{-1} (\mathbf{x}_k - \mathbf{x}_k^-) \right]. \tag{5.50}$$

The first term is a soft constraint keeping $\{\mathbf{x}_k\}_1^N$ close to the observations $\{y_k\}_1^N$. The smaller the measurement noise variance σ_n^2, the stronger this constraint will be. The second term keeps the state estimates and model estimates mutually consistent with the model structure. This constraint will be strong when the state is highly deterministic (i.e., $\mathbf{R}^{\mathrm{v}}$ is small).

$J^j(\mathbf{x}_1^N, \mathbf{w})$ should be minimized with respect to both $\{\mathbf{x}_k\}_1^N$ and $\mathbf{w}$ to find the estimates that maximize the joint density $\rho_{\mathbf{y}_1^N \mathbf{x}_1^N | \mathbf{w}}$. This is a difficult optimization problem because of the high degree of coupling between the unknown quantities $\{\mathbf{x}_k\}_1^N$ and $\mathbf{w}$. In general, we can classify approaches as being either *direct* or *decoupled*. In direct approaches, both the signal and the state are determined jointly as a multivariate optimization problem. Decoupled approaches optimize one variable at a time while the other variable is fixed, and then alternating. Direct algorithms include the joint EKF algorithm (see Section 5.1), which attempts to minimize the cost sequentially by combining the signal and weights into a single (joint) state vector. The decoupled approaches are elaborated below.

Decoupled Estimation To minimize $J^j(\mathbf{x}_1^N, \mathbf{w})$ with respect to the signal, the cost function is evaluated using the current estimate $\hat{\mathbf{w}}$ of the weights to generate the predictions. The simplest approach is to substitute the predictions $\hat{\mathbf{x}}_k^- \triangleq \mathbf{F}(\mathbf{x}_{k-1}, \hat{\mathbf{w}})$ directly into Eq. (5.50), obtaining

$$J^j(\mathbf{x}_1^N, \hat{\mathbf{w}}) = \sum_{k=1}^{N} \left[\frac{(y_k - \mathbf{C}\mathbf{x}_k)^2}{\sigma_n^2} + (\mathbf{x}_k - \hat{\mathbf{x}}_k^-)^T (\mathbf{R}^{\mathbf{v}})^{-1} (\mathbf{x}_k - \hat{\mathbf{x}}_k^-) \right]. \tag{5.51}$$

This cost function is then minimized with respect to $\{\mathbf{x}_k\}_1^N$. To minimize the joint cost function with respect to the weights, $J^j(\mathbf{x}_1^N, \mathbf{w})$ is evaluated using the current signal estimate $\{\hat{\mathbf{x}}_k\}_1^N$ and the associated (redefined) predictions $\hat{\mathbf{x}}_k^- \triangleq \mathbf{F}(\hat{\mathbf{x}}_{k-1}, \mathbf{w})$. Again, this results in a straightforward substitution in Eq. (5.50):

$$J^j(\hat{\mathbf{x}}_1^N, \mathbf{w}) = \sum_{k=1}^{N} \left[\frac{(y_k - \mathbf{C}\hat{\mathbf{x}}_k)^2}{\sigma_n^2} + (\hat{\mathbf{x}}_k - \hat{\mathbf{x}}_k^-)^T (\mathbf{R}^{\mathbf{v}})^{-1} (\hat{\mathbf{x}}_k - \hat{\mathbf{x}}_k^-) \right]. \tag{5.52}$$

An alternative simplified cost function can be used if it is assumed that only $\hat{\mathbf{x}}_k^-$ is a function of the weights:

$$J_i^j(\hat{\mathbf{x}}_1^N, \mathbf{w}) = \sum_{k=1}^{N} (\hat{\mathbf{x}}_k - \hat{\mathbf{x}}_k^-)^T (\mathbf{R}^{\mathbf{v}})^{-1} (\hat{\mathbf{x}}_k - \hat{\mathbf{x}}_k^-). \tag{5.53}$$

This is essentially a type of prediction error cost, where the model is trained to predict the *estimated* state. Effectively, the method maximizes $\rho_{\mathbf{x}_1^N|\mathbf{w}}$, evaluated at $\mathbf{x}_1^N = \hat{\mathbf{x}}_1^N$. A potential problem with this approach is that it is not directly constrained by the actual data $\{y_k\}_1^N$. An inaccurate (yet self-consistent) pair of estimates $(\hat{\mathbf{x}}_1^N, \hat{\mathbf{w}})$ could conceivably be obtained as a solution. Nonetheless, this is essentially the approach used in [14] for robust prediction of time series containing outliers.

In the decoupled approach to joint estimation, by separately minimizing each cost with respect to its argument, the values are found that maximize (at least locally) the joint conditional density function. Algorithms that fall into this class include a sequential two-observation form of the dual EKF algorithm [21], and the errors-in-variables (EIV) method applied to batch-style minimization [18, 19]. An alternative approach, referred to as *error coupling*, makes the extra step of taking the errors in the estimates into account. However, this error-coupled approach (investigated in [22]) does not appear to perform reliably, and is not described further in this chapter.

5.3.2 Marginal Estimation Methods

Recall that in marginal estimation, the joint density function is expanded as

$$\rho_{\mathbf{x}_1^N \mathbf{w}|\mathbf{y}_1^N} = \rho_{\mathbf{x}_1^N|\mathbf{w}\mathbf{y}_1^N} \rho_{\mathbf{w}|\mathbf{y}_1^N}, \tag{5.54}$$

and $\hat{\mathbf{x}}_1^N$ is found by maximizing the first factor on the right-hand side, while $\hat{\mathbf{w}}$ is found by maximizing the second factor. Note that only the first factor ($\rho_{\mathbf{x}_1^N|\mathbf{w}\mathbf{y}_1^N}$) is dependent on the state. Hence, maximizing this factor for the state will yield the same solution as when maximizing the joint density (assuming the optimal weights have been found). However, because both factors also depend on $\mathbf{w}$, maximizing the second ($\rho_{\mathbf{w}|\mathbf{y}_1^N}$) alone with respect to $\mathbf{w}$ is *not* the same as maximizing the joint density $\rho_{\mathbf{x}_1^N \mathbf{w}|\mathbf{y}_1^N}$ with respect to $\mathbf{w}$. Nonetheless, the resulting estimates $\hat{\mathbf{w}}$ are consistent and unbiased, if conditions of sufficient excitation are met [37].

The marginal estimation approach is exemplified by the maximum-likelihood approaches [8, 9] and EM approaches [11, 12]. Motivation for these methods usually comes from considering only the marginal density $\rho_{\mathbf{w}|\mathbf{y}_1^N}$ to be the relevant quantity to maximize, rather than the joint density $\rho_{\mathbf{x}_1^N \mathbf{w}|\mathbf{y}_1^N}$. However, in order to maximize the marginal density, it is necessary to generate signal estimates that are invariably produced by maximizing the first term $\rho_{\mathbf{x}_1^N|\mathbf{w}\mathbf{y}_1^N}$.

Maximum-Likelihood Cost To derive a cost function for weight estimation, we further expand the marginal density as

$$\rho_{\mathbf{w}|\mathbf{y}_1^N} = \frac{\rho_{\mathbf{y}_1^N|\mathbf{w}} \rho_{\mathbf{w}}}{\rho_{\mathbf{y}_1^N}}. \tag{5.55}$$

If there is no prior information on $\mathbf{w}$, maximizing this posterior density is equivalent to maximizing the likelihood function $\rho_{\mathbf{y}_1^N|\mathbf{w}}$. Assuming Gaussian statistics, the chain rule for conditional probabilities can be used to express this likelihood function as:

$$\rho_{\mathbf{y}_1^N|\mathbf{w}} = \prod_{k=1}^{N} \frac{1}{\sqrt{2\pi\sigma_{\varepsilon_k}^2}} \exp\left[-\frac{(y_k - \overline{y_{k|k-1}})^2}{2\sigma_{\varepsilon_k}^2} \right], \tag{5.56}$$

where

$$\overline{y_{k|k-1}} \triangleq E[y_k | \{y_t\}_1^{k-1}, \mathbf{w}] \tag{5.57}$$

is the conditional mean (and optimal prediction), and $\sigma^2_{\varepsilon_k}$ is the prediction error variance. Taking the logarithm yields the following *maximum-likelihood* cost function:

$$J^{ml}(\mathbf{w}) = \sum_{k=1}^{N} \left[\log(2\pi\sigma^2_{\varepsilon_k}) + \frac{(y_k - \overline{y_{k|k-1}})^2}{\sigma^2_{\varepsilon_k}} \right]. \tag{5.58}$$

Note that the log-likelihood function takes the same form whether the measurement noise is colored or white. In evaluating this cost function, the term $\overline{y_{k|k-1}} = \mathbf{C}\hat{\mathbf{x}}_k^-$ must be computed. Thus, the signal estimate must be determined as a step to weight estimation. For linear models, this can be done exactly using an ordinary Kalman filter. For nonlinear models, however, the expectation is approximated by an extended Kalman filter, which equivalently attempts to minimize the joint cost $J^j(\mathbf{x}_1^k, \hat{\mathbf{w}})$ defined in Section 5.3.1 by Eq. (5.51).

An iterative maximum-likelihood approach for linear models is described in [7] and [8]; this chapter presents a sequential maximum-likelihood approach for nonlinear models, developed in [21].

Prediction Error Cost Often the variance $\sigma^2_{\varepsilon_k}$ in the maximum-likelihood cost is assumed (incorrectly) to be independent of the weights $\mathbf{w}$ and the time index k. Under this assumption, the log likelihood can be maximized by minimizing the squared prediction error cost function:

$$J^{pe}(\mathbf{w}) = \sum_{k=1}^{N} (y_k - \overline{y_{k|k-1}})^2. \tag{5.59}$$

The basic dual EKF algorithm described in the previous section minimizes this simplified cost function with respect to the weights $\mathbf{w}$, and is an example of a recursive prediction error algorithm [6, 19]. While questionable from a theoretical perspective, these algorithms have been shown in the literature to be quite useful. In addition, they benefit from reduced computational cost, because the derivative of the variance $\sigma^2_{\varepsilon_k}$ with respect to $\mathbf{w}$ is not computed.

EM Algorithm Another approach to maximizing $\rho_{\mathbf{w}|\mathbf{y}_1^N}$ is offered by the expectation-maximization (EM) algorithm [10, 12, 38]. The EM algorithm can be derived by first expanding the log-likelihood as

$$\log \rho_{\mathbf{y}_1^N|\mathbf{w}} = \log \rho_{\mathbf{y}_1^N \mathbf{x}_1^N|\mathbf{w}} - \log \rho_{\mathbf{x}_1^N|\mathbf{w}\mathbf{y}_1^N}. \tag{5.60}$$

Taking the conditional expectation of both sides using the conditional density $\rho_{\mathbf{x}_1^N|\mathbf{w}\mathbf{y}_1^N}$ gives

$$\log \rho_{\mathbf{y}_1^N|\mathbf{w}} = E_{\mathbf{X}|\mathbf{Y}\mathbf{W}}[\log \rho_{\mathbf{y}_1^N \mathbf{x}_1^N|\mathbf{w}}|\mathbf{y}_1^N, \hat{\mathbf{w}}] - E_{\mathbf{X}|\mathbf{Y}\mathbf{W}}[\log \rho_{\mathbf{x}_1^N|\mathbf{w}\mathbf{y}_1^N}|\mathbf{y}_1^N, \hat{\mathbf{w}}], \tag{5.61}$$

where the expectation over $\mathbf{X}$ of the left-hand side has no effect, because $\mathbf{X}$ does not appear in $\log \rho_{\mathbf{y}_1^N|\mathbf{w}}$. Note that the expectation is conditional on a previous estimate of the weights, $\hat{\mathbf{w}}$. The second term on the right is concave by Jensen's inequality [39],[3] so choosing $\mathbf{w}$ to maximize the first term on the right-hand side alone will always increase the log-likelihood on the left-hand side. Thus, the EM algorithm repeatedly maximizes $E_{\mathbf{X}|\mathbf{Y}\mathbf{W}}[\log \rho_{\mathbf{y}_1^N \mathbf{x}_1^N|\mathbf{w}}|\mathbf{y}_1^N, \hat{\mathbf{w}}]$ with respect to $\mathbf{w}$, each time setting $\hat{\mathbf{w}}$ to the new maximizing value. The procedure results in maximizing the original marginal density $\rho_{\mathbf{y}_1^N|\mathbf{w}}$.

For the white-noise case, it can be shown (see [12, 22]) that the EM cost function is

$$\begin{aligned} J^{em} = E_{\mathbf{X}|\mathbf{Y}\mathbf{W}}\Bigg[\sum_{k=1}^{N}\Bigg\{&\log(2\pi\sigma_n^2) + \frac{(y_k - \mathbf{C}\mathbf{x}_k)^2}{\sigma_n^2} \\ &+ \log(2\pi|\mathbf{R}^{\mathrm{v}}|) + (\mathbf{x}_k - \mathbf{x}_k^-)^T(\mathbf{R}^{\mathrm{v}})^{-1}(\mathbf{x}_k - \mathbf{x}_k^-)\Bigg\}z\Bigg|\mathbf{y}_1^N, \hat{\mathbf{w}}\Bigg], \end{aligned} \tag{5.62}$$

where $\mathbf{x}_k^- \triangleq \mathbf{F}(\mathbf{x}_{k-1}, \mathbf{w})$, as before. The evaluation of this expectation is computable on a term-by-term basis (see [12] for the linear case). However, for the sake of simplicity, we present here the resulting

[3] Jensen's inequality states that $E[g(x)] \le g(E[x])$ for a concave function $g(\cdot)$.

expression for the special case of time-series estimation, represented in Eq. (5.37). As shown in [22], the expectation evaluates to

$$J^{em} = N \log(4\pi^2 \sigma_v^2 \sigma_n^2) + \sum_{k=1}^{N} \Bigg[\frac{(y_k - \hat{x}_{k|N})^2 + p_{k|N}}{\sigma_n^2} + \frac{(\hat{x}_{k|N} - \hat{x}_{k|N}^{-})^2 + p_{k|N} - 2p_{k|N}^{\dagger} + p_{k|N}^{-}}{\sigma_v^2} \Bigg], \tag{5.63}$$

where $\hat{x}_{k|N}$ and $p_{k|N}$ are defined as the conditional mean and variance of x_k given $\hat{\mathbf{w}}$ and *all* the data, $\{y_k\}_1^N$. The terms $\hat{x}_{k|N}^{-}$ and $p_{k|N}^{-}$ are the conditional mean and variance of $x_k^{-} = f(\mathbf{x}_{k-1}, \mathbf{w})$, given all the data. The additional term $p_{k|N}^{\dagger}$ represents the cross-variance of x_k and x_k^{-}, conditioned on all the data. Again we see that determining state estimates is a necessary step to determining the weights. In this case, the estimates $\hat{x}_{k|N}$ are found by minimizing the joint cost $J^j(\mathbf{x}_1^N, \hat{\mathbf{w}})$, which can be approximated using an extended Kalman smoother. A sequential version of EM can be implemented by replacing $\hat{x}_{k|N}$ with the usual causal estimates $\hat{x}_k$, found using the EKF.

Summary of Cost Functions The various cost functions given in this section are summarized in Table 5.4. No explicit signal estimation cost is given for the marginal estimation methods, because signal estimation is only an *implicit* step of the marginal approach, and uses the joint cost $J^j(\mathbf{x}_1^N, \hat{\mathbf{w}})$. These cost functions, combined with specific optimization methods, lead to the variety of algorithms that appear in the literature.

Table 5.4 Summary of dual estimation cost functions

	Symbol	Name of cost	Density	Eq.
Joint	$J^j(\mathbf{x}_1^N, \mathbf{w})$	Joint	$\rho_{\mathbf{x}_1^N \mathbf{w} \mid \mathbf{y}_1^N}$	(5.50)
	$J^j(\mathbf{x}_1^N, \hat{\mathbf{w}})$	Joint signal	$\rho_{\mathbf{x}_1^N \mathbf{w} \mid \mathbf{y}_1^N}$	(5.51)
	$J^j(\hat{\mathbf{x}}_1^N, \mathbf{w})$	Joint weight	$\rho_{\mathbf{x}_1^N \mathbf{w} \mid \mathbf{y}_1^N}$	(5.52)
	$J_i^j(\hat{\mathbf{x}}_1^N, \mathbf{w})$	Joint weight (independent)	$\rho_{\mathbf{x}_1^N \mid \mathbf{w}}$	(5.53)
Marginal	$J^{pe}(\mathbf{w})$	Prediction error	$\sim\rho_{\mathbf{w} \mid \mathbf{y}_1^N}$	(5.59)
	$J^{ml}(\mathbf{w})$	Maximum likelihood	$\rho_{\mathbf{w} \mid \mathbf{y}_1^N}$	(5.58)
	$J^{em}(\mathbf{w})$	EM	*n.a.*	(5.62)

In the next section, we shall show how each of these cost functions can be minimized using a general dual EKF-based approach.

5.3.3 Dual EKF Algorithms

In this section, we show how the dual EKF algorithm can be modified to minimize any of the cost functions discussed earlier. Recall that the basic dual EKF as presented in Section 5.2.3 minimized the prediction error cost of Eq. (5.59). As was shown in the last section, all approaches use the same joint cost function for the state-estimation component. Thus, the state EKF remains unchanged. Only the weight EKF must be modified. We shall show that this involves simply redefining the error term $\boldsymbol{\mathfrak{e}}_k$.

To develop the method, consider again the general state-space formulation for weight estimation (Eq. (5.11)):

$$\mathbf{w}_{k+1} = \mathbf{w}_k + \mathbf{r}_k, \tag{5.64}$$

$$\mathbf{d}_k = \mathbf{G}(\mathbf{x}_k, \mathbf{w}_k) + \mathbf{e}_k. \tag{5.65}$$

We may reformulate this state-space representation as

$$\mathbf{w}_k = \mathbf{w}_{k-1} + \mathbf{r}_k, \tag{5.66}$$

$$0 = -\boldsymbol{\mathfrak{e}}_k, +\mathbf{e}_k, \tag{5.67}$$

where $\boldsymbol{\mathfrak{e}}_k = \mathbf{d}_k - \mathbf{G}(\mathbf{x}_k, \mathbf{w}_k)$ and the target "observation" is fixed at zero. This *observed error* formulation yields the exact same set of Kalman equations as before, and hence minimizes the same prediction error cost, $J(\mathbf{w}) = \sum_{t=1}^{k} [\mathbf{d}_t - \mathbf{G}(\mathbf{x}_t, \mathbf{w})]^T (\mathbf{R}^e)^{-1} [\mathbf{d}_t - \mathbf{G}(\mathbf{x}_t, \mathbf{w})] = \sum_{t=1}^{k} J_t$. However, if we consider the modified-Newton algorithm interpretation, it can be shown [22] that the EKF weight filter is also equivalent to the recursion

$$\hat{\mathbf{w}}_k = \hat{\mathbf{w}}_k^- + \mathbf{P}_{\mathbf{w}_k} (\mathbf{C}_k^{\mathbf{w}})^T (\mathbf{R}^e)^{-1} (0 + \boldsymbol{\mathfrak{e}}_k), \tag{5.68}$$

where

$$\mathbf{C}_k^{\mathbf{w}} \triangleq \left. \frac{\partial(-\boldsymbol{\mathfrak{e}}_k)}{\partial \mathbf{w}} \right|_{\mathbf{w}=\mathbf{w}_k}^{T} \tag{5.69}$$

and

$$\mathbf{P}_{\mathbf{w}_k}^{-1} = (\lambda^{-1}\mathbf{P}_{\mathbf{w}_{k-1}})^{-1} + (\mathbf{C}_k^{\mathbf{w}})^T(\mathbf{R}^{\mathrm{e}})^{-1}\mathbf{C}_k^{\mathbf{w}}. \tag{5.72}$$

The weight update in Eq. (5.68) is of the form

$$\hat{\mathbf{w}}_k = \hat{\mathbf{w}}_k^- - \mathbf{S}_k \nabla_{\mathbf{w}} J(\hat{\mathbf{w}}_k^-)^T, \tag{5.73}$$

where $\nabla_{\mathbf{w}} J$ is the gradient of the cost J with respect to $\mathbf{w}$, and $\mathbf{S}_k$ is a symmetric matrix that approximates the inverse Hessian of the cost. Both the gradient and Hessian are evaluated at the previous value of the weight estimate. Thus, we see that by using the observed error formulation, it is possible to redefine the error term $\mathfrak{e}_k$, which in turn allows us to minimize an arbitrary cost function that can be expressed as a sum of instantaneous terms $J_k = \mathfrak{e}_k^T \mathfrak{e}_k$. This basic idea was presented by Puskorius and Feldkamp [40] for minimizing an entropic cost function; see also Chapter 2. Note that $J_k = \mathfrak{e}_k^T \mathfrak{e}_k$ does not uniquely specify $\mathfrak{e}_k$, which can be vector-valued. The error must be chosen such that the gradient and inverse Hessian approximations (Eqs. (5.70) and (5.72)) are consistent with the desired batch cost.

In the following sections, we give the exact specification of the error term (and corresponding gradient $\mathbf{C}_k^{\mathbf{w}}$) necessary to modify the dual EKF algorithm to minimize the different cost functions. The original set of dual EKF equations given in Table 5.3 remains the same, with only $\mathfrak{e}_k$ being redefined. Note that for each case, the full evaluation of $\mathbf{C}_k^{\mathbf{w}}$ requires taking recursive gradients. The procedure for this is analogous to that taken in Section 5.2.3. Furthermore, we restrict ourselves to the autoregressive time-series model with state-space representation given in Eqs. (5.38) and (5.39).

Joint Estimation Forms The corresponding weight cost function (see also Eq. (5.52)) and error terms are given in Table 5.5. Note that this represents a special *two-observation* form of the weight filter, where

$$\hat{x}_t^- = f(\hat{\mathbf{x}}_{t-1}, \mathbf{w}), \qquad e_k \stackrel{\Delta}{=} (y_k - \hat{x}_k), \qquad \tilde{\hat{x}}_k \stackrel{\Delta}{=} (\hat{x}_k - \hat{x})_k^-,$$

Note that this dual EKF algorithm represents a sequential form of the *decoupled* approach to joint optimization; that is, the two EKFs minimize the overall joint cost function by alternately optimizing one argument at a

Table 5.5 Joint cost function observed error terms for the dual EKF weight filter

$$J^j(\hat{\mathbf{x}}_1^k, \mathbf{w}) = \sum_{t=1}^{k} \left[\frac{(y_t - \hat{x}_t)^2}{\sigma_n^2} + \frac{(\hat{x}_t - \hat{x}_t^-)^2}{\sigma_v^2} \right], \tag{5.74}$$

$$\mathfrak{e}_k \triangleq \begin{bmatrix} \sigma_n^{-1} e_k \\ \sigma_v^{-1} \tilde{\hat{x}}_k \end{bmatrix}, \quad \text{with} \quad \mathbf{C}_k^{\mathbf{w}} = - \begin{bmatrix} \sigma_n^{-1} \nabla_{\mathbf{w}}^T e_k \\ \sigma_v^{-1} \nabla_{\mathbf{w}}^T \tilde{\hat{x}}_k] \end{bmatrix}.$$

time while the other argument is fixed. A *direct* approach found using the *joint* EKF is described later in Section 5.3.4.

Marginal Estimation Forms–Maximum-Likelihood Cost The corresponding weight cost function (see Eq. (5.58)) and error terms are given in Table 5.6, where

$$\varepsilon_k = y_k - \hat{x}_k^-, \qquad l_{\varepsilon,k} = \frac{\sigma_{\varepsilon,k}^2}{3\varepsilon_k^2 - 2\sigma_{\varepsilon,k}^2}.$$

Note that the prediction error variance is given by

$$\sigma_{\varepsilon_k}^2 = E[(y_k - \overline{y_{k|k-1}})^2 | \{y_t\}_1^{k-1}, \mathbf{w}] \tag{5.75a}$$

$$= E[(n_k + x_k - \hat{x}_k^-)^2 | \{y_t\}_1^{k-1}, \mathbf{w}] \tag{5.75b}$$

$$= \sigma_n^2 + \mathbf{C}\mathbf{P}_k^- \mathbf{C}^T, \tag{5.75c}$$

where $\mathbf{P}_k^-$ is computed by the Kalman signal filter (see [22] for a discussion of the selection and interpretation of $l_{\varepsilon,k}$).

Table 5.6 Maximum-likelihood cost function observed error terms for dual EKF weight filter

$$J^{ml}(\mathbf{w}) = \sum_{k=1}^{N} \left[\log(2\pi\sigma_{\varepsilon_k}^2) + \frac{(y_k - \hat{x}_k^-)^2}{\sigma_{\varepsilon_k}^2} \right],$$

$$\mathfrak{e}_k \triangleq \begin{bmatrix} (l_{\varepsilon,k})^{1/2} \\ \sigma_{\varepsilon_k}^{-1} \varepsilon_k \end{bmatrix}, \qquad \mathbf{C}_k^{\mathbf{w}} = \begin{bmatrix} -\frac{1}{2} \frac{(l_{\varepsilon,k})^{-1/2}}{\sigma_{\varepsilon_k}^2} \nabla_{\mathbf{w}}^T (\sigma_{\varepsilon_k}^2) \\ -\frac{1}{\sigma_{\varepsilon_k}} \nabla_{\mathbf{w}}^T \varepsilon_k + \frac{\varepsilon_k}{2(\sigma_{\varepsilon_k}^2)^{3/2}} \nabla_{\mathbf{w}}^T (\sigma_{\varepsilon_k}^2) \end{bmatrix}.$$

Table 5.7 Prediction error cost function observed error terms for the dual EKF weight filter

$$J^{pe}(\mathbf{w}) = \sum_{k=1}^{N} \varepsilon_k^2 = (y_k - \hat{x}_k^-)^2, \tag{5.76}$$

$$\mathfrak{e}_k \overset{\triangle}{=} \varepsilon_k = (y_k - \hat{x}_k^-), \qquad \mathbf{C}_k^{\mathbf{w}} = -\nabla_{\mathbf{w}} \varepsilon_k = \mathbf{C} \frac{\partial \hat{\mathbf{x}}_k^-}{\partial \mathbf{w}} \bigg|_{\mathbf{w}=\hat{\mathbf{w}}_k^-}.$$

Marginal Estimation Forms–Prediction Error Cost If $\sigma_{\varepsilon_k}^2$ is assumed to be independent of $\mathbf{w}$, then we are left with the formulas corresponding to the original basic dual EKF algorithm (for the time-series case); see Table 5.7.

Marginal Estimation Forms–EM Cost The dual EKF can be modified to implement a sequential EM algorithm. Note that the M-step, which relates to the weight filter, corresponds to a *generalized* M-step, in which the cost function is *decreased* (but not necessarily minimized) at each iteration. The formulation is given in Table 5.8, where $\tilde{\tilde{x}}_{k|k} = \hat{x}_k - \hat{x}_{k|k}^-$. Note that $J_k^{em}(\mathbf{w})$ was specified by dropping terms in Eq. (5.63) that are independent of the weights (see [22]). While $\hat{x}_k$ are found by the usual state EKF, the variance terms $p_{k|k}^\dagger$, and $p_{k|k}^-$, as well as $\hat{x}_{k|k}^-$ (a noncausal prediction), are not typically computed in the normal implementation of the state EKF. To compute these, the state vector is augmented by one additional lagged value of the signal:

$$\mathbf{x}_k^+ = \begin{bmatrix} \mathbf{x}_k \\ x_{k-M} \end{bmatrix} = \begin{bmatrix} x_k \\ \mathbf{x}_{k-1} \end{bmatrix}, \tag{5.78}$$

Table 5.8 EM cost function observed error terms for the dual EKF weight filter

$$J_k^{em}(\mathbf{w}) = \frac{(\hat{x}_k - \hat{x}_{k|k}^-)^2 - 2p_{k|k}^\dagger + p_{k|k}^-}{\sigma_v^2}, \tag{5.77}$$

$$\mathfrak{e}_k = \begin{bmatrix} \sigma_v^{-1} \tilde{\tilde{x}}_{k|k} \\ \sqrt{-2}\sigma_v^{-1}(p_{k|k}^\dagger)^{1/2} \\ \sigma_v^{-1}(p_{k|k}^-)^{1/2} \end{bmatrix}, \qquad \mathbf{C}_k^{\mathbf{w}} = \begin{bmatrix} -\frac{1}{\sigma_v} \nabla_{\mathbf{w}}^T \tilde{\tilde{x}}_{k|k} \\ -\dfrac{\sqrt{-2}(p_{k|k}^\dagger)^{-1/2}}{2\sigma_v} \nabla_{\mathbf{w}}^T p_{k|k}^\dagger \\ -\dfrac{(p_{k|k}^-)^{-1/2}}{2\sigma_v} \nabla_{\mathbf{w}}^T p_{k|k}^- \end{bmatrix}.$$

The state Kalman filter is then modified by adding a final zero element to the vectors **B** and **C** (see Eqs. (5.38) and (5.39)), and the linearized state transition matrix is redefined as

$$\mathbf{A}_k \triangleq \begin{bmatrix} \nabla_{\mathbf{x}}^T f \\ \mathbf{I} \quad 0 \end{bmatrix}.$$

Now the estimate $\hat{\mathbf{x}}_k^+$ will contain $\hat{\mathbf{x}}_{k-1|k}$ in its last M elements. As shown in [22], the variance terms are then approximated by

$$p_{k|k}^- = \mathbf{C}\mathbf{A}_{k-1|k}(\mathbf{P}_{k-1|k})\mathbf{A}_{k-1|k}^T\mathbf{C}^T, \qquad p_{k|k}^\dagger = \mathbf{C}(\mathbf{P}_k^\sharp)\mathbf{A}_{k-1|k}^T\mathbf{C}^T, \tag{5.79}$$

where the covariance $\mathbf{P}_{k-1|k}$ is provided as the lower right block of the augmented covariance $\mathbf{P}_k^+$, and $\mathbf{P}_k^\sharp$ is the upper right block of $\mathbf{P}_k^+$. The usual error covariance $\mathbf{P}_k$ is provided in the upper left block of $\mathbf{P}_k^+$. Furthermore, $\mathbf{A}_{k-1|k}$ is found by linearizing $f(\cdot)$ at $\hat{\mathbf{x}}_{k-1|k}$. The noncausal prediction $\hat{x}_{k|k}^- = E[f(\mathbf{x}_{k-1}, \mathbf{w})|\{y_t\}_1^k, \hat{\mathbf{w}}] \approx f(\hat{\mathbf{x}}_{k-1|k}, \mathbf{w})$.

Finally, the necessary gradient terms in the dual EKF algorithm can be evaluated as follows:

$$\nabla_{\mathbf{w}}^T \tilde{\tilde{x}}_{k|k} = -\nabla_{\mathbf{w}} \hat{x}_{k|k}^- = -\nabla_{\mathbf{w}} f(\hat{\mathbf{x}}_{k-1|k}, \mathbf{w}), \tag{5.80}$$

which is evaluated at $\hat{\mathbf{w}}_k$. The ith element of the gradient vector $\nabla_{\mathbf{w}} p_{k|k}^-$ is constructed from the expression

$$\frac{\partial p_{k|k}^-}{\partial w^{(i)}} = \mathbf{C}\left[\frac{\partial \mathbf{A}_{k-1|k}}{\partial w^{(i)}}(\mathbf{P}_{k-1|k})\mathbf{A}_{k-1|k}^T + \mathbf{A}_{k-1|k}(\mathbf{P}_{k-1|k})\frac{\partial \mathbf{A}_{k-1|k}^T}{\partial w^{(i)}}\right]\mathbf{C}^T, \tag{5.81}$$

and the elements of the gradient $\nabla_{\mathbf{w}} p_{k|k}^\dagger$ are given by

$$\frac{\partial p_{k|k}^\dagger}{\partial w^{(i)}} = \mathbf{C}\left[\frac{\partial \mathbf{A}_{k-1|k}}{\partial w^{(i)}}(\mathbf{P}_k^\sharp)^T\right]\mathbf{C}^T. \tag{5.82}$$

Note that all components in the cost function are recursive functions of $\hat{\mathbf{w}}$, but not of $\mathbf{w}$. Hence, no recurrent derivative computations are required for the EM algorithm. Furthermore, it can be shown that the actual error variances $p_{k|k}^-$ and $p_{k|k}^\dagger$ (not their gradients) cancel out in the formula for $\mathbf{C}_k^{\mathbf{w}}$, and thus should be replaced with large constant values to obtain a good Hessian approximation.

5.3.4 Joint EKF

In the previous section, the dual EKF was modified to minimize the joint cost function. This implementation represented a *decoupled* type of approach, in which *separate* state-space representations were used to estimate x_k and $\mathbf{w}$. An alternative *direct* approach is given by the joint EKF, which generates *simultaneous* MAP estimates of $\mathbf{x}_k$ and $\mathbf{w}$. This is accomplished by defining a new joint state-space representation with concatenated state:

$$\mathbf{z}_k = \begin{bmatrix} \mathbf{x}_k \\ \mathbf{w}_k \end{bmatrix}. \tag{5.83}$$

It is clear that maximizing the density $\rho_{\mathbf{z}_k|\mathbf{y}_1^k}$ is equivalent to maximizing $\rho_{\mathbf{x}_k\mathbf{w}|\mathbf{y}_1^k}$. Hence, the MAP-optimal estimate of $\mathbf{z}_k$ will contain the values of x_k and $\mathbf{w}_k$ that minimize the batch cost $J(\mathbf{x}_1^k, \mathbf{w})$. Running the single EKF with this state vector provides a sequential estimation algorithm. The joint EKF first appeared in the literature for the estimation of linear systems (in which there is a bilinear relation between the states and weights) [1, 2]. The general equations for nonlinear systems are given in Table 5.9.

Note that because the gradient of $f(\mathbf{z})$ with respect to $\mathbf{w}$ is taken with the other elements (namely, $\hat{\mathbf{x}}_k$) fixed, it will *not* involve recursive derivatives of $\hat{\mathbf{x}}_k$ with respect to $\mathbf{w}$. This fact is cited in [3] and [6] as a potential source of convergence problems for the joint EKF. Additional results and citations in [5] corroborate the difficulties of the approach, although the cause of divergence is linked there to the linearization of the coupled system, rather than the lack of recurrent derivatives (note that no divergence problems were encountered in preparing the experimental results in this chapter). Although the use of recurrent derivatives is suggested in [3] and [6], there is no theoretical justification for this. In summary, the joint EKF provides an alternative to the dual EKF for sequential minimization of the joint cost function. Note that the joint EKF cannot be readily adapted to minimize other cost functions discussed in this chapter.

5.4 DUAL EKF VARIANCE ESTIMATION

The implementation of the EKF requires the noise variance, σ_v^2 and σ_n^2 as parameters in the algorithm. Often these can be determined from physical knowledge of the problem (e.g., sensor accuracy or ambient noise

Table 5.9 The joint extended Kalman filter equations (time-series case)

Initialize with

$$\hat{\mathbf{z}}_0 = E[\mathbf{z}_0], \tag{5.84}$$

$$\mathbf{P}_0 = E[(\mathbf{z}_0 - \hat{\mathbf{z}}_0)(\mathbf{z}_0 - \hat{\mathbf{z}}_0)^T]. \tag{5.85}$$

For $k \in \{1, \ldots, \infty\}$, the time-update equations of the Kalman filter are

$$\hat{\mathbf{z}}_k^- = \bar{\mathbf{F}}(\hat{\mathbf{z}}_{k-1}), \tag{5.86}$$

$$\mathbf{P}_k^- = \bar{\mathbf{A}}_{k-1}\mathbf{P}_{k-1}\bar{\mathbf{A}}_{k-1}^T + \bar{\mathbf{V}}_k, \tag{5.87}$$

and the measurement update equations are

$$\bar{\mathbf{K}}_k = \mathbf{P}_k^-\bar{\mathbf{C}}^T(\bar{\mathbf{C}}\mathbf{P}_k^-\bar{\mathbf{C}}^T + \sigma_n^2)^{-1}, \tag{5.88}$$

$$\hat{\mathbf{z}}_k = \hat{\mathbf{z}}_k^- + \bar{\mathbf{K}}_k(y_k - \bar{\mathbf{C}}\hat{\mathbf{z}}_k^-), \tag{5.89}$$

$$\mathbf{P}_k = (\mathbf{I} - \bar{\mathbf{K}}_k\bar{\mathbf{C}})\mathbf{P}_k^-, \tag{5.90}$$

where

$$\bar{\mathbf{V}}_k \triangleq \mathrm{Cov}\begin{bmatrix} \mathbf{B}v_k \\ \mathbf{u}_k \end{bmatrix} = \begin{bmatrix} \mathbf{B}\sigma_v^2\mathbf{B}^T & 0 \\ 0 & \mathbf{R}^{\mathbf{r}} \end{bmatrix}, \tag{5.91}$$

$$\bar{\mathbf{F}}(\mathbf{z}_{k-1}) \triangleq \begin{bmatrix} \mathbf{F}(\mathbf{x}_{k-1}, \mathbf{w}_{k-1}) \\ \mathbf{I} \cdot \mathbf{w}_{k-1} \end{bmatrix}, \qquad \bar{\mathbf{C}} \triangleq [\mathbf{C} \quad 0 \quad \ldots \quad 0], \tag{5.92}$$

$$\bar{\mathbf{A}}_k \triangleq \left.\frac{\partial \bar{\mathbf{F}}(\mathbf{z})}{\partial \mathbf{z}}\right|_{\mathbf{z}=\hat{\mathbf{z}}_k} = \begin{bmatrix} \begin{bmatrix} \dfrac{\partial f(\hat{\mathbf{x}}_k, \mathbf{w})^T}{\partial \mathbf{x}} \\ \mathbf{I} \quad 0 \end{bmatrix} & \begin{bmatrix} \dfrac{\partial f(\hat{\mathbf{x}}_k, \mathbf{w})^T}{\partial \mathbf{w}} \\ 0 \quad 0 \end{bmatrix} \\ 0 & \mathbf{I} \end{bmatrix}. \tag{5.93}$$

measurements). However, if the variances are unknown, their values can be estimated within the dual EKF framework using cost functions similar to those derived in Section 5.3. A full treatment of the variance estimation filter is presented in [22]; here we focus on the maximum-likelihood cost function

$$J^{ml}(\sigma^2) = \sum_{k=1}^{N}\left[\log(2\pi\sigma_{\varepsilon_k}^2) + \frac{(y_k - \overline{y_{k|k-1}})^2}{\sigma_{\varepsilon_k}^2}\right]. \tag{5.94}$$

Note that this cost function is identical to the weight-estimation cost function, except that the argument is changed to emphasize the estimation of the unknown variances.

As with the weight filter, a modified-Newton algorithm can be found for each variance by using an observed error form of the Kalman filter and modeling the variances as

$$\sigma_{k+1}^2 = \sigma_k^2 + r_k, \tag{5.95}$$

$$0 = \mathfrak{e}_k + e_k, \tag{5.96}$$

which gives a one-dimensional state-space representation. However, a peculiar difficulty in the estimation of variances is that these quantities must be positive-valued. Because this constraint is not built explicitly into the cost functions, it is conceivable that negative values can be obtained.

One solution to this problem (inspired by [41]) is to estimate $l \triangleq \log(\sigma^2)$ instead. Negative values of l map to small positive values of σ^2, and $l = -\infty$ maps to $\sigma^2 = 0$. The logarithm is a monotonic function, so a one-to-one mapping exists between the optimal value of l and the optimal value of σ^2. An additional benefit of the logarithmic function is that it expands the dynamic range near $\sigma^2 = 0$, where the solution is more likely to reside; this can improve the numerical properties of the optimization.

Of course, this new formulation requires computing the gradients and Hessians of the cost J with respect to l. Fortunately, the change is fairly straightforward; these expressions are simple functions of the derivatives with respect to σ^2. The variance estimation filter is given in Table 5.10. Note that the dimension of the state space is 1 in the case of variance estimation, while the observation $\mathfrak{e}_k$ is generally multidimensional. For this reason, the covariance form of the KF is more efficient than the forms shown earlier for signal or weight estimation, which employ the matrix inversion lemma and use a Kalman gain term. This form of the variance filter is used in the experiments in the next section.

In Table 5.10, the mean $\overline{y_{k|k-1}}_k$ and variance $\sigma_{\varepsilon_k}^2$ are computed in the same way as before (see Eq. (5.75)), except that the unknown variance σ^2 is now an additional conditioning argument in the expectations. Hence, the derivatives are

$$\frac{\partial \overline{y_{k|k-1}}_k}{\partial \sigma^2} = -\frac{\partial \hat{x}_k^-}{\partial \sigma^2}, \tag{5.103}$$

$$\frac{\partial \sigma_{\varepsilon_k}^2}{\partial \sigma^2} = \frac{\partial \sigma_n^2}{\partial \sigma^2} + \frac{\partial \mathbf{P}_k-(1,1)}{\partial \sigma^2}, \tag{5.104}$$

where $\mathbf{P}_k^-(1,1)$ is the upper left element of the matrix, and $\partial \sigma_n^2 / \partial \sigma^2$ is either 1 or 0, depending on whether σ^2 is σ_n^2 or σ_v^2. The other derivatives

Table 5.10 Variance estimation filter of the dual EKF

Initialize with

$$\hat{\sigma}_0^2 = E[\sigma^2], \qquad p_{\sigma_0} = E[(\sigma^2 - \hat{\sigma}_0^2)(\sigma^2 - \hat{\sigma}_0^2)^T].$$

For $k \in \{1, \ldots, \infty\}$, the time-update equations for the variance filter are

$$\hat{\sigma}_k^{2-} = \hat{\sigma}_{k-1}^2, \tag{5.97}$$

$$p_{\sigma_k}^- = p_{\sigma_{k-1}} + \sigma_u^2, \qquad \sigma_u^2 = \left(\frac{1}{\lambda} - 1\right) p_{\sigma_{k-1}}, \tag{5.98}$$

and the measurement equations are

$$p_{\sigma_k} = \left[(p_{\sigma_k}^-)^{-1} + (\mathbf{C}_k^\sigma)^T \sigma_r^{-2} \mathbf{C}_k^\sigma (\hat{\sigma}_k^{2-})^2 + (\mathbf{C}_k^\sigma)^T \sigma_r^{-2} \boldsymbol{\mathfrak{e}}_k \hat{\sigma}_k^{2-} \right]^{-1}, \tag{5.99}$$

$$\hat{l}_k^- = \log(\hat{\sigma}_k^{2-}), \tag{5.100}$$

$$\hat{l}_k = \hat{l}_k^- + p_{\sigma_k} (\mathbf{C}_k^\sigma)^T \sigma_r^{-2} \boldsymbol{\mathfrak{e}}_k \hat{\sigma}_k^{2-}, \tag{5.101}$$

$$\hat{\sigma}_k^2 = e^{\hat{l}_k}. \tag{5.102}$$

The observed error term and gradient are

$$\boldsymbol{\mathfrak{e}}_k \triangleq \begin{bmatrix} (l_{\varepsilon,k})^{1/2} \\ \sigma_{\varepsilon_k}^{-1} \varepsilon_k \end{bmatrix}, \qquad \mathbf{C}_k^\sigma = \begin{bmatrix} -\dfrac{1}{2} \dfrac{(l_{\varepsilon,k})^{-1/2}}{\sigma_{\varepsilon_k}^2} \dfrac{\partial \sigma_{\varepsilon_k}^2}{\partial \sigma^2} \\ -\dfrac{1}{\sigma_{\varepsilon_k}} \dfrac{\partial \varepsilon_k}{\partial \sigma^2} + \dfrac{\varepsilon_k}{2(\sigma_{\varepsilon_k}^2)^{(3/2)}} \dfrac{\partial \sigma_{\varepsilon_k}^2}{\partial \sigma^2} \end{bmatrix},$$

where σ^2 represents either σ_n^2 or σ_v^2, and where $\dfrac{\partial \varepsilon_k}{\partial \sigma^2} = -\dfrac{\partial \overline{y_{k|k-1}}_k}{\partial \sigma^2}$.

are computed by taking the derivative of the Kalman filter equations with respect to either variance term (represented by σ^2). This results in the following system of recursive equations:

$$\frac{\partial \hat{\mathbf{x}}_{k+1}^-}{\partial \sigma^2} = \frac{\partial \mathbf{F}(\hat{\mathbf{x}}, \hat{\mathbf{w}})}{\partial \hat{\mathbf{x}}_k} \frac{\partial \hat{\mathbf{x}}_k}{\partial \sigma^2}, \tag{5.105}$$

$$\frac{\partial \hat{\mathbf{x}}_k}{\partial \sigma^2} = (\mathbf{I} - \mathbf{K}_k^{\mathbf{x}} \mathbf{C}) \frac{\partial \hat{\mathbf{x}}_k^-}{\partial \sigma^2} + \frac{\partial \mathbf{K}_k^{\mathbf{x}}}{\partial \sigma^2} (y_k - \mathbf{C} \hat{\mathbf{x}}_k^-), \tag{5.106}$$

where $\partial \mathbf{F}(\hat{\mathbf{x}}, \hat{\mathbf{w}}) / \partial \hat{\mathbf{x}}_k$ is evaluated at $\hat{\mathbf{w}}_k$, and represents a static linearization of the neural network. Note that $[\partial \mathbf{F}(\hat{\mathbf{x}}, \hat{\mathbf{w}}) / \partial \hat{\mathbf{w}}_k][\partial \hat{\mathbf{w}} / \partial \sigma^2]$ does not appear in Eq. (5.105), under the assumption that $\partial \hat{\mathbf{w}} / \partial \sigma^2 = 0$. The last term in

Eq. (5.106) may be dropped if we assume that the Kalman gain $\mathbf{K}^{\mathbf{x}}$ is independent of σ^2. However, for accurate computation of the recursive derivatives, $\partial \mathbf{K}_k^{\mathbf{x}} / \partial \sigma^2$ must be calculated; this is shown along with the derivative $\partial \mathbf{P}_k^-(1, 1)/\partial \sigma^2$ in Appendix A.

5.5 APPLICATIONS

In this section, we present the results of using the dual EKF methods on a number of different applications.

5.5.1 Noisy Time-Series Estimation and Prediction

For the first example, we return to the noisy time-series example of Section 5.2. Figure 5.4 compares the performance of the various dual EKF and joint EKF methods presented in this chapter. Box plots are used to show the mean, median, and quartile values based on 10 different runs (note that the higher mean for the maximum-likelihood cost is a consequence of a single outlier). The figure also compares performances for both known variances and estimated variances. The results in the example are fairly consistent with our findings on a number of controlled

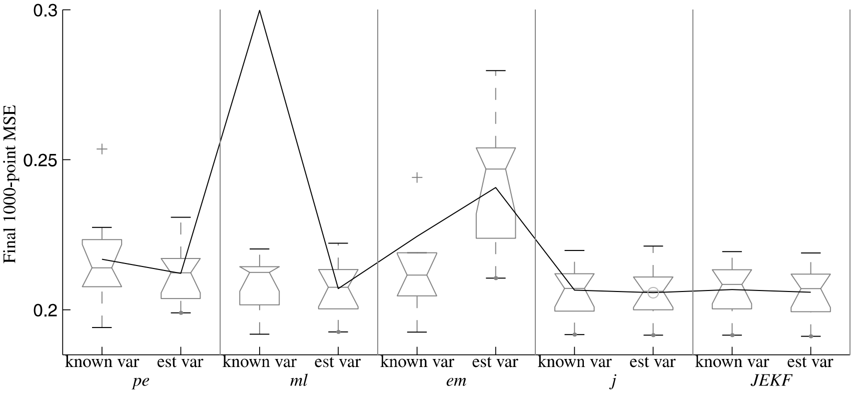


Figure 5.4 Performance of the dual EKF and joint EKF on a chaotic neural network time series described by Eqs. (5.37). MSE values are computed for the final 1000 points of the series. For each cost function, the horizontal lines indicate the median, upper and lower quartile values, and range of the overall NMSE. The mean values are linked by thin lines, and the algorithm with the lowest mean MSE is indicated by a superimposed circle.

experiments using different time series and parameter settings [22]. For both white and colored noises, the maximum-likelihood weight-estimation cost $J^{ml}(\mathbf{w})$ generally produces the best results, but often exhibits instability. This fact is analyzed in [22], and reduces the desirability of using this cost function. The joint cost function $J^{j}(\mathbf{w})$ has better stability properties and produces excellent results for colored measurement noise. For white noise, the prediction error cost performs nearly as well as $J^{ml}(\mathbf{w})$, but without stability problems. Hence, the dual EKF cost functions $J^{pe}(\mathbf{w})$ and $J^{j}(\mathbf{w})$ are generally the best choices for white and colored measurement noise, respectively. The joint EKF and dual EKF perform similarly in many cases, although the joint EKF is considerably less robust to inaccuracies in the assumed model structure and noise variances.

Chaotic Hénon Map As a second time-series example, we consider modeling the long-term behavior of the chaotic Hénon map:

$$a_{k+1} = 1 - 1.4a_k^2 + b_k, \tag{5.107}$$

$$b_{k+1} = 0.3b_k. \tag{5.108}$$

To obtain a one-dimensional time series for the following experiment, the signal is defined as $x_k = a_k$. The phase plot of x_{k+1} versus x_k (Fig. 5.5*a*) shows the chaotic attractor. A neural network (5-7-1) can easily be trained as a single-step predictor on this clean signal. The network is then iterated–feeding back the predictions of the network as future inputs–to produce the attractor shown in Figure 5.5*b*. However, if the signal is corrupted by white noise at 10 dB SNR (Fig. 5.5*c*), and a neural network with the same architecture is trained on these noisy data, the dynamics are not adequately captured. The iterated predictions exhibit limit-cycle behavior with far less complexity.

In contrast, using the dual EKF to train the neural network on the noisy data captures significantly more of the chaotic dynamics, as shown in Figure 5.5*d*. Here, $J^{pe}(\mathbf{w})$ is used for weight estimation, and the maximum-likelihood cost is used for estimating σ_v^2. The measurement-noise variance is assumed to be known. Parameter covariances are initialized at 0.1, and the initial signal covariance is $\mathbf{P}_{\mathbf{x}_0} = \mathbf{I}$. Forgetting factors are $\lambda_{\mathbf{w}} = 0.9999$ and $\lambda_{\sigma_v^2} = 0.9993$. Although the attractor is not reproduced with total fidelity, its general structure has been extracted from the noisy data.

This example also illustrates an interesting interpretation of dual EKF prediction training. During the training process, estimations from the output of the predictor are fed back as inputs, which are optimally

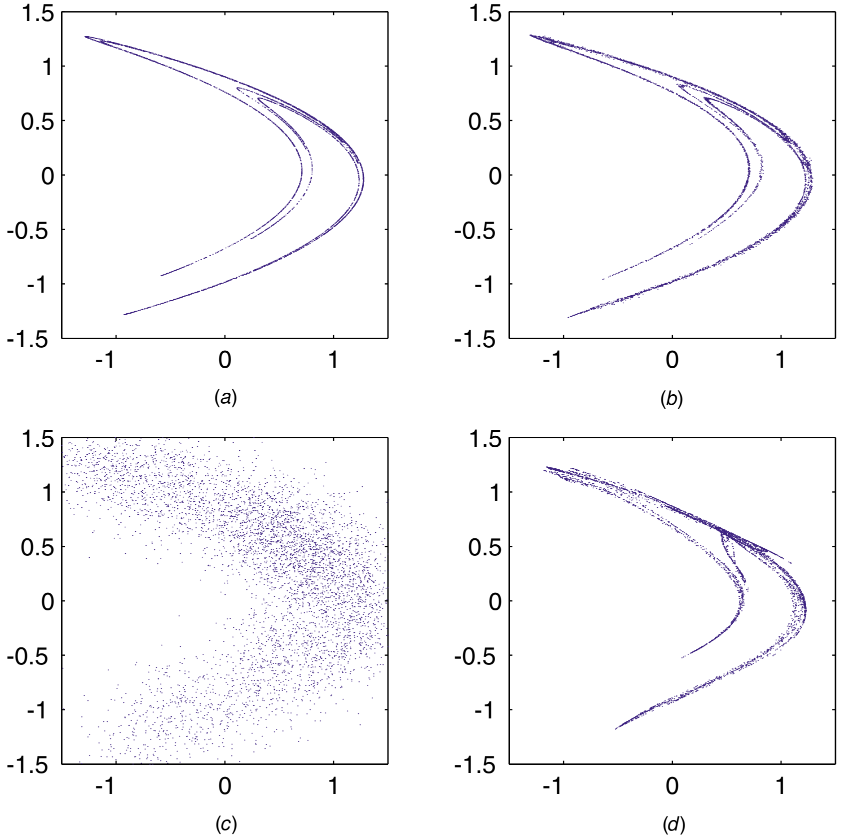


Figure 5.5 Phase plots of x_{k+1} versus x_k for the original Hénon series (*a*), the series generated by a neural network trained on x_k (*b*), the series generated by a neural network trained on y_k (*c*), and the series generated by a neural network trained on y_k, using the dual EKF (*d*).

weighted by the Kalman gain with the next noisy observation. The effective recurrent learning that takes place is analogous to *compromise* methods [42], which use the same principle of combining new observations with output predictions during training, in order to improve robustness of iterated performance.

5.5.2 Economic Forecasting–Index of Industrial Production

Economic and financial data are inherently noisy. As an example, we consider the *index of industrial production* (IP). As with most macroeconomic series, the IP is a composite index of many different economic indicators, each of which is generally measured by a survey of some kind. The monthly IP data are shown in Figure 5.6*a* for January 1950 to January 1990. To remove the trend, the differences between the $\log_2$ values for

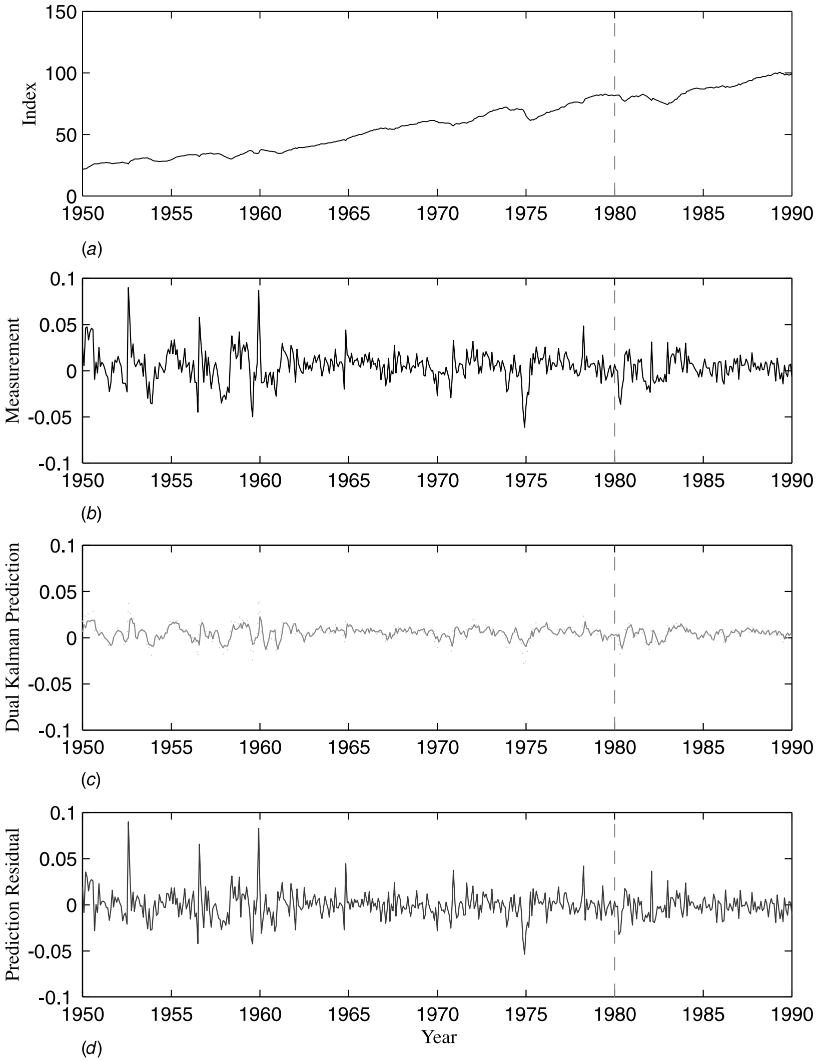


Figure 5.6 (*a*) Index of industrial production (IP) in the United States from January 1940 through March 2000. Data available from Federal Reserve (43). (*b*) Monthly rate of return of the IP in the United States, 1950–1990. (*c*) The dual KF prediction for a typical run (middle), along with the signal estimates (dotted line). (*d*) The prediction residual.

adjacent months are computed; this is called the IP monthly rate of return, and is shown in Figure 5.6*b*.

An important baseline approach is to predict the IP from its past values, using a standard linear autoregressive model. Results with an AR-14 model are reported by Moody et al. [44]. For comparison, both a linear AR-14 model and neural network (14 input, 4 hidden unit) model are tested using the dual EKF methods. Consistent with experiments reported in [44], data from January 1950 to December 1979 are used for a training set, and the remainder of the data is reserved for testing. The dual KF (or dual EKF) is iterated over the training set for several epochs, and the resultant model–consisting of $\hat{\mathbf{w}}$, $\hat{\sigma}_v^2$, and $\hat{\sigma}_n^2$–is used with a standard KF (or EKF) to produce causal predictions on the test set.

All experiments are repeated 10 times with different initial weights $\hat{\mathbf{w}}_0$ to produce the boxplots in Figure 5.7*a*. The advantage of dual estimation on linear models in comparison to the benchmark AR-14 model trained with least-squares (LS) is clear. For the nonlinear model, overtraining is a serious concern, because the algorithm is being run repeatedly over a very short training set (only 360 points). This effect is shown in Figure 5.7. Based on experience with other types of data (which may not have been optimal in this case) all runs were halted after only 5 training epochs. Nevertheless, the dual EKF with $J^j(\mathbf{w})$ cost produces significantly better results.

Although better results are reported on this problem in [44] using models with external inputs from other series, the dual EKF single-time-series results are quite competitive. Future work to incorporate additional inputs would, of course, be a straightforward extension within the dual EKF framework.

5.5.3 Speech Enhancement

Speech enhancement is concerned with the processing of noisy speech in order to improve the *quality* or *intelligibility* of the signal. Applications range from front-ends for automatic speech recognition systems, to telecommunications in aviation, military, teleconferencing, and cellular environments. While there exist a broad array of traditional enhancement techniques (e.g., spectral subtraction, signal-subspace embedding, time-domain iterative approaches, etc. [45]), such methods frequently result in audible distortion of the signal, and are somewhat unsatisfactory in real-world noisy environments. Different neural-network-based approaches are reviewed in [45].

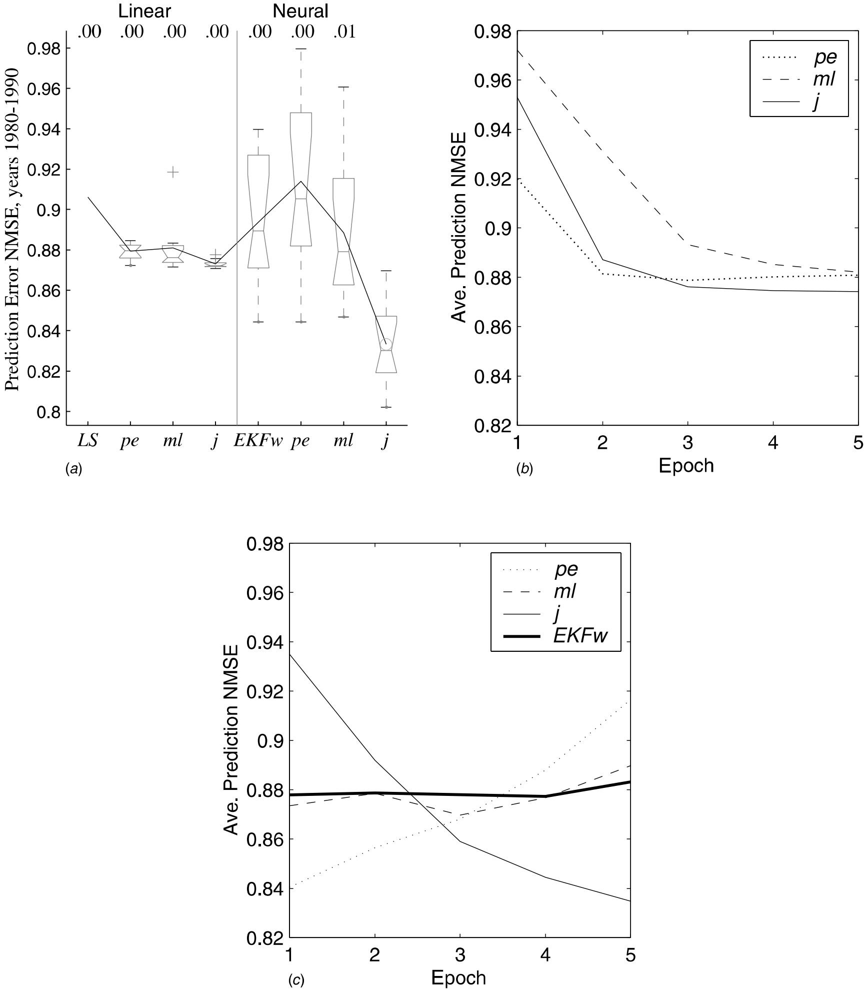


Figure 5.7 (*a*) Boxplots of the prediction NMSE on the test set (1980-1990). (*b*) and (*c*) show the average convergence behavior of linear and neural network model structures, respectively.

The dual estimation algorithms presented in this chapter have the advantage of generating estimates using only the noisy signal itself. To address its nonstationarity, the noisy speech is segmented into short overlapping windows. The dual estimation algorithms are then iterated over each window to generate the signal estimate. Parameters learned from one window are used to initialize the algorithm for the next window. Effectively, a *sequence* of time-series models is trained on the specific noisy speech signal of interest, resulting in a nonstationary model that can be used to remove noise from the given signal.

A number of controlled experiments using 8 kHz sampled speech have been performed in order to compare the different algorithms (joint EKF versus dual EKF with various costs). It was generally concluded that the best results were obtained with the dual EKF with $J^{ml}(\mathbf{w})$ cost, using a 10-4-1 neural network (versus a linear model), and window length set at 512 samples (overlap of 64 points). Preferred nominal settings were found to be: $\mathbf{P}_{\mathbf{x}_0} = \mathbf{I}$, $\mathbf{P}_{\mathbf{w}_0} = 0.01\mathbf{I}$, $p_{\sigma_0} = 0.1$, $\lambda_{\mathbf{w}} = 0.9997$, and $\lambda_{\sigma_v^2} = 0.9993$. The process-noise variance σ_v^2 is estimated with the dual EKF using $J^{ml}(\sigma_v^2)$, and is given a lower limit (e.g., 10^{-8}) to avoid potential divergence of the filters during silence periods. While, in practice, the additive-noise variance σ_n^2 could be estimated as well, we used the common procedure of estimating this from the start of the recording (512 points) where no speech is assumed present. In addition, linear AR-12 (or 10) filters were used to model colored additive noise. Using the settings found from these controlled experiments, several enhancement applications are reviewed below.

SpEAR Database The dual-EKF algorithm was applied to a portion of CSLU's Speech Enhancement Assessment Resource (SpEAR [47]). As opposed to artificially adding noise, the database is constructed by *acoustically* combining prerecorded speech (e.g., TIMIT) and noise (e.g., SPIB database [48]). Synchronous playback and recording in a room is used to provide exact time-aligned references to the clean speech such that objective measures can still be calculated. Table 5.11 presents sample results in terms of average *segmental* SNR.[4]

Car Phone Speech In this example, the dual EKF is used to process an actual recording of a woman talking on her cellular telephone while driving on the highway. The signal contains a significant level of road and engine noise, in addition to the distortion introduced by the telephone channel. The results appear in Figure 5.8, along with the noisy signal.

[4] Segmental SNR is considered to be a more perceptually relevant measure than standard SNR, and is computed as the average of the SNRs computed within 240-point windows, or *frames* of speech: SSNR $= (\#\text{ frames})^{-1} \sum_i \max(\text{SNR}_i, -10\,\text{dB})$. Here, SNR_i is the SNR of the ith frame (weighted by a Hanning window), which is thresholded from below at -10 dB. The thresholding reduces the contribution of portions of the series where no speech is present (i.e., where the SNR is strongly negative) [49], and is expected to improve the measure's perceptual relevance.

Table 5.11 Dual EKF enhancement results using a portion of the SpEAR database[a]

	Male voice (segmental SNR)			Female voice (segmental SNR)		
Noise	Before	After	Static	Before	After	Static
F-16	−2.27	2.65	1.69	0.16	4.51	3.46
Factory	−1.63	2.58	2.48	1.07	4.19	4.24
Volvo	1.60	5.60	6.42	4.10	6.78	8.10
Pink	−2.59	1.44	1.06	−0.23	4.39	3.54
White	−1.35	2.87	2.68	1.05	4.96	5.05
Bursting	1.60	5.05	4.24	7.82	9.36	9.61

[a] Different noise sources are used for the same male and female speaker. All results are in dB, and represent the segmental SNR averaged over the length of the waveform. Results labeled "static" were obtained using the static approximation to the derivatives. For reference, in this range of values, an improvement of 3 dB in segmental SNR relates to approximately an improvement of 5 dB in normal SNR.

Spectrograms of both the noisy speech and estimated speech are included to aid in the comparison. The noise reduction is most successful in nonspeech portions of the signal, but is also apparent in the visibility of formants of the estimated signal, which are obscured in the noisy signal. The perceptual quality of the result is quite good, with an absence of the "musical noise" artifacts often present in spectral subtraction results.

Seminar Recording The next example comes from an actual recording made of a lecture at the Oregon Graduate Institute. In this instance, the audio recording equipment was configured improperly, resulting in a very loud buzzing noise throughout the entire recording. The noise has a fundamental frequency of 60 Hz (indicating that improper grounding was the likely culprit), but many other harmonics and frequencies are present as well owing to some additional nonlinear clipping. As suggested by Figure 5.9, the SNR is extremely low, making for an unusually difficult audio enhancement problem.

Digit Recognition As the final example, we consider the application of speech enhancement for use as a front-end to automatic speech recognition (ASR) systems. The effectiveness of the dual EKF in this

[5] The authors wish to thank Edward Kaiser for his invaluable assistance in this experiment.

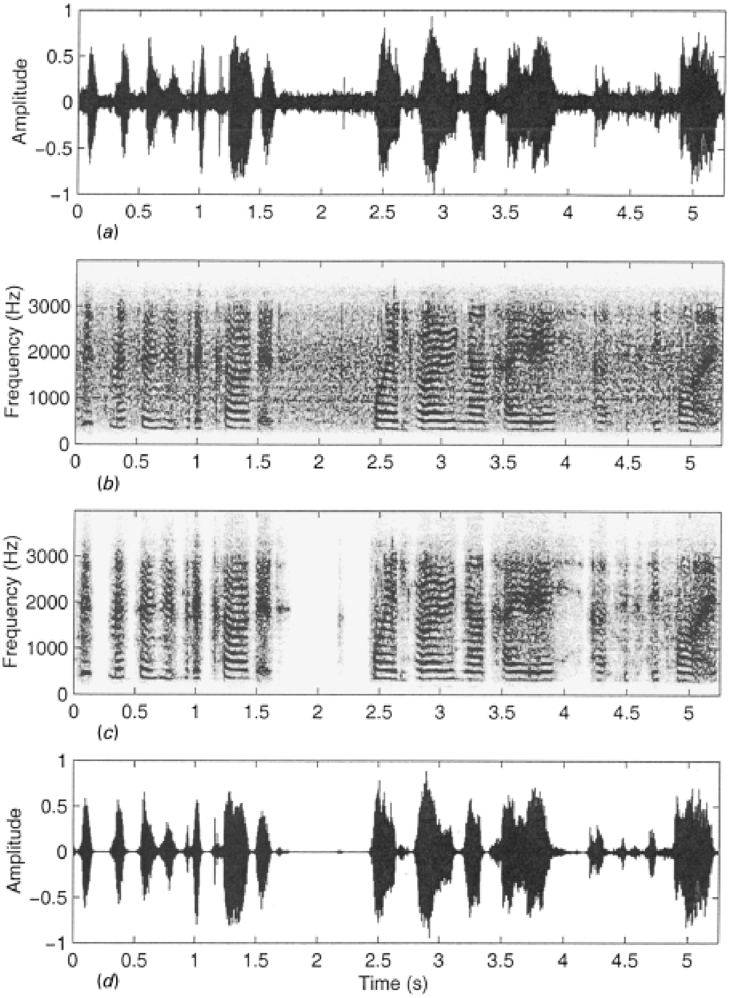


Figure 5.8 Enhancement of car phone speech. The noisy waveform appears in (*a*), with its spectrogram in (*b*). The spectrogram and waveform of the dual EKF result are shown in (*c*) and (*d*), respectively. To make the spectrograms easier to view, the spectral tilt is removed, and their histograms are equalized according to the range of intensities of the enhanced speech spectrogram.

application is demonstrated using a speech corpus and ASR system[5] developed at the Oregon Graduate Institute's Center for Spoken Language Understanding (CSLU). The speech corpus consists of zip-codes, addresses, and other digits read over the telephone by various people; the

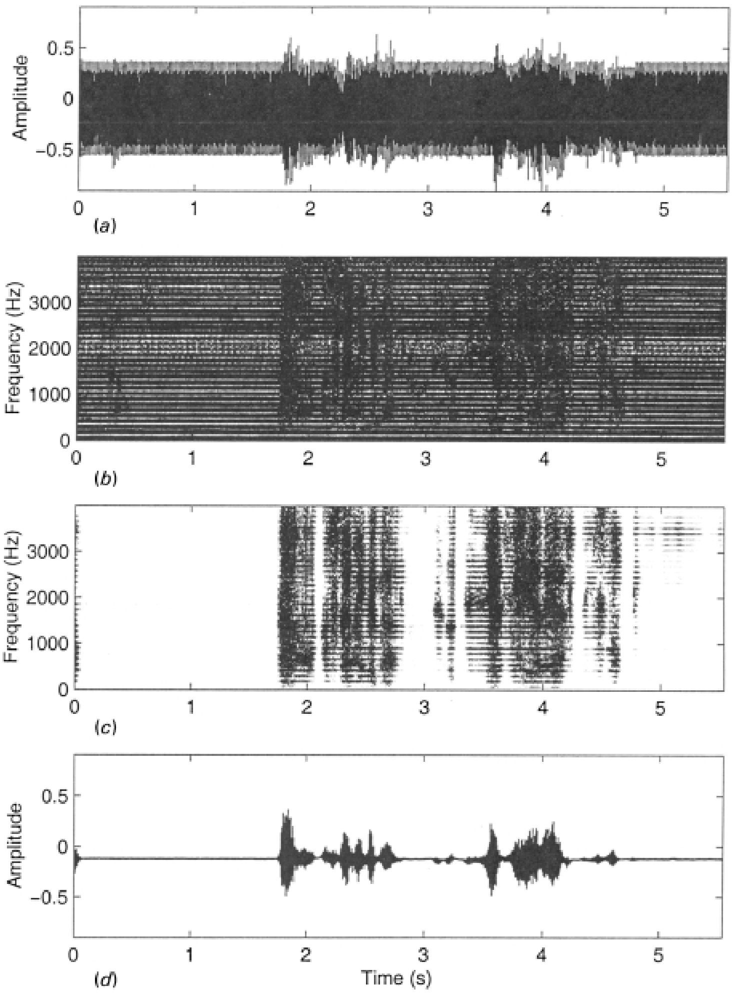


Figure 5.9 Enhancement of high-noise seminar recording. The noisy waveform appears in (*a*), with its spectrogram in (*b*). The spectrogram and waveform of the dual EKF result are shown in (*c*) and (*d*), respectively.

ASR system is a speaker-independent digit recognizer, trained exclusively to recognize numbers from zero to nine when read over the phone.

A subset of 599 sentences was used in this experiment. As seen in Table 5.12, the recognition rates on the clean telephone speech are quite good. However, adding white Gaussian noise to the speech at 6 dB significantly reduces the performance. In addition to the dual EKF, a standard spectral subtraction routine and an enhancement algorithm built into the speech codec TIA/EIA/IS-718 for digital cellular phones (published by the

Table 5.12 Automatic speech recognition rates for clean recordings of telephone speech (spoken digits), as compared with the same speech corrupted by white noise, and subsequently processed by spectral subtraction (SSUB), a cellular phone enhancement standard (IS-718), and the dual EKF

	Correct words	Correct sentences	
Clean	96.37%	85.81%	(514/599)
Noisy	59.21%	21.37%	(128/599)
SSUB	77.45%	38.06%	(228/599)
IS-718	67.32%	29.22%	(175/599)
Dual EKF	82.19%	52.92%	(317/599)

Telecommunications Industry Association) was used for comparison. As shown by Table 5.12, the dual EKF outperforms both the IS-718 and spectral subtraction recognition rates by a significant amount.

5.6 CONCLUSIONS

This chapter has detailed a unified approach to dual estimation based on a maximum *a posteriori* perspective. By maximizing the joint conditional density $\rho_{\mathbf{x}_1^N,\mathbf{w}|\mathbf{y}_1^N}$, the most probable values of the signal and parameters are sought, given the noisy observations. This probabilistic perspective elucidates the relationships between various dual estimation methods proposed in the literature, and allows their categorization in terms of methods that maximize the *joint* conditional density function directly, and those that maximize a related *marginal* conditional density function.

Cost functions associated with the joint and marginal densities have been derived under a Gaussian assumption. This approach offers a number of insights about previously developed methods. For example, the prediction error cost is viewed as an approximation to the maximum-likelihood cost; moreover, both are classified as marginal estimation cost functions. Thus, the recursive prediction error method of [5] and [6] is quite different from the joint EKF approach of [1] and [2], which minimizes a joint estimation cost.[6] Furthermore, the joint EKF and errors-in-variables algorithms are shown to offer two different ways of minimizing the same joint cost function; one is a sequential method and the other is iterative.

The dual EKF algorithm has been presented, which uses two extended Kalman filters run concurrently–one for state estimation and one for

[6] This fact is overlooked in [6], which emphasizes the similarity of these two algorithms.

weight estimation. By modification of the weight filter into an *observed error* form, it is possible to minimize each of the cost functions that are developed.[7] This provided a common algorithmic platform for the implementation of a broad variety of methods. In general, the dual EKF algorithm represents a *sequential* approach, which is applicable to both linear and nonlinear models, and which can be used in the presence of white or colored measurement noise. In addition, the algorithm has been extended to provide estimation of noise variance parameters within the same theoretical framework; this contribution is crucial in applications for which this information is not otherwise available.

Finally, a number of examples have been presented to illustrate the performance of the dual EKF methods. The ability of the dual EKF to capture the underlying dynamics of a noisy time series has been illustrated using the chaotic Hénon map. The application of the algorithm to the IP series demonstrates its potential in a real-world prediction context. On speech enhancement problems, the lack of musical noise in the enhanced speech underscores the advantages of a time-domain approach; the usefulness of the dual EKF as a front-end to a speech recognizer has also been demonstrated. In general, the state-space formulation of the algorithm makes it applicable to a much wider variety of contexts than has been explored here. The intent of this chapter was to show the utility of the dual EKF as a fundamental method for solving a range of problems in signal processing and modeling.

ACKNOWLEDGMENTS

This work was sponsored in part by the NSF under Grants ECS-0083106 and IRI-9712346.

APPENDIX A: RECURRENT DERIVATIVE OF THE KALMAN GAIN

(1) With Respect to the Weights

In the state-estimation filter, the derivative of the Kalman gain with respect to the weights $\mathbf{w}$ is computed as follows. Denoting the derivative of $\mathbf{K}_k^{\mathbf{x}}$

[7] Note that Kalman algorithms are *approximate* MAP optimization procedures for nonlinear systems. Hence, future work considers alternative optimization procedures (e.g., unscented Kalman filters [29]), which can still be cast within the same theoretically motivated dual estimation framework.

with respect to the ith element of $\hat{\mathbf{w}}$ by $\partial\mathbf{K}_k^{\mathbf{x}}/\partial\hat{w}(i)$ (the ith column of $\partial\mathbf{K}_k^{\mathbf{x}}/\partial\hat{\mathbf{w}}$) gives

$$\frac{\partial\mathbf{K}_k^{\mathbf{x}}}{\partial\hat{w}(i)} = \frac{\mathbf{I} - \mathbf{K}_k^{\mathbf{x}}\mathbf{C}}{\mathbf{C}\mathbf{P}_{\mathbf{x}_k}^{-}\mathbf{C}^T + \sigma_n^2}\frac{\partial\mathbf{P}_{\mathbf{x}_k}^{-}}{\partial\hat{w}(i)}\mathbf{C}^T, \tag{5.109}$$

where the derivatives of the error covariances are

$$\frac{\partial\mathbf{P}_{\mathbf{x}_k}^{-}}{\partial\hat{w}(i)} = \frac{\partial\mathbf{A}_{k-1}}{\partial\hat{w}(i)}\mathbf{P}_{\mathbf{x}_{k-1}}\mathbf{A}_{k-1}^T + \mathbf{A}_{k-1}\frac{\partial\mathbf{P}_{\mathbf{x}_{k-1}}}{\partial\hat{w}(i)}\mathbf{A}_{k-1}^T + \mathbf{A}_{k-1}\mathbf{P}_{\mathbf{x}_{k-1}}\frac{\partial\mathbf{A}_{k-1}^T}{\partial\hat{w}(i)}, \tag{5.110}$$

$$\frac{\partial\mathbf{P}_{\mathbf{x}_{k-1}}}{\partial\hat{w}(i)} = -\frac{\partial\mathbf{K}_{k-1}^{\mathbf{x}}}{\partial\hat{w}(i)}\mathbf{C}\mathbf{P}_{k-1}^{-} + (\mathbf{I} - \mathbf{K}_{k-1}^{\mathbf{x}}\mathbf{C})\frac{\partial\mathbf{P}_{\mathbf{x}_{k-1}}^{-}}{\partial\hat{w}(i)}. \tag{5.111}$$

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

$$\frac{\partial\mathbf{A}_{k-1}}{\partial\hat{w}(i)} = \frac{\partial^2\mathbf{F}}{\partial\hat{\mathbf{x}}_{k-1}\partial\hat{w}(i)} + \frac{\partial^2\mathbf{F}}{(\partial\hat{\mathbf{x}}_{k-1})^2}\frac{\partial\hat{\mathbf{x}}_{k-1}}{\partial\hat{w}(i)}, \tag{5.112}$$

where the first term is the static derivative of $\mathbf{A}_{k-1} = \partial\mathbf{F}/\partial\mathbf{x}_{k-1}$ with $\hat{\mathbf{x}}_{k-1}$ fixed, and the second term includes the recurrent derivative of $\hat{\mathbf{x}}_{k-1}$. The term $\partial^2\mathbf{F}/(\partial\hat{\mathbf{x}}_{k-1})^2$ actually represents a three-dimensional tensor (rather than a matrix), and care must be taken with this calculation. However, when $\mathbf{A}_{k-1}$ takes on a sparse structure, as with time-series applications, its derivative with respect to $\mathbf{x}$ contains mostly zeroes, and is in fact entirely zero for linear models.

(2) With Respect to the Variances

In the variance-estimation filter, the derivatives $\partial\mathbf{K}_k^{\mathbf{x}}/\partial\sigma^2$ may be calculated as follows:

$$\frac{\partial\mathbf{K}_k^{\mathbf{x}}}{\partial\sigma^2} = \frac{\mathbf{I} - \mathbf{K}_k^{\mathbf{x}}\mathbf{C}}{\mathbf{C}\mathbf{P}_k^{-}\mathbf{C}^T + \sigma_n^2}\frac{\partial\mathbf{P}_{\mathbf{x}_k}^{-}}{\partial\sigma^2}\mathbf{C}^T, \tag{5.113}$$

where

$$\frac{\partial \mathbf{P}_{\mathbf{x}_k}^-}{\partial \sigma^2} = \frac{\partial \mathbf{A}_{k-1}}{\partial \sigma^2} \mathbf{P}_{k-1} \mathbf{A}_{k-1}^T + \mathbf{A}_{k-1} \frac{\partial \mathbf{P}_{\mathbf{x}_{k-1}}}{\partial \sigma^2} \mathbf{A}_{k-1}^T + \mathbf{A}_{k-1} \mathbf{P}_{k-1} \frac{\partial \mathbf{A}_{k-1}}{\partial \sigma^2}, \quad (5.114)$$

$$\frac{\partial \mathbf{P}_{\mathbf{x}_{k-1}}}{\partial \sigma^2} = -\frac{\partial \mathbf{K}_{k-1}^{\mathbf{x}}}{\partial \sigma^2} \mathbf{C} \mathbf{P}_{k-1}^- + (\mathbf{I} - \mathbf{K}_{k-1}^{\mathbf{x}} \mathbf{C}) \frac{\partial \mathbf{P}_{\mathbf{x}_{k-1}}^-}{\partial \sigma^2}. \quad (5.115)$$

Because $\mathbf{A}_{k-1}$ depends on the linearization point, $\hat{\mathbf{x}}_{k-1}$, its derivative is

$$\frac{\partial \mathbf{A}_{k-1}}{\partial \sigma^2} = \frac{\partial \mathbf{A}_{k-1}}{\partial \hat{\mathbf{x}}_{k-1}} \frac{\partial \hat{\mathbf{x}}_{k-1}}{\partial \sigma^2}, \quad (5.116)$$

where again the derivative $\partial \hat{\mathbf{w}} / \partial \sigma^2$ is assumed to be zero.

APPENDIX B: DUAL EKF WITH COLORED MEASUREMENT NOISE

In this appendix, we give dual EKF equations for additive *colored* noise. Colored noise is modeled as a linear AR process:

$$n_k = \sum_{i=1}^{M_n} a_n^{(i)} n_{k-i} + v_{n,k}, \quad (5.128)$$

where the parameters $a_n^{(i)}$ are assumed to be known, and v_{nk} is a white Gaussian process with (possibly unknown) variance $\sigma_{v_n}^2$. The noise n_k can now be thought of as a second signal added to the first, but with the distinction that it has been generated by a known system. Note that the constraint $y_k = x_k + n_k$, requires that the estimates $\hat{x}_k$ and $\hat{n}_k$ must also sum to y_k. To enforce this constraint, both the signal and noise are incorporated into a combined state-space representation:

$$\boldsymbol{\mathfrak{e}}_k = \mathbf{F}_c(\boldsymbol{\mathfrak{e}}_{k-1}, \mathbf{w}, \mathbf{a}_n) + \mathbf{B}_c \mathbf{v}_{c,k}, \quad (5.129)$$

$$\begin{bmatrix} \mathbf{x}_k \\ \mathbf{n}_k \end{bmatrix} = \begin{bmatrix} \mathbf{F}(\mathbf{x}_{k-1}, \mathbf{w}) \\ \mathbf{A}_n \cdot \mathbf{n}_{k-1} \end{bmatrix} + \begin{bmatrix} \mathbf{B} & 0 \\ 0 & \mathbf{B}_n \end{bmatrix} \begin{bmatrix} v_k \\ v_{n,k} \end{bmatrix}, \quad (5.130)$$

$$y_k = \mathbf{C}_c \boldsymbol{\xi}_k,$$

$$y_k = [\mathbf{C} \quad \mathbf{C}_n] \begin{bmatrix} \mathbf{x}_k \\ \mathbf{n}_k \end{bmatrix},$$

where

$$\mathbf{A}_n \triangleq \begin{bmatrix} a_n^{(1)} & a_n^{(2)} & \dots & a_n^{(M_n)} \\ 1 & 0 & 0 & 0 \\ 0 & \ddots & 0 & \vdots \\ 0 & 0 & 1 & 0 \end{bmatrix}, \qquad \mathbf{C}_n = \mathbf{B}_n^T = [1 \quad 0 \quad \dots \quad 0].$$

The *effective* measurement noise is zero, and the process noise $\mathbf{v}_{c,k}$ is white, as required, with covariance

$$\mathbf{V}_c = \begin{bmatrix} \sigma_v^2 & 0 \\ 0 & \sigma_{v_n}^2 \end{bmatrix}.$$

Because n_k can be viewed as a second signal, it should be estimated on an equal footing with x_k. Consider, therefore, maximizing $\rho_{\mathbf{x}_1^N \mathbf{n}_1^N \mathbf{w} | \mathbf{y}_1^N}$ (where $\mathbf{n}_1^N$ is a vector comprising elements in $\{n_k\}_1^N$) instead of $\rho_{\mathbf{x}_1^N \mathbf{w} | \mathbf{y}_1^N}$. We can write this term as

$$\rho_{\mathbf{x}_1^N \mathbf{n}_1^N \mathbf{w} | \mathbf{y}_1^N} = \frac{\rho_{\mathbf{y}_1^N \mathbf{x}_1^N \mathbf{n}_1^N | \mathbf{w}} \rho_{\mathbf{w}}}{\rho_{\mathbf{y}_1^N}}, \tag{5.131}$$

and (in the absence of prior information about $\mathbf{w}$) maximize $\rho_{\mathbf{y}_1^N \mathbf{x}_1^N \mathbf{n}_1^N | \mathbf{w}}$ alone. As before, the cost functions that result from this approach can be categorized as joint or marginal costs, and their derivations are similar to those for the white noise case. The associated dual EKF algorithm for colored noise is given in Table 5.13. Minimization of different cost function is again achieved by simply redefining the error term. These modifications are presented without derivation below.

Joint Estimation Forms

The corresponding weight cost function and error terms for a *decoupled* approach is given in Table 5.14.

Table 5.13 The dual extended Kalman filter equations for colored measurement noise. The definitions of $\mathfrak{e}_k$ and $\mathbf{C}_k^{\mathbf{w}}$ will depend on the cost function used for weight estimation

Initialize with

$$\hat{\mathbf{w}}_0 = E[\mathbf{w}], \qquad \mathbf{P}_{\mathbf{w}_0} = E[(\mathbf{w} - \hat{\mathbf{w}}_0)(\mathbf{w} - \hat{\mathbf{w}}_0)^T],$$

$$\hat{\boldsymbol{\xi}}_0 = E[\boldsymbol{\xi}_0], \qquad \mathbf{P}_{\boldsymbol{\xi}_0} = E[(\boldsymbol{\xi}_0 - \hat{\boldsymbol{\xi}}_0)(\boldsymbol{\xi}_0 - \hat{\boldsymbol{\xi}}_0)^T].$$

For $k \in \{1, \ldots, \infty\}$, the time-update equations for the weight filter are

$$\hat{\mathbf{w}}_k^- = \hat{\mathbf{w}}_{k-1}, \tag{5.117}$$

$$\mathbf{P}_{\mathbf{w}_k}^- = \mathbf{P}_{\mathbf{w}_{k-1}} + \mathbf{R}_{k-1}^{\mathbf{r}}, \tag{5.118}$$

and those for the signal filter are

$$\hat{\boldsymbol{\xi}}_k^- = \mathbf{F}(\hat{\boldsymbol{\xi}}_{k-1}, \hat{\mathbf{w}}_k^-), \tag{5.119}$$

$$\mathbf{P}_{\boldsymbol{\xi}_k}^- = \mathbf{A}_{k-1}\mathbf{P}_{\boldsymbol{\xi}_{k-1}}\mathbf{A}_{k-1}^T + \mathbf{B}_c\mathbf{R}_c^{\mathbf{v}}\mathbf{B}_c^T. \tag{5.120}$$

The measurement-update equations for the signal filter are

$$\mathbf{K}_k^{\boldsymbol{\xi}} = \mathbf{P}_{\boldsymbol{\xi}_k}^- \mathbf{C}_c^T(\mathbf{C}_c\mathbf{P}_{\boldsymbol{\xi}_k}^-\mathbf{C}_c^T)^{-1}, \tag{5.121}$$

$$\hat{\boldsymbol{\xi}}_k = \hat{\boldsymbol{\xi}}_k^- + \mathbf{K}_k^{\boldsymbol{\xi}}(y_k - \mathbf{C}_c\hat{\boldsymbol{\xi}}_k^-), \tag{5.122}$$

$$\mathbf{P}_{\boldsymbol{\xi}_k} = (\mathbf{I} - \mathbf{K}_k^{\boldsymbol{\xi}}\mathbf{C})\mathbf{P}_{\boldsymbol{\xi}_k}^-, \tag{5.123}$$

and those for the weight filter are

$$\mathbf{K}_k^{\mathbf{w}} = \mathbf{P}_{\mathbf{w}_k}^-(\mathbf{C}_k^{\mathbf{w}})^T[\mathbf{C}_k^{\mathbf{w}}\mathbf{P}_{\mathbf{w}_k}^-(\mathbf{C}_k^{\mathbf{w}})^T + \mathbf{R}^{\mathbf{e}}]^{-1}, \tag{5.124}$$

$$\hat{\mathbf{w}}_k = \hat{\mathbf{w}}_k^- + \mathbf{K}_k^{\mathbf{w}} \cdot \mathfrak{e}_k, \tag{5.125}$$

$$\mathbf{P}_{\mathbf{w}_k} = (\mathbf{I} - \mathbf{K}_k^{\mathbf{w}}\mathbf{C}_k^{\mathbf{w}})\mathbf{P}_{\mathbf{w}_k}^-, \tag{5.126}$$

where

$$\mathbf{A}_{k-1} \overset{\triangle}{=} \left.\frac{\partial \mathbf{F}(\boldsymbol{\xi}, \hat{\mathbf{w}}_k^-, \mathbf{a}_n)}{\partial \boldsymbol{\xi}}\right|_{\hat{\boldsymbol{\xi}}_{k-1}}, \quad \mathfrak{e}_k = y_k - \mathbf{C}_c\hat{\boldsymbol{\xi}}_k^-, \quad \mathbf{C}_k^{\mathbf{w}} \overset{\triangle}{=} -\frac{\partial \mathfrak{e}_k}{\partial \mathbf{w}} = \mathbf{C}_c \left.\frac{\partial \hat{\boldsymbol{\xi}}_k^-}{\partial \mathbf{w}}\right|_{\mathbf{w}=\hat{\mathbf{w}}_k^-}. \tag{5.127}$$

Marginal Estimation–Maximum-Likelihood Cost

The corresponding weight cost function, and error terms are given in Table 5.15, where

$$\varepsilon_k \overset{\triangle}{=} y_k - (\hat{x}_k^- + \hat{n}_k^-), \qquad l_{\varepsilon,k} = \frac{\sigma_{\varepsilon,k}^2}{3\varepsilon_k^2 - 2\sigma_{\varepsilon,k}^2}$$

Table 5.14 Colored-noise joint cost function: observed error terms for the dual EKF weight filter

$$J(\hat{\mathbf{x}}_1^k, \hat{\mathbf{n}}_1^k, \mathbf{w}) + \sum_{t=1}^{k}\left[\frac{(\hat{x}_t - \hat{x}_t^-)^2}{\sigma_v^2} + \frac{(\hat{n}_t - \hat{n}_t^-)^2}{\sigma_{v_n}^2}\right], \quad \text{or} \quad J_k = \sum_{t=1}^{k}\left(\frac{\tilde{\hat{x}}_k^2}{\sigma_v^2} + \frac{\tilde{\hat{n}}_k^2}{\sigma_{v_n}^2}\right),$$

$$\mathbf{e}_k \overset{\triangle}{=} \begin{bmatrix} \sigma_v^{-1}\tilde{\hat{x}}_k \\ \sigma_{v_n}^{-1}\tilde{\hat{n}}_k \end{bmatrix}, \qquad \mathbf{C}_k^{\mathbf{w}} = -\begin{bmatrix} \sigma_v^{-1}\nabla_{\mathbf{w}}^T\tilde{\hat{x}}_k \\ \sigma_{v_n}^{-1}\nabla_{\mathbf{w}}^T\tilde{\hat{n}}_k \end{bmatrix}. \tag{5.132}$$

Table 5.15 Maximum-likelihood cost function: observed error terms for the dual EKF weight filter

$$J_c^{ml}(\mathbf{w}) = \sum_{k=1}^{N}\left[\log(2\pi\sigma_{\varepsilon_k}^2) + \frac{(y_k - \hat{x}_k^- - \hat{n}_k^-)^2}{\sigma_{\varepsilon_k}^2}\right],$$

$$\mathbf{e}_k \overset{\triangle}{=} \begin{bmatrix} (l_{\varepsilon,k}) \\ \sigma_{\varepsilon_k}^{-1}\varepsilon_k \end{bmatrix}^{1/2}, \qquad \mathbf{C}_k^{\mathbf{w}} = \begin{bmatrix} -\dfrac{1}{2}\dfrac{(l_{\varepsilon,k})^{-1/2}}{\sigma_{\varepsilon_k}^2}\nabla_{\mathbf{w}}^T(\sigma_{\varepsilon_k}^2) \\ -\dfrac{1}{\sigma_{\varepsilon_k}}\nabla_{\mathbf{w}}^T\varepsilon_k + \dfrac{\varepsilon_k}{2(\sigma_{\varepsilon_k}^2)^{3/2}}\nabla_{\mathbf{w}}^T(\sigma_{\varepsilon_k}^2) \end{bmatrix}.$$

Marginal Estimation Forms–Prediction Error Cost

If $\sigma_{\varepsilon_k}^2$ is assumed to be independent of $\mathbf{w}$, then we have the prediction error cost shown in Table 5.16. Note that an alternative prediction error form may be derived by including $\nabla_{\mathbf{w}}\hat{n}_k^-$ in the calculation of $\mathbf{C}_k^{\mathbf{w}}$. However, the performance appears superior if this term is neglected.

Table 5.16 Colored-noise prediction-error cost function: observed error terms for the dual EKF weight filter

$$J_c^{pe}(\mathbf{w}) = \sum_{k=1}^{N}\varepsilon_k^2 = \sum_{k=1}^{N}(y_k - \hat{x}_k^- - \hat{n}_k^-)^2, \tag{5.133}$$

$$\mathbf{e}_k \overset{\triangle}{=} y_k - \hat{n}_k - \hat{x}_k^-, \qquad \mathbf{C}_k^{\mathbf{w}} = -\nabla_{\mathbf{w}}\hat{x}_k^-.$$

Marginal Estimation–EM Cost

The cost and observed error terms for weight estimation with colored noise are identical to those for the white-noise case, shown in Table 5.8. In this case, the on-line statistics are found by augmenting the combined state vector with one additional lagged value for both the signal and noise. Specifically,

$$\boldsymbol{\xi}_k^+ = \begin{bmatrix} \mathbf{x}_k \\ x_{k-M} \\ \mathbf{n}_k \\ n_{k-M_n} \end{bmatrix} = \begin{bmatrix} x_k \\ \mathbf{x}_{k-1} \\ n_k \\ \mathbf{n}_{k-1} \end{bmatrix}, \tag{5.134}$$

so that the estimate $\hat{\boldsymbol{\xi}}_k^+$ produced by a Kalman filter will contain $\hat{\mathbf{x}}_{k-1|k}$ in elements $2, \ldots, 1+M$, and $\hat{\mathbf{n}}_{k-1|k}$ in its last M_n elements. Furthermore, the error variances $p_{k|k}^-$ and $p_{k|k}^\dagger$ can be obtained from the covariance $\mathbf{P}_{c,k}^+$ of $\boldsymbol{\xi}_k^+$ produced by the KF.

REFERENCES

[1] R.E. Kopp and R.J. Orford, “Linear regression applied to system identification for adaptive control systems,” *AIAA Journal*, **1**, 2300–2006 (1963).

[2] H. Cox, “On the estimation of state variables and parameters for noisy dynamic systems,” *IEEE Transactions on Automatic Control*, **9**, 5–12 (1964).

[3] L. Ljung, “Asymptotic behavior of the extended Kalman filter as a parameter estimator for linear systems,” *IEEE Transactions on Automatic Control*, **24**, 36–50 (1979).

[4] M. Niedźwiecki and K. Cisowski, “Adaptive scheme of elimination of broadband noise and impulsive disturbances from AR and ARMA signals,” *IEEE Transactions on Signal Processing*, **44**, 528–537 (1996).

[5] L.W. Nelson and E. Stear, “The simultaneous on-line estimation of parameters and states in linear systems,” *IEEE Transactions on Automatic Control*, Vol. AC-12, 438–442 (1967).

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

[7] H. Akaike, “Maximum likelihood identification of Gaussian autoregressive moving average models,” *Biometrika*, **60**, 255–265 (1973).

[8] N.K. Gupta and R.K. Mehra, “Computational aspects of maximum likelihood estimation and reduction in sensitivity function calculations,” *IEEE Transaction on Automatic Control*, **19**, 774–783 (1974).

[9] J.S. Lim and A.V. Oppenheim, “All-pole modeling of degraded speech,” *IEEE Transactions on Acoustics, Speech and Signal Processing*, **26**, 197–210 (1978).

[10] A. Dempster, N.M. Laird, and D.B. Rubin, “Maximum-likelihood from incomplete data via the EM algorithm,” *Journal of the Royal Statistical Society, Ser. B*, **39**, 1–38 (1977).

[11] B.R. Musicus and J.S. Lim, “Maximum likelihood parameter estimation of noisy data,” in *Proceedings of the International Conference on Acoustics, Speech, and Signal Processing*, *IEEE, April 1979*, pp. 224–227.

[12] R.H. Shumway and D.S. Stoffer, “An approach to time series smoothing and forecasting using the EM algorithm,” *Journal of Time Series Analysis*, **3**, 253–264 (1982).

[13] E. Weinstein, A.V. Oppenheim, M. Feder, and J.R. Buck, “Iterative and sequential algorithms for multisensor signal enhancement,” *IEEE Transactions on Signal Processing*, **42**, 846–859 (1994).

[14] J.T. Connor, R.D. Martin, and L.E. Atlas, “Recurrent neural networks and robust time series prediction,” *IEEE Transactions on Neural Networks*, **5**(2), 240–254 (1994).

[15] S.C. Stubberud and M. Owen, “Artificial neural network feedback loop with on-line training, in *Proceedings of International Symposium on Intelligent Control, IEEE, September* 1996, pp. 514–519.

[16] Z. Ghahramani and S.T. Roweis, “Learning nonlinear dynamical systems using an EM algorithm,” in *Advances in Neural Information Processing Systems 11: Proceedings of the 1998 Conference*. Cambridge, MA: MIT Press, 1999.

[17] T. Briegel and V. Tresp, “Fisher scoring and a mixture of modes approach for approximate inference and learning in nonlinear state space models,” in *Advances in Neural Information Processing Systems 11: Proceedings of the 1998 Conference*. Cambridge, MA: MIT Press, 1999.

[18] G. Seber and C. Wild, *Nonlinear Regression*. New York: Wiley, 1989, pp. 491–527.

[19] 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*. Cambridge, MA: MIT Press, 1997.

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

[21] A.T. Nelson and E.A. Wan, "A two-observation Kalman framework for maximum-likelihood modeling of noisy time series," in *Proceedings of International Joint Conference on Neural Networks, IEEE/INNS, May 1998.*

[22] A.T. Nelson, "Nonlinear Estimation and Modeling of Noisy Time-Series by Dual Kalman Filter Methods," PhD thesis, Oregon Graduate Institute of Science and Technology, 2000.

[23] R.E. Kalman, "A new approach to linear filtering and prediction problems," *Transactions of the ASME, Ser. D, Journal of Basic Engineering*, **82**, 35–45 (1960).

[24] J.F.G. de Freitas, M. Niranjan, A.H. Gee, and A. Doucet, "Sequential Monte Carlo methods for optimisation of neural network models," Technical Report TR-328, Cambridge University Engineering Department, November 1998.

[25] R. van der Merwe, N. de Freitas, A. Doucet, and E. Wan, "The unscented particle filter," Technical Report CUED/F-INFENG/TR 380, Cambridge University Engineering Department, August 2000.

[26] S.J. Julier, J.K. Uhlmann, and H. Durrant-Whyte, "A new approach for filtering nonlinear systems," in *Proceedings of the American Control Conference, 1995*, pp. 1628–1632.

[27] S.J. Julier and J.K. Uhlmann, "A general method for approximating nonlinear transformations of probability distributions," Technical Report, RRG, Department of Engineering Science, University of Oxford, November 1996. http://www.robots.ox.ac.uk/~siju/work/publications/letter_size/Unscented.zip.

[28] E.A. Wan and R. van der Merwe, "The unscented Kalman filter for nonlinear estimation," in *Proceedings of Symposium 2000 on Adaptive Systems for Signal Processing, Communication and Control (AS-SPCC), Lake Louise, Alberta, Canada, IEEE, October 2000.*

[29] E.A. Wan, R. van der Merwe, and A.T. Nelson, "Dual estimation and the unscented transformation," in *Advances in Neural Information Processing Systems 12: Proceedings of the 1999 Conference*. Cambridge, MA: MIT Press, 2000.

[30] S. Singhal and L. Wu, "Training multilayer perceptrons with the extended Kalman filter," in *Advances in Neural Information Processing Systems 1.* San Mateo, CA: Morgan Kauffman, 1989, pp. 133–140.

[31] G.V. Puskorius and L.A. Feldkamp, "Neural control of nonlinear dynamic systems with Kalman filter trained recurrent networks," *IEEE Transactions on Neural Networks*, **5** (1994).

[32] E.S. Plumer, "Training neural networks using sequential-update forms of the extended Kalman filter," Informal Report LA-UR-95-422, Los Alamos National Laboratory, January 1995.

[33] A.H. Sayed and T. Kailath, "A state-space approach to adaptive RLS filtering," *IEEE Signal Processing Magazine*, **11**(3), 18–60 (1994).

[34] S. Haykin, *Adaptive Filter Theory*, 3rd ed. Upper Saddle River, NJ: Prentice-Hall, 1996.

[35] R. Williams and D. Zipser, "A learning algorithm for continually running fully recurrent neural networks," *Neural Computation*, **1**, 270–280 (1989).

[36] F.L. Lewis, *Optimal Estimation*. New York: Wiley, 1986.

[37] R.K. Mehra, Identification of stochastic linear dynamic systems using Kalman filter representation. *AIAA Journal*, **9**, 28–31 (1971).

[38] B.D. Ripley, *Pattern Recognition and Neural Networks*. Cambridge University Press, 1996.

[39] T.M. Cover and J.A. Thomas, *Elements of Information Theory*. New York: Wiley, 1991.

[40] G.V. Puskorius and L.A. Feldkamp, "Extensions and enhancements of decoupled extended Kalman filter training, in *Proceedings of International Conference on Neural Networks, ICNN'97, IEEE, June 1997*, Vol. 3.

[41] N.N. Schraudolph, "Online local gain adaptation for multi-layer perceptrons," Technical report, IDSIA, Lugano, Switzerland, March 1998.

[42] P.J. Werbos, *Handbook of Intelligent Control*. New York: Van Nostrand Reinhold, 1992, pp. 283–356.

[43] FRED: Federal Reserve Economic Data. Available on the Internet at http://www.stls.frb.org/fred/. Accessed May 9, 2000.

[44] J. Moody, U. Levin, and S. Rehfuss, "Predicting the U.S. index of industrial production," *Neural Network World*, **3**, 791–794 (1993).

[45] J.H.L. Hansen, J.R. Deller, and J.G. Praokis, *Discrete-Time Processing of Speech Signals*. New York: Macmillan, 1993.

[46] E.A. Wan and A.T. Nelson, "Removal of noise from speech using the dual EKF algorithm," in *Proceedings of International Conference on Acoustics, Speech, and Signal Processing, IEEE, May 1998*.

[47] Center for Spoken Language Understanding, *Speech Enhancement Assessment Resource* (SpEAR). Available on the Internet at http://cslu.ece.ogi.edu/nsel/data/index.html. Accessed May 2000.

[48] Rice University, *Signal Processing Information Base* (SPIB). Available on the Internet at http://spib.ece.rice.edu/signal.html. Accessed September 15, 1999.

[49] J.H.L. Hansen and B.L. Pellom, "An effective quality evaluation protocol for speech enhancement algorithms," in *Proceedings of International Conference on Spoken Language Processing, ICSLP-98*, *Sidney, 1998*.